Propagation of Roundoff Errors in Finite Precision Computations: a Semantics Approach[1]

Matthieu Martel

CEA - Recherche Technologique
LIST-DTSI-SLA
CEA F91191 Gif-Sur-Yvette Cedex, France
e-mail : mmartel@cea.fr

Abstract. We introduce a concrete semantics for floating-point operations which describes the propagation of roundoff errors throughout a computation. This semantics is used to assert the correctness of an abstract interpretation which can be straightforwardly derived from it.
In our model, every elementary operation introduces a new first order error term, which is later combined with other error terms, yielding higher order error terms. The semantics is parameterized by the maximal order of error to be examined and verifies whether higher order errors actually are negligible. We consider also coarser semantics computing the contribution, to the final error, of the errors due to some intermediate computations.

Keywords: Numerical Precision, Abstract Interpretation, Floating-point Arithmetic, IEEE Standard 754.

1 Introduction

Often, the results of numerical programs are accepted with suspicion because of the approximate numbers used during the computations. In addition, it is often hard for a programmer to understand how precise a result is, or to understand the nature of the imprecision [6, 13]. In this article, we present the theoretical basis of an abstract interpreter which estimates the accuracy of a numerical result and which detects the elementary operations that introduce the most imprecision. The implementation of this tool is described in [9]. Compared with other approaches, we do not attempt to compute a better estimation of what a particular execution of a numerical program would return if the computer were using real number. Instead, we statically point out the places in the code which possibly introduce significant errors for a large set of executions.

From our knowledge, the propagation of roundoff errors and the introduction of new errors at each stage of a computation is a phenomenon which has almost never been studied in a semantics framework. Some dynamic stochastic methods have been proposed but they cannot guarantee the correctness of the estimation

[1] This work was supported by the RTD project IST-1999-20527 ”DAEDALUS” of the European FP5 programme.

in all cases [1, 2, 11, 16, 17]. Recently, Eric Goubault has proposed a static analysis based on abstract interpretation [3] which can compute the contribution of each first order error, due to the inaccuracy of one floating-point operation, to the final error associated with a result [7]. This new approach differs from the existing ones in that it not only attempts to estimate the accuracy of a result, but also provides indications on the source of the imprecision. It also differs from much other work in that it models the propagation of errors on initial data (sensitivity analysis) as well as the propagation of roundoff errors due to the intermediate floating-point computations (this can dominate the global error in some cases).

We develop a general concrete semantics $\mathcal{S}^{\mathcal{L}^*}$ for floating-point operations which explains the propagation of roundoff errors during a computation. Elementary operations introduce new first order error terms which, once combined, yield higher order error terms. $\mathcal{S}^{\mathcal{L}^*}$ models the roundoff error propagation in finite precision computations for error terms of any order and is based on IEEE Standard 754 for floating-point numbers [12]. By modelling the propagation of errors in the general case, $\mathcal{S}^{\mathcal{L}^*}$ contributes to the general understanding of this problem and provide a theoretical basis for many static analyses. In particular, $\mathcal{S}^{\mathcal{L}^*}$ can be straightforwardly adapted to define abstract interpretations generalizing the one of [7].

Next we propose some approximations of $\mathcal{S}^{\mathcal{L}^*}$. We show that for any integer n, $\mathcal{S}^{\mathcal{L}^*}$ can be approximated by another semantics $\mathcal{S}^{\mathcal{L}^n}$ which only computes the contribution to the global error of the error terms of order at most n, as well as the residual error, i.e. the global error due to the error terms of order higher than n. Approximations are proven correct by means of Galois connections. For example, $\mathcal{S}^{\mathcal{L}^1}$ computes the contribution to the global error of the first order errors. In addition, in contrast to [7], $\mathcal{S}^{\mathcal{L}^1}$ does verifie that the contribution to the global error of the error terms of order greater than one is negligible. Finally, we introduce coarser semantics which compute the contribution, to the global error in the result of a computation, of the errors introduced by some pieces of code in the program. These semantics provide less information than the most general ones but use non-standard values of smaller size. In practice, this allows the user to first detect which functions of a program introduce most errors and, next, to examine in more detail which part of an imprecise function increases the error [9]. We introduce a partial order relation $\dot{\subseteq}$ on the set of the partitions of the program points and we show that, for two partitions $\mathcal{J}_1 \dot{\subseteq} \mathcal{J}_2$, the semantics based on $\mathcal{J}_1$ approximates the semantics based on $\mathcal{J}_2$.

Section 2 gives an overview of the techniques developed in this article and Section 3 briefly describes some aspects of IEEE Standard 754. In Section 4 and Section 5, we introduce the semantics detailing the contribution, to the global error, of the error terms of any order and of order at most n, respectively. In Section 6, we introduce a coarser semantics which computes the contribution of the error introduced in pieces of code partitioning the program. We show that the semantics based on certain partitions are comparable. Section 7 concludes.

2 Overview

In this section, we illustrate how the propagation of roundoff errors is treated in our model using as an example a simple computation involving two values $a_{\mathbb{F}} = 621.3$ and $b_{\mathbb{F}} = 1.287$. For the sake of simplicity, we assume that $a_{\mathbb{F}}$ and $b_{\mathbb{F}}$ belong to a simplified set of floating-point numbers composed of a mantissa of four digits written in base 10. We assume that initial errors are attached to $a_{\mathbb{F}}$ and $b_{\mathbb{F}}$, and we write $a = 621.3+0.05\varepsilon_1$ and $b = 1.287+0.0005\varepsilon_2$ to indicate that the value of the initial error on $a_{\mathbb{F}}$ (resp. $b_{\mathbb{F}}$) is 0.05 (resp. 0.0005). ε_1 and ε_2 are formal variables related to static program points. a and b are called *floating-point numbers with errors*. Let us focus on the product $a_{\mathbb{F}} \times b_{\mathbb{F}}$ whose exact result is $a_{\mathbb{F}} \times b_{\mathbb{F}} = 799.6131$. This computation carried out with floating-point numbers with errors yields $a \times b = 799.6131 + 0.06435\varepsilon_1 + 0.31065\varepsilon_2 + 0.000025\varepsilon_1 \times \varepsilon_2$ The difference between $\alpha_{\mathbb{F}} \times b_{\mathbb{F}}$ and 621.35×1.2875 is 0.375025 and this error stems from the fact that the initial error on a (resp. b) was multiplied by b (resp. a) and that a second order error corresponding to the multiplication of both errors was introduced. So, at the end of the computation, the contribution to the global error of the initial error on a (resp. b) is 0.06435 (resp. 0.31065) and corresponds to the coefficient attached to the formal variable ε_1 (resp. ε_2). The contribution of the second order error due to the initial errors on both a and b is given by the term $0.000025\varepsilon_1 \times \varepsilon_2$ which we will write as $0.000025\varepsilon_{12}$ in the following. Finally, the number 799.6131 has too many digits to be representable in our floating-point number system. Since IEEE Standard 754 ensures that elementary operations are correctly rounded, we may claim that the floating-point number computed by our machine is 799.6 and that a new error term $0.0131\varepsilon_\times$ is introduced by the multiplication. To sum up, we have

$$a \times b = 799.6 + 0.06435\varepsilon_1 + 0.31065\varepsilon_2 + 0.000025\varepsilon_{12} + 0.0131\varepsilon_\times$$

At first sight, one could believe that the precision of the computation mainly depends on the initial error on a since it is 100 times larger than the one on b. However the above result indicates that the final error is mainly due to the initial error on b. Hence, to improve the precision of the final result, one should first try to increase the precision on b (whenever possible). Note that, as illustrated by the above example and in contrast to most of existing methods [14, 16], we do not attempt to compute a better approximation of the real number that the program would output if the computer were using real numbers. Since we are interested in detecting the errors possibly introduced by floating-point numbers, we always work with the floating-point numbers used by the machine and we compute the errors attached to them.

