Static bounds for straight-line programs*

JORIS VAN DER HOEVEN[a], GRÉGOIRE LECERF[b], ARNAUD MINONDO[c]

Laboratoire d'informatique de l'École polytechnique (LIX, UMR 7161 CNRS)
CNRS, École polytechnique, Institut Polytechnique de Paris
Bâtiment Alan Turing, CS35003
1, rue Honoré d'Estienne d'Orves
91120 Palaiseau, France

a. *Email:* vdhoeven@lix.polytechnique.fr
b. *Email:* lecerf@lix.polytechnique.fr
c. *Email:* minondo@lix.polytechnique.fr

Preliminary version of June 18, 2025

How to automatically determine reliable error bounds for a numerical computation? One traditional approach is to systematically replace floating point approximations by intervals or balls that are guaranteed to contain the exact numbers one is interested in. However, operations on intervals or balls are more expensive than operations on floating point numbers, so this approach involves a non-trivial overhead.

In this paper, we present several approaches to remove this overhead, under the assumption that the function f that we wish to evaluate is given as a straight-line program (SLP). We will first study the case when the arguments of our function lie in fixed balls. For polynomial SLPs, we next consider the "global" case where this restriction on the arguments is removed. We will also investigate the computation of bounds for first and higher order derivatives of f.

KEYWORDS: straight-line program, ball arithmetic, error bound, reliable computing

1. INTRODUCTION

Interval arithmetic is a popular technique to calculate guaranteed error bounds for approximate results of numerical computations [2, 13, 16–21]. The idea is to systematically replace floating point approximations by small intervals around the exact numbers that we are interested in. Basic arithmetic operations on floating point numbers are replaced accordingly with the corresponding operations on intervals. When computing with complex numbers or when working with multiple precision, it is more convenient to use balls instead of intervals. In this paper, we will always do so and this variant of interval arithmetic is called *ball arithmetic* [8, 14].

Unfortunately, ball arithmetic suffers from a non-trivial overhead: floating point balls take twice the space of floating point numbers and basic arithmetic operations are between two and approximately ten times more expensive. For certain applications, it may therefore be preferable to avoid the systematic use of balls for individual operations. Instead,

*. Grégoire Lecerf and Arnaud Minondo have been supported by the French ANR-22-CE48-0016 NODE project. Joris van der Hoeven has been supported by an ERC-2023-ADG grant for the ODELIX project (number 101142171).
Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

one may analyze the error for larger groups of operations. For instance, when naively multiplying two double precision $n \times n$ matrices whose coefficients are all bounded by 1 in norm (say), then it is guaranteed that the maximum error of an entry of the result is bounded by $n^2 2^{-51}$, without having to do any individual operations on balls.

The goal of this paper is to compute reliable error bounds in a systematic fashion, while avoiding the overhead of ball arithmetic. We will focus on the case when the function f that we wish to evaluate is given by a *straight-line program* (SLP). Such a program is essentially a sequence of basic arithmetic instructions like additions, subtractions, multiplications, and possibly divisions [5]. For instance, $n \times n$ matrix multiplication can be computed using an SLP. The SLP framework is actually surprisingly general: at least conceptually, the trace of the execution of a more general program that involves loops or subroutines can often be regarded as an SLP [11, section 3].

So consider a function $f: \mathbb{K}^m \to \mathbb{K}^n$ that can be computed using an SLP, where $\mathbb{K} = \mathbb{R}$ or $\mathbb{R} = \mathbb{C}$. Given $a \in \mathbb{K}$ and $r \in \mathbb{R}^{\geqslant}$, let $\mathcal{B}(a, r)$ denote the ball with center a and radius r. We will consider several problems:

Q1. Given an approximate evaluation $b = (b_1, \ldots, b_n) \approx f(a_1, \ldots, a_m) = f(a)$ using floating point arithmetic, can we efficiently compute a bound for the error?

Q2. Given $b \approx f(a)$ and $r_1, \ldots, r_m \in \mathbb{R}^{\geqslant}$, can we efficiently compute $s_1, \ldots, s_n \in \mathbb{R}^{\geqslant}$ such that $f(\mathcal{B}(x_1, r_1) \times \cdots \times \mathcal{B}(x_m, r_m)) \subseteq \mathcal{B}(b_1, s_1) \times \cdots \times \mathcal{B}(b_n, s_n)$?

Q3. Given balls $\mathcal{B}(a_1, r_1), \ldots, \mathcal{B}(a_m, r_m)$, an index $i \in \{1, \ldots, n\}$, and $k_1, \ldots, k_\ell \in \{1, \ldots, m\}$, can we efficiently compute a bound for $\left| \frac{\partial^r f_i}{\partial a_{k_1} \cdots \partial a_{k_\ell}} \right|$ on $\mathcal{B}(a_1, r_1) \times \cdots \times \mathcal{B}(a_m, r_m)$?

Here "efficiently" means that the cost of the bound computation should not exceed $O(m + n)$, ideally speaking. In particular, the cost should not depend on the length of the SLP, but we do allow ourselves to perform precomputations with such a higher cost. We may regard **Q1** as a special case of **Q2** by taking $r_1 = \cdots = r_m = 0$. If $\ell = 0$, then **Q3** becomes essentially a special case of **Q2**, since we may use $|b_i| + s_i$ as the required bound.

Of course, there is a trade-off between the cost of bound computations and the sharpness of the obtained bounds. We will allow our bounds to be less sharp than those obtained using traditional ball arithmetic. But we still wish them to be as tight as possible under the constraint that the cost of the bound computations should remain small.

We start with the special case when the centers $a_1, \ldots, a_m$ lie in some fixed balls $B_1, \ldots, B_m$. For this purpose we introduce a special variant of ball arithmetic, called *matryoshka arithmetic*: see section 2. A matryoshka is a ball whose center is itself a ball and its radius is a number in $\mathbb{R}^{\geqslant}$. Intuitively speaking, a matryoshka allows us to compute enclosures for "balls inside balls". By evaluating f at matryoshki with centers $B_1, \ldots, B_m$ and zero radii, we will be able to answer question **Q1**: see section 3. Using this to compute bounds for the gradient of f on $B_1 \times \cdots \times B_m$, we will also be able to answer question **Q2** in the case when $\mathcal{B}(a_1, r_1) \subseteq B_1, \ldots, \mathcal{B}(a_m, r_m) \subseteq B_m$.

In section 3, we will actually describe a particularly efficient way to evaluate SLPs at matryoshki. For this, we will adapt transient ball arithmetic from [10]. This variant of ball arithmetic has the advantage that, during the computations of error bounds for individual operations, no adjustments are necessary to take rounding errors into account. We originally developed this technique for SLPs that only use ring operations. In section 4, we will extend it to SLPs that may also involve divisions.

For polynomial SLPs that do not involve any divisions, we will show in section 5 that it is actually possible to release the condition that $a_1, \ldots, a_m$ must be contained in fixed

balls $B_1,\ldots,B_m$. The idea is to first reduce to the case when the components $f_i\colon \mathbb{K}^n \to \mathbb{K}$ with $i=1,\ldots,n$ are homogeneous. Then $f_i(\lambda a)=\lambda^{d_i} f(a)$ for some d_i, so we may always rescale a such that it fits into the unit poly-ball $\mathcal{B}(0,1)^m$. We may then apply the theory from sections 2 and 3. Reliable numeric homotopy continuation [4, 6, 7, 9, 15] is a typical application for which it is important to efficiently evaluate polynomial SLPs at arbitrary balls.

Our final section 6 is devoted to question **Q3**. The main idea is to compute bounds for the $|f_i|$ on balls $\mathcal{B}(a_1, r_1+\delta_1),\ldots,\mathcal{B}(a_m, r_m+\delta_m)$ with the same centers but larger radii. We will then use Cauchy's formula to obtain bounds for the derivatives of f without having to explicitly compute these derivatives. Bounds for the derivatives of f are in particular useful when developing reliable counterparts of Runge–Kutta methods for the integration of systems of ordinary differential equations. We intend to provide more details about this application in an upcoming paper.

2. DIFFERENT TYPES OF BALL ARITHMETIC

2.1. IEEE floating point arithmetic and notation

Throughout this paper, we assume that we work with a fixed floating point format that conforms to the IEEE 754 standard. We write p for the bit precision, i.e. the number of fractional bits of the mantissa plus one. We also denote the minimal and maximal allowed exponents by $E_{\min}$ and $E_{\max}$. For IEEE 754 double precision numbers, this means that $p=53$, $E_{\min}=-1022$ and $E_{\max}=1023$. We denote the set of hardware floating point numbers by $\mathbb{R}_p$. Given an $\mathbb{R}$-algebra $\mathbb{A}$, we will also denote the corresponding approximate version by $\mathbb{A}_p$. For instance, if $\mathbb{A}=\mathbb{C}=\mathbb{R}[\mathrm{i}]$, then we have $\mathbb{A}_p=\mathbb{R}_p[\mathrm{i}]$.

The IEEE 754 standard imposes correct rounding of all basic arithmetic operations. In this paper we will systematically use the *rounding to nearest* mode. We denote by $x_\circ$ the result of rounding $x\in\mathbb{R}$ according to this mode. The quantity $\varepsilon_\circ(x) := |x_\circ - x|$ stands for the corresponding rounding error, which may be $+\infty$. Given a single operation $*\in\{+,-,\cdot,\ldots\}$, it will be convenient to write $x *_\circ y$ for $(x*y)_\circ$. For compound expressions φ, we will also write $\circ[\varphi]$ for the full evaluation of φ using the rounding mode $\circ$. For instance, $\circ[xy+a^2b] = x_\circ \cdot_\circ y_\circ +_\circ (a_\circ \cdot_\circ a_\circ)\cdot_\circ b_\circ$.

We denote by $\bar\varepsilon_\circ$ any upper bound function for $\varepsilon_\circ$ that is easy to compute. In absence of underflow, one may take $\bar\varepsilon_\circ(x) = |x_\circ|\,2^{-p}$. If we want to allow for underflows during computations, then we can take $\bar\varepsilon_\circ(x) = |x_\circ|\,2^{-p} + 2^{E_{\min}-p+1}$ instead, where $2^{E_{\min}-p+1}$ is the smallest positive subnormal number in $\mathbb{R}_p$. If $x,y\in\mathbb{R}_p$, then we may still take $\bar\varepsilon_\circ(x\pm y) = |x\pm_\circ y|\,\epsilon_\circ$ since no underflow occurs in that special case. See [10, section 2.1] for more details about these facts.

For bounds on rounding errors, the following lemma will be useful.

LEMMA 2.1. *Let $q\in\mathbb{R}$ and $\epsilon>0$ be such that $q^2\leqslant\epsilon^{-1}$. Then $(1+\epsilon)^q\leqslant 1+(q+1)\,\epsilon$.*

Proof. Let $\binom{q}{k} := \prod_{i=0}^{q-1}\frac{(q-i)}{i+1}$ for all $k\in\mathbb{N}$. Since $\epsilon<1$, we have

$$\begin{aligned}(1+\epsilon)^q &= 1+q\epsilon+\frac{q\,(q-1)}{2}\epsilon^2+\sum_{k\geqslant 3}\binom{q}{k}\epsilon^k\\ &\leqslant 1+q\epsilon+\frac{q^2\epsilon^2}{2}+\frac{|q|^3\epsilon^3}{6}\mathrm{e}^{|q|\epsilon}\\ &\leqslant 1+q\epsilon+\left(\frac{1}{2}+\frac{\mathrm{e}}{6}\right)\epsilon\\ &\leqslant 1+(q+1)\,\epsilon.\end{aligned}$$

□

2.2. Ball arithmetic

Let $\mathbb{A}$ be an $\mathbb{R}$-algebra and let $|\cdot|$ be a norm on $\mathbb{A}$. We will typically take $\mathbb{A}=\mathbb{R}$ or $\mathbb{A}=\mathbb{C}$, but more general normed algebras are also allowed. Given $c\in\mathbb{A}$ and $r\in\mathbb{R}$, let $\mathcal{B}(c,r) := \{z\in\mathbb{A}, |z-c|\leqslant r\}$ be the closed ball with center c and radius r. We denote by $\mathcal{B}(\mathbb{A},\mathbb{R})$ the set of all such balls.

The aim of ball arithmetic is to provide a systematic way to bound the errors of numerical computations. The idea is to systematically replace numerical approximations of elements in $\mathbb{A}$ by balls that are guaranteed to contain the true mathematical values. It will be useful to introduce a separate notation $\circ\!\!-$ for this type of semantics: given a ball $\boldsymbol{x}\in\mathcal{B}(\mathbb{A},\mathbb{R})$ and a number, we say that $\boldsymbol{x}$ *encloses* x if $\boldsymbol{x}\ni x$, i.e.

$$\boldsymbol{x} \circ\!\!- x \quad\Longleftrightarrow\quad x\in\boldsymbol{x}.$$

We also introduce poly-balls, which are vectors of balls, for situations where it is required to reason "coordinate wise". For all $a := (a_1,\ldots,a_m)\in\mathbb{A}^m$ and all $r := (r_1,\ldots,r_m)\in\mathbb{R}^m$, we denote the poly-ball $\mathcal{B}(a,r) := (\mathcal{B}(a_1,r_1),\ldots,\mathcal{B}(a_m,r_m))\in\mathcal{B}(\mathbb{A},\mathbb{R})^m$. We extend this enclosure relation to poly-balls $\boldsymbol{x}\in\mathcal{B}(\mathbb{A},\mathbb{R})^m$ and $x\in\mathbb{R}^m$ as follows:

$$\boldsymbol{x} \circ\!\!- x \quad\Longleftrightarrow\quad \boldsymbol{x}_1 \circ\!\!- x_1 \wedge \cdots \wedge \boldsymbol{x}_m \circ\!\!- x_m. \tag{2.1}$$

In the sequel, we will use a bold font for ball enclosures and a normal font for values at actual points. Note that the smallest enclosure of $x\in\mathbb{A}$ is the "exact" ball $\mathcal{B}(x,0)$, and we will regard $\mathbb{A}$ as being embedded into $\mathcal{B}(\mathbb{A},\mathbb{R})$ in this way.

Let us now recall how to perform arithmetic operations in a way that is compatible with the "enclosure semantics". Given a function $f:\mathbb{A}^m\mapsto\mathbb{A}^n$, a *ball lift* of f is a function $f:\mathcal{B}(\mathbb{A},\mathbb{R})^m\mapsto\mathcal{B}(\mathbb{A},\mathbb{R})^n$ that satisfies the *inclusion property*

$$\boldsymbol{x} \circ\!\!- x \quad\Longrightarrow f(\boldsymbol{x}) \circ\!\!- f(x)$$

for all $\boldsymbol{x}\in\mathcal{B}(\mathbb{A},\mathbb{R})^m$ and $x\in\mathbb{A}^m$. (Note that we voluntarily used the same name f for the function and its lift.) For instance, the basic arithmetic operations admit the following ball lifts:

$$\mathcal{B}(a,r)\pm\mathcal{B}(b,s) := \mathcal{B}(a\pm b, r+s) \tag{2.2}$$

$$\mathcal{B}(a,r)\cdot\mathcal{B}(b,s) := \mathcal{B}(a\,b, (|a|+r)\,s+|b|\,r). \tag{2.3}$$

Other operations can be lifted in a similar way (in section 4 below, we will in particular study division). We can also extend the norm from $\mathbb{A}$ to $\mathcal{B}(\mathbb{A},\mathbb{R})$ via

$$\begin{aligned} |\cdot|:\mathcal{B}(\mathbb{A},\mathbb{R}) &\longrightarrow \mathbb{R}\\ \mathcal{B}(c,r) &\longmapsto |c|+r. \end{aligned} \tag{2.4}$$

The extended norm is sub-additive, sub-multiplicative, and positive definite.

