On the maximum relative error when computing x^n in floating-point arithmetic

Stef Graillat, Vincent Lefèvre, Jean-Michel Muller

On the maximum relative error when computing x^n in floating-point arithmetic

Stef Graillat
Université Pierre et Marie Curie Paris 6
Laboratoire LIP6

Vincent Lefèvre
Inria, Laboratoire LIP
Université de Lyon

Jean-Michel Muller
CNRS, Laboratoire LIP
Université de Lyon

February 11, 2014

Abstract

In this paper, we improve the usual relative error bound for the computation of x^n through iterated multiplications by x in binary floating-point arithmetic. The obtained error bound is only slightly better than the usual one, but it is simpler. We also discuss the more general problem of computing the product of n terms.

Keywords: floating-point arithmetic, rounding error, accurate error bound, exponentiation

AMS Subject Classifications: 15-04, 65G99, 65-04

1 Introduction

1.1 Floating-point arithmetic and rounding errors

In general, computations in floating-point arithmetic are not errorless: a small rounding error occurs each time an arithmetic operation is performed. Depending on the calculation being done, the global influence of these individual rounding errors can rank anywhere between completely negligible and overwhelming. Hence, it is always important to have some information on the numerical quality of a computed result. Furthermore, when critical applications are at stake, one may need *certain* yet *tight* error bounds. The manipulation of these error bounds (either paper-and-pencil manipulation or—if one wishes to do some dynamical error analysis—numerical manipulation) will also be made easier if these bound are *simple*.

In the following, we assume a radix-2, precision-p, floating-point (FP) arithmetic. To simplify the presentation, we assume an unbounded exponent range: our results will be

applicable to "real life" floating-point systems, such as those that are compliant with the IEEE 754-2008 Standard for Floating-Point Arithmetic [3, 6], provided that no underflow or overflow occurs. In such an arithmetic, a floating-point number is either zero or a number of the form

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

where X and e_x are integers, with $2^{p-1} \leq |X| \leq 2^p - 1$. The number X is called the *integral significand* of x, $X \cdot 2^{-p+1}$ is called the *significand* of x, and e_x is called the *exponent* of x.

As said above, since in general the sum, product, quotient, etc., of two FP numbers is not a FP number, it must be *rounded*. The IEEE 754-2008 Standard requires that the arithmetic operations should be *correctly rounded*: a rounding function must be chosen among five possible functions defined by the standard. If $\circ$ is the rounding function, when the arithmetic operation $(a \top b)$ is performed, the value that must be returned is the FP number $\circ(a \top b)$. The default rounding function is *round to nearest ties to even*, denoted RN_{even}, defined as follows:

(*i*) for all FP numbers y, $|\mathrm{RN}_{even}(t) - t| \leq |y - t|$;

(*ii*) if there are two FP numbers that satisfy (*i*), $\mathrm{RN}_{even}(t)$ is the one whose integral significand is even.

The IEEE 754-2008 standard defines another round-to-nearest rounding function, namely *round to nearest ties to away*, where (*ii*) is replaced by

(*ii'*) if there are two FP numbers that satisfy (*i*), $\mathrm{RN}_{away}(t)$ is the one whose integral significand has the largest magnitude.

In the following, RN is one of these two round-to-nearest functions. More precisely: unless stated otherwise, the bounds we give are applicable to both rounding functions. However, when we build examples (for instance for checking how tight are the obtained bounds), we use RN_{even}.

Recently, classic error bounds for summation and dot product have been improved by Jeannerod and Rump [8, 5]. They have considered the problem of calculating the sum of n FP numbers $x_1, x_2, \ldots, x_n$. If we call $\mathrm{float}(\sum_{i=1}^n x_i)$ the computed result and $u = 2^{-p}$ the *rounding unit*, they have shown that

$$\left| \mathrm{float}\left(\sum_{i=1}^n x_i \right) - \sum_{i=1}^n x_i \right| \leq (n-1) \cdot u \sum_{i=1}^n |x_i| \tag{1}$$

which is better than the previous bound [2, p.63]

$$\left| \mathrm{float}\left(\sum_{i=1}^n x_i \right) - \sum_{i=1}^n x_i \right| \leq \gamma_{n-1} \sum_{i=1}^n |x_i|$$

where

$$\gamma_n = \frac{n \cdot u}{1 - n \cdot u} = n \cdot u + n^2 \cdot u^2 + n^3 \cdot u^3 + \cdots = n \cdot u + \mathcal{O}(u^2). \tag{2}$$

We are interested in finding if a similar simplification is possible in the particular case of the computation of an integer power x^n, that is we wish to know if the result

computed using the "naive algorithm" (Algorithm 1 below) is always within relative error $(n-1)\cdot u$ from the exact result. This is "experimentally true" in binary32/single precision arithmetic. More precisely, we did an exhaustive check for all $x \in [1; 2[$ in binary32 (2^{23} numbers to be checked) until overflow for x^n. For the smallest number larger than 1, namely $x = 1 + 2u$, $n \approx 7.5 \times 10^8$ is needed to reach overflow. Our test used a 100-bit interval arithmetic provided by the MPFI [7] package.

In this paper, we prove—under mild hypotheses—that this result holds for all "reasonable" floating-point formats (we need the precision p to be larger than or equal to 5, which is always true in practice).

1.2 Relative error due to roundings

Let t be a positive real number between 2^e and 2^{e+1}, where $e \in \mathbb{Z}$. The rounding $\mathrm{RN}(t)$ is between 2^e and 2^{e+1} too, and we have

$$|\,\mathrm{RN}(t) - t| \leq 2^{e-p}. \tag{3}$$

From this, we easily deduce a bound on the relative error due to rounding t

$$\left|\frac{\mathrm{RN}(t) - t}{t}\right| \leq 2^{-p} = u. \tag{4}$$

This is illustrated by Figure 1.

