When Double Rounding is Odd

Sylvie Boldo Guillaume Melquiond

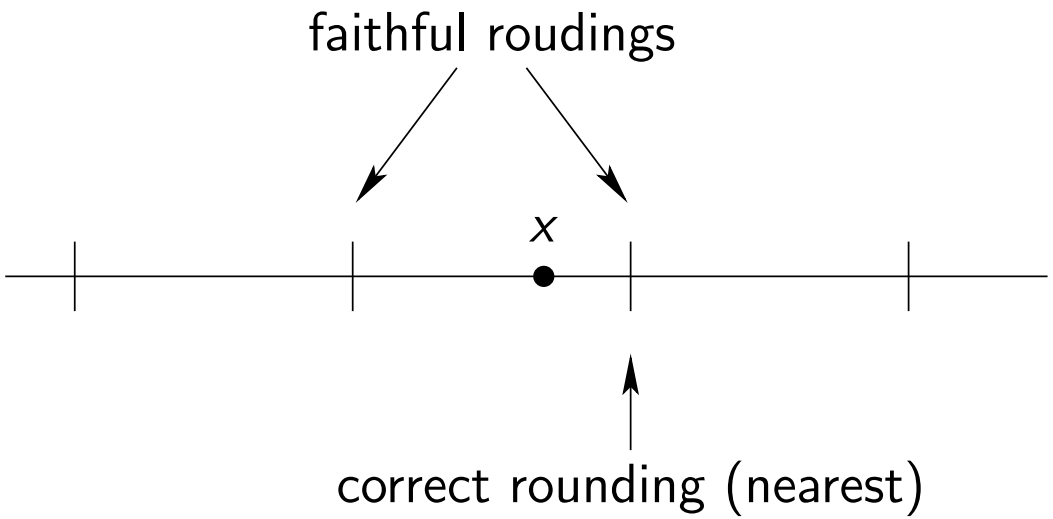
Arénaire, LIP, ENS Lyon

July 11th 2005

Rounding real numbers to floating-point numbers

Floating-point formats suffer from a limited precision: the result of an operation may not be exactly representable.

Faithful and correct roundings:



The real number will usually be rounded to the nearest number representable in the destination format.

Double rounding

On x86 architectures, when the three variables are double precision floating-point numbers, the code `c = a + b;` is compiled as:

```
fldl  b   ; load one operand
faddl a   ; load the other operand and
          ; add it in extended precision
fstpl c   ; store the result in double precision
```

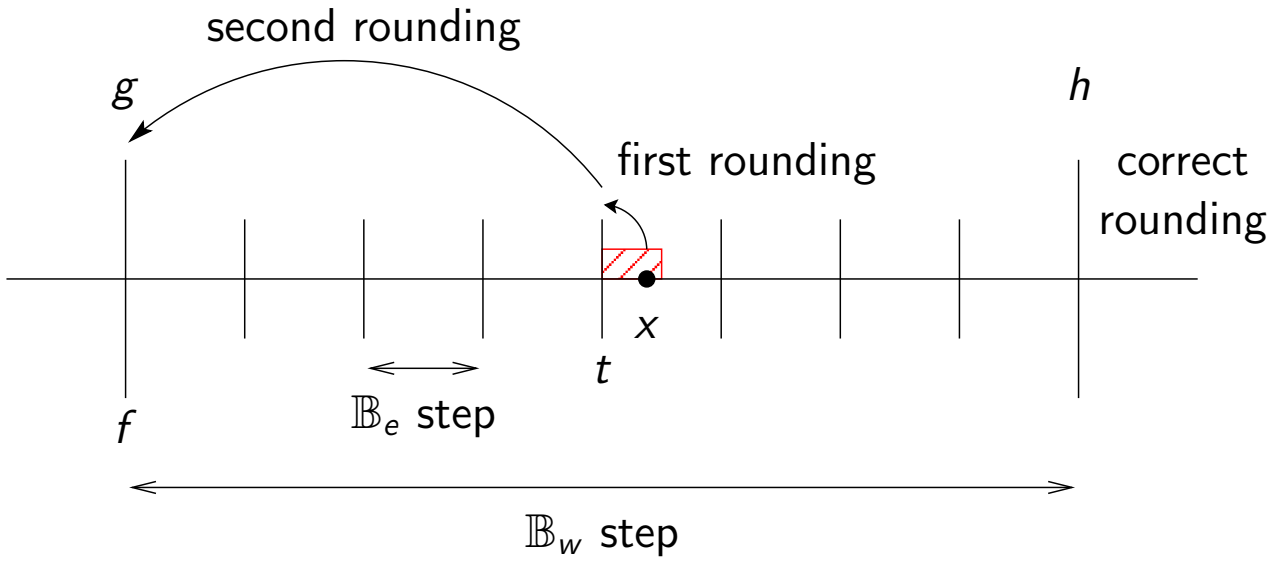
By default, two successive roundings will occur:

- first in extended precision during the addition,
- then in a lower precision when storing the value into memory.

