



HAL
open science

Error in ulps of the multiplication or division by a correctly-rounded function or constant in binary floating-point arithmetic

Nicolas Brisebarre, Jean-Michel Muller, Joris Picot

► To cite this version:

Nicolas Brisebarre, Jean-Michel Muller, Joris Picot. Error in ulps of the multiplication or division by a correctly-rounded function or constant in binary floating-point arithmetic. 2023. hal-04044716v1

HAL Id: hal-04044716

<https://hal.science/hal-04044716v1>

Preprint submitted on 24 Mar 2023 (v1), last revised 11 Jul 2023 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Error in ulps of the multiplication or division by a correctly-rounded function or constant in binary floating-point arithmetic

Nicolas Brisebarre, Jean-Michel Muller, *Fellow, IEEE*, Joris Picot

Abstract—Assume we use a binary floating-point arithmetic and that RN is the round-to-nearest function. Also assume that c is a constant or a real function of one or more variables, and that we have at our disposal a correctly rounded implementation of c , say $\hat{c} = \text{RN}(c)$. For evaluating $x \cdot c$ (resp. x/c or c/x), the natural way is to replace it by $\text{RN}(x \cdot \hat{c})$ (resp. $\text{RN}(x/\hat{c})$ or $\text{RN}(\hat{c}/x)$), that is, to call function \hat{c} and to perform a floating-point multiplication or division. This can be generalized to the approximation of n/d by $\text{RN}(\hat{n}/\hat{d})$ and the approximation of $n \cdot d$ by $\text{RN}(\hat{n} \cdot \hat{d})$, where $\hat{n} = \text{RN}(n)$ and $\hat{d} = \text{RN}(d)$, and n and d are functions for which we have at our disposal a correctly rounded implementation. We discuss tight error bounds in ulps of such approximations. From our results, one immediately obtains tight error bounds for calculations such as $x * \text{pi}$, $\ln(2)/x$, $x/(y+z)$, $(x+y) * z$, $x/\text{sqrt}(y)$, $\text{sqrt}(x)/y$, $(x+y)(z+t)$, $(x+y)/(z+t)$, $(x+y)/(zt)$, etc. in floating-point arithmetic.

Index Terms—Floating-point arithmetic, ulp, numerical error, correct rounding, multiplication by a constant

I. INTRODUCTION

A. Purpose of this paper and notation

MULTIPLYING a floating-point number by a real constant (such as π or $\ln(2)$), or multiplying or dividing it by a correctly rounded function of one or more variables (such as \sqrt{x} , $x+y$, or xy), or dividing a constant or correctly-rounded function by a floating-point number are very frequent operations in numerical computing. When last-bit accuracy is desired, one can use specifically-designed solutions (see for instance [1] for multiplication by a constant). However, here, we are interested by the error of the straightforward approach. For instance, when a programmer writes the statement

$$s = x * \text{pi};$$

he or she most probably wants to compute $x \cdot \pi$ as accurately as possible. However the variable `pi` is already a floating-point approximation of the real π , and, as a result, the error in this computation is larger than the mere approximation caused by the floating-point multiplication. We are also interested in calculations such as

$$s = x/\text{sqrt}(y); \quad \text{or} \quad s = (x+y) * (z+t);$$

A tight bound on the *relative error* of such operations is very easily obtained (see Section I-B). And yet, although relative

errors are frequently easier to manipulate, for basic functions that can be regarded as “atomic”, errors in ulps are frequently preferred, because they somehow convey more information. The purpose of this paper is to give very tight bounds on the error in ulps on the multiplication or division of a floating-point number by a real constant or a correctly-rounded function, or on the product or quotient of two correctly-rounded functions.

In the following, we assume a binary, precision- p , floating-point (FP) arithmetic, with correctly rounded (to nearest) floating-point operations. A floating-point number is zero or a number of the form

$$x = M_x \cdot 2^{e_x - p + 1},$$

where M_x and e_x are integers, with $2^{p-1} \leq |M_x| \leq 2^p - 1$. Here, we do not assume bounds on the exponents e_x which means that our results apply to real-life arithmetics such as the ones specified by the IEEE 754 Standard for Floating-Point arithmetic [2] whenever underflow and overflow do not occur. In the following, if t is a real number then $\text{RN}(t)$ is the floating-point number nearest to t (a tie-breaking choice is necessary if t is halfway between two consecutive FP numbers: our bounds are valid whatever the tie-breaking choice, however our examples use the ties-to-even rule, which is the default in IEEE-754 arithmetic). If $t \in \mathbb{R}$, $t \neq 0$ then $\text{ulp}(t)$ (*unit in the last place* of t) is $2^{\lfloor \log_2 |t| \rfloor - p + 1}$ and $\text{uflp}(t)$ (*unit in the first place* [3] of t) is $2^{\lfloor \log_2 |t| \rfloor}$. The unit round-off is the number $u = 2^{-p}$.

In the following, c is either a real constant or a real function of one or more floating-point variables. We assume that $\hat{c} = \text{RN}(c)$ is available. If c is a constant, \hat{c} is just the FP number nearest c . If c is an arithmetic function (e.g., $c(x, y) = x + y$ or $c(x, y) = xy$), or the square root, then \hat{c} is provided by the underlying IEEE-754 arithmetic. If c is a more complex function, such as $\exp(x)$, $\ln(x)$, etc., \hat{c} can be provided by a correctly-rounded function such as the ones included in the CORE-MATH library [4].

We first compute tight bounds on the error, expressed in ulps, of approximating the product $c \cdot x$ (resp. the quotient x/c or the quotient c/x), where x is a floating-point number, by

$$s = \text{RN}(\hat{c} \cdot x) \quad (\text{resp. } \text{RN}(x/\hat{c}) \text{ or } \text{RN}(\hat{c}/x)).$$

This is done in Section II for $x \cdot c$, in Section III for x/c , and in Section IV for c/x .

When c is a constant, we assume that it is not a floating-point number (otherwise, the well-known bound 0.5 ulp applies and is optimal). Our results apply for instance to the computation of

$$x \cdot \cos\left(\frac{2k\pi}{N}\right),$$

that appears in discrete cosine transforms, assuming that either the values $\text{RN}(\cos(2k\pi/N))$ are precomputed and stored, or that N is a power of 2, k is less than 2^p and a correctly-rounded function `cospi` is available (there is one in the CORE-MATH library). Our results also apply to the calculation of expressions of the form $(x+y) \cdot z$, $z/(x+y)$, $(x+y)/z$, $(x \cdot y) \cdot z$, $z/(x \cdot y)$, $x \cdot \sqrt{y}$, x/\sqrt{y} , \sqrt{y}/x , $e^x \cdot y$, $x \cdot \ln(y)$, x/π , π/x , etc., where x , y , and z are FP numbers and assuming that the exponential and logarithm are correctly rounded.

