The Classical Relative Error Bounds for Computing $\sqrt{a2 + b2}$ and $c/\sqrt{a2 + b2}$ in Binary Floating-Point Arithmetic are Asymptotically Optimal

Claude-Pierre Jeannerod, Jean-Michel Muller, Antoine Plet

The Classical Relative Error Bounds for Computing $\sqrt{a^2+b^2}$ and $c/\sqrt{a^2+b^2}$ in Binary Floating-Point Arithmetic are Asymptotically Optimal

Claude-Pierre Jeannerod*, Jean-Michel Muller†, and Antoine Plet‡

*Inria, †CNRS, ‡ENS de Lyon
Université de Lyon – Laboratoire LIP (CNRS, Inria, ENS de Lyon, UCBL), Lyon, France

***Abstract*—We study the accuracy of classical algorithms for evaluating expressions of the form $\sqrt{a^2+b^2}$ and $c/\sqrt{a^2+b^2}$ in radix-2, precision-p floating-point arithmetic, assuming that the elementary arithmetic operations $\pm$, $\times$, $/$, $\sqrt{}$ are rounded to nearest, and assuming an unbounded exponent range.**

Classical analyses show that the relative error is bounded by $2u+\mathcal{O}(u^2)$ for $\sqrt{a^2+b^2}$, and by $3u+\mathcal{O}(u^2)$ for $c/\sqrt{a^2+b^2}$, where $u = 2^{-p}$ is the unit roundoff. Recently, it was observed that for $\sqrt{a^2+b^2}$ the $\mathcal{O}(u^2)$ term is in fact not needed [1]. We show here that it is not needed either for $c/\sqrt{a^2+b^2}$. Furthermore, we show that these error bounds are asymptotically optimal. Finally, we show that both the bounds and their asymptotic optimality remain valid when an FMA instruction is used to evaluate a^2+b^2.

***Index Terms*—binary floating-point arithmetic; rounding error analysis; relative error; hypotenuse function**

I. Introduction

Expressions of the form $\sqrt{a^2+b^2}$ and $c/\sqrt{a^2+b^2}$ are basic building blocks of numerical computing that frequently appear in calculations (computation of 2D-norms, Givens rotations, etc.). In this paper, we give tight error bounds for the evaluation of these expressions.

In the following, we assume a radix-2, precision-p floating-point (FP) arithmetic, with unlimited exponent range. (Hence in practice all our results apply to IEEE floating-point arithmetic [2] as long as underflow and overflow do not occur). Here, an FP number of exponent e and integral significand M is a number of the form $M\cdot 2^{e-p+1}$, where M and e are integers, and $|M| \leqslant 2^p-1$.

Throughout the paper, $\mathrm{RN}(t)$ means t rounded to a nearest FP number, ties-to-even (so that, for example, $\mathrm{RN}(a^2)$ is the result of the floating-point multiplication `a*a`, assuming the round-to-nearest mode), $\mathrm{RD}(t)$ is t rounded towards $-\infty$, and $u = 2^{-p}$ denotes the unit roundoff. We have in particular the following relative error bounds: $\mathrm{RN}(t) = t(1+\epsilon)$ with $|\epsilon| \leqslant \frac{u}{1+u} < u$, and $\mathrm{RD}(t) = t(1+\epsilon')$ with $|\epsilon'| < 2u$; see [3], [4].

It is well known (see for example [5]) that the relative errors of the classical algorithms for $\sqrt{a^2+b^2}$ and $c/\sqrt{a^2+b^2}$ can be bounded by $2u+\mathcal{O}(u^2)$ and $3u+\mathcal{O}(u^2)$, respectively, as $u \to 0$. We consider algorithms that only use floating-point addition, multiplication, and square root, as well as algorithms that take advantage of the possible availability of a fused multiply-add (FMA) instruction to speed up the calculation of a^2+b^2.

Recently, it was shown in [1] that the $\mathcal{O}(u^2)$ term is not needed for $\sqrt{a^2+b^2}$ (that is, the relative error is always bounded by $2u$). In the following, we show that it is not needed either for $c/\sqrt{a^2+b^2}$. Furthermore, we show that these error bounds are asymptotically optimal, and that such optimality results are not impacted by the use of the FMA.

Let us clarify what we mean by "asymptotically optimal." When giving for some algorithm a relative error bound that is a function $B(p)$ of the precision p,

- if we can show that there exist some FP inputs parameterized by p and for which the bound is attained for every $p \geqslant p_0$ (for a given $p_0 \geqslant 2$), then we say that the bound is *optimal*;
- if we can show that there exist some FP inputs parameterized by p and for which the relative error $E(p)$ satisfies $E(p)/B(p) \to 1$ as $p \to \infty$, then we say that the bound is *asymptotically optimal*.

Knowing that a bound is asymptotically optimal has its importance: it shows that there is no need to try to obtain a substantially better bound.

The paper is organized as follows. Section II deals with the computation of $\sqrt{a^2+b^2}$ by the classical method (Algorithm 1) or its FMA-based variant (Algorithm 2). We recall the relative error bound $2u$ from [1] and prove its asymptotic optimality. In Section III, we bound by $3u$ the relative error of classical methods for $c/\sqrt{a^2+b^2}$ (Algorithm 3 and, with an FMA, Algorithm 4), and show that this bound is asymptotically optimal.

II. Computation of $\sqrt{a^2+b^2}$

A. Error bound

Given two FP numbers a and b, we consider the evaluation of $\sqrt{a^2+b^2}$ by means of one of the following two algorithms (without or with an FMA):

Algorithm 1 Computation of $\sqrt{a^2+b^2}$ in binary precision-p floating-point arithmetic.