In most cases, the final result is the correct rounding of the sum of the two floating-point numbers. But not always.

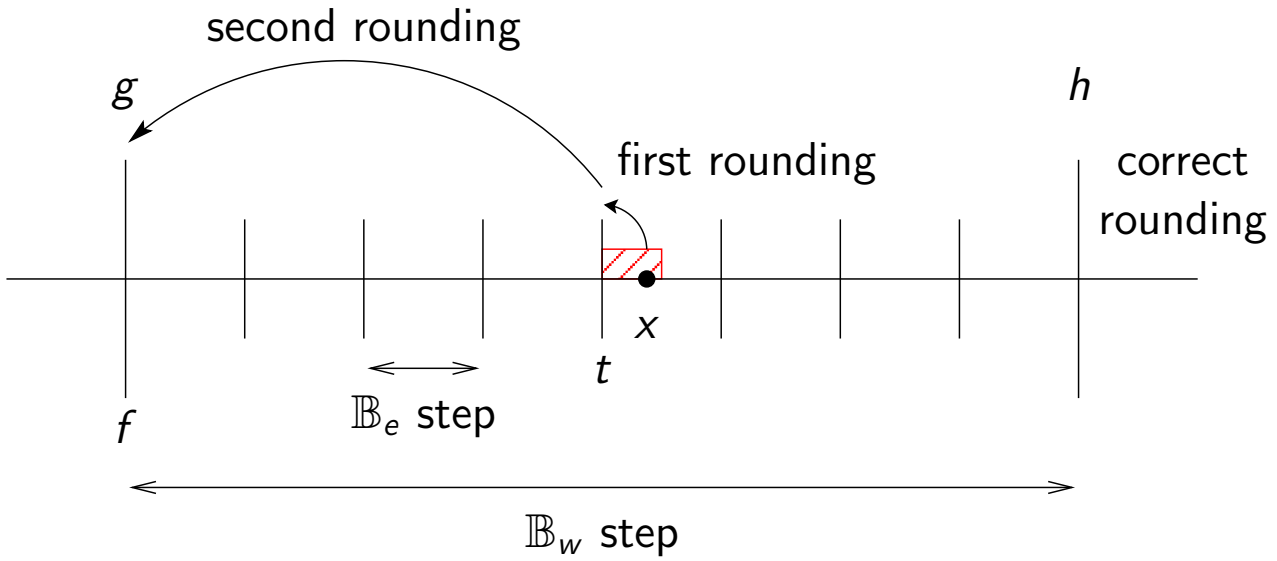
Double rounding going bad

If the first rounding to the format $\mathbb{B}_e$ gives the number t at equal distance from two consecutive floating-point numbers g and h in the second format $\mathbb{B}_w$, x could get rounded to the wrong end.



Double rounding going bad

If the first rounding to the format $\mathbb{B}_e$ gives the number t at equal distance from two consecutive floating-point numbers g and h in the second format $\mathbb{B}_w$, x could get rounded to the wrong end.



This situation would not occur if the first stage was never rounding x to the midpoint t (unless $x = t$).

The formalization of floating-point arithmetic

- Our formal proofs are based on
  - the formalization of M. Daumas, L. Rideau, and L. Théry,
  - the corresponding Coq library written by S. Boldo and L. Théry.
- A floating-point number is a pair (n, e) of relative integers. It is interpreted as the real number $n \cdot 2^e$.
- A number is representable in a floating-point format $\mathbb{B} = (p, E)$ if it satisfies $|n| \leq 2^p$ and $e \geq -E$.
- A rounding mode is defined as a relation between a real number and a floating-point number.

Rounding to odd

Rounding to odd, also called Von Neumann's rounding:

$$\square_{\text{odd}}(x) = \begin{cases} x & \text{if } x \text{ is representable} \\ \triangle(x) & \text{if the mantissa of } \triangle(x) \text{ is odd} \\ \triangledown(x) & \text{otherwise} \end{cases}$$

Rounding to odd

Rounding to odd, also called Von Neumann's rounding:

$$\square_{\text{odd}}(x) = \begin{cases} x & \text{if } x \text{ is representable} \\ \triangle(x) & \text{if the mantissa of } \triangle(x) \text{ is odd} \\ \triangledown(x) & \text{otherwise} \end{cases}$$

As a consequence, the mantissa of $\square_{\text{odd}}(x)$ is even only if x is a representable floating-point number: $x = \square_{\text{odd}}(x)$.