3 Preliminary Definitions

3.1 IEEE Standard 754

IEEE Standard 754 was introduced in 1985 to harmonize the representation of floating-point numbers as well as the behavior of the elementary floating-point operations [6, 12]. This standard is now implemented in almost all modern processors and, consequently, it provides a precise semantics, used as a basis in this article, for the basic operations occurring in high-level programming languages. First of all, a *floating-point number* x in base β is defined by

$$x = s \cdot (d_0.d_1 \dots d_{p-1}) \cdot \beta^e = s \cdot m \cdot \beta^{e-p+1}$$

where $s \in \{-1, 1\}$ is the sign, $m = d_0 d_1 \dots d_{p-1}$ is the *mantissa* with digits $0 \leq d_i < \beta$, $0 \leq i \leq p-1$, p is the *precision* and e is the exponent, $e_{min} \leq e \leq e_{max}$. A floating-point number x is *normalized* whenever $d_0 \neq 0$. Normalization avoids multiple representations of the same number. IEEE Standard 754 specifies a few values for p, e_{min} and e_{max}. Simple precision numbers are defined by $\beta = 2$, $p = 23$, $e_{min} = -126$ and $e_{max} = +127$; Double precision numbers are defined by $\beta = 2$, $p = 52$, $e_{min} = -1022$ and $e_{max} = +1023$. $\beta = 2$ is the only allowed basis but slight variants also are defined by IEEE Standard 854 which, for instances, allows $\beta = 2$ or $\beta = 10$. IEEE Standard 754, also introduces *denormalized numbers* which are floating-point numbers with $d_0 = d_1 = \dots = d_k = 0$, $k < p-1$ and $e = e_{min}$. Denormalized numbers make underflow gradual [6]. Finally, the following special values also are defined:

- NaN (Not a Number) resulting from an illegal operation,
- the values $\pm\infty$ corresponding to overflows,
- the values $+0$ and -0 (signed zeros[2]).

We do not consider the extended simple and extended double formats, also defined by IEEE Standard 754, whose implementations are machine-dependent. In the rest of this paper, the notation $\mathbb{F}$ indifferently refers to the set of simple or double precision numbers, since our assumptions conform to both types. $\mathbb{R}$ denotes the set of real numbers.

IEEE 754 Standard defines four rounding modes for elementary operations between floating-point numbers. These modes are towards $-\infty$, towards $+\infty$, towards zero and to the nearest. We write them $\circ_{-\infty}$, $\circ_{+\infty}$, $\circ_0$ and $\circ_\sim$ respectively. Let $\mathbb{R}$ denote the set of real numbers and let $\uparrow_\circ\, : \ \mathbb{R} \to \mathbb{F}$ be the function which returns the roundoff of a real number following the rounding mode $\circ \in \{\circ_{-\infty}, \circ_{+\infty}, \circ_0, \circ_\sim\}$. IEEE standard 754 specifies the behavior of the elementary operations $\diamond \in \{+,\ -,\ \times,\ \div\}$ between floating-point numbers by

$$f_1 \diamond_{\mathbb{F},\circ} f_2 \ = \uparrow_\circ (f_1 \diamond_{\mathbb{R}} f_2) \tag{1}$$

IEEE Standard 754 also specifies how the square root function must be rounded in a similar way to Equation (1) but does not specify, for theoretical reasons [15], the roundoff of transcendental functions like sin, log, etc.

In this article, we also use the function $\downarrow_\circ$: $\mathbb{R} \to \mathbb{R}$ which returns the error ω due to using $\uparrow_\circ (r)$ instead of f. We have $\downarrow_\circ (r) = r - \uparrow_\circ (r)$.

Let us remark that the elementary operations are total functions on $\mathbb{F}$, i.e. that the result of operations involving special values are specified. For instance,

[2] Remark that 0 is neither a normalized nor denormalized number.

$1 \div +\infty = 0$, $+\infty \times 0 = \text{NaN}$, etc. [6,10]. However, for the sake of simplicity, we do not consider these special values, assuming that any operation has correct operands and does not return an overflow or a NaN.

3.2 Standard Semantics

To study the propagation of errors along a sequence of computations, we consider arithmetic expressions annotated by static labels ℓ, ℓ_1, ℓ_2, etc. and generated by the grammar of Equation (2).

$$a^\ell ::= r^\ell \mid a_0^{\ell_0} +^\ell a_1^{\ell_1} \mid a_0^{\ell_0} -^\ell a_1^{\ell_1} \mid a_0^{\ell_0} \times^\ell a_1^{\ell_1} \mid a_0^{\ell_0} \div^\ell a_1^{\ell_1} \mid F^\ell(a_0^{\ell_0}) \qquad (2)$$

r denotes a value in the domain of floating-point numbers with errors and F denotes a transcendental function $\sqrt{}$, sin, etc. For any term generated by the above grammar, a unique label is attached to each sub-expression. These labels, which correspond to the nodes of the syntactic tree, are used to identify the errors introduced during a computation by an initial datum or a given operator. For instance, in the expression $r_0^{\ell_0} +^\ell r_1^{\ell_1}$ the initial errors corresponding to r_0 and r_1 are attached to the formal variables $\boldsymbol{\varepsilon}_{\ell_0}$ and $\boldsymbol{\varepsilon}_{\ell_1}$ and the new error introduced by the addition during the computation is attached to $\boldsymbol{\varepsilon}_\ell$.

In the remainder of this article, the set of all the labels occurring in a program is denoted $\mathcal{L}$. We use the small step operational semantics defined by the reduction rules below, where $\diamond \in \{+, -, \times, \div\}$. These rules correspond to a left to right evaluation strategy of the expressions.

$$\frac{a_0^{\ell_0} \to a_2^{\ell_2}}{a_0^{\ell_0} \diamond^\ell a_1^{\ell_1} \to a_2^{\ell_2} \diamond^\ell a_1^{\ell_1}} \qquad \frac{a_1^{\ell_1} \to a_2^{\ell_2}}{r_0^{\ell_0} \diamond^\ell a_1^{\ell_1} \to r_0^{\ell_0} \diamond^\ell a_2^{\ell_2}}$$

$$\frac{r = r_0 \diamond^\ell r_1}{r_0^{\ell_0} \diamond^\ell r_1^{\ell_1} \to r^\ell} \qquad \frac{a_0^{\ell_0} \to a_1^{\ell_1}}{F^\ell(a_0^{\ell_0}) \to F^\ell(a_1^{\ell_1})} \qquad \frac{r = F^\ell(r_0)}{F^\ell(r_0^{\ell_0}) \to r^\ell}$$

In the following, we introduce various domains for the values r and we specify various implementations of the operators $\diamond$. We only deal with arithmetic expressions because the semantics of the rest of the language (which is detailed in [8]) presents little interest. The only particularity concerns loops and conditionals, when the result of a comparison between floating-point numbers differs from the same comparison in $\mathbb{R}$. In this case, the semantics in $\mathbb{F}$ and $\mathbb{R}$ lead to the execution of different pieces of code. Our semantics mimics what the computer does and follows the execution path resulting from the evaluation of the test in $\mathbb{F}$. However, the errors cannot be computed any longer.

Labels are only attached to the arithmetic expressions because no roundoff error is introduced elsewhere. A label ℓ is related to the syntactic occurrence of an operator. If an arithmetic operation $\diamond^\ell$ is executed many times then the coefficient attached to $\boldsymbol{\varepsilon}_\ell$ denotes the sum of the roundoff errors introduced by each instance of $\diamond$. For instance if the assignment $x = r_1^{\ell_1} \diamond^\ell r_2^{\ell_2}$ is carried out inside a loop then, in the floating-point number with error denoting the value of x after n iterations, the coefficient attached to $\boldsymbol{\varepsilon}_\ell$ is the sum of the roundoff errors introduced by $\diamond^\ell$ during the n first iterations.

4 Floating-point numbers with errors

In this section, we define the general semantics $\mathcal{S}^{\mathcal{L}^*}$ of floating-point numbers with errors. $\mathcal{S}^{\mathcal{L}^*}$ computes the errors of any order made during a computation. Intuitively, a number r^ℓ occurring at point ℓ and corresponding to an initial datum r is represented by the floating-point number with errors $r^{\ell\mathcal{L}^*} = (f\boldsymbol{\varepsilon} + \omega^\ell \boldsymbol{\varepsilon}_\ell) \in \mathbb{R}^{\mathcal{L}^*}$ where $f = \uparrow_\circ r$ is the floating-point number approximating r and $\omega^\ell = \downarrow_\circ r$. The functions $\uparrow_\circ$ and $\downarrow_\circ$ are defined in Section 3.1. f and ω^ℓ are written as coefficients of a formal series and $\boldsymbol{\varepsilon}$ and $\boldsymbol{\varepsilon}_\ell$ are formal variables related to the value f known by the computer and the error between r and f.