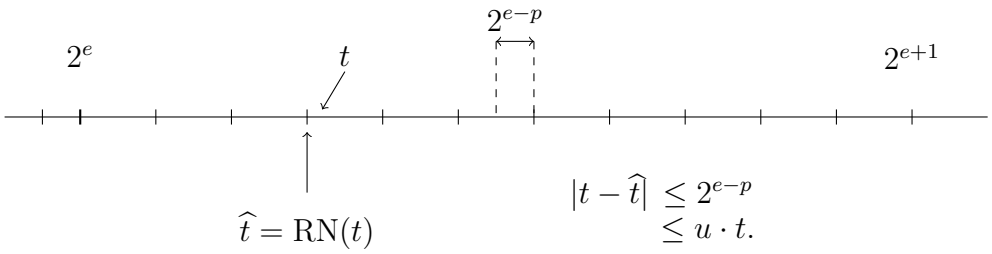

Figure 1: *In precision-p binary floating-point arithmetic, in the normal range, the relative error due to rounding to nearest is always bounded by* $u = 2^{-p}$.

For instance, when we perform a floating-point multiplication, if a and b are the input FP operands, $z = ab$ is the exact result, and $\widehat{z} = \mathrm{RN}(z)$ is the computed result, then we have

$$(1 - u) \cdot z \leq \widehat{z} \leq (1 + u) \cdot z. \tag{5}$$

Assume that we wish to evaluate the product

$$a_1 \cdot a_2 \cdots a_n,$$

of n floating-point numbers, and that the product is evaluated as

$$\mathrm{RN}(\cdots \mathrm{RN}(\mathrm{RN}(a_1 \cdot a_2) \cdot a_3) \cdot \cdots) \cdot a_n). \tag{6}$$

Define π_n as the exact value of $a_1 \cdots a_n$, and $\widehat{\pi}_n$ as the computed value. A simple induction, based on (5), allows one to show

Theorem 1. *Let $a_1, \ldots, a_n$ be floating-point numbers, $\pi_n = a_1 \cdots a_n$, and $\widehat{\pi}_n$ the computed value using* (6)*. Then we have*

$$(1-u)^{n-1}\pi_n \leq \widehat{\pi}_n \leq (1+u)^{n-1}\pi_n. \tag{7}$$

See [1] for some results concerning the computation of the product of floating-point numbers. Therefore, the relative error of the computation, namely $|\widehat{\pi_n} - \pi_n|/\pi_n$ is upper-bounded by

$$\psi_{n-1} = (1+u)^{n-1} - 1.$$

One easily shows that, as long as $ku < 1$ (which always holds in practical cases),

$$k \cdot u \leq \psi_k \leq \gamma_k,$$

where γ_k is defined by (2). Although the bound ψ_{n-1} on the relative error of the computation of $a_1 \cdot a_2 \cdots a_n$ is very slightly[1] better than γ_{n-1}, the classical bound found in the literature is γ_{n-1}. The reason for this is that it is easier to manipulate in calculations.

And yet, in all our experiments, we observed a relative error less than $(n-1) \cdot u$. If we could prove that this is a valid bound, this would be even easier to manipulate. In the general case of an iterated product, we did not succeed in proving that. We could only automatically build cases, for each value of the precision p, for which the attained relative error is extremely close to, yet not larger than, $(n-1) \cdot u$ (see Section 5). However, in the particular case $n \leq 4$, one can prove that the relative error is less than $(n-1) \cdot u$. This is done as follows.

First, as noticed by Jeannerod and Rump [4], one may remark that the bound on the relative error due to rounding—i.e., (4)—can be slightly improved. Assume that t is a real number between 2^e and 2^{e+1}. We already know that $|t - \mathrm{RN}(t)| \leq 2^{e-p} = u \cdot 2^e$. Therefore:

- if $t \geq 2^e \cdot (1+u)$, then $|t - \mathrm{RN}(t)|/t \leq u/(1+u)$;
- if $t < 2^e \cdot (1+u)$, then $\mathrm{RN}(t) = 2^e$. Let $t = 2^e \cdot (1+\tau \cdot u)$, we have: $|t - \mathrm{RN}(t)|/t = \tau \cdot u/(1+\tau \cdot u)$. An elementary study shows that for $\tau \in [0,1)$, $\tau \cdot u/(1+\tau \cdot u) < u/(1+u)$.

Therefore the maximum relative error due to rounding is bounded[2] by $u/(1+u)$. A consequence of this is that u can be replaced by $u/(1+u)$ in (7). This is illustrated by Figure 2 (see p. 6). In the general case (that is, for any n), this improvement does not suffice to show Theorem 2, and yet, when $n \leq 4$, we can use the following result.

Property 1. *If $k \leq 3$ then*

$$\left(1 + \frac{u}{1+u}\right)^k < 1 + k \cdot u.$$

Proof. The simplest way to prove Property 1 is to separately consider the cases $k = 1, 2$, and 3:

[1] As long as nu is small enough in front of 1.

[2] Incidentally, if $\mathrm{RN} = \mathrm{RN}_{even}$, that error is attained when $t = 1 + u$, which shows that the bound cannot be improved further.

- the case $k = 1$ is straightforward:
- if $k = 2$, we have

$$\left(1 + \frac{u}{1+u}\right)^2 - (1+2u) = -\frac{u^2 \cdot (1+2u)}{(1+u)^2} < 0;$$

- if $k = 3$, we have

$$\left(1 + \frac{u}{1+u}\right)^3 - (1+3u) = -\frac{u^3 \cdot (3u+2)}{(1+u)^3} < 0.$$

□

By taking $k = n - 1$, we immediately deduce that for $n \leq 4$, the relative error of the iterative product of n FP numbers is bounded by $(n-1) \cdot u$.