By “expressing the error bound in ulps” we mean that we want to find some real α , as small as possible, such that

$$|s - cx| \leq \alpha \cdot \text{ulp}(cx).$$

For an explanation of why it is wiser to measure errors in terms of ulps of the *exact result* instead of ulps of the *computed result*, see [5, Section 2.5].

In Section V, we generalize our work to the approximation of $m \cdot n$, where m and n are either real constants or correctly-rounded functions, by $s = \text{RN}(\hat{m} \cdot \hat{n})$, where $\hat{m} = \text{RN}(m)$ and $\hat{n} = \text{RN}(n)$. The obtained results apply to functions such as $(x+y) \cdot (z+t)$, $\sqrt{x} \cdot (y \cdot z)$, $\ln(x) \cdot \ln(y)$ (assuming a correctly-rounded logarithm), etc. Finally, in Section VI, we consider the quotient of two correctly-rounded functions.

B. Just a few words on relative errors

Obtaining relative errors on the calculation of xc or x/c is straightforward. We just quickly address that question, for the sake of completeness and to show that the obtained results do not suffice for deducing a tight bound on the error in ulps. The relative error due to rounding to nearest a real number is bounded by $u/(1+u)$ and that bound is optimal [6, p. 232] [7, p. 74] and has been used to obtain tight relative error bounds on various operations (see e.g. [8]). Therefore

$$c \cdot \left(1 - \frac{u}{1+u}\right) \leq \hat{c} \leq c \cdot \left(1 + \frac{u}{1+u}\right),$$

$$\hat{c}x \cdot \left(1 - \frac{u}{1+u}\right) \leq \text{RN}(\hat{c}x) \leq \hat{c}x \cdot \left(1 + \frac{u}{1+u}\right),$$

and

$$\frac{x}{\hat{c}} \cdot \left(1 - \frac{u}{1+u}\right) \leq \text{RN}\left(\frac{x}{\hat{c}}\right) \leq \frac{x}{\hat{c}} \cdot \left(1 + \frac{u}{1+u}\right).$$

From this, we easily deduce that the numbers $s_1 = \text{RN}(\hat{c}x)$ and $s_2 = \text{RN}(x/\hat{c})$ satisfy

$$cx \cdot \left(1 - \frac{u}{1+u}\right)^2 \leq s_1 \leq cx \cdot \left(1 + \frac{u}{1+u}\right)^2,$$

and

$$\frac{x}{c} \cdot \frac{1 - \frac{u}{1+u}}{1 + \frac{u}{1+u}} \leq s_2 \leq \frac{x}{c} \cdot \frac{1 + \frac{u}{1+u}}{1 - \frac{u}{1+u}},$$

so that the relative error of the multiplication by c is bounded by

$$\left(1 + \frac{u}{1+u}\right)^2 - 1 = \frac{2u + 3u^2}{1 + 2u + u^2} < 2u,$$

and the relative error of the division by c is bounded by

$$\max \left\{ 1 - \frac{1 - \frac{u}{1+u}}{1 + \frac{u}{1+u}}; \frac{1 + \frac{u}{1+u}}{1 - \frac{u}{1+u}} - 1 \right\} = \frac{1 + \frac{u}{1+u}}{1 - \frac{u}{1+u}} - 1 = 2u.$$

Very similarly, the relative error of the division of c by x is bounded by $(1 + \frac{u}{1+u})^2 - 1 < 2u$, the relative error due to the approximation of $m \cdot n$ by $\text{RN}(\hat{m} \cdot \hat{n})$ is bounded by

$$\left(1 + \frac{u}{1+u}\right)^3 - 1 < 3u,$$

and the relative error due to the approximation of n/d by $\text{RN}(\hat{n}/\hat{d})$ is bounded by

$$\frac{\left(1 + \frac{u}{1+u}\right)^2}{1 - \frac{u}{1+u}} - 1 = \frac{3u + 4u^2}{1 + u} < 3u + u^2 \simeq 3u.$$

Hence the relative error of xc , x/c and c/x is bounded by $2u$, and the relative error of $m \cdot n$ and n/d are bounded by around $3u$. These bounds are very tight (for instance, in binary32 arithmetic ($p = 24$), error $1.99902u$ is attained for the multiplication of $c = 16779263$ and $x = 8392705$). From these bounds the best bounds on the error in ulps one can deduce [9, Section 2.3.3] are 2 ulp for xc , x/c and c/x (resp. around 3 ulp for $m \cdot n$ and n/d). And yet, we are going to show the significantly better bounds 1.5 ulp (resp. 2.5 ulp).

II. MULTIPLICATION OF A FP NUMBER BY A CONSTANT OR A CORRECTLY-ROUNDED FUNCTION

Let us first establish an error bound in ulps on the computation of $x \cdot c$, where x is a FP number. We will first consider “general” bounds, applicable to any c . Then we will try to improve these bounds in the particular case where c is a constant.

A. First steps

Without loss of generality, we assume $1 \leq c < 2$ and $1 \leq x < 2$. Since \hat{c} is c rounded to nearest, we have

$$|c - \hat{c}| \leq \frac{1}{2} \text{ulp}(c),$$

and therefore

$$|cx - \hat{c}x| \leq \frac{x}{2} \text{ulp}(c). \quad (1)$$

We also have

$$|s - \hat{c}x| \leq \frac{1}{2} \text{ulp}(\hat{c}x), \quad (2)$$

so that, by the triangular inequality

$$|s - cx| \leq \frac{x}{2} \text{ulp}(c) + \frac{1}{2} \text{ulp}(\hat{c}x). \quad (3)$$

Equation (3) expresses the error in terms of $\text{ulp}(c)$ and $\text{ulp}(\hat{c}x)$, whereas we need to express it in terms of $\text{ulp}(cx)$. Since $x \geq 1$ and ulp is an increasing function, $\text{ulp}(c)$ can be bounded (although possibly non-optimally) by $\text{ulp}(cx)$. The

case of $\text{ulp}(\hat{c}x)$ is more difficult to handle: $\hat{c}x$ and cx are very close values, between 1 and 4, so that we will almost always have $\text{ulp}(\hat{c}x) = \text{ulp}(cx)$. However, there may be some corner cases when one of $\hat{c}x$ and cx is less than 2 and the other one is larger than or equal to 2. In these cases $\text{ulp}(\hat{c}x)$ will be either $\frac{1}{2} \text{ulp}(cx)$ or $2 \text{ulp}(cx)$. Let us quickly eliminate the latter case, by showing that when $\text{ulp}(\hat{c}x) > \text{ulp}(cx)$ the computation is very accurate (in that case, the error will be significantly less than the general error bound we give later on).

