Numer Algor (2015) 70:653–667
DOI 10.1007/s11075-015-9967-8

ORIGINAL PAPER

On the maximum relative error when computing integer powers by iterated multiplications in floating-point arithmetic

Stef Graillat · Vincent Lefèvre · Jean-Michel Muller

Received: 31 March 2014 / Accepted: 18 January 2015 / Published online: 1 February 2015

Abstract 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

Mathematics Subject Classification (2010) 15-04 · 65G99 · 65-04

1 Introduction

1.1 Floating-point arithmetic and rounding errors

When critical applications are at stake, one may need *certain* yet *tight* error bounds on the results of numerical computations. The manipulation of these error bounds (either paper-and-pencil manipulation, or dynamical error analysis) will be made easier if these bounds are *simple*. This paper deals with the calculation of a certain, tight and

S. Graillat (✉)
Sorbonne Universités, UPMC Univ Paris 06, UMR 7606, LIP6, F-75005, Paris, France
e-mail: stef.graillat@lip6.fr

S. Graillat
CNRS, UMR 7606, LIP6, F-75005, Paris, France

V. Lefèvre
Inria, Laboratoire LIP, Université de Lyon, Lyon, France

J. M. Muller
CNRS, Laboratoire LIP, Université de Lyon, Lyon, France

simple error bound for the evaluation of integer powers by the iterative algorithm in floating-point arithmetic.

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, 7], provided that no underflow (i.e., no subnormal values are generated) or overflow occurs (the underflow/overflow issues are briefly discussed in Section 5). 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 $2^{p-1} \le |X| \le 2^p - 1$.

We assume that the arithmetic operations are *correctly rounded*, and that rounding to nearest is used. We denote by RN the rounding function (which means that when the operation $a \top b$ is performed, the returned value is $\mathrm{RN}(a \top b)$). Our error bounds will be given assuming, for the rounding function RN, any choice in case of a tie. However, when we build examples (for instance for checking how tight are the obtained bounds), we use round to nearest *ties to even*.

Recently, classic error bounds for summation and dot product have been improved by Jeannerod and Rump [5, 8]. 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

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

(notice that there is no restriction on n), which is better than the previous bound [2, p.63]

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

where[1]

$$\gamma_n = \frac{n \cdot u}{1 - n \cdot u} \tag{2}$$

We are interested in finding if a similar simplification is possible in the particular case of the computation of an integer power x^n. More precisely, we wish to know if for "reasonable" values of n the result computed using the "naive", iterative, algorithm (Algorithm 1 below) is always within relative error $(n-1) \cdot u$ from the exact result.

We performed exhaustive tests in binary32 (single precision) for all $x \in [1; 2[$ until overflow for x^n, and the relative error was always less than $(n-1) \cdot u$.

[1] We assume that $n \cdot u < 1$ when using γ_n.

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), provided that n is less than $\sqrt{2^{1/3}-1}/\sqrt{u}$. This restriction on n, discussed in Section 3.3, is not a problem for wide FP formats (e.g., binary64 or larger). It may be a significant constraint for small formats (binary32 or smaller).

1.2 Relative error due to roundings

Let a and b be floating-point numbers whose product $z = ab$ is positive. Let $\widehat{z} = \mathrm{RN}(z)$ be their computed product. It is well known that

$$(1-u)\cdot z \leq \widehat{z} \leq (1+u)\cdot z. \quad (3)$$

Now, assume that we wish to evaluate the non-negative product $a_1 \cdot a_2 \cdots a_n$ of n floating-point numbers, and that the product is evaluated as

$$\mathrm{RN}(\mathrm{RN}(\cdots \mathrm{RN}(\mathrm{RN}(a_1 \cdot a_2)\cdot a_3)\cdot \cdots)\cdot a_n). \quad (4)$$

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 (3), allows one to show

Theorem 1 *Let $a_1, \ldots, a_n$ be floating-point numbers whose product is nonnegative, $\pi_n = a_1 \cdots a_n$, and $\widehat{\pi}_n$ the computed value using (4). Then we have*

$$(1-u)^{n-1}\pi_n \leq \widehat{\pi}_n \leq (1+u)^{n-1}\pi_n. \quad (5)$$

Therefore, the relative error of the computation, namely $|\widehat{\pi}_n - \pi_n|/\pi_n$ is upper-bounded by $(1+u)^{n-1}-1$, which is less than γ_{n-1} as long as $(n-1)\cdot u < 1$ (which always holds in practical cases). See for instance [1].