If the midpoint of two consecutive floating-point numbers is always even, rounding to odd would be a correct first stage of a double rounding.

Some basic properties formally proven

- Rounding to odd is a rounding mode:
  - each real number can be rounded to odd,
  - any odd rounding is a faithful rounding,
  - rounding to odd is monotone.

Some basic properties formally proven

- Rounding to odd is a rounding mode:
  - each real number can be rounded to odd,
  - any odd rounding is a faithful rounding,
  - rounding to odd is monotone.
- For each real, there is a unique odd rounding.
  So rounding to odd can be expressed as a function.
- Rounding to odd is symmetric:
  if $f = \square_{\mathrm{odd}}(x)$, then $-f = \square_{\mathrm{odd}}(-x)$.

The main property: double rounding

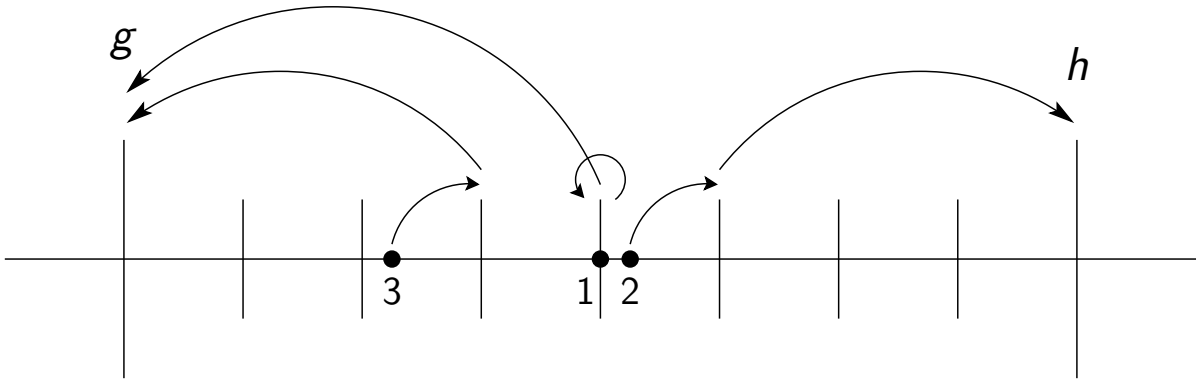
The formats are $\mathbb{B}_w = (p, E_w)$ and $\mathbb{B}_e = (p + k, E_e)$.

Assuming that $p \geq 2$, $k \geq 2$, and $E_e \geq 2 + E_w$, first rounding to odd in the extended format then rounding to nearest in the working format is equivalent to directly rounding to nearest in the working format.

$$\forall x \in \mathbb{R}, \quad \circ^w(x) = \circ^w\left(\square^e_{\mathrm{odd}}(x)\right)$$

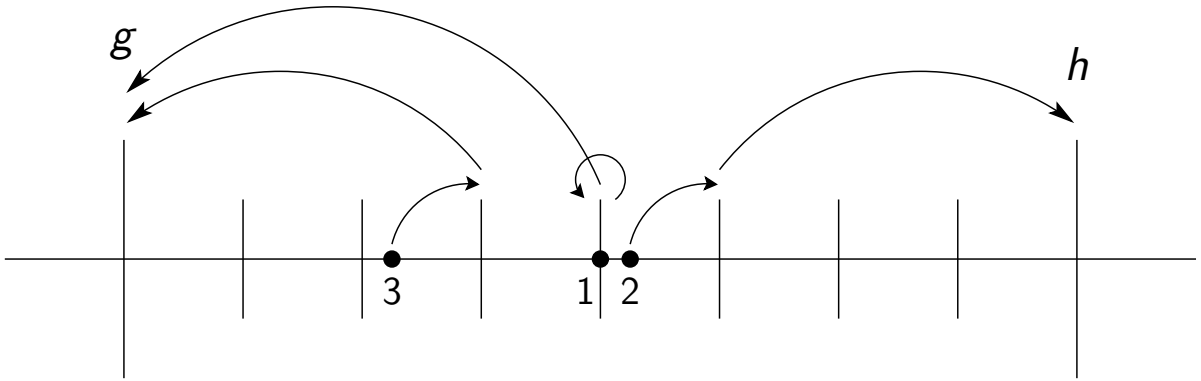
Some details about the proof

$k \geq 2$ ensures that the midpoint of two consecutive representable numbers in $\mathbb{B}_w$ has an even mantissa when represented in $\mathbb{B}_e$.