B. The special case $\text{ulp}(\hat{c}x) > \text{ulp}(cx)$

For that case to happen, we must have $cx < 2 \leq \hat{c}x$. Since $\text{ulp}(c) = 2^{1-p} = 2u$ and $\hat{c} = \text{RN}(c)$, we have

$$c < \hat{c} \leq c + u,$$

and we therefore deduce

$$cx < 2 \leq \hat{c}x \leq cx + ux.$$

Since x is a FP number and $x < 2$, we have $x \leq 2 - 2u$, and therefore

$$cx < 2 \leq \hat{c}x \leq cx + (2u - 2u^2).$$

Hence, $2 \leq \hat{c}x < 2 + 2u - 2u^2$. Since RN is an increasing function and $\text{RN}(2) = \text{RN}(2 + 2u - 2u^2) = 2$ we deduce that $s = \text{RN}(\hat{c}x) = 2$. Therefore,

$$|s - cx| = |2 - cx| \leq 2u - 2u^2 = (1 - u) \cdot \text{ulp}(cx).$$

C. The general case $\text{ulp}(\hat{c}x) \leq \text{ulp}(cx)$

1) *A bound that does not depend on c :* In that case, (2) gives

$$|s - \hat{c}x| \leq \frac{1}{2} \text{ulp}(cx), \quad (4)$$

and (3) implies

$$|s - cx| \leq \frac{x}{2} \text{ulp}(c) + \frac{1}{2} \text{ulp}(cx). \quad (5)$$

As $x \geq 1$, $\text{ulp}(c) \leq \text{ulp}(cx)$. Since $x \leq 2 - 2u$, we finally obtain

$$|s - cx| \leq \left(\frac{3}{2} - u\right) \cdot \text{ulp}(cx). \quad (6)$$

Example 1 below shows that the bound (6) is asymptotically optimal (as $u \rightarrow 0$ or, equivalently, as $p \rightarrow +\infty$): one cannot improve it (at least at order 0 in u) without further assumptions on c . Making further assumptions on c makes no sense if c is the result of a previous operation or function (so that the bound (6) will not be improved at order 0 in u for calculations such as $(x + y) \cdot z$, $(x \cdot y) \cdot z$, $x \cdot \sqrt{y}$, $e^x \cdot y$, $x \cdot \ln(y)$, etc.). However, if c is a real constant (such as π , $\ln(2)$, or one of the terms of the form $\cos(2k\pi/N)$ that appear in Fourier-related transforms), it makes sense to try to find a sharper bound that depends on c .

2) *Improving the bound when c is a constant:* We obtained (5) by adding (4) and (1). Now, we will keep (4)—which cannot be improved, unless \hat{c} is a power of 2—unchanged, and try to improve on (1), by introducing a new bound on $|cx - \hat{c}x|$ that depends on c . We start from

$$|cx - \hat{c}x| = x \cdot |c - \hat{c}|.$$

Two cases may occur:

- if $x < 2/c$ then $\text{ulp}(cx) = \text{ulp}(c) = 2^{1-p} = 2u$, and therefore

$$|cx - \hat{c}x| < 2^p \cdot \left| \frac{c - \hat{c}}{c} \right| \cdot \text{ulp}(cx); \quad (7)$$

- if $x \geq 2/c$ then $\text{ulp}(cx) = 2 \text{ulp}(c) = 2^{2-p}$, and therefore

$$|cx - \hat{c}x| < 2^{p-1} \cdot |c - \hat{c}| \cdot \text{ulp}(cx). \quad (8)$$

Since c is between 1 and 2, the bound (8) is always less than the bound (7). Hence, the bound (7) holds in all cases.

We immediately deduce

$$|s - cx| \leq \left(\frac{1}{2} + 2^p \cdot \left| \frac{c - \hat{c}}{c} \right| \right) \cdot \text{ulp}(cx). \quad (9)$$

Note that the bound (9) varies with p . One may obtain a bound that no longer depends on p , but is in general looser, by defining

$$\mu(c) = \frac{c}{\text{ulp}(c)} = \frac{c}{2^{\lfloor \log_2(c) \rfloor}}$$

(it is the “infinite precision significand” of c —up to now, in our proofs, we used $1 \leq c < 2$, in which case $\mu(c) = c$, but to give a final result that is valid for all positive c , we need here to use function μ). If we note that $|c - \hat{c}| \leq u \cdot \text{ulp}(c)$, we obtain

$$|s - cx| \leq \left(\frac{1}{2} + \frac{1}{\mu(c)} \right) \cdot \text{ulp}(cx).$$

3) *Summary:* We finally obtain

Property 2.1: Assume radix-2, precision- p FP arithmetic. If c is a real constant or a real function of one or more variables, x is a FP number, and $\hat{c} = \text{RN}(c)$ is the correctly-rounded (to nearest) implementation of c then, barring underflow and overflow, the FP number $s = \text{RN}(\hat{c} \cdot x)$ satisfies

$$|s - cx| \leq \left(\frac{3}{2} - u \right) \cdot \text{ulp}(cx). \quad \square$$

Property 2.2: Assume radix-2, precision- p FP arithmetic. If c is a nonzero real constant, x is a FP number, and $\hat{c} = \text{RN}(c)$ then, barring underflow and overflow, the FP number $s = \text{RN}(\hat{c} \cdot x)$ satisfies

$$|s - cx| \leq \left(\frac{1}{2} + \frac{1}{\mu(c)} \right) \cdot \text{ulp}(cx),$$

$$|s - cx| \leq \left(\frac{1}{2} + 2^p \cdot \left| \frac{c - \hat{c}}{c} \right| \right) \cdot \text{ulp}(cx). \quad \square$$

The bound of Property 2.1 is general. The first bound of Property 2.2 is tighter but requires c to be a constant and depends on its value. The second bound of Property 2.2 is even tighter but depends on the values of c and p .