Although we conjecture that this remains true for larger values of n, we did not succeed in proving that (notice that Property 1 is no longer true when $k \geq 4$). However, in the particular case of the computation of x^n, for some given FP number x and some positive integer n, we could prove the bound $(n-1) \cdot u$: our main result is Theorem 2 below.

1.3 The particular case of computing powers

In the following, we are interested in computing x^n, where x is a FP number and n is an integer. It is not difficult to show by induction that the bound provided by Theorem 1 applies not only to the case that was discussed above (computation of $\mathrm{RN}(\cdots \mathrm{RN}(\mathrm{RN}(x \cdot x) \cdot x) \cdot \cdots) \cdot x$) but to the larger class of recursive algorithms where the approximation to $x^{k+\ell}$ is deduced from approximations to x^k and x^ℓ by a FP multiplication. However, we will prove a (slightly) better bound only in the case where the algorithm used for computing x^n is Algorithm 1 below.

Algorithm 1 (naive-power(x, n))**.**

$y \leftarrow x$
for $k = 2$ *to* n ***do***
  $y \leftarrow \mathrm{RN}(x \cdot y)$
end for
return y

We will define $\widehat{x}_j$ as the value of variable y after the iteration corresponding to $k = j$ in the **for** loop of Algorithm 1. We have $\widehat{x}_2 = \mathrm{RN}(x^2)$, and $\widehat{x}_k = \mathrm{RN}(x \cdot \widehat{x}_{k-1})$. We wish to prove

Theorem 2. *Assume $p \geq 5$ (which holds in all practical cases). If*

$$n \leq \sqrt{2^{1/2} - 1} \cdot 2^{p/2},$$

then

$$|\widehat{x}_n - x^n| \leq (n-1) \cdot u \cdot x^n.$$

To prove Theorem 2, it suffices to prove it in the case $1 \leq x < 2$: in the following we will therefore assume that x lies in that range.

We prove Theorem 2 in Section 3. Before that, in Section 2, we give some preliminary results. In Section 4, we discuss the tightness of our new bound. Section 5 is devoted to a discussion on the possible generalization of this bound to the product of n floating-point numbers.

2 Preliminary results

In this section we give some preliminary results that will help to improve the bound of Theorem 1 in the specific case of the computation of integer powers. Let us start with an easy remark.

Remark 1. *Since $(1-u)^{n-1} \geq 1-(n-1)\cdot u$ for all $n \geq 2$ and $u \in [0,1]$, the left-hand bound of (7) suffices to show that $(1-(n-1)\cdot u)\cdot x^n \leq \widehat{x}_n$. In other words, to establish Theorem 2, we only need to improve on the right-hand bound of (7).*

Now, for $t \neq 0$, define

$$\overline{t} = \frac{t}{2^{\lfloor \log_2 |t| \rfloor}}.$$

We have,

Lemma 1. *Let t be a real number. If*

$$2^e \leq w \cdot 2^e \leq |t| < 2^{e+1}, e \in \mathbb{Z} \tag{8}$$

(in other words, if $|\overline{t}|$ is lower-bounded by w) then

$$\left| \frac{\mathrm{RN}(t) - t}{t} \right| \leq \frac{u}{w}.$$

Figure 2 illustrates Lemma 1, and Figure 3 illustrates this "wobbling" maximal relative error due to rounding.

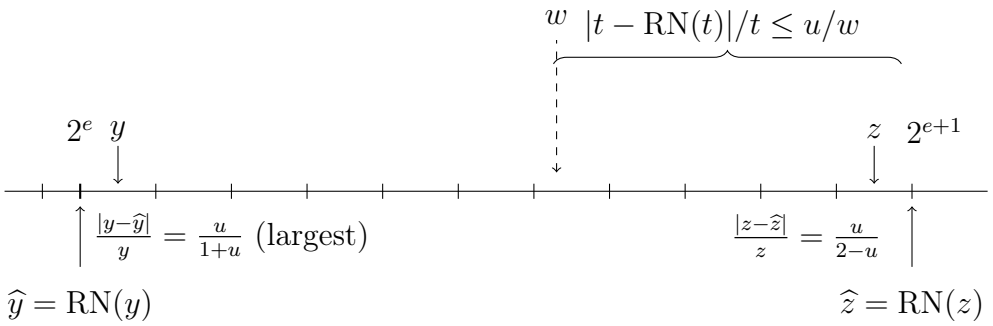

Figure 2: *The bound on the relative error due to rounding to nearest can be reduced to $u/(1+u)$. Furthermore, if we know that $\overline{t} = t/2^e$ is larger than w, then $|\mathrm{RN}(t) - t|/t$ is less than u/w.*