2.3. Matryoshka arithmetic

For any ball $\boldsymbol{x}=\mathcal{B}(c,r)$, we have $\boldsymbol{x}+(-\boldsymbol{x})=\mathcal{B}(c,r)+\mathcal{B}(-c,r)=\mathcal{B}(0,2r)\neq 0$, so $\mathcal{B}(\mathbb{A},\mathbb{R})$ is clearly not an additive group in the mathematical sense. Nonetheless, we may consider $\mathcal{B}(\mathbb{A},\mathbb{R})$ to be a normed $\mathbb{R}$-algebra from a computer science perspective, because all the relevant operations $+,-,\cdot$, and $|\cdot|$ are "implemented". Allowing ourselves this shortcut, we may apply the theory from the previous subsection and formally obtain a ball arithmetic on $\mathcal{B}(\mathcal{B}(\mathbb{A},\mathbb{R}),\mathbb{R})$. It turns out that this construction actually makes sense from a mathematical perspective, provided that we appropriately adapt the semantics for the notion of "enclosure".

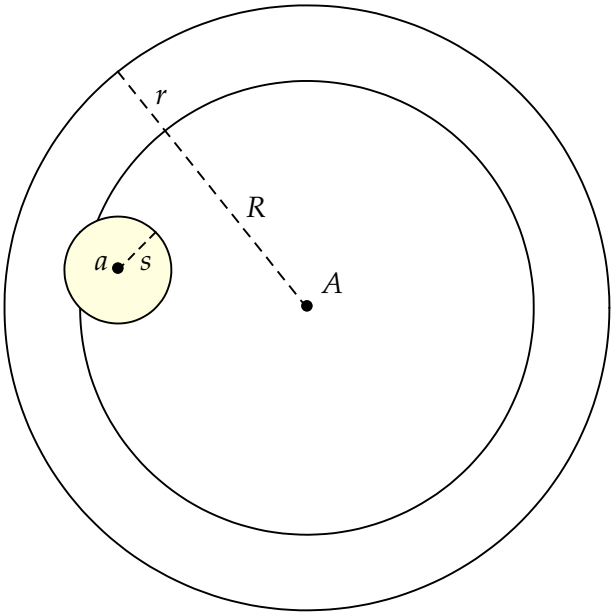


Figure 2.1. Representation of an embedded ball inside a matryoshka.

Let $\mathcal{B}(\mathbb{A},\mathbb{R},\mathbb{R}) := \mathcal{B}(\mathcal{B}(\mathbb{A},\mathbb{R}),\mathbb{R})$. An element $\mathcal{B}(A,R,r) := \mathcal{B}(\mathcal{B}(A,R),r)$ of $\mathcal{B}(\mathbb{A},\mathbb{R},\mathbb{R})$ will be called a *matryoshka*. Given a matryoshka $\boldsymbol{A} = \mathcal{B}(A,R,r)$, we call A its *center*, R its *large radius*, and r its *small radius*. The appropriate enclosure relation for matryoshki is defined as follows: given a matryoshka $\boldsymbol{A} = \mathcal{B}(A,R,r)$ and a ball $\boldsymbol{a} = \mathcal{B}(a,s)$, we define

$$\boldsymbol{A} \circ\!\!- \boldsymbol{a} \quad\Longleftrightarrow\quad \mathcal{B}(A,R) \circ\!\!- a \text{ and } s \leqslant r.$$

Conceptually, the "small embedded ball" $\boldsymbol{a}$ is contained in the matryoshka, which corresponds to the "big ball" $\mathcal{B}(A,R+r)$: see Figure 2.1. The enclosure relation naturally extends to vectors as in (2.1).

We use the abbreviation $\mathcal{B}(\mathbb{A},\mathbb{R},\mathbb{R}) := \mathcal{B}(\mathcal{B}(\mathbb{A},\mathbb{R}),\mathbb{R})$ for the set of matryoshki. A function $f : \mathcal{B}(\mathbb{A},\mathbb{R})^m \to \mathcal{B}(\mathbb{A},\mathbb{R})^n$ is said to *lift* to $f : \mathcal{B}(\mathbb{A},\mathbb{R},\mathbb{R})^m \to \mathcal{B}(\mathbb{A},\mathbb{R},\mathbb{R})^n$ if it satisfies the inclusion principle:

$$\boldsymbol{A} \circ\!\!- \boldsymbol{a} \quad\Longrightarrow\quad f(\boldsymbol{A}) \circ\!\!- f(\boldsymbol{a})$$

for all $\boldsymbol{A} \in \mathcal{B}(\mathbb{A},\mathbb{R},\mathbb{R})^m$ and $\boldsymbol{a} \in \mathcal{B}(\mathbb{A},\mathbb{R})^n$. Exactly the same formulas (2.2) and (2.3) can be used in order to lift the basic arithmetic operations:

$$\begin{aligned} \mathcal{B}(\boldsymbol{A},r) \pm \mathcal{B}(\boldsymbol{B},s) &:= \mathcal{B}(\boldsymbol{A} \pm \boldsymbol{B}, r+s) \\ \mathcal{B}(\boldsymbol{A},r) \cdot \mathcal{B}(\boldsymbol{B},s) &:= \mathcal{B}(\boldsymbol{A} \cdot \boldsymbol{B}, (|\boldsymbol{A}|+r)\, s + |\boldsymbol{B}|\, r), \end{aligned}$$

for all $\mathcal{B}(\boldsymbol{A},r), \mathcal{B}(\boldsymbol{B},s) \in \mathcal{B}(\mathbb{A},\mathbb{R},\mathbb{R})$. (Note that it was this time convenient to use $\boldsymbol{A}, \boldsymbol{B}$ as a notation for the centers of our matryoshki.) We may also extend the norm on $\mathcal{B}(\mathbb{A},\mathbb{R})$ to matryoshki via

$$\begin{aligned} |\cdot| : \mathcal{B}(\mathbb{A},\mathbb{R},\mathbb{R}) &\longrightarrow \mathbb{R} \\ \mathcal{B}(\boldsymbol{A},r) &\longmapsto |\boldsymbol{A}| + r \end{aligned}$$

which again remains sub-additive, sub-multiplicative, and positive definite.

Remark 2.2. In principle, we could repeat the process and recursively consider matryoshki as centers of even larger matryoshki. While attractive from a folkloric perspective, it turns out that the basic matryoshki are more efficient for the applications we know of.

2.4. Rounded and transient ball arithmetic

We will write $\mathcal{B}(\mathbb{A}_p, \mathbb{R}_p)$ and $\mathcal{B}(\mathbb{A}_p, \mathbb{R}_p, \mathbb{R}_p)$ for the approximate versions of $\mathcal{B}(\mathbb{A}, \mathbb{R})$ and $\mathcal{B}(\mathbb{A}, \mathbb{R}, \mathbb{R})$ when working with machine floating point numbers in $\mathbb{R}_p$ instead of exact real numbers in $\mathbb{R}$. In that case, the formulas from section 2.2 and 2.3 need to be adjusted in order to take into account rounding errors. For instance, if $\mathbb{A} = \mathbb{R}$, then one may replace the formulas (2.2), (2.3), and (2.4) by

$$\begin{aligned} \mathcal{B}(a,r) \pm_\circ \mathcal{B}(b,s) &:= \mathcal{B}(a \pm_\circ b, \circ[(r+s+\bar{\varepsilon}_\circ(a \pm b))\,(1+4\cdot 2^{-p})]) \\ \mathcal{B}(a,r) \cdot_\circ \mathcal{B}(b,s) &:= \mathcal{B}(a \cdot_\circ b, \circ[((|a|+r)\,s+|b|\,r+\bar{\varepsilon}_\circ(a\cdot b))\,(1+6\cdot 2^{-p})]) \\ |\mathcal{B}(a,r)|_\circ &:= \circ[(|a|+r)\,(1+2\cdot 2^{-p})]. \end{aligned} \tag{2.5}$$

See, e.g., [10]. We will call this *rounded ball arithmetic*.

Unfortunately, these formulas are far more complicated than (2.2), (2.3), and (2.4), so bounding the rounding errors in this way gives rise to a significant additional computational overhead. An alternative approach is to continue to use the non-adjusted formulas

$$\begin{aligned} \mathcal{B}(a,r) \pm_\circ \mathcal{B}(b,s) &:= \mathcal{B}(a \pm_\circ b, \circ[r+s]) \\ \mathcal{B}(a,r) \cdot_\circ \mathcal{B}(b,s) &:= \mathcal{B}(a \cdot_\circ b, \circ[(|a|+r)\,s+|b|\,r]) \\ |\mathcal{B}(a,r)|_\circ &:= \circ[|a|+r]. \end{aligned} \tag{2.6}$$

This type of arithmetic was called *transient ball arithmetic* in [10]. This arithmetic is not a ball lift, but we will see below how to take advantage of it by inflating the radii of the input balls of a given SLP.

Of course, we need to carefully examine the amount of inflation that is required to ensure the correctness of our final bounds. This will be the purpose of the next section in the case where we evaluate an entire SLP using transient ball arithmetic. We denote by $\epsilon_{\mathbb{A}_p} \in \mathbb{R}_p \cap (\mathbb{N}\,2^{-p})$ a quantity that satisfies $\epsilon_{\mathbb{A}_p} \leqslant 1/16$ and

$$\begin{aligned} |a *_\circ b - a * b| &\leqslant |a *_\circ b|\,\epsilon_{\mathbb{A}_p} \\ ||a|_\circ - |a|| &\leqslant |a|_\circ\,\epsilon_{\mathbb{A}_p}, \end{aligned} \tag{2.7}$$

for all $a, b \in \mathbb{A}_p$ and $* \in \{+, -, \cdot\}$, in absence of underflows and overflows. One may for instance take $\epsilon_{\mathbb{R}_p} := 2^{-p}$ and $\epsilon_{\mathbb{C}_p} := 4 \cdot 2^{-p}$, whenever $p \geqslant 16$; see [10, Appendix A of the preprint version].

We recall the following lemma, which provides a useful error estimate for the radius of a transient ball product.

LEMMA 2.3. [10, Lemma 3] *For all $a, b \in \mathbb{A}_p$ and $r, s \in \mathbb{R}_p$ such that the computation of*

$$R = \circ[(|a|+r)\,s+|b|\,r]$$

involves no underflows or overflows, we have $(|a|+r)\,s+|b|\,r \leqslant R\,(1+\epsilon_{\mathbb{A}_p})^4$.

In the presence of overflows or underflows, additional adjustments are required. Overflows actually cause no problems, because the IEEE 754 standard rounds every number beyond the largest representable floating point number to infinity, so the radii of balls automatically become infinite whenever an overflow occurs. In order to protect ourselves against underflows, the requirements (2.7) need to be changed into

$$\begin{aligned} |(a \pm_\circ b) - (a \pm b)| &\leqslant |a \pm_\circ b|\, \epsilon_{\mathbb{A}_p} \\ |a \cdot_\circ b - a \cdot b| &\leqslant |a \cdot_\circ b|\, \epsilon_{\mathbb{A}_p} + \eta_{\mathbb{A}_p} \\ ||a|_\circ - |a|| &\leqslant |a|_\circ\, \epsilon_{\mathbb{A}_p} + \eta_{\mathbb{A}_p}. \end{aligned} \tag{2.8}$$

Here we recall that additions and subtractions never lead to underflows. If $\mathbb{A}_p = \mathbb{R}_p$, then we may take $\eta_{\mathbb{R}_p} := 2^{E_{\min}+1}$. If $\mathbb{A}_p = \mathbb{C}_p$, then a safe choice is $\eta_{\mathbb{C}_p} := 8 \cdot 2^{(E_{\min}+1)/2}$ (in fact, $\eta_{\mathbb{C}_p} := 5 \cdot 2^{E_{\min}+1}$ is sufficient for multiplication). We refer to [10, section 3.3] for details.

2.5. Transient matryoshka arithmetic

Taking $\mathcal{B}(\mathbb{A}_p, \mathbb{R}_p)$ instead of $\mathbb{A}_p$ for our center space, the formulas (2.6) naturally give rise to transient matryoshka arithmetic:

$$\begin{aligned} \mathcal{B}(\boldsymbol{A}, r) \pm_\circ \mathcal{B}(\boldsymbol{B}, s) &:= \mathcal{B}(\boldsymbol{A} \pm_\circ \boldsymbol{B}, \circ[r+s]) \\ \mathcal{B}(\boldsymbol{A}, r) \cdot_\circ \mathcal{B}(\boldsymbol{B}, s) &:= \mathcal{B}(\boldsymbol{A} \cdot_\circ \boldsymbol{B}, \circ[(|\boldsymbol{A}| + r)\, s + |\boldsymbol{B}|\, r]) \\ |\mathcal{B}(\boldsymbol{A}, r)|_\circ &:= \circ[|\boldsymbol{A}| + r]. \end{aligned}$$

Here we understand the operations $\pm_\circ$, $\cdot_\circ$, and $|\cdot|_\circ$ on the centers are done using transient ball arithmetic.

We will need an analogue of Lemma 2.3 for matryoshki. Given balls $\boldsymbol{a} = \mathcal{B}(a, r)$ and $\boldsymbol{b} = \mathcal{B}(b, s)$ in $\mathcal{B}(\mathbb{A}, \mathbb{R})$, we define $\boldsymbol{a} -_{\mathrm{vec}} \boldsymbol{b} := \mathcal{B}(\boldsymbol{a} - \boldsymbol{b}, |r - s|)$. In what follows, let $\epsilon_{\mathcal{B}(\mathbb{A}_p, \mathbb{R}_p)} \in \mathbb{R}_p \cap (\mathbb{N}\, 2^{-p})$ be such that $\epsilon_{\mathcal{B}(\mathbb{A}_p, \mathbb{R}_p)} \leqslant 1/16$ and

$$\begin{aligned} |\boldsymbol{a} *_\circ \boldsymbol{b} -_{\mathrm{vec}} \boldsymbol{a} * \boldsymbol{b}| &\leqslant |\boldsymbol{a} *_\circ \boldsymbol{b}|\, \epsilon_{\mathcal{B}(\mathbb{A}_p, \mathbb{R}_p)} \\ ||\boldsymbol{a}|_\circ - |\boldsymbol{a}|| &\leqslant |\boldsymbol{a}|_\circ\, \epsilon_{\mathcal{B}(\mathbb{A}_p, \mathbb{R}_p)}, \end{aligned} \tag{2.9}$$

for all $\boldsymbol{a}, \boldsymbol{b} \in \mathcal{B}(\mathbb{A}_p, \mathbb{R}_p)$ and $* \in \{+, -, \cdot\}$, in absence of underflows and overflows.

Let $\epsilon := \max(\epsilon_{\mathbb{A}_p}, \epsilon_{\mathbb{R}_p})$. Given $\boldsymbol{a} := \mathcal{B}(a, r) \in \mathcal{B}(\mathbb{A}_p, \mathbb{R}_p)$ and $\boldsymbol{b} := \mathcal{B}(b, s) \in \mathcal{B}(\mathbb{A}_p, \mathbb{R}_p)$, Lemma 2.3 gives $(|a| + r)\, s + |b|\, r \leqslant \circ[(|a| + r)\, s + |b|\, r]\, (1 + \epsilon)^4$; similarly we have

$$\circ[(|a| + r)\, s + |b|\, r]\, (1 - \epsilon)^4 \leqslant (|a| + r)\, s + |b|\, r.$$

This implies

$$\begin{aligned} |\boldsymbol{a} \cdot_\circ \boldsymbol{b} -_{\mathrm{vec}} \boldsymbol{a} \cdot \boldsymbol{b}| &\leqslant |\boldsymbol{a} \cdot_\circ \boldsymbol{b}| \max((1+\epsilon)^4 - 1, 1 - (1-\epsilon)^4) \\ &\leqslant |\boldsymbol{a} \cdot_\circ \boldsymbol{b}|\, (5\epsilon). \end{aligned}$$

It thus suffices to take $\epsilon_{\mathcal{B}(\mathbb{A}_p, \mathbb{R}_p)} := 5 \max(\epsilon_{\mathbb{A}_p}, \epsilon_{\mathbb{R}_p})$ for (2.9) to hold in the case of multiplication. Simpler similar computations show that this is also sufficient for the other operations. We can now state the analogue of Lemma 2.3.

