Integer points close to a transcendental curve and correctly-rounded evaluation of a function

Nicolas Brisebarre, Guillaume Hanrot

INTEGER POINTS CLOSE TO A TRANSCENDENTAL CURVE AND CORRECTLY-ROUNDED EVALUATION OF A FUNCTION

NICOLAS BRISEBARRE AND GUILLAUME HANROT

Abstract. Despite several significant advances over the last 30 years, guaranteeing the correctly rounded evaluation of elementary functions, such as $\cos, \exp, \sqrt[3]{\cdot}$ for instance, is still a difficult issue. This can be formulated as a Diophantine approximation problem, called the Table Maker's Dilemma, which reduces to determining points with integer coordinates that are close to a curve. In this article, we propose two algorithmic approaches to tackle this problem, closely related to a celebrated work by Bombieri and Pila and to the so-called Coppersmith's method. We establish the underlying theoretical foundations, prove the algorithms, study their complexity and present practical experiments; we also compare our approach with previously existing ones. In particular, our results show that the development of a correctly rounded mathematical library for the binary128 format is now possible at a much smaller cost than with previously existing approaches.

Contents

2020 *Mathematics Subject Classification.* Primary 11Y99, 41A05, 65G50; Secondary 11D75, 11J25.

This work was partly supported by the TaMaDi, FastRelax and NuSCAP projects of the French *Agence Nationale de la Recherche*.

1. Introduction

Modelling real numbers on a computer is by no means a trivial task. Until the mid-80s, processor manufacturers developed their own representations and conventions, leading to a difficult era – a time of weird, unexpected and dangerous behaviours [34]. This motivated the publication in 1984 of the IEEE-754 standard [17, 3], since then revised in 2008 [30, 49] and 2019 [29], for binary floating-point (FP) arithmetic[1], which remains the best trade-off for representing real numbers on a computer [52]. This put an end to this dangerous era of "numerical insecurity".

In particular, the IEEE-754 standard clearly specifies the formats of the FP 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.

The issue that we address in this paper is the problem of correctly-rounded evaluation of a one-variable function. Usually, when one wants to evaluate a function such as the cube root or the exponential functions, one actually evaluates a very good approximation of it (such as a polynomial for instance). This raises a problem, that is correct rounding: how can one guarantee that the rounding of the value of the function coincides with the rounding of the value of the approximation? This issue is related to a problem called Table's Maker Dilemma (TMD), which we shall describe in further detail in Section 1.2.

[1] and the radix independent IEEE-854 [16, 4] standard that followed

This paper presents two heuristic approaches to address the TMD. Both mix ingredients from approximation theory and algorithmic number theory (actually Euclidean lattice basis reduction). The first approach can be viewed as an effective variant of Bombieri and Pila's approach developed in [6]. The second one is an improvement over the algorithmic approaches developed in [63, 61]. Rather than reducing the problem for f to the same problem for an approximation (Taylor) polynomial for f as it is done in [63, 61], we work with the function f itself as long as possible. The difference may seem subtle, but it raises significant difficulties, while bringing two major improvements: smaller matrices and the prereduction trick (see Section 8).

In particular, we give the first significant results for the binary128 format and this work paves the way for the first development of an efficient correctly rounded mathematical library in the three fundamental formats binary32, binary64 and binary128. As of today, the library `CRlibm` [39] offers correctly rounded evaluation of the binary64 precision C99 standard elementary functions.

We believe that our results are interesting in themselves. In particular, beyond their application to the Table Maker's Dilemma for which we improve on some of the theoretical and practical results of [40, 42, 41, 63, 61], they offer a practical means to compute integer points in a strip around a transcendental analytic curve[2].

Note that we restrict ourselves in the present paper to transcendental function. Our methods, as we describe them, are bound to fail for algebraic functions of small degree. They may however be adapted in this case (similarly to Bombieri and Pila's adaptation in the algebraic case). We intend to come back to this in a sequel of this paper.

1.1. **Arithmetic framework.** We first recall the definition of a FP number.

Definition 1.1. Let $\beta, p, E_{\min}, E_{\max} \in \mathbb{Z}$, $\beta, p \geqslant 2, E_{\min} < 0 < E_{\max}$, a (normal) radix-$\beta$ FP number in precision p with exponent range $[E_{\min}, E_{\max}]$ is a number of the form

$$x = (-1)^s \frac{M}{\beta^{p-1}} \cdot \beta^E,$$

where :

- the *exponent* $E \in \mathbb{Z}$ is such that $E_{min} \leqslant E \leqslant E_{max}$,
- the *integral significand* $M \in \mathbb{N}$ represented in radix β satisfies $\beta^{p-1} \leqslant M \leqslant \beta^p - 1$,
- $s \in \{0, 1\}$ is the sign bit of x.

In the sequel, we shall leave the exponent range implicit unless it is explicitly required, and simply talk about "radix-β FP numbers in precision p".

Remark 1.2. For the sake of clarity, we chose not to mention subnormal FP numbers since they will not appear in the text. One can find the complete definition in [52, Chap. 2.1].

The number zero is a special case, cf. [52, Chap. 3], that we add to the set of radix-β and precision-p FP numbers. This yields a set denoted $\mathcal{F}_{\beta,p}$.

Remark 1.3. In this paper, we use radix 2 for the sake of clarity but our approach remains valid for any radix, in particular radix 10, the importance of which grows at a steady pace. The set $\mathcal{F}_{2,p}$ will thus be denoted $\mathcal{F}_p$.

[2]We mean here a representative curve of a transcendental analytic function.

	precision p	minimal exponent E_{min}	maximal exponent E_{max}
binary32	24	-126	127
binary64	53	-1022	1023
binary128	113	-16382	16383

TABLE 1. Main parameters of the three basic binary formats (up to 128 bits) specified by the standard [29].

Table 1 gives the main parameters of the three basic binary formats specified by IEEE 754-2019.

The result of an arithmetic operation whose input values belong to $\mathcal{F}_p$ may not belong to $\mathcal{F}_p$ (in general it does not). Hence that result must be rounded. The IEEE standard defines 5 different rounding functions; in the sequel, x is any real number to be rounded:

- round toward $+\infty$, or upwards: $\circ_u(x)$ is the smallest element of $\mathcal{F}_p$ that is greater than or equal to x;
- round toward $-\infty$, or downwards: $\circ_d(x)$ is the largest element of $\mathcal{F}_p$ that is less than or equal to x;
- round toward 0: $\circ_z(x)$ is equal to $\circ_u(x)$ if $x < 0$, and to $\circ_d(x)$ otherwise;
- round to nearest *ties to even*, denoted $\circ_{ne}(x)$ and round to nearest *ties to away*, denoted $\circ_{na}(x)$. If x is exactly halfway between two consecutive elements of $\mathcal{F}_p$, $\circ_{ne}(x)$ is the one for which the integral significand M is an even number and $\circ_{na}(x)$ is the one for which the integral significand M is largest. Otherwise, both return the element of $\mathcal{F}_p$ that is the closest to x.

The first three rounding functions are called directed rounding functions.

The following real numbers will play a key role in the problem that we address.

Definition 1.4. A rounding breakpoint (or simply, a breakpoint) is a point where the rounding function changes (namely a discontinuity point). For round-to-nearest functions, the rounding breakpoints are the exact middles of consecutive floating-point numbers. For the other rounding functions, they are the floating-point numbers themselves.

1.2. **Correct rounding, Table Maker's Dilemma.** The standard requires that the user should be able to choose one rounding function among these ones, called the *active rounding function*. An active rounding function 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*.

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.

While the IEEE 754-1985 and 854-1987 standards required correctly rounded arithmetic operations and square root, they did not do it for the most common

mathematical functions, such as simple algebraic[3] functions like $1/\sqrt{\cdot}$, $\sqrt[3]{\cdot}$, ... and also a few transcendental[4] 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[5]. A subset of these functions is usually available from the `libm`s delivered with compilers or operating systems.

This lack of requirement is mainly due to a difficult problem known as the Table Maker's Dilemma (TMD), a term coined by Kahan. When evaluating most elementary functions, one has to compute an approximation to the exact result, using an intermediate precision somewhat larger than the "target" precision p. The TMD is the problem of determining, given a function f, what this intermediate precision should be in order to make sure that rounding that approximation yields the same result as rounding the exact result. Ideally, we aim at getting the minimal such precision $\mathrm{htr}_f(p)$, that we call *hardness to round* of f (see Definition 2.3).

If we have N FP numbers in the domain being considered, it is expected that $\mathrm{htr}_f(p)$ is of the order of $p+\log_2(N)$ (hence $2p$ for most usual functions and binades[6]). This is supported by a probabilistic heuristic approach that is presented in detail in [52, 51]. It has been studied in [10] where O. Robert and the authors of the present paper gave, under some mild hypothesis on f'', solid theoretical foundations to some instances of this probabilistic heuristic, targeting in particular the cases that the `CRlibm` library uses in practice.

1.3. **Fast and cheap correctly-rounded function evaluation in binary64.** Diophantine approximation-type methods yield – not fully satisfactory — upper bounds for $\mathrm{htr}_f(p)$ for algebraic functions: the precision to which the computations must be performed is, in general, grossly overestimated [31, 38, 11]. On the other hand, regarding transcendental functions, either no theoretical statement exists or they provide results that are off by such a margin that they cannot be used in practical computations [53, 36, 35].

Therefore, algorithmic approaches to the TMD [40, 42, 41, 63, 27] had to be developed. They allowed for solving the TMD for the IEEE binary64 format (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), $\sin(x)$, $\cos(x)$, $\tan(x)$, $\arcsin(x)$, $\arccos(x)$, $\arctan(x)$, $\arctan(y/x)$, $\sin(\pi x)$, $\cos(\pi x)$, $\tan(\pi x)$, $\arcsin(x)/\pi$, $\arccos(x)/\pi$, $\arctan(x)/\pi$, $\arctan(y/x)/\pi$, $\sinh(x)$, $\cosh(x)$, $\tanh(x)$, $\sinh^{-1}(x)$, $\cosh^{-1}(x)$, $\tanh^{-1}(x)$.

Thanks to these results, it is now possible to obtain correct rounding in binary64 in two steps only (inspired by a strategy developed by A. Ziv [72] and implemented

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

[4]A function is transcendental if it is not algebraic.

[5]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 [12, Def. 5.1.4].

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

in the `libultim` library[7]), which one may then optimize separately. This is the approach used in `CRlibm`:

- the first quick step is as fast as a current `libm`, and provides a relative 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. [43, 42, 62, 63, 61]. In particular, there is no need for arbitrary multiple precision anymore.

This approach [21, 22] leads to correctly-rounded function evaluation routines that are fast and have a reasonable memory consumption. Unfortunately, the lack of useful information about the TMD in binary128 has so far prevented the development of an extension of `CRlibm` to this format.

1.4. **Goal and outline of the paper.** In this paper, we present two new algorithmic approaches to tackle the TMD. For both, we follow the standard strategy to subdivide the interval under study into subintervals; but instead of approximating the function f by a polynomial function using Taylor expansion at the center of such a tiny interval I, as it was done in [40, 42, 41, 63, 61], we approximate f by an algebraic function using uniform approximation: if we assume for instance $f : [1/2, 1) \to [1/2, 1)$ (hence every involved FP number has denominator 2^p),we search for P_0 and $P_1 \in \mathbb{Z}[X, Y]$ that are small on the "weighted" curve $(2^p x, 2^p f(x))$ (first approach) or in a strip around this "weighted" curve (second approach). This smallness implies that the bad cases for rounding are common roots to P_0 and P_1. Then, we use a heuristic argument of coprimality of P_0 and P_1, analogous to the one used in [7, 63, 61] to obtain these bad cases.

In order to compute P_0 and P_1, we use ideas and techniques developed by the first author and S. Chevillard [9]. Very roughly speaking, if we still assume $f : [1/2, 1) \to [1/2, 1)$, the key idea is to find $P \in \mathbb{Z}[X_1, X_2]$ that is small at some points $(2^p x_i, 2^p f(x_i))$ of the "weighted" curve (first approach) or $(2^p x_i, 2^p (f(x_i)+y_i))$ of a strip around the "weighted" curve (second approach). If the points x_i (resp. the pairs (x_i, y_i)) are carefully chosen, these discrete smallness constraints imply uniform smallness over the curve, resp. the strip around the curve, cf. Section 4.1. The discrete constraints can be reformulated as the fact that the values of P at the $(2^p x_i, 2^p f(x_i))$ (resp. the $(2^p x_i, 2^p (f(x_i) + y_i))$) are the coordinates of a certain short vector in a Euclidean lattice. The celebrated LLL algorithm, cf. Section 4.2, then allows for computing a reasonable candidate for P.

The first approach which favours smallness on the curve $(2^p x, 2^p f(x)), x \in I$ is somehow akin to [6] whereas the second one, which forces smallness on a strip around this curve, is somehow analogous to [63, 61].

As our reader will see, our work does not lead, with respect to the previous algorithmic approaches, to an improvement for the determination of worst cases for rounding and optimal values of $\mathrm{htr}_f(p)$. On the other hand, for certain elementary or special functions evaluated in binary128, we are able to provide:

- upper bounds for $\mathrm{htr}_f(p)$ that are useful in practice. For instance, for the exponential function, which plays a central role for correctly-rounded evaluation of the elementary functions of the C mathematical library, we

[7] `libultim` was released by IBM.

provide a roadmap to reach, in practice[8], the upper bound $\mathrm{htr}_f(p) \leqslant 12p$ for $p = 113$ which corresponds to the binary128 format;
- an effective determination of the FP values whose evaluation by f is exactly an FP number or the middle of two consecutive FP numbers. This is a key issue in the development of correctly-rounded evaluation routines. An exhaustive evaluation is possible in binary32 and theoretical results [32] yield lists of these values for some restricted classes of functions and algorithmic approaches [43, 42, 62, 63, 61] make it possible to address this problem in binary64. However, before the present paper, the only practical means to tackle the binary128 format was S. Torres' implementation of [61] in his PhD thesis [66] and we will show in Section 8 that our work significantly improves the situation. As an example, we address the case of the Euler function Γ, a function of the C mathematical library, for which theoretical results from Transcendental Number Theory are almost nonexistent.

This hopefully paves the way to an extension of `CRlibm` to the binary128 format, provided that we adopt the following three step strategy for this format:

- a first quick step identical to the one mentioned in the previous subsection;
- a second step, slower but with a reasonable bounded execution time, where the evaluation is performed using a precision of $260 = 2p + 34$ bits, say. Heuristically, this should cover all the hardest-to-round cases;
- a third step, where the evaluation is performed using a precision of $12p = 1356$ bits. Heuristically, this step should never be called, so it is important to write routines simple enough to be formally proved in order to guarantee its validity.

We will formalize the problem we address in Section 2. We then give a state of the art in Section 3. The theoretical results are presented in Section 3.1, including applications of [36, 35] that, to the best of our knowledge, are reviewed for the first time and offer theoretical upper bounds for $\mathrm{htr}_f(p)$ in the binary64 and binary128 cases which greatly improve upon the existing ones. The existing algorithmic approaches are sketched in Section 3.2. Our approach relies on tools from Approximation Theory and Euclidean lattice basis reduction and an idea presented in [9, 15]. We recall them in Section 4. Our first approach is presented in Section 5 and our second one in Section 6. We present a comparison with previous work in Section 7 and we conclude with experimental results in Section 8.

2. Formalization of the problem

Assume we wish to correctly round a real-valued function φ. Note that if x is a bad case for φ (i.e., $\varphi(x)$ is difficult to round), 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 $\varphi(x) \geqslant 0$.

We consider that all input values are elements of $\mathcal{F}_p \cap [2^{e_1}, 2^{e_1+1})$. The method must be applied for each possible integer value of e_1.

If the values of $\varphi(x)$, for $x \in [2^{e_1}, 2^{e_1+1})$, are not all included in the binade $[2^{e_2}, 2^{e_2+1})$, we split the input interval into subintervals such that for each subinterval, there is an 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 restrict to one of those subintervals I included in $[2^{e_1}, 2^{e_1+1})$.

[8] A few days for a binade, see Section 8.

For directed rounding functions, the problem to be solved is the following:

Problem 2.1 (TMD, directed rounding functions)**.** *What is the minimum $\mu(p) \in \mathbb{Z}$ such that, for $2^{p-1} \leqslant X \leqslant 2^p - 1$ (and, possibly, the restrictions implied by $X/2^{-e_1+p-1} \in I$) such that $\varphi\left(X2^{e_1-p+1}\right) \notin \mathcal{F}_p$ and for $2^{p-1} \leqslant Y \leqslant 2^p - 1$, we have*

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

For rounding to nearest functions, the problem to be solved is the following:

Problem 2.2 (TMD, rounding to nearest functions)**.** *What is the minimum $\mu(p) \in \mathbb{Z}$ such that, for $2^{p-1} \leqslant X \leqslant 2^p - 1$ (and, possibly, the restrictions implied by $X/2^{-e_1+p-1} \in I$) such that $\varphi\left(X2^{e_1-p+1}\right)$ is not the middle of two consecutive elements of $\mathcal{F}_p$ and for $2^{p-1} \leqslant Y \leqslant 2^p - 1$, we have*

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

These statements lead to the following definition.

Definition 2.3 (hardness to round)**.** Let a precision p be given, $\circ$ be a rounding function and φ be a real valued function. Let x be a FP number in precision p and $e_2 \in \mathbb{Z}$ be the unique integer such that $\varphi(x) \in [2^{e_2}, 2^{e_2+1})$ (here again, we assume x and $\varphi(x) \geqslant 0$, since the extension to the other cases is straightforward).

The hardness to round $\varphi(x)$, denoted $\mathrm{htr}_{\varphi,\{x\},\circ}(p)$ is equal to:

- $-\infty$ if $\varphi(x)$ is a breakpoint;
- the smallest integer m such that the distance of $\varphi(x)$ to the nearest breakpoint is larger than or equal to 2^{-m+e_2}.

The hardness to round φ over an interval I, denoted $\mathrm{htr}_{\varphi,I,\circ}(p)$, is then the maximum of the hardness to round $\varphi(x)$ for all FP $x \in \mathcal{F}_p \cap I$, while the hardness to round φ is the hardness to round φ over $\mathbb{R}$, simply denoted $\mathrm{htr}_{\varphi,\circ}(p)$. When there is no ambiguity over the rounding function, we get rid of the symbol $\circ$.

Remark 2.4. Note that both Problem 2.1 and Problem 2.2 for precision p are subproblems of Problem 2.1 for precision $p+1$.

Remark 2.5. If we assume that φ admits an inverse φ^{-1} and is differentiable over I and that we have a precise control over the image of φ' over I, it follows from the mean value theorem that addressing Problems 2.1 and 2.2 for φ over I is analogous to addressing Problems 2.1 and 2.2 for φ^{-1} over $\varphi(I)$. For instance, one can think of exp and log or $x \mapsto \sqrt[3]{x}$ and $x \mapsto x^3$.

The problem that we actually tackle in this paper is the following.

Problem 2.6. *Let $a, b \in \mathbb{R}$, $a < b$, $f : [a, b] \to \mathbb{R}$ be a transcendental function analytic in a (complex) neighbourhood of $[a, b]$. Let $u, v, w \in \mathbb{N} \setminus \{0\}$, determine the integers X, $a \leqslant X/u \leqslant b$ for which there exists $Y \in \mathbb{Z}$ satisfying*

$$\left|f\left(\frac{X}{u}\right) - \frac{Y}{v}\right| < \frac{1}{w}. \tag{2.1}$$

This problem encompasses the TMD: consider $a = 2^{e_1}, b = 2^{e_1+1} - 1, u = 2^{p-e_1-1}, v = 2^{p-e_2-1}$ and $f = \varphi$ (for directed rounding functions) or $f = \varphi - 1/(2v)$ (for rounding to nearest functions).

It also includes a generalization of the question addressed in [6], which corresponds to the case $a = 0, b = 1, u = v$.

Remark 2.7. Note that the status of w in Problem 2.6 may vary. In Section 5, we'll consider w as an *output* of the algorithm: on input u, v, a, b, Algorithm 2 heuristically returns a value of w and a set (X, Y) of solutions of (2.1). This comes from the fact that Algorithm 2 is primarily devoted to finding solutions to $Y/v = f(X/u)$, and that it happens that from the work done *on* this curve, one can deduce information *close to* the curve. A parameter ω_0 gives some influence on w, but no complete control.

On the other hand, in Section 6, w will be an *input* of the problem: on input u, v, w, a, b, Algorithm 4 heuristically returns the set (X, Y) of solutions of (2.1).

3. State of the art

In this section, we review the state of the art on the TMD; we shall start by discussing results which can be derived from previous estimates in Diophantine approximation, then shall account on the algorithmic approaches which have been developed since the late 90s. One can find a more complete (but slightly outdated since the results of [36, 35] on the exponential function are not considered) state of the art in [52, Chap. 12].

3.1. Diophantine approximation results: the exponential and the logarithm functions.

In this section, we use the conventions and notations that we introduced in Section 2.

The exponential function is central in the study of correctly-rounded evaluation of the elementary functions of `libms`: a relevant information on its hardness to round yields relevant information as well on trigonometric and hyperbolic functions, and their respective reciprocals, see [52, §12.4.4] and Remark 2.5, the logarithm function and inverse trigonometric functions.

Following the works [53] and [36], Khémira and Voutier proved in [35] a lower bound (called transcendence measure) for the expression $\left|e^{\beta} - \alpha\right|$, where α and β are algebraic numbers, $\beta \neq 0$. When specialized in FP numbers, their result provides interesting upper bounds for $\mathrm{htr}_{\exp}(p)$.

Let m and $n \in \mathbb{N}$, we put

$$d_n = \text{l.c.m.}(1, \dots, n) \text{ and } D_{m,n} = \frac{m!}{\prod\limits_{\substack{q \leqslant n, \\ q \text{ prime}}} q^{v_q(m!)}},$$

where $v_q(m!)$ is the q-adic valuation of $m!$. We can now state Khémira and Voutier's Theorem in the particular case where α is a FP number and β is a FP number (directed rounded functions) or the middle of two consecutive FP numbers (round-to-nearest functions). As mentioned above, we can get a similar result for the logarithm function.

Theorem 3.1 (Khemira and Voutier [35], specialized here to FP numbers, directed rounded functions)**.** *Let a precision p be given, let $x \neq 0$ and $y \in \mathcal{F}_p$ such that y and e^x are in the same binade. We denote by e_x, resp. e_y, the exponent of x, resp. y. We have $e_y = \lfloor \log_2(\exp(x)) \rfloor = \lfloor x/\log(2) \rfloor$. Let K and $L \in \mathbb{N} \setminus \{0\}$, $K \geqslant 2$,*[9]

[9]The condition $K \geqslant 2$ is not stated in [35] but it is actually necessary to have Inequality (3.1) satisfied here.

$L \geqslant 2$ and $E \in (1, +\infty)$ which satisfy

$$\begin{aligned} KL\log E \geqslant\ & KL\log 2 + (K-1)(1+\log(\sqrt{3L}d_{L-1})) + \log(D_{K-1,L-1}) \\ & + (1+2\log 2)(L-1) + \log\left(\min\left(d_{L-2}^{K-1}, (L-2)!\right)\right) + \log((K-1)!) \\ & + (K-1)\max(p-2-e_x, -1)\log 2 + LE|x| + L\log E \\ & + (L-1)\max(p-2-e_y, -1)\log 2. \end{aligned} \tag{3.1}$$

Then we have $|e^x - y| \geqslant E^{-KL}$.

Remark 3.2. For the round-to-nearest functions, we assume that y is the middle of two consecutive FP numbers and that the numbers y and e^x are in the same binade. If we denote again $e_y = \lfloor x/\log(2) \rfloor$, the conclusion of the theorem remains valid if we replace $\max(p-2-e_y, -1)$ with $\max(p-1-e_y, -1)$ in the last line of (3.1).

For instance, using the following sets of parameters, we are able to compute the following upper bounds for the hardness to round exp on $[1/4, 1/2)$:

- In binary64 ($p = 53$), the triple $(K, L, E) = (61, 29, 81.29...)$ yields $\mathrm{htr}_{\exp,[1/4,1/2)}(53) \leqslant 11225 \sim 211p$.
- In double extended precision ($p = 64$), the triple $(K, L, E) = (62, 37, 82.62...)$ yields $\mathrm{htr}_{\exp,[1/4,1/2)}(64) \leqslant 14610 \sim 228p$.
- In binary128 ($p = 113$), the triple $(K, L, E) = (84, 59, 109.44...)$ yields $\mathrm{htr}_{\exp,[1/4,1/2)}(113) \leqslant 33573 \sim 297p$.

3.2. **Algorithmic approaches.** In view of the lack of practicality (or in order to improve on it) of fundamental results discussed in the previous subsection, algorithmic approaches have been developed and used in an extensive way since the late 90s.

A first straightforward idea consists in testing all possible FP values x; for each value of x one computes a sufficiently accurate interval approximation to $f(x)$ and determines the hardness to round $f(x)$. The cost of the approach is obviously proportional to the number of different FP numbers of the format under study i.e., $2^{p+E_{\max}-E_{\min}}$, which makes it basically tractable for binary32 [60]. In [23], the exhaustive evaluations are performed on an FPGA using a tabulated difference approach, which makes it possible to address the binary64 case. Currently, the double extended or binary128 formats seem completely out of reach of such approaches.

More subtle ideas proceed by splitting the domain into subintervals and replacing the function (assumed to be sufficiently smooth) by a polynomial, in practice a Taylor approximation, over the interval under study; one is then reduced to study the problem in the polynomial case.

Lefèvre, together with Muller [40, 42, 41], studied the degree 1 case; in this case, the remaining Diophantine problem is to find two integers x, y, $|x| \leqslant X, |y| \leqslant Y$ such that $|\alpha x + \beta - y|$ is minimal, which is solved by elementary Diophantine arguments, either the three distance theorem, or continued fractions (see e.g., [5]). These ideas lead to an algorithm of complexity $\tilde{O}(2^{2p/3})$ for floating-point numbers of precision p, as $p \to \infty$, which computes all worst cases for rounding in the domains under consideration.

Highly optimized and parallel implementations of this method have proved invaluable to find optimal values of $\mathrm{htr}_f(53)$ for several functions of the standard `libm`. This was a key step towards the development of `CRlibm`.

Higher degree approximations give rise to more complicated Diophantine problems. Stehlé, Lefèvre and Zimmermann [63], further refined by Stehlé [61], make use of a technique due to Coppersmith [19, 20] and based on lattice basis reduction to solve it. We recall Corollaries 4 & 5 of [61], adapted to our context.

Theorem 3.3 (Stehlé [61])**.** *For all $\varepsilon > 0$, there exists a heuristic algorithm of complexity $2^{p(1+\varepsilon)/2}$ which, given a function f, returns all FP numbers $x \in [1/2, 1)$ of precision p such that the hardness to round $f(x)$ is $\geqslant 2p$.*

There exists a polynomial-time heuristic algorithm which returns all FP numbers $x \in [1/2, 1)$ of precision p such that the hardness to round $f(x)$ is $\geqslant 4p^2$; the latter works by reducing a lattice of dimension $O(p^2)$ of $\mathbb{R}^m$ for some $m = O(p^4)$.

The heuristic character of the algorithm is rather mild (i.e., the algorithm works in practice as expected on almost all inputs).

We use a somewhat different approach: the algorithmic content of our method remains based on lattice basis reduction, but rather than reducing the problem to a polynomial problem, we keep the problem linked to the function, which we shall make possible thanks to rigorous uniform approximation techniques based on Chebyshev interpolation. In order to develop our approach, we thus now need to give a short survey of Chebyshev interpolation/approximation and lattice basis reduction.

4. A quick overview of uniform approximation and lattice basis reduction

We shall require some tools from uniform approximation theory [24, 59, 58, 14, 8, 50, 67] and algorithmic geometry of numbers [48, 26, 18, 13, 55, 69].

4.1. Relation between uniform approximation and interpolation.

Let $n \in \mathbb{N}$, the n-th Chebyshev polynomial of the first kind is defined by $T_n \in \mathbb{R}_n[x]$ and $T_n(\cos t) = \cos(nt)$ for all $t \in [0, \pi]$. The T_n's can also be defined by

$$T_0(x) = 1, T_1(x) = x, T_{n+2}(x) = 2xT_{n+1}(x) - T_n(x), \forall n \in \mathbb{N}.$$

4.1.1. *Interpolation at the Chebyshev nodes.*

The zeros of T_{n+1} are

$$\mu_{k,n} = \cos\left(\frac{(n-k+1/2)\pi}{n+1}\right), k = 0, \ldots, n.$$

They are called $(n+1)$-Chebyshev nodes of the first kind. Polynomials interpolating functions at this family give rise to very good uniform approximations over $[-1, 1]$ to these functions [50, 67]. To be able to work on an interval $[a, b]$, we will need scaled versions of Chebyshev polynomials and nodes. We then define, for $n \in \mathbb{N}$,

$$T_{n,[a,b]} := T_n\left(\frac{2x-b-a}{b-a}\right), \mu_{k,n,[a,b]} := \frac{(b-a)\mu_{k,n}+a+b}{2}, k = 0, \ldots, n. \tag{4.1}$$

Here again, when there is no ambiguity, we denote the nodes as $\mu_{k,[a,b]}$. Note that $T_{n,[a,b]}\left(\mu_{k,n,[a,b]}\right) = T_n(\mu_{k,n})$.

Let $N \geqslant 1$, let f be a function defined over $[a, b]$, if we interpolate f by a polynomial in $\mathbb{R}_{N-1}[x]$ at the scaled Chebyshev nodes of the first kind, we have the

following expressions for the interpolation polynomial P [50, Chap. 6]:

$$\begin{aligned} P(x) &= {\sum_{0\leqslant k\leqslant N-1}}' c_k T_{k,[a,b]}(x) \in \mathbb{R}_{N-1}[x] \text{ with} \\ c_k &= \frac{2}{N} \sum_{0\leqslant \ell\leqslant N-1} f(\mu_{\ell,N-1,[a,b]}) T_{k,[a,b]}(\mu_{\ell,N-1,[a,b]}) \text{ for } k=0,\ldots,N-1, \\ &= \frac{2}{N} \sum_{0\leqslant \ell\leqslant N-1} f(\mu_{\ell,N-1,[a,b]}) T_k(\mu_{\ell,N-1}). \end{aligned}$$

The symbol ${\sum}'$ means that the first coefficient has to be halved. Note that, if we introduce $\widehat{f} : z \in [-1,1] \mapsto f\left(z\frac{b-a}{2} + \frac{a+b}{2}\right)$, the coefficients c_k are also the coefficients of the interpolation polynomial in $\mathbb{R}_{N-1}[x]$ of $\widehat{f}$ at the Chebyshev nodes of the first kind.

4.1.2. *Uniform approximation using interpolation polynomials.* Let $\rho > 1$, if $a < b$ are two real numbers, we define the ellipse

$$\mathcal{E}_{\rho,a,b} = \left\{ \frac{b-a}{2}\frac{\rho e^{i\theta} + \rho^{-1}e^{-i\theta}}{2} + \frac{a+b}{2}, \theta \in [0,2\pi] \right\}$$

and let $E_{\rho,a,b}$ be the closed region bounded by the ellipse $\mathcal{E}_{\rho,a,b}$. For f a function analytic in a neighbourhood of $E_{\rho,a,b}$, we define $M_{\rho,a,b}(f) = \max_{z\in\mathcal{E}_{\rho,a,b}} |f(z)|$.

Let $N \in \mathbb{N}$, $N \geqslant 1$, we also define

$$\eta_{\rho,0} = 1 \text{ and } \eta_{\rho,k} = \frac{\rho^2+1}{\rho^2-1} \text{ for } k = 1,\ldots,N-1.$$

The following two propositions establish Cauchy's inequalities for interpolation polynomials at (scaled) Chebyshev nodes.

Proposition 4.1. *Let $\rho > 1$, $a < b$, let $N \in \mathbb{N}$, $N \geqslant 1$, f be a function analytic in a neighbourhood of $E_{\rho,a,b}$, the coefficients $c_k, k = 0,\ldots,N-1$, of the interpolation polynomial p_{N-1} of f at the (scaled) Chebyshev nodes of the first kind over $[a,b]$, $(\mu_{k,N-1,[a,b]})_{0\leqslant k\leqslant N-1}$ satisfy*

$$|c_k| \leqslant 2\frac{M_{\rho,a,b}(f)}{\rho^k}\eta_{\rho,k} \text{ for } k = 0,\ldots,N-1,$$

Moreover, we have

$$\|f - p_{N-1}\|_{\infty,[a,b]} \leqslant \frac{4M_{\rho,a,b}(f)}{\rho^{N-1}(\rho-1)}.$$

Proof. See Appendix A. □

Let $\rho_1,\rho_2 > 1$, $a_1 < b_1$, $a_2 < b_2$, we define $\mathcal{E}_{\rho_1,a_1,b_1,\rho_2,a_2,b_2} = \mathcal{E}_{\rho_1,a_1,b_1} \times \mathcal{E}_{\rho_2,a_2,b_2}$ and $E_{\rho_1,a_1,b_1,\rho_2,a_2,b_2} = E_{\rho_1,a_1,b_1} \times E_{\rho_2,a_2,b_2}$.