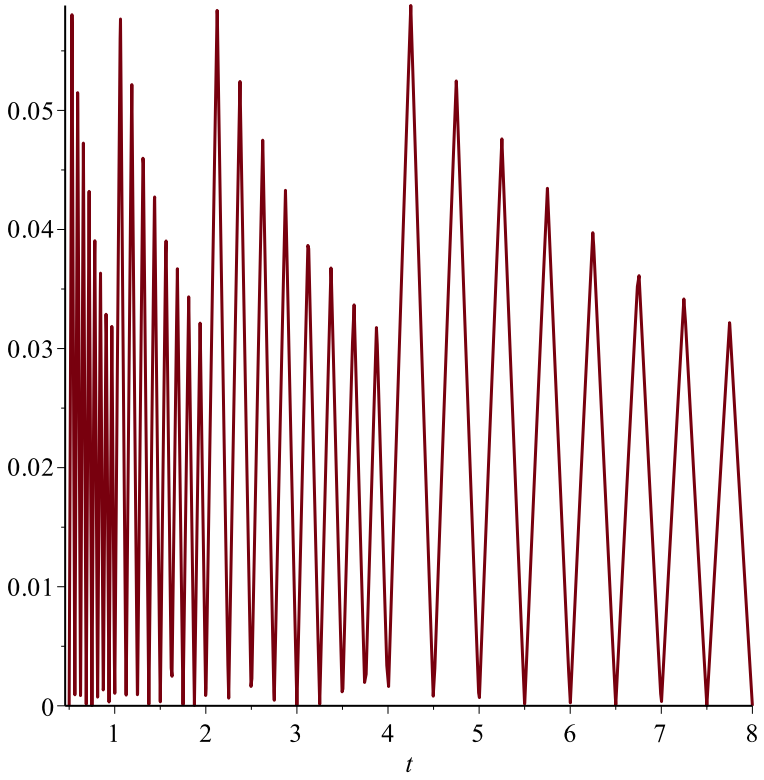

Figure 3: The relative error due to rounding, namely $|\mathrm{RN}(t) - t|/t$, for t between $1/5$ and 8.

Lemma 1 is an immediate consequence of (3) and (8). It is at the heart of our study: our problem will be to show that at least once in the execution of Algorithm 1 the number $x \cdot y$ is such that $\overline{x \cdot y}$ is large enough to sufficiently reduce the error bound on the corresponding FP multiplication $y \leftarrow \mathrm{RN}(x \cdot y)$, so that the overall relative error bound becomes smaller than $(n-1) \cdot u$. More precisely, we will show that, under some conditions, at least once, $\overline{x \cdot y}$ is larger than $1 + n^2u$, so that in (7) the term $(1+u)^{n-1}$ can be replaced by

$$(1+u)^{n-2} \cdot \left(1 + \frac{u}{1+n^2u}\right).$$

Therefore, we need to bound this last quantity. We have,

Lemma 2. *If* $0 \leq u \leq 2/(3n^2)$ *then*

$$(1+u)^{n-2} \cdot \left(1 + \frac{u}{1+n^2u}\right) \leq 1 + (n-1) \cdot u. \tag{9}$$

Proof. Proving Lemma 2 reduces to proving that the polynomial

$$P(u) = (1 + (n-1)u)(1+n^2u) - (1+u)^{n-2}(1+n^2u+u)$$

is ≥ 0 for $0 \leq u \leq 2/(3n^2)$.

Notice that for $u \geq 0$, we have

$$\ln(1+u) \leq u - \frac{u^2}{2} + \frac{u^3}{3}.$$

From $\ln(1+u) \leq u$ we also deduce that $(n-2)\ln(1+u) \leq (n-2)u \leq 1/(2n)$. For $0 \leq t \leq 1/6$, $e^t \leq 1 + t + \frac{3}{5}t^2$. Therefore, for $0 \leq u \leq 2/3n^2$, to prove that $P(u) \geq 0$ it suffices to prove that

$$\begin{gathered} Q(n,u) = (1+(n-1)\,u)\,(n^2u+1) \\ -\Big(1+(n-2)\,(u-1/2\,u^2+1/3\,u^3)+3/5\;(n-2)^2\,(u-1/2\,u^2+1/3\,u^3)^2\Big) \\ \times\,(n^2u+u+1) \geq 0. \end{gathered} \tag{10}$$

By defining $a = n^2u$, $Q(n,u) = R(n,a)$, with

$$\begin{gathered} R(n,a) = -\tfrac{1}{5}\,\tfrac{a^2(3\,a-2)}{n^2} + \tfrac{1}{10}\,\tfrac{a^2(29\,a+19)}{n^3} + \tfrac{1}{5}\,\tfrac{a^2\left(3\,a^2-17\,a-7\right)}{n^4} \\ -\tfrac{1}{30}\,\tfrac{a^3(82\,a-5)}{n^5} - \tfrac{1}{60}\,\tfrac{a^3\left(33\,a^2-187\,a+20\right)}{n^6} + \tfrac{1}{15}\,\tfrac{a^4(33\,a-8)}{n^7} \\ +\tfrac{1}{60}\,\tfrac{a^4\left(12\,a^2-153\,a+52\right)}{n^8} - \tfrac{1}{5}\,\tfrac{a^5(4\,a-7)}{n^9} - \tfrac{1}{15}\,\tfrac{a^5\left(a^2-14\,a+21\right)}{n^{10}} \\ +\tfrac{4}{15}\,\tfrac{a^6(a-2)}{n^{11}} - \tfrac{1}{15}\,\tfrac{a^6(5\,a-8)}{n^{12}} \\ +\tfrac{4}{15}\,\tfrac{a^7}{n^{13}} - \tfrac{4}{15}\,\tfrac{a^7}{n^{14}} \end{gathered} \tag{11}$$

Multiplying $R(n,a)$ by $5n^2/a^2$, we finally obtain

$$\begin{gathered} S(n,a) = -3\,a + 2 + \left(\tfrac{29}{2}\,a + \tfrac{19}{2}\right)n^{-1} + \tfrac{3\,a^2-17\,a-7}{n^2} - \tfrac{1}{6}\,\tfrac{a(82\,a-5)}{n^3} \\ -\tfrac{1}{12}\,\tfrac{a\left(33\,a^2-187\,a+20\right)}{n^4} + \tfrac{1}{3}\,\tfrac{a^2(33\,a-8)}{n^5} + \tfrac{1}{12}\,\tfrac{a^2\left(12\,a^2-153\,a+52\right)}{n^6} \\ -\tfrac{a^3(4\,a-7)}{n^7} - \tfrac{1}{3}\,\tfrac{a^3\left(a^2-14\,a+21\right)}{n^8} + \tfrac{4}{3}\,\tfrac{a^4(a-2)}{n^9} - \tfrac{1}{3}\,\tfrac{a^4(5\,a-8)}{n^{10}} \\ +\tfrac{4}{3}\,\tfrac{a^5}{n^{11}} - \tfrac{4}{3}\,\tfrac{a^5}{n^{12}} \end{gathered} \tag{12}$$