LEMMA 2.4. *For all $\boldsymbol{a}, \boldsymbol{b} \in \mathcal{B}(\mathbb{A}_p, \mathbb{R}_p)$ and $r, s \in \mathbb{R}_p$ such that the computation of*

$$R = \circ[(|\boldsymbol{a}| + r)\, s + |\boldsymbol{b}|\, r]$$

involves no underflows or overflows, we have $(|\boldsymbol{a}| + r)\, s + |\boldsymbol{b}|\, r \leqslant R\, (1 + \epsilon_{\mathcal{B}(\mathbb{A}_p, \mathbb{R}_p)})^4$.

Proof. The same as the proof of [10, Lemma 1], *mutatis mutandis*. □

3. EVALUATING STRAIGHT LINE PROGRAMS

3.1. Straight-line programs

A *straight-line program* Γ over a ring $\mathbb{A}$ is a sequence $\Gamma_1, \ldots, \Gamma_l$ of instructions of the form

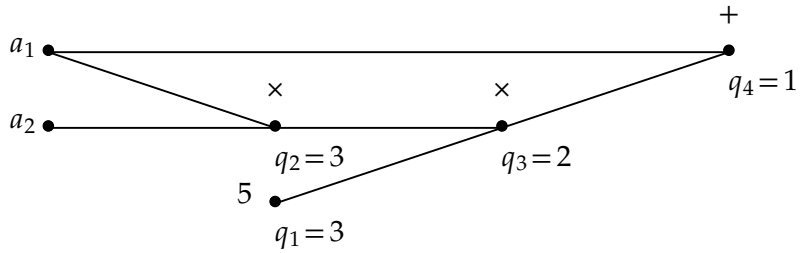
$$\begin{aligned} \Gamma_k &\equiv X_k := C_k \qquad\qquad \text{or} \\ \Gamma_k &\equiv X_k := Y_k * Z_k, \end{aligned}$$

where X_k, Y_k, Z_k are variables in a finite ordered set $\mathcal{V}$, C_k constants in $\mathbb{A}$, and $* \in \{+, -, \cdot\}$. Variables that appear for the first time in the sequence in the right-hand side of an instruction are called *input variables*. A distinguished subset of the set of all variables occurring at the left-hand side of instructions is called the set of *output variables*. Variables which are not used for input or output are called *temporary variables* and determine the amount of auxiliary memory needed to evaluate the program. The length $l_\Gamma = l$ of the sequence is called the *length* of Γ.

Let $I_1, \ldots, I_m$ be the input variables of Γ and $O_1, \ldots, O_n$ the output variables, listed in increasing order. Then we associate an *evaluation function* $E_\Gamma \colon \mathbb{A}^m \to \mathbb{A}^n$ to Γ as follows: given $(a_1, \ldots, a_m) \in \mathbb{A}^m$, we assign a_i to I_i for $i = 1, \ldots, m$, then evaluate the instructions of Γ in sequence, and finally read off the values of $O_1, \ldots, O_n$, which determine $E_\Gamma(a_1, \ldots, a_m)$.

To each instruction Γ_k, one may associate the *remaining path lengths* q_k as follows. Let $q_l = 1$, and assume that $q_{k+1}, \ldots, q_l$ have been defined for some $k \in \{1, \ldots, l\}$. Then we take $q_k = \max(q_{i_1}, \ldots, q_{i_n}) + 1$, where $i_1 > \cdots > i_n$ are those indices $i > k$ such that Γ_i is of the form $X_i := Y_i * Z_i$ with $X_k \in \{Y_i, Z_i\}$, $* \in \{+, -, \cdot\}$, and $X_k \notin \{X_{k+1}, \ldots, X_{i-1}\}$. If no such indices i exist, then we set $q_k = 1$. Similarly, for each input variable I_k we define $q_{I_k} = \max(q_{i_1}, \ldots, q_{i_n}) + 1$, where $i_1 > \cdots > i_n$ are those indices such that Γ_i is of the form $X_i := Y_i * Z_i$ with $I_k \in \{Y_i, Z_i\}$, $* \in \{+, -, \cdot\}$, and $I_k \notin \{X_1, \ldots, X_{i-1}\}$. We also define $q_\Gamma = \max(q_{I_1}, \ldots, q_{I_m}, q_{i_1}, \ldots, q_{i_n})$, where $I_1, \ldots, I_m$ are the input variables of Γ and $i_1, \ldots, i_n$ all indices i such that Γ_i is of the form $X_i := C_i$.

Example 3.1. Let us consider $\Gamma = (x_1 := 5, x_2 := a_1 \cdot a_2, x_1 := x_1 \cdot x_2, x_3 := x_1 + a_1)$, of length $l = 4$. The input variables are a_1 and a_2, and we distinguish x_3 as the sole output variable. This SLP thus computes the function $5a_1a_2 + a_1$. The associated computation graph, together with remaining path lengths are as pictured:



3.2. Transient ball evaluations

Our goal is to evaluate SLPs using transient ball arithmetic from section 2.4 instead of rounded ball arithmetic. Transient arithmetic is faster, but some adjustments are required in order to guarantee the correctness of the end-results. In order to describe and analyze the correctness of these adjustments, it is useful to introduce one more variant of ball arithmetic.

In *semi-exact ball arithmetic*, the operations on centers are approximate, but correctly rounded, whereas operations on radii are exact and certified:

$$\begin{aligned}\mathcal{B}(a,r^*) \pm^* \mathcal{B}(b,s^*) &:= \mathcal{B}(\circ[a \pm b], r^* + s^* + \bar{\varepsilon}_\circ(a \pm b))\\ \mathcal{B}(a,r^*) \cdot^* \mathcal{B}(b,s^*) &:= \mathcal{B}(\circ[a \cdot b], (|a| + r^*)\, s^* + |b| r^* + \bar{\varepsilon}_\circ(a \cdot b)),\end{aligned}$$

where $a, b \in \mathbb{A}_p$ and $r^*, s^* \in \mathbb{R}_p$. The extra terms $\bar{\varepsilon}_\circ(a * b)$ in the radius ensure that these definitions satisfy the inclusion principle.

In order to measure how far transient arithmetic can deviate from this idealized semi-exact arithmetic, let $H_k := \frac{1}{1} + \cdots + \frac{1}{k}$ be the k-th harmonic number and let $H_{k,l} := H_l - H_k$ for all $l \geqslant k$. The following theorem shows how to certify transient ball evaluations:

THEOREM 3.2. [10, Theorem 5] *Let Γ be an SLP of length l as above and let $\alpha > 0$ be an arbitrary parameter such that $1 + \alpha > (1+\epsilon)^{4q_\Gamma}$, where $\epsilon = \epsilon_{\mathbb{A}_p}$ is as in (2.7). Consider two evaluations of Γ for different types of ball arithmetic. For the first evaluation, we use semi-exact ball arithmetic with $\bar{\varepsilon}_\circ(x) = |x_\circ|\epsilon$. For the second evaluation, we use transient ball arithmetic with the additional property that any input or constant ball $\mathcal{B}(a,r^*)$ is replaced by a larger ball $\mathcal{B}(a,r)$ with*

$$r \geqslant \max\left(|a|\left((1+\epsilon)^{\beta q_\Gamma} - 1\right), (1+\alpha)\, r^*\right),$$

where

$$\beta \geqslant \max\left(3, \frac{1+\alpha}{\alpha}\gamma\right), \quad \gamma \geqslant H_{q_\Gamma}(1+\epsilon)^{4q_\Gamma}\frac{\alpha}{1+\alpha}\left(1 - \frac{(1+\epsilon)^{4q_\Gamma}}{1+\alpha}\right)^{-1}.$$

Assume that no underflow or overflow occurs during the second evaluation. For any corresponding outputs $\mathcal{B}(c,t^*)$ and $\mathcal{B}(c,t)$ for the first and second evaluations, we then have $t^* \leqslant t$.

Given a fixed α and $q_\Gamma = O(\epsilon^{-1})$, we note that we may take β and γ such that $(1+\epsilon)^{\beta q_\Gamma} - 1 = O(\epsilon q_\Gamma \log q_\Gamma)$. The value of the parameter α may be optimized as a function of the SLP and the input. Without entering details, α should be taken large when the input radii are small, and small when these radii are large. The latter case occurs typically within subdivision algorithms. For our main application when we want to compute error bounds, the input radii are typically small or even zero.

For our implementations in MATHEMAGIX [12] and JIL [1], we found it useful to further simplify the bounds from Theorem 3.2, assuming that ϵ is "sufficiently small".

COROLLARY 3.3. *With the notation from Theorem 3.2, assume that $(4q_\Gamma)^2 \leqslant \epsilon^{-1}$. In order to apply Theorem 3.2, it is sufficient to take $\alpha > \eta := (4q_\Gamma + 1)\,\epsilon$ and*

$$\beta \geqslant \max\left(3, (\ln q_\Gamma + 1)\frac{(1+\eta)\,(1+\alpha)}{\alpha - \eta}\right).$$

Proof. The harmonic series satisfies the well-known inequality $\ln q_\Gamma \leqslant H_{q_\Gamma} \leqslant \ln q_\Gamma + 1$, of which we only use the right-hand part. Lemma 2.1 also yields $(1+\epsilon)^{4q_\Gamma} \leqslant 1 + \eta$. Given that $\alpha > \eta$, it follows that

$$\left(1 - \frac{(1+\epsilon)^{4q_\Gamma}}{1+\alpha}\right)^{-1} = \frac{1+\alpha}{1+\alpha-(1+\epsilon)^{4q_\Gamma}} \leqslant \frac{1+\alpha}{\alpha-\eta}.$$

Therefore,

$$H_{q_\Gamma}(1+\epsilon)^{4q_\Gamma}\left(1 - \frac{(1+\epsilon)^{4q_\Gamma}}{1+\alpha}\right)^{-1} \leqslant (\ln(q_\Gamma)+1)\frac{(1+\eta)\,(1+\alpha)}{\alpha-\eta}. \qquad \square$$

3.3. From balls to matryoshki

The semi-exact ball arithmetic from the previous subsection naturally adapts to matryoshki as follows: given $\boldsymbol{A}, \boldsymbol{B} \in \mathcal{B}(\mathbb{A}_p, \mathbb{R})$ and $r^*, s^* \in \mathbb{R}$, we define

$$\begin{aligned} \mathcal{B}(\boldsymbol{A}, r^*) \pm^* \mathcal{B}(\boldsymbol{B}, s^*) &:= \mathcal{B}(\boldsymbol{A} \pm^* \boldsymbol{B}, r^* + s^* + \bar{\varepsilon}_\circ(\boldsymbol{A} \pm \boldsymbol{B})), \\ \mathcal{B}(\boldsymbol{A}, r^*) \cdot^* \mathcal{B}(\boldsymbol{B}, s^*) &:= \mathcal{B}(\boldsymbol{A} \cdot^* \boldsymbol{B}, (|\boldsymbol{A}| + r^*)\, s^* + |\boldsymbol{B}|\, r^* + \bar{\varepsilon}_\circ(\boldsymbol{A} \cdot \boldsymbol{B})), \end{aligned}$$

where $\bar{\varepsilon}_\circ$ is extended to balls *via*

$$\bar{\varepsilon}_\circ(\boldsymbol{A}) := \bar{\varepsilon}_\circ(|\boldsymbol{A}|).$$

Recall that $\bar{\varepsilon}_\circ(|\boldsymbol{A}|)$ denotes an upper bound on the rounding errors involved in the computation of the norm of $\boldsymbol{A}$. Operations on centers use the semi-exact arithmetic from the previous subsection. This arithmetic is therefore a matryoshka lift of the elementary operations. The following theorem specifies the amount of inflation that is required in order to certify SLP evaluations using "transient matryoshka arithmetic".

THEOREM 3.4. *Let $\epsilon = \epsilon_{\mathbb{A}_p, \mathbb{R}_p}$, and let α, β, γ be as in Theorem 3.2. Consider two evaluations of Γ for two different types of arithmetic. For the first evaluation, we use semi-exact matryoshka arithmetic with $\bar{\varepsilon}_\circ(x) = |x_\circ|\,\epsilon$. For the second evaluation, we use transient matryoshka arithmetic with the additional property that any input or constant ball $\mathcal{B}(A, R^*, r^*)$ is replaced by a larger ball $\mathcal{B}(A, R, r) = \mathcal{B}(\boldsymbol{A}, r)$ with*

$$\begin{aligned} R &\geqslant \max(|A|\,((1+\epsilon)^{\beta q_\Gamma} - 1), (1+\alpha)\, R^*), \\ r &\geqslant \max(|\boldsymbol{A}|\,((1+\epsilon)^{\beta q_\Gamma} - 1), (1+\alpha)\, r^*). \end{aligned}$$

Assume that no underflow or overflow occurs during the second evaluation. For any corresponding outputs $\mathcal{B}(C, T^*, t^*)$ and $\mathcal{B}(C, T, t)$ for the first and second evaluations, we then have $T^* \leqslant T$ and $t^* \leqslant t$.

Proof. The theorem essentially follows by applying Theorem 3.2 twice: once for the big and once for the small radii. This is clear for the big radii T^* and T since the formulas for the big radii of matryoshki correspond with the formula for the radii of the corresponding ball arithmetic. For the small radii, we use essentially the same proof as in [10]. For convenience of the reader, we reproduce it here with the required minor adaptations.

Let $\mathcal{B}(C_k, T_k, t_k) = \mathcal{B}(\boldsymbol{C}_k, t_k)$ be the value of the variable X_k after the evaluation of $\Gamma_1, \ldots, \Gamma_k$ using transient matryoshki arithmetic. It will be convenient to systematically use star superscripts for the corresponding value $\mathcal{B}(C_k, T_k^*, t_k^*) = \mathcal{B}(\boldsymbol{C}_k^*, t_k^*)$ when using semi-exact matryoshka arithmetic. Let us show by induction on k that:

$$t_k \geqslant |\boldsymbol{C}_k|\,((1+\epsilon)^{\beta q_k} - 1). \tag{3.1}$$

If Γ_k is of the form $X_k := \mathcal{B}(\boldsymbol{C}_k, t_k)$, then we are done since $q_k \leqslant q_\Gamma$. Otherwise, Γ_k is of the form $X_k := Y_k * Z_k$. Writing $Y_k = \mathcal{B}(\boldsymbol{A}, r)$, we claim that

$$r \geqslant |\boldsymbol{A}|\,((1+\epsilon)^{\beta(q_k+1)} - 1).$$

This holds by assumption if Y_k is an input variable. Otherwise, let $i < k$ be the largest index such that $Y_k = X_i$. Then $q_i \geqslant q_k + 1$ by construction of q_i, whence our claim follows from the induction hypothesis. Similarly, writing $Z_k = \mathcal{B}(\boldsymbol{B}, s)$, we have $s \geqslant |\boldsymbol{B}|\,((1+\epsilon)^{\beta(q_k+1)} - 1)$.