A number r occurring at point ℓ_0 and corresponding to the result of the evaluation of an expression $a_0^{\ell_0}$ is represented by the series

$$r^{\ell_0\mathcal{L}^*} = f\boldsymbol{\varepsilon} + \sum_{u \in \overline{\mathcal{L}^+}} \omega^u \boldsymbol{\varepsilon}_u \qquad (3)$$

where $\mathcal{L}$ is a set containing all the labels occurring in $a_0^{\ell_0}$ and $\overline{\mathcal{L}^+}$ is a subset of the words on the alphabet $\mathcal{L}$. $\overline{\mathcal{L}^+}$ is formally defined further in this Section. In Equation (3), f is the floating-point number approximating r and is always attached to the formal variable $\boldsymbol{\varepsilon}$. Let ℓ be a word made of one character. In the formal series $\sum_{u \in \overline{\mathcal{L}^+}} \omega^u \boldsymbol{\varepsilon}_u$, $\omega^\ell \boldsymbol{\varepsilon}_\ell$ denotes the contribution to the global error of the first order error introduced by the computation of the operation labelled ℓ during the evaluation of $a_0^{\ell_0}$. $\omega^\ell \in \mathbb{R}$ is the scalar value of this error term and $\boldsymbol{\varepsilon}_\ell$ is a formal variable. For a given word $u = \ell_1\ell_2 \ldots \ell_n$ such that $n \geq 2$, $\boldsymbol{\varepsilon}_u$ denotes the contribution to the global error of the n^{th} order error due to the combination of the errors made at points $\ell_1 \ldots \ell_n$. For instance, let us consider the multiplication at point ℓ_3 of two initial data $r_1^{\ell_1} = (f_1\boldsymbol{\varepsilon} + \omega_1^{\ell_1}\boldsymbol{\varepsilon}_{\ell_1})$ and $r_2^{\ell_2} = (f_2\boldsymbol{\varepsilon} + \omega_2^{\ell_2}\boldsymbol{\varepsilon}_{\ell_2})$.

$$r_1^{\ell_1} \times^{\ell_3} r_2^{\ell_2} = \uparrow_\circ (f_1 f_2)\boldsymbol{\varepsilon} + f_2\omega_1^{\ell_1}\boldsymbol{\varepsilon}_{\ell_1} + f_1\omega_2^{\ell_2}\boldsymbol{\varepsilon}_{\ell_2} + \omega_1^{\ell_1}\omega_2^{\ell_2}\boldsymbol{\varepsilon}_{\ell_1\ell_2} + \downarrow_\circ (f_1 f_2)\boldsymbol{\varepsilon}_{\ell_3} \quad (4)$$

As shown in Equation (4), the floating-point number computed by this multiplication is $\uparrow_\circ (f_1 f_2)$. The initial first order errors $\omega_1^{\ell_1}\boldsymbol{\varepsilon}_{\ell_1}$ and $\omega_2^{\ell_2}\boldsymbol{\varepsilon}_{\ell_2}$ are multiplied by f_2 and f_1 respectively. In addition, the multiplication introduces a new first order error $\downarrow_\circ (f_1 f_2)\boldsymbol{\varepsilon}_{\ell_3}$, due to the approximation made by using the floating-point number $\uparrow_\circ (f_1 f_2)$ instead of the real number $f_1 f_2$. Finally, this operation also introduces an error whose weight is $\omega_1^{\ell_1}\omega_2^{\ell_2}$ and which is a second order error. We attach this coefficient to the formal variable $\boldsymbol{\varepsilon}_{\ell_1\ell_2}$ denoting the contribution to the global error of the second order error in ℓ_1 and ℓ_2.

From a formal point of view, let $\mathcal{L}^*$ denote the set of words of finite length on the alphabet $\mathcal{L}$. ϵ denotes the empty word, $|u|$ denotes the size of the word u and $u.v$ denotes the concatenation of the words u and v. We introduce the equivalence relation $\sim$ which identifies the words made of the same letters. $\sim$ makes concatenation commutative.

Definition 1 $\sim\ \subseteq \mathcal{L}^* \times \mathcal{L}^*$ *is the greatest equivalence relation* R *such that* $u\ R\ v$ *implies* $u = \ell.u'$, $v = v'.\ell.v''$ *and* $u'\ R\ v'.v''$.

Let $\overline{\mathcal{L}^*}$ be the quotient set $\mathcal{L}^*/\sim$. An equivalence class in $\overline{\mathcal{L}^*}$ is denoted by the smallest element u of the class w.r.t. the lexicographical order. $\overline{\mathcal{L}^+}$ denotes the set $\overline{\mathcal{L}^*} \setminus \{\epsilon\}$. For any word $u \in \overline{\mathcal{L}^+}$, the formal variable $\boldsymbol{\varepsilon}_u$ is related to a n^{th} order error whenever $n = |u|$. $\boldsymbol{\varepsilon}_\epsilon = \boldsymbol{\varepsilon}$ is related to the floating-point number f known by the computer instead of the real value. In this article, the symbols f and ω^ϵ are used indifferently to denote the coefficient of the variable $\boldsymbol{\varepsilon}_\epsilon$.

Let $\mathcal{F}(\mathcal{D}, \overline{\mathcal{L}^*}) = \{\sum_{u \in \overline{\mathcal{L}^*}} \omega^u \boldsymbol{\varepsilon}_u\ :\ \forall u,\ \omega^u \in \mathcal{D}\}$ be the domain of the formal series (ordered componentwise) whose formal variables are annotated by elements of $\overline{\mathcal{L}^*}$ and which coefficients ω^u belong to $\mathcal{D}$. The semantics $\mathcal{S}^{\mathcal{L}^*}$ uses the domain $\mathbb{R}^{\mathcal{L}^*} = \mathcal{F}(\mathbb{R}, \overline{\mathcal{L}^*})$ for floating-point numbers with errors. The elementary operations on $\mathbb{R}^{\mathcal{L}^*}$ are defined in Figure 1. $\mathcal{W}$ denotes a set of words on an alphabet containing $\mathcal{L}$ and $\mathcal{W}^+$ denotes $\mathcal{W} \setminus \{\epsilon\}$. For now, $\mathcal{W} = \overline{\mathcal{L}^*}$ but some alternatives are treated later in this article.

$$r_1 +^{\ell_i} r_2 \stackrel{\text{def}}{=} \uparrow_\circ (f_1 + f_2)\boldsymbol{\varepsilon} + \sum_{u \in \mathcal{W}^+} (\omega_1^u + \omega_2^u)\boldsymbol{\varepsilon}_u + \downarrow_\circ (f_1 + f_2)\boldsymbol{\varepsilon}_{\ell_i} \tag{5}$$

$$r_1 -^{\ell_i} r_2 \stackrel{\text{def}}{=} \uparrow_\circ (f_1 - f_2)\boldsymbol{\varepsilon} + \sum_{u \in \mathcal{W}^+} (\omega_1^u - \omega_2^u)\boldsymbol{\varepsilon}_u + \downarrow_\circ (f_1 - f_2)\boldsymbol{\varepsilon}_{\ell_i} \tag{6}$$

$$r_1 \times^{\ell_i} r_2 \stackrel{\text{def}}{=} \uparrow_\circ (f_1 f_2)\boldsymbol{\varepsilon} + \sum_{\substack{u \in \mathcal{W} \\ v \in \mathcal{W} \\ |u.v| > 0}} \omega_1^u \omega_2^v \boldsymbol{\varepsilon}_{u.v} + \downarrow_\circ (f_1 f_2)\boldsymbol{\varepsilon}_{\ell_i} \tag{7}$$

$$(r_1)^{-1^{\ell_i}} \stackrel{\text{def}}{=} \uparrow_\circ (f_1^{-1})\boldsymbol{\varepsilon} + \frac{1}{f_1} \sum_{n \geq 1} (-1)^n \left(\sum_{u \in \mathcal{W}^+} \frac{\omega^u}{f_1} \boldsymbol{\varepsilon}_u \right)^n + \downarrow_\circ (f_1^{-1})\boldsymbol{\varepsilon}_{\ell_i} \tag{8}$$

$$r_1 \div^{\ell_i} r_2 \stackrel{\text{def}}{=} r_1 \times^{\ell_i} (r_2)^{-1^{\ell_i}} \tag{9}$$

Fig. 1. Elementary operations for the semantics $\mathcal{S}^{\mathcal{L}^*}$.