We wish to show that $S(n,a) \geq 0$ for $0 \leq a \leq 2/3$. Let us examine the terms of $S(n,a)$ separately. For a in the interval $[0, 2/3]$ and $n \geq 3$:

- the term $-3\,a + 2$ is always larger than 0;
- the term $\left(\frac{29}{2}\,a + \frac{19}{2}\right)n^{-1}$ is always larger than $19/(2n)$;
- the term $\frac{3\,a^2-17\,a-7}{n^2}$ is always larger than $-6/n$;
- the term $-\frac{1}{6}\,\frac{a(82\,a-5)}{n^3}$ is always larger than $-7/(10n)$;
- the term $-\frac{1}{12}\,\frac{a\left(33\,a^2-187\,a+20\right)}{n^4}$ is always larger than $-17/(10000n)$;
- the term $\frac{1}{3}\,\frac{a^2(33\,a-8)}{n^5}$ is always larger than $-3/(10000n)$;

- the term $\frac{1}{12}\,\frac{a^2\left(12\,a^2-153\,a+52\right)}{n^6}$ is always larger than $-69/(10000n)$;
- the term $-\frac{a^3(4\,a-7)}{n^7}$ is always larger than 0;
- the term $-\frac{1}{3}\,\frac{a^3\left(a^2-14\,a+21\right)}{n^8}$ is always larger than $-6/(10000n)$;
- the term $\frac{4}{3}\,\frac{a^4(a-2)}{n^9}$ is always larger than $-6/(100000n)$;
- the term $-\frac{1}{3}\,\frac{a^4(5\,a-8)}{n^{10}}$ is always larger than 0;
- the term $\frac{4}{3}\,\frac{a^5}{n^{11}}$ is always larger than 0;
- the term $-\frac{4}{3}\,\frac{a^5}{n^{12}}$ is always larger than $-1/(1000000n)$.

By summing all these lower bounds, we find that for $0 \le a \le 2/3$ and $n \ge 3$, $S(n,a)$ is always larger than $2790439/(1000000n)$. □

Let us now raise some remarks, that are direct consequences of Lemma 2.

Remark 2. *Assume $n \le \sqrt{2/3}\cdot 2^{p/2}$. If for some $k \le n$, we have $\mathrm{RN}(x\cdot\widehat{x}_{k-1}) \le x\cdot\widehat{x}_{k-1}$ (i.e., if in Algorithm 1 at least one rounding is done downwards), then $\widehat{x}_n \le (1+(n-1)\cdot u)x^n$.*

Proof. We have

$$\widehat{x}_n \le (1+u)^{n-2}x^n.$$

Lemma 2 implies that $(1+u)^{n-2}$ is less than $1+(n-1)\cdot u$. Therefore,

$$\widehat{x}_n \le (1+(n-1)\cdot u)x^n.$$

□

Remark 3. *Assume $n \le \sqrt{2/3}\cdot 2^{p/2}$. If there exists k, $1 \le k \le n-1$, such that $\overline{x\cdot\widehat{x}_k} \ge 1+n^2\cdot u$, then $\widehat{x}_n \le (1+(n-1)\cdot u)x^n$.*

Proof. By combining Lemma 1 and Lemma 2, if there exists k, $1 \le k \le n-1$, such that

$$\overline{x\cdot\widehat{x}_k} \ge 1+n^2\cdot u,$$

then

$$\widehat{x}_n \le (1+u)^{n-2}\cdot\left(1+\frac{u}{1+n^2u}\right)\cdot x^n \le (1+(n-1)\cdot u)\cdot x^n.$$

□

3 Proof of Theorem 2

The proof is articulated as follows

- first, we show that if x is close enough to 1, then when computing $\mathrm{RN}(x^2)$, the rounding is done downwards (i.e., $\mathrm{RN}(x^2) \leq x^2$), which implies, from Remark 2, that $\widehat{x}_n \leq (1 + (n-1) \cdot u)x^n$. This is the purpose of Lemma 3.
- then, we show that in the other cases, there is at least one $k \leq n-1$ such that $\overline{x \cdot \widehat{x}_k} \geq 1 + n^2 \cdot u$, which implies, from Remark 3, that $\widehat{x}_n \leq (1 + (n-1) \cdot u)x^n$.

Lemma 3. *Let $x = 1 + k \cdot 2^{-p+1} = 1 + 2ku, k \in \mathbb{N}$ (all FP numbers between 1 and 2 are of that form). We have $x^2 = 1 + 2k \cdot 2^{-p+1} + k^2 \cdot 2^{-2p+2}$, so that if $k < 2^{p/2-1}$, i.e., if $1 \leq x < 1 + 2^{p/2}u$, then $\widehat{x}_2 = 1 + 2k \cdot 2^{-p+1} < x^2$, which, by Remark 2, implies $\widehat{x}_n \leq (1 + (n-1)u) \cdot x^n$.*

Remark 3 and Lemma 3 imply that to prove Theorem 2, we are reduced to examine the case where $1 + 2^{p/2}u \leq x < 2$ and we assume $u \leq 2/(3n^2)$, i.e., $n < \sqrt{2/3} \cdot 2^{p/2}$ (later on, we will see that a stronger assumption is necessary). For that, we distinguish between the cases where $x^2 \leq 1 + n^2u$ and $x^2 > 1 + n^2u$.