Proposition 4.2. *Let $\rho_1,\rho_2 > 1$, $a_1 < b_1$, $a_2 < b_2$, let $M_1, M_2 \in \mathbb{N}, M_1, M_2 \geqslant 2$, f be a function analytic in a neighbourhood of $E_{\rho_1,a_1,b_1,\rho_2,a_2,b_2}$, the coefficients $c_{k_1,k_2}, k_1 = 0,\ldots,M_1-1, k_2 = 0,\ldots,M_2-1$ of the interpolation polynomial P_{M_1-1,M_2-1} of f at pairs of Chebyshev nodes of the first kind satisfy, for $k_1 = 0,\ldots,M_1-1$, $k_2 = 0,\ldots,M_2-1$,*

$$|c_{k_1,k_2}| \leqslant 4\frac{M_{\rho_1,a_1,b_1,\rho_2,a_2,b_2}(f)}{\rho_1^{k_1}\rho_2^{k_2}}\eta_{\rho_1,k_1}\eta_{\rho_2,k_2},$$

where $M_{\rho_1,a_1,b_1,\rho_2,a_2,b_2}(f) = \max_{z\in\mathcal{E}_{\rho_1,a_1,b_1,\rho_2,a_2,b_2}} |f(z)|$. *Moreover, we have*

$$\|f - P_{M_1-1,M_2-1}\|_{\infty,[a_1,b_1]\times[a_2,b_2]} \leqslant \frac{16\rho_1\rho_2 M_{\rho_1,\rho_2}(f)}{(\rho_1-1)(\rho_2-1)}\left(\frac{1}{\rho_1^{M_1}}+\frac{1}{\rho_2^{M_2}}\right).$$

Proof. See Appendix A. □

4.2. **A quick reminder on Euclidean lattices and the LLL algorithm.** In this subsection, we shortly review basic facts concerning lattices and lattice basis reduction algorithms.

Definition 4.3. Let $M \in \mathbb{N}$, $M \geqslant 1$, a lattice of $\mathbb{R}^M$ is a discrete subgroup of $\mathbb{R}^M$; equivalently, a lattice $L \subset \mathbb{R}^M$ is the set of integer linear combinations of a family $(b_1,\dots,b_N)$ of $\mathbb{R}$-linearly independent vectors of $\mathbb{R}^M$. We shall then say that $(b_i)_{1\leqslant i\leqslant N}$ is a basis of L, and that $N \leqslant M$ is the dimension (or the rank) of L.

Proposition 4.4. *The sets* $B = (b_i)_{1\leqslant i\leqslant N}$, $C = (c_i)_{1\leqslant i\leqslant N}$ *are two bases of the same lattice, given in (row) matrix form if and only if there exists* $U \in \mathcal{M}_N(\mathbb{Z})$, $\det U \in \{\pm 1\}$, *such that* $C = UB$. *As a consequence, the quantity* $(\det CC^t)^{1/2} = (\det BB^t)^{1/2}$ *is independent of the basis and is associated to the lattice itself – we shall call it the* volume *of the lattice and denote it by* $\operatorname{vol} L$.

Given a basis $(b_1,\dots,b_N)$ of L as input, finding a shortest nonzero vector in L is called the *shortest vector problem*. The decision version of this problem has been shown [1] to be hard under randomized reductions; in practice, one thus has to content oneself with approximation algorithms, such as the LLL algorithm [44]:

Theorem 4.5 (Lenstra, Lenstra, Lovász, 1982)**.** *The LLL algorithm, given* N $\mathbb{R}$*-linearly independent vectors* $(b_1,\dots,b_N) \in \mathbb{Z}^M$ *, returns a basis* $(c_1,\dots,c_N)$ *such that* $\|c_1\|_2 \leqslant 2^{(N-1)/4}(\operatorname{vol} L)^{1/N}$, *and* $\|c_1\|_2 \leqslant \left(2^{(N-1)/4}\right)^2 \min_{x\in L-\{0\}}\|x\|_2$. *One also has* $\|c_2\|_2 \leqslant 2^{(N-1)/4}(\operatorname{vol} L)^{1/(N-1)}$.

The time complexity of the LLL algorithm is polynomial in the maximal bit-length of the coefficients of the b_i*'s, the lattice rank* N*, and the space dimension* M.

Proof. See Theorems 9 and 10 from [55, Chap. 2], except for the last inequality on $\|c_2\|$ which is a consequence of the proof of Fact 3.3 in [7]. □

The constant 2 in the terms $2^{(N-1)/4}$ of the inequalities of the theorem is arbitrary, and could be replaced by any real number $> 4/3$.

We now discuss shortly an improvement due to Akhavi & Stehlé [2] to the LLL algorithm in the case where N is much smaller than M, the dimension of the ambient space. Let A be an $N \times M$ matrix, the rows of which generate the lattice L; the idea is to reduce a smaller $N \times N$ matrix obtained by a random projection (i.e., multiplying A on the right by a random $M \times N$ matrix), and apply the same transformation to the original matrix.

Theorem 4.6 (Akhavi & Stehlé, 2008)**.** *For all* N*, there is an* $n_0(N)$ *such that for* $M \geqslant n_0(N)$*, if* P *is an* $M \times N$ *matrix whose columns are independent random vectors picked up uniformly independently inside the* M*-th dimensional unit ball, and* $A' = \operatorname{LLL}(A \cdot P)$*; then, with probability* $\geqslant 1 - 2^{-N}$*, the first column vector of the matrix* $A'(A\cdot P)^{-1}A$ *is a vector of* L *of norm* $\leqslant 2^{4N}(\operatorname{vol} L)^{1/N}$.

S. Torres [66] showed that this idea indeed improves the practical outcome of [61]. As for us, cf. Section 8, we noticed that this idea works even for simpler models

for the random matrix P such as random, uniform $\{0, \pm 1\}$ coefficients, which give equally good results.

5. The one-variable method, à la Liouville

Let $u, v \in \mathbb{N} \setminus \{0\}$, $a < b$ be two real numbers and $f : [a, b] \to \mathbb{R}$. The starting point of this approach to Problem 2.6 follows a simple (but fundamental!) idea due to J. Liouville [45, 46, 47], which we now recall.

For any $P \in \mathbb{Z}[X_1, X_2]$, for any $x \in [a, b]$, $y \in [f(x) - 1/v, f(x) + 1/v]$, we know, from the mean value theorem, that there exists z between $vf(x)$ and vy (hence $z \in [vf(x) - 1, vf(x) + 1]$), such that

$$P(ux, vf(x)) - P(ux, vy) = v(f(x) - y)\frac{\partial P}{\partial y}(ux, z). \tag{5.1}$$

Then we compute, by combining Chebyshev interpolation and lattice reduction, two polynomials $P_0, P_1 \in \mathbb{Z}[X_1, X_2]$ such that, for $i = 0, 1$, for all $x \in [a, b]$, $z \in [vf(x) - 1, f(x) + 1)], |P_i(ux, vf(x))| < 1/2$, while $\left|\frac{\partial P_i}{\partial y}(ux, z)\right|$ is bounded by "not too large" an M. Let $x_0 = X/u \in [a, b], X \in \mathbb{Z}$, $y_0 = Y/v, Y \in \mathbb{Z}$, we have $P_i(ux_0, vy_0) = P_i(X, Y) \in \mathbb{Z}$. If $P_i(X, Y)$ is non zero for some i, say $i = 0$, then it follows from (5.1)

$$\underbrace{|P_0(X, Y)|}_{\geqslant 1} - \underbrace{|P_0(X, vf(X/u))|}_{<1/2} \leqslant v|f(X/u) - Y/v| \underbrace{\left|\frac{\partial P_0}{\partial y}(ux_0, z_0)\right|}_{\leqslant M},$$

hence $|vf(X/u) - Y| > 1/(2M)$. Otherwise, we have $P_0(X, Y) = P_1(X, Y) = 0$. We now use our heuristic assumption, that is P_0 and P_1 have no nonconstant common factor: we then perform elimination of one of the variables and retrieve the list of all the bad cases, i.e., the X such that $|vf(X/u) - Y| \leqslant 1/(2M)$. In the sequel of this section, we give all the details of this approach: we first give estimates of the determinants of the lattices that we use, we present our algorithm, the proof of its correctness and analyse its complexity.

5.1. Volume estimates for rigorous interpolants at the Chebyshev nodes.

Let $N \geqslant 2$, for $i = 0, \ldots, N - 1$, let f_i be a function defined over $[a, b]$ and Q_i be its interpolation polynomials in $\mathbb{R}_{N-1}[x]$ at Chebyshev nodes of the first kind. We shall use the following results for the functions f_i defined in (5.5).

Let DCT-II denote the discrete cosine transform of type 2:

$$\begin{aligned}&\text{DCT-II} : \mathbb{R}^N \to \mathbb{R}^N\\ &(x_0, \ldots, x_{N-1}) \mapsto (X_0, \ldots, X_{N-1}) \text{ with}\\ &\qquad X_k = \sum_{0 \leqslant \ell \leqslant N-1} x_\ell \cos\left(\frac{k(\ell + 1/2)\pi}{N}\right), \text{ for } k = 0, \ldots, N - 1.\end{aligned}$$

This function is often introduced with slightly different normalisations [64, 57] and we can take advantage of fast algorithms [57, §6.3] that make possible to compute it in at most $\mathcal{O}(N \log N)$ operations for a fixed and given precision. Recall, cf.

Section 4.1.1, that for $i = 0, \dots, N-1$,

$$Q_i(x) = {\sum_{0 \leqslant k \leqslant N-1}}' c_{k,i} T_{k,[a,b]}(x) \in \mathbb{R}_{N-1}[x] \text{ with}$$

$$(c_{0,i}, \dots, c_{N-1,i}) = \frac{2}{N} \text{DCT-II}(f_i(\mu_{N-1,[a,b]}), \dots, f_i(\mu_{0,[a,b]})). \tag{5.2}$$

Let us introduce two real parameters $\rho > 1$ and $\omega_0 \geqslant 0$ (to be chosen later on).

We now assume that all the f_i's are analytic in a neighbourhood of $E_{\rho,a,b}$. For $i = 0, \dots, N-1$, let $R_i = 4M_{\rho,a,b}(f_i)/(\rho^{N-1}(\rho-1))$. We have, by Proposition 4.1, $\|f_i - Q_i\|_{\infty,[a,b]} \leqslant R_i$. For B, C two complex matrices with the same number of rows r and respectively m and n columns, we shall denote by $(B|C)$ the $r \times (m+n)$ matrix obtained by concatenating these two matrices. If δ_{ij} denotes the Kronecker delta, we introduce the $N \times 2N$ matrix $A = (A_1|A_2)$, where

$$A_1 = \left(c_{j,i}/2^{\delta_{j0}}\right)_{0 \leqslant i,j \leqslant N-1}, \quad A_2 = (\delta_{ij}\rho^{\omega_0} R_i)_{0 \leqslant i,j \leqslant N-1}. \tag{5.3}$$

Its rows generate the lattice that will be reduced in our algorithm. The (diagonal) right half of the matrix are weights that will be used for two tasks:

- (remainders) controlling that $P_0(ux, vf(x))$, $P_1(ux, vf(x))$ are uniformly small, where P_0 , P_1 are two polynomials, with integer coefficients, output by the lattice reduction process. Matrix A_1 helps us securing uniform smallness of the interpolation polynomial of P_0, resp. P_1, and matrix A_2 helps us securing smallness of the corresponding approximation remainders (this accounts for the presence of the R_i term);
- (coefficients) controlling the size of the coefficients of the polynomial, hence the quality of the lower bound deduced from the output of Algorithm 2; this accounts for the presence of the ρ^{ω_0} term. We shall assume later on (in Section 5.2) that $4\rho^{\omega_0} v M_{\rho,a,b}(f)/(\rho^{N-1}(\rho-1)) < 1$. This is a necessary condition for the success of the method; otherwise, most of the $\rho^{\omega_0} R_i$ are too large, and the method is bound to fail. Note that this assumption can be made without loss of generality on f, as for fixed f, a, b, ω_0, ρ it holds for N large enough.

We now establish a slightly improved version of [61, Theorem 2][10]. Let an $N \times M$ matrix B whose rows span a lattice L. We assume that the entries of B satisfy: $|B_{i,j}| \leqslant \mathfrak{r}_i \cdot \mathfrak{c}_j$, for some $\mathfrak{r}_i$'s and $\mathfrak{c}_j$'s, $0 \leqslant i \leqslant N-1$ and $0 \leqslant j \leqslant M-1$. As mentioned in [61], this is typical for Coppersmith-type lattice bases.

Theorem 5.1. *Let B be an $N \times M$ matrix (with $M \geqslant N$), the entries of which are bounded by the product of some quantities $\mathfrak{r}_i$'s and $\mathfrak{c}_j$'s as described above. Let L be the lattice spanned by the rows of the matrix B, and $\mathfrak{P}$ the product of the N largest $\mathfrak{c}_j$'s. We have:*

$$\operatorname{vol} L = (\det BB^t)^{1/2} \leqslant \binom{M}{N}^{1/2} N^{N/2} \left(\prod_{i=0}^{N-1} \mathfrak{r}_i\right) \mathfrak{P}.$$

Proof. Let us denote $\mathbf{C}_0, \dots, \mathbf{C}_{M-1}$ the columns of B. The classical Lagrange's identity, which is a particular case of Cauchy-Binet formula [25, 37], then states

[10] Note that there is an inaccuracy in the statement of [61, Theorem 2] : $\sqrt{n}$ should be replaced with $\binom{n}{d}^{1/2} d!^{1/2}$.

that

$$\det BB^t = \sum_{0\leqslant j_1<\cdots<j_N\leqslant M-1} \det(\mathbf{C}_{j_1},\dots,\mathbf{C}_{j_N})^2 \leqslant \binom{M}{N} \max_{0\leqslant j_1<\cdots<j_N\leqslant M-1} \det(\mathbf{C}_{j_1},\dots,\mathbf{C}_{j_N})^2. \tag{5.4}$$

We can assume $\left(\prod_{i=0}^{N-1}\mathfrak{r}_i\right)\neq 0$: otherwise, there is at least one row of B that is identically 0, hence $\operatorname{vol} L = 0$.

Now, for a given $0\leqslant j_1<\cdots<j_N\leqslant M-1$, if one of the $\mathfrak{c}_j$ is zero, it follows that at least one column of $(\mathbf{C}_{j_1},\dots,\mathbf{C}_{j_N})$ is zero, hence $\det(\mathbf{C}_{j_1},\dots,\mathbf{C}_{j_N})=0$. Otherwise, we consider the matrix $(\mathbf{C}'_{j_1}\dots\mathbf{C}'_{j_N})$ obtained from $(\mathbf{C}_{j_1}\dots\mathbf{C}_{j_N})$ after having divided the i-th row by $\mathfrak{r}_i$ for all $i=0,\dots,N-1$ and the j_k-th column by cc_{j_k} for all $k=1,\dots,N$. All the coefficients of $(\mathbf{C}'_{j_1}\dots\mathbf{C}'_{j_N})$ have an absolute value less or equal to 1. Hadamard's inequality then implies $\det(\mathbf{C}'_{j_1},\dots,\mathbf{C}'_{j_N})^2\leqslant N^N$. It follows

$$\det(\mathbf{C}_{j_1},\dots,\mathbf{C}_{j_N})^2 = \left(\prod_{i=0}^{N-1}\mathfrak{r}_i\right)^2\left(\prod_{k=1}^{N}\mathfrak{c}_{j_k}\right)^2\det(\mathbf{C}'_{j_1},\dots,\mathbf{C}'_{j_N})^2 \leqslant \left(\prod_{i=0}^{N-1}\mathfrak{r}_i\right)^2\mathfrak{P}^2N^N.$$

We conclude by combining the last inequality with (5.4). □

Then, we upper bound $\det AA^t$, where A is the matrix defined by (5.3).

Theorem 5.2. *Let $\rho>1$, $\omega_0\geqslant 0$, $a<b$, $N\geqslant 2$, $f_0,\dots,f_{N-1}$ be functions analytic in a neighbourhood of $E_{\rho,a,b}$. We have*

$$(\det AA^t)^{1/2}\leqslant (64N)^{N/2}\left(\frac{\rho}{\rho-1}\right)^N\frac{\prod_{i=0}^{N-1}M_{\rho,a,b}(f_i)}{\rho^{N(N-1)/2+\lfloor\omega_0\rfloor(\lfloor\omega_0\rfloor-2\omega_0+1)/2}}$$

Proof. For $0\leqslant i,j\leqslant N-1$, we have from Proposition 4.1 and Lemma A.6,

$$|A_{1,i,j}|\leqslant\left|\frac{c_{j,i}}{2^{\delta_{j0}}}\right|\leqslant 2M_{\rho,a,b}(f_i)\frac{1}{\rho^j}\frac{\rho^2+1}{\rho^2-1}\leqslant 2\frac{M_{\rho,a,b}(f_i)}{\rho^j}\frac{\rho}{\rho-1}.$$

Now, let us write, for $0\leqslant i,j\leqslant N-1$,

$$|A_{2,i,j}|\leqslant\rho^{\omega_0}R_i = 4\rho^{\omega_0}\frac{M_{\rho,a,b}(f_i)}{\rho^{N-1}(\rho-1)} = 4\rho^{\omega_0}\frac{\rho}{\rho-1}\frac{M_{\rho,a,b}(f_i)}{\rho^N}.$$

In view of these estimates, we can apply Theorem 5.1 with $\mathfrak{r}_i = 2\rho/(\rho-1)M_{\rho,a,b}(f_i)$ for $i=0,\dots,N-1$,

$$\mathfrak{c}_j=\begin{cases}\rho^{-j}, & 0\leqslant j\leqslant N-1,\\ 2\rho^{\omega_0-N}, & N\leqslant j\leqslant 2N-1.\end{cases}$$

Hence, $\mathfrak{P}$ is the maximum, for $s=0,\dots,N$, of

$$\prod_{k=1}^{s}\frac{1}{\rho^{k-1}}\prod_{k=s+1}^{N}\frac{2\rho^{\omega_0}}{\rho^N} = \frac{1}{\rho^{s(s-1)/2}}\frac{(2\rho^{\omega_0})^{N-s}}{\rho^{N(N-s)}}.$$

Finally, Theorem 5.1 yields

$$(\det AA^t)^{1/2} \leqslant \binom{2N}{N}^{1/2} N^{N/2} 2^N \left(\frac{\rho}{\rho-1}\right)^N \max_{0\leqslant s\leqslant N} \frac{(2\rho^{\omega_0})^{N-s}}{\rho^{s(s-1)/2+N(N-s)}} \prod_{i=0}^{N-1} M_{\rho,a,b}(f_i)$$
$$\leqslant (64N)^{N/2} \left(\frac{\rho}{\rho-1}\right)^N \max_{0\leqslant s\leqslant N} \frac{\rho^{\omega_0(N-s)}}{\rho^{s(s-1)/2+N(N-s)}} \prod_{i=0}^{N-1} M_{\rho,a,b}(f_i).$$

Then, if $P_s = \rho^{\omega_0(N-s)}/\rho^{s(s-1)/2+N(N-s)}$, we have $P_{s+1}/P_s = \rho^{N-s-\omega_0}$, hence P_s is maximal for $s = N - \lfloor \omega_0 \rfloor$, which completes the proof of the Theorem. □

We now specialize the previous estimate to our situation, where we shall use the ordered list of functions

$$[f_i, 0 \leqslant i \leqslant (d+1)(d+2)/2 - 1] \tag{5.5}$$
$$= [x \mapsto u^k x^k v^\ell f(x)^\ell, \ell = 0, \dots, d, k = 0, \dots, d-\ell].$$

Let the $c_{j,i}$'s be defined by (5.2), using the same ordering for the functions and $[R_{2,i}, i = 0, \dots, N-1]$ be the ordered list

$$\left[4\frac{u^k M_{\rho,a,b}(x)^k v^\ell M_{\rho,a,b}(f)^\ell}{\rho^{N-1}(\rho-1)}; \ell = 0, \dots, d, k = 0, \dots, d-\ell\right].$$

We denote again $A = (A_1|A_2)$ the $N \times 2N$ matrix defined[11] by

$$A_1 = \left(c_{j,i}/2^{\delta_{j0}}\right)_{0\leqslant i,j\leqslant N-1}, A_2 = \left(\delta_{ij}\rho^{\omega_0} R_{2,i}\right)_{0\leqslant i,j\leqslant N-1}. \tag{5.6}$$

Corollary 5.3. *Let $\rho > 1$, $a < b$, f be a function analytic in a neighbourhood of $E_{\rho,a,b}$. Let $d \geqslant 1$, $N = (d+1)(d+2)/2$, $\omega_0 \geqslant 0$, $u, v \in \mathbb{N} \setminus \{0\}$. Define*

$$f_{k,\ell}(x) = u^k x^k v^\ell f(x)^\ell, 0 \leqslant \ell \leqslant d, 0 \leqslant k \leqslant d-\ell,$$

the matrices A_1, A_2, $A = (A_1|A_2)$ as in (5.6), and the quantity $\Delta_{N,[a,b],\omega_0} := (\det AA^t)^{1/2}$. We have

$$\Delta_{N,[a,b],\omega_0}^{1/(N-1)} \leqslant 30\sqrt{N} \left(\frac{\rho}{\rho-1}\right)^{N/(N-1)} \frac{(uv)^{2N/(3(d+3))}}{\rho^{N/2+\lfloor\omega_0\rfloor(\lfloor\omega_0\rfloor-2\omega_0+1)/(2(N-1))}} \tag{5.7}$$
$$\left(\frac{b-a}{2}\left(\frac{\rho+\rho^{-1}}{2}\right) + \left|\frac{b+a}{2}\right|\right)^{2N/(3(d+3))} M_{\rho,a,b}(f)^{2N/(3(d+3))}.$$

Proof. Follows from Theorem 5.2, the facts that $M_{\rho,a,b}(x^k f(x)^\ell) \leqslant M_{\rho,a,b}(x)^k M_{\rho,a,b}(f)^\ell$, the inequality

$$M_{\rho,a,b}(x) \leqslant \frac{b-a}{2}\left(\frac{\rho+\rho^{-1}}{2}\right) + \left|\frac{b+a}{2}\right|,$$

and finally the fact that $(8\sqrt{N})^{N/(N-1)} \leqslant 30\sqrt{N}$ for $N \geqslant 3$. □

[11] This is the same definition as (5.3) where we have specialized the f_i.

5.2. **Statement of the algorithms.** Our main routine is Algorithm 2. It comes together with Algorithm 1 that mainly constructs the lattice to be reduced in Algorithm 2. We shall make use of the following notation: for any $x \in \mathbb{R}$, $[x]_0 = \lfloor x \rfloor$ if $x \geqslant 0$ and $\lceil x \rceil$ otherwise. In the sequel, as in Corollary 5.3, we define $N = (d+1)(d+2)/2$.

Before writing the algorithm, we explain how we turn our problem, which leads to the reduction of a sublattice of $\mathbb{R}^{2N}$, into a problem leading to the reduction of a sublattice of $\mathbb{Z}^{2N}$.

From a mathematical point of view, the lattice that we would ideally work with is the one generated by the rows of A defined in (5.6), the volume of which is estimated in Corollary 5.3. And yet, in order to perform lattice reduction computations, it is safer to work with lattices given by vectors defined over $\mathbb{Z}$, hence the introduction of $\hat{A} = (\hat{A}_1 | \hat{A}_2)$:

$$\begin{aligned} \hat{A}_1 &= \left(\left[2^{\mathsf{tprec}} A_1[i,j] \right]_0 / 2^{\mathsf{tprec}} \right)_{0 \leqslant i,j \leqslant N-1}, \\ \hat{A}_2 &= \left(\lfloor 2^{\mathsf{tprec}} A_2[i,j] \rfloor / 2^{\mathsf{tprec}} \right)_{0 \leqslant i,j \leqslant N-1}, \end{aligned} \tag{5.8}$$

where $\mathsf{tprec} = \lceil -\log_2(\min_{0 \leqslant i \leqslant N-1} A_2[i,i]) + \log_2(N) \rceil + 5$. The integer tprec corresponds to a truncation precision that will allow us to work over $\mathbb{Z}^N$ and keep enough information from A at the same time.

Remark 5.4. By construction, $\left| \hat{A}[i,j] \right| \leqslant |A[i,j]|$ for all i, j. Hence, Theorem 5.2 and its corollaries, which proceed by upper bounding the absolue values of the coefficients of A and applying Theorem 5.1, also hold for $(\det \hat{A}\hat{A}^t)^{1/2}$.

Note that the matrices M_c and M_r computed in Algorithm 1 correspond to the scaled matrices $2^{\mathsf{tprec}}\hat{A}_1$ and $2^{\mathsf{tprec}}\hat{A}_2$.

The rows of $\hat{A}$ generate the lattice that will be reduced in our algorithm.

Lemma 5.5. *The $\mathbb{Z}$-module generated by the rows of $\hat{A}$ is a lattice of rank N.*

Proof. Recall that $\mathsf{tprec} = \lceil -\log_2(\min_{0 \leqslant i \leqslant N-1} A_2[i,i]) + \log_2(N) \rceil + 5$. Thus, for $i = 0, \ldots, N-1$, $2^{\mathsf{tprec}} A_2[i,i] \geqslant 2^5 N$, hence $\hat{A}_2[i,i] \geqslant 2^{5-\mathsf{tprec}} N > 0$. This shows that the matrix $\hat{A}_2$ is an invertible diagonal matrix, so that the matrix $\hat{A}$ has full rank N. □

We now give an equivalent but more convenient form for tprec. In order to do that, we henceforth assume that the set $u[a,b]$, resp. $vf([a,b])$, contains at least one nonzero integer n_x, resp. n_f; note that this assumption is made without loss of generality with respect to our problem, since if the assumption does not hold the problem is trivial.

Lemma 5.6. *We have*

$$\begin{aligned} \mathsf{tprec} &= \lceil -\log_2(\rho^{\omega_0} R_{2,0}) + \log_2(N) \rceil + 5 \\ &= \lceil (N - \omega_0 - 1)\log_2(\rho) + \log_2(\rho - 1) + \log_2(N) \rceil + 3 \end{aligned}$$

and

$$8N\rho^{N-\omega_0-1}(\rho-1) \leqslant 2^{\mathsf{tprec}} \leqslant 16N\rho^{N-\omega_0-1}(\rho-1).$$

Proof. From our assumption, we have $uM_{\rho,a,b}(x) \geqslant |n_x| \geqslant 1$ and $vM_{\rho,a,b}(f) \geqslant |n_f| \geqslant 1$. It then follows $R_{2,i} \geqslant 4/(\rho^{N-1}(\rho-1)) = R_{2,0}$ for all i. Therefore, we get $\mathsf{tprec} = \lceil -\log_2(\rho^{\omega_0} R_{2,0}) + \log_2(N) \rceil + 5$. □

Algorithm 1 Computation of the lattice to be reduced (1D approach)

Input: Two real numbers $a < b$, f a transcendental function analytic in a complex neighbourhood of $[a, b]$, three positive integers d, u, v, two real numbers $\rho > 1, \omega_0 \geqslant 0$ such that $4\rho^{\omega_0} v M_{\rho,a,b}(f) < \rho^{N-1}(\rho - 1)$, where $N = (d+1)(d+2)/2$.

Output: An integer tprec which is the truncation precision, two matrices $M_c, M_r \in \mathcal{M}_N(\mathbb{Z})$, respectively storing scaled values of the coefficients and of the remainders, namely $2^{\mathsf{tprec}}\hat{A}_1$ and $2^{\mathsf{tprec}}\hat{A}_2$, $\hat{A}_1$ and $\hat{A}_2$ being defined in (5.8).

1: $R_{\omega_0} \leftarrow \frac{4\rho^{\omega_0}}{\rho^{N-1}(\rho-1)}$, $\mathsf{tprec} \leftarrow \lceil -\log_2(R_{\omega_0}) + \log_2(N) \rceil + 5$
2: $L_{cheb} \leftarrow \left[\frac{b-a}{2}\cos\left((j+1/2)\frac{\pi}{N}\right) + \frac{a+b}{2}\right]_{0\leqslant j\leqslant N-1}$ // Computation of the Chebyshev nodes, listed in reverse order
3: $M_c \leftarrow [0]_{N\times N}$; $M_r \leftarrow [0]_{N\times N}$
4: $B_x \leftarrow \left|\frac{a+b}{2}\right| + \frac{b-a}{4}(\rho + \rho^{-1})$
5: $g \leftarrow \left(x \mapsto \left|f\left(\frac{a+b}{2} + \frac{b-a}{4}(\rho\exp(ix) + \rho^{-1}\exp(-ix))\right)\right|\right)$
6: $B_f \leftarrow \max\left(g([0, 2\pi])\right)$
7: **for** $\ell = 0$ **to** d **do**
8: **for** $k = 0$ **to** $d - \ell$ **do**
9: $\varphi \leftarrow (x \mapsto (ux)^k (vf(x))^\ell)$
 // We compute the coefficient matrix : for each function, we compute its value at points of L_{cheb}, use DCT and scale.
10: $U \leftarrow [\varphi(L_{cheb}[0]), \ldots, \varphi(L_{cheb}[N-1])]$
11: $L_{\mathrm{DCT}} \leftarrow \frac{2}{N}\text{DCT-II}(U)$, $L_{\mathrm{DCT}}[0] \leftarrow \frac{1}{2}L_{\mathrm{DCT}}[0]$
12: **for** $j = 0$ **to** $N - 1$ **do**
13: $M_c[i, j] \leftarrow [2^{\mathsf{tprec}} L_{\mathrm{DCT}}[j]]_0$ // Scaling of the coefficient matrix
14: **end for**
 // We compute the scaled remainder matrix.
15: $M_r[i, i] \leftarrow \left\lfloor 2^{\mathsf{tprec}} R_{\omega_0} (uB_x)^k (vB_f)^\ell \right\rfloor$, $i \leftarrow i + 1$
16: **end for**
17: **end for**
18: Return tprec, M_c, M_r

5.2.1. *Heuristic character of Algorithm 2.* We will see in the sequel of the section that, up to a suitable choice of parameters, the condition stated at Step 4 can always be satisfied. On the other hand, we do not know how to simultaneously guarantee both this condition and the condition stated at Step 8. It is even likely that it is not possible when f is close to an algebraic function of small height.

5.3. **Practical remarks.** Algorithm 1 has been written with readability in mind. We now add some practical clarifications. We will discuss some experiments in Section 8.

5.3.1. *Efficiency.* The number of DCT calls (see Step 11 in Algorithm 1) can be reduced from $O(N)$ to $O(d)$ by noticing that the DCT δ' of the vector u' associated to $x\varphi(x)$ can be deduced from the DCT δ of the vector u associated to $\varphi(x)$ via the following formulas, which are easily deduced from the recurrence relation $2xT_n(x) = T_{n+1}(x) + T_{n-1}(x)$:

- $\delta'[0] = (b-a)\delta[1]/4 + (b+a)\delta[0]/2$,
- $\delta'[k] = (b-a)(\delta[k-1] + \delta[k+1])/4 + (b+a)\delta[k]/2$, $1 \leqslant k \leqslant n-2$,
- $\delta'[n-1] = (b-a)\delta[n-2]/4 + (b+a)\delta[n-1]/2$.

Algorithm 2 1D approach to Problem 2.6

Input: Two real numbers $a < b$, f a transcendental function analytic in a complex neighbourhood of $[a, b]$, three positive integers d, u, v, two real numbers $\rho > 1, \omega_0 \geqslant 0$ such that $4\rho^{\omega_0} v M_{\rho,a,b}(f) < \rho^{N-1}(\rho - 1)$, where $N = (d+1)(d+2)/2$.

Output: If successful, return $K \in \mathbb{R}_{>0}$ and a list $\mathcal{L}$ of integers, $\#\mathcal{L} \leqslant d^2$, such that for all integers X, $a \leqslant X/u \leqslant b$ and $X \notin \mathcal{L}$, for all integers Y, we have $\left|f\left(\frac{X}{u}\right) - \frac{Y}{v}\right| \geqslant 1/K$. The bound K is guaranteed to be at most $\frac{d\rho^{N-\omega_0-1}(\rho-1)}{2M_{\rho,a,b}(f)}$.