In Figure 1, the formal series $\sum_{u \in \overline{\mathcal{L}^*}} \omega^u \boldsymbol{\varepsilon}_u$ related to the result of an operation $\diamond^{\ell_i}$ contains the combination of the errors on the operands as well as a new error term $\downarrow_\circ (f_1 \diamond_{\mathbb{R}} f_2)\boldsymbol{\varepsilon}_{\ell_i}$ corresponding to the error introduced by the operation $\diamond_{\mathbb{F}}$ occurring at point ℓ_i. The rules for addition and subtraction are natural. The elementary errors are added or subtracted componentwise in the formal series and the new error due to point ℓ_i corresponds to the roundoff of the result. Multiplication requires more care because it introduces higher-order errors due to the multiplication of the elementary errors. Higher-order errors appear when multiplying error terms. For instance, for $\ell_1, \ell_2 \in \mathcal{L}$, and for first order errors $\omega_1^{\ell_1}\boldsymbol{\varepsilon}_{\ell_1}$ and $\omega_2^{\ell_2}\boldsymbol{\varepsilon}_{\ell_2}$, the operation $\omega_1^{\ell_1}\boldsymbol{\varepsilon}_{\ell_1} \times \omega_2^{\ell_2}\boldsymbol{\varepsilon}_{\ell_2}$ introduces a second-order error term written $(\omega_1^{\ell_1} \times \omega_2^{\ell_2})\varepsilon_{\ell_1\ell_2}$. The formal series resulting from the computation of $r_1^{-1^{\ell_i}}$ is obtained by means of a power series development. Note that, since a power series is only defined as long as it is convergent, the above technique cannot be used for any value of r_1. In the case of the inverse function, the convergence disc of the series $\sum_{n\geq 0}(-1)^n x^n = (1+x)^{-1}$ has radius $\rho = 1$. So, in Equation (8), we require $-1 < \sum_{u\in\mathcal{W}^+} \frac{\omega^u}{f_1} < 1$. This constraint means that Equation (8) is correct as long as the absolute value of the sum of the elementary errors is less than the related floating-point number.

For an expression $a_0^{\ell_0}$ such that $\mathrm{Lab}(a_0^{\ell_0}) \subseteq \mathcal{L}$, the semantics $\mathcal{S}^{\mathcal{L}^*}$ is defined by the domain $\mathbb{R}^{\mathcal{L}^*}$ for values and by the reduction rules $\rightarrow^{\mathcal{L}}$ obtained by substituting the operators on $\mathbb{R}^{\mathcal{L}^*}$ to the operators $\diamond$ in the reduction rules of Section 3.

Concerning the correctness of the operations defined by equations (5) to (9), it is a trivial matter to verify that both sides of these equations denote the same quantity, i.e. for any operator $\diamond$ and any numbers $r_1 = \sum_{u\in\mathcal{W}} \omega_1^u \varepsilon_u$, $r_2 = \sum_{u\in\mathcal{W}} \omega_2^u \varepsilon_u$ and $r = r_1 \diamond^{\ell_i} r_2 = \sum_{u\in\mathcal{W}} \omega^u \varepsilon_u$, we have

$$\sum_{u\in\mathcal{W}} \omega^u = \left(\sum_{u\in\mathcal{W}} \omega_1^u\right) \diamond \left(\sum_{u\in\mathcal{W}} \omega_2^u\right) \tag{10}$$

However, this is too weak a correctness criterion since it does not examine whether a given error term is correctly propagated during a computation. For example, Equation (11) incorrectly models the propagation of errors since it inverts the errors attached to ε_{ℓ_1} and ε_{ℓ_2}.

$$(f_1\varepsilon + \omega^{\ell_1}\varepsilon_{\ell_1}) +^{\ell_i} (f_2\varepsilon + \omega^{\ell_2}\varepsilon_{\ell_2}) \overset{\text{bad!}}{=} \uparrow_\circ (f_1+f_2)\varepsilon + \omega^{\ell_1}\varepsilon_{\ell_2} + \omega^{\ell_2}\varepsilon_{\ell_1} + \downarrow_\circ (f_1+f_2)\varepsilon_{\ell_i} \tag{11}$$

Defining addition by a generalization of Equation (11) leads to an undesirable formula satisfying the correctness criterion of Equation (10). We aim at showing that no such confusion was made in our definitions of the elementary operations, mainly for multiplication and division. So we compare the variations of the terms occurring in both sides of the equations (5) to (9) as a finite number of the coefficients ω_1^u and ω_2^u are slightly modified. The variations of r_1 and r_2 are given by $\frac{\partial^n}{\partial\omega_{k_1}^{u_1}\ldots\omega_{k_n}^{u_n}}$ for a finite subset $\omega_{k_1}^{u_1}, \ldots \omega_{k_n}^{u_n}$ of the coefficients, with $k_i = 1$ or 2, $u_i \in \mathcal{W}^+$, $1 \leq i \leq n$. We first introduce Lemma 2 which deals with first order partial derivatives.

Lemma 2 *Let* $\diamond \in \{+, -, \times, \div\}$ *be a usual operator on formal series and let* $\diamond^{\ell_i} \in \{+^{\ell_i}, -^{\ell_i}, \times^{\ell_i}, \div^{\ell_i}\}$ *be the operators defined in Equations (5) to (9). For any* $r_1 = \sum_{u\in\mathcal{W}} \omega_1^u \varepsilon_u$, $r_2 = \sum_{u\in\mathcal{W}} \omega_2^u \varepsilon_u$ *and for any* $u_0 \in \mathcal{W}^+ \setminus \{\ell_i\}$ *we have*

$$\frac{\partial(r_1 \diamond r_2)}{\partial\omega_1^{u_0}} = \frac{\partial(r_1 \diamond^{\ell_i} r_2)}{\partial\omega_1^{u_0}} \quad \textit{and} \quad \frac{\partial(r_1 \diamond r_2)}{\partial\omega_2^{u_0}} = \frac{\partial(r_1 \diamond^{\ell_i} r_2)}{\partial\omega_2^{u_0}} \tag{12}$$

Lemma 2 ensures that the variation of a single error term in the input series is correctly managed in our model. Proofs are detailed in [8]. Proposition 3 below generalizes Lemma 2 to the variation of a finite number of coefficients.

Proposition 3 *Let $\diamond \in \{+, -, \times, \div\}$ denote an usual operator on formal series and let $\diamond^{\ell_i} \in \{+^{\ell_i}, -^{\ell_i}, \times^{\ell_i}, \div^{\ell_i}\}$ denote the operators defined in Equations (5) to (9). For any $r_1 = \sum_{u \in \mathcal{W}} \omega_1^u \boldsymbol{\varepsilon}_u$, $r_2 = \sum_{u \in \mathcal{W}} \omega_2^u \boldsymbol{\varepsilon}_u$ and for any $\omega_{k_1}^{u_1}, \dots \omega_{k_n}^{u_n}$, $k_i = 1$ or 2, $u_i \in \mathcal{W}^+ \setminus \{\ell_i\}$, $1 \le i \le n$, we have:*

$$\frac{\partial^n (r_1 \diamond r_2)}{\partial \omega_{k_1}^{u_1} \dots \omega_{k_n}^{u_n}} = \frac{\partial^n (r_1 \diamond^{\ell_i} r_2)}{\partial \omega_{k_1}^{u_1} \dots \omega_{k_n}^{u_n}} \tag{13}$$

As a conclusion, let us remark that the incorrect definition of addition given in Equation (11) satisfies neither Lemma 2, nor Proposition 3.

For transcendental functions, given a number $r^{\mathcal{L}^n} = \sum_{u \in \overline{\mathcal{L}^n}} \omega^u \boldsymbol{\varepsilon}_u$ and a function F, we aim at determining how a given error term ω^u related to $r^{\mathcal{L}^n}$ is modified in $F(r^{\mathcal{L}^n})$. This is done by means of power series developments, as for the inverse function of Equation (8). However, let us remark that the functions $\uparrow_\circ (F(x))$ and $\downarrow_\circ (F(x))$ must be used carefully, since IEEE Standard 754 only specifies how to roundoff the elementary operations $+$, $-$, $\times$ and $\div$ and the square root function. For another function $F \in \{\sin, \exp, \dots\}$ machine-dependent criteria must be considered to determine $\uparrow_\circ (F(x))$ [8].