In our experiments, we always observed a relative error less than $(n-1)\cdot u$ for n until $\lesssim 2^{p/2}$. If this was a valid bound, it would be slightly better, and easier to manipulate than γ_{n-1}. In the general case of an iterated product, we did not succeed in proving that. We could only automatically build cases for which the attained relative error is extremely close to, yet not larger than, $(n-1)\cdot u$ for $n \lesssim 2^{p/2}$ (see Section 6). We also found counterexamples[2] (in the special case of the computation of x^n) for $n \approx 2^p$. However, in the particular case $n \leq 4$, one can easily prove that the relative error is less than $(n-1)\cdot u$. This is done as follows.

First, as noticed by Knuth [6], and later on used by Jeannerod and Rump [4] to improve classical results, the bound u on the relative error due to rounding can be

[2] Good candidates are machine numbers less than but very close to $2^{m/q}$, where m and q are small integers, such that $\widehat{\pi}_{k+q}$ and $\widehat{\pi}_k$ have the same significand for some k.

slightly improved: if t is a floating-point number, then $|t - \mathrm{RN}(t)|/t \leq u/(1+u)$ (incidentally, if RN is round-to-nearest ties to even, that bound is attained when $t = 1 + u$, which shows that the bound cannot be improved further).

A consequence of this is that u can be replaced by $u/(1+u)$ in (5). In the general case (that is, for any n), this improvement does not suffice to show that the relative error is less than $(n-1) \cdot u$, 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 Straightforward by separately considering the cases $k = 1, 2$, and 3. □

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 $(n-1) \cdot u$.

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. One shows 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 (i.e., we compute powers using iterated multiplications). Incidentally, when n is not known at compile-time (i.e., it is not a constant), computing x^n using a "smart" algorithm such as exponentiation by squaring is not so efficiently implementable in modern pipeline architectures, and it also requires tests that may be slow (because of non-predictable branches) in front of a floating-point multiplication. Hence, although the logarithmic-time smart algorithms necessarily beat the linear-time iterated product algorithm ultimately, our tests show that this is not the case until n is around 10. Also, there are a few applications where one needs to know all the powers x^i, $i \leq n$ of a given number x. For these applications, obviously, the iterated product algorithm is of interest.

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

$\widehat{x}_1 \leftarrow x$
for $k = 2$ *to* n ***do***
  $\widehat{x}_k \leftarrow \mathrm{RN}(x \cdot \widehat{x}_{k-1})$
end for
return $\widehat{x}_n$

We wish to prove

Theorem 2 *Assume $p \geq 5$. If*

$$n \leq \sqrt{2^{1/3} - 1} \cdot 2^{p/2},$$

then

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

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, and in Section 5, we raise brief remarks about possible underflow/overflow issues. Section 6 is devoted to a discussion on the possible generalization of this bound to the product of n floating-point numbers.

2 Preliminary results

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 $u \in [0, 1]$, (5) 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 show that $\widehat{x}_n \leq (1 + (n-1) \cdot u) \cdot x^n$.

We also have,

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

$$2^e \leq w \cdot 2^e \leq |t| < 2^{e+1}, e \in \mathbb{Z} \tag{6}$$

then

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

Lemma 1 is a straightforward consequence of the relations $|\,\mathrm{RN}(t) - t| \leq u \cdot 2^e$ and $w \cdot 2^e \leq |t|$.

For $t \neq 0$, we will define $\overline{t}$ as the *significand* of t, namely

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

Lemma 1 is at the heart of our study: if at least once in the execution of Algorithm 1, $x \cdot \widehat{x}_{k-1}$ is such that $\overline{x \cdot \widehat{x}_{k-1}}$ is large enough to sufficiently reduce the error bound on the corresponding FP multiplication $\widehat{x}_k \leftarrow \mathrm{RN}(x \cdot \widehat{x}_{k-1})$, then the overall relative error bound becomes smaller than $(n-1) \cdot u$. More precisely, we will show that, under some conditions, at least once, $\overline{x \cdot \widehat{x}_{k-1}}$ is larger than $1 + n^2u$, so that in (5) 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* $0 \le u \le 2/(3n^2)$ *then*

$$(1+u)^{n-2} \cdot \left(1 + \frac{u}{1+n^2u}\right) \le 1 + (n-1) \cdot u. \tag{7}$$

Proof Proving Lemma 2 reduces to proving that the polynomial

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

is ≥ 0 for $0 \le u \le 2/(3n^2)$.