TABLE I

THE VARIOUS BOUNDS GIVEN BY PROPERTIES 2.1 AND 2.2 IN THE CASE $c = \pi$, COMPARED WITH THE ACTUAL LARGEST ERROR OBTAINED THROUGH EXHAUSTIVE TESTING (FOR SMALL VALUES OF p ONLY).

p	8	16	24	53	113
bound of Prop. 2.1	1.496093750	1.499984741	1.499999940	1.500000000	1.500000000
1st bound of Prop. 2.2	1.136619772	1.136619772	1.136619772	1.136619772	1.136619772
2nd bound of Prop. 2.2	.5788515082	.6858466083	.9668685680	.8511161042	.7866483180
actual largest error	.5176877776	.6825298419			

TABLE II

THE VARIOUS BOUNDS GIVEN BY PROPERTIES 2.1 AND 2.2 IN THE CASE $c = \cos(5\pi/32)$, COMPARED WITH THE ACTUAL LARGEST ERROR OBTAINED THROUGH EXHAUSTIVE TESTING (FOR SMALL VALUES OF p ONLY).

p	8	16	24	53	113
bound of Prop. 2.1	1.496093750	1.499984741	1.499999940	1.500000000	1.500000000
1st bound of Prop. 2.2	1.066944035	1.066944035	1.066944035	1.066944035	1.066944035
2nd bound of Prop. 2.2	.7587037370	.9626486317	1.013690470	.7026621871	.8537866473
actual largest error	.7004712694	.9585313311			

TABLE III

THE VARIOUS BOUNDS GIVEN BY PROPERTIES 2.1 AND 2.2 IN THE CASE $c = 263/256$, COMPARED WITH THE ACTUAL LARGEST ERROR OBTAINED THROUGH EXHAUSTIVE TESTING (FOR SMALL VALUES OF p ONLY).

p	8	16	24	53	113
bound of Prop. 2.1	1.496093750	1.499984741	1.499999940	1.500000000	1.500000000
1st bound of Prop. 2.2	1.473384030	1.473384030	1.473384030	1.473384030	1.473384030
2nd bound of Prop. 2.2	1.473384030	.5000000000	.5000000000	.5000000000	.5000000000
actual largest error	1.437500000	.5000000000			

n	2nd bound of Prop. 2.2
4	1.0140506566
5 – 7	1.2984865212
8	1.3717040024
9 – 12	1.4501519783
13	1.4616997574
14	1.4761273962
15 – 18	1.4940991041
19 – 20	1.4970223192
...	...
32	1.4999503673

TABLE IV

FOR VARIOUS VALUES OF n , WE GIVE THE LARGEST VALUE OF THE 2ND BOUND OF PROP. 2.2 FOR ALL CONSTANTS $\cos(2k\pi/2^n)$ THAT APPEAR IN A FFT OR DCT OF SIZE 2^n , ASSUMING $p = 24$, WHICH CORRESPONDS TO BINARY32 ARITHMETIC.

We can now assume $1+2u \leq x \leq 2-2u$ and $1+2u \leq \hat{c} \leq 2$ (so that $1+u \leq c \leq 2$). We have

$$\left|s - \frac{x}{c}\right| \leq \left|\text{RN}\left(\frac{x}{\hat{c}}\right) - \frac{x}{\hat{c}}\right| + \left|\frac{x}{c} - \frac{x}{\hat{c}}\right|.$$

- If $c \leq x$ then $\hat{c} \leq x$ (since RN is an increasing function and $\text{RN}(x) = x$) and therefore $\text{ulp}(x/c) = \text{ulp}(x/\hat{c})$. Also,

$$\left|\text{RN}\left(\frac{x}{\hat{c}}\right) - \frac{x}{\hat{c}}\right| \leq u,$$

and

$$\left|\frac{x}{c} - \frac{x}{\hat{c}}\right| \leq x \cdot \frac{u}{\hat{c}c} \leq (2-2u) \cdot \frac{u}{(1+2u)(1+u)},$$

so that

$$\begin{aligned} \left|s - \frac{x}{c}\right| &\leq u \cdot \left(1 + \frac{2-2u}{(1+2u)(1+u)}\right) \\ &= 3u - \frac{4u^2(2+u)}{1+3u+2u^2}, \end{aligned}$$

and $\text{ulp}(x/c) = 2u$, so that

$$\begin{aligned} \left|s - \frac{x}{c}\right| &\leq \left(\frac{3}{2} - \frac{4u+2u^2}{1+3u+2u^2}\right) \text{ulp}\left(\frac{x}{c}\right) \\ &\leq \left(\frac{3}{2} - 4u + 10u^2\right) \text{ulp}\left(\frac{x}{c}\right). \end{aligned}$$

- If $c > x$ then $\hat{c} \geq x$ (since RN is an increasing function). We have

$$\left|\text{RN}\left(\frac{x}{\hat{c}}\right) - \frac{x}{\hat{c}}\right| \leq \frac{u}{2},$$

and

$$\left|\frac{x}{c} - \frac{x}{\hat{c}}\right| \leq x \cdot \frac{u}{\hat{c}c} = \left(\frac{x}{c}\right) \cdot \frac{u}{\hat{c}} < 1 \cdot \frac{u}{1+2u},$$

and $\text{ulp}(x/c) = u$, so that

$$\begin{aligned} \left|s - \frac{x}{c}\right| &\leq \left(\frac{1}{2} + \frac{1}{1+2u}\right) \text{ulp}\left(\frac{x}{c}\right) \\ &= \left(\frac{3}{2} - \frac{2u}{1+2u}\right) \text{ulp}\left(\frac{x}{c}\right) \\ &\leq \left(\frac{3}{2} - 2u + 4u^2\right) \text{ulp}\left(\frac{x}{c}\right). \end{aligned}$$

Putting all this together, we obtain

Property 3.1: Assume radix-2, precision- p FP arithmetic. If c is a real constant or a real function of one or more variables, x is a FP number, and $\hat{c} = \text{RN}(c)$ is the correctly-rounded (to nearest) implementation of c then, barring underflow and overflow, the FP number $s = \text{RN}(x/\hat{c})$ satisfies

$$\begin{aligned} \left| s - \frac{x}{c} \right| &\leq \left(\frac{3}{2} - \frac{2u}{1+2u} \right) \text{ulp} \left(\frac{x}{c} \right) \\ &\leq \left(\frac{3}{2} - 2u + 4u^2 \right) \text{ulp} \left(\frac{x}{c} \right). \quad \square \end{aligned}$$

The following example shows that when c is an arbitrary real constant, the bound of Property 3.1 is asymptotically optimal.