$s_a \leftarrow \mathrm{RN}(a^2)$
$s_b \leftarrow \mathrm{RN}(b^2)$
$s \leftarrow \mathrm{RN}(s_a + s_b)$
$\rho \leftarrow \mathrm{RN}(\sqrt{s})$
return ρ

Algorithm 2 FMA-based computation of $\sqrt{a^2+b^2}$ in binary precision-p floating-point arithmetic.

$s_b \leftarrow \mathrm{RN}(b^2)$
$s \leftarrow \mathrm{RN}(a^2 + s_b)$
$\rho \leftarrow \mathrm{RN}(\sqrt{s})$
return ρ

As we shall see, these two algorithms have their relative error bounded by $2u$ and this bound is asymptotically optimal. Incidentally, this means that both algorithms are rather equivalent in terms of *worst-case* error. This does not necessarily mean that they are *always* equivalent: the result of Algorithm 2 tends to be more often closer to the exact result than the result of Algorithm 1. To compare both algorithms, we have performed experiments in binary64 IEEE arithmetic and obtained the following results.

For $1,000,000$ randomly chosen pairs (a,b) of binary64 FP numbers with the same exponent:

- both algorithms yield the same error in 90.08% of cases;
- Algorithm 2 is more accurate in 6.26% of cases;
- Algorithm 1 is more accurate in 3.65% of cases.

For $100,000$ randomly chosen pairs (a,b) of binary64 FP numbers with respective exponents e_a and e_b satisfying $e_a - e_b = -26$, we always have the same error with both algorithms.

For $100,000$ randomly chosen pairs (a,b) of binary64 FP numbers with $e_a - e_b = +26$:

- both algorithms yield the same error in 83.90% of cases;
- Algorithm 2 is more accurate in 13.79% of cases;
- Algorithm 1 is more accurate in 2.32% of cases.

Let us now focus on the bound on the relative error of Algorithms 1 and 2. As said above, it is well known that this relative error is upper bounded by $2u + \mathcal{O}(u^2)$. Recently, the error analysis of the most common FP operations was revisited in [1] and it was noted in particular that the relative error of Algorithm 1 is at most $2u$ (that is, there is no $\mathcal{O}(u^2)$ term). The proof from [1] is easily adapted to Algorithm 2: the same bound holds for that algorithm. In practice, this bound is tight. For example, in IEEE binary64 arithmetic ($p = 53$), with the two FP numbers $a = 1723452922282957/2^{64}$ and $b = 4503599674823629/2^{52}$, Algorithms 1 and 2 return the same result, which corresponds to a relative error of the form $1.99999993022\ldots u$.

B. Asymptotic optimality of the bound

As said above, what we are interested in is to show that the bound $2u$ is asymptotically optimal. We establish the following result.

Theorem II.1. *For $p \geqslant 12$, there exist floating-point inputs a and b for which the result ρ of Algorithm 1 or Algorithm 2 satisfies*

$$\left|\frac{\rho - \sqrt{a^2+b^2}}{\sqrt{a^2+b^2}}\right| = 2u - \epsilon, \quad |\epsilon| = \mathcal{O}(u^{3/2}).$$

Before giving the values a and b of Theorem II.1 and proving that they lead to an error asymptotically equivalent to $2u$, let us explain how they are built. We mainly focus on Algorithm 1 (but we also give the details specific to Algorithm 2):

1) We restrict to a and b such that $0 < a < b$.
2) b is chosen such that the largest possible absolute error (that is, half an ulp of b^2) is committed when computing b^2. To make sure that this absolute error leads to a relative error that is large enough asymptotically (that is, of the order of u), b must also be slightly above a power of 2. We choose
   - $b = 1 + 2^{-p/2}$ if p is even;
   - $b = 1 + \left\lceil \sqrt{2} \cdot 2^{\frac{p-3}{2}} \right\rceil \cdot 2^{-p+1}$ if p is odd.

   (Notice that although these choices of b seem very different, they can be viewed as the same value $1 + \lceil 2^{p/2-1} \rceil \cdot 2^{-p+1}$.)
3) In Algorithm 1, we have $s_a \leqslant s_b$ since $a < b$. When adding s_a and s_b, the significand of s_a is right-shifted by a number of positions equal to the difference of their exponents. One can find the form s_a should have in order to produce a relative error that is large enough; this is illustrated in Figure 1.
4) We just choose a equal to the square root of that value, adequately rounded.

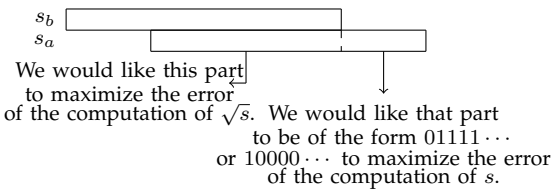

Fig. 1. Some objectives when constructing suitable generic inputs to Algorithms 1 and 2.

1) If p is even: As said above, we choose

$$b = 1 + 2^{-p/2},$$

which due to the ties-to-even rule gives

$$s_b = 1 + 2^{1-p/2}.$$

Note that s_b is b^2 rounded *down*. To force the rounding errors to accumulate (and not compensate), we must ensure that $s_a + s_b$ and $\sqrt{s}$ are rounded down as well. Define

$$G = \left\lceil 2^{\frac{p}{2}} \left(\sqrt{2} - 1\right) + \delta \right\rceil \cdot 2^{\frac{p}{2}+1} + 2^{\frac{p}{2}}$$

with

$$\delta = \begin{cases} 1 & \text{if } \left\lceil 2^{\frac{p}{2}}\sqrt{2} \right\rceil \text{ is odd,} \\ 2 & \text{otherwise,} \end{cases} \tag{1}$$

and let

$$a = \mathrm{RD}\left(2^{-\frac{3p}{4}}\sqrt{G}\right).$$

By definition of RD, this implies that a^2 satisfies

$$2^{-\frac{3p}{2}} G \left(1 - 2^{-p+1}\right)^2 < a^2 \leqslant 2^{-\frac{3p}{2}} G.$$

