Number Theory and Floating-Point Numbers

(Applications de la théorie des nombres aux ordinateurs)

Paul Zimmermann

INRIA Lorraine/LORIA, Nancy, France

Rencontre *Théorie des nombres et applications*, CIRM, Luminy, 14-18 janvier 2002

What is a floating-point (fp) number?

$$x = m \cdot 2^e$$

m is the *mantissa*: $2^{p-1} \leq m < 2^p$

p is the *precision*

e is the *exponent*: $E_{\min} \leq e \leq E_{\max}$

special numbers: $x = \pm 0$, $x = \pm\infty$, $x = \mathrm{NaN}$

The IEEE 754 standard

Discussions started in 1977, standard was approved in 1985.

Cf *An Interview with the Old Man of Floating-Point*, interview from William Kahan (Turing Award 1989) by Charles Severance, `http://cch.loria.fr/documentation/IEEE754/`

Standardize the representation of fp numbers, operations and their semantics, exceptions (invalid operation, division by zero, overflow, underflow, inexact result).

Under revision: `http://www.validlab/com/754R/`

The IEEE 754 standard: Representation

format	total bits	mantissa	exponent	$E_{\min}, E_{\max}$
single precision	32	23+1	8	$-126, 127$
double precision	64	52+1	11	$-1022, 1023$
double extended	80	64	15	$-16382, 16383$
quadruple precision	128	112+1	15	$-16382, 16383$

Implicit leading bit: for "normal" numbers, the leading bit of the mantissa is always 1, and is not stored.

Special values ($\pm 0, \pm\infty, \mathrm{NaN}$, subnormal numbers) correspond to special exponent values.

The IEEE 754 standard: semantics

Let x a fp number: $f(x)$ is not always a fp number.

Four rounding modes: to nearest, towards $-\infty$ (down), towards $+\infty$ (up), towards 0 (truncation). "Even rounding rule" for nearest mode.

For all four basic arithmetic operations $(+, -, \times, \div)$, and the square root $(\sqrt{\ })$, the result of $x \diamond y$ should be the fp number nearest from the exact result for the given rounding mode.

Proposed extension in 754R to *fused multiply-add* (fma): $x \cdot y + z$.

Conversion between binary and decimal

Should be correctly rounded up to $\overbrace{999999999}^{9} \cdot 10^{\pm 13}$ in single precision, $\underbrace{99999999999999999}_{17} \cdot 10^{\pm 27}$ in double precision.

Otherwise the additional error should not exceed 0.47 *unit in last place* (ulp) in the round to nearest mode, 1.47 ulps in the directed rounding modes.

For nearest mode, binary $\circ$ decimal $=$ Id for the following bounds:

format	decimal to binary	binary to decimal
single	$\underbrace{999...999}_{9} \cdot 10^{\pm 99}$	$\underbrace{999...999}_{9} \cdot 10^{\pm 53}$
double	$\underbrace{999...999}_{17} \cdot 10^{\pm 999}$	$\underbrace{999...999}_{17} \cdot 10^{\pm 340}$

Density of Invertible Numbers

Jean-Michel Muller, *Some algebraic properties of floating-point arithmetic*, Third “Real Numbers and Computers”, April 2000.

We say that a fp number x of precision k admits a fp-reciprocal if there exists a fp number y of precision k such that:

$$\mathcal{N}(xy) = 1.$$

y is not unique: for $k = 5$, consider $x = \frac{3}{2}$, $y_1 = \frac{21}{32}$ and $y_2 = \frac{11}{16}$:

$$\mathcal{N}(xy_1) = \mathcal{N}(\frac{63}{64}) = 1 = \mathcal{N}(\frac{33}{32}) = \mathcal{N}(xy_2).$$

Does not always exist: for $k = 6$, consider $x = \frac{29}{16}$, $y_1 = \frac{35}{64}$, $y_2 = \frac{9}{16}$:

$$\mathcal{N}(xy_1) = \frac{63}{64} < 1 < \frac{33}{32} = \mathcal{N}(xy_2).$$

n	$\gamma(n)$	$\gamma(n)/2^{n-1}$
≤ 5	0	0
6	1	0.03125
7	1	0.015625
8	6	0.046875
9	12	0.046875
10	28	0.0546875
11	55	0.0537109375
12	140	0.068359375
$\vdots$	$\vdots$	$\vdots$
23	286165	0.06822705269...