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

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

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

$$\begin{aligned} Q(n,u) = (1 + (n-1)\,u)\left(n^2u+1\right) \\ -\left(1 + (n-2)\left(u - 1/2\,u^2 + 1/3\,u^3\right) + 3/5\ (n-2)^2\left(u - 1/2\,u^2 + 1/3\,u^3\right)^2\right) \\ \times \left(n^2u + u + 1\right) \ge 0. \end{aligned} \tag{8}$$

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

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

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

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

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 $\frac{\frac{29}{2}\,a + \frac{19}{2}}{n}$ 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 $0 \leq a \leq 2/3$ and $n \geq 3$, $S(n, a)$ is always larger than $2790439/(1000000n)$. □

Let us deduce two consequences of Lemma 2. The most important is Lemma 3 below, which is the basis of almost all subsequent results. It says that if in Algorithm 1 at least one rounding is done towards zero, the desired result is obtained.

Lemma 3 *Assume* $n \leq \sqrt{2/3} \cdot 2^{p/2}$. *If for some* $k \leq n$, *we have* $\mathrm{RN}(x \cdot \widehat{x}_{k-1}) \leq x \cdot \widehat{x}_{k-1}$, *then* $\widehat{x}_n \leq (1 + (n-1) \cdot u)x^n$.

Proof We have

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

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

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

□

Now, 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

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

so that:

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

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., $\mathrm{RN}(x^2) \le x^2$), which implies, from Lemma 3, that $\widehat{x}_n \le (1 + (n-1) \cdot u)x^n$. This is the purpose of Lemma 4.
- Then, we show that in the other cases, there is at least one $k \le n-1$ such that $\overline{x \cdot \widehat{x}_k} \ge 1 + n^2 \cdot u$, which implies, from Remark 2, that $\widehat{x}_n \le (1 + (n-1) \cdot u)x^n$.

Lemma 4 *Let $x = 1 + k \cdot 2^{-p+1} = 1 + 2ku$, where $k \in \mathbb{N}$ (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 $1 \le x < 1 + 2^{p/2}u$, then $\widehat{x}_2 = 1 + 2k \cdot 2^{-p+1} < x^2$, which, by Lemma 3, implies $\widehat{x}_n \le (1 + (n-1)u) \cdot x^n$.*

Assume $u \le 2/(3n^2)$, i.e., $n < \sqrt{2/3} \cdot 2^{p/2}$ (later on, we will see that a stronger assumption is necessary). Remark 2 and Lemma 4 imply that to prove Theorem 2, we are reduced to examine the case where $1 + 2^{p/2}u \le x < 2$. For that, we distinguish between the cases where $x^2 \le 1 + n^2 u$ and $x^2 > 1 + n^2 u$.

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

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

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

so that, from Lemma 3, we can assume that

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

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

- if $\widehat{x}_{n-1}x < 2$, then $\overline{\widehat{x}_{n-1}x} \ge (1 + n^2 u)$, so that, from Remark 2, $x^n \le (1 + (n-1) \cdot u) \cdot x^n$;

- 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

  $$\widehat{x}_{k-2} \cdot x \geq \frac{2}{\sqrt{1+n^2u} \cdot (1+u)}. \tag{11}$$

  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

  $$n \leq \alpha_p \cdot 2^{p/2}, \tag{12}$$

  then

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

  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 (11), 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 2, 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 2 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 2). 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

$$x \cdot \widehat{x}_{n-1} < (2 + 2n^2u)^{n/2} \cdot (1 + u)^{n-2}. \tag{13}$$

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

$$(2 + 2n^2u)^{n/2} \cdot (1 + u)^{n-2} \leq 2^{2n/3} \cdot (1 + u)^{n-2}. \tag{14}$$

- if $n = 3$, the bound on $x \cdot \widehat{x}_{n-1}$ derived from (13) and (14) is $4 \cdot (1+u)$. 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 know from Lemma 3 that the conclusion of Theorem 2 holds);
- if $n \geq 4$, consider function

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

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

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

  Hence, g has a single root, 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 (13) and (14), 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,[4] therefore,

$$\overline{x \cdot \widehat{x}_{k-1}} \geq x > \sqrt{2}. \tag{15}$$

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

$$1 + n^2u \leq 1 + \beta^2 = 2^{1/3} < \sqrt{2}. \tag{16}$$

By combining (15) and (16) we obtain

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

Therefore, using Remark 2, we deduce that $\widehat{x}_n \leq (1 + (n-1) \cdot u) \cdot x^n$.