Now, for $p \geqslant 12$, we have $2^{p-1} < G < 2^p$, from which it follows that $2^{-\frac{3p}{2}} G$ is an FP number and that $2^{-\frac{3p}{2}} G \left(1 - 2^{-p+1}\right)^2$ is lower bounded by $2^{-\frac{3p}{2}}(G - 4)$, which is also an FP number:

$$2^{-\frac{3p}{2}}(G-4) < a^2 \leqslant 2^{-\frac{3p}{2}} G. \tag{2}$$

Using the monotonicity of RN, we deduce that the variable $s_a = \mathrm{RN}(a^2)$ of Algorithm 1 satisfies

$$2^{-\frac{3p}{2}}(G-4) \leqslant s_a \leqslant 2^{-\frac{3p}{2}} G.$$

This means that s_a lies between the two FP numbers whose binary expansions are as in Figures 2 and 3.

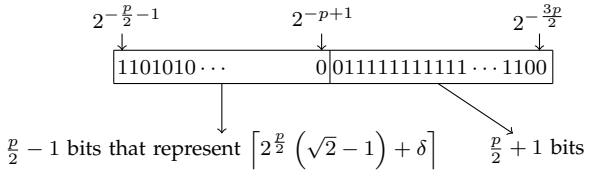

Fig. 2. Bit pattern of the lower bound on s_a.

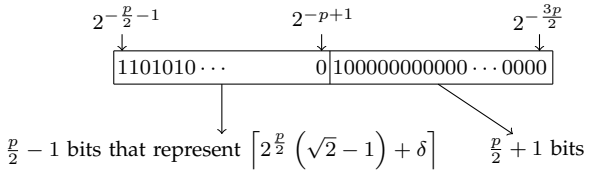

Fig. 3. Bit pattern of the upper bound on s_a.

Note that the choice of δ in (1) implies that the bit of s_a of weight 2^{-p+1} is always zero. Hence, thanks to the ties-to-even rule, $s_a + s_b$ is always rounded *down* to $s = \mathrm{RN}(s_a + s_b)$, whose binary expansion is as in Figure 4.

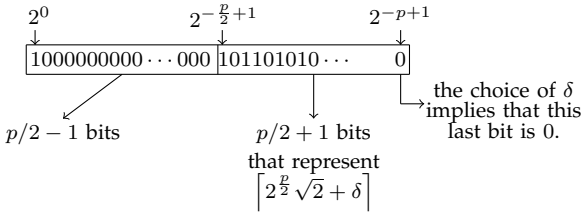

Fig. 4. Bit pattern of s.

Consequently, $s = \mathrm{RN}(s_a + s_b)$ has the form

$$s = 1 + \omega, \qquad \omega = 2^{-p+1} \left\lceil 2^{p/2}\sqrt{2} + \delta \right\rfloor. \tag{3}$$

For Algorithm 2, we can deduce in exactly the same way, by replacing s_a with a^2 and by using (2), that $\mathrm{RN}(a^2 + s_b)$ also has the form (3). Hence, for our chosen values of a and b, Algorithm 2 will return the same final value ρ as Algorithm 1.

Let us now give an explicit formula for that value ρ. First, notice that $1 + \frac{\omega}{2}$ is an FP number thanks to the choice of δ in (1). Since $\omega > 0$ and since $\delta \leqslant 2$ and $p \geqslant 12$, this implies further $1 + 2^{-p+1} \leqslant 1 + \frac{\omega}{2} < 2$. Then, writing f to denote the FP predecessor of $1 + \omega/2$, we have

$$f = 1 + \frac{\omega}{2} - 2^{-p+1}, \tag{4}$$

which is an FP number in $[1, 2)$. Third, we will show now that the exact square root $\sqrt{1+\omega}$ satisfies

$$f \leqslant \sqrt{1+\omega} < f + 2^{-p}. \tag{5}$$

Since $u = 2^{-p}$, for the lower bound it suffices to check that $(1 + \frac{\omega}{2} - 2u)^2 \leqslant 1 + \omega$, or, equivalently, that the polynomial $P(x) = \frac{1}{4}x^2 - 2ux - 4u + 4u^2$ satisfies $P(\omega) \leqslant 0$. The latter inequality holds, since the roots of $P(x)$ are $\pm 4\sqrt{u} + 4u$ and since

$$\begin{aligned} \omega &= 2u\left\lceil \sqrt{2}/\sqrt{u} + \delta \right\rceil \\ &< 2\sqrt{2}\sqrt{u} + 6u \qquad \text{for } \delta \leqslant 2, \\ &\leqslant 4\sqrt{u} + 4u \qquad \text{for } p \geqslant 2. \end{aligned}$$

To establish the upper bound in (5), it suffices to check that $1 + \omega < (1 + \frac{\omega}{2} - u)^2$, or, equivalently, that $Q(x) = \frac{1}{4}x^2 - ux - 2u + u^2$ satisfies $Q(\omega) > 0$. This is true since the roots of $Q(x)$ are $\pm 2\sqrt{2}\sqrt{u} + 2u$ and since $\omega > 2\sqrt{2}\sqrt{u} + 2u$ for $\delta \geqslant 1$ and p even.

