Computer Arithmetic

Vasily Arzhanov

Reactor Physics, KTH

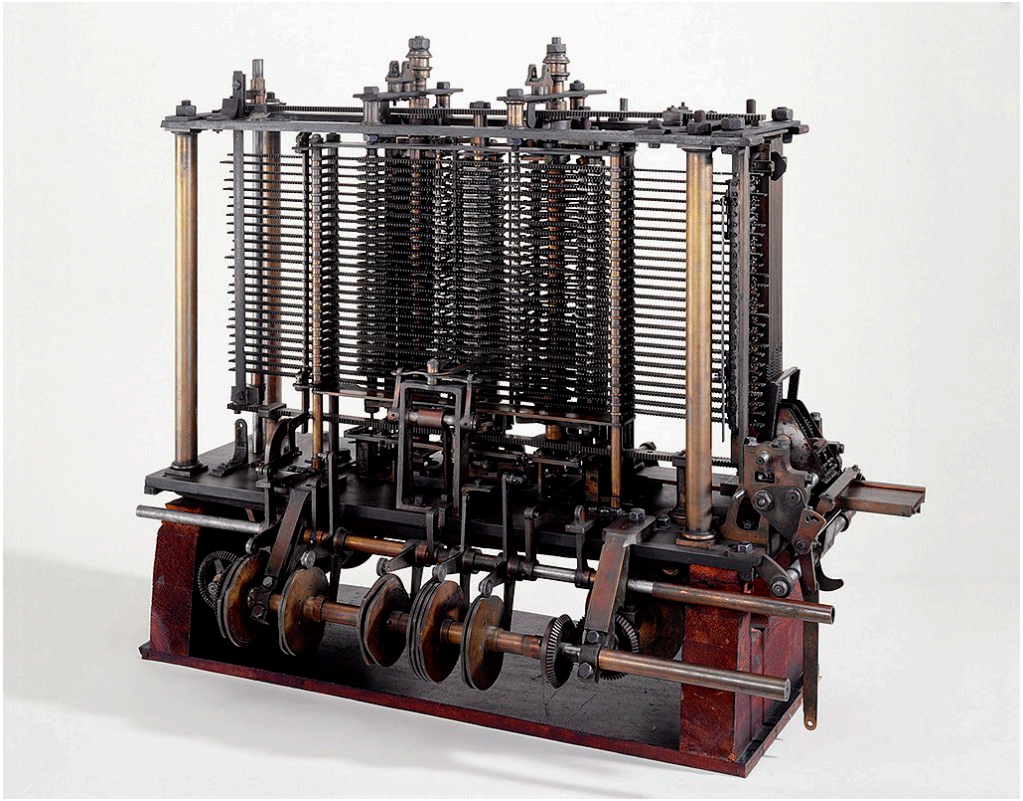
Etimology

- 1613: First known use of the word ”computer” in the book by Richard Braithwait, “*The Yong Mans Gleanings*”:
- “I haue [sic] read the truest computer of Times, and the best Arithmetician that euer [sic] breathed, and he reduceth thy dayes into a short number”.
- The first meaning is ”human computer”.
- 1833: Charles Babbage – Analytical Engine; Ada Lovelace.
- 1897: The first usage meaning ”calculating machine” of any type.
- 1937: Turing machine.
- By 1943, the most (human) computers were women.
- 1945: Computer = Programmable digital electronic computer.

Analytical Engine

- Arithmetic logic unit
- Control flow:
  - Conditional brunching
  - Loops
- Integrated memory

First general-pupose computer



Overview

- FP Number System
- Special Quantities
- Machine Epsilon
- Behaviour of Discretisation Error
- Rounding
- Loss of Significance

Sources of Error

- Physical Model
- Mathematical Model
- Initial Data
- Descrete Model
- Solution Method
- Data Representation in Computers
- Round off

Floating Point Numbers

Number range: $10^{100} \div 10^{-100}$

Nearly all computers use floating point numbers

Decimal numbers: $x = \pm 0.d_1 d_2 \cdots d_p \times 10^e$

Binary numbers: $x = \pm 0.d_1 d_2 \cdots d_p \times 2^e$

Generally: $x = \pm 0.d_1 d_2 \cdots d_p \times \beta^e$

β = the number base;
e = the exponent;
p = the precision;
$0.d_1 d_2 \ldots d_p$ = the fraction (sometimes mantissa, discouraged!).

$$0 \le d_i \le \beta - 1$$

Terminology

123.45 can be represented in several standard forms:

- 0.12345×10^3 – scientific;
- 1.2345×10^2 – normalized (engineering);
- 12345×10^{-2} – decimal.

$$0.12345 = \begin{cases} \text{Mantissa} & \leq 2005 \\ \text{Significand} & \text{nowadays} \end{cases}$$

Mantissa was commonly used until 2005.
Nowadays, this use of mantissa is discouraged by the IEEE floating point committee and by some professionals (W. Kahan, D. Knuth)

$123.45 = 123 + 0.45$

$[123.45] = 123$ (integer part)

$\{123.45\} = 0.45$ (fraction part = mantissa in American English)

Floating Point System

It is also termed as the real number system.