Having shown our claim, let us first consider the case where $* \in \{+,-\}$. In this case we obtain

$$\begin{aligned} r+s &\geqslant (|\boldsymbol{A}|+|\boldsymbol{B}|)\,((1+\epsilon)^{\beta(q_k+1)}-1)\\ &\geqslant |\boldsymbol{A}*\boldsymbol{B}|\,((1+\epsilon)^{\beta(q_k+1)}-1). \end{aligned}$$

Combined with (2.9) and the inequalities

$$\begin{aligned} 1-\epsilon &\geqslant (1+\epsilon)^{-2}\\ ((1+\epsilon)^{A}-1)\,(1+\epsilon)^{-1} &\geqslant (1+\epsilon)^{A-1}-1, \end{aligned} \tag{3.2}$$

we deduce:

$$\begin{aligned} t_k &= \circ[r+s]\\ &\geqslant (r+s)\,(1+\epsilon)^{-1}\\ &\geqslant |\boldsymbol{A}*\boldsymbol{B}|\,((1+\epsilon)^{\beta(q_k+1)}-1)\,(1+\epsilon)^{-1}\\ &\geqslant |\boldsymbol{A}*_\circ\boldsymbol{B}|\,((1+\epsilon)^{\beta(q_k+1)}-1)\,(1+\epsilon)^{-1}\,(1-\epsilon)\\ &\geqslant |\boldsymbol{A}*_\circ\boldsymbol{B}|\,((1+\epsilon)^{\beta(q_k+1)}-1)\,(1+\epsilon)^{-3}\\ &\geqslant |\boldsymbol{A}*_\circ\boldsymbol{B}|\,((1+\epsilon)^{\beta q_k}-1). \end{aligned}$$

In case when $* = \cdot$, we obtain

$$\begin{aligned} |\boldsymbol{A}|s+|\boldsymbol{B}|r+rs &\geqslant |\boldsymbol{A}|s+|\boldsymbol{B}|r\\ &\geqslant 2\,|\boldsymbol{A}\cdot\boldsymbol{B}|\,((1+\epsilon)^{\beta(q_k+1)}-1). \end{aligned}$$

Combined with Lemma 2.4, the inequalities (3.2), (2.9), and $2\,(1+\epsilon)^{-4} > (1+\epsilon)^{-1}$ successively imply

$$\begin{aligned} t_k &= \circ[\,(|\boldsymbol{A}|+r)\,s+|\boldsymbol{B}|\,r]\\ &\geqslant (|\boldsymbol{A}|s+|\boldsymbol{B}|r+rs)\,(1+\epsilon)^{-4}\\ &\geqslant 2\,|\boldsymbol{A}\cdot\boldsymbol{B}|\,((1+\epsilon)^{\beta(q_k+1)}-1)\,(1+\epsilon)^{-4}\\ &\geqslant |\boldsymbol{A}\cdot_\circ\boldsymbol{B}|\,((1+\epsilon)^{\beta(q_k+1)}-1)\,(1-\epsilon)\,(1+\epsilon)^{-1}\\ &\geqslant |\boldsymbol{A}\cdot_\circ\boldsymbol{B}|\,((1+\epsilon)^{\beta q_k}-1), \end{aligned}$$

which achieves the induction. Then, for all $k \in \{1,\ldots,l\}$, we define

$$\gamma_k := \left(\frac{1}{1+\alpha}+\frac{\alpha}{\gamma\,(1+\alpha)}H_{q_k,q_\Gamma}\right)(1+\epsilon)^{4(q_\Gamma-q_k)}$$

so that $(1+\alpha)^{-1} \leqslant \gamma_k$. By the inequality that we imposed on γ, we have $\gamma_k \leqslant 1$. Using a second induction over k, let us next prove that

$$t_k^* \leqslant \gamma_k t_k. \tag{3.3}$$

Assume that this inequality holds up until order $k-1$. If Γ_k is of the form $X_k := \mathcal{B}(\boldsymbol{C}_k, t_k)$ then we are done by the fact that $\gamma_k \geqslant (1+\alpha)^{-1}$. If Γ_k is of the form $X_k := Y_k \pm Z_k$, then let i, $j<k$ be the largest integers so that $X_i = Y_k$ and $X_j = Z_k$, so we have $\min(q_i,q_j) \geqslant q_k+1$,

$$\max(\gamma_i,\gamma_j) \leqslant \left(\frac{1}{1+\alpha}+\frac{\alpha}{\gamma\,(1+\alpha)}H_{q_k+1,q_\Gamma}\right)(1+\epsilon)^{4(q_\Gamma-(q_k+1))},$$

and

$$\eta_k := \frac{|\boldsymbol{C}_k|\,\epsilon}{t_k} \leqslant \frac{\epsilon}{((1+\epsilon)^{\beta q_k}-1)} \leqslant \frac{1}{\beta q_k} \leqslant \frac{\alpha}{q_k\gamma\,(1+\alpha)},$$

using (3.1). In particular, we obtain $(\max(\gamma_i,\gamma_j)+\eta_k)(1+\epsilon)^4 \leqslant \gamma_k$. We denote by $\mathcal{B}(\boldsymbol{A}^*, r^*)$ (resp. $\mathcal{B}(\boldsymbol{B}^*, s^*)$) the matryoshka corresponding to the value of Y_k (resp. Z_k) in the semi-exact evaluation. Their corresponding values for the transient evaluation are denoted without the star superscript. Thanks to Theorem 3.2, we know that $T_k^* \leqslant T_k$, whence $|\boldsymbol{C}_k^*| \leqslant |\boldsymbol{C}_k|$ since $C_k^* = C_k$. With r and s as above, using (3.3) and (2.9), it follows that

$$\begin{aligned} t_k^* &= r^* + s^* + |\boldsymbol{C}_k^*|\,\epsilon \\ &\leqslant r^* + s^* + |\boldsymbol{C}_k|\,\epsilon \\ &\leqslant ((r+s)\max(\gamma_i,\gamma_j) + \eta_k t_k) \\ &\leqslant (\max(\gamma_i,\gamma_j)+\eta_k)\,t_k\,(1+\epsilon) \\ &\leqslant \gamma_k t_k. \end{aligned}$$

If Γ_k is of the form $X_k := Y_k \cdot Z_k$, then, thanks to Lemmas 2.3, 2.4, and inequality (3.3) a similar calculation yields

$$\begin{aligned} t_k^* &= |\boldsymbol{A}^*|\,s^* + (|\boldsymbol{B}^*| + s^*)\,r^* + |\boldsymbol{C}_k^*|\,\epsilon \\ &\leqslant |\boldsymbol{A}|\,s^* + (|\boldsymbol{B}| + s)\,r^* + \eta_k t_k \\ &\leqslant (|\boldsymbol{A}|\,s + (|\boldsymbol{B}| + s)\,r)\max(\gamma_i,\gamma_j) + \eta_k t_k \\ &\leqslant (\max(\gamma_i,\gamma_j)+\eta_k)\,t_k\,(1+\epsilon)^4 \\ &\leqslant \gamma_k t_k, \end{aligned}$$

which completes the second induction and the proof of this theorem. □

Remark 3.5. In [10, section III.C], it is discussed how to manage underflows and overflows in Theorem 3.2. A similar discussion applies to Theorem 3.4.

3.4. Applications

Let us now give two direct applications of matryoshka arithmetic. Assume that we are given an SLP Γ that computes a function $f = E_\Gamma : \mathbb{A}^m \to \mathbb{A}^n$ and denote by $f_\circ : \mathbb{A}_p^m \to \mathbb{A}_p^n$ the function obtained by evaluating Γ using approximate floating point arithmetic over $\mathbb{A}_p$. Matryoshki yield an efficient way to statically bound the error $f_\circ(a) - f(a)$ for a inside some fixed poly-ball, as follows:

PROPOSITION 3.6. *Let $\boldsymbol{A} = (\boldsymbol{A}_1, \ldots, \boldsymbol{A}_m) \in \mathcal{B}(\mathbb{A}_p, \mathbb{R}_p)^m$ be a fixed poly-ball and $\mathcal{B}(\boldsymbol{A}, 0) := (\mathcal{B}(\boldsymbol{A}_1, 0), \ldots, \mathcal{B}(\boldsymbol{A}_m, 0)) \in \mathcal{B}(\mathbb{A}_p, \mathbb{R}_p, \mathbb{R}_p)^m$. Let $\boldsymbol{F} : \mathcal{B}(\mathbb{A}_p, \mathbb{R}_p, \mathbb{R}_p)^m \to (\mathbb{A}_p, \mathbb{R}_p, \mathbb{R}_p)^n$ be a poly-matryoshka lift of f and $\mathcal{B}(\boldsymbol{B}, E) := (\mathcal{B}(\boldsymbol{B}_1, E_1), \ldots, \mathcal{B}(\boldsymbol{B}_n, E_n)) := \boldsymbol{F}(\mathcal{B}(\boldsymbol{A}, 0))$. Then for all $a \in \mathbb{A}_p^m$ with $a_1 \in \boldsymbol{A}_1, \ldots, a_m \in \boldsymbol{A}_m$ and $i = 1, \ldots, n$, we have*

$$|f_{\circ,i}(a) - f_i(a)| \leqslant E_i.$$

Proof. Let $\boldsymbol{f} : \mathcal{B}(\mathbb{A}_p, \mathbb{R}_p)^m \to \mathcal{B}(\mathbb{A}_p, \mathbb{R}_p)^n$ be a ball lift for which $\boldsymbol{F}$ satisfies the matryoshka lift property. Since $\boldsymbol{A} \circ\!\!- a$ by assumption, we have

$$\boldsymbol{F}(\mathcal{B}(\boldsymbol{A}, 0)) = \mathcal{B}(\boldsymbol{B}, E) \circ\!\!- \boldsymbol{f}(\mathcal{B}(a, 0)) = \mathcal{B}(f_\circ(a), r),$$

for some $r \in \mathbb{R}_p^n$. The ball lift property also yields

$$\mathcal{B}(f_\circ(a), r) \circ\!\!- f(a),$$

so $|f_{\circ,i}(a) - f_i(a)| \leqslant r_i \leqslant E_i$ for $i = 1, \ldots, n$. □

As our second application, we wish to construct a ball lift of f that is almost as efficient to compute as f itself, provided that the input balls are included into fixed large balls $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_m \in \mathcal{B}(\mathbb{A}_p, \mathbb{R}_p)$ as above. For this, we first compute bounds $B_{i,j} \in \mathbb{R}_p^{\geqslant}$ for the Jacobian matrix of f:

$$\sup_{A \circ\!\!-\!\!- a} \left| \frac{\partial f_i}{\partial x_j}(a) \right| \leqslant B_{i,j}. \tag{3.4}$$

For this, it suffices to evaluate a ball lift of the Jacobian of f at $\boldsymbol{A}$, which yields a matrix $\boldsymbol{J} \in \mathcal{B}(\mathbb{A}_p, \mathbb{R}_p)^{n \times m}$, after which we take $B_{i,j} := |\boldsymbol{J}_{i,j}|$. We recall that an SLP for the computation of the Jacobian can be constructed using the algorithm by Baur and Strassen [3].

PROPOSITION 3.7. *Let $\lg k := \lceil \log_2 k \rceil$ for all integers $k \geqslant 1$ and $\eta := \eta_{\mathbb{R}_p}$. With the above notation, assume that $(\lg m + 5)^2 \leqslant \epsilon^{-1}$, and let E be as in Proposition 3.6. For every $\boldsymbol{a} = \mathcal{B}(a, r) \in \mathcal{B}(\mathbb{A}_p, \mathbb{R}_p)^m$ with $\boldsymbol{a}_1 \subseteq \boldsymbol{A}_1, \ldots, \boldsymbol{a}_m \subseteq \boldsymbol{A}_m$, let*

$$\boldsymbol{f}_*(\boldsymbol{a}) := \mathcal{B}(f_\circ(a), \circ[(E + B\,r)\,(1 + (\lg m + 8)\,\epsilon) + (m+1)\,\eta]).$$

Then $\boldsymbol{f}_*$ is a ball lift of f. (Note that this lift is only defined on the set of $\boldsymbol{a} \in \mathcal{B}(\mathbb{A}_p, \mathbb{R}_p)^m$ such that $\boldsymbol{a}_1 \subseteq \boldsymbol{A}_1, \ldots, \boldsymbol{a}_m \subseteq \boldsymbol{A}_m$.)

Proof. Let $b_1 \in \boldsymbol{a}_1, \ldots, b_m \in \boldsymbol{a}_m$. Then, for $i = 1, \ldots, n$, we have

$$|f_i(b) - f_i(a)| \leqslant B_{i,1} r_1 + \cdots + B_{i,m} r_m,$$

by (3.4) and the mean value theorem. In other words,

$$\mathcal{B}(f(a), B\,r) \circ\!\!-\!\!- f(b).$$

Applying Proposition 3.6, it follows that

$$\mathcal{B}(f_\circ(a), E + B\,r) \circ\!\!-\!\!- f(b).$$

By computing the sum $B_{i,1} r_1 + \cdots + B_{i,m} r_m$ via a balanced binary tree and using (2.8), we obtain

$$\begin{aligned}
& E_i + B_{i,1} r_1 + \cdots + B_{i,m} r_m \\
\leqslant\ & \circ[(E_i + B_{i,1} r_1 + \cdots + B_{i,m} r_m)]\,(1+\epsilon)^{\lg m + 1} + m\,\eta \\
\leqslant\ & (\circ[E_i + B_{i,1} r_1 + \cdots + B_{i,m} r_m]\,(1 + (\lg m + 8)\,\epsilon) + (m+1)\,\eta)\,(1+\epsilon)^{-7} \\
\leqslant\ & \circ[(E_i + B_{i,1} r_1 + \cdots + B_{i,m} r_m)\,(1 + (\lg m + 8)\,\epsilon) + (m+1)\,\eta],
\end{aligned}$$

where we also used Lemma 2.1 and the fact that the computation of $\lg m + 6$ is exact. □

4. DIVISION

It is classical to extend ball arithmetic to other operations like division, exponentiation, logarithm, trigonometric functions, etc. In this section, we will focus on the particular case of division. It will actually suffice to study reciprocals $\iota(x) := x^{-1}$, since $x/y = x\,\iota(y)$.

Divisions and reciprocals raise the new difficulty of divisions by zero. We recall that the IEEE 754 provides a special not-a-number element NaN in $\mathbb{R}_p$ that corresponds to undefined results. All arithmetic operations on NaN return NaN, so we may check whether some error occurred during a computation, simply by checking whether one of the return values is NaN. For exact arithmetic, we will assume that $\mathbb{R}$ is extended in a similar way with an element NaN such that $\iota(0) := \text{NaN}$.

4.1. Reciprocals of balls

In the remainder of this section, assume that $\mathbb{A} = \mathbb{R}$ or $\mathbb{A} = \mathbb{C}$. In order to extend reciprocals to balls, it is convenient to introduce the function

$$\bar{\iota}(x) := \begin{cases} \iota(x), & \text{if } x > 0 \\ \text{NaN}, & \text{if } x \leqslant 0 \end{cases}$$

Then the exact reciprocal of a ball $\mathcal{B}(a,r) \in \mathcal{B}(\mathbb{A},\mathbb{R})$ can be defined as follows:

$$\iota(\mathcal{B}(a,r)) \;:=\; \mathcal{B}(\iota(a), r\,\bar{\iota}((|a|-r)\,|a|)), \tag{4.1}$$

where we note that

$$r\,\iota((|a|-r)\,|a|) = \frac{r}{(|a|-r)\,|a|} = \frac{1}{|a|-r} - \frac{1}{|a|}.$$

If $\mathcal{B}(a,r) \in \mathcal{B}(\mathbb{A}_p,\mathbb{R}_p)$ then we may also use the following formula

$$\iota_\circ(\mathcal{B}(a,r)) \;:=\; \mathcal{B}(\iota_\circ(a), \circ[r\,\bar{\iota}((|a|-r)\,|a|)]) \tag{4.2}$$

for transient ball arithmetic and

$$\iota^*(\mathcal{B}(a,r^*)) \;:=\; \mathcal{B}(\iota_\circ(a), r\,\bar{\iota}((|a|-r)\,|a|) + \bar{\varepsilon}_\circ(\iota(a))). \tag{4.3}$$

for semi-exact ball arithmetic.