From (5) we deduce that rounding *down* occurs when evaluating $\rho = \mathrm{RN}(\sqrt{s})$, so that

$$\rho = 1 + \frac{\omega}{2} - 2^{-p+1}. \tag{6}$$

Finally, to estimate the relative error of this approximation ρ, we must determine (or, at least, adequately approximate) the exact value $\sqrt{a^2 + b^2}$. From $b = 1 + 2^{-p/2}$

TABLE I
RELATIVE ERRORS OF ALGORITHM 1 OR ALGORITHM 2 FOR OUR GENERIC VALUES a AND b OF SECTION II-B1 AND FOR VARIOUS *even* VALUES OF p BETWEEN 16 AND 64.

p	relative error
16	$1.\mathbf{97519352187392\ldots}u$
20	$1.\mathbf{99418559548869\ldots}u$
24	$1.\mathbf{99873332158282\ldots}u$
28	$1.\mathbf{99967582969338\ldots}u$
32	$1.\mathbf{99990783760560\ldots}u$
36	$1.\mathbf{99997442258505\ldots}u$
40	$1.\mathbf{99999449547633\ldots}u$
44	$1.\mathbf{99999835799502\ldots}u$
48	$1.\mathbf{99999967444005\ldots}u$
52	$1.\mathbf{99999989989669\ldots}u$
56	$1.\mathbf{99999997847972\ldots}u$
60	$1.\mathbf{99999999397377\ldots}u$
64	$1.\mathbf{99999999}849587\ldots u$

and (2) we deduce that

$$1+2^{-p/2+1}+2^{-p}+2^{-3p/2}(G-4)<a^2+b^2 \leqslant 1+2^{-p/2+1}+2^{-p}+2^{-3p/2}G.$$

Replacing G by its expression $\lceil 2^{\frac{p}{2}}\left(\sqrt{2}-1\right)+\delta\rceil\cdot 2^{\frac{p}{2}+1}+2^{\frac{p}{2}}$ and using (3), we obtain

$$1+\omega+2^{-p+1}-2^{-3p/2+2}<a^2+b^2\leqslant 1+\omega+2^{-p+1}.$$

From this, we deduce that for $p\to\infty$,

$$\sqrt{a^2+b^2}=1+\frac{\omega}{2}+2^{-p}-\frac{\omega^2}{8}+\mathcal{O}\big(2^{-3p/2}\big)$$

and, noting that (3) implies $\frac{\omega^2}{8}=2^{-p}+\mathcal{O}\big(2^{-3p/2}\big)$,

$$\sqrt{a^2+b^2}=1+\frac{\omega}{2}+\mathcal{O}\big(2^{-3p/2}\big). \tag{7}$$

Combining (6) and (7) gives $\left|\rho-\sqrt{a^2+b^2}\right|=2^{-p+1}+\mathcal{O}\big(2^{-3p/2}\big)$ and, on the other hand, it follows from (3) and (7) that $\sqrt{a^2+b^2}=1+\mathcal{O}\big(2^{-p/2}\big)$. Hence, recalling that $u=2^{-p}$, we conclude that the relative error has the form $2u+\mathcal{O}(u^{3/2})$ as $u\to 0$. This terminates the proof of Theorem II.1 when p is even. □

Table I gives the relative errors obtained with the values a and b of our generic example for various *even* values of p between 16 and 64. We see that the a priori bound $2u$ is tight already for reasonably small precisions.

2) If p is odd: As stated above, we choose

$$b=1+\eta,\qquad \eta=\left\lceil\sqrt{2}\cdot 2^{\frac{p-3}{2}}\right\rceil\cdot 2^{-p+1}. \tag{8}$$

From $\sqrt{2}\cdot 2^{\frac{-p-1}{2}}<\eta<\sqrt{2}\cdot 2^{\frac{-p-1}{2}}+2^{-p+1}$, we easily deduce that

$$2^{-p}<\eta^2<2^{-p}+\sqrt{2}\cdot 2^{\frac{-3p+3}{2}}+2^{-2p+2}. \tag{9}$$

Since $b=1+\eta$, it follows from (9) that the number $b^2=1+2\eta+\eta^2$ is slightly above $1+2\eta+2^{-p}$, that is, slightly above the middle of the two consecutive FP numbers $1+2\eta$ and $1+2\eta+2^{-p+1}$. This implies that rounding *up* occurs when computing $s_b=\mathrm{RN}(b^2)$:

$$s_b=1+2\eta+2^{-p+1}. \tag{10}$$

Now define

$$H=2^{\frac{-p+3}{2}}-2\eta-3\cdot 2^{-p}+2^{\frac{-3p+3}{2}}. \tag{11}$$

Notice that H is close to $(1-1/\sqrt{2})\cdot 2^{(-p+3)/2}$. We choose

$$a=\mathrm{RN}\big(\sqrt{H}\,\big). \tag{12}$$

By definition of RN, this implies

$$H\cdot(1-2^{-p})^2<a^2<H\cdot(1+2^{-p})^2. \tag{13}$$

Hence, the variable $s_a=\mathrm{RN}(a^2)$ of Algorithm 1 satisfies

$$H\cdot(1-2^{-p})^3<s_a<H\cdot(1+2^{-p})^3. \tag{14}$$

This gives (as soon as $p\geqslant 2$)

$$H\cdot(1-3\cdot 2^{-p})<s_a<H\cdot(1+4\cdot 2^{-p}),$$

or, equivalently, using (11),

$$\begin{aligned}2^{\frac{-p+3}{2}}-2\eta-3\cdot 2^{-p}-2\cdot 2^{\frac{-3p+3}{2}}&\\ +6\eta\cdot 2^{-p}+9\cdot 2^{-2p}-3\cdot 2^{\frac{-5p+3}{2}}&\\ <s_a<2^{\frac{-p+3}{2}}-2\eta-3\cdot 2^{-p}+5\cdot 2^{\frac{-3p+3}{2}}&\\ -8\eta\cdot 2^{-p}-12\cdot 2^{-2p}+4\cdot 2^{\frac{-5p+3}{2}}.&\end{aligned}$$