5 Restriction to errors of the n^{th} order

The semantics $\mathcal{S}^{\mathcal{L}^*}$, introduced in Section 4, computes the errors of any order made during a computation. This is a general model for error propagation but it is commonly admitted that, in practice, errors of order greater than one or (rarely) two are negligible [5]. However, even if, from a practical point of view, we are only interested in detailing the contribution of the first n order errors to the global error, for $n = 1$ or $n = 2$, a safe semantics must check that higher order errors actually are negligible.

We introduce a family $(\mathcal{S}^{\mathcal{L}^n})_{n \in \mathbb{N}}$ of semantics such that the semantics $\mathcal{S}^{\mathcal{L}^n}$ details the contribution to the global error of the errors of order at most n. In addition, $\mathcal{S}^{\mathcal{L}^n}$ collapses into the coefficient of a single formal variable of the series the whole contribution of the errors of order higher than n. A value r is represented by

$$r = f\boldsymbol{\varepsilon} + \sum_{u \in \overline{\mathcal{L}^+},\ |u| \le n} \omega^u \boldsymbol{\varepsilon}_u + \omega^\varsigma \boldsymbol{\varepsilon}_\varsigma \tag{14}$$

The term $\omega^\varsigma \boldsymbol{\varepsilon}_\varsigma$ of the series aggregates the elementary errors of order higher than n. Starting with $n = 1$, one can examine the contribution of the first order errors to the global error in a computation. If the ω^ς coefficient is negligible in the result, then the semantics $\mathcal{S}^{\mathcal{L}^1}$ provides enough information to understand the nature of the error. Otherwise, $\mathcal{S}^{\mathcal{L}^1}$ states that there is a higher order error which is not negligible but does not indicate which operation mainly makes this error grow. This information can be obtained by the semantics $\mathcal{S}^{\mathcal{L}^n}$ for an adequate value of n.

Let $\mathcal{L}^n$ be the set of words of length at most n on the alphabet $\mathcal{L}$ and let $\overline{\mathcal{L}^n} = (\mathcal{L}^n / \sim) \cup \{\varsigma\}$. $\varsigma \notin \mathcal{L}^*$ is a special word representing all the words of size greater than n. We define the new concatenation operator

$$u \cdot_n v = \begin{cases} u.v \text{ if } |u.v| \leq n \text{ and } u, v \neq \varsigma \\ \varsigma \text{ otherwise} \end{cases} \tag{15}$$

For the sake of simplicity, we write $u.v$ instead of $u \cdot_n v$ whenever it is clear that u and v belong to $\overline{\mathcal{L}^n}$. The domain of floating-point numbers with errors of order at most n is $\mathbb{R}^{\mathcal{L}^n} = \mathcal{F}(\mathbb{R}, \overline{\mathcal{L}^n})$. The elementary operations on $\mathbb{R}^{\mathcal{L}^n}$ are defined by the equations (5) to (9) of Section 4 in which $\mathcal{W} = \overline{\mathcal{L}^n}$.

Let $\mathcal{S}^{\mathcal{L}^n}$ denote the semantics defined by the domain $\mathbb{R}^{\mathcal{L}^n}$ for values and by the reduction rules of Section 2. The semantics $\mathcal{S}^{\mathcal{L}^n}$ indicates the contribution to the global error of the elementary errors of order at most n.

The correctness of the semantics $(\mathcal{S}^{\mathcal{L}^n})_{n \in \mathbb{N}}$ stems from the fact that, for any n, $\mathcal{S}^{\mathcal{L}^n}$ is a conservative approximation of $\mathcal{S}^{\mathcal{L}^*}$. Furthermore, for any $1 \leq m \leq n$, $\mathcal{S}^{\mathcal{L}^m}$ is a conservative approximation of $\mathcal{S}^{\mathcal{L}^n}$. So, the semantics of order m can always be considered as being a safe approximation of the semantics of order n, for any $n > m$. To prove the previous claims we introduce the following Galois connections [3, 4] in which $m \leq n$.

$$\langle \wp(\mathcal{F}(\mathbb{R}, \overline{\mathcal{L}^n})), \subseteq \rangle \quad \underset{\alpha^{n,m}}{\overset{\gamma^{m,n}}{\leftrightarrows}} \quad \langle \mathcal{F}(\mathbb{R}, \overline{\mathcal{L}^m}), \sqsubseteq \rangle \tag{16}$$

$\wp(X)$ denotes the power set of X and $\sqsubseteq$ denotes the componentwise ordering on formal series. $\alpha^{n,m}$ and $\gamma^{m,n}$ are defined by

$$\alpha^{n,m}\left(\bigcup_{i \in I} \sum_{u \in \overline{\mathcal{L}^n}} \omega_i^u \boldsymbol{\varepsilon}_u\right) \stackrel{\text{def}}{=} \sum_{u \in \overline{\mathcal{L}^m} \setminus \{\varsigma\}} \left(\bigvee_{i \in I} \omega_i^u\right) \boldsymbol{\varepsilon}_u + \bigvee_{i \in I} \left(\sum_{u \in (\overline{\mathcal{L}^n} \setminus \overline{\mathcal{L}^m}) \cup \{\varsigma\}} \omega^u\right) \boldsymbol{\varepsilon}_\varsigma$$

$$\gamma^{m,n}\left(\sum_{u \in \overline{\mathcal{L}^m}} \nu^u \boldsymbol{\varepsilon}_u\right) \stackrel{\text{def}}{=} \left\{ \sum_{u \in \overline{\mathcal{L}^n}} \omega^u \boldsymbol{\varepsilon}_u \ : \ \left| \begin{array}{l} \omega^u \leq \nu^u \text{ if } u \in \overline{\mathcal{L}^m} \setminus \{\varsigma\} \\ \sum_{u \in (\overline{\mathcal{L}^n} \setminus \overline{\mathcal{L}^m}) \cup \{\varsigma\}} \omega^u \leq \nu^\varsigma \end{array} \right. \right\}$$

The abstraction of the coefficients attached to $\boldsymbol{\varepsilon}_u$, for any $u \in \overline{\mathcal{L}^m} \setminus \{\varsigma\}$, is natural. $\alpha^{n,m}$ also adds the coefficients of the terms of order higher than m and appends the result to $\boldsymbol{\varepsilon}_\varsigma$. Next the supremum is taken between the terms $\nu_i^\varsigma \boldsymbol{\varepsilon}_\varsigma$, $i \in I$. $\gamma^{m,n}$ maps a series $\sum_{u \in \overline{\mathcal{L}^m}} \nu^u \boldsymbol{\varepsilon}_u \in \mathcal{F}(\mathbb{R}, \overline{\mathcal{L}^m})$ to the set of series of the form $\sum_{u \in \overline{\mathcal{L}^n}} \omega^u \boldsymbol{\varepsilon}_u \in \mathcal{F}(\mathbb{R}, \overline{\mathcal{L}^n})$ whose coefficients ω^u are less than ν^u for any $u \in \overline{\mathcal{L}^m} \setminus \{\varsigma\}$ and such that ν^ς is greater than the sum of the remaining terms. The correctness of the elementary operations in $\mathcal{F}(\mathbb{R}, \overline{\mathcal{L}^m})$, w.r.t. the correctness of the same operations in $\mathcal{F}(\mathbb{R}, \overline{\mathcal{L}^n})$ stems from Lemma 4.

Lemma 4 *Let ℓ_i be a program point, let $R^{\mathcal{L}^n}, S^{\mathcal{L}^n} \in \wp(\mathcal{F}(\mathbb{R}, \overline{\mathcal{L}^n}))$ be sets of floating-point numbers with errors and let $r^{\mathcal{L}^m} = \alpha^{n,m}(R^{\mathcal{L}^n})$, $s^{\mathcal{L}^m} = \alpha^{n,m}(S^{\mathcal{L}^n})$, $1 \leq m \leq n$. For any operator $\diamond \in \{+, -, \times, \div\}$ we have*

$$R^{\mathcal{L}^n} \diamond^{\ell_i} S^{\mathcal{L}^n} \subseteq \gamma^{m,n}(r^{\mathcal{L}^m} \diamond^{\ell_i} s^{\mathcal{L}^m})$$

Proofs are given in [8]. To extend Lemma 4 to sequences of reduction steps, we introduce the mapping $\mathcal{R}$ defined by Equation (17). $\mathrm{Lab}(a_0^{\ell_0})$ is the set of labels occurring in $a_0^{\ell_0}$.

$$\mathcal{R}(a_0^{\ell_0}) \ : \ \left| \begin{array}{ccl} \mathrm{Lab}(a_0^{\ell_0}) & \rightarrow & \mathcal{F}(\mathbb{R}, \overline{\mathcal{L}^n}) \\ \ell & \mapsto & \begin{cases} r \text{ if } a^\ell = r^\ell \\ \perp \text{ otherwise} \end{cases} \end{array} \right. \tag{17}$$

$\mathcal{R}(a_0^{\ell_0})(\ell)$ returns a value if the sub-expression a^ℓ in $a_0^{\ell_0}$ is a value and $\perp$ otherwise.

