Stupid is as Stupid Does: Taking the Square Root of the Square of a Floating-Point Number

Sylvie Boldo

Stupid is as Stupid Does: Taking the Square Root of the Square of a Floating-Point Number

Sylvie Boldo[1,2]

Inria,
LRI, bâtiment 650,
Université Paris-Sud,
F-91405 Orsay Cedex, France

Abstract

Floating-point experts know that mathematical formulas may fail or give imprecise results when implemented in floating-point arithmetic. This article describes an example where, surprisingly, it is absolutely not the case. Indeed, using radix 2 and an unbounded exponent range, the computation of the square root of the square of a floating-point number a is exactly $|a|$. A consequence is the fact that the floating-point computation of $a/\sqrt{(a^2+b^2)}$ is always in the interval $[-1,1]$. This removes the need for a test when calling an arccos or an arcsin on this value. For more guarantees, this property was formally checked using the Coq proof assistant and the Flocq library. The conclusion will give hints on what happens without assumptions and in other radices, where the behavior is very different.

Keywords: Floating-point, formal proof, square, square root, radix, Coq.

1 Introduction

Floating-point (FP) arithmetic is seen as intricate because too few people have sufficient knowledge to understand how it works. For people having been only trained with mathematics, facts such that $(x+y)+z$ may be different from $x+(y+z)$ for certain values or the fact that there exists x such that $x \neq 0$, but $x^2 = 0$ is beyond comprehension. This is the reason why mathematical formulas are most of the time programmed as they stand in mathematical textbooks even if this may not be a good idea [1]. The main shortcoming is the limited precision: as the exact mathematical result cannot be exactly represented, it has to be rounded to the nearest floating-point values (other roundings are available [6] but are seldom used, except in interval arithmetic).

[1] This work was supported by the VERASCO (ANR-11-INSE-003) and the FastRelax (ANR-14-CE25-0018-01) projects of the French National Agency for Research (ANR). The author is also indebted to Jean-Michel Muller for having described this problem to her a long long time ago.

[2] Email: sylvie.boldo@inria.fr

An example where it might happen is the following one. Given a right triangle with sides a, b and the hypotenuse h, we want to compute one of the angle θ, that is such that $\sin(\theta) = a/h = a/\sqrt{(a^2+b^2)}$. Therefore, to compute θ, one has to take the arcsin of

$$\frac{a}{\sqrt{(a^2+b^2)}}.$$

This may be a problem: the arcsin function assumes an input between -1 and 1, but this FP computation may be incorrect. How can we be sure that the arcsin computation will not fail? A simple solution is to compute in FP arithmetic

$$x = \circ\left(\frac{a}{\circ\left(\sqrt{\circ\left(\circ\left(a^2\right)+\circ\left(b^2\right)\right)}\right)}\right),$$

where $\circ$ is a FP rounding. If $x > 1$, we return 1. If $x < -1$, we return -1. Then, we are sure that the result will be between -1 and 1 and that the call to arcsin will not fail. Another more complicated solution (if possible) is to prove that x will always be between -1 and 1, hence getting rid of the tests. This is a substantial performance benefit: when the test is not correctly guessed, the pipeline has to be flushed, which is very costly.

This is the goal of this article. It is easy to see that the worst case corresponds to $b = 0$ where x reduces to $\circ\left(\frac{a}{\circ\left(\sqrt{\circ(a^2)}\right)}\right)$. It is sufficient (but not necessary!) to prove that $\circ\left(\sqrt{\circ\left(a^2\right)}\right) = |a|$ as it implies that x will be either 1 or -1. This is the reason we will study what happens when taking the square root of the square of a floating-point number.

This is not a new problem: Cody and Waite use this property as a test for the square root function [4] (pages 12 and 28). It is also stated in one of Kahan's web papers [5] (page 29).There is no detailed proof in any of there references. In particular, the minimal precision is never explicit.

More than a pen-and-paper proof, this work gives a high guarantee of its correctness and gives a precise hypothesis on the needed precision. We will rely on the Coq proof assistant. From the formal methods point of view, we will base our proof on the Flocq library [3]. Flocq is a formalization in Coq that offers a multi-radix and multi-precision formalization for various floating- and fixed-point formats (including FP with or without gradual underflow) with a comprehensive library of theorems. Its usability and practicality have been established against test-cases [2].

We will here assume we are using rounding to nearest, ties to even in radix 2, denoted by $\circ$, with a precision $p > 1$. In particular, the square root is assumed to be computed correctly. Note that $p > 1$ will be the only requirement on the precision in the whole paper. We will also assume that there is neither underflow nor overflow: the exponent range is unbounded. This corresponds to the FLX format of the Flocq library [3]. What happens using a bounded exponent range is described in the conclusion. The proof is available in the example sub-directory of the Flocq library from version 2.3.0 available at

http://flocq.gforge.inria.fr/.