We assume that the rounded counterpart of the reciprocal verifies the condition

$$|\iota_\circ(a) - \iota(a)| \leqslant |\iota_\circ(a)|\,\epsilon_{\mathbb{A}_p}, \tag{4.4}$$

for all invertible a in $\mathbb{A}_p$. For IEEE 754 arithmetic, this condition is naturally satisfied for $\epsilon_{\mathbb{R}_p} \geqslant 2^{-p}$ when $\mathbb{A}_p = \mathbb{R}_p$, and in absence of overflows and underflows. If $\mathbb{A}_p = \mathbb{C}_p$ and $a, b \in \mathbb{R}_p$, then we compute reciprocals using the formula

$$\iota_\circ(a + \mathrm{i}\,b) \;:=\; \circ[(a - \mathrm{i}\,b)\,\iota(a^2+b^2)].$$

For this definition, we have

$$|\iota_\circ(a+\mathrm{i}\,b) - \iota(a+\mathrm{i}\,b)| \;\leqslant\; |a-\mathrm{i}\,b|\,|\circ[\iota(a^2+b^2)] - \iota(a^2+b^2)| + |\iota_\circ(a+\mathrm{i}\,b)|\,\epsilon_{\mathbb{R}_p}.$$

Now

$$\begin{aligned} a^2+b^2 \;&\leqslant\; (a \cdot_\circ a + b \cdot_\circ b)\,(1+\epsilon_{\mathbb{R}_p}) \;\leqslant\; \circ[a^2+b^2]\,(1+\epsilon_{\mathbb{R}_p})^2 \\ a^2+b^2 \;&\geqslant\; (a \cdot_\circ a + b \cdot_\circ b)\,(1-\epsilon_{\mathbb{R}_p}) \;\geqslant\; \circ[a^2+b^2]\,(1-\epsilon_{\mathbb{R}_p})^2, \end{aligned}$$

whence

$$\begin{aligned} \iota(a^2+b^2) \;&\geqslant\; \iota(\circ[a^2+b^2])\,(1+\epsilon_{\mathbb{R}_p})^{-2} \;\geqslant\; \circ[\iota(a^2+b^2)]\,(1+\epsilon_{\mathbb{R}_p})^{-2}(1-\epsilon_{\mathbb{R}_p}) \\ \iota(a^2+b^2) \;&\leqslant\; \iota(\circ[a^2+b^2])\,(1-\epsilon_{\mathbb{R}_p})^{-2} \;\leqslant\; \circ[\iota(a^2+b^2)]\,(1-\epsilon_{\mathbb{R}_p})^{-2}(1+\epsilon_{\mathbb{R}_p}). \end{aligned}$$

For $\epsilon_{\mathbb{R}_p} \leqslant 1/16$, it follows that

$$\begin{aligned} |a-\mathrm{i}\,b|\,|\circ[\iota(a^2+b^2)] - \iota(a^2+b^2)| \;&\leqslant\; |a-\mathrm{i}\,b|\,(\circ[\iota(a^2+b^2)]\,(3\,\epsilon_{\mathbb{R}_p} + 6\,\epsilon_{\mathbb{R}_p}^2)) \\ &\leqslant\; |\circ[(a-\mathrm{i}\,b)\,\iota(a^2+b^2)]|\,(3\,\epsilon_{\mathbb{R}_p} + 10\,\epsilon_{\mathbb{R}_p}^2) \\ &=\; |\iota_\circ(a+\mathrm{i}\,b)|\,(3\,\epsilon_{\mathbb{R}_p} + 10\,\epsilon_{\mathbb{R}_p}^2) \\ &\leqslant\; |\iota_\circ(a+\mathrm{i}\,b)|\,(4\,\epsilon_{\mathbb{R}_p}). \end{aligned}$$

Consequently,

$$|\iota_\circ(a+\mathrm{i}\,b) - \iota(a+\mathrm{i}\,b)| \;\leqslant\; |\iota_\circ(a+\mathrm{i}\,b)|\,(5\,\epsilon_{\mathbb{R}_p}).$$

This shows that we may take $\epsilon_{\mathbb{C}_p} := 5\,\epsilon_{\mathbb{R}_p}$.

Let us denote $\epsilon := \max(\epsilon_{\mathbb{A}_p}, \epsilon_{\mathbb{R}_p})$ and assume that $\epsilon \leqslant 1/16$. We are now in a position to prove a counterpart of Lemma 2.3 for division.

LEMMA 4.1. *Let $\mathcal{B}(a,r) \in \mathcal{B}(\mathbb{A}_p, \mathbb{R}_p)$ and $\kappa \in \mathbb{R}_p^{\geqslant}$ be such that $0 \leqslant \frac{r}{|a|-r} \leqslant \kappa$. Assume that the computation of*

$$R := \circ[r\,\iota((|a|-r)\,|a|)]$$

involves no overflows and no underflows. Then $r\,\iota((|a|-r)\,|a|) \leqslant R\,(1+\epsilon)^{\kappa+7}$.

Proof. We have

$$\begin{aligned} |a|_\circ - r &\leqslant |a|\,(1-\epsilon)^{-1} - r \\ &= (|a| - r + r\epsilon)\,(1-\epsilon)^{-1} \\ &\leqslant (|a|-r)\,(1+\kappa\epsilon)\,(1-\epsilon)^{-1}, \end{aligned}$$

whence

$$\begin{aligned} (|a|-r)\,|a| &\geqslant (|a|_\circ - r)\,|a|\,(1-\epsilon)\,(1+\kappa\epsilon)^{-1} \\ &\geqslant (|a|_\circ - r)\,|a|_\circ\,(1-\epsilon)^2\,(1+\kappa\epsilon)^{-1} \\ &\geqslant (|a|_\circ -_\circ r)\,|a|_\circ\,(1-\epsilon)^3\,(1+\kappa\epsilon)^{-1} \\ &\geqslant (|a|_\circ -_\circ r) \cdot_\circ |a|_\circ\,(1-\epsilon)^4\,(1+\kappa\epsilon)^{-1}, \end{aligned}$$

whence

$$\begin{aligned} r\,\iota((|a|-r)\,|a|) &\leqslant r\,\iota((|a|_\circ -_\circ r) \cdot_\circ |a|_\circ)\,(1-\epsilon)^{-4}\,(1+\kappa\epsilon) \\ &\leqslant r\,\iota_\circ((|a|_\circ -_\circ r) \cdot_\circ |a|_\circ)\,(1-\epsilon)^{-4}\,(1+\kappa\epsilon)\,(1+\epsilon) \\ &\leqslant \circ[r\,\iota((|a|-r)\,|a|)]\,(1-\epsilon)^{-4}\,(1+\kappa\epsilon)\,(1+\epsilon)^2. \end{aligned}$$

We conclude by observing that $(1-\epsilon)^{-4}\,(1+\kappa\epsilon)\,(1+\epsilon)^2 \leqslant (1+\epsilon)^{\kappa+7}$, since $\epsilon \leqslant 1/16$. □

4.2. Transient evaluation

We adapt the definition of the remaining path length to our extended notion of SLPs with reciprocals. We now take $q_k = \max(q_{i_1}, \ldots, q_{i_n}) + 1$, where $i_1 > \cdots > i_n$ are those indices $i > k$ such that Γ_i is of the form $X_i := Y_i * Z_i$ with $X_k \in \{Y_i, Z_i\}$, $* \in \{+, -, \cdot\}$, and $X_k \notin \{X_{k+1}, \ldots, X_{i-1}\}$, or Γ_i is of the form $X_i := \iota(X_k)$ and $X_k \notin \{X_{k+1}, \ldots, X_{i-1}\}$. Lemma 4.1 allows us to extend Theorem 3.2 as follows.

THEOREM 4.2. *With the setup from Theorem 3.2, assume that $1 + \alpha \geqslant (1+\epsilon)^{(\kappa+7)q_\Gamma}$ and that Γ may also contain reciprocals. During the second evaluation, assume that no underflow or overflow occurs and that $0 \leqslant \frac{r}{|a|-r} \leqslant \kappa$ for every computation of a reciprocal of $\mathcal{B}(a,r)$, where $\kappa \in \mathbb{R}_p^{\geqslant}$. Assume in addition that $(\beta q_\Gamma)^2 \leqslant \epsilon^{-1}$ and that the following two inequalities hold:*

$$\gamma \geqslant H_{q_\Gamma}(1+\epsilon)^{(\kappa+7)q_\Gamma} \frac{\alpha}{1+\alpha} \left(1 - \frac{(1+\epsilon)^{(\kappa+7)q_\Gamma}}{1+\alpha}\right)^{-1}, \qquad \beta \geqslant \max\left(\frac{\kappa+9}{2}, \frac{1+\alpha}{\alpha}\gamma\right).$$

Then $t^* \leqslant t$ for any corresponding outputs $\mathcal{B}(c,t^*)$ and $\mathcal{B}(c,t)$ for the two evaluations.

Proof. We adapt the proof of [10, Theorem 5], by showing that the two inductions still go through in the case of reciprocals. For the first induction, consider an instruction $X_k := \iota(Y_k)$, where $Y_k = \mathcal{B}(a,r)$ and $0 \leqslant \frac{r}{|a|-r} \leqslant \kappa$. Recall that

$$r \geqslant |a|\,((1+\epsilon)^{\beta(q_k+1)} - 1).$$

Using $\beta \geqslant 4$, $(\beta q_\Gamma)^2 \leqslant \epsilon^{-1}$, and Lemma 2.1, we note that

$$\begin{aligned}(1+\epsilon)^{\beta-2}(1-\beta\epsilon) &\leqslant (1+(\beta-1)\epsilon)(1-\beta\epsilon)\\ &= 1-\epsilon-\beta(\beta-1)\epsilon^2\\ &\leqslant 1-\epsilon. \end{aligned} \tag{4.5}$$

Combined with Lemmas 4.1 and 2.1, we deduce that

$$\begin{aligned} t_k &= \circ[r\,\iota((|a|-r)\,|a|)]\\ &\geqslant \frac{r}{|a|\,(|a|-r)}(1+\epsilon)^{-(\kappa+7)}\\ &= |\iota(a)|\frac{1}{|a|/r-1}(1+\epsilon)^{-(\kappa+7)}\\ &\geqslant |\iota(a)|\frac{((1+\epsilon)^{\beta(q_k+1)}-1)}{2-(1+\epsilon)^{\beta(q_k+1)}}(1+\epsilon)^{-(\kappa+7)}\\ &\geqslant |\iota_\circ(a)|\,(1-\epsilon)\frac{((1+\epsilon)^{\beta(q_k+1)}-1)}{2-(1+\beta\epsilon)}(1+\epsilon)^{-(\kappa+7)}\\ &\geqslant |\iota_\circ(a)|\,(1-\epsilon)\frac{((1+\epsilon)^{\beta q_k}-1)\,(1+\epsilon)^{\beta}}{2-(1+\beta\epsilon)}(1+\epsilon)^{-(\kappa+7)}\\ &\geqslant |\iota_\circ(a)|\,((1+\epsilon)^{\beta q_k}-1)\,(1+\epsilon)^{2\beta-(\kappa+9)}.\\ &\geqslant |\iota_\circ(a)|\,((1+\epsilon)^{\beta q_k}-1). \end{aligned}$$

For the second induction, we redefine

$$\gamma_k := \left(\frac{1}{1+\alpha}+\frac{\alpha}{\gamma(1+\alpha)}H_{q_k,q_\Gamma}\right)(1+\epsilon)^{(\kappa+7)(q_\Gamma-q_k)}.$$

Assume $X_k := \iota(Y_k)$, let $\mathcal{B}(a,r^*)$ (resp. $\mathcal{B}(a,r)$) be the value of Y_k in the semi-exact evaluation (resp. transient evaluation), and let i be the biggest index so that $Y_k = X_i$. Then

$$\begin{aligned} t_k^* &= r^*\,\iota((|a|-r^*)\,|a|)+\bar{\varepsilon}_\circ(\iota(a))\\ &\leqslant r^*\,\iota((|a|-r^*)\,|a|)+\eta_k t_k\\ &\leqslant \gamma_i r\,\iota((|a|-r)\,|a|)+\eta_k t_k\\ &\leqslant \gamma_i(1+\epsilon)^{\kappa+7}t_k+\eta_k t_k\\ &\leqslant \gamma_k t_k. \end{aligned}$$

The two inductions still hold for the other operations since we increased β and γ. □

If the conditions of the theorem are all satisfied for a given input poly-ball $\mathcal{B}(b,s)$, then we note that the numeric evaluation of the SLP at any point $\tilde{b}$ with $\mathcal{B}(b,s) \circ\!\!-\!\!- \tilde{b}$ never results in a division by zero. Conversely however, even when all these numerical evaluations are legit, it may happen that the condition that $\frac{r}{|a|-r} \leqslant \kappa$ is violated for the computation of some reciprocal $\iota(\mathcal{B}(a,r))$. This can either be due to an overly optimistic choice of κ or to the classical phenomenon of overestimation.

Intuitively speaking, a low choice of κ means that balls that need to be inverted should neatly steer away from zero, even when inflating the radius by a small constant factor. In fact, this is often a reasonable requirement, in which case one may simply take $\kappa := 1$. A somewhat larger choice like $\kappa := 10$ remains appropriate and has the advantage of increasing the resilience of reciprocal computations (the condition $\frac{r}{|a|-r} \leqslant \kappa$ being easier to satisfy). Large values of κ deteriorate the quality of the obtained bounds for a dubious further gain in resilience.

4.3. Reciprocals of matryoshki

The formulas (4.1), (4.2), and (4.3) for reciprocals can again be specialized to matryoshki modulo the precaution that we should interpret $|a|$ as a lower bound for the norm. For $\boldsymbol{A} := \mathcal{B}(a, R)$ and $\boldsymbol{A}^* := \mathcal{B}(a, R^*)$, this leads to the definitions

$$\begin{aligned}
\iota(\mathcal{B}(\boldsymbol{A}, r)) &:= \mathcal{B}(\iota(\boldsymbol{A}), r\,\bar{\iota}((\lfloor \boldsymbol{A} \rfloor - r)\,\lfloor \boldsymbol{A} \rfloor)) \\
\iota_\circ(\mathcal{B}(\boldsymbol{A}, r)) &:= \mathcal{B}(\iota_\circ(\boldsymbol{A}), \circ[r\,\bar{\iota}((\lfloor \boldsymbol{A} \rfloor - r)\,\lfloor \boldsymbol{A} \rfloor)]) \\
\iota^*(\mathcal{B}(\boldsymbol{A}^*, r^*)) &:= \mathcal{B}(\iota_\circ(\boldsymbol{A}^*), r^*\,\bar{\iota}((\lfloor \boldsymbol{A}^* \rfloor - r^*)\,\lfloor \boldsymbol{A}^* \rfloor) + \bar{\varepsilon}_\circ(\iota(\boldsymbol{A}^*))),
\end{aligned}$$

where we use the lower bound notation $\lfloor \boldsymbol{A} \rfloor := |a| - R$. In addition to (2.9), we will assume that the rounded counterpart of the reciprocals also verifies the condition

$$|\iota(\boldsymbol{a}) -_{\mathrm{vec}} \iota_\circ(\boldsymbol{a})| \leqslant |\iota_\circ(\boldsymbol{a})|\,\epsilon_{\mathcal{B}(\mathbb{A}_p, \mathbb{R}_p)} \tag{4.6}$$

for all invertible balls $\boldsymbol{a}$ in $\mathcal{B}(\mathbb{A}_p, \mathbb{R}_p)$. For what follows, let $\epsilon := \epsilon_{\mathcal{B}(\mathbb{A}_p, \mathbb{R}_p)}$.

LEMMA 4.3. *Let* $\mathrm{K}, \kappa \in \mathbb{R}_p^{\geqslant}$ *be such that* $\mathrm{E} := 1 - (1 + \mathrm{K}\,\epsilon)^{-1}\,(1 - \epsilon)^2 \leqslant 1/16$. *Let* $\mathcal{B}(\boldsymbol{A}, r) = \mathcal{B}(A, R, r) \in \mathcal{B}(\mathbb{A}_p, \mathbb{R}_p, \mathbb{R}_p)$ *be such that* $0 \leqslant \frac{R}{\lfloor a \rfloor - R} \leqslant \mathrm{K}$ *and* $0 \leqslant \frac{r}{\lfloor A \rfloor - r} \leqslant \kappa$. *If the computation of*

$$R := \circ[r\,\iota((\lfloor \boldsymbol{A} \rfloor - r)\,\lfloor \boldsymbol{A} \rfloor)]$$

involves no overflows and no underflows, then $r\,\iota((\lfloor \boldsymbol{A} \rfloor - r)\,\lfloor \boldsymbol{A} \rfloor) \leqslant R\,(1 + \epsilon)^{(\mathrm{K}+3)(\kappa+2)+5}$.

