Properties of the subtraction valid for any floating point system

Sylvie Boldo and Marc Daumas

Laboratoire de l’Informatique du Parallélisme
UMR 5668 - CNRS - ENS de Lyon - INRIA

1 Introduction

In 1974, Sterbenz [20] presented a theorem about the exact subtraction of two floating point numbers \( x \) and \( y \) when they are very close one from another, that is

\[
\frac{y}{2} \leq x \leq 2y.
\]

The theorem stating that \( x - y \) is exact under the preceding condition was presented for any radix provided the hardware was accurate enough. More recently, other authors [8,10] presented similar results with an emphasis on didactic aspects.

We have recognized in [6] that Sterbenz’s theorem is not a property of the computing hardware but rather a property of the floating point number representation. Given \( x \) and \( y \), the question is to know whether or not \( x - y \) can be represented in the working floating point system. This is clearly the key necessary condition for the implemented floating point subtraction \( x \ominus y \) to return the exact result \( x - y \).

With IEEE-like behavior, any floating point operation is cut down to two steps. An intermediate result is first computed to sufficient accuracy and then rounded. The designer must guarantee that the system always returns the result as if the infinitely precise mathematical operation were rounded. For example the subtraction is implemented as the composition of two mathematical functions, namely, the subtraction \((-\)) and the user specified rounding function \((\circ)\)

\[
x \ominus y = \circ(x - y).
\]

The details of the implementation are not relevant to the user since knowing the rounding function is sufficient to deduce the value returned by any operation. Users usually expect the rounding function to be a monotonous (non decreasing) projection of the real numbers over the set of the machine floating point numbers. The later property implies that for any floating point number \( v \),

\[
\circ(v) = v.
\]

©2002 Published by Elsevier Science B. V.
Establishing Sterbenz’s equality does not require any additional knowledge on
the rounding function provided it is a projection.

In Section 2, we describe more precisely our formalization of the floating-
point system. In Section 3 we discuss key properties of this system. The
proposed representation is very redundant but we will see that any machine
number has one single canonical representation that can be used in hardware.
We will quickly present that a floating point system must handle denormal
numbers in order to verify Sterbenz’s theorem on very small numbers. Finally
as the existence of a negation will become key to Sterbenz’s theorem, we will
present a strong necessary and sufficient condition for a generic number system
to be stable by negation.

Section 4 presents our results about Sterbenz’s theorem and relates them
to real hardware implementations. We have focused on IEEE 754 standard
implementations and on Texas Instrument TMS 320C3x series described in
Section 5 with its SMJ military grade processes. The SMJ 320C3x circuits can
be used in avionics and military applications such as the flight control primary
or secondary computer (FCPC / FCSC) [14]. Past studies have proved that
an automatic proof checker must be used for such critical systems [16]. This
work ends with concluding remarks and perspectives for further developments.

2 Our formalization of a floating point system

All the results have been developed and validated using Coq [11]. It is a
theorem checking system based on the Curry-Howard isomorphism. Systems
like Coq allow the user to define new objects and to derive consequences of
these definitions formally while checking every detail. The Coq tool is based
on higher-order logic. With such an expressive logic, it is possible to state
properties in their most general form. For example, universal quantification
has been used to state properties that are true for an arbitrary rounding mode.
Theorem provers have already been successfully used to mechanically check
the correctness of floating-point algorithms [17,9], and with a strong emphasis
for avionics [2].

We will present in this text the behavior of a generic floating point system
in regard to Sterbenz’s theorem. We define a generic floating point system
from a mapping of $\mathbb{Z}^2$ onto $\mathbb{R}$

$$(n, e) \mapsto n\beta^e$$

where $\beta$ is a constant integer strictly greater than one called the radix of the
floating point system. Later $n$ will be called the mantissa and $e$ the amplitude.