1: $(\mathsf{tprec}, M_c, M_r) \leftarrow$ Algorithm 1 $(a, b, f, d, u, v, \rho, \omega_0)$
2: $M_{LLL} \leftarrow$ LLL-reduce the rows of $(M_c \mid M_r)$
3: $U \leftarrow M_{LLL,r} M_r^{-1}$ // This is the LLL change of basis matrix; $M_{LLL,r}$ is the right part of the matrix M_{LLL}. Note that M_r is diagonal.
4: **if** $\max(\|(M_{LLL}[0,j])_{0\leqslant j\leqslant 2N-1}\|_2, \|(M_{LLL}[1,j])_{0\leqslant j\leqslant 2N-1}\|_2) \leqslant 2^{\mathsf{tprec}}/(2N)$ **then**
5: $[L_m[j], j = 0, \ldots, N-1] \leftarrow [X_1^k X_2^\ell$ for $k = 0$ to $d - \ell$ for $\ell = 0$ to $d]$ // List of monomials, ordered in a way compatible with Algorithm 1, Steps 7–9.
6: $P_0 \leftarrow \sum_{j=0}^{N-1} U[0,j] L_m[j]$, $P_1 \leftarrow \sum_{j=0}^{N-1} U[1,j] L_m[j]$,
7: $R(X_1) \leftarrow \mathrm{Res}_{X_2}(P_0(X_1, X_2), P_1(X_1, X_2))$
8: **if** $R(X_1) \neq 0$ **then**
9: $\mathcal{L} \leftarrow \{t \in \mathbb{Z}; R(t) = 0\}$
10: $(B_0, B_1) \leftarrow (0, 0)$
// Monomials in X_1 do not contribute to the B_i's; given the ordering of L_m we can thus start the loop at $k = d + 1$.
11: **for** $k = d+1$ **to** $N-1$ **do**
12: $J \leftarrow \frac{dL_m[k]}{dX_2}(X_1 = u \max(|a|, |b|), X_2 = v \max|f|([a, b]))$,
13: $B_0 \leftarrow B_0 + |U[0,k]|J$, $B_1 \leftarrow B_1 + |U[1,k]|J$
14: **end for**
15: return $K = 4v \max(B_0, B_1, d/2)$, $\mathcal{L}$
16: **else**
17: return "FAIL"
18: **end if**
19: **else**
20: return "FAIL"
21: **end if**

This gives, in practice, a significant speedup in the construction of the matrix for large d. Note that these formulas must be applied to L_{DCT} before the renormalisation instruction $L_{\mathrm{DCT}}[0] \leftarrow \frac{1}{2} L_{\mathrm{DCT}}[0]$.

A similar strategy applies for the computation of the remainder matrix M_r.

5.3.2. *Overestimation issues.* We now discuss the instruction $B_f \leftarrow \max(g([0, 2\pi]))$, presented at Step 6 of Algorithm 1. For some functions such as exp or Γ, we can take advantage of a closed expression for this maximum. Otherwise, either we develop a dedicated routine to derive a tight estimate of this value, or we can use interval or ball arithmetic [68, 33] to quickly obtain an upper bound, which then may raise

overestimation issues. So far, our experiments, which use Arb[12], resp. MPFI[13], for every ball, resp. interval, arithmetic based computation, did not show any problematic overestimation – we thus did not have to develop dedicated routines.

This remark leads to the fact that we may overestimate B_f, thus M_r, in implementations of the Algorithm. Still, if the condition stated at Step 4 of Algorithm 2 is satisfied with these (possibly overestimated) computed values, then this condition is also satisfied for the actual values and Theorem 5.8 and Corollary 5.11 apply.

5.3.3. *Rounding issues.* In Algorithm 1, the computation of tprec at Step 1, as well as those performed at Steps 13 and 15, as written, require correct rounding, and may raise issues such as those this paper aims at solving.

In this context, we can, however quite easily avoid them, using the classical remark that if we let $\tilde{x}$ be an underapproximation of x with $\tilde{x} \leqslant x \leqslant \tilde{x} + 1/2$, then we have $\lceil x \rceil \in \{\lceil \tilde{x} \rceil, \lceil \tilde{x} \rceil + 1\}$. Similarly, if $\tilde{x}$ is an approximation of a nonzero x such that $|\tilde{x}| - 1/2 \leqslant |x| \leqslant \tilde{x}$, we get $[x]_0 \in \{[\tilde{x}]_0, [\tilde{x}]_0 - \mathrm{sgn}(x)\}$.

It is well known that such approximations are easy to compute, either by using floating-point with sufficient intermediate precision and ensuring that we work with over/under-approximations using the suitable rounding mode for each operation, or using ball arithmetic as provided, for instance, by Arb.

In the sequel, we denote by $\mathsf{tprec}_{\mathsf{comp}}$, $M_{c,\mathsf{comp}}$, $M_{r,\mathsf{comp}}$ the quantities computed in this way. Note that M_c and M_r are then defined with $\mathsf{tprec}_{\mathsf{comp}}$ instead of tprec. We have $\mathsf{tprec}_{\mathsf{comp}} \in \{\mathsf{tprec}, \mathsf{tprec}+1\}$, $M_c[i,j] - M_{c,\mathsf{comp}}[i,j] \in \{0, \mathrm{sgn}(M_c[i,j])\}$, and $M_r[i,j] - M_{r,\mathsf{comp}}[i,j] \in \{0,1\}$ for $i,j = 0,\dots,N-1$.

The question of the intermediate precision required will be addressed in Appendix C. Note that in this setting, Remark 5.4 must be replaced by:

Remark 5.7. Let $\hat{A}_{\mathsf{comp}} = 2^{-\mathsf{tprec}_{\mathsf{comp}}}(M_{c,\mathsf{comp}}\, M_{r,\mathsf{comp}})$ be the actual computed matrix, we notice that since $\left|\hat{A}_{\mathsf{comp}}[i,j]\right| \leqslant |A[i,j]|$ for all i,j, the same argument as in Remark 5.4 applies: Theorem 5.2 and its corollaries hold for $\hat{A}_{\mathsf{comp}}$.

5.3.4. *Newton polynomials.* In practice, we replace the monomial functions $\{u^k x^k\}_{0\leqslant k\leqslant d}$ with Newton polynomial functions $\{ux(ux-1)\cdots(ux-k+1)/k!\}_{0\leqslant k\leqslant d}$. In both cases, the substitution $x = X/u$ yields integer values - respectively $\{X^k\}_{0\leqslant k\leqslant d}$ and $\{X(X-1)\cdots(X-k+1)/k!\}_{0\leqslant k\leqslant d}$. Likewise, we replace the "monomials" $\{v^k f(x)^k\}_{0\leqslant k\leqslant d}$ with "Newton polynomials" $\{vf(x)(vf(x)-1)\cdots(vf(x)-k+1)/k!\}_{0\leqslant k\leqslant d}$. Hence, the changes to operate are:

- Step 9, Algorithm 1,
  $\varphi \leftarrow \left(x \mapsto \left(\prod_{j=1}^{k}(ux-j+1)/j\right)\left(\prod_{j=1}^{\ell}(vf(x)-j+1)/j\right)\right)$,
- Step 15, Algorithm 1,
  $M_r[i,i] \leftarrow \left\lfloor 2^{\mathsf{tprec}} \frac{4}{\rho^{N-\omega_0-1}(\rho-1)} \prod_{j=1}^{k} \frac{uB_x+j-1}{j} \prod_{j=1}^{\ell} \frac{vB_f+j-1}{j} \right\rfloor$,
- Step 5, Algorithm 2,
  $L_m \leftarrow \left[\prod_{j=1}^{k} \frac{X_1-j+1}{j} \prod_{j=1}^{\ell} \frac{X_2-j+1}{j} \text{ for } k=0 \text{ to } d-\ell \text{ for } \ell = 0 \text{ to } d\right]$.

The use of Newton polynomials leads to smaller uniform norms, hence makes it possible to tackle larger intervals for a same d. This improvement is asymptotically negligible (it contributes to a lower order term) but is quite significant in practice.

[12] https://arblib.org/
[13] https://gitlab.inria.fr/mpfi

Note that the optimizations described in Section 5.3.1 can easily be adapted to the case of Newton polynomials.

5.3.5. *Choice of norms.* Beside the heuristic coprimality condition, the success condition of Algorithm 2 is expressed at Step 4 in terms of the Euclidean norm of the vectors. This is a mere convenience related to the fact that the bounds on the LLL algorithm are expressed in terms of this norm, making it more tractable in our proofs of correctness / complexity analysis.

Alternatively, one may make the choice of a success condition expressed in terms of the 1-norm, namely

$$\max_{i=0,1}\left(\|(M_{LLL}[i,j])_{0\leqslant j\leqslant 2N-1}\|_1+\frac{\|(M_{LLL}[i,j])_{N\leqslant j\leqslant 2N-1}\|_1}{16}\right)<2^{\mathsf{tprec}-1}. \tag{5.9}$$

Indeed, as we shall see in the proof of Theorem 5.8, this condition means that from the vector we can derive a polynomial P such that $P(ux,vf(x))=\sum_{i=0}^{N-1}c_iT_{i,[a,b]}(x)+R(x)$, with $\sum_{i=0}^{N-1}|c_i|+\|R(x)\|_\infty<1/2$. Since $\|T_{i,[a,b]}\|_\infty=1$ for all i, we see that (5.9) guarantees that the polynomial P verifies $\|P(ux,vf(x))\|_\infty<1/2$ over $[a,b]$, which is the key criterion for the success of the algorithm. As this condition is slightly more efficient in practice, we recommend using it in any implementation of our algorithm.

5.4. **Proof of correctness.** In this section, we shall prove the correctness of Algorithm 2. This is done in two steps: first, in Subsection 5.4.1 we prove that if the Algorithm does not return FAIL, then the output is indeed as specified; second, in Subsection 5.4.2, we prove that for suitable choices of parameters, Algorithm 2 may not return FAIL at Step 20. Recall $N=(d+1)(d+2)/2$.

5.4.1. *Proof of correctness of the output in case of success.* This part is devoted to the proof of the following.

Theorem 5.8. *Let d,u,v be nonzero integers, $N=(d+1)(d+2)/2$, $(f_j)_{0\leqslant j\leqslant N-1}=(u^kx^kv^\ell f(x)^\ell)_{\substack{0\leqslant\ell\leqslant d\\0\leqslant k\leqslant d-\ell}}$. Let $\omega_0\geqslant 0$, $\rho>1$ such that there exists $\Lambda=(\lambda_{k,\ell})_{\substack{0\leqslant\ell\leqslant d\\0\leqslant k\leqslant d-\ell}}\in\mathbb{Z}^N$ with $\|\Lambda\hat{A}\|_2\leqslant 1/(2N)$, and let $P(X_1,X_2)=\sum_{0\leqslant k+\ell\leqslant d}\lambda_{k,\ell}X_1^kX_2^\ell$.*

Then, we have

(1) $\max_{x\in[a,b]}|P(ux,vf(x))|<1/2$;

(2) $\max_{\substack{x\in[a,b],z\in f([a,b])\\0\leqslant|y-z|\leqslant|z|/(2d)}}\left|\frac{P(ux,vz)-P(ux,vy)}{z-y}\right|<2vB<\frac{d\rho^{N-\omega_0-1}(\rho-1)}{4M_{\rho,a,b}(f)}$, *where*

$$B=\sum_{\substack{1\leqslant\ell\leqslant d\\0\leqslant k\leqslant d-\ell}}\ell|\lambda_{k,\ell}|u^k\max(|a|,|b|)^kv^{\ell-1}\|f\|_{\infty,[a,b]}^{\ell-1}.$$

Proof. For $j=0,\dots,2N-1$, we have $|(\Lambda A)[j]-(\Lambda\hat{A})[j]|_1\leqslant\sum_{0\leqslant k+\ell\leqslant d}|\lambda_{k,\ell}|2^{-\mathsf{tprec}}\leqslant\sum_{0\leqslant k+\ell\leqslant d}|\lambda_{k,\ell}|\min_i\hat{A}_2[i,i]\frac{2^{-5}}{N}$, cf. proof of Lemma 5.5. As

$$\sum_{0\leqslant k+\ell\leqslant d}|\lambda_{k,\ell}|\min_i\hat{A}_2[i,i]\leqslant\|\Lambda\hat{A}_2\|_1,$$

we get $|(\Lambda A)[j]-(\Lambda\hat{A})[j]|\leqslant\frac{1}{32N}\|\Lambda\hat{A}_2\|_1$. Then, it comes $\|\Lambda A\|_1\leqslant\|\Lambda\hat{A}\|_1+\frac{\|\Lambda\hat{A}_2\|_1}{16}\leqslant\|\Lambda\hat{A}\|_1+\sqrt{N}\frac{\|\Lambda\hat{A}_2\|_2}{16}$ thanks to Cauchy-Schwarz inequality. Finally, we obtain $\|\Lambda A\|_1\leqslant\|\Lambda\hat{A}\|_1+\frac{1}{32N^{1/2}}$ from the assumption $\|\Lambda\hat{A}\|_2\leqslant 1/(2N)$.

Let P be as in the statement of the Theorem, and $Q(x) = \sum_{j=0}^{N-1} q_j T_{j,[a,b]}(x)$ be the interpolation polynomial of $P(ux, vf(x))$ at the order N Chebyshev nodes of the first kind. Then, the coordinates of ΛA_1 are exactly q_j, $0 \leqslant j \leqslant N-1$: indeed, the matrix A_1 contains the DCT of the functions $(ux)^k (vf(x))^\ell$, so that ΛA_1 is the DCT of $P(ux, vf(x))$, meaning (see (5.2)) that it contains the coefficients of the interpolation polynomial $P(ux, vf(x))$ in the Chebyshev basis $(T_{j,[a,b]}(x))_{0\leqslant j\leqslant N-1}$.

Proposition 4.1 shows that

$$\begin{aligned}\max_{x\in[a,b]} |Q(x) - P(ux, vf(x))| &\leqslant 4\frac{M_{\rho,a,b}(P)}{\rho^{N-1}(\rho-1)}\\ &\leqslant 4\sum_{0\leqslant k+\ell\leqslant d} |\lambda_{k,\ell}| \frac{u^k M_{\rho,a,b}(x)^k v^\ell M_{\rho,a,b}(f)^\ell}{\rho^{N-1}(\rho-1)},\end{aligned}$$

hence

$$\max_{x\in[a,b]} |P(ux, vf(x)| \leqslant \max_{x\in[a,b]} |Q(x)| + 4\sum_{0\leqslant k+\ell\leqslant d} |\lambda_{k,\ell}| \frac{u^k M_{\rho,a,b}(x)^k v^\ell M_{\rho,a,b}(f)^\ell}{\rho^{N-1}(\rho-1)}.$$

As $\max_{x\in[a,b]} |T_{k,[a,b]}(x)| = 1$ for all k, we have $\max_{x\in[a,b]} |Q(x)| \leqslant \sum_{0\leqslant j\leqslant N-1} |q_j|$, so that:

$$\begin{aligned}\max_{x\in[a,b]} |P(ux, vf(x)| \leqslant &\sum_{0\leqslant j\leqslant N-1} |q_j| +\\ &4\rho^{\omega_0} \sum_{0\leqslant k+\ell\leqslant d} |\lambda_{k,\ell}| \frac{u^k M_{\rho,a,b}(x)^k v^\ell M_{\rho,a,b}(f)^\ell}{\rho^{N-1}(\rho-1)} \qquad (5.10)\\ &= \|\Lambda A\|_1 \leqslant \|\Lambda \hat{A}\|_1 + 1/(2^5\sqrt{N})\\ &\leqslant 1/(2^5\sqrt{N}) + \sqrt{2N}\|\Lambda\hat{A}\|_2 \text{ thanks to Cauchy-Schwarz inequality}\\ &\leqslant 1/(2^5\sqrt{N}) + 1/\sqrt{2N} < 1/2 \text{ since } N \geqslant 3. \qquad (5.11)\end{aligned}$$

Finally, let $x \in [a,b]$, $z \in f([a,b])$ such that $0 \leqslant |y-z| \leqslant |z|/(2d)$. Notice that the quantity $\frac{P(ux,vy)-P(ux,vz)}{y-z}$ is actually a polynomial, so is well defined for $y = z$.

First, we note that $\max(|a|,|b|) \leqslant M_{\rho,a,b}(x)$ and $\|f\|_{\infty,[a,b]} \leqslant M_{\rho,a,b}(f)$ from the maximum modulus principle. Then, we have

$$\begin{aligned}\left|\frac{P(ux,vy)-P(ux,vz)}{y-z}\right| &\leqslant \sum_{\substack{1\leqslant\ell\leqslant d\\ 0\leqslant k\leqslant d-\ell}} \ell|\lambda_{k,\ell}|u^k|x|^k v^\ell \max(|z|,|y|)^{\ell-1}\\ &\leqslant \sum_{\substack{1\leqslant\ell\leqslant d\\ 0\leqslant k\leqslant d-\ell}} \ell|\lambda_{k,\ell}|u^k|x|^k v^\ell |z|^{\ell-1} \underbrace{\left(1+\frac{1}{2d}\right)^{\ell-1}}_{<2}\\ &< 2\sum_{\substack{1\leqslant\ell\leqslant d\\ 0\leqslant k\leqslant d-\ell}} \ell|\lambda_{k,\ell}|u^k \max(|a|,|b|)^k v^\ell \|f\|_{\infty,[a,b]}^{\ell-1} =: 2vB \qquad (5.12)\\ &\leqslant 2d\sum_{\substack{1\leqslant\ell\leqslant d\\ 0\leqslant k\leqslant d-\ell}} |\lambda_{k,\ell}|u^k M_{\rho,a,b}(x)^k v^\ell M_{\rho,a,b}(f)^{\ell-1}\\ &< \frac{d\rho^{N-\omega_0-1}(\rho-1)}{4M_{\rho,a,b}(f)}. \qquad (5.13)\end{aligned}$$

The last inequality follows from the comparison of (5.10) to (5.11). We take the supremum over the compact set $x \in [a,b], z \in f([a,b]), 0 \leqslant |y-z| \leqslant |z|/(2d)$; as, again, the quantity under study is actually a polynomial, this supremum is actually a maximum, which concludes the proof. □

Remark 5.9. Note that $\|\Lambda A\|_1 \geqslant 4\sum_{0\leqslant k+\ell\leqslant d} |\lambda_{k,\ell}| \frac{u^k M_{\rho,a,b}(x)^k v^\ell M_{\rho,a,b}(f)^\ell}{\rho^{N-1}(\rho-1)}$ in particular. Since the constraint $\|\Lambda\hat{A}\|_2 \leqslant 1/(2N)$ implies $\|\Lambda A\|_1 < 1/2$, it comes either $\lambda_{k,\ell} = 0$ or $4\frac{u^k M_{\rho,a,b}(x)^k v^\ell M_{\rho,a,b}(f)^\ell}{\rho^{N-1}(\rho-1)} < 1$ for any k,ℓ. Also, the proof of Lemma 5.6 shows in particular that $R_{2,i} \geqslant R_{2,d+2} = 4vM_{\rho,a,b}(f)/(\rho^{N-1}(\rho-1))$ for all $i \geqslant d+2$. Hence, if $\rho^{\omega_0} R_{2,d+2} \geqslant 1$, we thus have $\rho^{\omega_0} R_{2,i} \geqslant 1$ for all $i \geqslant 1$ and $\lambda_{k,\ell} = 0$ for any $1 \leqslant \ell \leqslant d$, $0 \leqslant k \leqslant d-\ell$: the only functions taken into account are the $u^k x^k$'s and the method fails as claimed at the beginning of this section. This explains the condition $4v\rho^{\omega_0} M_{\rho,a,b}(f) < \rho^{N-1}(\rho-1)$.

Remark 5.10. The proof should be slightly adapted if Subsection 5.3.3 is used. Recall that $\hat{A}_{\mathsf{comp}} = 2^{-\mathsf{tprec}_{\mathsf{comp}}}(M_{c,\mathsf{comp}}\, M_{r,\mathsf{comp}})$, we obtain for $j = 0,\dots,2N-1$,

$$|(\Lambda A)[j] - (\Lambda \hat{A}_{\mathsf{comp}})[j]| \leqslant \sum_{0\leqslant k+\ell\leqslant d} |\lambda_{k,\ell}| 2^{1-\mathsf{tprec}_{\mathsf{comp}}} \leqslant \sum_{0\leqslant k+\ell\leqslant d} |\lambda_{k,\ell}| \min_i \hat{A}_2[i,i] \frac{2^{-4}}{N}$$

from which follows $\|\Lambda A\|_1 \leqslant \|\Lambda\hat{A}\|_1 + \frac{\|\Lambda\hat{A}_2\|_1}{2^3} \leqslant \|\Lambda\hat{A}\|_1 + \frac{1}{2^4 N^{1/2}}$. The upper bound in Inequality (5.11) becomes $1/(2^4 N^{1/2}) + 1/\sqrt{2N} < 1/2$ since $N \geqslant 3$.

Note also that the success condition (5.9) becomes

$$\max_{i=0,1}\left(\|(M_{LLL}[i,j])_{0\leqslant j\leqslant 2N-1}\|_1 + \frac{\|(M_{LLL}[i,j])_{N\leqslant j\leqslant 2N-1}\|_1}{8}\right) < 2^{\mathsf{tprec}-1}.$$

From Theorem 5.8, we can deduce a lower bound for $|Y/v - f(X/u)|$ in the following way:

Corollary 5.11. *With the notations and assumptions of Theorem 5.8, we have, for all $X \in \mathbb{Z}$, $a \leqslant X/u \leqslant b$, all $Y \in \mathbb{Z}$, either:*

$$P(X,Y) = 0 \text{ or } \left|\frac{Y}{v} - f\left(\frac{X}{u}\right)\right| > \frac{1}{2v\max(2B,d)} > \frac{2M_{\rho,a,b}(f)}{d\rho^{N-\omega_0-1}(\rho-1)}.$$

Proof. The inequality $\frac{1}{4vB} > \frac{2M_{\rho,a,b}(f)}{d\rho^{N-\omega_0-1}(\rho-1)}$ follows from the second point of Theorem 5.8.

We have, for any $x \in [a,b], y \in \mathbb{R}$,

$$P(ux,vy) = P(ux,vf(x)) + (y-f(x))\frac{P(ux,vy) - P(ux,vf(x))}{y-f(x)}. \tag{5.14}$$

As $P \in \mathbb{Z}[X_1,X_2]$ and $X,Y \in \mathbb{Z}$, we must have $P(X,Y) \in \mathbb{Z}$, so that either $P(X,Y) = 0$ or $|P(X,Y)| \geqslant 1$. In the former case, there is nothing to prove. In the latter case, we plug $x = X/u$ and $y = Y/v$ into (5.14) and obtain

$$\left|\frac{Y}{v} - f\left(\frac{X}{u}\right)\right| \left|\frac{P(X,Y) - P(X,vf(X/u))}{Y/v - f(X/u)}\right| \geqslant |P(X,Y)| - |P(X,vf(X/u))| > \frac{1}{2} \tag{5.15}$$

from the first point of Theorem 5.8. If we assume $|Y/v - f(X/u)| \leqslant |f(X/u)|/(2d)$, we then derive the expected result from the second point of Theorem 5.8.

We now assume $|Y/v - f(X/u)| > |f(X/u)|/(2d)$.

- If $|f(X/u)| < 1/v$, then either $Y = -1, 0, 1$ or $|Y/v - f(X/u)| \geqslant 1/v$. For $Y = -1, 0, 1$, we have

$$\left|\frac{P(X,Y) - P(X, vf(X/u))}{Y/v - f(X/u)}\right| \leqslant \sum_{\substack{1\leqslant \ell\leqslant d\\ 0\leqslant k\leqslant d-\ell}} |\lambda_{k,\ell}||X|^k v \sum_{0\leqslant j\leqslant \ell-1} \underbrace{|Y|}_{\leqslant 1}{}^{j} \underbrace{|vf(X/u)|}_{\leqslant 1}{}^{\ell-1-j}$$

$$\leqslant \sum_{\substack{1\leqslant \ell\leqslant d\\ 0\leqslant k\leqslant d-\ell}} \ell|\lambda_{k,\ell}|u^k|X/u|^k v \leqslant \sum_{\substack{1\leqslant \ell\leqslant d\\ 0\leqslant k\leqslant d-\ell}} \ell|\lambda_{k,\ell}|u^k \max(|a|,|b|)^k v \underbrace{(v\|f\|_{\infty,[a,b]})}_{\geqslant |n_f|\geqslant 1}{}^{\ell-1}$$

  where n_f was introduced just before Lemma 5.6. The conclusion follows by combining this upper bound with (5.15).

  If $|Y/v - f(X/u)| \geqslant 1/v$, the conclusion holds since not all $\lambda_{k,\ell}$ are zero and

$$B = 4v \sum_{\substack{1\leqslant \ell\leqslant d\\ 0\leqslant k\leqslant d-\ell}} \ell \underbrace{|\lambda_{k,\ell}|}_{\in\mathbb{N}} \underbrace{u^k \max(|a|,|b|)^k}_{|n_x|^k\geqslant 1} \underbrace{v^{\ell-1}\|f\|^{\ell-1}_{\infty,[a,b]}}_{|n_f|^{\ell-1}\geqslant 1} \geqslant 4v.$$

- Likewise, if $|f(X/u)| \geqslant 1/v$, we have

$$|Y/v - f(X/u)| > \frac{|f(X/u)|}{2d} \geqslant \frac{1}{2dv} > \frac{2M_{\rho,a,b}(f)}{d\rho^{N-\omega_0-1}(\rho-1)},$$

  since $1 > \frac{4vM_{\rho,a,b}(f)}{\rho^{N-\omega_0-1}(\rho-1)}$ thanks to Remark 5.9, the conclusion holds again.

□

We now have all the elements to prove the correctness of Algorithm 2. First, recall that the matrices M_c and M_r computed in Algorithm 1 correspond to the scaled matrices $2^{\mathsf{tprec}}\hat{A}_1$ and $2^{\mathsf{tprec}}\hat{A}_2$. If the condition stated at Step 4 of Algorithm 2 is satisfied, then Theorem 5.8 and Corollary 5.11 prove that, given an integer pair (X, Y), either $P_0(X, Y) = P_1(X, Y) = 0$, or the lower bound of Corollary 5.11 holds. If the test at Step 8 succeeded, P_0 and P_1 are coprime and since the total degree of each P_i is at most d, we deduce from Bézout's theorem that the cardinal of $\mathcal{L}$, the list of integer roots of P_0 and P_1 is at most d^2.

Steps 7 & 9 of Algorithm 2 deal with the former case, by trying to solve the polynomial system $P_0(x, y) = P_1(x, y) = 0$. If P_0 and P_1 are not coprime, however, this will fail; this is what makes our algorithm (and actually all bivariate versions of Coppersmith's method) heuristic. When this situation occurs, one convenient solution is to subdivide the interval of interest into two subintervals and to apply the algorithm again to these subintervals.

When $X \notin \mathcal{L}$, then at least one of the values $P_0(X, Y), P_1(X, Y)$ is non-zero. Then, we deduce from Corollary 5.11 that $|f(X/u) - Y/v| > 1/K > \frac{2M_{\rho,a,b}(f)}{d\rho^{N-\omega_0-1}(\rho-1)}$, where K is the value computed at Step 15 of Algorithm 2.

5.4.2. *Examination of the success of Algorithm 2.* If we apply the LLL lattice basis reduction algorithm to $\hat{A}$, we obtain:

Corollary 5.12. *Assume that* $\det(\hat{A}\hat{A}^t)^{1/2(N-1)} \leqslant \frac{2^{-(N+3)/4-\mathsf{tprec}/(N-1)}}{N}$*; then Theorem 5.8 applies with* Λ *equal to any of the first two vectors of an LLL-reduced basis of the lattice generated by the rows of* $\hat{A}$*.*

Proof. Let $\tilde{A} = 2^{\mathsf{tprec}}\hat{A} \in \mathcal{M}_n(\mathbb{Z})$, we know from Theorem 4.5 that if w_1 and w_2 denote these first two vectors, we have

$$\|2^{\mathsf{tprec}} w_i\|_2 \leqslant 2^{(N-1)/4} \max(\det(\tilde{A}\tilde{A}^t)^{1/2(N-1)}, \det(\tilde{A}\tilde{A}^t)^{1/(2N)}).$$

As $\tilde{A}$ is an integer matrix, its determinant is an integer, so that $\det(\tilde{A}\tilde{A}^t)^{1/2(N-1)} \geqslant \det(\tilde{A}\tilde{A}^t)^{1/(2N)}$; we thus have

$$2^{\mathsf{tprec}}\|w_i\|_2 \leqslant 2^{(N-1)/4} 2^{N\mathsf{tprec}/(N-1)} \det(\hat{A}\hat{A}^t)^{1/2(N-1)},$$

hence

$$\|w_i\|_2 \leqslant 2^{(N-1)/4} 2^{\mathsf{tprec}/(N-1)} \det(\hat{A}\hat{A}^t)^{1/2(N-1)} \leqslant \frac{1}{2N}.$$

□

We shall base our analysis on Inequality (5.7). The important term in the analysis is $(uv)^{2N/(3(d+3))} M_{\rho,a,b}(f)^{2N/(3(d+3))}/\rho^{N/2+\cdots}$. The quality of the bound thus depends on ρ and the growth of f.

We shall start by giving a general result. This result will then be turned into more readable versions under various sets of assumptions in Theorems 5.18, 5.26 and 5.27.

Proposition 5.13. *Let f be analytic in a neighbourhood of the closed disc $\mathcal{D}_{a,b,K} = \{z \in \mathbb{C} : |z-(a+b)/2| \leqslant K/2\}$, d be an integer $\geqslant 1$, $N = (d+1)(d+2)/2$, and $\omega_0 \geqslant 0$, $\rho = K/(b-a) \geqslant 2$ be two real parameters. Let $M_{\mathcal{D}_{a,b,K}}(f) := \max_{z\in\mathcal{D}_{a,b,K}} |f(z)|$ and recall that $\Delta_{N,[a,b],\omega_0} = (\det AA^t)^{1/2}$.*

Then, if

$$b - a < K\left(2^{6(d+3)} uv(|a+b|+K) M_{\mathcal{D}_{a,b,K}}(f)\right)^{-\frac{2dN}{3(N(N-3)+2\omega_0+\lfloor\omega_0\rfloor(\lfloor\omega_0\rfloor-2\omega_0+1))}},$$

we have $\Delta_{N,[a,b],\omega_0}^{1/(N-1)} < \frac{2^{-(N+3)/4-\mathsf{tprec}/(N-1)}}{N}$.

Proof. We first notice that under our assumption $\rho = K/(b-a) \geqslant 2$, we have $E_{\rho,a,b} \subset \mathcal{D}_{a,b,K}$. Thanks to Corollary 5.3, in view of $(\rho/(\rho-1))^{N/(N-1)} \leqslant 2^{3/2}$, we have

$$\Delta_{N,[a,b],\omega_0}^{1/(N-1)} \leqslant 60\sqrt{2N}\frac{(uv(|a+b|+K)/2)^{2N/(3(d+3))}}{\rho^{N/2+\lfloor\omega_0\rfloor(\lfloor\omega_0\rfloor-2\omega_0+1)/(2(N-1))}} M_{\mathcal{D}_{a,b,K}}(f)^{2N/(3(d+3))}.$$

Thus, for $\Delta_{N,[a,b],\omega_0}^{1/(N-1)} < 2^{-(N+3)/4} 2^{-\mathsf{tprec}/(N-1)}/N$, it suffices, using Lemma 5.6 that

$$\rho > \Big(60 \cdot 2^{(N+5)/4-2N/(3(d+3))+3/(N-1)} N^{3/2} \\ (uv(|a+b|+K) M_{\mathcal{D}_{a,b,K}}(f))^{2N/(3(d+3))}\Big)^{\frac{2(N-1)}{N(N-3)+2\omega_0+\lfloor\omega_0\rfloor(\lfloor\omega_0\rfloor-2\omega_0+1)}}.$$

We observe that

$$60 \cdot 2^{(N+5)/4-2N/(3(d+3))+3/(N-1)} N^{3/2} < 2^{4N}$$

for $d \geqslant 1$, from which the proposition follows. □

Corollary 5.14. *Under the assumptions of Proposition 5.13, Algorithm 2 over $[a,b]$ produces at Step 6 two polynomials P_0, P_1 such that $\max_{x\in[a,b]} |P_i(ux, vf(x))| < 1/2$ for $i \in \{0,1\}$. In particular, Algorithm 2 never reaches Step 20 and its output is valid.*

Proof. It suffices to apply Proposition 5.13, Corollary 5.12, Theorem 5.8 and Corollary 5.11 (in order to get the estimate on $|f(X/u) - Y/v|$ which is part of the output of Algorithm 2). □

Note that the algorithm may still return "FAIL" at Step 17, precisely in the case where P_0 and P_1 are not coprime – this makes the algorithm heuristic.

5.5. **Complexity analysis.** In this subsection, we deduce estimates for the complexity of our algorithm applied to a fixed interval $[\alpha, \beta)$. This actually requires several things:

- Evaluating the complexity of the basic blocks, namely Algorithms 1 and 2 (Subsection 5.5.1), and the precision required for all intermediate computations;
- Evaluating, thanks to Corollary 5.14, the size of a subinterval $[a, b]$ which can be treated at once by those algorithms; the general case will be treated in Subsection 5.5.2, whereas the case where f is entire allows for an asymptotic improvement in the estimates by letting ρ tend to infinity with d; we shall discuss this in Subsection 5.5.4;
- Investigating the interplay between d and ω_0, two parameters which have an impact on both the complexity and the quality of the final bound on $1/w$. (Subsection 5.5.3).