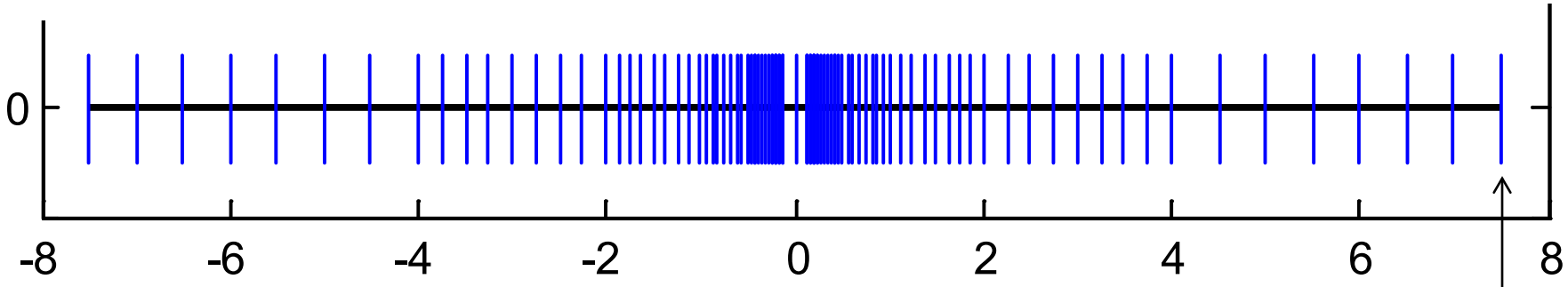
$$F(\beta, p, L, U) = \left\{ x \in \mathbb{Q} \;\middle|\; x = \pm\left(\frac{d_1}{\beta} + \frac{d_2}{\beta^2} + \ldots + \frac{d_p}{\beta^p} \right) \times \beta^e \right\}$$
$$0 \le d_i < \beta; \qquad L \le e \le U$$

If $\forall x \neq 0 \longrightarrow d_1 \neq 0$ The system is said to be normalized.

Example

$$\beta = 2;\ \ p = 4;\ L = -2;\ \ U = 3.$$

$$|F| = 2(\beta - 1)\beta^{p-1}(U - L + 1) + 1 = 97$$



$$x_{\max} = 0.1111_2 \times 2^3$$

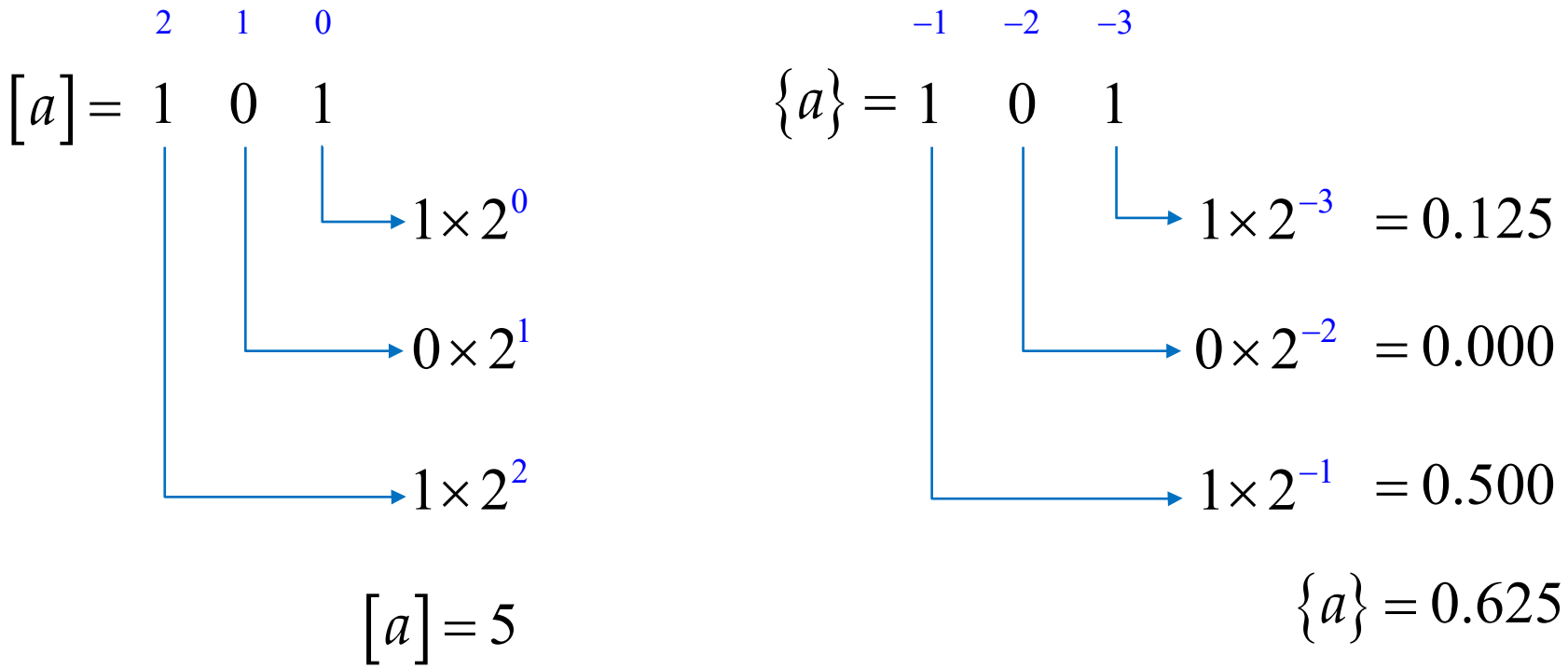
$$x_{\max} = \left(1 - \beta^{-p}\right)\beta^{U} = 7.5$$