This article is organized as follows: Section 2 describes some intermediate rather generic lemmas. Section 3 describes the main results. Section 4 concludes and gives hints on what happens in practice and with radices other than 2.

2 Intermediate Lemmas

The ulp (unit in the place) denotes the value of the last bit of the mantissa of a FP number or the corresponding value for a real number [6,3].

Lemma 1 (round_le_half_an_ulp) *Let v be a real and u be a positive FP number. Assume that $v < u + \mathrm{ulp}(u)/2$, then $\circ(v) \leq u$.*

Proof. It is already proved in Flocq that the generic rounding to nearest (with any tie-breaking rule) is monotone, that is to say, if $x < y$, then $\circ_1(x) \leq \circ_2(y)$ whatever the two tie-breaking-rules. As $v < u + \mathrm{ulp}(u)/2$ and if we consider the even tie-breaking rule for v and the tie-breaking rule down (or towards zero) for $u + \mathrm{ulp}(u)/2$, we get $\circ(v) \leq u$, as $u + \mathrm{ulp}(u)/2$ is the midpoint between u and its successor. □

Lemma 2 (round_ge_half_an_ulp) *Let v be a real and u be a positive FP number. First assume that u is not a power of 2. Assume also that $u - \mathrm{ulp}(u)/2 < v$, then $u \leq \circ(v)$.*

This inequality is more complicated than the previous one, with a proof about four times bigger.

Proof. A key point is the fact that u is not a power of the radix. Then, its predecessor has the same exponent, and the same ulp. This fact takes about half of the proof. It relies on the fact that the FP format is floating-point with an unbounded exponent range.

The rest of the proof is similar to the previous one, relying on the monotony of rounding to nearest, whatever the ties. An additional difficulty is the need to prove that the predecessor of u is positive, which is also a consequence of the unbounded exponent range. □

The last lemma of this section may seem strange, but it is easier to prove it before the main theorem. Moreover, this is the only place where we rely on the fact that the radix is 2.

Lemma 3 (ulp_sqr_ulp_lt) *Let u be a positive real number. Then*

$$\mathrm{ulp}\left(u^2\right) + \frac{(\mathrm{ulp}(u))^2}{2} < 2\,u\,\mathrm{ulp}(u).$$

Proof. There is no subtlety here. Let e be such that $\mathrm{ulp}(u) = 2^{e-p}$ and i such that $\mathrm{ulp}\left(u^2\right) = 2^{i-p}$. then we either have $i = 2\,e - 1$ or $i = 2\,e$. We then split into the two cases, handle substitutions and inequalities and prove the results holds in both cases. □

3 Main Results

With the previous lemmas, we can now prove the required equality on positive FP numbers.

Lemma 4 (round_flx_sqr_sqrt_exact_aux) *Let u be a positive FP number. Then*

$$\circ\left(\sqrt{\circ\left(u^2\right)}\right) = u.$$

Proof. The first case is when u is a power of the radix. This is easy: as the exponent range is unbounded, u^2 is computed exactly and then the square root is also computed exactly and there is no rounding error at all. Let us now assume that u is not a power of the radix. Let us denote $y = \circ\left(u^2\right)$.

To prove that u is smaller or equal to the computation, we use the previous Lemma 2. We are left to prove that $u - \mathrm{ulp}(u)/2 < \sqrt{y}$. We replace it equivalently with $u \times (1 - \mathrm{ulp}(u)/2/u) < \sqrt{y}$. As $|y - u^2| \leq \mathrm{ulp}\left(u^2\right)/2$, we simplify it to only prove $u \times (1 - \mathrm{ulp}(u)/2/u) < \sqrt{u^2(1 - \mathrm{ulp}\left(u^2\right)/2/u^2)}$. Squaring the inequality and simplifying by u^2, it amounts to $(1 - \mathrm{ulp}(u)/2/u)^2 < 1 - \mathrm{ulp}\left(u^2\right)/2/u^2$, that can be simplified (after some work) exactly into the formula of Lemma 3.

To prove the other inequality, the proof is similar, but easier than the preceding one. We apply Lemma 1. We have left to prove that $\sqrt{y} < u + \mathrm{ulp}(u)/2$. Taking advantage of the $\mathrm{ulp}\left(u^2\right)/2$ distance between y and u^2, using similar formulas such as $u \times (1 + \mathrm{ulp}(u)/2/u)$ and Lemma 3, we prove this inequality. Note that the use of Lemma 3 could have probably been prevented.

Of course, many positiveness and non-zero lemmas were required and formally proved, but it is useless to detail them here as they were formally verified. □

Theorem 5 (round_flx_sqr_sqrt_exact) *Let u be a FP number. Then*

$$\circ\left(\sqrt{\circ\left(u^2\right)}\right) = |u|.$$

Proof. This is very straightforward as it holds when u is either positive (Theorem 4), negative (Theorem 4 on $-u$),or zero (as $\circ(0) = 0$). □

Theorem 6 (sqrt_sqr) *Let a be a FP number and b be a real. Then*

$$-1 \leq \circ\left(\frac{a}{\circ\left(\sqrt{\circ\left(\circ\left(a^2\right) + \circ\left(b^2\right)\right)}\right)}\right) \leq 1.$$

Note that b can be any real number: it is not required to be a FP number. This means less statements to prove when applying this theorem.