We start by giving complexity estimates for Algorithms 1 and 2.

5.5.1. *Complexity of Algorithms 1 and 2.* In this subsection, we denote $\mathsf{M}(n)$ the complexity of multiplying two n-bit integers (or two precision n floating-point numbers). Using naive arithmetic, we have $M(n) = O(n^2)$ whereas the best known bound as of today is $M(n) = O(n \log n)$, see [28].

We assume that interval evaluation at precision p of a function f uses $O(1)$ evaluations of f at precision p. In our implementation, we used the Arb library [33] and the MPFI library.

Lemma 5.15. *Put $\mathcal{M} = \max(u, v, |a|, |b|, \rho, B_f, \max_{[a,b]} |f'(x)|)$. The computations of Algorithm 1 on input $a, b, f, d, u, v, \rho, \omega_0$ can be made in floating-point precision $\mathfrak{p} = \mathsf{tprec} + O(d \log \mathcal{M})$.*

In particular, for fixed a, b, ρ, ω_0, f, for $u, v = 2^p$, with $p \geqslant d$, the required precision is $O(dp)$.

Proof. It is a corollary of Theorem C.9, see Appendix C. □

Proposition 5.16. *On input $a, b, f, d, u, v, \rho, \omega_0$, assuming that evaluating f in precision P costs $C_{f,P}$, and a DCT of size n in precision q has cost $O(n\mathsf{M}(q))$, Algorithm 1 has complexity*

$$O(d^4 \mathsf{M}(\mathfrak{p}) + d^2 C_{f,\mathfrak{p}}),$$

where $\mathfrak{p}$ is as in Lemma 5.15.

Proof. The most costly steps of Algorithm 1 appear in the loop 7-17 which can be performed using $O(d^2)$ evaluations of f at precision $O(\mathsf{tprec})$, and $O(d^4)$ multiplications of real numbers in precision $O(\mathsf{tprec})$, plus $O(d^2)$ DCTs of size N in precision tprec. □

In particular, for fixed a, b, ρ, $u = v = 2^p$, $p \geqslant d$, $C_{f,P} = \tilde{O}(\mathsf{M}(P))$, $\mathsf{M}(n) = \tilde{O}(n^\kappa)$, and ignoring the $\omega_0 \log \rho$ term in tprec, we obtain a complexity of $\tilde{O}(d^{4+\kappa} p^\kappa)$. Here, we use the $\tilde{O}(\cdot)$ notation defined as $f(n) = \tilde{O}(g(n))$ iff. there exists a nonnegative integer k such that $f(n) = O(g(n) \log^k g(n))$ (g is implicitly assumed to tend to $+\infty$ at ∞).

We now turn to the analysis of Algorithm 2. We shall limit ourselves to the analysis of Steps 1–6, which compute the two auxiliary polynomials. This is, in any case, the core of the algorithm, but also the choice made in previous papers, and thus allows for a better comparison.

Proposition 5.17. *On input $a, b, f, d, u, v, \rho, \omega_0$, under the assumption $C_{f,P} = O(P^2)$, Steps 1–6 of Algorithm 2 have complexity $O(d^6 \mathsf{M}(d^2)(d^2 + \mathfrak{p})\mathfrak{p})$.*

Proof. The main steps of Algorithm 2 are:

- A call to Algorithm 1;
- A call to LLL on a lattice of dimension N with entries of size $O(\mathfrak{p})$.

For the second part, we use the L^2 algorithm [54] on a lattice of dimension $N = O(d^2)$, embedded into $\mathbb{R}^{2N}$; we thus have complexity $O(d^6 \mathsf{M}(d^2)(d^2 + \mathfrak{p})\mathfrak{p})$. This cost dominates the cost of Algorithm 2. □

Note that in typical situations (for instance, either $u, v = 2^p, p \geqslant d$ or $\omega_0 \neq N - o(N)$, $\rho \geqslant 2$) we have $d^2 = O(\mathfrak{p})$ and the complexity simplifies to $O(d^6 \mathsf{M}(d^2)\mathfrak{p}^2)$.

5.5.2. *Number of subintervals for fixed d.* Thanks to the results of Subsection 5.4.2, we can estimate the maximum size of an interval $[a, b] \subset [\alpha, \beta]$, with α, β fixed, for which Algorithm 2 succeeds (in the sense of Corollary 5.14). This follows from Proposition 5.13, and yields at the same time the number of subintervals to be considered if one wants to deal with a full interval $[\alpha, \beta]$.

Theorem 5.18. *Given fixed f, a fixed parameter d, two fixed real numbers α and β, Problem 2.6 can heuristically be solved for $u, v \to \infty$ over $[\alpha, \beta]$ using*

$$O\left((\beta - \alpha)(uv)^{\frac{2Nd}{3(N(N-3)+2\omega_0+\lfloor\omega_0\rfloor(\lfloor\omega_0\rfloor-2\omega_0+1))}}\right) \tag{5.16}$$

calls to Algorithm 2 with parameter d.

We then obtain a value

$$w = O\left((uv)^{\frac{2N(N-\omega_0)d}{3(N(N-3)+2\omega_0+\lfloor\omega_0\rfloor(\lfloor\omega_0\rfloor-2\omega_0+1))}}\right). \tag{5.17}$$

When $d \to \infty$, both statements remain valid if $N - \omega_0 = \Theta(N)$, or if one replaces uv by $uv2^{O(d)}$.

Proof. This is a direct consequence of Proposition 5.13 and Corollary 5.14, where we choose $K = 2$. Recall that the heuristic nature of this result comes from the possibility that the two polynomials obtained in Algorithm 2 are not coprime, in which case one cannot recover the solutions X, Y from those two polynomials.

Finally, thanks to Corollary 5.11, the upper bound on w is $O(\rho^{(N-\omega_0)})$, from which the second part of the result follows.

For $d \to \infty$, we need to take into account the term $2^{6(d+3)} = 2^{O(d)}$ of Proposition 5.13. However, if $\omega - N = \Theta(N)$, the global exponent in Proposition 5.13 is $O(1/d)$ and, overall, this term is absorbed by the O notation. □

Remark 5.19. If one is only interested with the smallest possible complexity, it should be noted that the exponent in (5.16) is minimal for $\omega_0 \in [1,2]$, and equal in this case to $8N/(3(d+3)(d^2+3d-2))$; for $\omega_0 = 2$ we then get $w = O\left((uv)^{4N/(3(d+3))}\right)$.

Remark 5.20. In the case where ω_0 is an integer, the bounds take the nicer form

$$O\left((\beta-\alpha)(uv)^{\frac{2Nd}{3(N-\omega_0)(N+\omega_0-3)}}\right) \text{ and } w = O\left((uv)^{\frac{2Nd}{3(N+\omega_0-3)}}\right).$$

In particular, this shows the limits of the approach: for ω_0 close to N, we decrease the exponent of the bound on w by a factor of 2, but can expect nothing better. We shall see in the next section how to go beyond this limitation.

We now discuss the case where we let $d \to \infty$. This has two goals:

- See for what value of d we can expect to treat a whole interval $[\alpha, \beta]$ at once; notice that better results will be obtained later (Subsection 5.5.4) on if f is entire and we have control on its growth at infinity;
- Give a simplified form of the estimates of Theorem 5.18, which, in reason of the technical parameter ω_0, are rather unpleasant and unintuitive.

Corollary 5.21. *Let again α, β be fixed real numbers. For $d \to \infty$, for $\omega_0 = \lambda N(1+o(1))$, $\lambda \in [0,1)$, Problem 2.6 can heuristically be solved for $u, v \to \infty$ over $[\alpha, \beta]$ using*

$$O\left((\beta-\alpha)(uv)^{\frac{4(1+o(1))}{3(1-\lambda^2)d}}\right) \tag{5.18}$$

calls to Algorithm 2 with parameter d, giving a bound

$$w = O\left((uv)^{\frac{2d(1+o(1))}{3(1+\lambda)}}\right). \tag{5.19}$$

Remark 5.22. Note that if we assume $d = \Theta(\log(uv))$, Corollary 5.21 states that one call is enough to address an interval of the size $\beta - \alpha = O(1)$ and we then obtain $w = e^{O(\log^2(uv))}$. We will improve this result in Section 5.5.4 under additional assumptions on the growth of f at infinity.

Remark 5.23. In all this section, our complexity estimates should be considered as slightly pessimistic for fixed d, at least for usual transcendental functions. Indeed, we base our complexity estimates on estimates on the size of the *second* vector of an LLL-reduced basis, estimates which can only be obtained under a (trivial, thus pessimistic in practice) lower bound on the size of the first vector.

For a "classical" function such as exp or Γ, we notice in practice that most of the time, the second vector has a size similar to the size of the first one. This yields the slighly better bound

$$O\left((\beta-\alpha)(uv)^{\frac{2dN}{3(N(N-3)+2\omega_0+N\lfloor\omega_0\rfloor(\lfloor\omega_0\rfloor-2\omega_0+1))}}\right).$$

Note that it is easy to build examples where this latter bound does not hold, by taking a function which has a very good algebraic approximation (which gives a very short first vector) over the interval under study.

Remark 5.24. The first part of the Corollary, when $\lambda = 0$, is akin to, asymptotically, Bombieri and Pila's result [6] on the number of real algebraic curves of degree $\leqslant d$ containing all integer points on a given transcendental curve. The only reasons why we do not get the exact same result as theirs are: the fact that we use a bound on the second vector (see previous remark); and the fact that in order to get a practical

algorithm, we truncate our matrix to get an integer matrix – this has a slight effect, asymptotically negligible, on the final bound.

5.5.3. *Tuning d and ω_0.* Again, in order to ease this very technical discussion, we shall focus on the situation of a fixed f, a, b for $uv \to \infty$.

Let us start by pointing that a tedious, but not difficult computation shows that for $d \geqslant 2$ the exponent in (5.17) is decreasing for $\omega_0 \in [0, N-1)$; the maximal value of this exponent, for $\omega_0 = 0$, is $4dN/(3(d-1)(d+4)) \approx 2d/3$, whereas its minimal value, for ω_0 close to $N-1$, is $dN/(3N-6) \approx d/3$. We thus have a *wall-type phenomenon*: if we want to get access to a good complexity (see (5.16)), we need to increase d; but then ω_0 fails to prevent the degradation of the estimate on w, at least in a significant way. In practice, if we target a sharp bound and let ω_0 grow with this purpose in mind, we observe that the lattice basis reduction step decreases d to some δ on its own simply by not using monomials of degree $> \delta$ for the first vectors.

We shall however see that setting ω_0 to a non-zero value still allows one to get a better complexity/w compromise, and shall study a different method giving complete control on w in the next section.

From now on, we thus fix a value of $w = (uv)^\mu$ and try to find a pair (d, ω_0) which minimizes the complexity required to achieve this value of w.

We start with the asymptotic situation, i.e., $d \to \infty$.

Proposition 5.25. *Let $d \to \infty$, and $\omega_0 = \lambda N(1 + o(1))$. The value of d such that the exponent of uv in (5.19) is μ while minimizing the exponent in (5.18) is $d = 2\mu(1+o(1))$, obtained for $\lambda = 1/3(1+o(1))$. This gives a number of subintervals*

$$O\left((\beta-\alpha)(uv)^{\frac{3}{4\mu}(1+o(1)}\right).$$

Proof. Elementary calculus. □

Led by this asymptotic statement, we have computed (experimentally), for small values of μ, the value of d giving the best estimate for the complexity in (5.16); it turns out that in all our computations, the optimal value was $d = \lfloor 2\mu \rfloor$, except when $\mu = r + 1/2$ is an half-integer, where the optimal d is $2r$.

Working out a closed form for the exponent of the complexity estimate as a function of μ seems thus possible, but would be moderately enlightening; it seems preferable to give a plot of the corresponding function. Figure 1 gives three curves. The dashed curve corresponds to the best exponent in Theorem 5.18 as a function of the exponent of uv in the bound on w. The dotted curve represents a similar function, but using a version of our bounds controlling only the first vector of the lattice. Finally, the plain curve is the asymptotic bound $3/(4\mu)$.

5.5.4. *The case $\rho(b-a) \to \infty$.* In this subsection, we shall now let K depend on d, namely we shall let it tend to ∞ with d. We shall thus need the function under study to be an entire function. Recall that if $f : \mathbb{C} \to \mathbb{C}$ is an entire function, and if $\theta = \limsup_{\rho\to\infty} \log\log \max_{|z|\leqslant\rho} |f(z)| / \log\rho$ is finite, the function f is said to have finite order θ.

The presence of a term $M_{\rho,a,b}(f)$, depending on the growth of f at infinity, shows that it is difficult to give a single ready-to-use result. We thus split the discussion into two parts: the case of entire functions of finite order, such as exp for instance, in which the value of the order gives sufficiently precise information on the growth

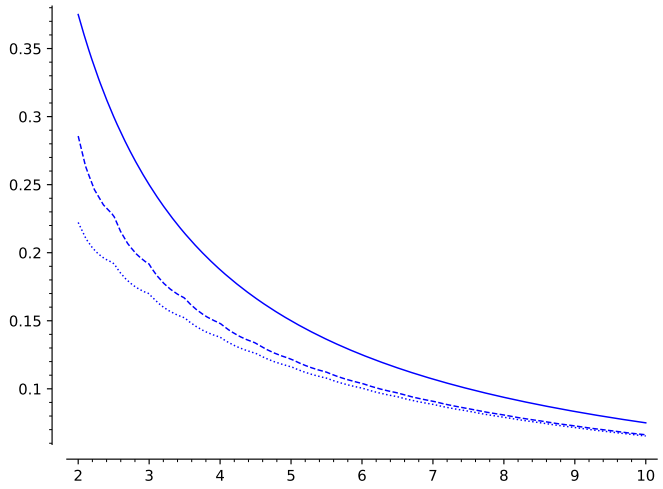

FIGURE 1. Exponent estimates

at infinity to obtain a general result; and then two examples of a function of order zero and of a function of order infinity – in those cases, case-by-case estimates of the growth at infinity are required.

Theorem 5.26. *Let $a < b$ two fixed real numbers, let θ be a positive real number, and let f be an entire function of finite order $\leqslant \theta$, $d \geqslant 2$ be an integer, $N = (d+1)(d+2)/2$, and let $\omega_0 = \lambda N + o(N)$, for some fixed $\lambda \in [0,1)$.*

For $uv \to \infty$, for any constant $\nu > \frac{4\theta}{3(1-\lambda^2)}$, for

$$d = \nu \frac{\log(uv)}{\log\log(uv)}(1+o(1)),$$

Algorithm 2 succeeds (in the sense of Corollary 5.14) over $[a,b]$ and yields a bound on w of the form:

$$w \leqslant (uv)^{\frac{\nu^2(1-\lambda)}{2\theta}\frac{\log(uv)}{\log\log(uv)}(1+o(1))}.$$

Proof. Let $\theta' > \theta$ be a parameter which will be fixed later on. We choose $K = d^{1/\theta'}/2$, which is $> 2(b-a)$ for d large enough. The disc $\mathcal{D}_{a,b,K}$ (see Proposition 5.13) is, for our choice of ρ, included into the ball $B(0, d^{1/\theta'})$ for d large enough; hence, the assumption that f has order $\leqslant \theta$ shows that there exist constants $C_{f,\theta'}, \sigma_{f,\theta'} \in \mathbb{R}$ such that

$$M_{\mathcal{D}_{a,b,K}}(f) \leqslant C_{f,\theta'} \exp(\sigma_{f,\theta'} d) = 2^{O(d)}.$$

Proposition 5.13 then implies that a sufficient condition for the conclusion of the theorem to hold is

$$b-a < \frac{1}{2}\Bigg(\underbrace{2^{6(d+3)}\left(|a+b| + \frac{d^{1/\theta'}}{2}\right) M_{\mathcal{D}_{a,b,K}}(f)}_{=:A_d}\Bigg)^{-4/(3d(1-\lambda^2))(1+o(1))} d^{1/\theta'}(uv)^{-4/(3d(1-\lambda^2))(1+o(1))},$$

or equivalently

$$d^{1/\theta'}(uv)^{-4/(3d(1-\lambda^2))(1+o(1))} > 2A_d^{4/(3d(1-\lambda^2))(1+o(1))}(b-a).$$

As a, b are fixed and $A_d = 2^{O(d)}$ when $d \to \infty$, the right hand side is bounded and a sufficient condition for this to hold is simply

$$d^{1/\theta'}(uv)^{-4/(3d(1-\lambda^2))(1+o(1))} \to \infty,$$

for which it suffices that, for some $\varepsilon' > 0$,

$$d \log d \geqslant \left(\frac{4\theta'}{3(1-\lambda^2)} + \varepsilon'\right) \log(uv),$$

which obviously holds under the assumption on d made in the theorem for uv large enough and $\theta' < 3(1-\lambda^2)\nu/4$.

As for the last part, the bound on w is $\leqslant \rho^{N(1-\lambda)} = K^{N(1-\lambda)(1+o(1))}$, namely we have

$$\log(w) = \frac{(1-\lambda)d^2 \log d}{2\theta'}(1+o(1)) \leqslant \frac{(1-\lambda)\nu^2 \log^2(uv)}{2\theta \log\log uv}(1+o(1)),$$

as claimed. □

In particular, for the TMD over $[1/4, 1/2)$ for the exponential function ($\theta = 1$), hence for $a = 1/4, b = 1/2$ and $u = 2^{p+1}$ and $v = 2^{p-1}$, for $p \to \infty$, we obtain the condition $d \geqslant \left(\frac{8\log 2}{3(1-\lambda^2)} + \varepsilon\right)\frac{p}{\log p}$ for the full interval $[1/4, 1/2)$, with a bound $w \leqslant 2^{\left(\frac{32\log 2}{9(1-\lambda^2)(1+\lambda)} + \varepsilon\right)\frac{p^2}{\log p}}$.

Two other examples. We illustrate, more generally, the fact that we get asymptotic results depending on the rate of growth of f at infinity: the slower the growth of f, the better the performance.

Theorem 5.27. *Let $f = \exp(\exp(z))$. For uv large enough, for any constant $\nu > \frac{4}{3(1-\lambda^2)}$, for $d = \lceil \nu \frac{\log(uv)}{\log\log\log(uv)} \rceil$, Algorithm 2 succeeds in the sense of Corollary 5.14 and we obtain*

$$w = (uv)^{\nu^2(1-\lambda)\frac{\log(uv)\log\log(uv)}{(\log\log\log(uv))^2}(1+o(1))}.$$

Let $g(z) = \sum_{n\geqslant 0} \exp(-n^2)z^n$. For uv large enough, for any constant $\nu > \frac{4}{3(1-\lambda^2)}$, for $d = \lceil \nu\sqrt{\log(uv)} \rceil$, Algorithm 2 succeeds in the sense of Corollary 5.14 and we obtain $w = (uv)^{\frac{3}{2}\nu^2(1-\lambda^2)(1-\lambda)\sqrt{\log(uv)}(1+o(1))}$.

Proof. The proof is similar to the proof of the previous theorem, with $K = \log d$ for f and $K = \exp(3(1-\lambda^2)d/2)$ for g. See Appendix B. □

6. The two-variable method

We consider $u, v \in \mathbb{N} \setminus \{0\}$, $a_1 < b_1$ and $a_2 < b_2$, and $f : [a_1, b_1] \to \mathbb{R}$ a function that is analytic in a neighbourhood of $[a_1, b_1]$. In this section, we develop a heuristic algorithmic approach to determine the integers X, Y such that

$$X/u \in [a_1, b_1] \text{ and } a_2 < Y/v - f(X/u) < b_2. \tag{6.1}$$

Note that this is a mere reformulation of Problem 2.6: let $w \in \mathbb{N} \setminus \{0\}$, we set $[a, b] = [a_1, b_1]$ and $b_2 = -a_2 = 1/w$. Actually, without loss of generality, we can assume $b_2 = -a_2$ by replacing f by $f + (a_2 + b_2)/2$.

We prefer to keep a_2 and b_2 arbitrary in the sequel to put more emphasis on the symmetry between (a_1, b_1) and (a_2, b_2) in the formulas and statements.

Our approach aims at building a trap for these pairs (X, Y). We compute, by combining two-dimensional Chebyshev interpolation and lattice reduction, two polynomials $P_0, P_1 \in \mathbb{Z}[X_1, X_2]$ such that, for $i = 0, 1$, for all $x \in [a_1, b_1]$, $t \in [a_2, b_2]$, we have $|P_i(ux, v(f(x)+t))| < 1$. Let $X \in \mathbb{Z}$ be such that $X/u =: x_0 \in [a_1, b_1]$ and let $Y \in \mathbb{Z}$ be such that $Y/v =: f(x_0)+t_0$ with $t_0 \in [a_2, b_2]$. Then $P_i(ux_0, v(f(x_0)+t_0)) = P_i(X, Y) \in \mathbb{Z} \cap (-1, 1) = \{0\}$, that is to say (X, Y) is a common root to P_0 and P_1. As in Section 5, we use our heuristic assumption: P_0 and P_1 are supposed to have no nonconstant common factor. We eliminate one of the variables and get the list of all the integers X, Y that satisfy (6.1).

In the sequel of this section, we start with estimates of the determinants of the lattices that we use, we present our algorithm, the proof of its correctness and analyse its complexity. When the proofs of the statements are similar to the ones presented in Section 5, we shall postpone them to Appendix E.

Throughout this section, $N_1, N_2 \geqslant 2$ and $N \geqslant 2$, will be three integers. In order to avoid degenerate situations and trivial output, we shall always assume $N_1 N_2 \geqslant N$.

6.1. **Volume estimates for rigorous interpolants at the Chebyshev nodes.** We start by introducing the two dimensional extension of the DCT-II:

$$\begin{array}{rccc} \text{2D-DCT-II}: & \mathbb{R}^{N_1} \times \mathbb{R}^{N_2} & \to & \mathbb{R}^{N_1} \times \mathbb{R}^{N_2} \\ & (x_{\ell_1,\ell_2})_{\substack{0 \leqslant \ell_1 \leqslant N_1-1 \\ 0 \leqslant \ell_2 \leqslant N_2-1}} & \mapsto & (X_{k_1,k_2})_{\substack{0 \leqslant k_1 \leqslant N_1-1 \\ 0 \leqslant k_2 \leqslant N_2-1}} \end{array}$$

with

$$X_{k_1,k_2} = \sum_{0 \leqslant \ell_1 \leqslant N_1-1} \sum_{0 \leqslant \ell_2 \leqslant N_2-1} x_{\ell_1,\ell_2} \cos\left(\frac{k_1(\ell_1+1/2)\pi}{N_1}\right) \cos\left(\frac{k_2(\ell_2+1/2)\pi}{N_2}\right),$$

for $k_1 = 0, \ldots, N_1 - 1, k_2 = 0, \ldots, N_2 - 1$.

Let $N \in \mathbb{N}, N \geqslant 2$, let $i = 0, \ldots, N-1$, let f_i a function defined over $[a_1, b_1] \times [a_2, b_2]$. We shall use the following results for the functions f_i defined in (6.8). If we interpolate f_i by $Q_i(x, t) \in \mathbb{R}_{N_1-1,N_2-1}[x, t]$[14] at pairs of Chebyshev nodes $\mu_{k_1,k_2} = (\mu_{k_1,N_1-1,[a_1,b_1]}, \mu_{k_2,N_2-1,[a_2,b_2]})_{\substack{0 \leqslant k_1 \leqslant N_1-1 \\ 0 \leqslant k_2 \leqslant N_2-1}}$, cf. Section 4.1, we have the following expressions for the interpolation polynomials (the proof is identical to the one variable case [50, Chap. 6]), for $i = 0, \ldots, N-1$:

$$Q_i(x, t) = \sum_{k_1=0}^{N_1-1}{}' \sum_{k_2=0}^{N_2-1}{}' c_{k_1,k_2,i} T_{k_1,[a_1,b_1]}(x) T_{k_2,[a_2,b_2]}(t) \in \mathbb{R}_{N_1-1,N_2-1}[x, t]$$

with

$$(c_{k_1,k_2,i})_{\substack{0 \leqslant k_1 \leqslant N_1-1 \\ 0 \leqslant k_2 \leqslant N_2-1}} = \frac{4}{N_1 N_2} \text{2D-DCT-II}\left((f_i(\mu_{N_1-1-\ell_1,N_2-1-\ell_2}))_{\substack{0 \leqslant \ell_1 \leqslant N_1-1 \\ 0 \leqslant \ell_2 \leqslant N_2-1}}\right). \tag{6.2}$$

Let $\rho_1, \rho_2 > 1$, $a_1 < b_1$, $a_2 < b_2$, we recall from Section 4.1.2, $\mathcal{E}_{\rho_1,a_1,b_1,\rho_2,a_2,b_2} = \mathcal{E}_{\rho_1,a_1,b_1} \times \mathcal{E}_{\rho_2,a_2,b_2}$ and $E_{\rho_1,a_1,b_1,\rho_2,a_2,b_2} = E_{\rho_1,a_1,b_1} \times E_{\rho_2,a_2,b_2}$. We also recall $M_{\rho_1,a_1,b_1,\rho_2,a_2,b_2}(g) := \max_{z \in \mathcal{E}_{\rho_1,a_1,b_1,\rho_2,a_2,b_2}} |g(z)|$ if g is analytic in a neighbourhood

[14] This denotes the set of polynomials in two indeterminates x and t with real coefficients, degree in x less than N_1 and degree in t less than N_2.

of $E_{\rho_1,a_1,b_1,\rho_2,a_2,b_2}$. Let $N_1, N_2 \geqslant 2$, and $N \leqslant N_1 N_2$. Let $f_0, \ldots, f_{N-1}$ be functions analytic in a neighbourhood of $E_{\rho_1,a_1,b_1,\rho_2,a_2,b_2}$. We introduce the $N \times (N_1N_2 + N)$ matrix $A = (A_1|A_2)$, defined by

$$\begin{aligned}(A_1)_{i,(k_1,k_2)} &= \left(\frac{c_{k_1,k_2,i}}{2^{\delta_{0k_1}+\delta_{0k_2}}}\right)_{\substack{0\leqslant i\leqslant N-1\\ 0\leqslant k_1\leqslant N_1-1,\, 0\leqslant k_2\leqslant N_2-1}},\\ (A_2)_{i,j} &= \delta_{ij}\frac{16\rho_1\rho_2 M_{\rho_1,a_1,b_1,\rho_2,a_2,b_2}(f_i)}{(\rho_1-1)(\rho_2-1)}\left(\frac{1}{\rho_1^{N_1}}+\frac{1}{\rho_2^{N_2}}\right), 0 \leqslant i,j \leqslant N-1.\end{aligned} \tag{6.3}$$

Recall from Proposition 4.2 that $\|f_i - Q_i\|_\infty \leqslant A_2[i,i]$, $i = 0, \ldots, N-1$.

Once again, the diagonal right part, A_2, of the matrix will be used for controlling that the functions $P_0(ux, v(f(x)+t)), P_1(ux, v(f(x)+t))$, output by the lattice basis reduction process, are uniformly small; this accounts for the presence of the $\frac{16\rho_1\rho_2 M_{\rho_1,a_1,b_1,\rho_2,a_2,b_2}(f_i)}{(\rho_1-1)(\rho_2-1)}\left(\frac{1}{\rho_1^{N_1}}+\frac{1}{\rho_2^{N_2}}\right)$ remainder term for the approximation of $P_i(ux, v(f(x)+t))$ by its interpolation polynomial at the 2D Chebyshev points.

We start with a convenient combinatorial lemma.

Lemma 6.1. *Let $\gamma \in \mathbb{R}, \gamma \geqslant 1$, N, N_1, N_2 positive integers. Consider the multiset*[15] *$S = \{k + \gamma k', (k,k') \in [0, N_1-1] \times [0, N_2-1]\} + \{\underbrace{N_1, \ldots, N_1}_{N \text{ times}}\}$, and order the elements of S as $\sigma_0 \leqslant \ldots \leqslant \sigma_{\operatorname{card} S - 1}$.*

Define

$$\Omega_\gamma(N_1, N_2, N) = \sigma_0 + \cdots + \sigma_{N-1},$$

the sum of the N smallest elements of S.

Let now $s \leqslant N_1$ be a real number, let $\mathcal{K}_s = \{(i,j) \in [0, N_1-1] \times [0, N_2-1] : i + \gamma j \leqslant s\}$. Then,

$$\Omega_\gamma(N_1, N_2, N) \geqslant s(N - \operatorname{card}\mathcal{K}_s) + \sum_{(i,j)\in\mathcal{K}_s}(i + \gamma j).$$

Proof. Put $\mathcal{M}_s = \{i + j\gamma, (i,j) \in \mathcal{K}_s\}$. Then, if $\operatorname{card}\mathcal{K}_s \leqslant N$, $\mathcal{M}_s \subset \{\sigma_0, \ldots, \sigma_{N-1}\}$, and any element in $\{\sigma_0, \ldots, \sigma_{N-1}\} \setminus \mathcal{M}_s$ is at least equal to s.

Otherwise, we have $\{\sigma_0, \ldots, \sigma_{N-1}\} \subset \mathcal{M}_s$, and any element in $\mathcal{M}_s \backslash \{\sigma_0, \ldots, \sigma_{N-1}\}$ is at most equal to s. □

Remark 6.2. The quantity $\Omega_\gamma(N_1, N_2, N)$ plays a key role in the analysis of our bivariate method, as $\rho_1^{\Omega_\gamma(N_1,N_2,N)}$ will turn to play the role that $\rho^{N(N-1)/2}$ played in the univariate case. For fixed values of N, N_1, N_2, γ, it is easy to compute explicit values of $\Omega_\gamma(N, N_1, N_2)$. We thus focus in the sequel on the asymptotic (for $N \to \infty$) behaviour of $\Omega_\gamma(N, N_1, N_2)$ and shall hence mostly study the asymptotic behaviour of this bivariate method – even though the analysis itself is not asymptotic by nature (see e.g. Theorem 6.4).

We now give explicit expressions for $\operatorname{card}\mathcal{K}_s$ and $\sum_{(i,j)\in\mathcal{K}_s}(i + j\gamma)$.

Lemma 6.3. *Let $s \in \mathbb{R}$, $s < N_1$ and $\gamma \geqslant N_1/N_2 \geqslant 1$. We have*

$$\operatorname{card}\mathcal{K}_s = \sum_{j=0}^{\lfloor s/\gamma\rfloor}(1 + \lfloor s - j\gamma\rfloor) = (1 + \lfloor s/\gamma\rfloor) + \sum_{j=0}^{\lfloor s/\gamma\rfloor}\lfloor s - j\gamma\rfloor \tag{6.4}$$

[15] By sum of multisets, we mean that the multiplicity of an element of the union is the sum of its multiplicities in the multisets.

and

$$\sum_{(i,j)\in\mathcal{K}_s}(i+j\gamma)=\sum_{j=0}^{\lfloor s/\gamma\rfloor}(1+\lfloor s-j\gamma\rfloor)\left(j\gamma+\frac{\lfloor s-j\gamma\rfloor}{2}\right). \tag{6.5}$$

This implies

$$(1+\lfloor s/\gamma\rfloor)(s-\gamma\lfloor s/\gamma\rfloor/2)\leqslant\operatorname{card}\mathcal{K}_s\leqslant(1+\lfloor s/\gamma\rfloor)(1+s-\gamma\lfloor s/\gamma\rfloor/2) \tag{6.6}$$

and

$$\begin{aligned}(1+\lfloor s/\gamma\rfloor)\frac{6s(s-1)+\gamma\lfloor s/\gamma\rfloor(3-\gamma-2\gamma\lfloor s/\gamma\rfloor)}{12}&\leqslant\sum_{(i,j)\in\mathcal{K}_s}(i+j\gamma)\\&\leqslant(1+\lfloor s/\gamma\rfloor)\frac{6s(s+1)+\gamma\lfloor s/\gamma\rfloor(3-\gamma-2\gamma\lfloor s/\gamma\rfloor)}{12}.\end{aligned} \tag{6.7}$$

Proof. First, note that $s/\gamma<N_1/\gamma\leqslant N_1/(N_1/N_2)=N_2$, hence $\lfloor s/\gamma\rfloor\leqslant N_2-1$. Let $(i,j)\in\mathcal{K}_s$, the largest possible value of j corresponds to the case $i=0$: we then have $j\gamma\leqslant s$, that is to say $j\leqslant\lfloor s/\gamma\rfloor$. Now, in order to count the elements of K_s, we enumerate, for $j=0,\dots,\lfloor s/\gamma\rfloor$, the elements of each slice $\{i+j\gamma\leqslant s, i\in[0,N_1-1]\}$: there are $1+\lfloor s-j\gamma\rfloor$ such elements, which proves (6.4).