$$x_1 = \beta^{L-1} = 0.125$$

$$\varepsilon_M = \beta^{1-p} = 0.125$$

Binary to Decimal

$$a = 101.101_2 = 101_2 + 0.101_2 = [a] + \{a\}$$

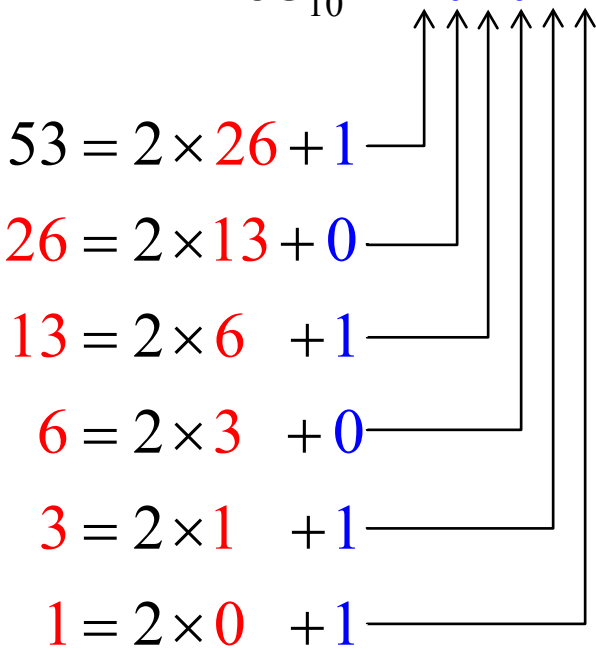


$$101.101_2 = 5.625$$

Decimal to Binary

$$a = 53.7_{10} = 53_{10} + 0.7_{10} = [a] + \{a\}$$

$53_{10} = 101011_2$ $\qquad$ $0.7_{10} = 0.1\,0110\,0110\ldots_2$



$$0.7 \times 2 = 0.4 + 1$$
$$\mathbf{0.4 \times 2 = 0.8 + 0}$$
$$0.8 \times 2 = 0.6 + 1$$
$$0.6 \times 2 = 0.2 + 1$$
$$0.2 \times 2 = 0.4 + 0$$
$$\mathbf{0.4 \times 2 = 0.8 + 0}$$
$$\vdots$$

$$53.7_{10} = 101011.\overline{10110}_2$$

Old Computers

Computer	β	p	L	U	ε_M
Univac 1108	2	27	-128	127	1.49×10^{-8}
Honeywell 6000	2	27	-128	127	1.49×10^{-8}
PDP-11	2	24	-128	127	1.19×10^{-7}
Control Data 6600	2	48	-975	1070	7.11×10^{-15}
Cray-1	2	48	-16384	8191	7.11×10^{-15}
Illiac-IV	2	48	-16384	16383	7.11×10^{-15}
SETUN	3	18	?	?	7.74×10^{-9}
Burroughs B5500	8	13	-51	77	1.46×10^{-11}
HP-45	10	10	-98	100	1.00×10^{-9}
TI SR-5x	10	12	-98	100	1.00×10^{-11}
IBM-360/370	16	6	-64	63	9.54×10^{-7}
IBM-360/370	16	14	-64	63	2.22×10^{-16}

Peculiarities

- (A+1)–A ≠ A+(1–A)
- IBM/370: every bit pattern is a valid number
- VAX: reserves certain bits for special qu.
- CDC 6600: reserves INDEF and INF
- IBM/360: SQRT(–4) → error message
- IBM/370: SQRT(–4) = 2
- IEEE : SQRT(–4) = NaN