Example 2: If p is even and larger than 4, the choice $x = 2 - 2^{-p/2}$ and $c = 1 + 2^{-p/2-1} - 2^{-p}$ gives $\hat{c} = 1 + 2^{-p/2-1}$, $x/c = 2 - 2^{-p/2+1} + 3 \cdot 2^{-p} + O(2^{-3p/2})$ and $\text{RN}(x/\hat{c}) = 2 - 2^{-p/2+1}$, resulting in an error

$$\left| \frac{x}{c} - \text{RN} \left(\frac{x}{\hat{c}} \right) \right| = \left(\frac{3}{2} + O(2^{-p/2}) \right) \text{ulp} \left(\frac{x}{c} \right).$$

If p is odd and larger than 5, the choice $x = 1$ and $c = 1 + 2^{-(p+1)/2} - 2^{-p}$ gives $\hat{c} = 1 + 2^{-(p+1)/2}$, $x/c = 1 - 2^{-(p+1)/2} + 3 \cdot 2^{-p-1} + O(2^{-3p/2})$ and $\text{RN}(x/\hat{c}) = 1 - 2^{-(p+1)/2}$, resulting again in an error

$$\left| \frac{x}{c} - \text{RN} \left(\frac{x}{\hat{c}} \right) \right| = \left(\frac{3}{2} + O(2^{-p/2}) \right) \text{ulp} \left(\frac{x}{c} \right).$$

For instance, if $p = 24$, our example gives

$$c = 16779263/2^{24}, x = 8191/4096,$$

and an error equal to $1.49957 \dots \text{ulp}(x/c)$; if $p = 53$, it gives

$$c = 9007199321849855/2^{53}, x = 1,$$

and an error $1.49999998137 \dots \text{ulp}(x/c)$; and if $p = 113$ it gives

$$\begin{aligned} c &= 10384593717069655329118586696368127/2^{113}, \\ x &= 1, \end{aligned}$$

and an error $1.4999999999999998265 \dots \text{ulp}(x/c)$.

Example 2 establishes the asymptotical optimality when c is a constant, but also gives information for some particular functions. It turns out that the values of c in Example 2 are the same as those in Example 1, so the reasoning is the same:

- since the values of c used in Example 2 can straightforwardly be obtained by adding two FP numbers, we also deduce the asymptotic optimality of the bound of Property 3.1 for the calculation of $z/(x+y)$.
- in the particular cases $p = 24$ (binary32 arithmetic), $p = 53$ (binary64) and $p = 113$ (binary128), factorizations of the particular values of c given in the example show that these values can be obtained by multiplying FP numbers. This immediately shows that errors $1.49957 \dots \text{ulp}(x/c)$ (in single precision), $1.49999998137 \dots \text{ulp}(x/c)$ (in double precision), and $1.4999999999999998265 \dots \text{ulp}(x/c)$ (in quad precision) can be attained when calculating $z/(x*y)$, showing that for that function, the bound of Property 3.1 is very tight.

For other functions, testings in small precisions show that the bounds are tight. For instance, if $p = 24$ (which corresponds to binary32/double-precision arithmetic), error

$1.4959 \text{ulp}(x/\sqrt{y})$ is attained when computing x/\sqrt{y} for $x = 16763899$ and $y = 8396805/2$. For the same function in binary64 arithmetic, error $1.49906 \text{ulp}(x/\sqrt{y})$ is attained for $x = 9007198105271337$ and $y = 4503599631275935/2^{52}$.

IV. DIVIDING A CORRECTLY-ROUNDED FUNCTION BY A FP NUMBER

Now we consider approximating c/x , where x is a FP number and c is either a real constant or a real function of one or more FP variables, by $s = \text{RN}(\hat{c}/x)$ where, as previously, $\hat{c} = \text{RN}(c)$. Since the reasoning is highly similar to that of Section III, we only outline the major points. The cases where c or x are a power of 2 are straightforward, so without loss of generality, we assume $1 < x < 2$ (so that, since x is a FP number, $1 + 2u \leq x \leq 2 - 2u$), and $1 < c < 2$ (which implies $1 \leq \hat{c} \leq 2$).

- if $c < x$ then $\hat{c} \leq x$, and therefore

$$\left| \frac{\hat{c}}{x} - \text{RN} \left(\frac{\hat{c}}{x} \right) \right| \leq \frac{u}{2},$$

and

$$\left| \frac{\hat{c}}{x} - \frac{c}{x} \right| = \frac{1}{x} \cdot |\hat{c} - c| \leq \frac{1}{1+2u} \cdot u,$$

so that

$$\left| s - \frac{c}{x} \right| \leq \frac{3u + 2u^2}{2 + 4u} = \frac{3 + 2u}{2 + 4u} \cdot \text{ulp} \left(\frac{c}{x} \right).$$

- if $c \geq x$ then $\hat{c} \geq x$, and therefore

$$\left| \frac{\hat{c}}{x} - \text{RN} \left(\frac{\hat{c}}{x} \right) \right| \leq u,$$

and

$$\left| \frac{\hat{c}}{x} - \frac{c}{x} \right| = \frac{1}{x} \cdot |\hat{c} - c| \leq \frac{1}{1+2u} \cdot u < u,$$

and therefore

$$\left| s - \frac{c}{x} \right| < 2u = \text{ulp} \left(\frac{c}{x} \right).$$

Putting all this together, we conclude

Property 4.1: Assume radix-2, precision- p FP arithmetic. If c is a real constant or a real function of one or more variables, x is a FP number, and $\hat{c} = \text{RN}(c)$ is the correctly-rounded (to nearest) implementation of c then, barring underflow and overflow, the FP number $s = \text{RN}(\hat{c}/x)$ satisfies

$$\begin{aligned} \left| s - \frac{c}{x} \right| &\leq \frac{3 + 2u}{2 + 4u} \cdot \text{ulp} \left(\frac{c}{x} \right) \\ &\leq \left(\frac{3}{2} - 2u + 4u^2 \right) \text{ulp} \left(\frac{c}{x} \right). \quad \square \end{aligned}$$