Now, for $j=0,\dots,\lfloor s/\gamma\rfloor$, we sum the values $i+j\gamma$ for i in the slice $\{i+j\gamma\leqslant s, i\in[0,N_1-1]\}$, i.e., $i\in[0,\lfloor s-j\gamma\rfloor]$. Hence, for $j=0,\dots,\lfloor s/\gamma\rfloor$, we sum the values $\lfloor s-j\gamma\rfloor(1+\lfloor s-j\gamma\rfloor)/2+(1+\lfloor s-j\gamma\rfloor)j\gamma$, from which (6.5) follows.

We use $s-j\gamma-1\leqslant\lfloor s-j\gamma\rfloor\leqslant s-j\gamma$ for $j=0,\dots,\lfloor s/\gamma\rfloor$ to derive from (6.4)

$$(1+\lfloor s/\gamma\rfloor)(s-\gamma\lfloor s/\gamma\rfloor/2)\leqslant\operatorname{card}\mathcal{K}_s\leqslant(1+\lfloor s/\gamma\rfloor)(1+s-\gamma\lfloor s/\gamma\rfloor/2).$$

Likewise, we derive from (6.5)

$$\sum_{j=0}^{\lfloor s/\gamma\rfloor}\frac{(s-j\gamma)(s+j\gamma-1)}{2}\leqslant\sum_{(i,j)\in\mathcal{K}_s}(i+j\gamma)\leqslant\sum_{j=0}^{\lfloor s/\gamma\rfloor}\frac{(1+s-j\gamma)(s+j\gamma)}{2},$$

which yields (6.7). □

The next result gives an upper bound for the volume of the lattice generated by the rows of A:

Theorem 6.4. *Let $\rho_1,\rho_2>1$, $a_1<b_1$, $a_2<b_2$. We further assume that $\rho_1^{N_1}\leqslant\rho_2^{N_2}$, and define $\gamma=\log\rho_2/\log\rho_1$. Let $f_0,\dots,f_{N-1}$ be functions analytic in a neighbourhood of $E_{\rho_1,a_1,b_1,\rho_2,a_2,b_2}$. Then, we have*

$$(\det AA^t)^{1/2}\leqslant\left(32\sqrt{N}\right)^N 2^{N_1N_2}\left(\frac{\rho_1\rho_2}{(\rho_1-1)(\rho_2-1)}\right)^N\frac{\prod_{i=0}^{N-1}M_{\rho_1,a_1,b_1,\rho_2,a_2,b_2}(f_i)}{\rho_1^{\Omega_\gamma(N,N_1,N_2)}}.$$

Proof. See Appendix E. □

We now give two statements on the behaviour of $\Omega_\gamma(N,N_1,N_2)$ when $N\to\infty$, for a fixed (essentially optimal) choice of N_1, N_2. The proofs are elementary, but long and we postpone them to Appendix D.

Proposition 6.5. *Let φ be the function from $[1,+\infty)$ to $[1,+\infty)$ defined by $\varphi(x)=(1+\lfloor x\rfloor)(x-\lfloor x\rfloor/2)$. Then φ is invertible. We further define ψ by*

$$\psi(x)=\frac{1+\lfloor\varphi^{-1}(x)\rfloor}{12x}\left(6\varphi^{-1}(x)^2-\lfloor\varphi^{-1}(x)\rfloor-2\lfloor\varphi^{-1}(x)\rfloor^2\right);$$

we then have, for any $y \in [1, +\infty)$,

$$\psi^{-1}(y) = \frac{k+1}{2}\left(2y - k + \sqrt{4y(y-k) + 2k(2k+1)/3}\right),$$

where $k = \lfloor 3y/2 + 1/4 \rfloor$. *Further, when* $x \to \infty$, $\varphi^{-1}(x) = \sqrt{2x} + O(1)$, $\psi(x) = 2\sqrt{2x}/3 + O(1)$.

Proof. See Corollary D.3. □

Note that for $x \geqslant 1$, we prove in Lemma D.4 the inequalities $\psi(x) - 2\sqrt{2x}/3 \in [-5/6, 0]$ and observe numerically that $\psi(x) - 2\sqrt{2x}/3 \in [-1/2, -0.44]$, meaning that for our purposes $\psi(x)$ is very well approximated by $2\sqrt{2x}/3 - 1/2$.

Proposition 6.6. *Let* $\gamma \in \mathbb{R}$ *such that* $3 \leqslant \gamma \leqslant N$. *Put* $N_1 = \lfloor \sqrt{2N\gamma} \rfloor$ *and* $N_2 = \lceil \sqrt{2N/\gamma} \rceil$. *Then, we have* $\gamma \geqslant N_1/N_2$ *and*

$$\Omega_\gamma(N, N_1, N_2) = \psi(N/\gamma)N\gamma + O(N).$$

In particular, for $\gamma = o(N)$, *we have*

$$\Omega_\gamma(N, N_1, N_2) = \frac{2\sqrt{2}}{3}N^{3/2}\gamma^{1/2} + O(N\gamma).$$

Proof. See Corollary D.3. □

Remark 6.7. We can obtain a similar result for $1 < \gamma < 3$ if we set $N_1 = 1 + \lfloor \sqrt{2N\gamma} \rfloor$ and $N_2 = 1 + \lceil \sqrt{2N/\gamma} \rceil$.

This allows us to give asymptotic versions of Theorem 6.4, which will be more convenient in the sequel.

Corollary 6.8. *Let* $\rho_1, \rho_2 > 1$ *such that* $\gamma = \log \rho_2 / \log \rho_1 \in [3, N]$. *Let* $a_1 < b_1$, $a_2 < b_2$, $s = \gamma\varphi^{-1}(N/\gamma)$, $N_1 = \lfloor \sqrt{2N\gamma} \rfloor$ *and* $N_2 = \lceil \sqrt{2N/\gamma} \rceil$. *Let* $f_0, \ldots, f_{N-1}$ *be functions analytic in a neighbourhood of* $E_{\rho_1,a_1,b_1,\rho_2,a_2,b_2}$.

Assume that $N \to \infty$, *we obtain*

$$(\det AA^t)^{1/2} \leqslant 2^{O(N \log N)} \left(\frac{\rho_1\rho_2}{(\rho_1 - 1)(\rho_2 - 1)}\right)^N \frac{\prod_{i=0}^{N-1} M_{\rho_1,a_1,b_1,\rho_2,a_2,b_2}(f_i)}{\rho_1^{\psi(N/\gamma)N\gamma + O(N)}}.$$

Again, we specialize this statement to the case of the ordered list of functions

$$[f_i, 0 \leqslant i \leqslant (d+1)(d+2)/2 - 1] = [x \mapsto u^k x^k v^\ell (f(x) + t)^\ell, \ell = 0, \ldots, d, k = 0, \ldots, d - \ell]. \tag{6.8}$$

Corollary 6.9. *Let* $\rho_1, \rho_2 > 1$ *such that* $\gamma = \log \rho_2 / \log \rho_1 \in [3, N]$. *Let* $a_1 < b_1$, $a_2 < b_2$, $s = \gamma\varphi^{-1}(N/\gamma)$, $N_1 = \lfloor \sqrt{2N\gamma} \rfloor$ *and* $N_2 = \lceil \sqrt{2N/\gamma} \rceil$. *Let* f *be a function analytic in a neighbourhood of* E_{ρ,a_1,b_1}. *Define*

$$f_{k,\ell}(x, t) = u^k x^k v^\ell (f(x) + t)^\ell, 0 \leqslant \ell \leqslant d, 0 \leqslant k \leqslant d - \ell,$$

the matrices A_1, A_2, $A=(A_1|A_2)$ as in (6.3), and the quantity $\Delta_{N,N_1,N_2,[a_1,b_1],[a_2,b_2],\rho_1,\rho_2} := (\det AA^t)^{1/2}$. We have, as $d\to+\infty$,

$$\begin{aligned}(6.9)\quad \Delta^{1/(N-1)}_{N,N_1,N_2,[a_1,b_1],[a_2,b_2],\rho_1,\rho_2} \leqslant 2^{O(1)}&\left(\sqrt{N}\frac{\rho_1\rho_2}{(\rho_1-1)(\rho_2-1)}\right)^{1+o(1)}\\ &\frac{(uv)^{d/3+O(1)}}{\rho_1^{\psi(N/\gamma)\gamma+O(1)}}\left(\frac{b_1-a_1}{2}\left(\frac{\rho_1+\rho_1^{-1}}{2}\right)+\left|\frac{b_1+a_1}{2}\right|\right)^{d/3+O(1)}\\ &\left(M_{\rho_1,a_1,b_1}(f)+\frac{b_2-a_2}{2}\left(\frac{\rho_2+\rho_2^{-1}}{2}\right)+\left|\frac{b_2+a_2}{2}\right|\right)^{d/3+O(1)}.\end{aligned}$$

Proof. Note that each $f_{k,\ell}$ is analytic in a neighbourhood of $E_{\rho_1,a_1,b_1,\rho_2,a_2,b_2}$. Also, since $\sum_{0\leqslant k+\ell\leqslant d} k = \sum_{0\leqslant k+\ell\leqslant d}\ell = dN/3$, the exponent of uv, $\frac{b_1-a_1}{2}\left(\frac{\rho_1+\rho_1^{-1}}{2}\right)+\left|\frac{b_1+a_1}{2}\right|$ and $M_{\rho_1,a_1,b_1}(f)+\frac{b_2-a_2}{2}\left(\frac{\rho_2+\rho_2^{-1}}{2}\right)+\left|\frac{b_2+a_2}{2}\right|$ is $\frac{dN}{3(N-1)}=\frac{d}{3}+O(1)$. □

6.2. **Statement of the algorithms.** Our main routine is Algorithm 4. It comes together with Algorithm 3 that mainly constructs the lattice to be reduced in Algorithm 4.

Let $[R_i, i=0,\dots,N-1]$ be the ordered list

$$\left[\frac{16\rho_1\rho_2u^kM_{\rho_1,a,b}(x)^kv^\ell M_{\rho_1,a_1,b_1,\rho_2,a_2,b_2}(f(x)+t)^\ell}{(\rho_1-1)(\rho_2-1)}\left(\frac{1}{\rho_1^{N_1}}+\frac{1}{\rho_2^{N_2}}\right);\ \ell=0,\dots,d, k=0,\dots,d-\ell\right].$$

Let $A=(A_1|A_2)$ and $\hat{A}=\left(\hat{A}_1|\hat{A}_2\right)$ be the $N\times(N_1N_2+N)$ matrices defined by

$$A_1=\left(\frac{c_{k_1,k_2,i}}{2^{\delta_{0k_1}+\delta_{0k_2}}}\right)_{\substack{0\leqslant i\leqslant N-1\\ 0\leqslant k_1\leqslant N_1-1,\,0\leqslant k_2\leqslant N_2-1}},\quad A_2=(\delta_{ij}R_i)_{0\leqslant i,j\leqslant N-1},$$

$$\hat{A}_1=\left(\left[2^{\mathsf{tprec}}A_1[i,j]\right]_0/2^{\mathsf{tprec}}\right)_{\substack{0\leqslant i\leqslant N-1\\ 0\leqslant j\leqslant N_1N_2-1}},\quad \hat{A}_2=\left(\lfloor 2^{\mathsf{tprec}}A_2[i,j]\rfloor/2^{\mathsf{tprec}}\right)_{0\leqslant i,j\leqslant N-1},$$

where[16] $\mathsf{tprec}=\lceil -\log_2(\min_{0\leqslant i\leqslant N-1}A_2[i,i])+\log_2(N)\rceil+2$. The reasons for introducing $\hat{A}$ and tprec are the same as in Section 5.

By construction, $\left|\hat{A}[i,j]\right|\leqslant|A[i,j]|$ for all i,j. Hence, Theorem 6.4 and its corollaries, which proceed by upper bounding the coefficients of A and applying Theorem 5.1, also hold for $(\det\hat{A}\hat{A}^t)^{1/2}$.

The rows of $\hat{A}$ generate the lattice that will be reduced in our algorithm.

Lemma 6.10. *The $\mathbb{Z}$-module generated by the rows of $\hat{A}$ is a lattice of rank N.*

Proof. Identical to the proof of Lemma 5.5. □

The matrices M_c and M_r computed in Algorithm 3 correspond to the scaled matrices $2^{\mathsf{tprec}}\hat{A}_1$ and $2^{\mathsf{tprec}}\hat{A}_2$.

We now derive an explicit expression for tprec in this bivariate context. In order to do so, we assume that the set $u[a_1,b_1]$, resp. $v(f([a_1,b_1])+[a_2,b_2])$, contains at least one nonzero integer n_x, resp. n_f. Again, this assumption is made without loss

[16] We shall prove in Lemma 6.11 that this value coincides with the definition of tprec at Step 1 of Algorithm 3.

of generality with respect to our problem, since if the assumption does not hold the problem is trivial.

Lemma 6.11. *We have*

$$\begin{aligned}\mathsf{tprec} &= \lceil -\log_2(R_0) + \log_2(N)\rceil + 2\\ &= \left\lceil \log_2(1-1/\rho_1) + \log_2(1-1/\rho_2) - \log_2(\rho_1^{-N_1}+\rho_2^{-N_2}) + \log_2(N)\right\rceil - 2\end{aligned}$$

and

$$\frac{N(\rho_1-1)(\rho_2-1)\rho_1^{N_1-1}\rho_2^{N_2-1}}{4(\rho_1^{N_1}+\rho_2^{N_2})} \leqslant 2^{\mathsf{tprec}} \leqslant \frac{N(\rho_1-1)(\rho_2-1)\rho_1^{N_1-1}\rho_2^{N_2-1}}{2(\rho_1^{N_1}+\rho_2^{N_2})}.$$

Proof. Under our assumption, it comes $v(M_{\rho_1,a_1,b_1,\rho_2,a_2,b_2}(f(x)+t)) \geqslant |n_f| \geqslant 1$ and $uM_{\rho_1,a_1,b_1}(x) \geqslant |n_x| \geqslant 1$. It then follows $R_i \geqslant 16\rho_1\rho_2(\rho_1^{-N_1}+\rho_2^{-N_2}))/((\rho_1-1)(\rho_2-1)) = R_0$ for all i. Therefore, we get $\mathsf{tprec} = \lceil -\log_2(R_0)+\log_2(N)\rceil + 2$. □

6.3. **Practical remarks.** All practical details and optimizations mentioned in Section 5.3 apply *mutatis mutandis* to Algorithms 3 and 4: optimization of the construction of the matrix using properties of the DCT (Section 5.3.1), overestimation issues (Section 5.3.2), rounding issues (Section 5.3.3), use of Newton polynomials (Section 5.3.4). Concerning Section 5.3.5, one should replace (5.9) by the following inequality, cf. proof of Theorem 6.12:

$$\max_{i=0,1}\left(\|(M_{LLL}[i,j])_{0\leqslant j\leqslant N+N_1N_2-1}\|_1 + (N+N_1N_2)\frac{\|(M_{LLL}[i,j])_{N_1N_2\leqslant j\leqslant N+N_1N_2-1}\|_1}{4N}\right) < 2^{\mathsf{tprec}}. \tag{6.10}$$

6.4. **Proof of correctness.** We shall now prove the correctness of Algorithm 4.

6.4.1. *Uniformly small polynomials in the vicinity of a transcendental analytic curve.* We now state a key result for the proof of Algorithm 4.

Theorem 6.12. *Let $d \geqslant 1$, $m \geqslant 2$ be two integers, $N = (d+1)(d+2)/2$, $u, v > 0$ and $(f_j)_{1\leqslant j\leqslant N} = (u^k x^k v^\ell (f(x)+t)^\ell)_{0\leqslant k+\ell\leqslant d}$. Let $\rho_1,\rho_2 > 1$, $a_1 < b_1$, $a_2 < b_2$, $N_1, N_2 \geqslant 2$, and $N \leqslant N_1N_2$. Let $\Lambda = (\lambda_{k,\ell})_{0\leqslant k+\ell\leqslant d} \in \mathbb{Z}^N$ be such that $\|\Lambda\hat{A}\|_2 \leqslant 1/(N+N_1N_2)$, and let $P(X,Y) = \sum_{0\leqslant k+\ell\leqslant d}\lambda_{k,\ell}X^kY^\ell$, we have*

$$\max_{\substack{x\in[a_1,b_1]\\ t\in[a_2,b_2]}} |P(ux, v(f(x)+t))| < 1.$$

Proof. See Appendix E. □

Remark 6.13. The proof of Theorem 6.12 yields in particular that

$$\|\Lambda A\|_1 \geqslant \sum_{0\leqslant k+\ell\leqslant d} |\lambda_{k,\ell}| \underbrace{\frac{16u^kM_{\rho_1,a_1,b_1}(x)^kv^\ell M_{\rho_1,a_1,b_1,\rho_2,a_2,b_2}(f(x)+t)^\ell\rho_1\rho_2}{(\rho_1-1)(\rho_2-1)}\left(\frac{1}{\rho_1^{N_1}}+\frac{1}{\rho_2^{N_2}}\right)}_{=:Q_{k,\ell}}.$$

Since the constraint $\|\Lambda\hat{A}\|_2 \leqslant 1/(N+N_1N_2)$ implies $\|\Lambda A\|_1 < 1$, cf. Appendix E, it comes either $\lambda_{k,\ell} = 0$ or $Q_{k,\ell} < 1$ for any k,ℓ. Also, the proof of Lemma 6.11

Algorithm 3 Computation of the lattice to be reduced (2D approach)

Input: Four real numbers $a_1 < b_1, a_2 < b_2$, f a transcendental function analytic in a complex neighbourhood of $[a_1, b_1]$, five positive integers d, N_1, N_2, u, v, two real numbers $\rho_1, \rho_2 > 1$ such that $N_1, N_2 \geqslant 2$, $N_1 N_2 \geqslant N := (d+1)(d+2)/2$ and $16\rho_1\rho_2(\rho_1^{-N_1} + \rho_2^{-N_2})vM_{\rho_1,a_1,b_1,\rho_2,a_2,b_2}(f(x)+t) < (\rho_1 - 1)(\rho_2 - 1)$.

Output: Two matrices $M_c \in \mathcal{M}_{N,N_1N_2}(\mathbb{Z}), M_r \in \mathcal{M}_N(\mathbb{Z})$, where $N = (d+1)(d+2)/2$, respectively storing scaled values of the coefficients and the remainders, an integer tprec which is the truncation precision.

$R_0 \leftarrow \frac{16(\rho_1^{1-N_1}\rho_2 + \rho_1\rho_2^{1-N_2})}{(\rho_1-1)(\rho_2-1)}$, $\mathsf{tprec} \leftarrow \lceil -\log_2(R_0) + \log_2(N) \rceil + 2$

// Computation of the Chebyshev nodes, listed in reverse order

$L_{cheb,x} \leftarrow \left[\frac{b_1-a_1}{2} \cos\left((j+1/2)\frac{\pi}{N_1} \right) + \frac{a_1+b_1}{2} \right]_{0 \leqslant j \leqslant N_1 - 1}$

$L_{cheb,t} \leftarrow \left[\frac{b_2-a_2}{2} \cos\left((j+1/2)\frac{\pi}{N_2} \right) + \frac{a_2+b_2}{2} \right]_{0 \leqslant j \leqslant N_2 - 1}$

$M_c \leftarrow [0]_{N \times N_1N_2}$; $M_r \leftarrow [0]_{N \times N}$

$B_x \leftarrow \left| \frac{a_1+b_1}{2} \right| + \frac{b_1-a_1}{4}(\rho_1 + \rho_1^{-1})$, $B_t \leftarrow \rho_2 \max(|a_2|, |b_2|)$

$g \leftarrow \left(x \mapsto \left| f\left(\frac{a_1+b_1}{2} + \frac{b_1-a_1}{4}(\rho_1 \exp(ix) + \rho_1^{-1}\exp(-ix)) \right) \right| \right)$

$B_f \leftarrow \max\left(g([0, 2\pi]) \right)$, $i \leftarrow 0$

for $\ell = 0$ **to** d **do**

  for $k = 0$ **to** $d - \ell$ **do**

    $\varphi \leftarrow ((x,t) \mapsto (ux)^k (v(f(x)+t))^\ell)$

    // We compute the coefficient matrix : for each function, we compute its value at points of $L_{cheb,x} \times L_{cheb,t}$, use DCT and scale.

    $U \leftarrow \frac{4}{N_1N_2}\text{2D-DCT-II}\left((\varphi(L_{cheb,x}[\ell_1], L_{cheb,t}[\ell_2]))_{\substack{0 \leqslant \ell_1 \leqslant N_1-1 \\ 0 \leqslant \ell_2 \leqslant N_2-1}} \right)$,

    for $k_1 = 0$ **to** $N_1 - 1$ **do**

      for $k_2 = 0$ **to** $N_2 - 1$ **do**

        $M_c[i, k_2 + k_1N_2] \leftarrow U[k_1, k_2]$.

      end for

    end for

    for $k_1 = 0$ **to** $N_1 - 1$ **do**

      $M_c[i, k_1N_2] \leftarrow \frac{1}{2}M_c[i, k_1N_2]$

    end for

    for $k_2 = 0$ **to** $N_2 - 1$ **do**

      $M_c[i, k_2] \leftarrow \frac{1}{2}M_c[i, k_2]$

    end for

    for $j = 0$ **to** $N_1N_2 - 1$ **do**

      $M_c[i, j] \leftarrow [2^{\mathsf{tprec}} M_c[i, j]]_0$

    end for

    // We compute the scaled remainder matrix.

    $M_r[i, i] \leftarrow \left\lfloor 2^{\mathsf{tprec}} R_0 (uB_x)^k (v(B_f + B_t))^\ell \right\rfloor$, $i \leftarrow i + 1$

  end for

end for

Return M_c, M_r, tprec

Algorithm 4 2D approach to Problem 2.6

Input: Four real numbers $a_1 < b_1, a_2 < b_2$, f a transcendental function analytic in a complex neighbourhood of $[a_1, b_1]$, five positive integers d, N_1, N_2, u, v, two real numbers $\rho_1, \rho_2 > 1$ such that $N_1, N_2 \geqslant 2$, $N_1 N_2 \geqslant N := (d+1)(d+2)/2$ and $16\rho_1\rho_2(\rho_1^{-N_1} + \rho_2^{-N_2}) v M_{\rho_1,a_1,b_1,\rho_2,a_2,b_2}(f(x)+t) < (\rho_1 - 1)(\rho_2 - 1)$.

Output: If successful, return a list $\mathcal{L}$ such that $\mathcal{L} \supset \{X \in \mathbb{Z}$ such that $a_1 \leqslant X/u \leqslant b_1$ and there exists $Y \in \mathbb{Z}$, $\frac{Y}{v} \in \left[f\left(\frac{X}{u}\right) + a_2, f\left(\frac{X}{u}\right) + b_2\right]\}$.

1: $(M_c, M_r, \mathsf{tprec}) \leftarrow$ Algorithm 3 $(a_1, b_1, a_2, b_2, f, d, N_1, N_2, u, v, \rho_1, \rho_2)$,
2: $M_{LLL} \leftarrow$ LLL-reduce the rows of $(M_c \mid M_r)$
3: $U \leftarrow M_{LLL,r} M_r^{-1}$ // This is the LLL change of basis matrix; $M_{LLL,r}$ is the right part of the matrix M_{LLL}. Note that M_r is diagonal.
4: **if** $\max(\|(M_{LLL}[0,j])_{0 \leqslant j \leqslant N+N_1N_2-1}\|_2, \|(M_{LLL}[1,j])_{0 \leqslant j \leqslant N+N_1N_2-1}\|_2) \leqslant 2^{\mathsf{tprec}}/(N + N_1N_2)$ **then**
5: $L_m \leftarrow [X_1^k X_2^\ell$ for $k = 0$ to $d - \ell$ for $\ell = 0$ to $d]$ // List of monomials, ordered in a way compatible with Algorithm 3, Steps 8–10.
6: $P_0 \leftarrow \sum_{j=0}^{N-1} U[0,j] L_m[j]$, $P_1 \leftarrow \sum_{j=0}^{N-1} U[1,j] L_m[j]$
7: $R(X_1) \leftarrow \mathrm{Res}_{X_2}(P_0(X_1, X_2), P_1(X_1, X_2))$
8: **if** $R(X_1) \neq 0$ **then**
9: $\mathcal{L} \leftarrow \{t \in \mathbb{Z}; R(t) = 0\}$
10: return $\mathcal{L}$
11: **else**
12: return "FAIL"
13: **end if**
14: **else**
15: return "FAIL"
16: **end if**

shows in particular that $R_i \geqslant R_{d+2} = 16 \frac{v M_{\rho_1,a_1,b_1,\rho_2,a_2,b_2}(f(x)+t)\rho_1\rho_2}{(\rho_1-1)(\rho_2-1)} \left(\frac{1}{\rho_1^{N_1}} + \frac{1}{\rho_2^{N_2}}\right)$ for all $i \geqslant d+2$. Hence, if $R_{d+2} \geqslant 1$, we thus have $R_i \geqslant 1$ for all $i \geqslant d+2$ and $\lambda_{k,\ell} = 0$ for any $1 \leqslant \ell \leqslant d$, $0 \leqslant k \leqslant d - \ell$: the only functions taken into account are the $u^k x^k$'s and the method fails. This explains the condition $16\rho_1\rho_2(\rho_1^{-N_1} + \rho_2^{-N_2}) v M_{\rho_1,a_1,b_1,\rho_2,a_2,b_2}(f(x)+t) < (\rho_1 - 1)(\rho_2 - 1)$ in the input of Algorithm 3 and 4.

We deduce the following corollary.

Corollary 6.14. *Under the assumptions and notations of the previous theorem, for all x, y such that $ux, vy \in \mathbb{Z}$, we have either $P(ux, vy) = 0$, or $y \notin [f(x) + a_2, f(x) + b_2]$.*

6.4.2. *Proof of success of Algorithm 4.* If we apply the LLL lattice basis reduction algorithm to $\hat{A}$, we obtain:

Corollary 6.15. *Assume that* $\det(\hat{A}\hat{A}^t)^{1/2(N-1)} \leqslant \frac{2^{-(N-1)/4-\mathsf{tprec}/(N-1)}}{N+N_1N_2}$*; then Theorem 6.12 applies with Λ any of the first two vectors of an LLL-reduced basis of the lattice generated by the rows of $\hat{A}$.*

Proof. Identical to the proof of Corollary 5.12. □

We now study the case $\rho_1 = K_1/(b_1-a_1)$ and similarly $\rho_2 = K_2/(b_2-a_2)$, where $K_1 > 2(b_1-a_1)$ and $K_2 > 2(b_2-a_2)$ are fixed real numbers (note that $\rho_1, \rho_2 > 2$); we further assume $\rho_1^{N_1} \leqslant \rho_2^{N_2}$.

Proposition 6.16. *Let f be analytic in a neighbourhood of the closed disc $\mathcal{D}_{a_1,b_1,K_1} = \{z \in \mathbb{C} : |z-(a_1+b_1)/2| \leqslant K_1/2\}$, d be an integer $\geqslant 2$, $N = (d+1)(d+2)/2$, $\rho_1 = K_1/(b_1-a_1) > 2$, $\rho_2 = K_2/(b_2-a_2) > 2$, $\gamma = \log\rho_2/\log\rho_1 \in [3, N]$, $N_1 = \lfloor\sqrt{2\gamma N}\rfloor, N_2 = \lceil\sqrt{2N/\gamma}\rceil$ two integers. Let $M_{\mathcal{D}_{a_1,b_1,K_1}}(f) := \max_{z\in\mathcal{D}_{a_1,b_1,K_1}} |f(z)|$.*

Then, for $d \to \infty$, if

$$b_1 - a_1 < K_1 2^{O\left(-\frac{N}{\psi(N/\gamma)\gamma}\right)} \left(\frac{uv}{2}(|a_1+b_1|+K_1)\left(M_{\mathcal{D}_{a_1,b_1,K_1}}(f) + \frac{K_2+|a_2+b_2|}{2}\right)\right)^{-\frac{d}{3\psi(N/\gamma)\gamma}(1+O(1/d))},$$

we have $\Delta^{1/(N-1)}_{N,N_1,N_2,[a_1,b_1],[a_2,b_2],\rho_1,\rho_2} < \frac{2^{-(N-1)/4-\mathsf{tprec}/(N-1)}}{N+N_1N_2}$.

Proof. Since $\rho_1 = K_1/(b_1-a_1) > 2$, we have $E_{\rho_1,a_1,b_1} \subset \mathcal{D}_{a_1,b_1,K_1}$. Thanks to Corollary 6.9, in view of $(\rho_i/(\rho_i-1))^{N/(N-1)} \leqslant 2^{3/2}$ for $i \in \{1,2\}$, we have

$$\Delta^{1/(N-1)}_{N,N_1,N_2,[a_1,b_1],[a_2,b_2],\rho_1,\rho_2} \leqslant 2^{O(1)} N^{1/2+o(1)} \frac{(uv(|a_1+b_1|+K_1)/2)^{d/3+O(1)}}{\rho_1^{\psi(N/\gamma)\gamma+O(1)}} \left(M_{\mathcal{D}_{a_1,b_1,K_1}}(f) + \frac{K_2+|a_2+b_2|}{2}\right)^{d/3+O(1)}.$$

Note that, using Lemma 6.11, as $\rho_1^{N_1} \leqslant \rho_2^{N_2}$,

$$\begin{aligned} 2^{-\mathsf{tprec}} &\geqslant \frac{4}{N}\left(\rho_1^{-N_1} + \rho_2^{-N_2}\right) \\ &\geqslant \frac{8}{N}\rho_2^{-N_2} = \frac{8}{N}\rho_1^{-\gamma N_2} \\ &\geqslant 2^{-o(N)}\rho_1^{-O(N)}, \text{as } \gamma N_2 < \sqrt{2N\gamma} + \gamma \leqslant N(1+\sqrt{2}). \end{aligned}$$

Thus, for $\Delta^{1/(N-1)}_{N,N_1,N_2,[a_1,b_1],[a_2,b_2],\rho_1,\rho_2} < 2^{-(N-1)/4-\mathsf{tprec}/(N-1)}/(N+N_1N_2)$, it suffices that $\Delta^{1/(N-1)}_{N,N_1,N_2,[a_1,b_1],[a_2,b_2],\rho_1,\rho_2} < 2^{-O(N)}\rho_1^{-O(1)}$, or again that

$$\rho_1 > 2^{O\left(\frac{N}{\psi(N/\gamma)\gamma}\right)} \left(\frac{uv}{2}(|a_1+b_1|+K_1)\left(M_{\mathcal{D}_{a_1,b_1,K_1}}(f) + \frac{K_2+|a_2+b_2|}{2}\right)\right)^{\frac{d}{3\psi(N/\gamma)\gamma}(1+O(1/d))}.$$

□

Corollary 6.17. *Under the assumptions of Proposition 6.16, Algorithm 4 over $[a_1,b_1]$ and $[a_2,b_2]$ produces at Step 6 two polynomials P_0, P_1 such that*

$$\max_{x\in[a_1,b_1],\, t\in[a_2,b_2]} |P_i(ux, v(f(x)+t))| < 1 \text{ for } i \in \{0,1\}.$$

In particular, Algorithm 4 never executes Step 15 and its output is valid.

Proof. It suffices to apply Proposition 6.16, Corollary 6.15, and Theorem 6.12. □

Note again that P_0 and P_1 may not be coprime, in which case the algorithm returns "FAIL" at Step 12. This is what makes the algorithm heuristic.

6.5. **Complexity analysis.** In this subsection, we deduce estimates for the complexity of our algorithm applied to a fixed interval $[\alpha, \beta)$. As in the univariate case, this actually requires several things:

- An evaluation of the complexity of the basic blocks, namely Algorithms 3 and 4.
- Use Corollary 6.17 to evaluate the size of a subinterval $[a_1, b_1]$ which can be treated at once by those algorithms.

We start by giving complexity estimates for Algorithms 3 and 4.

6.5.1. *Complexity of Algorithms 3 and 4.* In this subsection, we keep notations and assumptions of Section 5.5.1.

Proposition 6.18. *On input* $a_1, b_1, a_2, b_2, f, d, N_1, N_2, u, v, \rho_1, \rho_2$, *if*

$$\mathcal{M} := \max(u, v, |a_1|, |b_1|, \rho_1, |a_2|, |b_2|, \rho_2, B_f, \max_{[a_1,b_1]} |f'(x)|),$$

under the assumption $C_{f,\mathfrak{p}} = O(\mathfrak{p}^2)$, *the computations of Algorithm 3 can be made in floating-point precision* $\mathfrak{p} = \mathsf{tprec} + O(\max(d \log \mathcal{M}, |\log((\rho_1 - 1)(\rho_2 - 1))|)$. *Hence, Steps 1–6 of Algorithm 4 have complexity* $O(d^6 \mathsf{M}(d^2)(d^2 + \mathfrak{p})\mathfrak{p})$ *using the* L^2 *algorithm.*

Proof. Similar to Propositions 5.16 and 5.17. □

6.5.2. *Number of subintervals for fixed d.* Thanks to the results of the previous subsection, given a value γ, we can estimate the maximum size of an interval $[a_1, b_1] \subset [\alpha, \beta]$, with α, β fixed, for which Algorithm 4 succeeds (in the sense of Corollary 6.17) and yields an upper bound of the order of magnitude $w = O(|b_1 - a_1|^{-\gamma})$.