Proof. From

$$\begin{aligned}
\lfloor \boldsymbol{A} \rfloor_\circ &:= |A|_\circ -_\circ R \\
&\leqslant (|A|\,(1-\epsilon)^{-1} - R)\,(1-\epsilon)^{-1} \\
&= (|A| - R + R\,\epsilon)\,(1-\epsilon)^{-2} \\
&\leqslant (|A| - R)\,(1 + \mathrm{K}\,\epsilon)\,(1-\epsilon)^{-2} \\
&= \lfloor \boldsymbol{A} \rfloor\,(1 - \mathrm{E})^{-1}
\end{aligned}$$

and

$$\begin{aligned}
\lfloor \boldsymbol{A} \rfloor_\circ - r &\leqslant \lfloor \boldsymbol{A} \rfloor\,(1-\mathrm{E})^{-1} - r \\
&= (\lfloor \boldsymbol{A} \rfloor - r + r\,\mathrm{E})\,(1-\mathrm{E})^{-1} \\
&\leqslant (\lfloor \boldsymbol{A} \rfloor - r)\,(1 + \kappa\,\mathrm{E})\,(1-\mathrm{E})^{-1},
\end{aligned}$$

we obtain

$$\begin{aligned}
(\lfloor \boldsymbol{A} \rfloor - r)\,\lfloor \boldsymbol{A} \rfloor &\geqslant (\lfloor \boldsymbol{A} \rfloor_\circ - r)\,\lfloor \boldsymbol{A} \rfloor_\circ\,(1 + \kappa\,\mathrm{E})^{-1}\,(1-\mathrm{E})^2 \\
&\geqslant (\lfloor \boldsymbol{A} \rfloor_\circ -_\circ r)\,\lfloor \boldsymbol{A} \rfloor_\circ\,(1 + \kappa\,\mathrm{E})^{-1}\,(1-\mathrm{E})^2\,(1-\epsilon) \\
&\geqslant (\lfloor \boldsymbol{A} \rfloor_\circ -_\circ r) \cdot_\circ \lfloor \boldsymbol{A} \rfloor_\circ\,(1 + \kappa\,\mathrm{E})^{-1}\,(1-\mathrm{E})^2\,(1-\epsilon)^2.
\end{aligned}$$

It follows that

$$\begin{aligned}
r\,\iota((\lfloor \boldsymbol{A} \rfloor - r)\,\lfloor \boldsymbol{A} \rfloor) &\leqslant r\,\iota((\lfloor \boldsymbol{A} \rfloor_\circ -_\circ r) \cdot_\circ \lfloor \boldsymbol{A} \rfloor_\circ)\,(1 + \kappa\,\mathrm{E})\,(1-\mathrm{E})^{-2}\,(1-\epsilon)^{-2} \\
&\leqslant r\,\iota_\circ((\lfloor \boldsymbol{A} \rfloor_\circ -_\circ r) \cdot_\circ \lfloor \boldsymbol{A} \rfloor_\circ)\,(1 + \kappa\,\mathrm{E})\,(1-\mathrm{E})^{-2}\,(1-\epsilon)^{-2}\,(1+\epsilon) \\
&\leqslant \circ[r\,\iota((\lfloor \boldsymbol{A} \rfloor - r)\,\lfloor \boldsymbol{A} \rfloor)]\,(1 + \kappa\,\mathrm{E})\,(1-\mathrm{E})^{-2}\,(1-\epsilon)^{-2}\,(1+\epsilon)^2.
\end{aligned}$$

Using $\epsilon \leqslant 1/16$ and $\mathrm{E} \leqslant 1/16$, we have

$$(1-\mathrm{E})^{-1} \;=\; (1+\mathrm{K}\epsilon)\,(1-\epsilon)^{-2} \;\leqslant\; (1+\epsilon)^{\mathrm{K}+3}$$

and then

$$(1+\kappa\,\mathrm{E})\,(1-\mathrm{E})^{-2}\,(1-\epsilon)^{-2}\,(1+\epsilon)^{2} \;\leqslant\; (1-\mathrm{E})^{-(\kappa+2)}\,(1+\epsilon)^{5} \;\leqslant\; (1+\epsilon)^{(\mathrm{K}+3)(\kappa+2)+5}. \qquad \square$$

We are now ready to extend Theorem 3.4 to SLPs with reciprocals.

THEOREM 4.4. *Let* $\mathrm{K}, \kappa \in \mathbb{R}_p^{\geqslant}$ *be such that* $\mathrm{E} := 1-(1+\mathrm{K}\,\epsilon)^{-1}\,(1-\epsilon)^2 \leqslant 1/16$ *and let* $M := (\mathrm{K}+3)\,(\kappa+2)+5$. *With the setup from Theorem 3.4, assume that* $1+\alpha \geqslant (1+\epsilon)^{Mq_\Gamma}$ *and that* Γ *may also contain reciprocals. During the second evaluation, assume that no underflow or overflow occurs and that* $0 \leqslant \frac{R}{\lfloor A \rfloor} \leqslant \mathrm{K}$ *and* $0 \leqslant \frac{r}{\lfloor A \rfloor - r} \leqslant \kappa$ *hold for every computation of a reciprocal of* $\mathcal{B}(\boldsymbol{A}, r) = \mathcal{B}(A, R, r)$. *Assume in addition that* $(\beta q_\Gamma)^2 < \epsilon^{-1}$ *and that the following two inequalities hold:*

$$\begin{aligned}
\gamma \;&\geqslant\; H_{q_\Gamma}(1+\epsilon)^{Mq_\Gamma}\frac{\alpha}{1+\alpha}\left(1-\frac{(1+\epsilon)^{Mq_\Gamma}}{1+\alpha}\right)^{-1}\\
\beta \;&\geqslant\; \max\left(\frac{M+2}{2}, \frac{1+\alpha}{\alpha}\gamma\right).
\end{aligned}$$

Then $t^* \leqslant t$ *and* $T^* \leqslant T$ *for any corresponding outputs* $\mathcal{B}(C, T^*, t^*)$ *and* $\mathcal{B}(C, T, t)$ *for the first and second evaluations.*

Proof. We adapt the proof of Theorem 3.4 by showing that the two inductions still go through in the case of reciprocals. For the first induction, consider an instruction $X_k := \iota(Y_k)$, where $Y_k = \mathcal{B}(A, R, r) = \mathcal{B}(\boldsymbol{A}, r)$, with $0 \leqslant \frac{R}{\lfloor A \rfloor} \leqslant \mathrm{K}$ and $0 \leqslant \frac{r}{\lfloor A \rfloor - r} \leqslant \kappa$. Recall that

$$r \;\geqslant\; |\boldsymbol{A}|\,((1+\epsilon)^{\beta(q_k+1)}-1).$$

Using Lemma 4.3, the identity $|\iota(\boldsymbol{A})| = \frac{1}{\lfloor A \rfloor}$, as well as (4.5) and (3.2), we obtain

$$\begin{aligned}
t_k \;&=\; \circ[r\,\iota((\lfloor A \rfloor - r)\,\lfloor A \rfloor)]\\
&\geqslant\; \frac{r}{(\lfloor A \rfloor - r)\,\lfloor A \rfloor}\,(1+\epsilon)^{-M}\\
&=\; |\iota(\boldsymbol{A})|\,\frac{r}{\lfloor A \rfloor - r}\,(1+\epsilon)^{-M}\\
&=\; |\iota(\boldsymbol{A})|\,\frac{1}{\lfloor A \rfloor / r - 1}\,(1+\epsilon)^{-M}\\
&\geqslant\; |\iota(\boldsymbol{A})|\,\frac{1}{|\boldsymbol{A}| / r - 1}\,(1+\epsilon)^{-M}\\
&\geqslant\; |\iota(\boldsymbol{A})|\,\frac{(1+\epsilon)^{\beta(q_k+1)}-1}{2-(1+\epsilon)^{\beta(q_k+1)}-1}\,(1+\epsilon)^{-M}\\
&\geqslant\; |\iota_\circ(\boldsymbol{A})|\,(1-\epsilon)\,\frac{(1+\epsilon)^{\beta(q_k+1)}-1}{1-\beta\epsilon}\,(1+\epsilon)^{-M}\\
&\geqslant\; |\iota_\circ(\boldsymbol{A})|\,((1+\epsilon)^{\beta(q_k+1)}-1)\,(1+\epsilon)^{\beta-(M+2)}\\
&\geqslant\; |\iota_\circ(\boldsymbol{A})|\,((1+\epsilon)^{\beta q_k}-1)\,(1+\epsilon)^{\beta}\,(1+\epsilon)^{\beta-(M+2)}\\
&\geqslant\; |\iota_\circ(\boldsymbol{A})|\,((1+\epsilon)^{\beta q_k}-1).
\end{aligned}$$

For the second induction, we redefine

$$\gamma_k := \left(\frac{1}{1+\alpha} + \frac{\alpha}{\gamma\,(1+\alpha)}\,H_{q_k,q_\Gamma}\right)(1+\epsilon)^{M(q_\Gamma-q_k)}.$$

Assume $X_k := \iota(Y_k)$, let $\mathcal{B}(\boldsymbol{A}^*, r^*)$ (resp. $\mathcal{B}(\boldsymbol{A}, r)$) be the value of Y_k in the semi-exact evaluation (resp. transient evaluation), and let i be the biggest index so that $Y_k = X_i$. Then Lemma 4.3 implies

$$\begin{aligned} t_k^* &= r^* \iota((\lfloor \boldsymbol{A} \rfloor - r^*)\, \lfloor \boldsymbol{A} \rfloor) + \bar{\varepsilon}_\circ(\iota(\boldsymbol{A}^*)) \\ &\leqslant r^* \iota((\lfloor \boldsymbol{A} \rfloor - r^*)\, \lfloor \boldsymbol{A} \rfloor) + \eta_k t_k \\ &\leqslant \gamma_i r \iota((\lfloor \boldsymbol{A} \rfloor - r)\, \lfloor \boldsymbol{A} \rfloor) + \eta_k t_k \\ &\leqslant \gamma_i (1+\epsilon)^M t_k + \eta_k t_k \\ &\leqslant \gamma_k t_k. \end{aligned}$$

The two inductions still hold for the other operations since we increased β and γ. □

Similar comments as those made after Theorem 4.2 also apply to the above theorem. In particular, we recommend taking $\kappa, \mathrm{K} \in [1, 10]$.

5. GLOBAL PROJECTIVE BOUNDS

In the previous sections we have focussed on more efficient algorithms for the reliable evaluation of functions on fixed bounded poly-balls. This technique applies to very general SLPs that may involve operations like division, which are only locally defined. Nonetheless, polynomial SLPs, which only involve the ring operations $+, -, \cdot$ are important for many applications such as numerical homotopy continuation. In this special case, we will show how remove the locality restriction and allow for the global evaluation of such SLPs in a reliable and efficient way.

5.1. Homogenization of SLPs

Consider a polynomial map

$$\begin{aligned} P\colon \mathbb{A}^m &\longrightarrow \mathbb{A}^n \\ x = (x_1, \ldots, x_m) &\longmapsto P(x) = (P_1(x_1, \ldots, x_m), \ldots, P_n(x_1, \ldots, x_n)) \end{aligned}$$

for polynomials $P_1, \ldots, P_n \in \mathbb{A}[x_1, \ldots, x_m]$. Given $i \in \{1, \ldots, n\}$, the polynomial P_i is not homogeneous, in general, but there exists a homogeneous polynomial $P_i^{\mathrm{hom}} \in \mathbb{A}[x_1, \ldots, x_{m+1}]$ such that $P_i(x_1, \ldots, x_m) = P_i^{\mathrm{hom}}(x_1, \ldots, x_m, 1)$. This polynomial is unique up to multiplication by powers of x_{m+1}. The polynomials $P_1^{\mathrm{hom}}, \ldots, P_n^{\mathrm{hom}}$ give rise to a polynomial map

$$\begin{aligned} P^{\mathrm{hom}}\colon \mathbb{A}^{m+1} &\longrightarrow \mathbb{A}^n \\ x = (x_1, \ldots, x_{m+1}) &\longmapsto (P_1^{\mathrm{hom}}(x_1, \ldots, x_{n+1}), \ldots, P_n^{\mathrm{hom}}(x_1, \ldots, x_{n+1})), \end{aligned}$$

which we call a *homogenization* of P_i.

Assume now that the map $P = E_\Gamma$ can be computed using an SLP Γ of length l without divisions. Let us show how to construct an SLP Γ^{hom} such that $P^{\mathrm{hom}} := E_{\Gamma^{\mathrm{hom}}}$ is a homogenization of P. Consider the formal evaluation of Γ over $\mathbb{A}[x_1, \ldots, x_m]$, by applying E_Γ to the input arguments $x_1, \ldots, x_m$. Then the output value X_k and the input arguments Y_k and Z_k of an instruction Γ_k of the form $X_k := Y_k * Z_k$ ($* \in \{+, -, \cdot\}$) are polynomials in $\mathbb{A}[x_1, \ldots, x_{m+1}]$ that we denote by $\bar{X}_k$ $\bar{Y}_k$, and $\bar{Z}_k$, respectively (for instructions $X_k := C_k$, we set $\bar{X}_k := C_k$). We can recursively compute upper bounds $d_{\bar{X}_k, k}$, $d_{\bar{Y}_k, k}$, $d_{\bar{Z}_k, k}$ (abbreviated as $d_{\bar{X}_k}$, $d_{\bar{Y}_k}$, $d_{\bar{Z}_k}$) for the total degrees of these polynomials:

- If Γ_k is of the form $X_k := C_k$, then we take $d_{\bar{X}_k} := 0$.

- If Γ_k is of the form $X_k := Y_k \pm Z_k$, then $d_{\bar{X}_k} := \max(d_{\bar{Y}_k}, d_{\bar{Z}_k})$.
- If Γ_k is of the form $X_k := Y_k \cdot Z_k$, then $d_{\bar{X}_k} := d_{\bar{Y}_k} + d_{\bar{Z}_k}$.

Here $d_{\bar{Y}_k} := 1$ (resp. $d_{\bar{Z}_k} := 1$) whenever Y_k (resp. Z_k) is an input variable.

Now consider the program $\tilde{\Gamma}$ obtained by rewriting each instruction Γ_k as follows:

- If Γ_k is of the form $X_k := C_k$ or $X_k := Y_k \cdot Z_k$, then Γ_k is rewritten into itself.
- If Γ_k is of the form $X_k := Y_k \pm Z_k$ with $d_{\bar{Y}_k} = d_{\bar{Z}_k}$, then Γ_k is rewritten into itself.
- If Γ_k is of the form $X_k := Y_k \pm Z_k$ with $d_{\bar{Y}_k} < d_{\bar{Z}_k}$, then Γ_k is rewritten into two instructions $A_k := Y_k \cdot x_{m+1}^{d_{\bar{Z}_k} - d_{\bar{Y}_k}}$ and $X_k := A_k \pm Z_k$, for some new auxiliary variable A_k.
- If Γ_k is of the form $X_k := Y_k \pm Z_k$ with $d_{\bar{Z}_k} < d_{\bar{Y}_k}$, then Γ_k is rewritten into two instructions $A_k := Z_k \cdot x_{m+1}^{d_{\bar{Y}_k} - d_{\bar{Z}_k}}$ and $X_k := Y_k \pm A_k$, for some new auxiliary variable A_k.

By induction, one verifies that the image of Γ_k under this rewriting computes the unique homogenization $E_{\Gamma,k}^{\hom}$ of $E_{\Gamma,k}$ of degree $d_{\bar{X}_k}$ for $k = 1, \ldots, l$. We obtain $\Gamma^{\hom}$ by prepending $\tilde{\Gamma}$ with instructions that compute all powers x_{m+1}^i that occur in $\tilde{\Gamma}$. This can for instance be done using binary powering: $x_{m+1}^{2i} := x_{m+1}^i \cdot x_{m+1}^i$ and $x_{m+1}^{2i+1} := x_{m+1}^{2i} \cdot x_{m+1}$ for all $i \geqslant 1$.