Property 4.1 covers calculations such as $\ln(2)/x$, \sqrt{x}/y , $(x+y)/z$, ... In the general case, the bound given by Property 4.1 is very tight. For instance, if $p = 53$, $c = 2^{53} + 1$, and $x = 2^{52} + 2^{25}$, assuming RN breaks ties to even, we obtain $\hat{c} = 2^{53}$ and $s = 134217727/2^{26}$, resulting in an error

$$1.4999999888241291 \dots \text{ulp}(c/x).$$

That example can be transformed into a “generic” example, that shows that the bound of Property 4.1 is asymptotically

optimal for odd values of p . This is done as follows. Let p be an odd integer, let $c = 2^p + 1$ (hence $\hat{c} = 2^p$) and $x = 2^{p-1} + 2^{(p-3)/2}$. We have

$$\frac{\hat{c}}{x} = \frac{2^p}{2^{p-1} + 2^{(p-3)/2}} = 2 \frac{1}{1 + 2^{-(p+1)/2}} = 2(1 - 2^{-(p+1)/2} + \varepsilon_p),$$

with $|\varepsilon_p| < 2^{-(p+1)}$. Therefore, $\text{RN}(\hat{c}/x) = 2(1 - 2^{-(p+1)/2})$ and

$$\begin{aligned} \text{RN}\left(\frac{\hat{c}}{x}\right) - \frac{c}{x} &= 2(1 - 2^{-(p+1)/2}) - \frac{2 + 2^{-(p-1)}}{1 + 2^{-(p+1)/2}} \\ &= 2(1 - 2^{-(p+1)/2}) \\ &\quad - (2 + 2^{-(p-1)})(1 - 2^{-(p+1)/2} + 2^{-(p+1)} + \eta_p), \end{aligned}$$

where $|\eta_p| < 2^{-3(p+1)/2}$. It then follows

$$\begin{aligned} \left| \text{RN}\left(\frac{\hat{c}}{x}\right) - \frac{c}{x} \right| &= \left(\frac{3}{2} - 2^{-(p+1)/2} + 2^{-(p+1)} + \eta_p(1 + 2^p) \right) \text{ulp}\left(\frac{c}{x}\right) \\ &= \left(\frac{3}{2} + 2^{-(p+1)/2} \gamma_p \right) \text{ulp}\left(\frac{c}{x}\right), \end{aligned}$$

where $|\gamma_p| < 3$.

V. PRODUCT OF TWO CORRECTLY-ROUNDED FUNCTIONS

We now consider the approximation of $m \cdot n$, where m and n are either real constants or correctly-rounded functions, by $s = \text{RN}(\hat{m} \cdot \hat{n})$, where $\hat{m} = \text{RN}(m)$ and $\hat{n} = \text{RN}(n)$ (the case where *both* m and n are constants is of course of little interest). To deal with this case, we need the following lemma.

Lemma 5.1: Let x and y be real numbers satisfying $1 \leq x, y \leq 2$. If $xy \leq 2$ then $x + y \leq 3$.

The proof of Lemma 5.1 is simple: $xy \leq 2$ implies $x \leq 2/y$, therefore $x + y \leq 2/y + y$. The number $2/y + y - 3$ has the same sign as $P(y) = y^2 - 3y + 2$ since $y > 0$. The roots of P are 1 and 2 and we immediately deduce that $P(y) \leq 0$ for y between 1 and 2.

Let us now bound the error $|s - mn|$. Without loss of generality, we assume $1 \leq m < 2$ and $1 \leq n < 2$. We have

$$\begin{aligned} |s - mn| &= |\text{RN}(\hat{m} \cdot \hat{n}) - mn| \\ &\leq |\text{RN}(\hat{m} \cdot \hat{n}) - \hat{m} \cdot \hat{n}| + |\hat{m} \cdot \hat{n} - mn|, \end{aligned}$$

and

$$\begin{aligned} |\hat{m} \cdot \hat{n} - mn| &\leq \hat{m} \cdot |\hat{n} - n| + n \cdot |\hat{m} - m| \\ &\leq (\hat{m} + n) \cdot u. \end{aligned}$$

- if $mn \leq 2$ and $\hat{m}\hat{n} \leq 2$ then $|\text{RN}(\hat{m} \cdot \hat{n}) - \hat{m} \cdot \hat{n}| \leq u$. Furthermore, Lemma 5.1 implies $m+n \leq 3$ and therefore $\hat{m} + n \leq 3 + u$. All this gives $|s - mn| \leq 4u + u^2$, and since $\text{ulp}(mn) \geq 2u$, we find $|s - mn| \leq (2 + u/2) \text{ulp}(mn)$;
- if $mn > 2$ then $|\text{RN}(\hat{m} \cdot \hat{n}) - \hat{m} \cdot \hat{n}| \leq 2u$ and $\hat{m} + n \leq 4$, so that $|s - mn| \leq 6u = \frac{3}{2} \text{ulp}(mn)$;

- if $mn \leq 2$ and $\hat{m}\hat{n} > 2$ then $|\text{RN}(\hat{m} \cdot \hat{n}) - \hat{m} \cdot \hat{n}| \leq 2u$ and (from Lemma 5.1) $\hat{m} + n \leq 3 + u$, so that $|s - mn| \leq 5u + u^2$, and this last bound is $(\frac{5}{2} + \frac{u}{2}) \cdot \text{ulp}(mn)$ when $mn < 2$, and $(\frac{5}{4} + \frac{u}{4}) \cdot \text{ulp}(mn)$ when $mn = 2$.

We finally obtain

Property 5.2: Assume radix-2, precision- p FP arithmetic. If m and n are real constants or real functions of one or more variables, $\hat{m} = \text{RN}(m)$ and $\hat{n} = \text{RN}(n)$ then, barring underflow and overflow, the FP number $s = \text{RN}(\hat{m} \cdot \hat{n})$ satisfies

$$|s - mn| \leq \left(\frac{5}{2} + \frac{u}{2} \right) \text{ulp}(mn). \quad \square$$