3.1 First case: if $x^2 \leq 1 + n^2u$

From $x \geq 1 + 2^{p/2}u \geq 1 + nu$, we deduce

$$x^n \geq (1 + nu)^n > 1 + n^2u,$$

so that, from Remark 2, we can assume that

$$\widehat{x}_{n-1} \cdot x > (1 + n^2u)$$

(otherwise, at least one rounding was done downwards, which implies Theorem 2). Therefore

- if $\widehat{x}_{n-1}x < 2$, then $\overline{\widehat{x}_{n-1}x} \geq (1+n^2u)$, so that, from Remark 3, $x^n \leq (1+(n-1)\cdot u)\cdot x^n$;
- if $\widehat{x}_{n-1}x \geq 2$, then let k be the smallest integer such that $\widehat{x}_{k-1}x \geq 2$. Notice that since we have assumed that $x^2 \leq 1 + n^2u$, we necessarily have $k \geq 3$. We have

  $$\widehat{x}_{k-1} \geq \frac{2}{x} \geq \frac{2}{\sqrt{1 + n^2u}},$$

  hence

  $$\widehat{x}_{k-2} \cdot x \geq \frac{2}{\sqrt{1 + n^2u} \cdot (1 + u)}. \tag{13}$$

  Now, define

  $$\alpha_p = \sqrt{\left(\frac{2^{p+1}}{2^p + 1}\right)^{2/3} - 1}.$$

For all $p \geq 5$, $\alpha_p \geq \alpha_5 = 0.74509\cdots$, and $\alpha_p \leq \sqrt{2^{2/3}-1} = 0.7664209\cdots$. If

$$n \leq \alpha_p \cdot 2^{p/2}, \tag{14}$$

then

$$1 + n^2 u \leq \left(\frac{2^{p+1}}{2^p+1}\right)^{2/3},$$

so that

$$(1+n^2u)^{3/2} \cdot (1+u) \leq 2,$$

so that

$$\frac{2}{\sqrt{1+n^2u}\cdot(1+u)} \geq 1 + n^2u.$$

Therefore, from (13), we have

$$\widehat{x}_{k-2} \cdot x \geq 1 + n^2u.$$

Also, $\widehat{x}_{k-2} \cdot x$ is less than 2, since k was assumed to be the smallest integer such that $\widehat{x}_{k-1}x \geq 2$. Therefore

$$\overline{\widehat{x}_{k-2} \cdot x} \geq 1 + n^2u.$$

Which implies, by Remark 3, that $x^n \leq (1+(n-1)\cdot u)\cdot x^n$. So, to summarize this first case, if $x^2 \leq 1+n^2u$ and $n \leq \alpha_p \cdot 2^{p/2}$, then the conclusion of Theorem 2 holds.

3.2 Second case: if $x^2 > 1 + n^2u$

First, if $x^2 < 2$ then we deduce from Remark 3 that $x^n \leq (1+(n-1)\cdot u)\cdot x^n$. The case $x^2 = 2$ is impossible (x is a floating-point number, thus it cannot be irrational). Therefore let us now assume that $x^2 > 2$. We also assume that $x^2 < 2+2n^2u$ (otherwise, we would have $\overline{(x^2)} \geq 1 + n^2u$, so that we could apply Remark 3). Hence, we have

$$\sqrt{2} < x < \sqrt{2+2n^2u}.$$

From this we deduce

$$x^{n-1} < (2+2n^2u)^{\frac{n-1}{2}},$$

therefore, using Theorem 1,

$$\widehat{x}_{n-1} < (2+2n^2u)^{\frac{n-1}{2}} \cdot (1+u)^{n-2},$$

which implies

$$x \cdot \widehat{x}_{n-1} < (2+2n^2u)^{n/2} \cdot (1+u)^{n-2}. \tag{15}$$

Define

$$\beta = \sqrt{2^{1/3}-1} = 0.5098245285339\cdots$$

If $n \leq \beta \cdot 2^{p/2}$ then $2+2n^2u \leq 2^{4/3}$, so that we find

$$(2+2n^2u)^{n/2} \cdot (1+u)^{n-2} \leq 2^{2n/3} \cdot (1+u)^{n-2}. \tag{16}$$

- if $n = 3$, the bound on $x \cdot \widehat{x}_{n-1}$ derived from (15) and (16) is equal to $4 \cdot (1 + u)$. Therefore either $x \cdot \widehat{x}_{n-1} < 4$, or $x \cdot \widehat{x}_{n-1}$ will be rounded downwards when computing $\widehat{x}_n$ (in which case we already know from Remark 2 that the conclusion of Theorem 2 holds);

- if $n \geq 4$, consider function

$$g(t) = 2^{t-1} - 2^{2t/3} \left(1 + \frac{1}{2^p}\right)^{t-2} = 2^{2t/3} \left[2^{t/3-1} - \left(1 + \frac{1}{2^p}\right)^{t-2}\right].$$

  It is a continuous function, and it goes to $+\infty$ as $t \to +\infty$. We have:

$$g(t) = 0 \Leftrightarrow t = \frac{\log(2) + 2\log\left(1 + \frac{1}{2^p}\right)}{\frac{1}{3}\log(2) - \log\left(1 + \frac{1}{2^p}\right)}.$$

  Hence, function g has one root only, and as soon as $p \geq 5$, that root is strictly less than 4. From this, we deduce that if $p \geq 5$, then $g(t) > 0$ for all $t \geq 4$. Hence, using (15) and (16), we deduce that if $p \geq 5$ then $x \cdot \widehat{x}_{n-1} < 2^{n-1}$.