Proposition 5 *Let $a_0^{\ell_0 \mathcal{L}^m}$ and $a_0^{\ell_0 \mathcal{L}^n}$ be syntactically equivalent expressions such that for any $\ell \in \mathcal{L}$, $\mathcal{R}(a_0^{\ell_0 \mathcal{L}^n})(\ell) \in \gamma^{m,n}(\mathcal{R}(a_0^{\ell_0 \mathcal{L}^m})(\ell))$. If $a_0^{\ell_0 \mathcal{L}^m} \rightarrow a_1^{\ell_1 \mathcal{L}^m}$ then $a_0^{\ell_0 \mathcal{L}^n} \rightarrow a_1^{\ell_1 \mathcal{L}^n}$ such that $a_1^{\ell_1 \mathcal{L}^m}$ and $a_1^{\ell_1 \mathcal{L}^n}$ are syntactically equivalent expressions and for all $\ell \in \mathcal{L}$, $\mathcal{R}(a_1^{\ell_1 \mathcal{L}^n})(\ell) \in \gamma^{m,n}(\mathcal{R}(a_1^{\ell_1 \mathcal{L}^n})(\ell))$.*

Given an arithmetic expression $a_0^{\ell_0}$, Proposition 5 shows how to link $\mathcal{S}^{\mathcal{L}^n}(a_0^{\ell_0})$ to $\mathcal{S}^{\mathcal{L}^{n+1}}(a_0^{\ell_0})$ for any integer $n \geq 0$. The semantics $\mathcal{S}^{\mathcal{L}^n}$ is based on the domain $\mathbb{R}^{\mathcal{L}^n} = \mathcal{F}(\mathbb{R}, \overline{\mathcal{L}^n})$. The semantics $\mathcal{S}^{\mathcal{L}^*}$ can be viewed as a simple instance of this model, since the operations on $\mathbb{R}^{\mathcal{L}^*}$, as defined in Section 4, correspond to the ones of equations (5) to (9) with $\mathcal{W} = \mathcal{L}^*$. Conversely, the semantics $\mathcal{S}^{\mathcal{L}^0}$ uses floating-point numbers with errors of the form $\omega^\epsilon \varepsilon_\epsilon + \omega^\varsigma \varepsilon_\varsigma$ and computes the global error done during a computation. In short, there is a chain of Galois connections between the semantics of any order:

$$\mathcal{S}^{\mathcal{L}^*}(a_0^{\ell_0}) \ \leftrightarrows \ \ldots \ \mathcal{S}^{\mathcal{L}^n}(a_0^{\ell_0}) \ \leftrightarrows \ \mathcal{S}^{\mathcal{L}^{n-1}}(a_0^{\ell_0}) \ \ldots \ \leftrightarrows \ \mathcal{S}^{\mathcal{L}^0}(a_0^{\ell_0})$$

$\mathcal{S}^{\mathcal{L}^*}(a_0^{\ell_0})$ is the most informative result since it indicates the contribution of the elementary errors of any order. $\mathcal{S}^{\mathcal{L}^0}(a_0^{\ell_0})$ is the least informative result which only indicates the global error made during the computation.

6 Coarse grain errors

In this section, we introduce a new semantics that generalizes the ones of Section 4 and Section 5. Intuitively, we no longer consider elementary errors corresponding to the errors due to individual operations and we instead compute errors due to pieces of code partitioning the program. For instance, one may be interested in the contribution to the global error of the whole error due to an intermediate formula or due to a given line in the program code.

In practice, these new semantics are important to reduce the memory size used to store the values. From a theoretical point of view, it is necessary to prove that they are correct with respect to the general semantics $\mathcal{S}^{\mathcal{L}^*}$ of Section 4.

We show that the different partitions of the program points can be partially ordered in such a way that we can compare the semantics based on comparable partitions. Let $\mathcal{J} = \{J_1, J_2, \ldots, J_p\} \in \mathcal{P}(\mathcal{L})$ be a partition of the program points. We consider now the words on the alphabet $\mathcal{J}$. $\mathcal{J}^n$ denotes the words of maximal length n and $\overline{\mathcal{J}^n} = (\mathcal{J}^n/\sim) \cup \{\varsigma\}$. The concatenation operator $\cdot_n$ related to $\overline{\mathcal{J}^n}$ is defined in Equation (15).

For a maximal order of error $n \in \mathbb{N}$, we consider the family of domains $(\mathcal{F}(\mathbb{R}, \overline{\mathcal{J}^n}))_{\mathcal{J}\in\mathcal{P}(\mathcal{L})}$, equivalently denoted $(\mathbb{R}^{\mathcal{J}^n})_{\mathcal{J}\in\mathcal{P}(\mathcal{L})}$. Basically, a unique label identifies all the operations of the same partition. A value $r^{\mathcal{J}^n} \in \mathbb{R}^{\mathcal{J}^n}$ is written

$$r^{\mathcal{J}^n} = f\boldsymbol{\varepsilon} + \sum_{\substack{u \in \overline{\mathcal{J}^{n+}} \\ u = J_1 \dots J_k}} \Bigg(\sum_{\substack{v = \ell_1 \dots \ell_k \in \overline{\mathcal{L}^n} \\ \forall i,\ 1 \le i \le k,\ \ell_i \in J_i}} \omega^v \Bigg) \boldsymbol{\varepsilon}_u = f\boldsymbol{\varepsilon} + \sum_{u \in \overline{\mathcal{J}^{n+}}} \omega^u \boldsymbol{\varepsilon}_u$$

If $|u| = 1$, the word $u = J$ is related to the first order error due to the operations whose label belongs to J. The elementary operations on $\mathbb{R}^{\mathcal{J}^n}$ are defined by the equations (5) to (9) of Section 4 in which $\mathcal{W} = \overline{\mathcal{J}^n}$.

Remark that this semantics generalizes the semantics of Section 5 which is based on a particular partition $\mathcal{J} = \{\{\ell\} \ :\ \ell \in \mathcal{L}\}$. Another interesting partition consists of using singletons for the initial data and collapsing all the other program points. This enables us to determine the contribution, to the global error, of initial errors on the program inputs (sensitivity analysis).

In the rest of this section, we compare the semantics based on different partitions of the labels. Intuitively, a partition $\mathcal{J}_1$ is coarser than a partition $\mathcal{J}_2$ if $\mathcal{J}_1$ collapses some of the elements of $\mathcal{J}_2$. For a maximal order of error n and using this ordering, the partition $\mathcal{J} = \{\{\ell\} \ :\ \ell \in \mathcal{L}\}$ corresponds to $\mathcal{S}^{\mathcal{L}^n}$ and is the finest partition. We show that any semantics based on a partition $\mathcal{J}_2$, coarser than a partition $\mathcal{J}_1$, is an approximation of the semantics based on $\mathcal{J}_1$. Consequently, any semantics based on a partition of the program points is an approximation of the general semantics $\mathcal{S}^{\mathcal{L}^n}$ and $\mathcal{S}^{\mathcal{L}^*}$ defined in Section 4 and 5. The partial ordering on the set of partitions is defined below.