The bound given by Property 5.2 is very tight. It is asymptotically optimal for even values of p (this is shown when RN breaks ties to even by considering $m = 2^p + 2^{p/2} - 1$ and $n = 2^{p+1} - 2^{p/2+1} + 3$, for which we obtain $\hat{m} = 2^p + 2^{p/2}$, $\hat{n} = 2^{p+1} - 2^{p/2+1} + 4$, $mn = 2^{2p+1} - 2^p + 5 \cdot 2^{p/2} - 3$ and $s = 2^{2p+1} + 2^{p+2}$. With another tie-breaking rule, adding a very small $\epsilon > 0$ to m and n gives the same \hat{m} and \hat{n} and the same s). We conjecture that this is also the case for odd values of p but we have no proof. For instance, in binary32 arithmetic ($p = 24$), the error committed when computing $(x+y) \cdot (z+t)$, where x, y, z , and t are the 4 FP numbers defined as follows:

$$\begin{cases} x &= 2^{24} = 16777216; \\ y &= 2^{12} - 1 = 4095; \\ z &= 2^{25} - 2^{13} = 33546240; \\ t &= 3; \end{cases}$$

is equal to

$$2.4993897 \dots \text{ulp}((x+y) \cdot (z+t))$$

(this corresponds to Property 5.2, with $m = x + y$ and $n = z + t$). Even if we have no proof of asymptotic optimality for odd p , we have found cases for which the error is very close to the bound. For instance if $p = 53$, with $m = 9007199877083003$ and $n = 18014397264798047$, the error is $2.4999982 \text{ulp}(mn)$. As m is the product of the two binary64 numbers $e = 290554834744613$ and $f = 31$, and n is the product of the two binary64 numbers $g = 29$ and $h = 621186112579243$, error $2.4999982 \text{ulp}(efgh)$ is attained when computing $(e * f) * (g * h)$ in binary64/double-precision arithmetic, which is very close to the bound of Property 5.2. Property 5.2 covers calculations such as $\pi \cdot \sqrt{x}$, $(x+y) \cdot (z+t)$, etc. If an FMA instruction is available, it also covers computations of the form

$$(ax + b)(cy + d),$$

where a, b, c, d, x , and y are FP numbers.

VI. QUOTIENT OF TWO CORRECTLY-ROUNDED FUNCTIONS

We finally consider the approximation of n/d , where n and d are either real constants or correctly-rounded functions, by $s = \text{RN}(\hat{n}/\hat{d})$, where $\hat{n} = \text{RN}(n)$ and $\hat{d} = \text{RN}(d)$ (the case where *both* n and d are constants is of course of little interest). We assume that n and d are not FP numbers (that case is covered by Sections III and IV), so that without loss of

generality, we can assume $1 < n < 2$ and $1 < d < 2$, which implies $|\hat{n} - n| \leq u$ and $|\hat{d} - d| \leq u$. We have

$$\begin{aligned} \left| \frac{\hat{n}}{\hat{d}} - \frac{n}{d} \right| &\leq \left| \frac{\hat{n}}{\hat{d}} - \frac{n}{\hat{d}} \right| + \left| \frac{n}{\hat{d}} - \frac{n}{d} \right| \\ &= \frac{1}{\hat{d}} \cdot |\hat{n} - n| + \frac{n}{d} \cdot \left| \frac{\hat{d} - d}{\hat{d}} \right| \\ &\leq \left(1 + \frac{n}{d} \right) \cdot \frac{u}{\hat{d}}. \end{aligned}$$

If $\hat{n} = \hat{d}$ or $\hat{n} = 2\hat{d}$ or $\hat{n} = \hat{d}/2$ then $s = \hat{n}/\hat{d}$, and otherwise $n < d \Rightarrow \hat{d}/2 < \hat{n} < \hat{d}$ and $n > d \Rightarrow \hat{n} > \hat{d} > \hat{n}/2$ so that $\text{ulp}(\hat{n}/\hat{d}) = \text{ulp}(n/d)$. Therefore, in all cases,

$$\left| s - \frac{\hat{n}}{\hat{d}} \right| \leq \frac{1}{2} \text{ulp} \left(\frac{n}{d} \right).$$

Putting all this together, we obtain,

$$\left| s - \frac{n}{d} \right| \leq \frac{1}{2} \text{ulp} \left(\frac{n}{d} \right) + \left(1 + \frac{n}{d} \right) \cdot \frac{u}{\hat{d}},$$

which immediately gives:

- if $n \geq d$ then $\text{ulp}(n/d) = 2u$ and $(1 + \frac{n}{d}) \cdot \frac{u}{\hat{d}} \leq 3u$ so that

$$\left| s - \frac{n}{d} \right| \leq 2 \text{ulp} \left(\frac{n}{d} \right);$$

- if $n < d$ then $\text{ulp}(n/d) = u$ and $(1 + \frac{n}{d}) \cdot \frac{u}{\hat{d}} \leq 2u$ so that

$$\left| s - \frac{n}{d} \right| \leq \frac{5}{2} \text{ulp} \left(\frac{n}{d} \right).$$

We therefore obtain

Property 6.1: Assume radix-2, precision- p FP arithmetic. If n and d are real constants or real functions of one or more variables, $\hat{n} = \text{RN}(n)$ and $\hat{d} = \text{RN}(d)$ then, barring underflow and overflow, the FP number $s = \text{RN}(\hat{n}/\hat{d})$ satisfies

$$\left| s - \frac{n}{d} \right| \leq \frac{5}{2} \text{ulp} \left(\frac{n}{d} \right). \quad \square$$

Property 6.1 covers calculations such as π/\sqrt{x} , $(x+y)/(z+t)$, $(xy)/(z+t)$, etc. If an FMA instruction is available, it also covers computations of the form

$$\frac{ax + b}{cy + d},$$

where a, b, c, d, x , and y are FP numbers.

The bound given by Property 6.1 is very tight. In the general case, assuming RN breaks ties to even, it is asymptotically optimal, as one may check using the generic values

- if p is odd: $n = 2^p + 1$ and $d = 2^p + 2^{(p-1)/2} - 1$;
- if p is even: $n = 2^p + 2^{p/2-1} + 1$ and $d = 2^p + 2^{p/2} - 1$.