Conjecture (Muller, 2000): *The proportion $\gamma(n)/2^{n-1}$ of elements of F_n without a fp-reciprocal converges toward* $\frac{1}{2} - \frac{3}{2}\log\frac{4}{3} = 0.06847689...$

Guillaume Hanrot, Joël Rivat, Gérald Tenenbaum, P. Z., *Density Results on Floating-Point Invertible Numbers*, 2001, to appear in *Theoretical Computer Science.*

Theorem 1. For $r = 0, 1, 2$, let $\gamma_r(k)$ denote the number of $x \in F_k \cap [1, 2[$ having exactly r fp-reciprocals. Then:

$$\begin{aligned}
\gamma_0(k)/2^{k-1} &= \tfrac{1}{2} - \tfrac{3}{2}\log\tfrac{4}{3} + O(2^{-k/3}) = 0.0684768917... + O(2^{-k/3}) \\
\gamma_1(k)/2^{k-1} &= 1 - \tfrac{3}{2}\log\tfrac{9}{8} + O(2^{-k/3}) = 0.8233254464... + O(2^{-k/3}) \\
\gamma_2(k)/2^{k-1} &= -\tfrac{1}{2} + \tfrac{3}{2}\log\tfrac{3}{2} + O(2^{-k/3}) = 0.1081976622... + O(2^{-k/3})
\end{aligned}$$

Numbers of fp-reciprocals of $x = \frac{m}{2^{k-1}}$ is the number of integers n such that

$$\frac{2^{2k-1} - 2^{k-2} - 1}{m} < n \leq \frac{2^{2k-1} + 2^{k-1}}{m}.$$

$$N(m) = \lfloor \frac{2^{2k-1} + 2^{k-1}}{m} \rfloor - \lfloor \frac{2^{2k-1} - 2^{k-2} - 1}{m} \rfloor$$

$$\gamma(k) = \sum_{\substack{2^{k-1} \leq m < 2^k \\ N(m)=0}} 1.$$

```
N := proc(m,k)
   iquo(2^(2*k-1)+2^(k-1), m) - iquo(2^(2*k-1)-2^(k-2)-1, m)
end:

gam := proc(k)
   s:=0; for m from 2^(k-1) to 2^k-1 do if N(m,k)=0 then s:=s+1 fi od: s
end:

> seq(gam(k), k=2..12);
                     0, 0, 0, 0, 1, 1, 6, 12, 28, 55, 140
```

Proof of Theorem 1 uses sums of first Bernoulli function $B_1(u) = u - \lfloor u \rfloor - \frac{1}{2}$, then reduction to exponential sums. Links with Voronoï's asymptotic formula on the divisor problem, Van der Corput's method, results from Vaaler, Titchmarsh or Tenenbaum.

Theorems 2, 3, 4. *Every number of F_k admits at most one fp-inverse square root. The number δ^+ (resp. δ^-) of elements of $F_k \cap [1, 2[$ (resp. $F_k \cap [\frac{1}{2}, 1[$) admitting a fp-inverse square root satisfies:*

$$\begin{aligned}
\delta^+/2^{k-1} &= \tfrac{3\sqrt{2}-3}{2} + O(2^{-k/3}) = 0.621320343... + O(2^{-k/3}) \\
\delta^-/2^{k-1} &= \tfrac{3\sqrt{2}-3}{2\sqrt{2}} + O(2^{-k/3}) = 0.4393398278... + O(2^{-k/3})
\end{aligned}$$

The Table Maker's Dilemma

Let x a n-bit fp-number, f a function, $\circ$ a rounding mode

Compute an approximation y of $f(x)$ with precision $m > n$, and error at most one ulp

How large must be m to ensure that $\circ_n(y) = \circ_n(f(x))$?