Hence, recalling that $s_b=1+2\eta+2\cdot 2^{-p}$,

$$\begin{aligned}1+2^{\frac{-p+3}{2}}-2^{-p}+\varphi(\eta)<s_a+s_b&\\ <1+2^{\frac{-p+3}{2}}+\psi(\eta),&\end{aligned}$$

where φ and ψ depend linearly on η as follows:

$$\varphi(\eta)=-2\cdot 2^{\frac{-3p+3}{2}}+6\eta\cdot 2^{-p}+9\cdot 2^{-2p}-3\cdot 2^{\frac{-5p+3}{2}}$$

and

$$\psi(\eta)=-2^{-p}+5\cdot 2^{\frac{-3p+3}{2}}-8\eta\cdot 2^{-p}-12\cdot 2^{-2p}+4\cdot 2^{\frac{-5p+3}{2}}.$$

Now, using the definition of η in (8), it is easy to check that $\psi(\eta)<0<\varphi(\eta)$ as soon as $p\geqslant 3$. We then deduce immediately that

$$1+2^{\frac{-p+3}{2}}-2^{-p}<s_a+s_b<1+2^{\frac{-p+3}{2}}.$$

Hence rounding *up* occurs when evaluating $s=\mathrm{RN}(s_a+s_b)$, that is,

$$s=1+2^{\frac{-p+3}{2}}.$$

The enclosure in (14) being weaker than the one in (13), we can replace s_a with a^2 in the derivation above and conclude that $\mathrm{RN}(a^2+s_b)$ is also equal to $1+2^{\frac{-p+3}{2}}$. Consequently, for our choice of a and b both Algorithms 1 and 2 yield the same intermediate value $s=1+2^{\frac{-p+3}{2}}$ and thus the same final value $\rho=\mathrm{RN}(\sqrt{s})$.

TABLE II
RELATIVE ERRORS OF ALGORITHM 1 OR ALGORITHM 2 FOR OUR GENERIC VALUES a AND b OF SECTION II-B2 AND FOR VARIOUS *odd* VALUES OF p BETWEEN 53 AND 113.

p	relative error
53	$1.\mathbf{9999999}188175005308\ldots u$
57	$1.\mathbf{99999997}64537355319\ldots u$
61	$1.\mathbf{999999994}9811629228\ldots u$
65	$1.\mathbf{9999999988}096732861\ldots u$
69	$1.\mathbf{99999999970}55095283\ldots u$
73	$1.\mathbf{999999999}181918151\ldots u$
77	$1.\mathbf{9999999998}00815518\ldots u$
81	$1.\mathbf{99999999995}4499727\ldots u$
85	$1.\mathbf{99999999998}7674393\ldots u$
89	$1.\mathbf{999999999997}033180\ldots u$
93	$1.\mathbf{9999999999992}46073\ldots u$
97	$1.\mathbf{99999999999983}03397\ldots u$
101	$1.\mathbf{999999999999949}423\ldots u$
105	$1.\mathbf{9999999999999}86669\ldots u$
109	$1.\mathbf{99999999999999}6677\ldots u$
113	$1.\mathbf{999999999999999}175\ldots u$

Using the fact that $1+\frac{x}{2}-\frac{x^2}{8} < \sqrt{1+x} < 1+\frac{x}{2}$ for all $x>0$, we have in particular

$$1+2^{\frac{-p+1}{2}}-2^{-p} < \sqrt{1+2^{\frac{-p+3}{2}}} < 1+2^{\frac{-p+1}{2}},$$

so rounding *up* occurs when computing ρ:

$$\rho = 1+2^{\frac{-p+1}{2}}. \tag{15}$$

Finally, let us estimate the relative error of ρ. We have $b^2 = 1+2\eta+\eta^2$, and from (13), we deduce that

$$H\cdot(1-2^{-p+1}) < a^2 < H\cdot(1+2^{-p+1}+2^{-2p}),$$

which implies that a^2 is between

$$2^{\frac{-p+3}{2}}-2\eta-3\cdot 2^{-p}+2^{\frac{-3p+3}{2}}-2^{\frac{-3p+5}{2}} +2^{-p+2}\eta+3\cdot 2^{-2p+1}-2^{\frac{-5p+5}{2}}$$

and

$$2^{\frac{-p+3}{2}}-2\eta-3\cdot 2^{-p}+2^{\frac{-3p+3}{2}}+2^{\frac{-3p+5}{2}} -2^{-p+2}\eta-3\cdot 2^{-2p+1}+2^{\frac{-5p+5}{2}}+\mathcal{O}\big(2^{\frac{-5p}{2}}\big).$$

Therefore, using this last result and (9), we have $a^2+b^2 = 1+2^{\frac{-p+3}{2}}-2^{-p+1}+\mathcal{O}\big(2^{\frac{-3p}{2}}\big)$, from which we easily deduce that

$$\sqrt{a^2+b^2} = 1+2^{\frac{-p+1}{2}}-2^{-p+1}+\mathcal{O}\big(2^{\frac{-3p}{2}}\big). \tag{16}$$

For $u=2^{-p}$, we deduce from (15) and (16) that the relative error $|\rho-\sqrt{a^2+b^2}|/\sqrt{a^2+b^2}$ has the form $2u+\mathcal{O}(u^{3/2})$ as $u\to 0$. This concludes the proof of Theorem II.1 when p is odd. □

Table II gives the relative errors obtained with the values a and b of our generic example for various *odd* values of p between 53 and 113. Again, the a priori bound $2u$ is tight already for reasonably small precisions.