Let us quickly detail the case p odd: let $v = 2^{-(p+1)/2}$ so that $v^2 = 2^{-p+1} = 2u$. We have

$$\frac{n}{d} = \frac{1 + 2^{-p}}{1 + 2^{-(p-1)/2} - 2^{-p}} = \frac{1 + v^2/2}{1 + v/2 - v^2/2}.$$

From $\hat{n} = 2^p$ and $\hat{d} = 2^p + 2^{(p-1)/2}$ we obtain

$$\frac{\hat{n}}{\hat{d}} = \frac{1}{1 + v/2} = 1 - v/2 + v^2/4 - v^4/8 + \dots,$$

from which we obtain $\text{RN}(\hat{n}/\hat{d}) = 1 - v/2$. Hence the error is

$$\begin{aligned} \left| \left(1 - \frac{v}{2} \right) - \left(\frac{1 + \frac{v^2}{2}}{1 + \frac{v}{2} - \frac{v^2}{2}} \right) \right| &= \frac{5}{4}v^2 + O(v^3) \\ &= \left(\frac{5}{2} + O(u^{3/2}) \right) \text{ulp} \left(\frac{n}{d} \right). \end{aligned}$$

For instance, consider the computation of $(x+y)/(z+t)$ in binary64 arithmetic, where x, y, z and t are the four following binary64 numbers:

$$\begin{cases} x = 2^{53}, \\ y = 1, \\ z = 2^{53}, \\ t = 2^{26} - 1. \end{cases}$$

This corresponds to Property 6.1, with $n = x+y$ and $d = z+t$. We have

$$\frac{x+y}{z+t} = \frac{9007199254740993}{9007199321849855} = 0.999999992549419680631925 \dots$$

and the computed result is

$$\frac{134217727}{134217728} = 0.999999992549419403076 \dots,$$

resulting in an error

$$2.49999997392 \dots \text{ulp} \left(\frac{x+y}{z+t} \right).$$

Interestingly enough, these values of $n = 2^{53} + 1$ and $d = 2^{53} + 2^{26} - 1$ can be written as the product of two binary64 FP numbers, since $n = 321. \times 28059810762433.$ and $d = 34292630015. \times 262657..$ As a consequence, the same error 2.49999997392...ulp is also attained for computations of the form $(xy)/(z+t)$; $(x+y)/(zt)$; and $(xy)/(zt)$ in binary64 arithmetic. For other functions, we also found cases where the actual error is very close to the bound. For instance, in binary64 floating-point arithmetic ($p = 53$), error 2.4994 ulp is attained when computing

$$\frac{x_1 + y_1}{\sqrt{z}} \quad \text{or} \quad \frac{x_2 y_2}{\sqrt{z}},$$

with

$$\begin{cases} x_1 = 9007199312857556, \\ y_1 = 1, \\ x_2 = 1870953, \\ y_2 = 4814230669, \\ z = 4503599859833552, \end{cases}$$

(these two cases correspond to Property 6.1 with $n = x_1 + y_1 = x_2 y_2$ and $d = \sqrt{z}$)

CONCLUSION

We have given sharp error bounds in ulps for computations of the form $x \cdot c$, x/c , c/x , $m \cdot n$ and n/d , where x is a FP number and c, n, m and d are either real constants or correctly-rounded functions of one or more variables. Examples of functions for which our work gives tight bounds are $x * \text{pi}$, $\ln(2)/x$, $x/(y+z)$, $(x+y) * z$, $x/\text{sqrt}(y)$, $\text{sqrt}(x)/y$, $(x+y)(z+t)$, $(x+y)/(z+t)$, $(x+y)/(zt)$, etc. In several cases, we have been able to show that our bounds are asymptotically optimal.

ACKNOWLEDGEMENT

This work is partly sponsored by the ANR Nuscap (ANR-20-CE48-0014) project of the French Agence Nationale de la Recherche.

REFERENCES

- [1] N. Brisebarre and J.-M. Muller, “Correctly rounded multiplication by arbitrary precision constants,” *IEEE Trans. Comput.*, vol. 57, no. 2, pp. 165–174, Feb. 2008.
- [2] IEEE, *IEEE Standard for Floating-Point Arithmetic (IEEE Std 754-2019)*, Jul. 2019. [Online]. Available: <https://ieeexplore.ieee.org/servlet/opac?punumber=8766227>
- [3] S. M. Rump, T. Ogita, and S. Oishi, “Accurate floating-point summation, Part I: Faithful rounding,” *SIAM J. Sci. Comput.*, vol. 31, no. 1, pp. 189–224, 2008. [Online]. Available: <https://doi.org/10.1137/050645671>
- [4] A. Sibidanov, P. Zimmermann, and S. Gloudu, “The CORE-MATH project,” in *29th IEEE Symposium on Computer Arithmetic*, 2022, preprint available at <https://hal.inria.fr/hal-03721525>.
- [5] S. Boldo, C.-P. Jeannerod, G. Melquiond, and J.-M. Muller, “Floating-point arithmetic,” *Acta Numer.*, 2023, to appear.
- [6] D. E. Knuth, *The Art of Computer Programming*, 3rd ed. Addison-Wesley, Reading, MA, 1998, vol. 2.
- [7] P. H. Sterbenz, *Floating-Point Computation*. Englewood Cliffs, NJ: Prentice-Hall, 1974.
- [8] C.-P. Jeannerod and S. M. Rump, “On relative errors of floating-point operations: optimal bounds and applications,” *Mathematics of Computation*, vol. 87, pp. 803–819, 2018.
- [9] J.-M. Muller, N. Brunie, F. de Dinechin, C.-P. Jeannerod, M. Joldes, V. Lefèvre, G. Melquiond, N. Revol, and S. Torres, *Handbook of Floating-Point Arithmetic, 2nd edition*. Birkhäuser Boston, 2018, ACM G.1.0; G.1.2; G.4; B.2.0; B.2.4; F.2.1., ISBN 978-3-319-76525-9.



Joris Picot was born in Troyes, France, in 1985. He received his Ph.D. degree in 2013 from the Université de Toulouse. He spent seven years in the development of computational fluid dynamics software, and he is Research Engineer at École Normale Supérieure de Lyon, France. His research interests are in Rigorous Computing.



Nicolas Brisebarre was born in Bordeaux, France, in 1971. He received his Ph.D. in pure mathematics from the Université Bordeaux I, France, in 1998. He is *Chargé de recherche* (junior researcher) at CNRS, France, and he is a member of the LIP laboratory (LIP is a joint computer science laboratory of CNRS, École Normale Supérieure de Lyon, INRIA, and Université Claude Bernard Lyon 1). His research interests are in computer arithmetic, number theory, validated computing and computer algebra.



Jean-Michel Muller was born in Grenoble, France, in 1961. He received his Ph.D. degree in 1985 from the Institut National Polytechnique de Grenoble. He is *Directeur de Recherches* (senior researcher) at CNRS, France, and he is the co-head of GDR-IM. His research interests are in Computer Arithmetic. He is the author of several books, including “Elementary Functions, Algorithms and Implementation” (3rd edition, Birkhäuser, 2016), and he coordinated the writing of the “Handbook of Floating-Point Arithmetic” (2nd edition Birkhäuser, 2018). He is currently associate editor in chief of the IEEE Transactions on Emerging Topics in Computing. He is a fellow of the IEEE.