Now that we have shown that[3] if $n \leq \beta \cdot 2^{p/2}$ then

$$x \cdot \widehat{x}_{n-1} < 2^{n-1},$$

let us define k as the smallest integer for which $x \cdot \widehat{x}_{k-1} < 2^{k-1}$. We now know that $k \leq n$, and (since we are assuming $x^2 > 2$), we have $k \geq 3$. The minimality of k implies that $x \cdot \widehat{x}_{k-2} \geq 2^{k-2}$, which implies that $\widehat{x}_{k-1} = \mathrm{RN}(x \cdot \widehat{x}_{k-2}) \geq 2^{k-2}$. Therefore, $\widehat{x}_{k-1}$ and $x \cdot \widehat{x}_{k-1}$ belong to the same binade, therefore,

$$\overline{x \cdot \widehat{x}_{k-1}} \geq x > \sqrt{2}. \tag{17}$$

The constraint $n \leq \beta \cdot 2^{p/2}$ implies

$$1 + n^2 u \leq 1 + \beta^2 = 2^{1/3} < \sqrt{2}. \tag{18}$$

By combining (17) and (18) we obtain

$$\overline{x \cdot \widehat{x}_{k-1}} \geq 1 + n^2 u.$$

Therefore, using Remark 3, we deduce that $\widehat{x}_n \leq (1 + (n-1) \cdot u) \cdot x^n$.

3.3 Combining both cases

One easily sees that for all $p \geq 5$, α_p is larger than β. Therefore, combining the conditions found in the cases $x^2 \leq 1 + n^2 u$ and $x^2 > 1 + n^2 u$, we deduce that if $p \geq 5$ and $n \leq \beta \cdot 2^{p/2}$, then for all x,

$$(1 - (n-1) \cdot u) \cdot x^n \leq \widehat{x}_n \leq (1 + (n-1) \cdot u) \cdot x^n.$$

Q.E.D.

[3] Unless $n = 3$ and $x \cdot \widehat{x}_{n-1} \geq 4$ but in that case we have seen that the conclusion of Theorem 2 holds.

Notice that the condition $n \leq \beta \cdot 2^{p/2}$ is not a huge constraint. The table below gives the maximum value of n that satisfies that condition, for the various binary formats of the IEEE 754-2008 Standard for Floating-Point Arithmetic.

p	$n_{\max}$
24	2088
53	48385542
113	51953580258461959

For instance, in the binary32/single precision format, with the smallest n larger than that maximum value (i.e., 2089), x^n will underflow as soon as $x \leq 0.95905406$ and overflow as soon as $x \geq 1.0433863$. In the binary64/double precision format, with $n = 4385543$, x^n will underflow as soon as $x \leq 0.999985359$ and overflow as soon as $x \geq 1.000014669422$. With the binary113/quad precision format, the interval in which function x^n does not under- or overflow is even narrower and, anyway, computing $x^{51953580258461959}$ by Algorithm 1 would at best require months of computation on current machines.

4 Is the bound of Theorem 2 tight?

For very small values of p, it is possible to check all possible values of x (we can assume $1 \leq x < 2$, so that we need to check 2^{p-1} different values), using a Maple program that simulates a precision-p floating-point arithmetic. Hence, for small values of p and reasonable values of n it is possible to compute the actual maximum relative error of Algorithm 1. For instance, Tables 1 and 2 present the actual maximum relative errors for $p = 8$ and 9, respectively, and various values of n.

Table 1: Actual maximum relative error of Algorithm 1 assuming precision $p = 8$, compared with the usual bound γ_{n-1} and our bound $(n-1)u$. The term n_{max} designs the largest value of n for which Theorem 2 holds, namely $\sqrt{2^{1/2} - 1} \cdot 2^{p/2}$

n	actual maximum	γ_{n-1}	our bound
3	$1.35988u$	$2.0157u$	$2u$
4	$1.73903u$	$3.0355u$	$3u$
5	$2.21152u$	$4.06349u$	$4u$
6	$2.53023u$	$5.099601u$	$5u$
7	$2.69634u$	$6.1440u$	$6u$
$8 = n_{max}$	$3.42929u$	$7.1967u$	$7u$

For larger values, we have some results (notice that beyond single precision—$p = 24$—exhaustive testing is out of reach):

- for single precision arithmetic ($p = 24$) and $n = 6$, the actual largest relative error is $4.328005619u$. It is attained for $x = 8473808/2^{23} \approx 1.010156631$;
- for double precision arithmetic ($p = 53$) and $n = 6$, although finding the actual largest relative error is out of reach, we could find an interesting case: for

Table 2: Actual maximum relative error of Algorithm 1 assuming precision $p = 9$, compared with the usual bound γ_{n-1} and our bound $(n-1)u$. The term n_{max} designs the largest value of n for which Theorem 2 holds, namely $\sqrt{2^{1/2}-1}\cdot 2^{p/2}$

n	actual maximum	γ_{n-1}	our bound
6	$2.677u$	$5.049u$	$5u$
7	$2.975u$	$6.071u$	$6u$
8	$3.435u$	$7.097u$	$7u$
9	$4.060u$	$8.1269u$	$8u$
10	$3.421u$	$9.1610u$	$9u$
$11 = n_{max}$	$3.577u$	$10.199u$	$10u$

$x = 4507062722867963/2^{52} \approx 1.000768961671552714 7761$, the relative error is $4.7805779\cdots u$

- for quad precision arithmetic ($p = 113$) and $n = 6$, although finding the actual largest relative error is out of reach, we could find an interesting case: for

$$x = 5192324351407105984705482084151108/2^{112}$$
$$\approx 1.00000529493459780998863520374963 65983,$$

the relative error is $4.8827888\cdots u$

- for single precision arithmetic ($p = 24$) and $n = 10$, the actual largest relative error is $7.059603149u$. It is attained for $x = 8429278/2^{23} \approx 1.004848242$;

- for double precision arithmetic ($p = 53$) and $n = 10$, although finding the actual largest relative error is out of reach, we could find an interesting case: for $x = 4503796447992526/2^{52} \approx 1.0000437029572597 5026$, the relative error is $7.9534189\cdots u$.