[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.

[4] A *binade* is the interval between two consecutive integer powers of 2.

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 $x^2 \leq 1 + n^2u$ and $x^2 > 1 + n^2u$, 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.

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 years 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.

For larger values, we have some results (notice that beyond single precision—$p =$ 24—exhaustive testing is either very costly or 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 would require months of calculation, we could find an interesting case: for $x = 4507062722867963/2^{52} \approx 1.0007689616715527147761$, the relative error is $4.7805779 \cdots u$;

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 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

$$x = 5192324351407105984705482084151108/2^{112}$$
$$\approx 1.0000052949345978099886352037496365983,$$

  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 1.0000437029572597502$6, 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

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

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	$3.421u$	$9.1610u$	$9u$
$11 = n_{max}$	$3.577u$	$10.199u$	$10u$

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 $10 \le n \le 2088$, we always have

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

5 A brief remark on underflow and overflow

As stated in the introduction, the results presented in this paper (assuming an unbounded exponent range) apply to "real" floating-point arithmetic provided that no underflow or overflow occur. When considering "general" iterated products, intermediate underflows are a real concern: they may make a final result very inaccurate, and this may be rather difficult to notice when the IEEE 754 exceptions are not supported, since the returned final result may lie well in the normal floating-point range. Overflows are less deceiving, but they may be difficult to manage: one may have an overflow appearing in an intermediate result (leading to an infinite or NaN final result being returned) even when the exact product is of magnitude much smaller than the overflow threshold.

However, when we are concerned with powers only, these pitfalls disappear. One easily shows that when evaluating a power using Algorithm 1:

- if an intermediate result underflows then the final result will be less than or equal to the minimum positive normal number in absolute value, so that this will not go unnoticed;
- if an intermediate result overflows then the exact final result is larger than $\Omega/(1+u)^{n-1}$ in absolute value, where Ω is the largest finite floating-point number.

6 What about iterated products ?

Assume now that, still in precision-p binary FP arithmetic, we wish to evaluate the product $a_1 \cdot a_2 \cdot \cdots \cdot a_n$, 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, we have

$$(1 - u)^{n-1}\pi_n \le \widehat{\pi}_n \le (1 + u)^{n-1}\pi_n, \tag{17}$$

which implies that the relative error $|\pi_n - \widehat{\pi}_n|/\pi_n$ is upper-bounded by γ_{n-1}, defined for $(n - 1) \cdot u < 1$.

Here we seek to build a sequence $a_1, a_2, a_3, \ldots$, trying to maximize the relative error at each step. For this purpose, we will choose each a_n so that all the roundings occur in the same direction, and this direction must be the upward one to have a chance to get a relative error larger than $(n - 1) \cdot u$ at step n.

Table 3 Relative errors achieved with the values a_i generated by our method of Section 6

p	n	relative error
24	10	$8.99401809 \cdots u$
24	100	$98.92221853 \cdots u$
53	10	$8.99999971848 \cdots u$
53	100	$98.99999680546 \cdots u$
113	10	$8.99999999999999972714 \cdots u$
113	100	$98.99999999999999705984 \cdots u$

With the construction below, a_n will depend only on $\widehat{\pi}_{n-1}$, and all the a_n's will be close to 1, so that this sequence will be ultimately periodical. Over one period, the ratio $\widehat{\pi}_n/\pi_n$ will be multiplied by some constant ρ, and since all roundings will be performed upward, $\rho > 1$. Over m periods, the ratio $\widehat{\pi}_n/\pi_n$ will be multiplied by ρ^m, so that the relative error will grow exponentially, thus will become larger than $(n-1) \cdot u$ when n is large enough.