In Coq, the set is defined by the `float` type defined below in ASCII

Record float : Set := Float {
  Fnum: Z;
  Fexp: Z }.
and its value is obtained by using the $FtoR$ function

\[
\text{Definition } FtoR := \{x : \text{float} \mapsto \text{Rmult} (Fnum(x) \text{ (powerRZ (IZR \ radial) (Fexp(x))))}.
\]

Two pairs are equivalent if they are mapped to the same real value. This equality will be noted as $=_{\mathbb{R}}$. Coq files are hardly understandable for a non-Coq user, theorems and definitions can be presented using an integrated pretty printer. For example:

**Definition 2.1** $FtoR := x : \text{float} \mapsto \text{Fnum}(x) \times \beta^{\text{Fexp}(x)}$

All the quantities treated by a computer system must fit into a finite field, we focus our interest on pairs $(n, e)$ such that $n$ and $e$ are bounded. For practical reasons, we do not use an upper bound on the amplitude and a bounded floating point pair is such that

\[
n \in \{-N_i, \cdots, N_s\} \quad \text{and} \quad e \geq -E_i.
\]

That is to say in Coq:

**Definition 2.2** $F_{\text{bounded}} := b : F_{\text{bound}}, d : \text{float} \mapsto \neg \text{vNumInf}(b) \leq \text{Fnum}(d) \wedge (\text{Fnum}(d) \leq \text{vNumSup}(b)) \wedge (-\text{dExp}(b) \leq \text{Fexp}(d))$

A sectioning mechanism with implicit parameter management transforms Sterbenz’s theorem with our floating point library so that it states that

\text{“for any radix greater than one, for any floating-point system, for all floats } x \text{ and } y, \text{ if } x \text{ and } y \text{ are bounded, and if } \frac{y}{2} \leq x \text{ and } x \leq 2 \times y \text{ then there exists a bounded float } z \text{ such that } z =_{\mathbb{R}} x - y”.\]

Unfortunately, this assertion is false. For example, let the radix be two and the format such that the mantissa is between $-1100_2$ and $1111_2$. Let $x$ be $(1111_2, 0)$ and $y$ be $(1110_2, 1)$. Both $x$ and $y$ are bounded, they are such that $\frac{y}{2} \leq x \leq 2 \times y$ but $x - y$ is $-1101_2$ and this value cannot be represented exactly in this floating point system. We will later give a list of necessary conditions for the assertion to be true.

Proofs are built interactively using high-level tactics that may solve some of the “easy” subgoals. We used pcoq [1]: a working environment for the Coq theorem prover with a nice graphical interface and the pretty printer.

At the end of each proof, Coq records a proof object that contains all the details of the derivation and ensures that the theorem is valid. The object can be double checked for life critical applications by a tool such as BindLib, a program independent of the Coq development.

The proofs for this work can be downloaded through the Internet at the address

They include the current development of the floating point library available at

http://www-sop.inria.fr/lemme/AOC/.

All the following theorems have been proved using this very general formalization. Unless explicitly specified the properties hold for any radix greater than one and any bound on the mantissa and the amplitude.

3 Basic properties of the set of bounded floating point numbers

3.1 Multiple representations

Contrary to IEEE-like behavior, the proposed library defines possibly many bounded floating point pairs with the same value. For example, the three radix two floating point pairs \((1100_2, 4)\), \((110_2, 5)\) and \((11_2, 6)\) share the same real value \(3 \times 2^6 = 192\). This fact can be disturbing as one real value can be associated to many different bounded floating point pairs that do not have the same properties.

In order to retain common floating point behavior, we define a canonical pair for each bounded pair. This pair is meant to represent the actual fields stored in a computer that are associated to the number. A pair is normal if it is bounded and its amplitude cannot be reduced by multiplying the mantissa by the radix, that is

\[ n \times \beta \notin \{-N_i, \cdots, N_s\}. \]

A pair is denormal if it is bounded and the amplitude reduction is blocked by the fact that it uses already the minimal accepted amplitude despite the mantissa being small enough to be multiplied by the radix. That is

\[ n \times \beta \in \{-N_i, \cdots, N_s\} \quad \text{and} \quad e = -E_i. \]

Any bounded pair is equivalent to one unique pair either normal or denormal. The later pair is called the canonical representation. This fact is proved by several theorems. The first one, \(F\text{canonicIUnique}\) states that if \(p\) and \(q\) are two canonical floating-point numbers such that \(p =_R q\) then \(p\) and \(q\) are syntactically equal (Leibniz’s equality).