This follows from Proposition 6.16, and yields at the same time the number of subintervals to be considered if one wants to deal with a full interval $[\alpha, \beta]$.

Theorem 6.19. *Given fixed* f *and two fixed real numbers* α, β, *Problem 2.6 can heuristically be solved for* $u, v \to \infty$, $d \to \infty$, $\gamma \in [3, N]$, *over* $[\alpha, \beta]$ *using*

$$(\beta - \alpha) 2^{O\left(\frac{N}{\gamma \psi(N/\gamma)}\right)} (uv)^{\frac{d}{3\psi(N/\gamma)\gamma}(1+O(1/d))} \tag{6.11}$$

calls to Algorithm 4 with parameter d.

We then obtain a value

$$w = 2^{O(N/\psi(N/\gamma))} (uv)^{\frac{d}{3\psi(N/\gamma)}(1+O(1/d))}. \tag{6.12}$$

Proof. This is a direct consequence of Corollary 6.17, where we note that a_1, a_2, b_1, b_2 are bounded, and we choose $K_1 = 2(b_1 - a_1)$, $K_2 = 1$ and $\rho_2 = \rho_1^\gamma$.

The heuristic nature of this result comes from the possibility that the two polynomials obtained in Algorithm 4 are not coprime, in which case one cannot recover the solutions X, Y from those two polynomials.

Finally, we can take $1/w = b_2 - a_2$, thus the upper bound on w is $O((b_1 - a_1)^{-\gamma})$, from which the second part of the result follows. □

Remark 6.20. To get a better feeling of this result we should distinguish two cases:

- We let γ tend to infinity as N/κ, $\kappa \geqslant 1$. Then, we obtain an upper bound for the number of intervals $O\left((\beta - \alpha)(uv)^{\frac{2\kappa}{3d\psi(\kappa)}(1+O(1/d))}\right)$, with $w = (uv)^{\frac{d}{3\psi(\kappa)}(1+O(1/d))}$.

- If, on the other hand, $\gamma = o(N)$, N/γ tends to infinity and we can use the asymptotic estimate (see Proposition 6.5) $\psi(N/\gamma) = \frac{2}{3}\sqrt{\frac{2N}{\gamma}} + O(1)$ to get a bound on the number n_I of intervals and on w of the respective forms

$$n_I = (\beta-\alpha)2^{O(d/\sqrt{\gamma})}(uv)^{1/(2\sqrt{\gamma})(1+O(\sqrt{\gamma}/d))},$$
$$w = 2^{O(d\sqrt{\gamma})}(uv)^{\sqrt{\gamma}/2(1+O(\sqrt{\gamma}/d))}.$$

The first case resembles the results obtained in the previous section with the univariate algorithm (but with a different constant), whereas the second part is unattainable using the methods of the previous section.

For $u = v = 2^p$, $\gamma = 4 + o(1)$, $d = o(p)$, we recover Stehlé's result [61], namely the fact that we can solve the TMD (i.e., get the bound $1/w = 2^{-2p}$ in time $2^{p/2(1+o(1))}$.

For examples of practical values of d, γ, the reader might consult Table 4. This table shows that the relevant regime for the TMD problem seems to be γ/N bounded rather that $\gamma = o(N)$, the relevance of which seems more theoretical.

Remark 6.21. One can adapt Remark 5.22, Theorems 5.26 and 5.27 in the case of this bivariate method. However, for the last two results, the region where the discussion makes sense is restricted to $\gamma \gg N/(\log N)^2$, as otherwise the impact of choosing an optimal ρ_1 occurs only on second order terms.

Further, this leads to a somewhat delicate discussion and, in the end, yields the same result up to some improvement in the constants. We thus chose not to include the corresponding theorems.

7. Comparison with previous work

The following table summarizes the main results of the paper and compares them to [61]. The columns complexity and bound on w should be understood as the exponent of uv in the corresponding values. For the sake of readability,

- In the first row, we restrict to $\omega_0 \in \mathbb{Z}_{>0}$ just in order to get a more compact bound ;
- In the fourth row, we shall assume that $d = o(\log(uv))$;
- We shall omit all factors $(1 + o(1))$ in the asymptotic results.

Rows labelled "1D" refer to Section 5 (Algorithms 1 and 2) whereas rows labelled "2D" refer to Section 6 (Algorithms 3 and 4).

	References	Parameters to be chosen	Matrix dimensions	Complexity (exponent of uv)	Bound on w (exponent of uv)
1D	Rk. 5.20	$\omega_0 \in [0, N) \cap \mathbb{Z}_{>0}$	$N \times 2N$	$\frac{2Nd}{3(N-\omega_0)(N+\omega_0-3)}$	$\frac{2Nd}{3(N+\omega_0-3)}$
$1D_{d\to\infty}$	Cor. 5.21	$\lambda = \omega_0/N \in [0, 1)$	$N \times 2N$	$\frac{4}{3(1-\lambda^2)d}$	$\frac{2d}{3(1+\lambda)}$
$2D_{d\to\infty}$	Rk. 6.20	$\gamma = N/\kappa$, $\kappa \geqslant 1$	$\approx N \times 3N$	$\frac{2\kappa}{3d\psi(\kappa)}$	$\frac{d}{3\psi(\kappa)}$
$2D_{d\to\infty}$	Rk. 6.20	$\gamma = o(N)$	$\approx N \times 3N$	$\frac{1}{2\sqrt{\gamma}}$	$\sqrt{\gamma}/2$
$S_{\alpha\to\infty}$	[61, Thm. 3]	$\xi \geqslant 1$	$\frac{(\alpha+1)(\alpha+2)}{2}$ $\times O(\xi\alpha^2)$	$\frac{1}{2\sqrt{\xi}}$	$\sqrt{\xi}/2$

Table 2. Comparison of the main methods of this paper and [61]

Concerning [61], we have introduced a parameter ξ for a clearer comparison; namely, we have put (with Stehlé's notations) $n_1 - t = \log_2(uv)/(2\sqrt{\xi})$, which gives $n_2 + m = \sqrt{\xi}\log(uv)/2$. Note that Stehlé's α corresponds to our d, while his d is a technical parameter with a completely different meaning of our d; finally, Stehlé's t corresponds to our notations $p - 1 - \log_2(b_1 - a_1)$. To avoid any confusion, we will denote them $d_{\mathrm{Ste}}, \alpha_{\mathrm{Ste}}$ and t_{Ste} in the sequel.

Remark 7.1. Table 2 only estimates the exponential part of the complexity; it would remain to estimate the polynomial part, which is dominated by the cost of lattice basis reduction. Akhavi-Stehlé's trick reduces greatly the influence of the dimension, but the size of the integers involved in the different methods differ. Roughly speaking, and ignoring the dependency on f, a, b which is similar for the two methods:

- in the case of the univariate method, the size of the integers involved is $\approx \mathsf{tprec} + d\log(\max(u,v)) \approx d\log\max(u,v) + N\log\rho$, which is of the order of $O(d\log\max(u,v) + \log w)$. As for this method, $\log(w) = O(d\log(uv))$, the size of the integers is $O(d\log\max(u,v))$;
- in the case of the bivariate method, the size of the integers involved is $\approx d\log(\max(u,v)) + \mathsf{tprec} \approx d\log\max(u,v) + \log\max(\rho_1^{N_1}, \rho_2^{N_2})$, whereas the bound on w is of the order of ρ_2; as we expect, for optimal choices of parameters, that $N_1\log\rho_1$ and $N_2\log\rho_2$ have the same order of magnitude, this size is thus $O(d\log\max(u,v) + N_2\log\rho_2) = O(d\log\max(u,v) + N_2\log w)$. As $N_2 \asymp d/\sqrt{\gamma}$, this is $O(d\left(\log\max(u,v) + \log w/\sqrt{\gamma}\right))$;
- in Stehlé's paper, the integers involved are of the order of $(MN_2N_1^{d_{\mathrm{Ste}}})^{\alpha_{\mathrm{Ste}}}$, which, in our notations, means that their size is of the order of $O(d(\log w + \log\max(u,v)))$.

This difference in the size of the integers involved may, at least partially, explain the fact that lattice basis reduction performs somewhat better in our method than in the BaCSeL implementation of Stehlé's method (see Section 8).

7.1. **Univariate method and Bombieri and Pila's approach.** Our univariate method bears a strong resemblance to Bombieri & Pila's approach [6] of bounding the number of integer points on an analytic curve. The method that we develop is effective, and yields a way to not only control integer points on the curve, but close to the curve.

Note that we (asymptotically) recover Bombieri and Pila's estimate [6, Main Lemma] under the form $(uv)^{\frac{4}{3(d+3)}}$, which is the number of intervals required. We can thus recover, following Bombieri and Pila's arguments based on [6, Thm. 5] or [56], their bound for the number of points on the curve (without any heuristic). Our method also allows for the explicit determination of those points, but is on this point only (mildly) heuristic.

Our variation with the ω_0, on the other hand, is new; it worsens the quality of Bombieri-Pila's bound, but improves the distance around the curve in which we are able to detect points with denominator dividing v.

7.2. **Bivariate method vs. Stehlé's approach.** On the other hand, our bivariate method bears a strong resemblance with Stehlé's work [61]. Our approach mostly differs by the use of approximation-related techniques (Chebyshev interpolants) rather than a computer-algebra oriented vision of functions using Taylor expansions.

We have a dense representation of our auxiliary polynomials, which leads us to manipulate almost square matrices, whereas Stehlé's matrices are inherently more rectangular, because he has to represent all coefficients of the bivariate polynomials he manipulates.

Table 5 shows that our approach is better in practice. We propose an analysis of this favourable situation in Section 8.3.

Note two further facts :

- our analysis is sharper in the case of entire functions, as we are able to take into account the growth of the function at infinity. The same results could probably be derived in Stehlé's paper using Cauchy inequalities to estimate Taylor coefficients.

  This sharper analysis allows us to a obtain, in Problem 2.6, a lower bound $1/w \geqslant (uv)^{-O(p^2/\log p)}$ in the case where we want to solve the problem without any subdivision. This improves at the same time on Stehlé's method and, heuristically, Nesterenko-Waldschmidt paper which, though with different goals and through a purely theoretical (vs. algorithmic) method, both obtain the upper bound $w \leqslant (uv)^{-O(p^2)}$.
- Our analysis is also sharper in the more practical domain where we choose $\gamma = N/\kappa$ for some constant κ. We obtain better constants in the exponents for both the overall complexity of the method and the bound on w.

8. Experimental results

We have implemented TMD-oriented versions of our algorithms in SageMath[17]. Our codes are available from https://perso.ens-lyon.fr/nicolas.brisebarre/tmd.html. The tests hereafter were executed on an Intel Xeon E5620 2.40GHz CPU with a 64-bit Linux-based system.

In the two examples that we address, we cut the binades under consideration into subintervals of the same size and we apply the algorithms to each subinterval. For Algorithms 1 and 2, the subinterval will correspond to the interval $[a, b]$ considered in these algorithms. For Algorithms 3 and 4, the subinterval will correspond to the interval $[a_1, b_1]$, while $[a_2, b_2]$ will be equal to $[-1/w, 1/w]$, cf. Problem 2.6.

Currently, the most expensive part of our algorithms is the LLL reduction. In our implementations, the following two optimizations led to a significant speedup of the LLL reduction part:

(1) we use a random projection trick inspired from [2], cf. Theorem 4.6.
(2) If we consider two contiguous subintervals I_0 and I_1, the matrices M_{c,I_0} and M_{r,I_0}, M_{c,I_1} and M_{r,I_1}, output by Algorithm 1 (resp. Algorithm 3) applied to I_0 and I_1 will be close by construction. Hence, our optimization consists in:
   - retrieving the change-of-basis matrix U_{I_0} computed at Step 3 of Algorithm 2 (resp. Step 3 of Algorithm 4) applied to I_0,
   - then left-multiplying M_{c,I_1} and M_{r,I_1} by U_{I_0}, which operates in practice as a significant prereduction of the lattice built from M_{c,I_1} and M_{r,I_1},
   - eventually, we apply LLL to these prereduced matrices.

[17] https://www.sagemath.org/

Another optimization comes from the use of Newton polynomials instead of monomials (as pointed in Section 5.3.4): for given values of d, N, N_1, N_2, it makes it possible to process larger subintervals.

The timings and the values of $\log_2(w)$ presented with a * are estimated ones: we performed our computations on a subinterval and then extrapolate the timing to address the whole binade, and the corresponding value of $\log_2(w)$.

We chose to limit the evaluation of our algorithms on feasible computations in `binary128`, namely computations that could be performed in real life, possibly using a large cluster. In terms of the bound on w, we have thus excluded the optimal case of the TMD, namely $w \approx 2^{2p}$, and have started at $w \approx 2^{6p}$.

8.1. **Algorithms 1 and 2 in action: the TMD for the gamma function in binary128.** Euler's gamma [65, Chap. 3] is one of the functions of the C mathematical library. Very little is known about the Diophantine properties of its values at rational numbers: we have $\Gamma(k+1) = k!$ for any $k \in \mathbb{N}$ and the transcendence of the numbers $\Gamma(1/2), \Gamma(1/3), \Gamma(1/4), \Gamma(1/6), \Gamma(2/3), \Gamma(3/4), \Gamma(5/6)$ [70]. We used our implementation of Algorithms 1 and 2 to address the TMD over $[1, 2)$, for directed rounding functions and for the precision $p = 113$. More precisely, we address the following question, for various values of the parameters d and ω_0: compute $w > 0$ and all the integers X, $1 \leqslant X/2^{p-1} < 2$ for which there exists $Y \in \mathbb{Z}$ satisfying

$$\left| \Gamma\left(\frac{X}{2^{p-1}}\right) - \frac{Y}{2^p} \right| < \frac{1}{w}.$$

We report in Table 3 our results. We first set $p = 113, u = 2^{p-1}, v = 2^p$, then the integer d and the real number ω_0, and finally we choose ρ in order to maximize the width of the subinterval $[a, b]$ of $[1, 2)$ which we apply the algorithms to. The column $\|M\|_\infty$ stands for the size (in bits) of the largest coefficient of the integer matrix $M = (M_c \mid M_r)$ which is LLL-reduced during the algorithm.

d	N	ρ	$b-a$	ω_0	$\|M\|_\infty$	$\log_2(w)$	Timing
6	28	2^{32}	2^{-31}	0	≈ 1575 bits	$7.86p^*$	35.53^* years
8	45	2^{26}	$21/2^{29}$	0	≈ 2070 bits	$10.11p^*$	510^* days
10	66	2^{21}	$13/2^{24}$	0	≈ 2510 bits	$12.24p$	102 days
12	91	2^{19}	$25/2^{23}$	14	≈ 2810 bits	$12.92p$	84 days
12	91	2^{19}	$21/2^{22}$	8	≈ 2920 bits	$13.79p$	57 days
12	91	2^{18}	$15/2^{21}$	0	≈ 2985 bits	$14.44p$	46 days

TABLE 3. Algo. 1 and 2: The gamma function over the binade $[1, 2)$

8.2. **Using ω_0 in Algorithms 1 and 2 : the TMD for the exponential function in binary128 - experimental validation of Figure 1.** Using the exp function over $[1/4, 1/2)$, we have also studied the influence of ω_0, in order to build the experimental equivalent of the theoretical curves of Figure 1.

The exponential function is also part of the C mathematical library. We recall that Section 3.1 presents up-to-date, to the best of our knowledge, theoretical results regarding the TMD in the case of the exponential function.

Here, we address the TMD for exp over $[1/4, 1/2)$, for directed rounding functions and for the precision $p = 113$: we compute $w > 0$ and all the integers X, $1/4 \leqslant X/2^{p+1} < 1/2$ for which there exists $Y \in \mathbb{Z}$ satisfying

$$\left|\exp\left(\frac{X}{2^{p+1}}\right) - \frac{Y}{2^{p-1}}\right| < \frac{1}{w}.$$

Here, we set $u = 2^{p+1}$ and $v = 2^{p-1}$.

For each value of the bound on w we have tried to find, for various values of d, the ω_0 allowing to attain this bound in minimal time (in the case of exp, we use $\rho = d/(b-a)$, cf. Theorem 5.26). Figure 2 represents the $\log_2$ of the time to treat a binade (y-axis) as a function of $\log_2(w)/p$ (x-axis); this is indeed the experimental equivalent of Figure 1, up to a rescaling of the x-axis (by a factor of 2, as Figure 1 is expressed in terms of powers of $uv = 2^{2p}$) and y-axis.

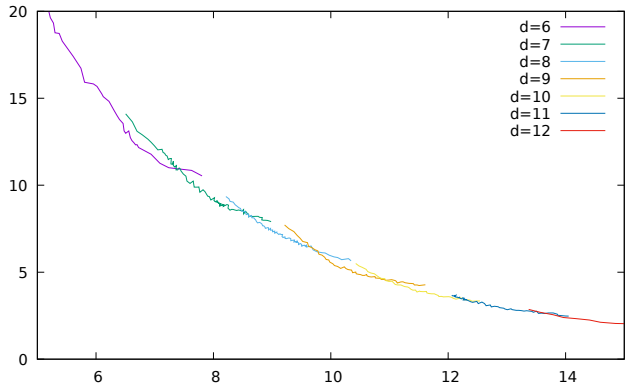

FIGURE 2. Exponent estimates

8.3. **Algorithms 3 and 4 in action: the TMD for the exponential function in binary128.** We used our implementation of Algorithms 3 and 4 to address the TMD over $[1/4, 1/2)$, for directed rounding functions and for the precision $p = 113$. More precisely, we address the following question, for various values of the parameter w: determine the integers X, $1/4 \leqslant X/2^{p+1} < 1/2$ for which there exists $Y \in \mathbb{Z}$ satisfying

$$\left|\exp\left(\frac{X}{2^{p+1}}\right) - \frac{Y}{2^{p-1}}\right| < \frac{1}{w}.$$

We report in Table 4 our results. We first set $p = 113, u = 2^{p+1}, v = 2^{p-1}, b_2 = -a_2 = 1/w$ and the value of d. The choice of the parameters N_1, N_2, ρ_1, a_1 and[18] b_1 is then made in order to maximize the width of the subinterval $[a_1, b_1]$ of $[1/4, 1/2)$. We finally fix $\rho_2 = \min(\rho_1^{N_1/N_2}, 1/b_2)$.

The column $\|M\|_\infty$ stands for the largest coefficient of the integer matrix $M = (M_c \mid M_r)$ which is to be LLL-reduced.

For the sake of practical comparison, we have attempted a comparison with BaCSeL-4.0[19], which implements [61]. For the latter, we used BaCSel-4.0 only for

[18] Actually, it is the value of $b_1 - a_1$ which matters and not the values of a_1 and b_1.

[19] https://gitlab.inria.fr/zimmerma/bacsel

$\log_2(w)$	d	N	N_1	N_2	ρ_1	$b_1 - a_1$	γ	$\|M\|_\infty$	Timing
$6p$	6	28	24	2	2^{36}	$7/2^{35}$	18.8	≈ 1540 bits	22.63* years
$6p$	12	91	60	3	2^{29}	$59/2^{30}$	23.4	≈ 3080 bits	3.41* years
$8p$	8	45	39	2	2^{30}	$15/2^{29}$	30.1	≈ 2065 bits	203* days
$8p$	12	91	59	3	2^{26}	2^{-21}	34.8	≈ 2870 bits	137 days
$10p$	10	66	59	2	2^{25}	$7/2^{23}$	45.2	≈ 2590 bits	25.8 days
$10p$	12	91	61	3	2^{24}	$7/2^{22}$	47.1	≈ 3260 bits	37.7 days
$12p$	12	91	80	2	2^{21}	$5/2^{19}$	64.6	≈ 3020 bits	8.7 days

TABLE 4. Algo. 3 and 4: exp over the binade $[1/4, 1/2)$

the generation of the corresponding matrix, and simply measured the cost of the LLL step (which dominates the total cost anyway), using the same implementation of fplll as in our code.

For the comparison to be fair, we have included the Akhavi-Stehlé's trick (cf. Theorem 4.6) in BaCSeL. We have also tried to include the prereduction trick but the latter, in the setting of [61], seems to make the reduction more costly. In this case, the complete timings are merely estimates for the cost of treating a whole binade.

The results are reported in Table 5. Again, the column $\|M\|_\infty$ stands for the largest coefficient of the integer matrix which is to be LLL-reduced.

$\log_2(w)$	α_{Ste}	d_{Ste}	t_{Ste}	$\|M\|\|_\infty$	Timing	Comparison with this paper	
						$\|M\|\|_\infty$	Timing
$6p$	6	16	78.7	≈ 3870 bits	169* years	$\times 2.5$	$\times 7.5$
$6p$	12	20	86.2	≈ 7780 bits	334* years	$\times 2.5$	$\times 98$
$8p$	8	30	86.3	≈ 7220 bits	16.82* years	$\times 3.5$	$\times 30$
$8p$	12	30	90.1	≈ 10230 bits	35.01* years	$\times 3.5$	$\times 93$
$10p$	10	40	90.8	≈ 11150 bits	7.37* years	$\times 4.3$	$\times 104$
$10p$	12	55	93.2	≈ 13545 bits	10.53* years	$\times 4.2$	$\times 102$
$12p$	12	60	94.6	≈ 16255 bits	3.81* years	$\times 5.4$	$\times 160$

TABLE 5. Stehlé's BaCSeL parameters and timings for the exponential function over the binade $[1/4, 1/2)$

We observe that we gain a significant constant factor, increasing with the value of α_{Ste} $(= d)$; our method allows for slightly larger intervals (by a factor around 2, which seems to decrease with d), but the main factors explaining the difference are the fact that our lattices seem somewhat easier to reduce and that the "prereduction trick" also plays an important role. These three terms each account for a small 2 to 5 (depending on the cases) factor, explaining overall the factors $\approx$ 9-100 that we observe above.

It should finally be recalled that the comparison is biased in favour of [61] : we are comparing optimized C code to an algorithm implemented in an interpreted language. The comparison remains rather fair as long as the lattice basis reduction dominates (which is the case for $\alpha_{\text{Ste}} = 10, 12$) but a low-level implementation of

our algorithms is required in order to get reliable results when our parameter $d \leqslant 8$, where the gap is probably larger than suggested by Table 5.

Remark 8.1. Note that, regarding the targets $\log_2(w) = 6p, 8p$, a more relevant comparison between Table 4 and Table 5 would probably be obtained by comparing

- Row 2 of Table 4 with Row 1 of Table 5, which corresponds to the best choice of parameters for the problem with $\log_2(w) = 6p$; with this criterion, the ratio is ≈ 50.
- Row 4 of Table 4 with Row 3 of Table 5, which corresponds to the best choice of parameters for the problem with $\log_2(w) = 8p$; with this criterion, the ratio is ≈ 45.

9. Conclusion

We expect this work to be used in practice to address the TMD for the binary128 format, but we also hope that it will be of practical use for the determination of integer points close to a transcendental curve. Moreover, we believe that the tools that we have developed are of a more general interest in the context of practical applications of Coppersmith's method, and allow easier analysis of some "rectangular" variants.

Regarding future work, we plan, first, to improve our current implementations and then to study the case of algebraic functions, for which we have some preliminary results.

Acknowledgements.

We wish to thank Jean-Michel Muller for so many invaluable discussions, Martin Albrecht for his kind and effective help about fplll and Paul Zimmermann for his thorough and extremely helpful rereading of this paper.

References

[1] M. Ajtai. The Shortest Vector Problem in L_2 is NP-hard for Randomized Reductions (Extended Abstract). In *Proceedings of the 30th ACM symposium on Theory of computing (STOC)*, pages 10–19, 1998.

[2] A. Akhavi and D. Stehlé. Speeding-up Lattice Reduction with Random Projections. In E. S. Laber, C. F. Bornstein, L. T. Nogueira, and L. Faria, editors, *LATIN 2008: Theoretical Informatics, 8th Latin American Symposium, Búzios, Brazil, April 7-11, 2008, Proceedings*, volume 4957 of *Lecture Notes in Computer Science*, pages 293–305. Springer, 2008.

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

[4] 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.

[5] V. Berthé and L. Imbert. Diophantine approximation, Ostrowski numeration and the double-base number system. *Discrete Math. Theor. Comput. Sci.*, 11(1):153–172, 2009.

[6] E. Bombieri and J. Pila. The number of integral points on arcs and ovals. *Duke Math. J.*, 59(2):337–357, 1989.

[7] D. Boneh and G. Durfee. Cryptanalysis of RSA with private key d less than $N^{0.292}$. *IEEE Trans. Inform. Theory*, 46(4):1339–1349, 2000.

[8] J. P. Boyd. *Chebyshev and Fourier spectral methods*. Dover Publications Inc., Mineola, NY, second edition, 2001. Available from http://www-personal.umich.edu/~jpboyd/BOOK_Spectral2000.html.

[9] N. Brisebarre and S. Chevillard. Efficient polynomial L^∞ approximations. In *ARITH '07: Proceedings of the 18th IEEE Symposium on Computer Arithmetic*, pages 169–176, Washington, DC, 2007. IEEE Computer Society.

[10] N. Brisebarre, G. Hanrot, and O. Robert. Exponential sums and correctly-rounded functions. *IEEE Trans. Comput.*, 66(12):2044–2057, 2017.

[11] N. Brisebarre and J.-M. Muller. Correct rounding of algebraic functions. *Theor. Inform. Appl.*, 41(1):71–83, 2007.

[12] M. Bronstein. *Symbolic integration. I*, volume 1 of *Algorithms and Computation in Mathematics*. Springer-Verlag, Berlin, second edition, 2005. Transcendental functions, With a foreword by B. F. Caviness.

[13] J. W. S. Cassels. *An introduction to the geometry of numbers*. Classics in Mathematics. Springer-Verlag, Berlin, 1997. Corrected reprint of the 1971 edition.

[14] E. W. Cheney. *Introduction to approximation theory*. AMS Chelsea Publishing, Providence, RI, 1998. Reprint of the second (1982) edition.

[15] S. Chevillard. *Évaluation efficace de fonctions numériques – Outils et exemples*. PhD thesis, École normale supérieure de Lyon – Université de Lyon, 2009. In French.

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

[17] 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.

[18] H. Cohen. *A course in computational algebraic number theory*, volume 138 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin, 1993.

[19] D. Coppersmith. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. *J. Cryptology*, 10(4):233–260, 1997.

[20] D. Coppersmith. Finding small solutions to small degree polynomials. In J. H. Silverman, editor, *Proceedings of Cryptography and Lattices (CaLC)*, volume 2146 of *Lecture Notes in Computer Science*, pages 20–31. Springer-Verlag, Berlin, 2001.

[21] 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.

[22] F. de Dinechin, C. Lauter, and J.-M. Muller. Fast and correctly rounded logarithms in double-precision. *Theor. Inform. Appl.*, 41(1):85–102, 2007.

[23] F. De Dinechin, J.-M. Muller, B. Pasca, and A. Plesco. An FPGA architecture for solving the Table Maker's Dilemma. In *Application-Specific Systems, Architectures and Processors (ASAP), 2011 IEEE International Conference on*, pages 187–194, Santa Monica, United States, Sept. 2011. IEEE Computer Society.

[24] L. Fox and I. B. Parker. *Chebyshev polynomials in numerical analysis*. Oxford University Press, London-New York-Toronto, Ont., 1968.

[25] F. Gramain. Sur le lemme de Siegel (d'après E. Bombieri et J. Vaaler). *Publications mathématiques de l'Univ. P. et M. Curie*, 64:fascicule 1, 1983-1984.

[26] P. M. Gruber and C. G. Lekkerkerker. *Geometry of numbers*, volume 37 of *North-Holland Mathematical Library*. North-Holland Publishing Co., Amsterdam, second edition, 1987.

[27] G. Hanrot, V. Lefèvre, D. Stehlé, and P. Zimmermann. Worst Cases of a Periodic Function for Large Arguments. In P. Kornerup and J.-M. Muller, editors, *18th IEEE Symposium in Computer Arithmetic*, pages 133–140, Montpellier, France, June 2007. IEEE.

[28] D. Harvey and J. van der Hoeven. Integer multiplication in time $O(n \log n)$. *Ann. of Math. (2)*, 193(2):563–617, 2021.

[29] IEEE. *IEEE Standard for Floating-Point Arithmetic (IEEE Std 754-2019)*. July 2019. Available at https://ieeexplore.ieee.org/servlet/opac?punumber=8766227.

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

[31] 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.

[32] C.-P. Jeannerod, N. Louvet, J.-M. Muller, and A. Panhaleux. Midpoints and exact points of some algebraic functions in floating-point arithmetic. *IEEE Trans. Comput.*, 60(2):228–241, 2011.
[33] F. Johansson. Arb: efficient arbitrary-precision midpoint-radius interval arithmetic. *IEEE Trans. Comput.*, 66(8):1281–1292, 2017.
[34] W. Kahan. Why do we need a floating-point standard? Technical report, Computer Science, UC Berkeley, 1981. Available at https://www.cs.berkeley.edu/ wkahan/ieee754status/why-ieee.pdf.
[35] S. Khémira and P. Voutier. Approximation diophantienne et approximants de Hermite-Padé de type I de fonctions exponentielles. *Ann. Sci. Math. Québec*, 35(1):85–116, 2011.
[36] S. Khémira. *Approximants de Hermite-Padé, déterminants d'interpolation et approximation diophantienne*. PhD thesis, Université Paris 6, Paris, France, 2005.
[37] O. Knill. Cauchy-Binet for pseudo-determinants. *Linear Algebra Appl.*, 459:522–547, 2014.
[38] 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.
[39] C. Q. Lauter. *Arrondi Correct de Fonctions Mathématiques*. PhD thesis, École Normale Supérieure de Lyon, Lyon, France, Oct. 2008.
[40] V. Lefèvre. *Moyens Arithmétiques Pour un Calcul Fiable*. PhD thesis, École Normale Supérieure de Lyon, Lyon, France, 2000.
[41] V. Lefèvre. New results on the distance between a segment and $\mathbb{Z}^2$. Application to the exact rounding. In *Proceedings of the 17th IEEE Symposium on Computer Arithmetic (ARITH-17)*, pages 68–75. IEEE Computer Society Press, Los Alamitos, CA, June 2005.
[42] 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.
[43] 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.
[44] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261(4):515–534, 1982.
[45] J. Liouville. Nouvelle démonstration d'un théorème sur les irrationnelles algébriques. *C. R. Acad. Sci. Paris*, 18:910–911, 1844.
[46] J. Liouville. Remarques relatives à des classes très-étendues de quantités dont la valeur n'est ni algébrique ni même réductible à des irrationnelles algébriques. *C. R. Acad. Sci. Paris*, 18:883–885, 1844.
[47] J. Liouville. Sur des classes très étendues de quantités dont la valeur n'est ni algébrique ni même réductible à des irrationnelles algébriques. *J. Math. Pures Appl.*, 16:133–142, 1851.
[48] L. Lovász. *An algorithmic theory of numbers, graphs and convexity*, volume 50 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1986.
[49] P. W. Markstein. The New IEEE-754 Standard for Floating-Point Arithmetic. In *Numerical Validation in Current Hardware Architectures*, number 08021 in Dagstuhl Seminar Proceedings. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2008.
[50] J. C. Mason and D. C. Handscomb. *Chebyshev polynomials*. CRC Press, 2002.
[51] J.-M. Muller. *Elementary Functions, Algorithms and Implementation*. Birkhäuser, Boston, 3rd edition, 2016.
[52] 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.
[53] 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`.
[54] P. Q. Nguyen and D. Stehlé. An LLL algorithm with quadratic complexity. *SIAM J. Comput.*, 39(3):874–903, 2009.
[55] P. Q. Nguyen and B. Vallée, editors. *The LLL Algorithm - Survey and Applications*. Information Security and Cryptography. Springer, 2010.