Example 5.1. Applying the homogenization procedure to the left-hand SLP below, we obtain the right-hand one:

$$
\begin{array}{ll}
 & x_3^2 := x_3 \cdot x_3 \\
 & x_3^3 := x_3^2 \cdot x_3 \\
 & x_3^6 := x_3^3 \cdot x_3^3 \\
 & x_3^7 := x_3^6 \cdot x_3 \\
V_1 := x_1 \cdot x_1 & V_1 := x_1 \cdot x_1 \\
V_2 := V_1 + x_2 & A_2 := x_2 \cdot x_3 \\
\qquad \xrightarrow{\text{homogenize}} & V_2 := V_1 + A_2 \\
V_3 := V_2 \cdot V_2 & V_3 := V_2 \cdot V_2 \\
V_4 := V_3 + x_2 & A_4 := x_2 \cdot x_3^3 \\
 & V_4 := V_3 + A_4 \\
V_5 := V_3 \cdot V_3 & V_5 := V_3 \cdot V_3 \\
V_6 := V_5 + x_2 & A_4 := x_2 \cdot x_3^7 \\
 & V_4 := V_3 + A_4
\end{array}
$$

In this example, $d_{\bar{V}_1} = d_{\bar{V}_2} = 2$, $d_{\bar{V}_3} = d_{\bar{V}_4} = 4$, and $d_{\bar{V}_5} = d_{\bar{V}_6} = 8$. In the last instruction, we thus have $d_{\bar{V}_5} = 8 > d_{\bar{x}_2} = 1$, whence the exponent $7 = d_{\bar{V}_5} - d_{\bar{x}_2}$ in $A_4 := x_2 \cdot x_3^7$.

Remark 5.2. In the worst case, computing the powers of x_{m+1} using binary powering may give rise to a logarithmic overhead. For instance, there is a straightforward SLP Γ of length l proportional to n that computes the polynomials $x^{3^k} + 1$ for $k = 1, \ldots, n$. But when computing using binary powering in order to compute its homogenization, the length of $\Gamma^{\hom}$ is proportional to $n \log n$.

An alternative way to compute the required powers is to systematically compute $x_{m+1}^{d_{\bar{X}_k}}$ and $(x_{m+1}^{-1})^{d_{\bar{X}_k}}$ for all k. In that case $x_{m+1}^{d_{\bar{Z}_k} - d_{\bar{Y}_k}}$ and $x_{m+1}^{d_{\bar{Y}_k} - d_{\bar{Z}_k}}$ can always be obtained using a simple multiplication and the length of $\Gamma^{\hom}$ remains bounded by $O(l)$ for an SLP Γ of length l. Of course, this requires division to be part of the SLP signature, or x_{m+1}^{-1} to be passed an argument, in addition to x_{m+1}.

Consider an output variable O_i with $i \in \{1, \ldots, n\}$ and denote by $E_{\Gamma,i}: \mathbb{A}^m \to \mathbb{A}$ and $E_{\Gamma^{\mathrm{hom}},i}: \mathbb{A}^{m+1} \to \mathbb{A}$ the i-th component of E_Γ and $E_{\Gamma^{\mathrm{hom}}}$, respectively. If $k \in \{1,\ldots,l\}$ is largest with $X_k = O_i$, then we define $d_i := d_{\bar{X}_i}$, and we note that $E_{\Gamma^{\mathrm{hom}},i}$ is the unique homogenization of $E_{\Gamma,i}$ of degree d_i. Let $d := (d_1, \ldots, d_n)$ and define $A\, t^d := (A_1\, t^{d_1}, \ldots, A_n\, t^{d_n})$ for any $A_1, \ldots, A_n \in \mathbb{A}$. For any $x \in \mathbb{A}^{m+1}$ and $t \in \mathbb{A}$, it then follows that

$$E_{\Gamma^{\mathrm{hom}}}(x\, t) = E_{\Gamma^{\mathrm{hom}}}(x)\, t^d. \tag{5.1}$$

We will denote by $P^{\mathrm{hom}} \in \mathbb{A}[x_1,\ldots,x_{m+1}]^n$ the polynomials with $E_{\Gamma^{\mathrm{hom}}}(x) = P^{\mathrm{hom}}(x)$ for all $x \in \mathbb{A}^{m+1}$.

5.2. Global bounds through homogenization

With the notation from the previous subsection, assume that $\mathbb{A} = \mathbb{K}$ with $\mathbb{K} = \mathbb{R}$ or $\mathbb{K} = \mathbb{C}$. Suppose that we wish to evaluate P at a point $x \in \mathbb{K}^m$ and let

$$t := \max(|x_1|, \ldots, |x_m|, 1), \qquad u := \frac{1}{t}, \qquad x^{\mathrm{hom}} := (u\, x_1, \ldots, u\, x_m, u). \tag{5.2}$$

Of course, we may simply evaluate Γ at x, which yields $P(x) = E_\Gamma(x)$. But thanks to (5.1), we also have the projective evaluation method

$$P(x) = E_{\Gamma^{\mathrm{hom}}}(x^{\mathrm{hom}})\, t^d.$$

The advantage here is that

$$|x^{\mathrm{hom}}|_\infty := \max(|x_1^{\mathrm{hom}}|, \ldots, |x_{m+1}^{\mathrm{hom}}|) \leqslant 1, \tag{5.3}$$

so we only evaluate Γ^{hom} at points in the poly-ball $\mathcal{B}(0,1)^{m+1}$. This allows us to apply the material from the previous sections concerning the evaluations of SLPs at points inside a fixed ball. Note that we may easily add the max function to SLPs since its computation is always exact. If $\mathbb{K} = \mathbb{C}$ then we apply max to the real and imaginary parts.

For instance, let $(\mathcal{B}(c_1, r_1), \ldots, \mathcal{B}(c_n, r_n)) \in \mathcal{B}(\mathbb{K}, \mathbb{R})^n$ be the evaluation of Γ^{hom} at $\mathcal{B}(0,1)^{m+1}$ using ball arithmetic and set $M_i := |c_i| + r_i$ for $i = 1, \ldots, n$. Then we have

$$|P_i(x)| \leqslant M_i\, t^{d_i} \tag{5.4}$$

for all $x \in \mathbb{K}^m$ and $i = 1, \ldots, n$. Consequently, we may compute upper bounds for $|P_1(x)|, \ldots, |P_n(x)|$ in time $O(\log d_1 + \cdots + \log d_n)$, which is typically much smaller than the length l of Γ.

We may use such global bounds for the construction of more efficient ball lifts, although the resulting error bounds may be less sharp. For this, we first evaluate the Jacobian matrix of $E_{\Gamma^{\mathrm{hom}}}$ at $\mathcal{B}(0,1)^{m+1}$, which yields bounds

$$\left| \frac{\partial E_{\Gamma^{\mathrm{hom}},i}}{\partial x_j}(x) \right| \leqslant M_{i,j}, \tag{5.5}$$

for all $i = 1, \ldots, n$, $j = 1, \ldots, m+1$, and $x \in \mathcal{B}(0,1)^{m+1}$. For any ball $\boldsymbol{x} = (\mathcal{B}(x_1, r_1), \ldots, \mathcal{B}(x_{m+1}, r_{m+1}))$ with $\mathcal{B}(x_i, r_i) \subseteq \mathcal{B}(0,1)$ for $i = 1, \ldots, m+1$, we then have

$$P_i^{\mathrm{hom}}(\boldsymbol{x}) \subseteq \mathcal{B}(E_{\Gamma^{\mathrm{hom}},i}(x), M_{i,1} r_1 + \cdots + M_{i,m+1} r_{m+1}). \tag{5.6}$$

This allows us to compute an enclosure of $P(\boldsymbol{x})$ using a single evaluation of Γ^{hom} at x plus $O(mn)$ extra operations (the quantity $O(mn)$ can be further reduced to $O(m+n)$ by replacing the $M_{i,j}$ by $\max(M_{1,j},\ldots,M_{n,j})$ or $\max(M_{i,1},\ldots,M_{i,m+1})$).

Similarly, if $\boldsymbol{x}=(\mathcal{B}(x_1,r_1),\ldots,\mathcal{B}(x_m,r_m))$ is any ball, then we define

$$t := \max(|x_1|+r_1,\ldots,|x_m|+r_m,1), \qquad u := \frac{1}{t}, \qquad \boldsymbol{x}^{\mathrm{hom}} := (u\,\boldsymbol{x}_1,\ldots,u\,\boldsymbol{x}_m,\mathcal{B}(u,0)). \tag{5.7}$$

For $i=1,\ldots,n$, the relations (5.1) and (5.6) now yield

$$P_i(\boldsymbol{x}) \subseteq \mathcal{B}(E_{\Gamma,i}(x),(M_{i,1}r_1+\cdots+M_{i,m}r_m)\,t^{d_i-1}). \tag{5.8}$$

(If $d_i=0$, then $E_{\Gamma,i}(x)$ is a constant, and we may use zero as the radius.) This allows us to compute an enclosure of $P(\boldsymbol{x})$ using one evaluation of Γ at x plus $O(mn+\log d_1+\cdots+\log d_n)$ extra operations in $\mathbb{K}$. Note that the homogenization Γ^{hom} was only required in order to compute the constants $M_{i,j}$, but not for evaluating the right-hand side of (5.8) at a specific $\boldsymbol{x}$.

5.3. Managing rounding errors

In the previous subsection, we assumed infinitely precise arithmetic over $\mathbb{R}$ or $\mathbb{C}$. Let us now show how to adapt this material to the case when $\mathbb{A}=\mathbb{K}_p$, where $\mathbb{K}_p=\mathbb{R}_p$ or $\mathbb{K}_p=\mathbb{C}_p$. As usual, assume $\epsilon := \max(\epsilon_{\mathbb{R}_p},\epsilon_{\mathbb{A}_p}) \leqslant 1/16$, and let us first show how to deal with rounding errors in the absence of overflows and underflows. We first replace (5.2) by

$$\begin{aligned} t_\circ &:= \circ[\max(|x_1|,\ldots,|x_m|,1)\,(1+3\epsilon)] \\ u_\circ &:= \circ[(1-3\epsilon)/t_\circ] \\ x_\circ^{\mathrm{hom}} &:= \circ[(u_\circ x_1,\ldots,u_\circ x_m,u_\circ)] \end{aligned}$$

and verify that

$$\begin{aligned} t_\circ &\geqslant \circ[\max(|x_1|,\ldots,|x_m|,1)]\,(1+3\epsilon)\,(1+\epsilon)^{-1} \geqslant t\,(1+3\epsilon)\,(1+\epsilon)^{-2} \geqslant t \\ t_\circ &\leqslant \circ[\max(|x_1|,\ldots,|x_m|,1)]\,(1+3\epsilon)\,(1-\epsilon)^{-1} \leqslant t\,(1+3\epsilon)\,(1-\epsilon)^{-2} \leqslant t\,(1+\epsilon)^6 \\ u_\circ &\geqslant (1-3\epsilon)/(t_\circ(1+\epsilon)) \geqslant (1-3\epsilon)\,u/(1+\epsilon)^7 \geqslant u\,(1+\epsilon)^{-11} \\ u_\circ &\leqslant (1-3\epsilon)/(t_\circ(1-\epsilon)) \leqslant (1-3\epsilon)\,u/(1-\epsilon) \leqslant u\,(1-\epsilon). \\ |(x_\circ^{\mathrm{hom}})_i| &\leqslant |u_\circ x_i|\,(1-\epsilon)^{-1} \leqslant u\,(1-\epsilon)\,t\,(1-\epsilon)^{-1} = 1, \qquad (i=1,\ldots,m). \end{aligned}$$

We have

$$\begin{aligned} |x_\circ^{\mathrm{hom}}-x^{\mathrm{hom}}|_\infty &\leqslant \left|\frac{u_\circ}{u}x^{\mathrm{hom}}-x^{\mathrm{hom}}\right|_\infty + \left|x_\circ^{\mathrm{hom}}-\frac{u_\circ}{u}x^{\mathrm{hom}}\right|_\infty \\ &\leqslant \left|\frac{u_\circ}{u}-1\right||x^{\mathrm{hom}}|_\infty + |x_\circ^{\mathrm{hom}}|_\infty\,\epsilon \\ &\leqslant \left|\frac{u_\circ}{u}-1\right|+\epsilon \\ &\leqslant 12\,\epsilon. \end{aligned}$$

Let $i\in\{1,\ldots,n\}$. Evaluating Γ^{hom} at $(\boldsymbol{U},\overset{(m+1)\times}{\ldots},\boldsymbol{U})$, for the matryoshka $\boldsymbol{U} := \mathcal{B}(0,1,12\,\epsilon)$, by Proposition 3.6 for instance, we may compute a $\Delta\in\mathbb{R}_p^n$ with

$$|P_i^{\mathrm{hom}}(x_\circ^{\mathrm{hom}})-P_i^{\mathrm{hom}}(x^{\mathrm{hom}})| \leqslant \Delta_i$$

for all $x\in\mathbb{K}_p^m$, as well as

$$|\circ[E_{\Gamma^{\mathrm{hom}},i}(x)]-P_i^{\mathrm{hom}}(x)| \leqslant \Delta_i$$

for all $x \in \mathbb{K}_p^{m+1}$ with $|x|_\infty \leqslant 1$. It follows that

$$\begin{aligned}
|\circ[E_{\Gamma^{\hom},i}(x_\circ^{\hom})]\, t_\circ^{d_i} - P_i(x)| &\leqslant |P_i^{\hom}(x_\circ^{\hom})\, t_\circ^{d_i} - P_i^{\hom}(x^{\hom})\, t^{d_i}| + \Delta_i t_\circ^{d_i} \\
&\leqslant \left|P_i(x)\left(\left(\tfrac{t_\circ}{t}\right)^{d_i} - 1\right)\right| + 2\Delta_i t_\circ^{d_i} \\
&\leqslant |P_i(x)|\,((1+\epsilon)^{6d_i} - 1) + 2\Delta_i t_\circ^{d_i}.
\end{aligned}$$

Setting

$$P_{i,\circ}(x) := \circ[E_{\Gamma^{\hom},i}(x_\circ^{\hom})\, t_\circ^{d_i}],$$

we then get

$$\begin{aligned}
|P_{i,\circ}(x) - P_i(x)| &\leqslant |\circ[E_{\Gamma^{\hom},i}(x_\circ^{\hom})]\, t_\circ^{d_i} - P_i(x)| + |\circ[E_{\Gamma^{\hom},i}(x_\circ^{\hom})]\, t_\circ^{d_i}|((1+\epsilon)^{d_i+1} - 1) \\
&\leqslant |\circ[E_{\Gamma^{\hom},i}(x_\circ^{\hom})]\, t_\circ^{d_i} - P_i(x)|\,(1+\epsilon)^{d_i+1} + |P_i(x)|((1+\epsilon)^{d_i+1} - 1) \\
&\leqslant (|P_i(x)|\,((1+\epsilon)^{6d_i} - 1) + 2\Delta_i t_\circ^{d_i})\,(1+\epsilon)^{d_i+1} + |P_i(x)|((1+\epsilon)^{d_i+1} - 1) \\
&\leqslant |P_i(x)|\,((1+\epsilon)^{7d_i+1} - 1) + 2\Delta_i t_\circ^{d_i}(1+\epsilon)^{d_i+1}. \qquad (5.9)
\end{aligned}$$

Note that for all $a \geqslant 2$ and $\epsilon \leqslant a^{-2}/2$ we have

$$a\log(1+\epsilon) \leqslant \log(1+a^2\epsilon),$$

whence $(1+\epsilon)^a - 1 \leqslant a^2\epsilon$. Consequently, provided that $\epsilon < (8d_i)^{-2}/2$, the bound (5.9) simplifies into

$$|P_{i,\circ}(x) - P_i(x)| \leqslant (8d_i)^2\epsilon|P_{i,\circ}(x)| + 3\Delta_i t_\circ^{d_i}. \qquad (5.10)$$

This yields an easy way to compute error bound

$$|P_{i,\circ}(x) - P_i(x)| \leqslant \circ[((8d_i)^2\epsilon|P_{i,\circ}(x)| + 3\Delta_i t_\circ^{d_i})\,(1+\epsilon)^{d_i+3}] \qquad (5.11)$$

for the approximate numeric evaluation of Γ at any point $x \in \mathbb{K}_p^m$.