Other theorems prove the correctness of the \(F\text{normalizeI}\) function defined below to construct the canonical representation from any bounded representation:

Fixpoint FNIAux [v, N, q : nat] : nat := Cases q of
0 => 0
| (S q’) => Cases
  (Zcompare (Zmult (Zpower_nat radix q’) v) (Zmult radix N)) of
    INFERIEUR => q’ | EGAL => q’ | _ => (FNIAux v N q’) end
end.

Definition FNI := [q, N : nat] (pred (FNIAux q N (S (S N)))).

Definition FnormalizeI :=
  [b : FboundI] [p : float]
  Cases (Zcompare ZERO (Fnum p)) of
  EGAL => (Float ZERO (Zopp (dExp b)))
  | INFERIEUR => (Fshift radix (min
    (FNI (absolu (Fnum p)) (vNumSup b))
    (absolu (Zplus (Fexp p) (dExp b)))) p)
  | SUPERIEUR => (Fshift radix (min
    (FNI (absolu (Fnum p)) (vNumI nf b))
    (absolu (Zplus (Fexp p) (dExp b)))) p)
end.

Expressing that the function is correct means that (i) if \( p \) is a bounded float, then the result \( FnormalizeI(p) \) is a bounded float (\( FnormalizeIBounded \)). It also means that (ii) the result is canonical (\( FnormalizeIFcanonicI \)) and such that (iii) the input pair \( p \) and the result pair are mapped to the same real value that is to say \( p =_F FnormalizeI(p) \) (\( FnormalizeICorrect \)). We omit these proofs as they are quite cumbersome but not difficult.

3.2 Negating a number

On IEEE-like number systems, the mantissa is stored with separate sign and magnitude, therefore \( N_i = N_s \). This fact is not true on all floating point systems. Some hardware designers decided to use two’s complement to store the mantissa as this is the case for Texas Instrument TMS 320C3x [21].

A bounded floating point number \( p \) can be negated if there exists another bounded floating point number \( q \) such that \( q =_F -p \). As we will see, almost any number can be negated even on systems based on the TMS 320C3x digital signal processors. The only two cases where a number cannot be negated cause either an overflow as the opposite of the least represented number is larger than the biggest number allowed in the number system or an underflow as the opposite of the least represented positive normal number is larger than the biggest negative number allowed in the number system. The second case would not occur on systems that handle denormal numbers.

The following theorem checked with Coq (\( FoppBounded \) and \( FoppBoundedInv \)) answers any question about negating a number. The cases study for a system that does not handle denormal numbers and for the upper bound on the amplitude are treated separately (\( FoppBoundedExp \)).

Theorem 3.1 On a floating point system bounded by \( N_i, N_s \) and \( E_i \) with
$N_i \neq N_s$, any bounded pair can be negated to a bounded float if and only if

$$|N_i - N_s| = 1 \quad \text{and} \quad \beta \mid \max(N_i, N_s).$$

Without loss of generality, we assume that $N_s > N_i$. As a consequence, any pair $(n, e)$ with $n \in \{-N_i, \cdots, N_i\}$ can be easily negated by negating its mantissa. The pairs $(n, e)$ with $n \in \{N_i + 1, \cdots, N_s\}$ can only be negated by manipulating the amplitude. Therefore, $\beta$ should divide all the $n \in \{N_i + 1, \cdots, N_s\}$. That is possible only for $N_i + 1 = N_s$ if $\beta$ divides $N_s$. On the contrary, if $N_s$ is a multiple of $\beta$ and $N_i = N_s - 1$, any bounded pair can be negated to find another bounded pair.

We have also proved that the negation is the only opposite on a system that handles denormal numbers: if $x \oplus y = 0$, then $y$ is the negation of $x$. Rephrasing [13] we first prove that the distance between two floating point numbers is at least $\beta^{-E_i}$, then we conclude in the OppositeUnique:

**Theorem 3.2** On a floating point system bounded by $N_i$, $N_s$ and $E_i$, let $P$ be any rounding mode and $x$ and $y$ be two bounded floats. If $y \neq_R -x$ and $z$ is a rounded result of $x + y$ then $|z| \geq \beta^{-E_i}$.