III. THE CASE OF $c/\sqrt{a^2+b^2}$

A. Error bound

We consider now the following computation of $c/\sqrt{a^2+b^2}$: first, we evaluate $\sqrt{a^2+b^2}$ using either Algorithm 1 (without FMA) or Algorithm 2 (with FMA), and then we perform the floating-point division of c by the obtained value. This corresponds to the two algorithms below.

Algorithm 3 Computation of $c/\sqrt{a^2+b^2}$ in binary precision-p floating-point arithmetic.

$s_a \leftarrow \mathrm{RN}(a^2)$
$s_b \leftarrow \mathrm{RN}(b^2)$
$s \leftarrow \mathrm{RN}(s_a+s_b)$
$\rho \leftarrow \mathrm{RN}(\sqrt{s})$
$g \leftarrow \mathrm{RN}(c/\rho)$
return g

Algorithm 4 FMA-based computation of $c/\sqrt{a^2+b^2}$ in binary precision-p floating-point arithmetic.

$s_b \leftarrow \mathrm{RN}(b^2)$
$s \leftarrow \mathrm{RN}(a^2+s_b)$
$\rho \leftarrow \mathrm{RN}(\sqrt{s})$
$g \leftarrow \mathrm{RN}(c/\rho)$
return g

A rounding error analysis valid for both algorithms is given by the following theorem.

Theorem III.1. *If $p\neq 3$, then the relative error committed when approximating $c/\sqrt{a^2+b^2}$ by the result g of Algorithm 3 or 4 is less than $3u$.*

Proof: If $p\neq 3$, then [6, Lemma 2] says that $s_a = a^2(1+\epsilon_1)$ and $s_b = b^2(1+\epsilon_2)$ with $|\epsilon_1|,|\epsilon_2| \leqslant \frac{u}{1+3u} =: u_3$. Furthermore, there exist ϵ_3 and ϵ_4 such that $|\epsilon_3|,|\epsilon_4| \leqslant \frac{u}{1+u} =: u_1$ and

$$s = \begin{cases} (s_a+s_b)(1+\epsilon_3) & \text{for Algorithm 3,} \\ (a^2+s_b)(1+\epsilon_4) & \text{for Algorithm 4.} \end{cases}$$

Hence, in both cases, s can be bounded as follows:

$$(a^2+b^2)(1-u_1)(1-u_3) \leqslant s \leqslant (a^2+b^2)(1+u_1)(1+u_3).$$

Recalling from [1] that the relative error of division in radix-2 FP arithmetic is at most $u-2u^2$, we have

$$g = \frac{c}{\sqrt{s}\,(1+\epsilon_5)}(1+\epsilon_6)$$

with $|\epsilon_5| \leqslant u_1$ and $|\epsilon_6| \leqslant u-2u^2$, and then

$$\frac{c}{\sqrt{s}}\cdot\frac{1-u+2u^2}{1+u_1} \leqslant g \leqslant \frac{c}{\sqrt{s}}\cdot\frac{1+u-2u^2}{1-u_1}.$$

Consequently, $\zeta\frac{c}{\sqrt{a^2+b^2}} \leqslant g \leqslant \zeta'\frac{c}{\sqrt{a^2+b^2}}$ with

$$\zeta := \frac{1}{\sqrt{(1+u_1)(1+u_3)}}\cdot\frac{1-u+2u^2}{1+u_1}$$

and

$$\zeta' := \frac{1}{\sqrt{(1-u_1)(1-u_3)}} \cdot \frac{1+u-2u^2}{1-u_1}.$$

To conclude, it suffices to check that $1-3u < \zeta$ and $\zeta' < 1+3u$ for all $u \leqslant 1/2$. ∎

B. Asymptotic optimality of the bound

We are going to show that the previously obtained bound is asymptotically optimal. More precisely, we establish the following result.

Theorem III.2. *For $p \geqslant 12$, there exist floating-point inputs a, b, and c for which the result g returned by Algorithm 3 or Algorithm 4 satisfies*

$$\left| \frac{g - \frac{c}{\sqrt{a^2+b^2}}}{\frac{c}{\sqrt{a^2+b^2}}} \right| = 3u - \epsilon, \quad |\epsilon| = \mathcal{O}(u^{3/2}).$$

The values of a and b we are going to choose to prove Theorem III.2 are the same as the ones we have chosen for $\sqrt{a^2+b^2}$ in Section II-B1 (if p is even) and in Section II-B2 (if p is odd). For these values, we have shown that Algorithms 1 and 2 return the same result. As a consequence, the variable ρ will have the same value in Algorithms 3 and 4. Therefore, for these chosen input values a and b, Algorithms 3 and 4 will return the same final value g. Hence, in the following, we do not need to analyze these algorithms separately.

1) If p is even: Let us show that when the precision p is even, the error bound given by Theorem III.1 is asymptotically optimal. To do this, we exhibit generic inputs a, b, c parameterized by p. For a and b, we use the same values as in Section II-B1, namely, $b = 1+2^{-p/2}$ and $a = \mathrm{RD}\big(2^{-\frac{3p}{4}}\sqrt{G}\big)$, where $G = \big\lceil 2^{\frac{p}{2}}\left(\sqrt{2}-1\right)+\delta \big\rceil \cdot 2^{\frac{p}{2}+1} + 2^{\frac{p}{2}}$ and

$$\delta = \begin{cases} 1 & \text{if } \left\lceil 2^{\frac{p}{2}}\sqrt{2} \right\rceil \text{ is odd,} \\ 2 & \text{otherwise.} \end{cases}$$