Some details about the proof

$k \geq 2$ ensures that the midpoint of two consecutive representable numbers in $\mathbb{B}_w$ has an even mantissa when represented in $\mathbb{B}_e$.



The double-rounding property was formally proved in Coq. The general case was easy to prove. Some special care had to be taken when $\circ^w(x)$ is a power of two though.

How to round to odd?

- Rounding to odd is not harder than rounding to zero.
  - The real number x is the theoretical result of an infinitely-precise arithmetic operation.
  - First round this result x toward zero.
  - Then set the lowest bit of the mantissa of $\mathcal{Z}(x)$ to one if the inexact flag was set during the rounding toward zero.
  - This computed number is $\square_{\mathrm{odd}}(x)$. The current inexact flag is still valid for the whole rounding to odd.

How to round to odd?

- Rounding to odd is not harder than rounding to zero.
  - The real number x is the theoretical result of an infinitely-precise arithmetic operation.
  - First round this result x toward zero.
  - Then set the lowest bit of the mantissa of $\mathcal{Z}(x)$ to one if the inexact flag was set during the rounding toward zero.
  - This computed number is $\square_{\mathrm{odd}}(x)$. The current inexact flag is still valid for the whole rounding to odd.
- Both rounding to zero and inexact flag are features from the IEEE-754 standard. An already compliant implementation of this standard would therefore have no difficulty to provide floating-point operations rounded to odd.

One computation, many formats

Thanks to the double-rounding property, storing a numerical value by rounding it to odd in precision $p + 2$ makes it available as a correctly rounded value to any precision $p' \leq p$.

One computation, many formats

Thanks to the double-rounding property, storing a numerical value by rounding it to odd in precision $p + 2$ makes it available as a correctly rounded value to any precision $p' \leq p$.

Corollary: if a correctly rounded number in precision $p + 2$ has an odd mantissa, it can be rounded again to a precision $p' \leq p$ and still be correctly rounded in this lower precision.

For example, the correctly rounded π constant on 80 bits can later be rounded to nearest to get the 32 and 64 bits version.

Correctly rounded summation of floating-point numbers

Let $(f_i)_{i \leq n}$ be a set of floating-point numbers. We want to compute their correctly rounded sum:

$$s = \circ^p \left(\sum_{i=1}^{n} f_i \right).$$

Let us compute $g_1 = f_1$ and $\forall i < n, \quad g_{i+1} = \square^{p+k}_{\text{odd}}(g_i + f_{i+1})$.

If $\forall i, |f_{i+1}| \geq 2|g_i|$, then $s = \circ^p(g_n)$.

Correctly rounded summation of floating-point numbers

Let $(f_i)_{i \leq n}$ be a set of floating-point numbers. We want to compute their correctly rounded sum:

$$s = \circ^p \left(\sum_{i=1}^{n} f_i \right).$$

Let us compute $g_1 = f_1$ and $\forall i < n, \quad g_{i+1} = \square^{p+k}_{\text{odd}}(g_i + f_{i+1})$.

If $\forall i, |f_{i+1}| \geq 2|g_i|$, then $s = \circ^p(g_n)$.

The property $\forall i, |f_{i+1}| \geq 2|g_i|$ may be hard to ensure. It is possible to avoid it, if the f_i are sorted by magnitude and they verify $\forall i, |f_{i+1}| \geq 3|f_i|$.

Conclusion

- Our Coq formalization is about one thousand lines of Coq. It defines the rounding to odd, and it proves the double-rounding property.

Conclusion

- Our Coq formalization is about one thousand lines of Coq. It defines the rounding to odd, and it proves the double-rounding property.
- The formal proofs ensure that the property is true even for the corner cases. In particular, it gives strict requirement on the minimal exponents of both formats so that the property is true even on subnormal numbers.

Conclusion

- Our Coq formalization is about one thousand lines of Coq. It defines the rounding to odd, and it proves the double-rounding property.
- The formal proofs ensure that the property is true even for the corner cases. In particular, it gives strict requirement on the minimal exponents of both formats so that the property is true even on subnormal numbers.
- This rounding mode and an extended precision allow to execute several floating-point operations and yet to get a correctly rounded result. It could even be used to emulate a fused-multiply-add operator without an extended precision.

Questions?

The Coq formalization:

- `http://lipforge.ens-lyon.fr/www/pff/`

E-mail addresses:

- `sylvie.boldo@ens-lyon.fr`
- `guillaume.melquiond@ens-lyon.fr`