[56] J. Pila. Density of integer points on plane algebraic curves. *Internat. Math. Res. Notices*, (18):903–912, 1996.
[57] G. Plonka, D. Potts, G. Steidl, and M. Tasche. *Numerical Fourier analysis*. Applied and Numerical Harmonic Analysis. Birkhäuser/Springer, Cham, 2018.
[58] M. J. D. Powell. *Approximation theory and methods*. Cambridge University Press, 1981.
[59] T. J. Rivlin. *The Chebyshev polynomials*. Wiley-Interscience [John Wiley & Sons], New York-London-Sydney, 1974. Pure and Applied Mathematics.
[60] M. J. Schulte and E. E. Swartzlander. Exact rounding of certain elementary functions. In E. E. Swartzlander, M. J. Irwin, and G. Jullien, editors, *Proceedings of the 11th IEEE Symposium on Computer Arithmetic*, pages 138–145. IEEE Computer Society Press, Los Alamitos, CA, June 1993.
[61] D. Stehlé. On the randomness of bits generated by sufficiently smooth functions. In F. Hess, S. Pauli, and M. E. Pohst, editors, *Proceedings of the 7th Algorithmic Number Theory Symposium, ANTS VII*, volume 4076 of *Lecture Notes in Computer Science*, pages 257–274. Springer-Verlag, Berlin, 2006.
[62] 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.
[63] D. Stehlé, V. Lefèvre, and P. Zimmermann. Searching Worst Cases of a One-Variable Function Using Lattice Reduction. *IEEE Trans. Comput.*, 54(3):340–346, Mar. 2005.
[64] G. Strang. The discrete cosine transform. *SIAM Rev.*, 41(1):135–147, 1999.
[65] N. M. Temme. *Special functions*. A Wiley-Interscience Publication. John Wiley & Sons, Inc., New York, 1996. An introduction to the classical functions of mathematical physics.
[66] S. Torres. *Tools for the Design of Reliable and Efficient Functions Evaluation Libraries*. PhD thesis, École normale supérieure de Lyon – Université de Lyon, Lyon, France, 2016.
[67] L. N. Trefethen. *Approximation Theory and Approximation Practice*. SIAM, 2013.
[68] W. Tucker. *Validated numerics, A short introduction to rigorous computations*. Princeton University Press, Princeton, NJ, 2011. A short introduction to rigorous computations.
[69] J. von zur Gathen and J. Gerhard. *Modern computer algebra*. Cambridge University Press, third edition, 2013.
[70] M. Waldschmidt. Transcendence of periods: the state of the art. *Pure Appl. Math. Q.*, 2(2, Special Issue: In honor of John H. Coates. Part 2):435–463, 2006.
[71] K. Xu. The Chebyshev points of the first kind. *Appl. Numer. Math.*, 102:17–30, 2016.
[72] A. Ziv. Fast evaluation of elementary mathematical functions with correctly rounded last bit. *ACM Trans. Math. Software*, 17(3):410–423, Sept. 1991.
[73] A. Zygmund. *Trigonometric series. Vol. I, II*. Cambridge Mathematical Library. Cambridge University Press, Cambridge, third edition, 2002.

Appendix A. Proofs of facts regarding Chebyshev polynomials

First, we recall the aliasing phenomenon in the case of Chebyshev nodes of the first kind.

Proposition A.1. *For all $N \geqslant 1$,*

- *for $k = 0, \dots, N-1$, the polynomials $T_k, -T_{2N-k}, -T_{2N+k}, T_{4N-k}, T_{4N+k}, \dots$ take the same values at the $\mu_{j,N-1}$, $j = 0, \dots, N-1$,*
- *for $j \geqslant 0$, let*

$$m = |(j+N-1) (\mathrm{mod} 2N) - N + 1| \text{ and } p = \left\lfloor \frac{N+j}{2N} \right\rfloor,$$

the polynomials T_j and $(-1)^p T_m$ take the same values at the $\mu_{j,N-1}$, $j = 0, \dots, N-1$.

Proof. These are Theorems 1 and 2 of [71]. □

Let f be Lipschitz continuous over $[-1,1]$, we know [73, Chap. VI] that f admits a series expansion $\sum'_{n\geqslant 0} a_n T_n(x)$ in $L_2\left([-1,1],(1-x^2)^{-1/2}\right)$ which converges uniformly to f. We now state a straightforward consequence of Proposition A.1.

Corollary A.2. *Let $N\in\mathbb{N}, N\geqslant 1$, Let f be Lipschitz continuous over $[-1,1]$, its Chebyshev coefficients $(a_k)_{k\geqslant 0}$ and the coefficients $c_k, k=0,\dots,N-1$, of the interpolation polynomial p_{N-1} of f at the Chebyshev nodes of the first kind satisfy, for $k=0,\dots,N-1$,*

$$\begin{aligned} c_k &= a_k - a_{2N-k} - a_{2N+k} + a_{4N-k} + a_{4N-k} - \cdots \\ &= \sum_{j=0}^{+\infty} (-1)^j a_{2jN+k} + \sum_{j=1}^{+\infty} (-1)^j a_{2jN-k}. \end{aligned} \tag{A.1}$$

Let $N\in\mathbb{N}$, $N\geqslant 1$, we also define

$$\gamma_{\rho,0,N-1}=1 \text{ and } \gamma_{\rho,k,N-1}=\frac{1}{1-\rho^{-2N}}\left(1+\frac{1}{\rho^{2(N-k)}}\right) \text{ for } k=1,\dots,N-1.$$

Proposition A.3. *Let $\rho>1$, let $N\in\mathbb{N}$, $N\geqslant 1$, f be a function analytic in a neighbourhood of E_ρ, the coefficients $(c_k)_{k=0,\dots,N-1}$ of the interpolation polynomial of f at the Chebyshev nodes of the first kind satisfy*

$$|c_k| \leqslant 2\frac{M_\rho(f)}{\rho^k}\gamma_{\rho,k,N-1}, k=0,\dots,N-1,$$

where $M_\rho(f)=\max_{z\in\mathcal{E}_\rho}|f(z)|$. Moreover, we have

$$\|f-p_{N-1}\|_{\infty,[-1,1]} \leqslant \frac{4M_\rho(f)}{\rho^{N-1}(\rho-1)}.$$

Proof. First, we use the following consequence of [67, Thm 8.1]: the Chebyshev coefficients satisfy

$$|a_0|\leqslant M_\rho(f) \text{ and } |a_k|\leqslant 2\frac{M_\rho(f)}{\rho^k}. \tag{A.2}$$

Then, we combine Equation (A.1) and Inequalities (A.2) to obtain, for $k=1,\dots,N-1$.

$$\begin{aligned} |c_k| &\leqslant |a_k| + |a_{2N-k}| + |a_{2N+k}| + |a_{4N-k}| + |a_{4N-k}| + \cdots, \\ &\leqslant 2\frac{M_\rho(f)}{\rho^k} + 2\frac{M_\rho(f)}{\rho^{2N-k}} + 2\frac{M_\rho(f)}{\rho^{2N+k}} + 2\frac{M_\rho(f)}{\rho^{4N-k}} + 2\frac{M_\rho(f)}{\rho^{4N+k}} + \cdots, \\ &\leqslant 2\frac{M_\rho(f)}{\rho^k}\left(1+\frac{1}{\rho^{2N-2k}}+\frac{1}{\rho^{2N}}+\frac{1}{\rho^{4N-2k}}+\frac{1}{\rho^{4N}}+\cdots\right), \\ &\leqslant 2\frac{M_\rho(f)}{\rho^k}\frac{1}{1-\rho^{-2N}}\left(1+\frac{1}{\rho^{2(N-k)}}\right). \end{aligned}$$

Moreover, recall that $c_0=\frac{2}{N}\sum_{1\leqslant\ell\leqslant N} f(\mu_\ell)$, hence $|c_0|=2\max_{x\in[-1,1]}|f(x)|\leqslant 2M_\rho(f)$ by the maximum principle.

Now, we turn to the estimate on the remainder. Corollary A.2 yields, for any $x\in[-1,1]$,

$$f(x)-p_{N-1}(x)=\sum_{k\geqslant N} a_k(T_k(x)-(-1)^p T_m(x))$$

where m and p are defined as in Proposition A.1. Hence, we have, for any $x \in [-1,1]$,

$$|f(x) - p_{N-1}(x)| \leqslant \sum_{k \geqslant N} |a_k||T_k(x) - (-1)^p T_m(x)| \leqslant 2 \sum_{k \geqslant N} |a_k| \leqslant 4M_\rho(f) \sum_{k \geqslant N} \rho^{-k} = \frac{4M_\rho(f)}{\rho^{N-1}(\rho-1)}.$$

□

Regarding the two variable case, we start by establishing results analogous to [67, Thms 8.1 and 8.2]. Let f in $L_2\left([-1,1]\times[-1,1],(1-x^2)^{-1/2}(1-y^2)^{-1/2}\right)$, we denote by $\sum_{n_1\geqslant 0}{}' \sum_{n_2\geqslant 0}{}' a_{n_1,n_2}T_{n_1}(x)T_{n_2}(y)$ its series expansion.

Proposition A.4. *Let $\rho_1,\rho_2 > 1$, f be a function analytic in a neighbourhood of E_{ρ_1,ρ_2}, the coefficients $a_{n_1,n_2}, n_1, n_2 \geqslant 0$, of the Chebyshev series of f satisfy, for all n_1 and $n_2 \in \mathbb{N}$,*

$$|a_{n_1,n_2}| \leqslant 4\frac{M_{\rho_1,\rho_2}(f)}{\rho_1^{n_1}\rho_2^{n_2}}, \tag{A.3}$$

where $M_{\rho_1,\rho_2}(f) = \max_{z\in\mathcal{E}_{\rho_1,\rho_2}} |f(z)|$. Moreover, we have, for all n_1 and $n_2 \in \mathbb{N}$,

$$\left\| f(x,y) - \sum_{k_1=0}^{n_1}{}' \sum_{k_2=0}^{n_2}{}' a_{k_1,k_2}T_{k_1}(x)T_{k_2}(y) \right\|_{\infty,[-1,1]\times[-1,1]} \leqslant \frac{4\rho_1\rho_2 M_{\rho_1,\rho_2}(f)}{(\rho_1-1)(\rho_2-1)}\left(\frac{1}{\rho_1^{n_1+1}} + \frac{1}{\rho_2^{n_2+1}}\right).$$

Proof. For $\rho > 0$, we define $\mathcal{C}_\rho = \{z \in \mathbb{C}, |z| = \rho\}$. Extending what is done in the proof of [67, Thm. 8.1], we now introduce the change of variables $x = (z_1 + z_1^{-1})/2, y = (z_2 + z_2^{-1})/2$ where $z_1, z_2 \in \mathcal{C}_1$ and the function

$$f(x,y) = F(z_1,z_2) = \sum_{n_1\geqslant 0}{}' \sum_{n_2\geqslant 0}{}' a_{n_1,n_2} \frac{z^{n_1}+z^{-n_1}}{2}\frac{z^{n_2}+z^{-n_2}}{2}.$$

Now we use Cauchy's integral formula in two variables: for all $n_1, n_2 \in \mathbb{N}$,

$$\frac{1}{(2i\pi)^2}\int_{\mathcal{C}_1\times\mathcal{C}_1} \frac{F(z_1,z_2)}{z_1^{n_1+1}z_2^{n_2+1}}\,\mathrm{d}z_1\mathrm{d}z_2 = \frac{1}{2^{\delta_{0n_1}+\delta_{0n_2}}}\frac{2^{\delta_{0n_1}+\delta_{0n_2}}}{4}a_{n_1,n_2}.$$

If $g : (z_1,z_2) \mapsto ((z_1+z_1^{-1})/2, (z_2+z_2^{-1})/2$, the domain E_{ρ_1,ρ_2} is the image of $\mathcal{R}_{\rho_1}\times\mathcal{R}_{\rho_2}$, where $\mathcal{R}_\rho = \{z\in\mathbb{C}, \rho^{-1} < |z| < \rho\}$, via the application g. Note that, since $F = f\circ g$, F is analytic in a neighbourhood of $\mathcal{R}_{\rho_1}\times\mathcal{R}_{\rho_2}$ since it is the composition of two analytic functions, hence, for all $n_1, n_2 \in \mathbb{N}$,

$$a_{n_1,n_2} = \frac{1}{(i\pi)^2}\int_{\mathcal{C}_{\rho_1}\times\mathcal{C}_{\rho_2}} \frac{F(z_1,z_2)}{z_1^{n_1+1}z_2^{n_2+1}}\,\mathrm{d}z_1\mathrm{d}z_2,$$

from which follows

$$|a_{n_1,n_2}| \leqslant \frac{(2\pi)^2\rho_1\rho_2}{\pi^2}\frac{\max_{(z_1,z_2)\in\mathcal{C}_{\rho_1}\times\mathcal{C}_{\rho_2}}|F(z_1,z_2)|}{\rho_1^{n_1+1}\rho_2^{n_2+1}} = \frac{4M_{\rho_1,\rho_2}(f)}{\rho_1^{n_1}\rho_2^{n_2}} \text{ for all } n_1,n_2 \in \mathbb{N}.$$

As for the remainder, for all $x, y \in [-1,1]$, for all n_1 and $n_2 \in \mathbb{N}$, we have

$$f(x,y) - \sum_{k_1=0}^{n_1}{}' \sum_{k_2=0}^{n_2}{}' a_{k_1,k_2} T_{k_1}(x) T_{k_2}(y)$$
$$= \sum_{k_1 \geqslant n_1+1} \sum_{k_2 \geqslant 0}{}' a_{k_1,k_2} T_{k_1}(x) T_{k_2}(y) + \sum_{k_1=0}^{n_1}{}' \sum_{k_2 \geqslant n_2+1} a_{k_1,k_2} T_{k_1}(x) T_{k_2}(y),$$

hence,

$$\left\| f(x,y) - \sum_{k_1=0}^{n_1}{}' \sum_{k_2=0}^{n_2}{}' a_{k_1,k_2} T_{k_1}(x) T_{k_2}(y) \right\|_{\infty,[-1,1]\times[-1,1]}$$
$$= \sum_{k_1 \geqslant n_1+1} \sum_{k_2 \geqslant 0}{}' |a_{k_1,k_2}| + \sum_{k_1=0}^{n_1}{}' \sum_{k_2 \geqslant n_2+1} |a_{k_1,k_2}|,$$
$$\leqslant 4 M_{\rho_1,\rho_2}(f) \left(\frac{1}{\rho_1^{n_1+1}} + \frac{1}{\rho_2^{n_2+1}} \right) \frac{\rho_1 \rho_2}{(\rho_1 - 1)(\rho_2 - 1)}.$$

□

Now we can prove:

Proposition A.5. *Let $\rho_1, \rho_2 > 1$, let $M_1, M_2 \in \mathbb{N}, M_1, M_2 \geqslant 2$, f be a function analytic in a neighbourhood of E_{ρ_1,ρ_2}, the coefficients $c_{k_1,k_2}, k_1 = 0, \ldots, M_1 - 1, k_2 = 0, \ldots, M_2 - 1$ of the interpolation polynomial P_{M_1-1,M_2-1} of f at pairs of Chebyshev nodes of the first kind satisfy, for $k_1 = 1, \ldots, M_1 - 1$, $k_2 = 1, \ldots, M_2 - 1$,*

$$|c_{k_1,k_2}| \leqslant 4 \frac{M_{\rho_1,\rho_2}(f)}{\rho_1^{k_1} \rho_2^{k_2}} \gamma_{\rho_1,k_1,M_1-1} \gamma_{\rho_2,k_2,M_2-1},$$

where $M_{\rho_1,\rho_2}(f) = \max_{z \in \mathcal{E}_{\rho_1,\rho_2}} |f(z)|$. Moreover, we have

$$\|f - P_{M_1-1,M_2-1}\|_{\infty,[-1,1]\times[-1,1]} \leqslant \frac{16 \rho_1 \rho_2 M_{\rho_1,\rho_2}(f)}{(\rho_1 - 1)(\rho_2 - 1)} \left(\frac{1}{\rho_1^{M_1}} + \frac{1}{\rho_2^{M_2}} \right).$$

Proof. Let $\sum_{n_1 \geqslant 0}{}' \sum_{n_2 \geqslant 0}{}' a_{n_1,n_2} T_{n_1}(x) T_{n_2}(y)$ the series expansion of f, the aliasing phenomenon presented above still exists: for $k_1 = 0, \ldots, M_1 - 1$, $k_2 = 0, \ldots, M_2 - 1$,

$$c_{k_1,k_2} = \sum_{p_1=0}^{+\infty} \sum_{p_2=0}^{+\infty} (-1)^{p_1+p_2} a_{2p_1M_1+k_1,2p_2M_2+k_2} \tag{A.4}$$
$$+ \sum_{p_1=0}^{+\infty} \sum_{p_2=1}^{+\infty} (-1)^{p_1+p_2} a_{2p_1M_1+k_1,2p_2M_2-k_2} + \sum_{p_1=1}^{+\infty} \sum_{p_2=0}^{+\infty} (-1)^{p_1+p_2} a_{2p_1M_1-k_1,2p_2M_2+k_2}$$
$$+ \sum_{p_1=1}^{+\infty} \sum_{p_2=1}^{+\infty} (-1)^{p_1+p_2} a_{2p_1M_1-k_1,2p_2M_2-k_2}.$$

We now combine Equation (A.4) and Inequalities (A.3) to obtain, for $k_1 = 1, \dots, M_1 - 1, k_2 = 1, \dots, M_2 - 1$,

$$\begin{aligned}
|c_{k_1,k_2}| &\leqslant \sum_{p_1=0}^{+\infty}\sum_{p_2=0}^{+\infty} |a_{2p_1M_1+k_1,2p_2M_2+k_2}| + \sum_{p_1=0}^{+\infty}\sum_{p_2=1}^{+\infty} |a_{2p_1M_1+k_1,2p_2M_2-k_2}| \\
&\quad + \sum_{p_1=1}^{+\infty}\sum_{p_2=0}^{+\infty} |a_{2p_1M_1-k_1,2p_2M_2+k_2}| + \sum_{p_1=1}^{+\infty}\sum_{p_2=1}^{+\infty} |a_{2p_1M_1-k_1,2p_2M_2-k_2}| \\
&\leqslant \sum_{p_1=0}^{+\infty}\sum_{p_2=0}^{+\infty} 4\frac{M_{\rho_1,\rho_2}(f)}{\rho_1^{2p_1M_1+k_1}\rho_2^{2p_2M_2+k_2}} + \sum_{p_1=0}^{+\infty}\sum_{p_2=1}^{+\infty} 4\frac{M_{\rho_1,\rho_2}(f)}{\rho_1^{2p_1M_1+k_1}\rho_2^{2p_2M_2-k_2}} \\
&\quad + \sum_{p_1=1}^{+\infty}\sum_{p_2=0}^{+\infty} 4\frac{M_{\rho_1,\rho_2}(f)}{\rho_1^{2p_1M_1-k_1}\rho_2^{2p_2M_2+k_2}} + \sum_{p_1=1}^{+\infty}\sum_{p_2=1}^{+\infty} 4\frac{M_{\rho_1,\rho_2}(f)}{\rho_1^{2p_1M_1-k_1}\rho_2^{2p_2M_2-k_2}} \\
&\leqslant 4\frac{M_{\rho_1,\rho_2}(f)}{\rho_1^{k_1}\rho_2^{k_2}} \frac{1}{1-\rho_1^{-2M_1}} \frac{1}{1-\rho_2^{-2M_2}} \\
&\quad \left(1 + \frac{1}{\rho_1^{2(M_1-k_1)}} + \frac{1}{\rho_2^{2(M_2-k_2)}} + \frac{1}{\rho_1^{2(M_1-k_1)}\rho_2^{2(M_2-k_2)}}\right).
\end{aligned}$$

If we use Equation (6.2), we get

$$\begin{aligned}
&|c_{0,0}| \leqslant 4M_{\rho_1,\rho_2}(f) \text{ thanks to the maximum principle,} \\
&|c_{k_1,0}| \leqslant 4\frac{M_{\rho_1,\rho_2}(f)}{\rho_1^{k_1}} \frac{1}{1-\rho_1^{-2M_1}} \left(1 + \frac{1}{\rho_1^{2(M_1-k_1)}}\right) \text{ for } k_1 = 1, \dots, M_1 - 1, \\
&|c_{0,k_2}| \leqslant 4\frac{M_{\rho_1,\rho_2}(f)}{\rho_2^{k_2}} \frac{1}{1-\rho_2^{-2M_2}} \left(1 + \frac{1}{\rho_2^{2(M_2-k_2)}}\right) \text{ for } k_2 = 1, \dots, M_2 - 1.
\end{aligned}$$

The last two inequalities are consequences of Proposition A.3.

As for the remainder, for all $x, y \in [-1, 1]$, for all n_1 and $n_2 \in \mathbb{N}$, we have thanks to the aliasing phenomenon

$$\begin{aligned}
&f(x,y) - {\sum_{k_1=0}^{M_1-1}}' {\sum_{k_2=0}^{M_2-1}}' c_{k_1,k_2} T_{k_1}(x) T_{k_2}(y) \\
&\quad = {\sum_{k_1 \geqslant M_1}} {\sum_{k_2 \geqslant 0}}' a_{k_1,k_2} (T_{k_1}(x) - (-1)^{p_1} T_{m_1}(x))(T_{k_2}(y) - (-1)^{p_2} T_{m_2}(y)) \\
&\qquad + {\sum_{k_1=0}^{M_1-1}}' {\sum_{k_2 \geqslant M_2}} a_{k_1,k_2} (T_{k_1}(x) - (-1)^{p_1} T_{m_1}(x))(T_{k_2}(y) - (-1)^{p_2} T_{m_2}(y)),
\end{aligned}$$

where m_1, m_2 and p_1, p_2 are defined as in Proposition A.1. Hence,

$$\begin{aligned}
\|f(x,y) - P_{M_1-1,M_2-1}(x,y)\|_{\infty,[-1,1]\times[-1,1]} &= {\sum_{k_1 \geqslant M_1}} {\sum_{k_2 \geqslant 0}}' 4|a_{k_1,k_2}| \\
&\quad + {\sum_{k_1=0}^{M_1-1}}' {\sum_{k_2 \geqslant M_2}} 4|a_{k_1,k_2}| \leqslant 16 M_{\rho_1,\rho_2}(f) \left(\frac{1}{\rho_1^{M_1}} + \frac{1}{\rho_2^{M_2}}\right) \frac{\rho_1\rho_2}{(\rho_1-1)(\rho_2-1)},
\end{aligned}$$

thanks to Proposition A.4. □

The next lemma eases the computations. Recall that we introduce in Section 4.1.2 $\eta_{\rho,0} = 1$ and $\eta_{\rho,k} = (\rho^2+1)(\rho^2-1)$ for $k = 1, \dots, N-1$.

Lemma A.6. *Let $\rho > 1$, $N \geqslant 2$, for $k = 0, \dots, N-1$, we have*

$$\gamma_{\rho,k,N-1} \leqslant \eta_{\rho,k}.$$

In particular, if $\rho \geqslant 2$, $\eta_{\rho,k} \leqslant 2$ for $k = 1, \dots, N-1$.

Proof. The case $k = 0$ is straightforward. For $k = 1, \dots, N-1$, we have

$$\gamma_{\rho,k,N-1} = \frac{1}{1-\rho^{-2N}}\left(1 + \frac{1}{\rho^{2(N-k)}}\right) = \frac{1}{\rho^{2N}-1}\left(\rho^{2N} + \rho^{2k}\right) \leqslant \frac{\rho^{2N}}{\rho^{2N}-1}\left(1+\rho^{-2}\right).$$

The function $u \mapsto u/(u-1)$ is strictly decreasing over $(1, +\infty)$, hence

$$\gamma_{\rho,k,N-1} \leqslant \frac{\rho^{2N}}{\rho^{2N}-1}\left(1+\rho^{-2}\right) \leqslant \frac{\rho^2}{\rho^2-1}\left(1+\rho^{-2}\right) = \frac{\rho^2+1}{\rho^2-1}.$$

The last statement is obvious. □

Proof of Proposition 4.1. We introduce $\widehat{f} : \widehat{f}(z) = f\left(z\frac{b-a}{2} + \frac{a+b}{2}\right)$ for any z in a suitable neighbourhood of E_ρ. The coefficients c_k are also the coefficients of the interpolation polynomial in $\mathbb{R}_{N-1}[x]$ of $\widehat{f}$ at the Chebyshev nodes of the first kind. Therefore, we obtain Proposition 4.1 by applying Proposition A.3 to $\widehat{f}$ and Lemma A.6.

Proof of Proposition 4.2. It is identical to the previous one: it suffices to introduce $\widehat{f} : \widehat{f}(z_1, z_1) = f\left(z_1\frac{b_1-a_1}{2} + \frac{a_1+b_1}{2}, z_2\frac{b_2-a_2}{2} + \frac{a_2+b_2}{2}\right)$ for any (z_1, z_2) in a suitable neighbourhood of E_{ρ_1,ρ_2}, and then to apply Proposition A.5 to $\widehat{f}$ and Lemma A.6.

Appendix B. Proof of Theorem 5.27

For $f(z) = \exp(\exp(z))$, we take $K = \log d$. A sufficient condition for success of Algorithm 2 is, in view of Proposition 5.13, (for d large enough)

$$b - a < \frac{1}{4}\left((|a+b| + \log d) M_{\mathcal{D}_{a,b,K}}(f)\right)^{-4/(3(1-\lambda^2)d)(1+o(1))} (uv)^{-4/(3(1-\lambda^2)d)(1+o(1))} \log d.$$

We have $M_{\mathcal{D}_{a,b,K}}(f) = 2^{O(d)}$; hence, as in the proof of Theorem 5.26, a sufficient condition is

$$(uv)^{-4/(3(1-\lambda^2)d)} \log d \to \infty,$$

for which it suffices that

$$d \log\log d \geqslant \left(\frac{4}{3(1-\lambda^2)} + \varepsilon'\right)\log(uv),$$

for some $\varepsilon' > 0$, which follows from the assumption of the Theorem. The statement on w follows from simple calculus using $w = O(K^{N(1-\lambda)})$.

For $g(z) = \sum_{n\geqslant 0} \exp(-n^2) z^n$, we have

$$\max_{|z|=\rho} |g(z)| = g(\rho) = \exp(\log^2\rho/4) \sum_{n\geqslant 0} \exp(-(n - \log\rho/2)^2) \leqslant \exp(\log^2\rho/4) \sum_{n\in\mathbb{Z}} \exp(-n^2) \leqslant 2\exp(\log^2\rho/4).$$

We choose $K = \exp(3(1-\lambda^2)d/2)$. In this case, we have $M_{\mathcal{D}_{a,b,K}}(f) = \exp(9(1-\lambda^2)^2d^2/16 + O(d))$.

A sufficient success condition for Algorithm 2 is thus, as $d \to \infty$,

$$b - a < \frac{1}{4}(|a+b| + \exp(3(1-\lambda^2)d/2))^{-4/(3(1-\lambda^2)d)(1+o(1))} \exp(3(1-\lambda^2)d/2)(uvM_{\mathcal{D}_{a,b,K}}(f))^{-4/(3(1-\lambda^2)d)(1+o(1))}.$$

It follows from

$$\frac{3(1-\lambda^2)d}{2} - \frac{4(1+o(1))}{3(1-\lambda^2)d}\left(\log(uv) + \frac{9(1-\lambda^2)^2d^2}{16}\right) \to \infty,$$

for which it suffices to have

$$d^2 \geqslant \left(\frac{16}{9(1-\lambda^2)^2} + \varepsilon'\right)\log(uv),$$

for some $\varepsilon' > 0$.

Again, the statement on w follows from simple calculus using $w = O(K^{N(1-\lambda)})$.

Appendix C. Precision required, 1D case

We now estimate the precision required for the computations performed in Algorithm 1. We shall use a computation model where our real numbers are represented by fixed point numbers, with $\mathfrak{p} \geqslant 1$ binary digits following the binary point. This means that for each elementary operation, the result differs from the ideal mathematical result by at most $2^{-\mathfrak{p}}$ (such a result is usually called faithful rounding in precision $\mathfrak{p}$). The notations used hereafter correspond to those of Algorithm 1. This somewhat artificial model is a simplification of the natural model, which is a floating-point model where the total precision is $\mathfrak{p} + P$, where P is the size of the largest real number encountered during the computation. It allows for a simpler, though probably slightly rougher, analysis; as a consequence, Theorem C.9 is valid for floating-point computations in precision $\mathfrak{p} + P$.

The following lemma summarizes basic facts on this model:

Lemma C.1. *Let x, y be real numbers and X, Y be fixed-point numbers in precision $\mathfrak{p}$ which are approximations of those, such that $|X - x| \leqslant \varepsilon_x, |Y - y| \leqslant \varepsilon_y$, with $\max(\varepsilon_x, \varepsilon_y) < 1/2$. Then, if $\oplus$ and $\otimes$ are the arithmetic operations of our computational model, we have*

$$|(X \oplus Y) - (x + y)| \leqslant \varepsilon_x + \varepsilon_y, \tag{C.1}$$

$$|X \otimes Y - x \cdot y| \leqslant \varepsilon_x Y + |x|\varepsilon_y + 2^{-\mathfrak{p}} \leqslant \varepsilon_x|y| + |x|\varepsilon_y + \varepsilon_x\varepsilon_y + 2^{-\mathfrak{p}}. \tag{C.2}$$

Further, if Z_1 is the fixed point result of the operation $\exp(X)$, and if g is a C^1 function over $[a, b]$, Z_2 the fixed point result of the operation $g(X)$, we have:

$$|Z_1 - \exp(x)| \leqslant 2^{-\mathfrak{p}} + 2\varepsilon_x \exp(x), |Z_2 - f(x)| \leqslant 2^{-\mathfrak{p}} + \varepsilon_x \max_{[x-\varepsilon, x+\varepsilon]} |g'|. \tag{C.3}$$

As a consequence, if $x_1, \ldots, x_n$ are real numbers and $X_1, \ldots, X_n$ be fixed-point numbers in precision $\mathfrak{p}$ which are approximations of those such that $\max_{1 \leqslant i \leqslant n} |x_i - X_i| \leqslant \varepsilon$, the error on the sum $x_1 + \cdots + x_n$ (which can be evaluated in any order) is at most $n\varepsilon$.

In the sequel, we shall put

$$C = \max(1, u, v) \max(1, B_x, B_f) \max(1, \max_{[a,b]} |f'|).$$

The error analysis of the DCT follows:

Proposition C.2. *Assume that each $\varphi(L_{cheb}[i])$ is given by an approximation with error $\varepsilon < 1$, with $\varepsilon \geqslant 2^{-\mathfrak{p}}$ and that the cosines involved in the DCT definition are given by an approximation $\leqslant 1$ with error $2^{-\mathfrak{p}}$. Assume that we compute each coefficient of the DCT by computing first the N products, then the sum. Then, we obtain an approximation Δ of* DCT-II(U) *at Step 11 of Algorithm 1 such that*

$$\|\Delta - \text{DCT-II}(U)\|_\infty \leqslant N(\varepsilon + 2^{-\mathfrak{p}}(1 + C^d)).$$

Proof. We deduce from (C.2) that each product $\varphi(L_{cheb}[i]) \cos(k(i + 1/2)\pi)/N$ incurs an error at most $\varepsilon + 2^{-\mathfrak{p}}|\varphi(L_{cheb}[i])| + 2^{-\mathfrak{p}} \leqslant \varepsilon + 2^{-\mathfrak{p}}(1 + C^d)$. The sum of those terms then, cf. (C.1), incurs an error $\leqslant N(\varepsilon + 2^{-\mathfrak{p}}(1 + C^d))$, from which the result follows. □

We now turn to the computation of α^k; our practical application cases have $\alpha \gg 1$, and we need all the values $\alpha, \ldots, \alpha^k$, so that we compute α^k by the recurrence $\alpha^k = \alpha^{k-1} \cdot \alpha$.

Proposition C.3. *Let x be a nonnegative real number, and X a fixed point number in precision $\mathfrak{p}$ approximating x, so that $|X - x| \leqslant \varepsilon$, with $1 \geqslant \varepsilon \geqslant 2^{-\mathfrak{p}}$. If k is an integer, and if we define $X_1 = X$ and $X_k = X \otimes X_{k-1}$ we have, for $k \geqslant 1$,*

$$|X_k - x^k| \leqslant k\varepsilon(x + 1)^{k-1}.$$

Proof. Induction on k, clear for $k = 1$. We let ε_k be $|X_k - x^k|$. Then, we have

$$\begin{aligned} |X_{k+1} - x^{k+1}| &= |X_{k+1} - X \cdot X_k| + |X \cdot (X_k - x^k)| + |X - x|x^k \\ &\leqslant 2^{-\mathfrak{p}} + (x + \varepsilon)\varepsilon_k + x^k\varepsilon, \end{aligned}$$

from (C.2) and the induction hypothesis, so that

$$\varepsilon_{k+1} \leqslant (x + \varepsilon)\varepsilon_k + x^k\varepsilon + 2^{-\mathfrak{p}} \leqslant (x + \varepsilon)\varepsilon_k + (x^k + 1)\varepsilon \leqslant (x + 1)\varepsilon_k + (x + 1)^k\varepsilon,$$

from which the result follows by induction. □

Corollary C.4. *Assume that $\varepsilon \leqslant 1/d$. Let α be a real number with $|\alpha| \leqslant B_x$, and β be a real number with $|\beta| \leqslant B_f$. If X is a fixed-point approximation of $u\alpha$ with error $\leqslant \varepsilon$ and Y is a fixed-point approximation of $v\beta$ with error $\leqslant \varepsilon$, and if X_k and Y_ℓ are defined as in Proposition C.3, we define $Z = X_k \otimes Y_\ell$. If $k + \ell \leqslant d$, we have*

$$|Z - (u\alpha)^k(v\beta)^\ell| \leqslant 2^{-\mathfrak{p}} + 2d\varepsilon(C + 1)^{d-1}.$$

Proof. By Proposition C.3, the error on X_k compared to $(u\alpha)^k$ is $\leqslant k\varepsilon(C + 1)^{k-1}$; the error on Y_ℓ compared to $(v\beta)^\ell$ is at most $\leqslant \ell\varepsilon(C + 1)^{\ell-1}$. Finally, we have

$$\begin{aligned} |Z - (u\alpha)^k(v\beta)^\ell| &\leqslant |Z - X_k \cdot Y_\ell| + |X_k - (u\alpha)^k||Y_\ell| + |u\alpha|^k|Y_\ell - (v\beta)^\ell| \\ &\leqslant 2^{-\mathfrak{p}} + k\varepsilon(C + 1)^{k-1}(C^\ell + \ell\varepsilon(C + 1)^{\ell-1}) + \ell\varepsilon(C + 1)^{\ell-1}C^k \\ &\leqslant 2^{-\mathfrak{p}} + 2d\varepsilon(C + 1)^{d-1}, \end{aligned}$$

using (C.2) and $\ell\varepsilon \leqslant d\varepsilon \leqslant 1$. □

We can now combine the previous results to get an estimate of the precision required for $M_c[i,j]$; for the sake of simplicity, we assume that approximations of the $\cos((k+1/2)\pi/N)$'s to the precision $2^{-\mathfrak{p}}$ are known and that all are less than 1.