Let us now turn to the bound (5.4). Using traditional ball arithmetic (formulas (2.5)) over $\mathcal{B}(\mathbb{K}_p, \mathbb{R}_p)$, we may still compute $M_i \in \mathbb{R}_p$ with $|P_i(x)| \leqslant M_i$ for all $x \in \mathbb{R}_p^m$ with $|x|_\infty \leqslant 1$. Then we simply have

$$|P_i(x)| \leqslant M_i t_\circ^{d_i}$$

for all $x \in \mathbb{K}_p^m$, since $t \leqslant t_\circ$. In a similar way, bounds $M_{i,j}$ that satisfy (5.5) can be computed using traditional ball arithmetic. For any ball $\boldsymbol{x} = (\mathcal{B}(x_1, r_1), \ldots, \mathcal{B}(x_m, r_m)) \in \mathcal{B}(\mathbb{K}_p, \mathbb{R}_p)^m$ and with we notation from (5.7), the enclosure (5.8) still holds. Combining

$$P_i(\boldsymbol{x}) \subseteq \mathcal{B}(E_{\Gamma,i}(x), (M_{i,1}r_1 + \cdots + M_{i,m}r_m)\max(|\boldsymbol{x}_1|_\circ, \ldots, |\boldsymbol{x}_m|_\circ, 1)^{d_i-1}).$$

with (5.10), this yields

$$P_i(\boldsymbol{x}) \subseteq \mathcal{B}(\circ[E_{\Gamma,i}(x)], (M_{i,1}r_1 + \cdots + M_{i,m}r_m)\, t_\circ^{d_i-1} + (8d_i)^2\epsilon|P_{i,\circ}(x)| + 3\Delta_i t_\circ^{d_i}).$$

Hence

$$P_i(\boldsymbol{x}) \subseteq \mathcal{B}(\circ[E_{\Gamma,i}(x)], \rho_i),$$

where

$$\rho_i := \circ[((M_{i,1}r_1 + \cdots + M_{i,m}r_m)\, t_\circ^{d_i-1} + (8d_i)^2\epsilon|P_{i,\circ}(x)| + 3\Delta_i t_\circ^{d_i})\,(1+\epsilon)^{2m+d_i+10}]. \qquad (5.12)$$

This yields a ball lift for P that is almost as efficient to compute as a mere numeric evaluation of E_Γ.

Let us finally analyze how to deal with underflows and overflows. Let $\eta := \eta_{\mathbb{R}_p} \in \mathbb{R}_p$ be such that (2.8) holds. As long as the computation of $u_\circ$ does not underflow, the inequality $|x_\circ^{\hom} - x^{\hom}|_\infty \leqslant 12\,\epsilon$ remains valid, even in the case of overflows (provided that $\epsilon \geqslant 2^{-p}$). Consequently, the relation (5.10) still holds, so it suffices to replace (5.11) by

$$|P_{i,\circ}(x) - P_i(x)| \;\leqslant\; \circ\big[((8\,d_i)^2\,\epsilon\,|P_{i,\circ}(x)| + 3\,\Delta_i\,t_\circ^{d_i})\,(1+\epsilon)^{d_i+3} + \eta\big].$$

This indeed counters the possible underflow for the multiplication of $(8\,d_i)^2\,\epsilon$ with $|P_{i,\circ}(x)|$. Note that the relation trivially holds in the case when the right-hand side overflows, which happens in particular when the computation of $t_\circ$ overflows. Similarly, it suffices to replace (5.12) by

$$\rho_i \;:=\; \circ\big[((M_{i,1}\,r_1 + \cdots + M_{i,m}\,r_m)\,t_\circ^{d_i-1} + (8\,d_i)^2\,\epsilon\,|P_{i,\circ}(x)| + 3\,\Delta_i\,t_\circ^{d_i})\,(1+\epsilon)^{2m+d_i+10} + m\,\eta\big].$$

If the computation of $t_\circ$ does not overflow, but the computation of $u_\circ$ underflows, then the IEEE 754 norm implies that we must have $t_\circ \geqslant 0.35\,\Omega$, where Ω is the largest positive floating point number in $\mathbb{R}_p$ with $\Omega < \infty$. Consequently, the computation $3\,\Delta_i\,t_\circ^{d_i}$ overflows whenever $d_i > 0$, provided that we compute this product as Δ_i times $3\,t_\circ^{d_i}$. The adjusted bounds are therefore valid in general.

6. BOUNDING DERIVATIVES

For several applications, it is useful to not only compute bounds for the function f itself, but also for some of its iterated derivatives.

6.1. The univariate case

Let us first assume that $f : \mathcal{U} \to \mathbb{C}$ is an analytic function defined on an open subset $\mathcal{U}$ of $\mathbb{C}$. Given a derivation order $k \in \mathbb{N}$ and a ball $\boldsymbol{a} = \mathcal{B}(a, R) \subseteq \mathcal{U}$, our goal is to compute a bound for $\sup_{\boldsymbol{a} \circ\!\!-\!\!- z} |f^{(k)}(z)|$. If f is actually an explicit polynomial

$$f(z) = f_d\,z^d + \cdots + f_0$$

and $a = 0$, then a first option is to use the crude bound

$$\sup_{\mathcal{B}(0,R) \circ\!\!-\!\!- z} |f^{(k)}(z)| \;\leqslant\; \sum_{i=0}^{d-k} \frac{(i+k)!}{i!}\,|f_{i+k}|\,R^i.$$

Of course, the case where $a \neq 0$ may be reduced to the case where $a = 0$ via the change of variables $z \to z + a$.

Assume now that f is given through an SLP, which possibly involves other operations as $+, -, \cdot$, such as division. In that case, the above crude bound typically becomes suboptimal, and does not even apply if f is no longer a polynomial. A general alternative method to compute iterated derivatives is to use Cauchy's formula

$$f^{(k)}(z) = \frac{k!}{2\,\mathrm{i}\,\pi} \oint_{\mathcal{C}} \frac{f(u)}{(u-z)^{k+1}}\,\mathrm{d}u, \tag{6.1}$$

where $\mathcal{C} \subseteq \mathcal{U}$ is some circle around $z \in \mathcal{B}(a, R)$. Taking $\mathcal{C} := \mathcal{C}(a, R + r)$ to be the circle with center a and radius $R + r$ for some $r > 0$, this yields the bound

$$\sup_{\mathcal{B}(a,R) \circ\!\!-\!\!- z} |f^{(k)}(z)| \;\leqslant\; k!\,\frac{[f(\mathcal{B}(a, R+r))]}{r^k}, \tag{6.2}$$

where $\lceil \mathcal{B}(b,s) \rceil := |b| + s$ denotes an upper bound for $|z|$ where $z \in \mathcal{B}(b,s)$.

It is an interesting question how to choose r. One practical approach is to start with any $r > 0$ such that $\mathcal{B}(a, R+r) \subseteq \mathcal{U}$. Next we may compute both $\lceil f(\mathcal{B}(a,R+r)) \rceil$ and $\lceil f(\mathcal{B}(a,R+r')) \rceil$ for some other $r' \neq r$ and check which value provided a better bound. If, say $r' := r/2$, yields a better bound, then we may next try $r'' := r/4$. If the bound for $r/2$ is better than the one for $r/4$, we may continue with $r''' := r/(2\sqrt{2})$ and so on until we find a bound that suits us.

For simple explicit functions f, it is also instructive to investigate what is the optimal choice for r. For instance, if $a = 0$ and $f = z^d$ with $d > k$, then we have $\lceil f(\mathcal{B}(0,R)) \rceil := R^d$, so the bound (6.2) reduces to

$$\sup_{\mathcal{B}(0,R) \circ\!\!-\!\!\!- z} |f^{(k)}(z)| \leqslant k! \frac{(R+r)^d}{r^k}.$$

The right-hand side is minimal when

$$r = \frac{k}{d-k} R. \tag{6.3}$$

In general, we may consider the power series expansion at a

$$f(a+z) = f_0 + f_1 z + f_2 z^2 + \cdots.$$

For such a power series expansion and a fixed $r > 0$ with $\mathcal{B}(a, R+r) \subseteq \mathcal{U}$, let $d \in \mathbb{N}$ be largest such that $|f_d| (R+r)^d$ is maximal. We regard $d = d(R+r)$ as the "numerical degree" of f on the disk $\mathcal{B}(a, R+r))$. Then the relation (6.3) suggests that we should have $r \approx kR/(d(R+r) - k)$ for the optimal value of r.

6.2. Bounds for derivatives of multivariate functions

For higher dimensional generalizations, consider an analytic map $f : \mathcal{U} \to \mathbb{C}^n$ for some open set $\mathcal{U} \subseteq \mathbb{C}^m$. Let $|\cdot|_2$ be the standard Euclidean norm on $\mathbb{C}^d$ for any $d \in \mathbb{N}$. Given $a \in \mathbb{C}^m$ and $R \in (\mathbb{R}^>)^m$, we denote by

$$\mathcal{D}(a,R) := \mathcal{B}(a_1, R_1) \times \cdots \times \mathcal{B}(a_m, R_m)$$

the polydisk associated to the poly-ball $\mathcal{B}(a,R)$. For $k \in \mathbb{N}$ and assuming that $\mathcal{D}(a,R) \subseteq \mathcal{U}$, we denote the operator norm of the k-th derivative $D^k f$ of f on $\mathcal{D}(a,R)$ by

$$\|D^k f\|_{\mathcal{D}(a,R)} := \sup_{z \in \mathcal{D}(a,R)} \left(\max_{|u_1|_2 = \cdots = |u_k|_2 = 1} |(D^k f)(z)(u_1, \ldots, u_k)|_2 \right).$$

Here we recall that

$$(D^k f)(z)(h_1, \ldots, h_k) = \sum_{i_1=1}^{m} \cdots \sum_{i_k=1}^{m} \frac{\partial^k f}{\partial z_{i_1} \cdots \partial z_{i_k}}(z)\, h_{1,i_1} \cdots h_{k,i_k}.$$

Given $r \in (\mathbb{R}^>)^m$ and $C := C(a, R+r) := C(a_1, R_1 + r_1) \times \cdots \times C(a_m, r_m)$ with $\mathcal{D}(a, R+r) \subseteq \mathcal{U}$, we have the following m-dimensional generalization of (6.1):

$$\frac{\partial^{k_1 + \cdots + k_m} f}{\partial z_1^{k_1} \cdots \partial z_m^{k_m}}(z) := \frac{k_1! \cdots k_m!}{(2\mathrm{i}\pi)^m} \oiint_C \frac{f(u)}{(u_1 - z_1)^{k_1+1} \cdots (u_m - z_m)^{k_m+1}} \mathrm{d}u_1 \cdots \mathrm{d}u_m,$$

for any $z \in \mathcal{D}(a,R)$. For any $\mathcal{B}(a,R) \circ\!\!-\!\!\!- z$, it follows that

$$\left| \frac{\partial^{k_1 + \cdots + k_m} f}{\partial z_1^{k_1} \cdots \partial z_m^{k_m}}(z) \right|_2 \leqslant k_1! \cdots k_m! \frac{\lceil f(\mathcal{B}(a,R+r)) \rceil_2}{r_1^{k_1} \cdots r_m^{k_m}},$$

where $\lceil \mathcal{B}(b,s) \rceil_2$ stands for an upper bound for $|z|_2$ with $\mathcal{B}(b,s) \circ\!\!-\!\!- z$. Translated into operator norms, this yields

$$\begin{aligned} \|D^k f\|_{\mathcal{D}(a,R)} &\leqslant \lceil f(\mathcal{B}(a,R+r)) \rceil_2 \sum_{k_1+\cdots+k_m=k} \binom{k}{k_1,\ldots,k_m} \frac{k_1!\cdots k_m!}{r_1^{k_1}\cdots r_m^{k_m}} \\ &= k! \lceil f(\mathcal{B}(a,R+r)) \rceil_2 \sum_{k_1+\cdots+k_m=k} \frac{1}{r_1^{k_1}\cdots r_m^{k_m}}. \end{aligned}$$

A practical approach to find an r which approximately minimizes the right-hand side is similar to what we did in the univariate case: starting from a given r, we vary it in a dichotomic way until we reach an acceptably good approximation. This time, we need to vary the m individual components $r_1, \ldots, r_m$ of r in turn, which makes the approximation process roughly m times more expensive. Multivariate analogues of (6.3) are more complicated and we leave this issue for future work.

BIBLIOGRAPHY

[1] A. Ahlbäck, J. van der Hoeven, and G. Lecerf. JIL: a high performance library for straight-line programs. https://sourcesup.renater.fr/projects/jil, 2025.

[2] G. Alefeld and J. Herzberger. *Introduction to interval computation*. Academic Press, New York, 1983.

[3] W. Baur and V. Strassen. The complexity of partial derivatives. *Theor. Comput. Sci.*, 22:317–330, 1983.

[4] C. Beltrán and A. Leykin. Robust certified numerical homotopy tracking. *Found. Comput. Math.*, 13(2):253–295, 2013.

[5] P. Bürgisser, M. Clausen, and M. A. Shokrollahi. *Algebraic complexity theory*, volume 315 of *Grundlehren der Mathematischen Wissenschaften*. Springer-Verlag, 1997.

[6] T. Duff and K. Lee. Certified homotopy tracking using the Krawczyk method. In *Proceedings of the 2024 International Symposium on Symbolic and Algebraic Computation*, ISSAC '24, pages 274–282. New York, NY, USA, 2024. ACM.

[7] A. Guillemot and P. Lairez. Validated numerics for algebraic path tracking. In *Proc. ISSAC 2024*, pages 36–45. ACM, 2024.

[8] J. van der Hoeven. Ball arithmetic. In A. Beckmann, C. Gaßner, and B. Löwe, editors, *Logical approaches to Barriers in Computing and Complexity*, number 6 in Preprint-Reihe Mathematik, pages 179–208. Ernst-Moritz-Arndt-Universität Greifswald, February 2010. International Workshop.

[9] J. van der Hoeven. Reliable homotopy continuation. Technical Report, HAL, 2011. https://hal.archives-ouvertes.fr/hal-00589948.

[10] J. van der Hoeven and G. Lecerf. Evaluating straight-line programs over balls. In *23nd IEEE Symposium on Computer Arithmetic (ARITH)*, pages 142–149. 2016. Extended preprint version with Appendix at https://hal.archives-ouvertes.fr/hal-01225979.

[11] J. van der Hoeven and G. Lecerf. Towards a library for straight-line programs. Technical Report, HAL, 2025. https://hal.science/hal-05075591.

[12] J. van der Hoeven, G. Lecerf, B. Mourrain et al. Mathemagix. 2002. http://www.mathemagix.org.

[13] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. Applied interval analysis. *Springer, London*, 2001.

[14] F. Johansson. Arb: a C library for ball arithmetic. *ACM Commun. Comput. Algebra*, 47(3/4):166–169, 2014.

[15] R. B. Kearfott. An interval step control for continuation methods. *SIAM J. Numer. Anal.*, 31(3):892–914, 1994.

[16] U. W. Kulisch. Computer arithmetic and validity. Theory, implementations and applications. *Studies in Mathematics.*, (33), 2008.

[17] R. E. Moore. *Interval analysis*. Prentice Hall, Englewood Cliff, 1966.

[18] R. E. Moore, R.B. Kearfott, and M. J. Cloud. *Introduction to interval analysis*. SIAM Press, 2009.

[19] A. Neumaier. *Interval methods for systems of equations*. Cambridge University Press, Cambridge, 1990.

[20] S. M. Rump. Fast and parallel interval arithmetic. *BIT*, 39(3):534–554, 1999.

[21] S. M. Rump. Verification methods: rigorous results using floating-point arithmetic. *Acta Numer*, 19:287–449, 2010.