The Table Maker's Dilemma (continued)

Worst case when $f(x) = \overbrace{b_1.b_2 ... b_n}^{n} \overbrace{000...000}^{m-n} ...$ or $\overbrace{b_1.b_2 ... b_n}^{n} \overbrace{111...111}^{m-n} ...$

Example for $n = 53$:

```
> Digits:=105:
> x:=4621447055448553/2^11:
> evalf(arctan(x));
1.570796326794453467989853834299733539837341807159746625005 93\
    366229299010382513844136484512822365141260 49


> convert(%, binary);
1.1001001000011111101101010100010001000010010101001100100000 0\
    0000000000000000000000000000000000000111011
```

Algebraic Functions

Tomas Lang, Jean-Michel Muller, *Bounds on Runs of Zeros and Ones for Algebraic Functions*, ARITH'15, 2001

function	run of ones	run of zeros
$1/x$	$\leq n-1$	$= n-1$
x/y	$= n-1$	$= n-1$
$\sqrt{x}$	$n+1$	$= n-1$
$1/\sqrt{x}$	$\leq 2n+1$	$\leq 2n+1$
$\sqrt{x^2+y^2}$	$\leq n+1$	$\leq n+1$
$x^{1/q}$	$\leq \lceil \log_2 q \rceil + (n+1)(q-1) - 1$	same
$\frac{x}{\sqrt{x^2+y^2}}$	$\leq 3n+2$	$\leq 3n+2$

Actual Maxima

Reciprocal. $x = 1 - 2^n \Longrightarrow \frac{1}{x} = 1 + 2^{-n} + 2^{-2n} + \ldots$ ($n-1$ zeros)

n odd: $1 - 2^{-\frac{n-1}{2}} + 2^{-n} \Rightarrow 1 + 2^{-\frac{n-1}{2}} + 2^{-n} - 2^{-2n} \ldots$ ($n-1$ ones)

n even: bound attained if $2^{2n}+1$ has a factor $2^{n-1} \leq b < 2^n$:

```
foo := proc(n)
  select(proc(b) 2^(n-1)<=b and b<2^n end,
         numtheory[divisors](2^(2*n)+1))
end:

> seq(2*k=foo(2*k), k=1..12);
2 = {}, 4 = {}, 6 = {}, 8 = {}, 10 = {}, 12 = {}, 14 = {},

    16 = {}, 18 = {}, 20 = {}, 22 = {}, 24 = {12648641}
```

```
> Digits:=50:
> evalf(1/12648641);
                                                                           -7
     .79059876867404174092695017591217902381765756495105 10

> convert(%, binary);
                                                                           -23
     .10101001110001111010000001111111111111111111111101 10
```

Open Problems

Is the bound of $2n+1$ attained for the reciprocal square root? (only $n+4$ for $n \leq 20$, $n = 24, 32, 53$)

Reduced to find lower bound for $2^{3n-1} - x^2 y$ where $2^{n-1} \leq x, y < 2^n$

Example for $n = 21$: $x = 1721436$, $y = 1556245$

$$x^2 y - 2^{62} = 1616$$

```
> Digits:=61: convert(evalf(1/sqrt(1556245)), binary);
                                                                              -10
.1101001000100010110111111111111111111111111111111111000000000 10
```

Elementary and Transcendental Functions

Remark: the set of fp-number with precision n is finite !

0. Compute a bound $m \leq B_f$ for function f
1. Compute an approximation y of $f(x)$ with precision $n + m$
2. Round that approximation wrt the given rounding mode

Problem: how to compute *efficiently* B_f?