Notice that we can use the maximum relative error of single precision and "inject it" in the inductive reasoning that led to Theorem 1 to show that *in single-precision arithmetic, and if* $n \geq 10$ then

$$(1-7.06u)(1-u)^{n-10}x^n \leq \widehat{x}_n \leq (1+7.06u)(1+u)^{n-10}x^n.$$

Then, by replacing u by 2^{-24} and through an elementary study of the function

$$\varphi(t) = \left[(1+7.06\cdot 2^{-24})(1+2^{-24})^{t-10}-1\right]\cdot 2^{24} - t$$

one easily deduces that for $10 \leq n \leq 2088$, we always have

$$\left|\frac{\widehat{x}_n - x^n}{x^n}\right| \leq (n-2.8104)\cdot u.$$

5 What about iterated products ?

Assume now that, still in precision-p binary FP arithmetic, we wish to evaluate the product

$$a_1 \cdot a_2 \cdots \cdots \cdot a_n,$$

of n floating-point numbers. We assume that the product is evaluated as

$$\mathrm{RN}(\cdots \mathrm{RN}(\mathrm{RN}(a_1 \cdot a_2) \cdot a_3) \cdot \cdots) \cdot a_n).$$

Define π_k as the exact value of $a_1 \cdots a_k$, and $\widehat{\pi}_k$ as the computed value. As already discussed in Section 1.2, ve have

$$(1-u)^{n-1}\pi_n \leq \widehat{\pi}_n \leq (1+u)^{n-1}\pi_n, \tag{19}$$

which implies that the relative error $|\pi_n - \widehat{\pi}_n|/\pi_n$ is upper-bounded by γ_{n-1}. We conjecture that the error is upper-bounded by $(n-1)u$. Let us now show how to build a_1, a_2, ..., a_n so that the relative error becomes extremely close to $(n-1)\cdot u$.

Define $a_1 = 1 + k_1 \cdot 2^{-p+1}$, and $a_2 = 1 + k_2 \cdot 2^{-p+1}$. We have

$$\pi_2 = a_1 a_2 = 1 + (k_1 + k_2) \cdot 2^{-p+1} + k_1 k_2 \cdot 2^{-2p+2}.$$

If k_1 and k_2 are not too large, $1 + (k_1 + k_2) \cdot 2^{-p+1}$ is a FP number. To maximize the relative error, we wish $k_1 + k_2$ to be as small as possible, while $k_1 k_2 \cdot 2^{-2p+2}$ is as close as possible to 2^{-p}. Hence a natural choice is

$$k_1 = k_2 = \left\lfloor 2^{\frac{p}{2}-1} \right\rfloor,$$

which gives $\widehat{\pi}_2 < \pi_2$. Now, if at step $i-1$ we have

$$\widehat{\pi}_i = 1 + g_i \cdot 2^{-p+1}, \text{ with } \widehat{\pi}_i < \pi_i,$$

we choose a_{i+1} of the form $1 + k_{i+1}2^{-p+1}$, with

- $k_{i+1} = \left\lceil \frac{2^{p-2}}{g_i} - 1 \right\rceil$ if $g_i \leq 2^{\frac{p}{2}-1}$;
- $k_{i+1} = -\left\lfloor \frac{2^{p-2}}{g_i} + 1 \right\rfloor$ otherwise.

For instance, in single precision ($p = 24$), the first values a_i generated by this strategy are

$$\begin{aligned} a_1 &= 4097/4096 \\ a_2 &= 4097/4096 \\ a_3 &= 8387583/8388608 \\ a_4 &= 8387241/8388608 \\ a_5 &= 262221/262144 \\ a_6 &= 8387601/8388608 \\ a_7 &= 8387279/8388608 \end{aligned}$$

Table 3 gives examples of the relative errors achieved with the values a_i generated by this method, for various values of p and n. As one can easily see, the relative error is always very close to, but less than $(n-1)\cdot u$.

Table 3: Relative errors achieved with the values a_i generated by our method of Section 5.

p	n	relative error
24	10	$8.99336984\cdots u$
24	100	$98.9371972591\cdots u$
53	10	$8.99999972447\cdots u$
53	100	$98.9999970091\cdots u$
113	10	$8.99999999999999973119\cdots u$
113	100	$98.99999999999999701662\cdots u$

6 Conclusion

We have shown that, under mild conditions, the relative error of the computation of x^n in floating-point arithmetic using the "naive" algorithm is upper bounded by $(n-1)\cdot u$. This bound is simpler and slightly better than the previous bound. We conjecture that the same bound holds in the more general case of the computation of the product of n floating-point numbers. In that case, we have provided examples that show that the actual error can be very close to $(n-1)\cdot u$.

References

[1] S. Graillat. Accurate floating point product and exponentiation. *IEEE Transactions on Computers*, 58(7):994–1000, 2009.

[2] N. J. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, Philadelphia, PA, 2nd edition, 2002.

[3] IEEE Computer Society. *IEEE Standard for Floating-Point Arithmetic*. IEEE Standard 754-2008, August 2008. Available at `http://ieeexplore.ieee.org/servlet/opac?punumber=4610933`.

[4] C.-P. Jeannerod and S. M. Rump. On relative errors of floating-point operations: optimal bounds and applications. Research report hal-00934443, available at `http://hal.inria.fr/hal-00934443`.

[5] C.-P. Jeannerod and S. M. Rump. Improved error bounds for inner products in floating-point arithmetic. *SIAM J. Matrix Anal. Appl.*, 34(2):338–344, 2013.

[6] J.-M. Muller, N. Brisebarre, F. de Dinechin, C.-P. Jeannerod, V. Lefèvre, G. Melquiond, N. Revol, D. Stehlé, and S. Torres. *Handbook of Floating-Point Arithmetic*. Birkhäuser Boston, 2010.

[7] N. Revol and F. Rouillier. *MPFI (Multiple Precision Floating-point Interval library)*, 2009. Available at `http://gforge.inria.fr/projects/mpfi`.

[8] S. M. Rump. Error estimation of floating-point summation and dot product. *BIT*, 52(1):201–220, 2012.