Recall from (3), (6), and (7) in Section II-B1 that

$$\rho = 1 + \frac{\omega}{2} - 2^{-p+1}$$

and

$$\sqrt{a^2+b^2} = 1 + \frac{\omega}{2} + \mathcal{O}\big(2^{-3p/2}\big), \tag{17}$$

where

$$\omega = 2^{-p+1}\left\lceil 2^{p/2}\sqrt{2}+\delta \right\rceil.$$

Now, let us define

$$c = 1 + 2^{-p+1} \cdot \left\lfloor 3\sqrt{2}\cdot 2^{p/2-2} \right\rfloor.$$

One can write $c = 1 + 2^{-p+1}(3\sqrt{2}\cdot 2^{p/2-2} + \alpha_1)$ with $-1 < \alpha_1 < 0$. Similarly, one can write

$$\omega = 2^{-p+1}\left(2^{p/2}\sqrt{2}+\delta+\alpha_2\right) \tag{18}$$

with $0 < \alpha_2 < 1$. Notice that the choice of δ implies that $\lceil 2^{p/2}\sqrt{2}+\delta \rceil = 2^{p/2}\sqrt{2}+\delta+\alpha_2$ is an even integer.

Defining

$$t = 2^{-p/2},$$

we obtain $c = 1 + \frac{3}{2}\sqrt{2}\,t + 2\alpha_1 t^2$ and $\rho = 1 + \sqrt{2}\,t + (\delta + \alpha_2 - 2)t^2$, so the exact quotient c/ρ has the form

$$\frac{c}{\rho} = 1 + \frac{\sqrt{2}}{2}t + (2\alpha_1 - \delta - \alpha_2 + 1)\,t^2 + \epsilon(t),$$

where $\epsilon(t) = \mathcal{O}(t^3)$ as $t \to 0$. Furthermore,

- we know that $3\sqrt{2}\cdot 2^{p/2-2} + \alpha_1$ is an integer, which implies that
$$A(t) = \frac{3}{2}\sqrt{2}\,t + 2\alpha_1 t^2$$
is a multiple of $2t^2 = 2^{-p+1}$;
- we know that $2^{p/2}\sqrt{2} + \delta + \alpha_2$ is an even integer, which implies that
$$B(t) = \sqrt{2}\,t + (\delta+\alpha_2)t^2$$
is a multiple of $2t^2$.

Consequently, we can rewrite c/ρ as

$$\begin{aligned} \frac{c}{\rho} &= 1 + A(t) - B(t) + t^2 + \epsilon(t) \\ &= \big(\text{multiple of } 2^{1-p}\big) + 2^{-p} + \epsilon(t). \end{aligned}$$

Now, it is easily checked that the ranges of δ, α_1, and α_2 imply that $-5 < 2\alpha_1 - \delta - \alpha_2 < -1$ and, consequently, that $0 \leqslant A(t) - B(t) < 1$ as soon as $p \geqslant 6$. This means that the integer $(1 + A(t) - B(t))/2^{1-p}$ is in $\{2^{p-1}, \ldots, 2^p - 1\}$.

On the other hand, one can check that $0 < \epsilon(t) \leqslant 2^{-p}$ as soon as $p \geqslant 6$. Indeed, the definitions of c, ρ, and ϵ imply that

$$\frac{\epsilon(t)}{t^2} = \frac{P_1 t + P_2 t^2}{1 + \sqrt{2}t + Q_2 t^2}$$

with

- $P_1 = \frac{\sqrt{2}}{2}\delta - 2\sqrt{2}\alpha_1 + \frac{\sqrt{2}}{2}\alpha_2$,
- $P_2 = Q_2(Q_2 + 1 - 2\alpha_1)$,
- $Q_2 = \delta + \alpha_2 - 2$.

Using again the ranges of δ, α_1, and α_2, one can check that $\frac{\sqrt{2}}{2} < P_1 < \frac{7}{2}\sqrt{2}$, $-1 < Q_2 < 1$, and $-3 < P_2 < 4$, and then, for $p \geqslant 6$ (that is, $t \leqslant 1/8$), that $1 + \sqrt{2}t + Q_2 t^2$ is larger than 1 and that $P_1 t + P_2 t^2$ is positive and less than 1.

Hence rounding *up* occurs when computing $g = \mathrm{RN}(c/\rho)$:

$$\begin{aligned} g &= 1 + A(t) - B(t) + 2^{-p+1} \\ &= 1 + \frac{\sqrt{2}}{2}t + (2\alpha_1 - \delta - \alpha_2 + 2)t^2. \end{aligned} \tag{19}$$

To estimate the relative error of g, it remains to estimate the exact value $c/\sqrt{a^2+b^2}$. First, from (17) and (18) we deduce that

$$\sqrt{a^2+b^2} = 1 + \sqrt{2}t + (\delta+\alpha_2)t^2 + \mathcal{O}(t^3).$$

Since $c = 1 + 2^{-p+1}(3\sqrt{2}\cdot 2^{p/2-2} + \alpha_1)$, this implies

$$\frac{c}{\sqrt{a^2+b^2}} = 1 + \frac{\sqrt{2}}{2}t + (2\alpha_1 - \delta - \alpha_2 - 1)t^2 + \mathcal{O}(t^3). \quad (20)$$

By combining (19) and (20), we find that the relative error committed when applying Algorithm 3 or Algorithm 4 to the above inputs a, b, and c has the form $3t^2+\mathcal{O}(t^3)$, or, equivalently, $3u+\mathcal{O}(u^{3/2})$. This concludes the proof of Theorem III.2 in the case where p is even. □

2) If p is odd: Let us show that when the precision p is odd, the error bound given by Theorem III.1 is asymptotically optimal. To do this, we exhibit a generic example parameterized by p. We will use the same values a and b as in Section II-B2, as defined in (8), (11), and (12). We remind the reader that for these particular values we proved in (15) that

$$\rho = 1 + 2^{\frac{-p+1}{2}}$$

and, in (16), that

$$\sqrt{a^2+b^2} = 1 + 2^{\frac{-p+1}{2}} - 2^{-p+1} + \mathcal{O}\big(2^{\frac{-3p}{2}}\big).$$