Linear Approximation

PhD thesis of Vincent Lefèvre

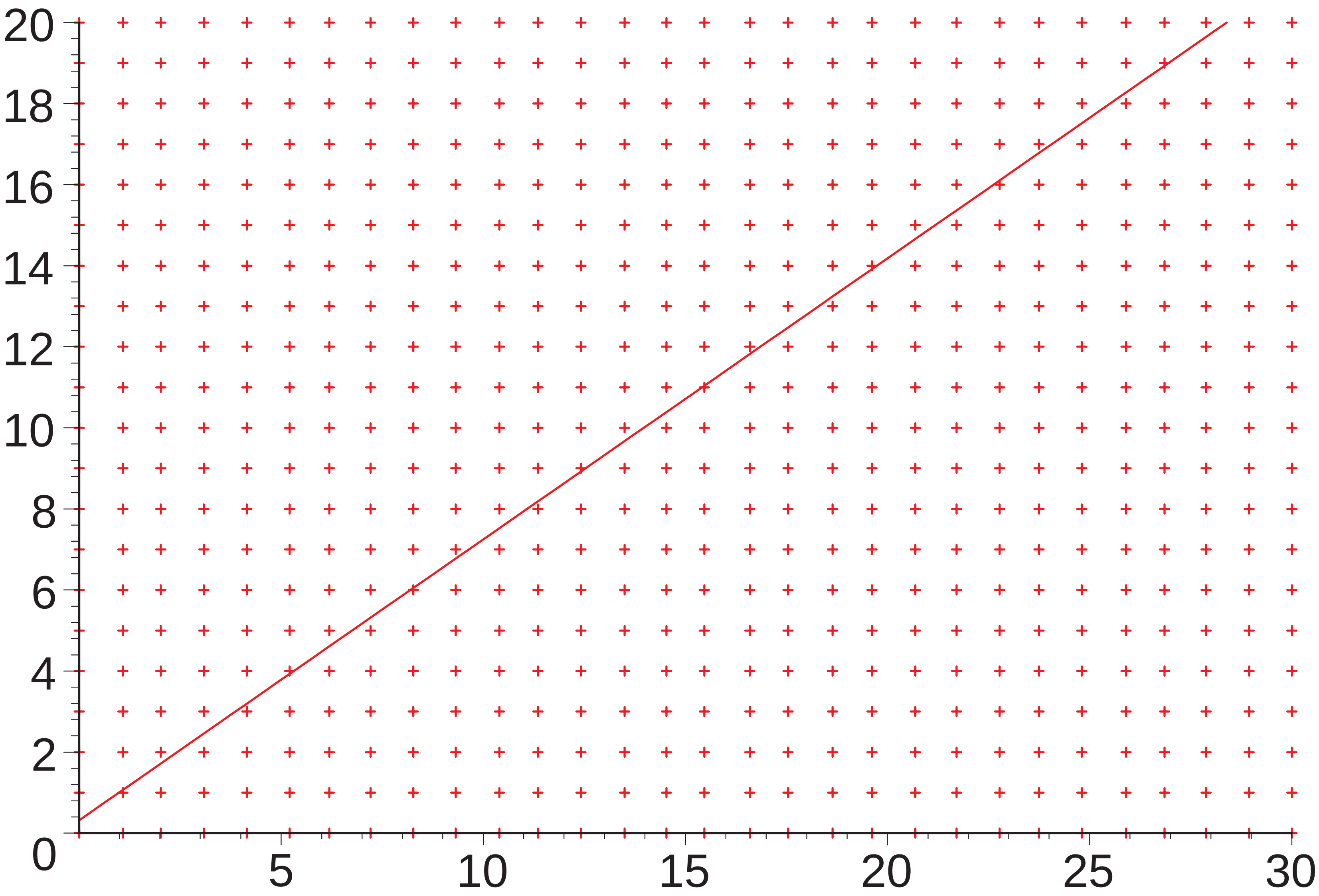
1. Cut the set of n-bit fp-numbers in small intervals with exponents of x and $f(x)$ constant

2. On each interval, approximate $f(x+t)$ by $a+bt$

3. As $t = i \cdot 2^e$, get $f(x+t) \cdot 2^{-e'} \sim \alpha + \beta i$

4. Find $\min_{|i|<N} \mathrm{frac}(\alpha + \beta i)$ with clever algorithm (three-distance theorem + modified extended gcd algorithm)

$\Longrightarrow$ double precision doable (with months of cpu time) but quadruple precision still out of reach