Theorem C.5. *Assume that a, b, u, v are exactly representable in our computation model. Then, the error on the values $uL_{cheb}[i]$ and $vf(L_{cheb}[i])$ is at most $2^{3-\mathfrak{p}}C$, and the error on the vector L_{DCT} is at most $d(C+1)^d 2^{4-\mathfrak{p}}$.*

Proof. The computations of $(b-a)/2$ and $(b+a)/2$ each incur an error $\leqslant 2^{-\mathfrak{p}}$. Hence, we deduce from Lemma C.1 that $L_{cheb}[i]$ is computed with an error

$$\leqslant \underbrace{2^{-\mathfrak{p}} + 1\cdot 2^{-\mathfrak{p}} + (b-a)/2\cdot 2^{-\mathfrak{p}}}_{\text{error on } (b-a)/2\cos((j+1/2)\pi/N)} + 2^{-\mathfrak{p}} = (3+(b-a)/2)\cdot 2^{-\mathfrak{p}}.$$

Hence, $f(L_{cheb}[i])$ incurs an error $\leqslant 2^{-\mathfrak{p}}\left(1+(3+(b-a)/2)\max_{[a,b]}|f'|\right)$, as we know that $L_{cheb}[i]$ is in $[a,b]$ and can always ensure that the approximation of $L_{cheb}[i]$ is also in $[a,b]$, up to replacing it by $\min(b,\max(a,L_{cheb}[i]))$.

Thus, the error incurred on ux and $vf(x)$ for $x = L_{cheb}[i]$ is at most, cf. (C.3),

$$2^{-\mathfrak{p}} + 2^{-\mathfrak{p}}\max(u\cdot(3+(b-a)/2), v(1+(3+(b-a)/2\max_{[a,b]}|f'|))) \leqslant 2^{-\mathfrak{p}}(1+5C) \leqslant 6\cdot 2^{-\mathfrak{p}}C.$$

Corollary C.4 finally bounds the error on U by $2^{-\mathfrak{p}}(1+12d(C+1)^{d-1}C) \leqslant 13d(C+1)^d 2^{-\mathfrak{p}}$.

Hence, thanks to Proposition C.2, the overall error on DCT-II(U) is at most $N(13d(C+1)^d 2^{-\mathfrak{p}} + 2^{-\mathfrak{p}}(1+C^d)) \leqslant N(13d+1)(C+1)^d 2^{-\mathfrak{p}}$. As N is exactly representable, after multiplication by $2/N$ (or $1/N$ for the zero-th coefficient), we obtain, from (C.2), an error on L_{DCT} of at most

$$(13d+1)(C+1)^d 2^{-\mathfrak{p}} + 2^{-\mathfrak{p}} \leqslant d(C+1)^d 2^{4-\mathfrak{p}}.$$

□

Remark C.6. This theorem can be read as a proof in this case of the rule of thumb valid in this computational model that one should use as a precision "the final precision required, plus the size of the largest element encountered in the computation, plus a few guard bits".

We now turn to similar estimates for the remainders. For the sake of simplicity again, we shall assume that $\rho, \rho-1, \omega$ are exactly representable in our computational model – which is, in practice, a very mild restriction. As N is an integer and $\mathfrak{p} \geqslant 0$, N and $N-1$ are also exactly representable in our computational model.

We compute R_{ω_0} as $\exp(-(N-1-\omega_0)\log\rho)/(\rho-1)/4)$.

Proposition C.7. *Define $C' = \max(1, (N-\omega_0)/(\rho-1))$. Then, the quantity R_{ω_0} can be computed with error at most $7\cdot 2^{-\mathfrak{p}}C'$, and the quantity $-\log_2(R_{\omega_0}) + \log_2(N)$ with error at most $5\cdot 2^{-\mathfrak{p}}(\rho-1)C'$.*

Proof. The error on $(N-1-\omega_0)\log\rho$ is at most, cf. (C.2), $2^{-\mathfrak{p}}(\log\rho + N - 1 - \omega_0 + 1) \leqslant 2C'(\rho-1)2^{-\mathfrak{p}}$.

The error on $\rho^{-(N-1-\omega_0)}$ is upper bounded by $2^{-\mathfrak{p}}(1+4C'(\rho-1))\rho^{-(N-1-\omega_0)} \leqslant 5\cdot 2^{-\mathfrak{p}}C'(\rho-1)\rho^{-(N-1-\omega_0)}$. Then, we deduce from (C.3) that the error on $\rho^{-(N-1-\omega_0)}/(\rho-1)$ is $\leqslant 2^{-\mathfrak{p}}(1+5C'\rho^{-(N-1-\omega_0)}) \leqslant 6C'2^{-\mathfrak{p}}$, and division by 4 incurs a precision loss of $2^{-\mathfrak{p}}$, so the total error on R_{ω_0} is $\leqslant 7\cdot 2^{-\mathfrak{p}}C'$.

Similarly, the error on $(N-1-\omega_0)\log_2\rho$ is at most $2^{-\mathfrak{p}}(\log_2\rho+N-1-\omega_0+1)\leqslant 3C'(\rho-1)2^{-\mathfrak{p}}$, while the errors on $\log_2(\rho-1)$ and $\log_2(N)$ are each at most $2^{-\mathfrak{p}}$. Hence, the total error on $-\log_2(R_{\omega_0})+\log_2(N)$ is at most $5C'(\rho-1)2^{-\mathfrak{p}}$. □

As we inherently have to allow for overestimation of the quantity B_f, as has been pointed, up to rounding upwards this overestimated quantity we shall assume that B_f is exactly representable.

Corollary C.8. *The error on $R_{\omega_0}(uB_x)^k(vB_f)^\ell$ is at most $2^{-\mathfrak{p}}C'(C+4u\rho)(C+1)^{d-1}(24d+8)$.*

Proof. The error on B_x is at most, cf. (C.1), the sum of the error on $(a+b)/2$, which is $\leqslant 2^{-\mathfrak{p}}$, and the error on the product $(b-a)(\rho+\rho^{-1})/4$, which is $\leqslant 2^{-\mathfrak{p}}((b-a)/4+\rho+\rho^{-1}+2^{-\mathfrak{p}})+2^{-\mathfrak{p}}\leqslant((b-a)/4+4\rho)2^{-\mathfrak{p}}$, cf. (C.2). Hence, the error on uB_x is at most $(C+4u\rho)2^{-\mathfrak{p}}$ and the error on vB_f at most $C\cdot 2^{-\mathfrak{p}}$; thus, Corollary C.4 bounds the error on $(uB_x)^k(vB_f)^\ell$ by $2^{-\mathfrak{p}}+2d(C+4u\rho)(C+1)^{d-1}2^{-\mathfrak{p}}\leqslant 3d(C+4u\rho)(C+1)^{d-1}2^{-\mathfrak{p}}$.

Further, note that $\max((uB_x)^k,(vB_f)^\ell)\leqslant C^d$ and $R_{\omega_0}\leqslant 1/(\rho-1)$, and recall that the error on R_{ω_0} is at most $7\cdot 2^{-\mathfrak{p}}C'$ (cf. Proposition C.7). Hence, finally, the error on the product is at most

$$7\cdot 2^{-\mathfrak{p}}C'(C^d+3d(C+4u\rho)(C+1)^{d-1}2^{-\mathfrak{p}})+3d(C+4u\rho)\frac{(C+1)^{d-1}}{\rho-1}2^{-\mathfrak{p}}+2^{-\mathfrak{p}},$$

which is in turn upper bounded by

$$2^{-\mathfrak{p}}C'(C+4u\rho)(C+1)^{d-1}(24d+8),$$

as claimed. □

Now we can state:

Theorem C.9. *Put $\mathcal{M}=\max(u,v,|a|,|b|,\rho,B_f,\max_{[a,b]}|f'|)$. For $\mathfrak{p}\geqslant \mathsf{tprec}+O(\max(d\log\mathcal{M},|\log(\rho-1)|))$, if the faithful rounding mode is set at each step, the computation in a fixed precision model with $\mathfrak{p}$ bits after the binary point allows for the computation of $\mathsf{tprec}_{\mathsf{comp}}, M_{c,\mathsf{comp}}, M_{r,\mathsf{comp}}$ with the property that $\mathsf{tprec}_{\mathsf{comp}}\in\{\mathsf{tprec},\mathsf{tprec}+1\}$, $M_c[i,j]-M_{c,\mathsf{comp}}[i,j]\in\{0,\operatorname{sgn}(M_c[i,j])\}$, and $M_r[i,j]-M_{r,\mathsf{comp}}[i,j]\in\{0,1\}$ for $i,j=0,\dots,N-1$.*

Proof. Follows from Proposition C.7, Theorems C.5 and C.8 and the discussion in Subsection 5.3.3. □

Note as a conclusion that as the largest real number encountered in this computation has size $O(\max(d\log\mathcal{M},|\log(\rho-1)|)$, the result stated in the theorem remains valid in a floating-point model with precision $\mathsf{tprec}+O(\max(d\log\mathcal{M},|\log(\rho-1)|))$.

Appendix D. Lemmata on φ,ψ

In this appendix, we group the facts concerning the function ψ of Section 6.

Lemma D.1. *Let φ be the function from $[1,+\infty)$ to $[1,+\infty)$ defined by $\varphi(x)=(1+\lfloor x\rfloor)(x-\lfloor x\rfloor/2)$. Then φ is continuous and strictly increasing, and defines a bijection from $[1,+\infty)$ to $[1,+\infty)$. For any $x\geqslant 1$, we have $x(x+1)/2\leqslant\varphi(x)\leqslant(x+1/2)^2/2$ and $\sqrt{2x}-1/2\leqslant\varphi^{-1}(x)\leqslant\sqrt{2x+1/4}-1/2$.*

Proof. For $x \notin \mathbb{Z}$, it is clear that φ is $\mathcal{C}^1$ in a neighbourhood of x and that $\varphi'(x) \geqslant 1$.

For x in $\mathbb{Z}$, we have $\varphi(x) = \lim_{t\to x^+} \varphi(t) = (1+x)x/2$, whereas $\lim_{t\to x^-} \varphi(t) = x(x-(x-1)/2) = x(x+1)/2$. This proves continuity, and the remaining assertions follow.

Let $k \in \mathbb{N}, a \in \mathbb{R}$ and $g_a(x) = (x+1-a)(x+a)/2$. We denote by φ_k the restriction of φ to $[k, k+1]$. For all $x \in [k, k+1]$, $(\varphi_k - g_a)'(x) = k + 1/2 - x$: the function $\varphi_k - g_a$ is decreasing over $[k, k+1/2]$ and increasing over $[k+1/2, k+1]$. Now, we remark that $\varphi_k(k) = g_0(k)$ and $\varphi_k(k+1) = g_0(k+1)$, which yields $\varphi_k(x) \geqslant g_0(x) = x(x+1)/2$ for all $x \in [k, k+1]$, and $\varphi_k(k+1/2) = g_{1/2}(k+1/2)$, which yields $\varphi_k(x) \leqslant g_{1/2}(x) = (x+1/2)^2/2$ for all $x \in [k, k+1]$.

The proof of the remaining inequalities is straightforward.

□

Lemma D.2. *Let $3 \leqslant \gamma \leqslant N$, $s = \gamma\varphi^{-1}(N/\gamma)$, $N_1 = \lfloor\sqrt{2N\gamma}\rfloor$ and $N_2 = \lceil\sqrt{2N/\gamma}\rceil$. We have $\gamma \geqslant N_1/N_2 \geqslant 1$, $s < N_1$ and $\sqrt{2N}(\sqrt{2N} - \sqrt{3}/3) \leqslant N_1N_2 \leqslant (2+\sqrt{2})N$.*

Assume that $N \to \infty$, then $\operatorname{card}\mathcal{K}_s \leqslant N + O(s/\gamma)$.

Proof. We have

$$\sqrt{2N\gamma} - \sqrt{2N/\gamma} = \sqrt{2N\gamma}(1 - 1/\gamma) \geqslant 1,$$

since $N \geqslant \gamma \geqslant 3$. It follows $N_1 = \lfloor\sqrt{2N\gamma}\rfloor \geqslant \lceil\sqrt{2N/\gamma}\rceil = N_2$, hence $N_1/N_2 \geqslant 1$. Moreover, for any $\gamma \geqslant 3$,

$$N_1 = \lfloor\sqrt{2N\gamma}\rfloor \leqslant \gamma\sqrt{2N/\gamma} \leqslant \gamma N_2.$$

Also,

$$(\sqrt{2N\gamma} - 1)\sqrt{2N/\gamma} \leqslant N_1N_2 \leqslant \sqrt{2N\gamma}(1 + \sqrt{2N/\gamma}),$$

hence

$$\sqrt{2N}(\sqrt{2N} - \sqrt{3}/3) \leqslant N_1N_2 \leqslant (2+\sqrt{2})N$$

since $N \geqslant \gamma \geqslant 2$.

Finally, from Lemma D.1 and the fact that $(\sqrt{8x+1} - 1)/2 < \sqrt{2x} - 3/8$ for all $x \geqslant 1$, we have $s = \gamma\varphi^{-1}(N/\gamma) < \gamma(\sqrt{2N/\gamma} - 3/8) < \sqrt{2N\gamma} - 1 \leqslant N_1$ since $N \geqslant \gamma \geqslant 3$. Therefore, we can apply Lemma 6.3: we have, from (6.6),

$$\operatorname{card}\mathcal{K}_s = \gamma(1+\lfloor s/\gamma\rfloor)(s/\gamma - \lfloor s/\gamma\rfloor/2) + O(s/\gamma) \leqslant \gamma\varphi(s/\gamma) + O(s/\gamma) \leqslant N + O(s/\gamma).$$

□

Corollary D.3. *With the assumptions of Lemma D.2, put $\lambda = N/\gamma$. Then, we have*

$$\Omega_\gamma(N, N_1, N_2) = \psi(\lambda)N\gamma + O(N),$$

where

$$\psi(\lambda) = \frac{1 + \lfloor\varphi^{-1}(\lambda)\rfloor}{12\lambda}\left(6\varphi^{-1}(\lambda)^2 - \lfloor\varphi^{-1}(\lambda)\rfloor - 2\lfloor\varphi^{-1}(\lambda)\rfloor^2\right).$$

Note that if $\gamma = o(N)$, we have $\lambda \to \infty$ and $\varphi^{-1}(\lambda) = \sqrt{2\lambda} + O(1)$, so that $\psi(\lambda) = 2\sqrt{2\lambda}/3 + O(1)$ and

$$\Omega_\gamma(N, N_1, N_2) = \frac{2\sqrt{2}}{3}N^{3/2}\gamma^{1/2} + O(N\gamma).$$

Proof. Lemma D.2 shows us that the assumptions of Lemma 6.3 are satisfied; we thus get

$$\sum_{(i,j)\in\mathcal{K}_s} (i+j\gamma) = \psi(\lambda)N\gamma + O(N).$$

Further, note that for our value of s, the term $s(N - \operatorname{card}\mathcal{K}_s)$ from Lemma 6.1 is $O(s^2/\gamma)$, which is $O(\gamma\varphi^{-1}(\lambda)^2) = O(N\varphi^{-1}(\lambda)^2/\lambda) = O(N)$, thanks to Lemma D.1. The result follows. □

Lemma D.4. *For $x \in [1, +\infty)$, we have*

$$-5/6 < \psi(x) - \frac{2\sqrt{2x}}{3} < 0.$$

Proof. We have

$$\psi(\varphi(x)) = \frac{6x^2 - \lfloor x\rfloor - 2\lfloor x\rfloor^2}{12x - 6\lfloor x\rfloor}.$$

For $v \leqslant u < v+1$, define

$$F(u,v) = \frac{6u^2 - v - 2v^2}{12u - 6v} - 2u/3,$$

so that $\psi(\varphi(x)) - 2x/3 = F(x, \lfloor x\rfloor)$.

We maximize $F(u,v)$ for fixed v, hence computing

$$\frac{\partial F}{\partial v}(u,v) = \frac{-2u^2 + 2uv + v}{3(4u^2 - 4uv + v^2)}.$$

By evaluating $2u^2 - 2uv - v = 0$ at v and $v+1$, one checks that for fixed v, there is a unique $u_0 \in [v, v+1)$ such that $F(u,v)$ increases over $[v, u_0]$ and decreases over $[v_0, v+1)$. Hence, for $u \in [v, v+1)$,

$$\frac{-1}{6} = \min(F(v,v) - 2v/3, F(v+1,v) - 2(v+1)/3) \leqslant F(u,v) - 2u/3 \leqslant F(u_0,v) - 2u_0/3.$$

Finally, we find

$$F(u_0,v) - \frac{2u_0}{3} = (v - u_0)/3 \leqslant 0.$$

(note that the optimal bound for the latter is actually $(1-\sqrt{3})/6$, obtained for $v = 1$, $u_0 = (1+\sqrt{3})/2$).

Hence, we have

$$2\varphi^{-1}(x)/3 - 1/6 \leqslant \psi(x) \leqslant 2\varphi^{-1}(x)/3,$$

which, in view of $\sqrt{2x} - 1 < \varphi^{-1}(x) < \sqrt{2x}$, gives

$$-5/6 < \psi(x) - 2\sqrt{2x}/3 < 0.$$

□

Remark D.5. Simple numerical experiments suggest that actually $\psi(x) - 2\sqrt{2x}/3 \in [-1/2, -0.44]$ for $x \geqslant 1$, so that the asymptotic expansion $\psi(x) = 2\sqrt{2x}/3 - 1/2 + o(1)$, once truncated, actually gives an excellent approximation for all $x \geqslant 1$.

The following two lemmas yield useful information on the function ψ: invertibility, and inverse function.

Lemma D.6. *The function $x \mapsto \psi(x)$ is continuous and increasing over $[1, \infty)$.*

Proof. For the first part, since φ^{-1} is continuous and $x > 0$, it suffices to prove that $x \mapsto x(\psi \circ \varphi(x))$ is continuous, namely that $F : x \mapsto (1 + \lfloor x \rfloor)(6x^2 - \lfloor x \rfloor - 2\lfloor x \rfloor^2)$ is continuous.

Obviously, F is continuous on $[1, \infty) \backslash \mathbb{Z}_{>0}$. If n is an integer, we check that $F(n) = (1+n)(4n^2-n)$ whereas $\lim_{x \to n^-} F(x) = n(4n^2+3n-1) = n(n+1)(4n-1) = F(n)$.

To prove that $x \mapsto \psi(x)$ is increasing, as φ is increasing it suffices to study $x \mapsto \psi(\varphi(x))$. As the latter function is continuous over each interval $(\varphi^{-1}(n), \varphi^{-1}(n+1))$, it suffices to prove that it increases over each of those intervals.

Over such an interval, we have $\psi(\varphi(x)) = Ax^2/\varphi(x) - B/\varphi(x)$ for some nonnegative constants A, B; it thus suffices to prove that $x^2/\varphi(x)$ is increasing, or that $\varphi(x)/x^2$ is decreasing. As this function is continuous and has the form $A'(x+B')/x^2$ for some nonnegative A', B' over each interval $(n, n+1)$ for integer n, we see that it is indeed decreasing. □

Proposition D.7. *We have, for any $\mu \in [1, +\infty)$,*

$$\psi^{-1}(\mu) = \frac{k+1}{2}\left(2\mu - k + \sqrt{4\mu(\mu-k) + 2k(2k+1)/3}\right),$$

where $k = \lfloor 3\mu/2 + 1/4 \rfloor$. For $\mu \to \infty$, we have $\psi^{-1}(\mu) = 9\mu^2/8 + O(\mu)$.

Proof. We start by noticing that for all integer ℓ, $\psi(\ell(\ell+1)/2) = (4\ell-1)/6$, which follows easily from $\varphi^{-1}(\ell(\ell+1)/2) = \ell$.

Let now $k = \lfloor 3\mu/2 + 1/4 \rfloor$; then, $\psi(k(k+1)/2) \leqslant \mu < \psi((k+1)(k+2)/2)$, so that

$$k(k+1)/2 \leqslant \psi^{-1}(\mu) < (k+1)(k+2)/2,$$

and $\lfloor \varphi^{-1}(\psi^{-1}(\mu)) \rfloor = k$. The asymptotic expansion follows from these inequalities.

As a consequence, $\psi^{-1}(\mu) = \varphi(\varphi^{-1}(\psi^{-1}(\mu))) = (k+1)(\varphi^{-1}(\psi^{-1}(\mu)) - k/2)$, so that $\varphi^{-1}(\psi^{-1}(\mu)) = \frac{\psi^{-1}(\mu)}{k+1} + k/2$.

Hence,

$$\mu = \psi(\psi^{-1}(\mu)) = \frac{1+k}{12\psi^{-1}(\mu)}\left(6\varphi^{-1}(\psi^{-1}(\mu))^2 - k - 2k^2\right),$$

so that

$$\frac{12}{1+k}\mu\psi^{-1}(\mu) = 6\left(\frac{\psi^{-1}(\mu)}{k+1} + k/2\right)^2 - k - 2k^2.$$

Finally, $\psi^{-1}(\mu)/(k+1)$ is the positive root of the equation

$$X^2 + (k - 2\mu)X - \frac{k(k+2)}{12} = 0,$$

which gives the explicit form

$$\psi^{-1}(\mu) = \frac{k+1}{2}\left(2\mu - k + \sqrt{4\mu(\mu-k) + 2k(2k+1)/3}\right).$$

□

Appendix E. Proofs of Theorems 6.4 and 6.12

Proof of Theorem 6.4. For $0 \leqslant i,j \leqslant N-1$, $0 \leqslant k_1 \leqslant N_1-1$, $0 \leqslant k_2 \leqslant N_2-1$, we have from Proposition 4.2,

$$\left|(A_1)_{i,(k_1,k_2)}\right| \leqslant \left|\frac{c_{k_1,k_2,i}}{2^{\delta_{0k_1}+\delta_{0k_2}}}\right| \leqslant 4\frac{M_{\rho_1,a_1,b_1,\rho_2,a_2,b_2}(f_i)}{\rho_1^{k_1}\rho_2^{k_2}}\frac{\rho_1^2+1}{\rho_1^2-1}\frac{\rho_2^2+1}{\rho_2^2-1} \leqslant \frac{4\rho_1\rho_2 M_{\rho_1,a_1,b_1,\rho_2,a_2,b_2}(f_i)}{(\rho_1-1)(\rho_2-1)}\frac{1}{\rho_1^{k_1}\rho_2^{k_2}}.$$

Moreover, from the definition of A_2 and the assumption $\rho_1^{N_1} \leqslant \rho_2^{N_2}$, we know that, for $0 \leqslant i,j \leqslant N-1$,

$$|A_{2,i,j}| \leqslant \frac{16\rho_1\rho_2 M_{\rho_1,a_1,b_1,\rho_2,a_2,b_2}(f_i)}{(\rho_1-1)(\rho_2-1)}\left(\frac{2}{\rho_1^{N_1}}\right).$$

We now apply Theorem 5.1. We put $\mathfrak{r}_i = \frac{32\rho_1\rho_2 M_{\rho_1,a_1,b_1,\rho_2,a_2,b_2}(f_i)}{(\rho_1-1)(\rho_2-1)}$ for $i = 0,\ldots,N-1$. Then, notice that the product of the N largest elements among the $\mathfrak{c}_j$'s

$$1,\ldots,\frac{1}{\rho_1^{k_1}\rho_2^{k_2}},\ldots,\frac{1}{\rho_1^{N_1-1}\rho_2^{N_2-1}},\underbrace{\frac{1}{\rho_1^{N_1}},\ldots,\frac{1}{\rho_1^{N_1}}}_{N\text{times}}.$$

is upper bounded by $\rho_1^{-\Omega_\gamma(N_1,N_2,N)}$ thanks to the assumption $\rho_1^\gamma = \rho_2$, or equivalently, $\rho_1^{-i}\rho_2^{-j} = \rho_1^{-i-\gamma j}$.

We can now apply Theorem 5.1 to obtain the bound

$$(\det AA^t)^{1/2} \leqslant \left(32\sqrt{N}\right)^N 2^{\frac{N_1N_2+N}{2}}\left(\frac{\rho_1\rho_2}{(\rho_1-1)(\rho_2-1)}\right)^N \frac{\prod_{i=1}^N M_{\rho_1,a_1,b_1,\rho_2,a_2,b_2}(f_i)}{\rho_1^{\Omega_\gamma(N,N_1,N_2)}},$$

from which the Theorem follows, in view of the preliminary assumption that $N_1N_2 \geqslant N$. □

Proof of Theorem 6.12. For $j = 0,\ldots,N+N_1N_2-1$, we have $|(\Lambda A)[j] - (\Lambda\hat{A})[j]|_1 \leqslant \sum_{0\leqslant k+\ell\leqslant d}|\lambda_{k,\ell}|2^{-\mathsf{tprec}} \leqslant \sum_{0\leqslant k+\ell\leqslant d}|\lambda_{k,\ell}|\min_i \hat{A}_2[i,i]\frac{2^{-2}}{N}$, cf. proof of Lemma 6.10. As $\sum_{0\leqslant k+\ell\leqslant d}|\lambda_{k,\ell}|\min_i \hat{A}_2[i,i] \leqslant \|\Lambda\hat{A}_2\|_1$, we get $|(\Lambda A)[j] - (\Lambda\hat{A})[j]| \leqslant \frac{1}{4N}\|\Lambda\hat{A}_2\|_1$. Then, it comes $\|\Lambda A\|_1 \leqslant \|\Lambda\hat{A}\|_1 + (N+N_1N_2)\frac{\|\Lambda\hat{A}_2\|_1}{4N} \leqslant \|\Lambda\hat{A}\|_1 + (N+N_1N_2)\frac{\|\Lambda\hat{A}_2\|_2}{4\sqrt{N}}$ thanks to Cauchy-Schwarz inequality. Finally, we obtain $\|\Lambda A\|_1 \leqslant \|\Lambda\hat{A}\|_1 + \frac{1}{4N^{1/2}}$ from the assumption $\|\Lambda\hat{A}\|_2 \leqslant 1/(N+N_1N_2)$.

Let now P be as in the statement of the Theorem, and

$$Q(x,t) = \sum_{\substack{0\leqslant j_1\leqslant N_1-1\\ 0\leqslant j_2\leqslant N_2-1}} q_{j_1,j_2}T_{j_1,[a_1,b_1]}(x)T_{j_2,[a_2,b_2]}(t)$$

be the interpolation polynomial for $P(ux, v(f(x)+t))$ at the order (N_1,N_2) pairs of Chebyshev nodes of the first kind. Then, the coordinates of ΛA_1 are exactly q_{j_1,j_2}, $0 \leqslant j_1 \leqslant N_1-1, 0 \leqslant j_2 \leqslant N_2-1$.

Proposition 4.2 shows that

$$\max_{\substack{x\in[a_1,b_1]\\ t\in[a_2,b_2]}} |Q(x,t)-P(ux,v(f(x)+t))| \leqslant 16\frac{M_{\rho_1,a_1,b_1,\rho_2,a_2,b_2}(P)\rho_1\rho_2}{(\rho_1-1)(\rho_2-1)}\left(\frac{1}{\rho_1^{N_1}}+\frac{1}{\rho_2^{N_2}}\right)$$

$$\leqslant 16\sum_{0\leqslant k+\ell\leqslant d}|\lambda_{k,\ell}|\frac{u^k M_{\rho_1,a_1,b_1}(x)^k v^\ell M_{\rho_1,a_1,b_1,\rho_2,a_2,b_2}(f(x)+t)^\ell\rho_1\rho_2}{(\rho_1-1)(\rho_2-1)}\left(\frac{1}{\rho_1^{N_1}}+\frac{1}{\rho_2^{N_2}}\right)$$

hence

$$\max_{\substack{x\in[a_1,b_1]\\ t\in[a_2,b_2]}} |P(ux,v(f(x)+t))| \leqslant \max_{\substack{x\in[a_1,b_1]\\ t\in[a_2,b_2]}} |Q(x,t)|+$$

$$16\sum_{0\leqslant k+\ell\leqslant d}|\lambda_{k,\ell}|\frac{u^k M_{\rho_1,a_1,b_1}(x)^k v^\ell M_{\rho_1,a_1,b_1,\rho_2,a_2,b_2}(f(x)+t)^\ell\rho_1\rho_2}{(\rho_1-1)(\rho_2-1)}\left(\frac{1}{\rho_1^{N_1}}+\frac{1}{\rho_2^{N_2}}\right)$$

$$\leqslant \sum_{\substack{0\leqslant j_1\leqslant N_1-1\\ 0\leqslant j_2\leqslant N_2-1}}|q_{j_1,j_2}|+ \quad \text{(recall that } \max_{x\in[a_i,b_i]}|T_{k,[a_i,b_i]}(x)|=1 \text{ for all } k)$$

$$16\sum_{0\leqslant k+\ell\leqslant d}|\lambda_{k,\ell}|\frac{u^k M_{\rho_1,a_1,b_1}(x)^k v^\ell M_{\rho_1,a_1,b_1,\rho_2,a_2,b_2}(f(x)+t)^\ell\rho_1\rho_2}{(\rho_1-1)(\rho_2-1)}\left(\frac{1}{\rho_1^{N_1}}+\frac{1}{\rho_2^{N_2}}\right)$$

$$= \|\Lambda A\|_1 \leqslant \|\Lambda\hat{A}\|_1 + 1/(4N^{1/2})$$

$$\leqslant 1/(4N^{1/2}) + \sqrt{N+N_1N_2}\|\Lambda\hat{A}\|_2 \text{ thanks to Cauchy-Schwarz inequality}$$

$$\leqslant 1/(4N^{1/2}) + 1/\sqrt{N+N_1N_2} < 1 \text{ since } N\geqslant 3, N_1, N_2 \geqslant 2. \tag{E.1}$$

□.

Remark E.1. The proof should be slightly adapted if the two-variable analogous of Subsection 5.3.3 is used. Recall that $\hat{A}_{\mathsf{comp}} = 2^{-\mathsf{tprec}_{\mathsf{comp}}}(M_{c,\mathsf{comp}}\, M_{r,\mathsf{comp}})$, we obtain for $j = 0,\ldots,N+N_1N_2-1$,

$$|(\Lambda A)[j]-(\Lambda\hat{A}_{\mathsf{comp}})[j]| \leqslant \sum_{0\leqslant k+\ell\leqslant d}|\lambda_{k,\ell}|2^{1-\mathsf{tprec}_{\mathsf{comp}}} \leqslant \sum_{0\leqslant k+\ell\leqslant d}|\lambda_{k,\ell}|\min_i \hat{A}_2[i,i]\frac{1}{2N}$$

from which follows $\|\Lambda A\|_1 \leqslant \|\Lambda\hat{A}\|_1 + (N+N_1N_2)\frac{\|\Lambda\hat{A}_2\|_1}{2N} \leqslant \|\Lambda\hat{A}\|_1 + \frac{1}{2N^{1/2}}$. The upper bound in Inequality (E.1) becomes $1/(2N^{1/2}) + 1/\sqrt{N+N_1N_2} < 1$ since $N\geqslant 3, N_1, N_2 \geqslant 2$.

Note also that the success condition (6.10) becomes

$$\max_{i=0,1}\left(\|(M_{LLL}[i,j])_{0\leqslant j\leqslant N+N_1N_2-1}\|_1 + (N+N_1N_2)\frac{\|(M_{LLL}[i,j])_{N_1N_2\leqslant j\leqslant N+N_1N_2-1}\|_1}{2N}\right) < 2^{\mathsf{tprec}}.$$

Université de Lyon, CNRS, ENS de Lyon, Inria, Université Claude-Bernard Lyon 1, Laboratoire LIP (UMR 5668), Lyon, France.
Email address: Nicolas.Brisebarre@ens-lyon.fr

Université de Lyon, CNRS, ENS de Lyon, Inria, Université Claude-Bernard Lyon 1, Laboratoire LIP (UMR 5668), Lyon, France.
Email address: Guillaume.Hanrot@ens-lyon.fr