Let us now choose $c = 1 + 3\cdot 2^{\frac{-p-1}{2}} + 2^{-p+1}$. Setting

$$t = 2^{\frac{-p+1}{2}}$$

gives $c = 1 + \frac{3}{2}t + t^2$ and $\rho = 1 + t$, and, since $t > 0$, it is then easily checked that

$$1 + \frac{t}{2} < \frac{c}{\rho} < 1 + \frac{t}{2} + \frac{t^2}{2}.$$

Since $1 + t/2 = 1 + 2^{\frac{-p-1}{2}}$ is an FP number in $[1, 2)$, and since $t^2/2 = 2^{-p}$, we deduce that rounding *down* occurs when evaluating $g = \mathrm{RN}(c/\rho)$:

$$g = 1 + \frac{t}{2}. \quad (21)$$

Now, the definition of t also implies that $\sqrt{a^2+b^2} = 1 + t - t^2 + \mathcal{O}(t^3)$ and, consequently,

$$\frac{c}{\sqrt{a^2+b^2}} = 1 + \frac{t}{2} + \frac{3}{2}t^2 + \mathcal{O}(t^3). \quad (22)$$

From (21) and (22) we deduce that the relative error committed when approximating $c/\sqrt{a^2+b^2}$ by the result g of Algorithm 3 or Algorithm 4 with a, b, and c as above has the form $\frac{3}{2}t^2 + \mathcal{O}(t^3)$. Since $t = 2^{\frac{-p+1}{2}} = \sqrt{2u}$, this is $3u+\mathcal{O}(u^{3/2})$, which concludes the proof of Theorem III.1 when p is odd. □

Table III illustrates the tightness of the a priori bound $3u$ by giving (for various common precisions p) the relative errors obtained with the values a, b, c introduced in Sections III-B1 and III-B2.

TABLE III
RELATIVE ERRORS OBTAINED, FOR VARIOUS PRECISIONS p, WHEN RUNNING ALGORITHM 3 OR ALGORITHM 4 WITH OUR GENERIC VALUES a, b, AND c OF SECTION III-B1 (FOR p EVEN) AND SECTION III-B2 (FOR p ODD).

p	relative error
24	$2.\mathbf{99}800258913676259676349 8\ldots u$
53	$2.\mathbf{99999989646575835154216 9}\ldots u$
64	$2.\mathbf{99999997}359196820010396\ldots u$
113	$2.\mathbf{99999999999999}896692295\ldots u$
128	$2.\mathbf{99999999999999999}566038\ldots u$

CONCLUSION

After having reminded the relative error bound for $\sqrt{a^2+b^2}$, slightly improved the bound for $c/\sqrt{a^2+b^2}$ (by showing that the quadratic term $\mathcal{O}(u^2)$ can be removed), and considered variants that take advantage of the possible availability of an FMA instruction, we have shown that these bounds are asymptotically optimal. Hence, trying to significantly refine them further is hopeless. In the paper we assumed an unbounded exponent range, so that our results hold provided that no underflow or overflow occurs. Several authors have suggested ways to avoid spurious overflows and underflows in the evaluation of $\sqrt{a^2+b^2}$, either using an exception handler and/or scaling the input values [7]–[11]. Of course, if the scaling introduces further rounding errors, then our error bounds may not hold anymore. However, if a and b are scaled by a power of 2 (as advocated in [8], [10], [11]), then our analyses still apply.

ACKNOWLEDGMENT

The work of C.-P. Jeannerod and J.-M. Muller was supported in part by the French National Research Agency under grant ANR-13-INSE-0007 (MetaLibm project).

REFERENCES

[1] C.-P. Jeannerod and S. M. Rump, "On relative errors of floating-point operations: optimal bounds and applications," *Mathematics of Computation*, 2016, to appear. [Online]. Available: https://hal.inria.fr/hal-00934443

[2] *IEEE Standard for Floating-Point Arithmetic, IEEE Standard 754-2008*. New York: IEEE Computer Society, 2008.

[3] D. E. Knuth, *The Art of Computer Programming, Volume 2, Seminumerical Algorithms*, 3rd ed. Reading, MA, USA: Addison-Wesley, 1998.

[4] P. H. Sterbenz, *Floating-Point Computation*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1974.

[5] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*. Oxford University Press, London, 1965.

[6] C.-P. Jeannerod, N. Louvet, J.-M. Muller, and A. Plet, "Sharp error bounds for complex floating-point inversion," *Numerical Algorithms*, vol. 73, no. 3, pp. 735–760, 2016.

[7] W. Kahan, "Branch cuts for complex elementary functions or much ado about nothing's sign bit," in *The State of the Art in Numerical Analysis*, A. Iserles and M. J. D. Powell, Eds. Oxford University Press, 1987, pp. 165–211.

[8] T. E. Hull, T. F. Fairgrieve, and P. T. P. Tang, "Implementing complex elementary functions using exception handling," *ACM Transactions on Mathematical Software*, vol. 20, no. 2, pp. 215–244, 1994.

[9] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, 3rd ed. Cambridge University Press, 1997.
[10] A. Ziv, "Sharp ULP rounding error bound for the hypotenuse function," *Mathematics of Computation*, vol. 68, no. 227, pp. 1143–1148, 1999.
[11] D. Bindel, J. Demmel, W. Kahan, and O. Marques, "On computing Givens rotations reliably and efficiently," *ACM Transactions on Mathematical Software*, vol. 28, no. 2, pp. 206–238, 2002.