Definition 6 *Let $\mathcal{J}_1$ and $\mathcal{J}_2$ be two partitions of the set $\mathcal{L}$. $\mathcal{J}_1$ is a coarser partition of $\mathcal{L}$ than $\mathcal{J}_2$ and we write $\mathcal{J}_1 \dot{\subseteq} \mathcal{J}_2$ iff $\forall J_2 \in \mathcal{J}_2,\ \exists J_1 \in \mathcal{J}_1 \ :\ J_2 \subseteq J_1$.*

If $\mathcal{J}_1 \dot{\subseteq} \mathcal{J}_2$ then some components of the partition $\mathcal{J}_2$ are collapsed in $\mathcal{J}_1$. $\dot{\subseteq}$ is used in the following to order the partitions of the set $\mathcal{L}$ of labels. The *translation* function $\tau_{\mathcal{J}_2^n,\mathcal{J}_1^n}$ maps words on the alphabet $\mathcal{J}_2^n$ to words on $\mathcal{J}_1^n$ as follows.

$$\tau_{\mathcal{J}_2^n,\mathcal{J}_1^n}(J_2.u) = J_1.\tau_{\mathcal{J}_2^n,\mathcal{J}_1^n}(u) \text{ where } J_1 \in \mathcal{J}_1,\ J_2 \subseteq J_1$$

The correctness of any semantics based on a partition $\mathcal{J}_1$ of $\mathcal{L}$ stems from the fact that, for any $\mathcal{J}_2 \in \mathcal{P}(\mathcal{L})$ such that $\mathcal{J}_1 \dot{\subseteq} \mathcal{J}_2$, there is a Galois connection

$$\langle \wp(\mathcal{F}(\mathbb{R}, \overline{\mathcal{J}_2^n})), \subseteq \rangle \underset{\alpha^{\mathcal{J}_2^n,\mathcal{J}_1^n}}{\overset{\gamma^{\mathcal{J}_1^n,\mathcal{J}_2^n}}{\leftrightarrows}} \langle \mathcal{F}(\mathbb{R}, \overline{\mathcal{J}_1^n}), \sqsubseteq \rangle$$

defined by

$$\alpha^{\mathcal{J}_2^n,\mathcal{J}_1^n}\left(\bigcup_{i \in I} \sum_{u \in \overline{\mathcal{J}_2^n}} \omega_i^u \boldsymbol{\varepsilon}_u \right) \stackrel{\text{def}}{=} \sum_{u \in \overline{\mathcal{J}_2^n}} \left(\bigvee_{i \in I} \omega_i^u \right) \boldsymbol{\varepsilon}_{\tau_{\mathcal{J}_2^n,\mathcal{J}_1^n}(u)}$$

$$\gamma^{\mathcal{J}_1^n,\mathcal{J}_2^n}\left(\sum_{v\in\overline{\mathcal{J}_1^n}} \nu^u \boldsymbol{\varepsilon}_v\right) \overset{\text{def}}{=} \left\{\sum_{u\in\overline{\mathcal{J}_2^n}} \omega^u \boldsymbol{\varepsilon}_u \ : \sum_{\tau_{\mathcal{J}_2^n,\mathcal{J}_1^n}(u)=v} \omega^u \le \nu^v\right\}$$

Let J be an element of the coarser partition $\mathcal{J}_1$. For any first order error term $\omega^u\boldsymbol{\varepsilon}_u$ attached to a floating-point number with errors $r^{\overline{\mathcal{J}_2^n}} = \sum_{u\in\overline{\mathcal{J}_2^n}} \omega^u \boldsymbol{\varepsilon}_u$, the abstraction $\alpha^{\mathcal{J}_2^n,\mathcal{J}_1^n}(\{r^{\overline{\mathcal{J}_2^n}}\})$ defines the coefficient ν_J attached to $\boldsymbol{\varepsilon}_J$ as being the sum of the coefficients $\omega_{J'}$ such that $J' \in \mathcal{J}_2$ and $J' \subseteq J$. The abstraction of sets of numbers is next defined in the usual way, by taking the supremum of the coefficients for each component. The function $\gamma^{\mathcal{J}_1^n,\mathcal{J}_2^n}$ returns the set of floating-point numbers with errors for which $\sum_{\tau_{\mathcal{J}_2^n,\mathcal{J}_1^n}(u)=v} \omega^u \le \nu^v$. Lemma 7 assesses the correctness of the operations defined by Equations (5) to (9) for the domains introduced in this section.

Lemma 7 *Let ℓ_i be a program point, let $\mathcal{J}_1$ and $\mathcal{J}_2$ be partitions of $\mathcal{L}$ such that $\mathcal{J}_1 \dot{\subseteq} \mathcal{J}_2$ and let $R^{\mathcal{J}_2^n}, S^{\mathcal{J}_2^n} \in \wp(\mathcal{F}(\mathbb{R}, \overline{\mathcal{J}_2^n}))$. If $r^{\mathcal{J}_1^n} = \alpha^{\mathcal{J}_2^n,\mathcal{J}_1^n}(R^{\mathcal{J}_2^n})$, $s^{\mathcal{J}_1^n} = \alpha^{\mathcal{J}_2^n,\mathcal{J}_1^n}(S^{\mathcal{J}_2^n})$ then for any operator $\diamond \in \{+,-,\times,\div\}$ we have*

$$R^{\mathcal{J}_2^n} \diamond^{\ell_i} S^{\mathcal{J}_2^n} \in \gamma^{\mathcal{J}_1^n,\mathcal{J}_2^n}(r^{\mathcal{J}_1^n} \diamond^{\ell_i} s^{\mathcal{J}_1^n})$$

The proof is given in [8]. The semantics defined by the domain $\mathbb{R}^{\mathcal{J}^n}$ for values and by the reduction rules of Section 3 is denoted $\mathcal{S}^{\mathcal{J}^n}$. Proposition 8 establishes the link between the semantics $\mathcal{S}^{\mathcal{J}_1^n}$ and $\mathcal{S}^{\mathcal{J}_2^n}$ for comparable partitions $\mathcal{J}_1 \dot{\subseteq} \mathcal{J}_2$ of the set $\mathcal{L}$ of labels.

Proposition 8 *Let $\mathcal{J}_1$ and $\mathcal{J}_2$ be partitions of $\mathcal{L}$ such that $\mathcal{J}_1 \dot{\subseteq} \mathcal{J}_2$ and let $a_0^{\ell_0 \mathcal{J}_1^n}$ and $a_0^{\ell_0 \mathcal{J}_2^n}$ be syntactically equivalent expressions such that for all $\ell \in \mathcal{L}$, $\mathcal{R}(a_0^{\ell_0 \mathcal{J}_2^n})(\ell) \in \gamma^{\mathcal{J}_1^n,\mathcal{J}_2^n}(\mathcal{R}(a_0^{\ell_0 \mathcal{J}_1^n})(\ell))$. If $a_0^{\ell_0 \mathcal{J}_1^n} \to a_1^{\ell_1 \mathcal{J}_1^n}$ then $a_0^{\ell_0 \mathcal{J}_2^n} \to a_1^{\ell_1 \mathcal{J}_2^n}$ such that $a_1^{\ell_1 \mathcal{J}_1^n}$ and $a_1^{\ell_1 \mathcal{J}_2^n}$ are syntactically equivalent expressions and for all $\ell \in \mathcal{L}$, $\mathcal{R}(a_1^{\ell_1 \mathcal{J}_2^n})(\ell) \in \gamma^{\mathcal{J}_1^n,\mathcal{J}_2^n}(\mathcal{R}(a_1^{\ell_1 \mathcal{J}_2^n})(\ell))$.*

As a consequence, for a given order of error n and for a given chain C of partitions, there is a chain of Galois connections between the semantics based on the partitions of C. Let us assume that

$$C = \mathcal{J}_0 = \{\mathcal{L}\} \ \dot{\subseteq} \ \ldots \ \dot{\subseteq} \ \ldots \ \dot{\subseteq} \ \mathcal{J}_k \ \dot{\subseteq} \ \ldots \ \dot{\subseteq} \ \{\{\ell\} \ : \ell \in \mathcal{L}\}$$

By combining Proposition 5 and Proposition 8, we can also link the semantics $\mathcal{S}^{\mathcal{J}_k^n}$ and $\mathcal{S}^{\mathcal{J}_k^{n+1}}$ for any $\mathcal{J}_k \in C$ and any $n \in \mathbb{N}$. This is summed up in Figure 2. For any integer n and partition $\mathcal{J}_k$, $\mathcal{S}^{\mathcal{J}_k^n}$ describes a particular semantics; $\mathcal{S}^{\mathcal{L}^*}$ is the most informative semantics and, conversely, the semantics $\mathcal{S}^{\mathcal{L}^0}$ that computes one global error term is the least informative semantics; for all $k > 0$ $\mathcal{S}^{\mathcal{J}_k^0} = \mathcal{S}^{\mathcal{L}^0}$ (for $n = 0$, any partition yields the same semantics); $\mathcal{S}^{\mathcal{J}_0^2}$ computes the global first order and second order errors made during a computation; finally, for any $\mathcal{J}_k$, $\mathcal{S}^{\mathcal{J}_k^1}$ computes the contribution to the global error of the first order errors made in the different pieces of code identified by $\mathcal{J}_k$.