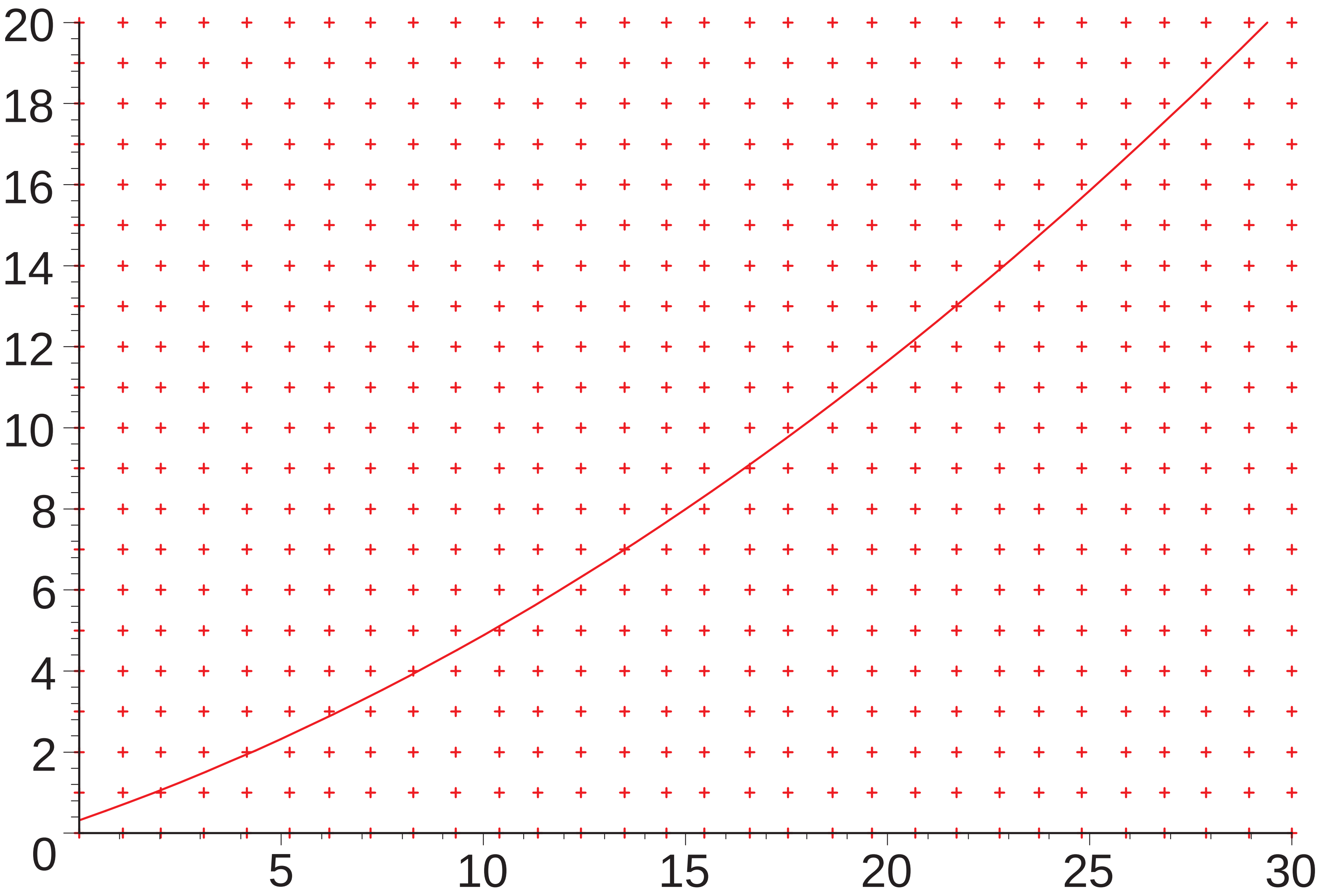

20
18
16
14
12
10
8
6
4
2
0
5
10
15
20
25
30

Polynomial Approximation

1. Cut the set of n-bit fp-numbers in small intervals with exponents of x and $f(x)$ constant
2. On each interval, approximate $f(x+t)$ by $a + bt + ct^2$
3. As $t = i \cdot 2^e$, get $f(x+t) \cdot 2^{-e'} \sim \alpha + \beta i + \gamma i^2$

Clever algorithm to find

$\min_{|i|<N} \operatorname{frac}(\alpha + \beta i + \gamma i^2)$?


20
18
16
14
12
10
8
6
4
2
0
5
10
15
20
25
30