We assume $p \geq 6$, so that it can be shown that the following construction behaves as wanted. At step $n \geq 2$, one can write:

$$\begin{aligned} a_n &= 1 + k_n \cdot 2^{-p+1}, \\ \widehat{\pi}_n &= 1 + g_n \cdot 2^{-p+1} = \mathrm{RN}(\widehat{\pi}_{n-1} \cdot a_n), \end{aligned}$$

where k_n will be an integer[5] and g_n will be a positive integer. We will deal with the initial step after giving the general rule. We have

$$\widehat{\pi}_{n-1} \cdot a_n = 1 + (g_{n-1} + k_n) \cdot 2^{-p+1} + g_{n-1}k_n \cdot 2^{-2p+2}.$$

We wish to maximize the relative error and have an upward rounding. If $g_{n-1} + k_n$ is less than 2^{p-1}, the number $1 + (g_{n-1} + k_n) \cdot 2^{-p+1}$ is a FP number. To maximize the relative error, we wish $g_{n-1} + k_n$ to be non-negative and as small as possible, while $g_{n-1}k_n \cdot 2^{-2p+2}$ should be as close as possible to, but larger than (for upward rounding), $\pm 2^{-p}$, i.e. $g_{n-1}k_n \simeq_> 2^{p-2}$; and we will get:

$$g_n = \begin{cases} g_{n-1} + k_n & \text{if } k_n < 0, \\ g_{n-1} + k_n + 1 & \text{if } k_n > 0. \end{cases}$$

However under these constraints, if g_n is very small, then one obtains large values for g_{n+1}, g_{n+2}, etc., which is not interesting as we want each g_i to remain small. For this reason, we will try to keep k_n and g_n balanced. Hence a good choice is

- $k_n = 1 + \left\lfloor \frac{2^{p-2}}{g_{n-1}} \right\rfloor$ if $g_{n-1} < \lfloor 2^{(p-1)/2} \rfloor$;
- $k_n = 1 - \left\lceil \frac{2^{p-2}}{g_{n-1}} \right\rceil$ otherwise.

[5] If k_n is negative, it could be a half-integer, but such a choice would not yield an interesting sequence.

Table 4 Relative errors for the smallest values of n such that the relative error is larger than $(n-1) \cdot u$ with the values a_i generated by our method of Section 6

p	n	relative error
6	106	$105.5728705 \cdots u$
7	124	$123.0487381 \cdots u$
8	119	$118.2293467 \cdots u$
9	156	$155.0673067 \cdots u$
24	27921	$27920.0002498 \cdots u$

For the initial step, as we want g_2 to be as small as possible, we will choose for k_1 $(= g_1)$ the smallest integer such that $k_2 < 0$, i.e.

$$k_1 = g_1 = \lfloor 2^{(p-1)/2} \rfloor.$$

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. Table 4 shows relative errors for the smallest values of n such that the relative error is larger than $(n-1) \cdot u$ with the values a_i generated by our method for various values of p.

7 Conclusion

We have shown that, under mild conditions (in particular, a reasonable bound on n), 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 when n is not too large. We have provided examples that show that the actual error can be very close to, but smaller than, $(n-1) \cdot u$ for small values of n, and becomes larger than $(n-1) \cdot u$ when n is large enough.

References

1. Graillat, S.: Accurate floating point product and exponentiation. IEEE Trans. Comput. **58**(7), 994–1000 (2009)
2. Higham, N.J. Accuracy and Stability of Numerical Algorithms, 2nd edn. SIAM, Philadelphia, PA (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. Jeannerod, C.-P., Rump, S.M.: On relative errors of floating-point operations: optimal bounds and applications. Research report hal-00934443, available at http://hal.inria.fr/hal-00934443
5. Jeannerod, C.-P., Rump, S.M.: Improved error bounds for inner products in floating-point arithmetic. SIAM J. Matrix Anal. Appl. **34**(2), 338–344 (2013)
6. Knuth, D.: The Art of Computer Programming, 3rd edition, volume 2, Seminumerical Algorithms. Addison-Wesley, Reading, MA (1998)
7. Muller, J.-M., Brisebarre, N., De Dinechin, F., Jeannerod, C.-P., Lefèvre, V., Melquiond, G., Revol, N., Stehlé, D., Torres, S.: Handbook of Floating-Point Arithmetic. Birkhäuser Boston (2010)
8. Rump, S.M.: Error estimation of floating-point summation and dot product. BIT **52**(1), 201–220 (2012)