Let us remark that the values in $\mathbb{R}^{\mathcal{J}_1^n}$ contain less terms than the ones in $\mathbb{R}^{\mathcal{J}_2^n}$ if $\mathcal{J}_1 \dot{\subseteq} \mathcal{J}_2$. Hence, using coarser partitions leads to significantly fewer computations.

$$\begin{tikzcd}
\mathcal{S}^{\mathcal{J}_0^*}(a_0^{\ell_0}) \arrow[r, shift right] \arrow[d, shift left] & \dots \arrow[l, shift right] \arrow[r, shift right, "\alpha^{n+1,n}"'] \arrow[d, shift left] & \mathcal{S}^{\mathcal{J}_0^n}(a_0^{\ell_0}) \arrow[l, shift right, "\gamma^{n,n+1}"'] \arrow[r, shift right, "\alpha^{n,n-1}"'] \arrow[d, shift left, "\gamma^{\mathcal{J}_0^n,\mathcal{J}_1^n}"] & \dots \arrow[l, shift right, "\gamma^{n-1,n}"'] \arrow[r, shift right] \arrow[d, shift left] & \mathcal{S}^{\mathcal{J}_0^0}(a_0^{\ell_0}) \arrow[l, shift right] \arrow[d, shift left] \\
\mathcal{S}^{\mathcal{J}_1^*}(a_0^{\ell_0}) \arrow[r, shift right] \arrow[u, shift left] \arrow[d, shift left] & \dots \arrow[l, shift right] \arrow[r, shift right] \arrow[u, shift left] \arrow[d, shift left] & \mathcal{S}^{\mathcal{J}_1^n}(a_0^{\ell_0}) \arrow[l, shift right] \arrow[r, shift right] \arrow[u, shift left, "\alpha^{\mathcal{J}_1^n,\mathcal{J}_0^n}"] \arrow[d, shift left] & \dots \arrow[l, shift right] \arrow[r, shift right] \arrow[u, shift left] \arrow[d, shift left] & \mathcal{S}^{\mathcal{J}_1^0}(a_0^{\ell_0}) \arrow[l, shift right] \arrow[u, shift left] \arrow[d, shift left] \\
\dots \arrow[r, shift right] \arrow[u, shift left] \arrow[d, shift left] & \dots \arrow[l, shift right] \arrow[r, shift right] \arrow[u, shift left] \arrow[d, shift left] & \dots \arrow[l, shift right] \arrow[r, shift right] \arrow[u, shift left] \arrow[d, shift left] & \dots \arrow[l, shift right] \arrow[r, shift right] \arrow[u, shift left] \arrow[d, shift left] & \dots \arrow[l, shift right] \arrow[u, shift left] \arrow[d, shift left] \\
\mathcal{S}^{\mathcal{L}^*}(a_0^{\ell_0}) \arrow[r, shift right] \arrow[u, shift left] & \dots \arrow[l, shift right] \arrow[r, shift right] \arrow[u, shift left] & \mathcal{S}^{\mathcal{L}^n}(a_0^{\ell_0}) \arrow[l, shift right] \arrow[r, shift right] \arrow[u, shift left] & \dots \arrow[l, shift right] \arrow[r, shift right] \arrow[u, shift left] & \mathcal{S}^{\mathcal{L}^0}(a_0^{\ell_0}) \arrow[l, shift right] \arrow[u, shift left]
\end{tikzcd}$$

Fig. 2. Links between the semantics $\mathcal{S}^{\mathcal{J}_k^n}$ for a given order of error n and for a chain of partitions $\mathcal{J}_0 \dot{\subseteq} \mathcal{J}_1 \dot{\subseteq} \dots \dot{\subseteq} \mathcal{J}_k \dot{\subseteq} \dots \dot{\subseteq} \{\{\ell\} : \ell \in \mathcal{L}\}$.

7 Conclusion

The semantics introduced in this article models the propagation of roundoff errors and the introduction of new errors at each stage of a computation. We use a unified framework, mainly based on the equations of Figure 1, to compute the contribution, to the global error, of the errors due to pieces of code partitioning the program and up to a maximal order of error. Lemma 2 and Proposition 3 are essential to ensure the correctness of the operators of Figure 1. They also represent a stronger correctness criterion for the operators introduced in [7]. Another important point is that $\mathcal{S}^{\mathcal{L}^n}$ not only details the propagation of the errors of order $\leq n$ but also verifies that higher order error terms actually are negligible.

A tool has been developed which implements an abstract interpretation based on the semantics introduced in this article. The real coefficients of the error series are abstracted by intervals of multi-precision floating-point numbers. This tool is described in [9]. Current work concerns the precision of the abstract interpretation in loops and is proceeding in two directions. First, because narrowings do not yield information to improve the precision of the error terms, we really need finely-tuned widening operators for our domain. This should enable us to restrict the number of cases where a loop is stable but is declared unstable by the abstract interpreter because of the approximations made during the analysis. The second way to improve the precision in loops consists of using a relational analysis. A first solution was proposed in [7] that can be used when the errors made at the iterations n and $n+1$ are related by a linear transformation. We are also working on the non-linear cases, using mathematical tools developed for the study of dynamical systems.

Acknowledgements

I would like to thank Eric Goubault, Sylvie Putot and Nicky Williams for their suggestions on earlier drafts of this paper.

References

1. F. Chaitin-Chatelin and V. Frayssé. *Lectures on Finite Precision Computations.* SIAM, 1996.
2. F. Chaitin-Chatelin and E. Traviesas. Precise, a toolbox for assessing the quality of numerical methods and software. In *IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics*, 2000.
3. P. Cousot and R. Cousot. Abstract interpretation: A unified lattice model for static analysis of programs by construction of approximations of fixed points. *Principles of Programming Languages 4*, pages 238–252, 1977.
4. P. Cousot and R. Cousot. Abstract interpretation frameworks. *Journal of Logic and Symbolic Computation*, 2(4):511–547, 1992.
5. M. Daumas and J. M. Muller, editors. *Qualité des Calculs sur Ordinateur.* Masson, 1997.
6. D. Goldberg. What every computer scientist should know about floating-point arithmetic. *ACM Computing Surveys*, 23(1), 1991.
7. E. Goubault. Static analyses of the precision of floating-point operations. In *Static Analysis Symposium, SAS'01*, number 2126 in LNCS. Springer-Verlag, 2001.
8. E. Goubault, M. Martel, and S. Putot. Concrete and abstract semantics of fp operations. Technical Report DRT/LIST/DTSI/SLA/LSL/01-058, CEA, 2001.
9. E. Goubault, M. Martel, and S. Putot. Asserting the precision of floating-point computations: a simple abstract interpreter. In *ESOP'02*, 2002.
10. J. R. Hauser. Handling floating-point exceptions in numeric programs. *ACM Transactions on Programming Languages and Systems*, 18(2), 1996.
11. W. Kahan. The improbability of probabilistic error analyses for numerical computations. Technical report, Berkeley University, 1991.
12. W. Kahan. Lecture notes on the status of IEEE standard 754 for binary floating-point arithmetic. Technical report, Berkeley University, 1996.
13. D. Knuth. *The Art of Computer Programming - Seminumerical Algorithms.* Addison Wesley, 1973.
14. P. Langlois and F. Nativel. Improving automatic reduction of round-off errors. In *IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics*, volume 2, 1997.
15. V. Lefevre, J.M. Muller, and A. Tisserand. Toward correctly rounded transcendentals. *IEEE Transactions on Computers*, 47(11), 1998.
16. M. Pichat and J. Vignes. The numerical study of unstable fixed points in a chaotic dynamical system. In *IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics,*, volume 2, 1997.
17. J. Vignes. A survey of the CESTAC method. In *Proceedings of Real Numbers and Computer Conference*, 1996.