### 3.3 Usual definitions of radix complement

Radix complement may or may not be used depending on the convention for negative numbers defined by the author for radix $\beta > 2$. The three common definitions are equivalent when $\beta = 2$.

The interpretation where the sign digit is $-1$ when $\beta - 1$ is stored in the most significant digit leads to the bounds

$$N_i = \beta^{p-1} \quad \text{and} \quad N_s = (\beta - 1) \cdot \beta^{p-1} - 1$$

with $p$ bits of mantissa ($p > 1$). If $\beta > 2$, $N_s - N_i > 1$ and some bounded pairs cannot be negated without rounding.

Some authors use the previous convention but restrict the leading digit to 0 or $\beta - 1$. In this case,

$$N_i = \beta^{p-1} \quad \text{and} \quad N_s = \beta^{p-1} - 1,$$

so any pair can be negated.

When the interpretation is read modulo $\beta^p$ and the digits are balanced evenly with possibly an additional digit to the negative set, the bound are

$$N_i = \left\lfloor \frac{\beta^p}{2} \right\rfloor \quad \text{and} \quad N_s = \left\lceil \frac{\beta^p}{2} \right\rceil - 1.$$

If $\beta$ is odd, the set is evenly balanced and $N_i = N_s$. If $\beta$ is even, $N_i = N_s + 1$ and $\beta$ divided $N_i$ since $p > 1$. 

137
As a conclusion, it seems natural to prefer a sign-magnitude or a two’s complement notation for the mantissa. We will see in Section 5 that all the existing implementations use one of these two classes.

3.4 Denormal numbers

The number system that we have just defined handles denormal pairs (gradual underflow) as this helps write more robust codes [7]. Sterbenz’s theorem cannot be true if denormal numbers are not allowed. Let $\lambda$ be the lowest positive normal number. Its value is

$$\lambda = \left(\left\lfloor \frac{N_s}{\beta} \right\rfloor + 1\right) \times \beta^{-E_i}$$

and the following floating point number is

$$\lambda^+ = \left(\left\lfloor \frac{N_s}{\beta} \right\rfloor + 2\right) \times \beta^{-E_i}.$$ 

The quantities $\lambda$ and $\lambda^+$ verify $\lambda^+/2 \leq \lambda \leq 2\lambda^+$ and

$$\lambda^+ - \lambda = \beta^{-E_i}$$

that is a denormal number. This example shows that without allowing denormal numbers, the subtraction of $x$ and $y$ under the conditions of Sterbenz’s theorem may not be represented.

3.5 Lexicographic order

Many authors, including [3], have recognized that it is a nice feature for lexicographic order of the floating point pairs to coincide with the order of the represented real values. As this fact is not necessary trivial in a generic floating point system, we establish the two following LexicoPosCanI and LexicoCanI theorems.

**Theorem 3.3** On a floating point system bounded by $N_i$, $N_s$ and $E_i$, for any canonical pair $(n_x, e_x)$ representing $x$ and any bounded pair $(n_y, e_y)$ representing $y$

$$0 \leq x \leq y \quad \text{implies} \quad e_x \leq e_y.$$ 

**Theorem 3.4** On a floating point system bounded by $N_i$, $N_s$ and $E_i$ with $|N_i - N_s| \leq 1$, for any canonical pair $(n_x, e_x)$ representing $x$ and any bounded pair $(n_y, e_y)$ representing $y$

$$|x| < |y| \quad \text{implies} \quad e_x \leq e_y.$$ 

The difference between the preceding theorems and the usual IEEE like situation arises from the fact that the magnitude of a floating point pair may
not be represented or may use another amplitude. We establish the following corollary.