Proof. The proof is quite simple from the previous lemmas:

$$\begin{aligned}\left|\circ\left(\frac{a}{\circ\left(\sqrt{\circ\left(\circ\left(a^2\right)+\circ\left(b^2\right)\right)}\right)}\right)\right| &\leq \left|\circ\left(\frac{a}{\circ\left(\sqrt{\circ\left(\circ\left(a^2\right)\right)}\right)}\right)\right| \\ &= \left|\circ\left(\frac{a}{\circ\left(\sqrt{\circ\left(a^2\right)}\right)}\right)\right| \\ &= \left|\circ\left(\frac{a}{|a|}\right)\right| \\ &= 1\end{aligned}$$

□

4 Conclusion & Openings to Other Radices

4.1 Conclusion and limits

We have proved that, using radix 2 and a precision greater than 1, the rounded computation of $\sqrt{a^2}$ is indeed $|a|$. This leads to the fact that the computation of $\frac{a}{\sqrt{(a^2+b^2)}}$ will always be between -1 and 1, so that further computations will not fail.

A problem ignored here is that of limited range for the exponent. Contrarily to $\left(\sqrt{a}\right)^2$, the computation of $\sqrt{a^2}$ may underflow and overflow. For very small positive a, the result will be 0 and for big finite a, it will be $+\infty$. This means this can be detected afterwards. Moreover, the preceding result is valid only for medium-range a, that is to say $|a|$ is about between 2^{-511} and 2^{511} in the `binary64` format. Scaling may be necessary.

Another question is if this still holds when using rounding to nearest, ties away from zero [6]. The answer is yes as no mid-point behavior is used. The bigger problem is that, if we want to express this property using any tie on any rounding, we must quantify on the 5 roundings that appear in the last theorem, which is cumbersome. If directed roundings are used, the results do not hold anymore.

4.2 Other radices

It seems that $x = \circ\left(\sqrt{\circ\left(x^2\right)}\right)$ still holds in any radix when the mantissa of x is small. But we have examples where $x > \circ\left(\sqrt{\circ\left(x^2\right)}\right)$, even by several ulps:

- With a radix $\beta = 10$ and a precision $p = 4$, if $x = 31.66$, then we have $\circ\left(\sqrt{\circ\left(x^2\right)}\right) = 31.65$.
- With $\beta = 1000$ and $p = 2$, if $x = 31.662$, then $\circ\left(\sqrt{\circ\left(x^2\right)}\right) = 31.654$.

All these examples are just above a multiple of $\sqrt{\beta}$, hinting where the worst case may lie. The number of ulps between x and $\circ\left(\sqrt{\circ\left(x^2\right)}\right)$ also seems to grow when the radix is increased.

A more interesting fact is that, even if $x \neq \circ\left(\sqrt{\circ\left(x^2\right)}\right)$ in many cases, the division returns 1 nonetheless on those cases. We were not able to find values such that the computation of $\circ\left(\frac{a}{\circ\left(\sqrt{\circ(a^2)}\right)}\right)$ is above 1. Some initial work seem to show that this holds for radices, the difficult cases being radix 3 and 5. Formal developments in Coq are in progress, with a focus on the smallest minimal precision needed.

References

[1] Sylvie Boldo. How to compute the area of a triangle: a formal revisit. In Alberto Nannarelli and Peter-Michael Seideland Ping Tak Peter Tang, editors, *Proceedings of the 21th IEEE Symposium on Computer Arithmetic*, pages 91–98, Austin, Texas, USA, April 2013.

[2] Sylvie Boldo. *Deductive Formal Verification: How To Make Your Floating-Point Programs Behave*. Thèse d'habilitation, Université Paris-Sud, October 2014.

[3] Sylvie Boldo and Guillaume Melquiond. Flocq: A unified library for proving floating-point algorithms in Coq. In Elisardo Antelo, David Hough, and Paolo Ienne, editors, *20th IEEE Symposium on Computer Arithmetic*, pages 243–252, Tübingen, Germany, 2011.

[4] W. Cody and W. Waite. *Software Manual for the Elementary Functions*. Prentice-Hall, Englewood Cliffs, NJ, 1980.

[5] William Kahan. Mathematics written in sand — the HP-15C, Intel 8087, etc. http://www.cs.berkeley.edu/~wkahan/MathSand.pdf, 1983.

[6] Microprocessor Standards Committee. IEEE Standard for Floating-Point Arithmetic. *IEEE Std. 754-2008*, pages 1–58, August 2008.