Validation for scientific computations properties that can be established in floating-point arithmetic

Cours de recherche master informatique

Nathalie Revol

Nathalie.Revol@ens-lyon.fr

13 octobre 2006

IEEE-754 Floating-Point Arithmetic : Sterbenz lemma

Let a and b be two positive floating-point numbers. If

$$\frac{a}{2} \leq b \leq 2a$$

then $a - b = a \ominus b$.
In other words, $a - b$ is exactly representable in floating-point arithmetic.

IEEE-754 Floating-Point Arithmetic : another provable property

With rounding to nearest.

If we compute using FP arithmetic

$$z := \frac{x}{\sqrt{x^2 + y^2}},$$

can we prove that $-1 \leq z \leq 1$?

Can we compute $t = \sqrt{1 - z^2}$ without getting a NaN ?

IEEE-754 Floating-Point Arithmetic : another provable property

With rounding to nearest.

If we compute using FP arithmetic

$$z := \frac{x}{\sqrt{x^2 + y^2}},$$

can we prove that $-1 \leq z \leq 1$?

Can we compute $t = \sqrt{1 - z^2}$ without getting a NaN ?

The answer is **yes**.

Computing the roundoff error using FP operations addition and subtraction

Theorem :
for any pair (x, y) of FP numbers and for rounding to nearest, there exists FP numbers r_+ and r_- such that

$$\begin{aligned} r_+ &= (x+y) - (x \oplus y) \\ r_- &= (x-y) - (x \ominus y) \end{aligned}$$

Furthermore, r_+ and r_- can be computed using FP operations.

Computing the roundoff error : $+$ and $-$

Why with "rounding to nearest" only ?

Counterexample for directed rounding

with basis 2 and at least $p > 4$ bits of mantissa, let's take

$$\begin{aligned} x &= -(2^{2p} + 2^{p+1}) \\ y &= 2^p - 3 \end{aligned}$$

then we have

$$\begin{aligned} x + y &= -2^{2p} - 2^p - 3 \\ x \oplus y &= -2^{2p} \\ (x + y) - (x \oplus y) &= -2^p - 3 \qquad \text{not representable.} \end{aligned}$$

Computing roundoff errors using FP operations : $+$

Let x and y be two normal FP numbers such that $|x| \geq |y|$ and the rounding mode be to nearest. Let also assume that $x \oplus y$ does not overflow.

$$\text{Algo Fast2Sum :} \quad \begin{array}{rcl} s & = & x \oplus y \\ z & = & s \ominus x \\ r & = & y \ominus z \end{array}$$

The mathematical equality holds :

$$s + r = x + y$$

i.e. r is the roundoff error on the addition of x and y.

Beware of the optimizations done by your compiler. . .

Proof of Fast2Sum

$$\text{Algo Fast2Sum :} \quad \begin{aligned} s &= x \oplus y \\ z &= s \ominus x \\ r &= y \ominus z \end{aligned}$$

- if x and y have the same sign : then $x \leq x + y \leq 2x$ thus $x \leq s \leq 2x$ s since $2x$ is representable and since the rounding mode is monotonic. By Sterbenz lemma, z exactly equals $s - x$. Since $(x+y)-s$ is exactly representable, then $r = (x \oplus y) \ominus s = (x+y)-s$ and since $y - z = r$, then $b \ominus z = r$ exactly.

Proof of Fast2Sum

- if x and y have opposite signs :
  - either $|y| \geq \frac{1}{2}|x|$ and Sterbenz lemma applies : $x \oplus y$ is exact, i.e. $s = x + y$, $z = b$ and $r = 0$;
  - or $|y| < \frac{1}{2}|x|$ and thus $|x + y| > \frac{1}{2}|x|$.
    This implies that $|s| \geq \frac{1}{2}|x|$, since $\frac{1}{2}|x|$ is representable and rounding is monotonic, then Sterbenz implies that $z = s \ominus x = s - x$ exactly.
    Since $(x + y) - s$ is exactly representable, then $r = (x \oplus y) \ominus s = (x + y) - s$ and since $y - z = r$, then $b \ominus z = r$ exactly.

This algo is also correct under the weaker assumption that the exponent of $x \geq$ the exponent of y.

Computing roundoff errors using FP operations : $+$

To avoid comparing x and y (a comparison can be costly), let us use

$$\text{Algo TwoSum :}\quad \begin{array}{rcl} s & = & x \oplus y \\ y' & = & s - x \\ x' & = & s - y' \\ \delta_y & = & y - y' \\ \delta_x & = & x - x' \\ r & = & \delta_x + \delta_y \end{array}$$

The mathematical equality holds : $s + r = x + y$
i.e. r is the roundoff error on the addition of x and y.

Computing roundoff errors using FP operations : $\times$

x, y two normal FP nbs, rounding to nearest, $x \otimes y$ does not overflow.

Theorem : $r = (x \times y) - (x \otimes y)$ is representable.

Denote by $s = \lceil \frac{p}{2} \rceil$.

Algo TwoMult :
(aka Dekker)

$$
\begin{array}{rcl}
x' & = & x \otimes (2^s \oplus 1) \\
x_h & = & (x \ominus x') \oplus x' \\
x_l & = & x \ominus x_h \\
\multicolumn{3}{l}{\text{ibid. for } y} \\
r_h & = & x \otimes y \\
r_l & = & (((x_h \otimes y_h \ominus r_h) \\
 & & \qquad \oplus x_h \otimes y_l) \oplus x_l \otimes y_h) \\
 & & \qquad\qquad \oplus x_l \otimes y_l
\end{array}
$$

The mathematical equality holds :

$$r_h + r_l = x \times y$$

i.e. r_l is the roundoff error on the multiplication of x and y.

17 operations : 7 $\otimes$ and 10 $\pm$.

Computing roundoff errors using FP operations : $\times$

Let x and y be two normal FP numbers and the rounding mode be to nearest. Let also assume that $x \oplus y$ does not overflow.

If a *fma* is available : *fused multiply-add*, it computes the rounding of $(a \times b + c)$.

$$\text{Algo TwoMultFMA :} \quad \begin{array}{rcl} r_h & = & x \otimes y \\ r_l & = & fma(x, y, -r) \end{array}$$

The mathematical equality holds :

$$r_h + r_l = x \times y$$

i.e. r_l is the roundoff error on the multiplication of x and y.