**Corollary 3.5** On a floating point system bounded by \( N_i, N_s \) and \( E_i \) with \( |N_i - N_s| \leq 1 \), for any canonical pair \((n_x, e_x)\) representing \( x \) and any bounded pair \((n_y, e_y)\) representing \( y \)

\[
e_x < e_y \quad \text{implies} \quad |x| \leq |y|.
\]

This means that when \( |N_i - N_s| \leq 1 \), our floating point system behaves like a IEEE compliant implementation as far as lexicographical order is concerned.

When \( |N_i - N_s| > 1 \), we have a very different behavior. Here is an example that also shows that the bound on the difference \( N_i - N_s \) is tight. We define a binary notation with the mantissa between \(-1001_2\) and \(111_2\). The pairs \((100_2, 1)_2\) and \((-1001_2, 0)_2\) are canonical yet their magnitudes and their amplitudes are not in the same order. This cannot happen in IEEE compliant systems or on the TMS 320C3x.

4 **Sterbenz’s theorem**

4.1 **A first very general theorem**

It is amazing to realize that the following theorem is true whatever the radix and the bounds \( N_i \) and \( N_s \). Moreover, the proof has been upgraded automatically by the Coq proof checker from the previous proof \text{SterbenzAux} \text{presented in} [6] \text{that was supposed to work only when} \( N_i = N_s \).

**Theorem 4.1** On a floating point system bounded by \( N_i, N_s \) and \( E_i \) with no assumption of a relation between \( N_i \) and \( N_s \), for any bounded pairs \((n_x, e_x)\) and \((n_y, e_y)\) representing \( x \) and \( y \) such that

\[
y \leq x \leq 2y,
\]

the difference \( x - y \) can be represented by a bounded pair \((n, e)\). Furthermore, the bounded mantissa \( n \) and the bounded amplitude \( e \) can be defined as

\[
n = n_x \beta^{e_x - \min(e_x, e_y)} - n_y \beta^{e_y - \min(e_x, e_y)}
\]

\[
e = \min(e_x, e_y).
\]

On a floating point system where any bounded pair can be negated without rounding such as presented section 3.2, Sterbenz theorem \text{SterbenzOppI} stated below is proved by applying twice Theorem 4.1.

**Theorem 4.2** On a floating point system bounded by \( N_i, N_s \) and \( E_i \) where any bounded pair can be negated to another bounded pair, for any bounded pairs \( x \) and \( y \) such that

\[
y \leq x \leq 2y,
\]
the difference $x - y$ can be represented by a bounded pair.

We prove the theorem correct when $y/2 \leq x \leq y$ by applying Theorem 4.1 to $X = y$ and $Y = x$ so that $X - Y = -(y - x)$ can be represented by a bounded pair.

4.2 Other systems

Although we presented in section 3 that it is most desirable to use a number system with a few natural properties including the fact that every bounded pair can be negated without rounding, we present now the SterbenzI very generic theorem. The details of the proof are available on the Internet.

**Theorem 4.3** On a floating point system bounded by $N_i$, $N_s$ and $E_i$ where $|N_i - N_s| \leq \delta$, for any canonical pair $(nx, ex)$ representing $x$ and any bounded pair $(ny, ey)$ representing $y$ such that

$$y + \delta \beta^{\min(ex, ey)} \leq 2 \delta y,$$

the difference $x - y$ can be represented by a bounded pair.

5 Concluding remarks

5.1 Overview of existing hardware implementing sign magnitude

Most general purpose widely available processors use a sign magnitude representation. Some books [10,15] even present the sign magnitude notation as the natural floating point notation. This notation is in use in the well-studied IEEE-754 compliant hardware. Some IBM systems use radix 10 [4] and a few of them retain a radix 16 compatibility mode [19]. Yet most systems use radix 2. For all these systems, Sterbenz’s equality holds with all the natural properties presented in this work.

The properties presented here were scattered in the litterature as most of them have been published over the time. Yet the main motivations of this work was to compare the most common general purpose IEEE 754 based behavior to other implementations such as IEEE 854 compatible circuits, almost IEEE 854 behavior and non IEEE behavior as we see in the following.

5.2 Texas Instrument’s two’s complement notation

Texas Instrument uses in its TMS 320C3x the two’s complement notation for the mantissa. This notation was also in use in Honeywell 6080N computer [18]. A different notation with the same mantissa range is studied in a exercise of [12]. This number system is well suited as all the natural properties (stability through negation, existence and unicity of an opposite and lexicographic order of the pairs) are still true and Sterbenz’s equality holds. We do not have any
knowledge of a working floating point unit that uses neither sign-magnitude nor the two’s complement notation for the mantissa encoding.

The following theorem (ReductRange and ReductRangeInv) can be used to deduce that the set of the represented numbers is almost identical with an IEEE-compatible unit and the TMS 320C3x. Should Texas Instrument decide to implement denormal pairs and precise rounding the unit could be functionnaly IEEE-compliant.

**Theorem 5.1** The set of the real numbers represented on a floating point system bounded by $N_i > 1$, $N_s > 1$ and $E_i$ is identical to the set of the numbers represented on a system bounded by $N_i$, $N_s - 1$ and $E_i$ (respectively $N_i - 1$, $N_s$ and $E_i$) if and only if

\[ \beta | N_s \quad (\text{respectively} \quad \beta | N_i). \]

All the theoretical results presented in this text prove that under a few assumptions there exists a bounded float that is the exact result of the subtraction. We have to look at the way the addition/subtraction is performed by the TMS 320C3x to be sure that this exact result is really returned by the unit.

Figure 1 presents a simplified version of the flowchart of the addition. This operation is first performed on extended precision and then rounded. The
mantissa of one of the inputs is possibly shifted depending on the actual value of the difference of the amplitudes. The result of the addition of the mantissas lies in 30 bits. That means that 8 additional bits are used for the intermediate result. Adding a new case to the result of [10], we see that one guard bit is sufficient for Sterbenz’s theorem to hold even using a different notation than the IEEE-like sign-magnitude for the mantissa. On the contrary, the Sterbenz’s theorem does not hold if the user manipulates extended numbers rather than single precision numbers. In this case, the operation is performed without any guard bit and the result is not necessarily found by the floating point unit.

If no exception is triggered, the mantissa is accurate enough to hold the exact result and the result before rounding is the expected exact result. As this result was proved to be bounded, the rounding does not change it and the final result is exact.

We deduce immediately from \( \frac{y}{2} \leq x \leq 2 \times y \) that \( x - y \in [-y/2; y] \) so no overflow can occur. If the exact result is a denormal number, the TMS 320C3x returns 0 as this processor does not handles such numbers.

In this text, we have shown that the floating point number system used for the TMS 320C3x defines almost the same real values as the system of an IEEE-compliant processor with a very different interpretation for the mantissa field. We have also shown that gradual underflow and correct rounding would be very sensible in such a system although neither was implemented. Finally, we have proved some very useful result about the TMS 320C3x such as Sterbenz’s theorem provided no underflow occurred.

5.3 On automatic proof checking

Without a strong incentive on formal analysis of the TMS 320C3x, such work would probably not have been carried out. It has been made possible by the very formal and generic development of the proofs used in Coq. Odds are that such conclusions would scarcely be trusted if they were not checked by an automatic proof checker since the proofs are very technical and prone to many small mistakes that would not have been ruled out by experimental knowledge.

We will continue to investigate natural properties of floating point number systems as they lead us to necessary conditions on the number systems. In the case of this work, Sterbenz’s equality and the possibility to negate a number are also key to analyze numerical software behavior such as [5].

6 Acknowledgments

We wish to thank Laurence Rideau and Laurent Thry for they work on the initial development of the Coq theory library.
References

URL http://www-sop.inria.fr/lemme/Laurence.Rideau/proof-pcoq.ps.gz

URL http://shemesh.larc.nasa.gov/fm/ftp/larc/vac/hug95.ps


URL http://computer.org/proceedings/arith/


URL http://link.springer.de/link/service/series/0558/bibs/2152/21520169.htm


URL http://www.cl.cam.ac.uk/users/jrh/papers/tang.ps.gz

URL http://www.ma.man.ac.uk/~higham/asna.html