IEEE Standards

- IEEE 754-1985
- IEEE 854-1987
- IEEE 754-2008 (current version)

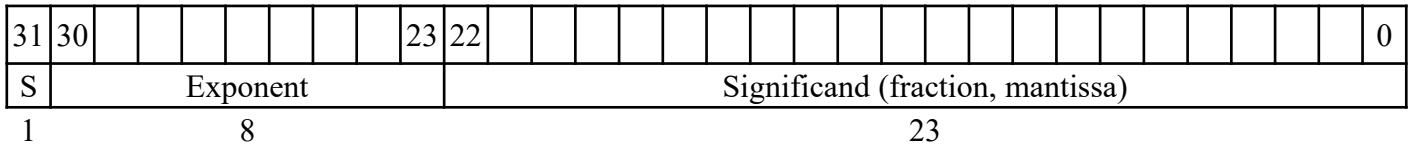
The Standard Defines:

- Arithmetic formats
  - Binary
  - Decimal
- Interchange formats
- Special numbers and values
  - Subnormal numbers
  - NaNs (qNaN, sNaN); ±0; ±∞
- Operations
- Rounding rules
- Exception handling

Formats

IEEE: $F(b,p,emin,emax) = \left\{ x = (-1)^S \times d_1 d_2 \cdots d_p \times b^e \right\}$

Name	Common Name	Base *b*	Digits *p*	*emin*	*emax*	Decimal *p*	Decimal *emax*
binary 16	Half precision	2	10+1	-14	+15	3.31	4.51
binary 32	Single precision	2	23+1	-126	+127	7.22	38.23
binary 64	Double precision	2	52+1	-1024	+1023	15.92	307.95
binary 128	Quadruple precision	2	112+1	-16382	+16383	34.02	4931.77
decimal 32	Single precision dec.	10	7	-95	+96	7	96
decimal 64	Double precision dec.	10	16	-383	+384	16	384
decimal 128	Quad. precision dec.	10	34	-6143	+6144	34	6144



Special Numbers and Values

- Subnormal numbers
- Signed Zeros
  - Plus zero, +0
  - Minus zero, -0
- Signed Infinities
  - Plus infinity, +∞
  - Minus infinity, -∞
- Not a Number, NaN
  - Quiet NaN, qNaN
  - Signaling NaN, sNaN

$$\text{Underflow} \rightarrow \pm 0$$
$$\text{Overflow} \rightarrow \pm \infty$$

$$+\infty = \frac{1}{+0}$$

$$+0 = \frac{1}{1/+0} \quad \left(x = \frac{1}{1/x} \right)$$

$$\text{NaN} = \frac{0}{0}, \frac{\infty}{\infty}, 0 \times \infty$$

Operations

- Arithmetic, $\sqrt{}$, Fused mult-add, Remainder
- Conversions (between formats, strings)
- Scaling, quantizing
- Manipulating the sign
- Comparison
- Classification and testing for NaNs
- Miscellaneous

Rounding

The standard requires exact rounding!

1) Round to nearest, ties to even
2) Round to nearest, ties away from zero
3) Round toward zero (truncation)
4) Round toward +∞ (round up, ceiling)
5) Round toward -∞ (round down, floor)

Guard Digits

Example: $10.1 - 9.93 = 0.17$ in $F(b = 10, p = 3)$

$$fl(10.1) = 0.101 \times 10^2$$
$$fl(9.93) = 0.993 \times 10^1$$

1) Allignment: 0.101×10^2
0.099×10^2

2) Substraction: $0.002 \times 10^2 = 0.2$

1) Extra digit: 0.1010×10^2
0.0993×10^2

2) Substraction: $0.0017 \times 10^2 = 0.17$

Exact Rounding

IEEE requires:

1) Perform operation exactly;
2) Round-off.

Binary arithmetic: 2 guard digits are enough!
Double precision: 64 bits (memory)
Intel processors: 80 bits (ALU)

● : operation on floating-point numbers
○ : infinite-precision operation

IEEE: $x \bullet y = \text{round}(x \circ y)$

Ties to Even

$$Round(1.25) = \begin{cases} 1.3 & \text{up} \\ 1.2 & \text{down} \end{cases}$$

0,1,2,3,4,5,6,7,8,9

Half the time up; another half down.

Least significant digit must be even:

$$Round(1.25) = 1.2$$
$$Round(1.35) = 1.4$$

Round up
($\beta = 10$; $p = 3$)

Let $x_0 = 1.00$ and $y = 0.555$
$x_0 + y = 1.555 = 1.56$
$x_1 = 1.56 - 0.555 = 1.005 = 1.01$
$x_1 + y = 1.01 + 0.555 = 1.565 = 1.57$
$x_2 = 1.57 - 0.555 = 1.015 = 1.02$

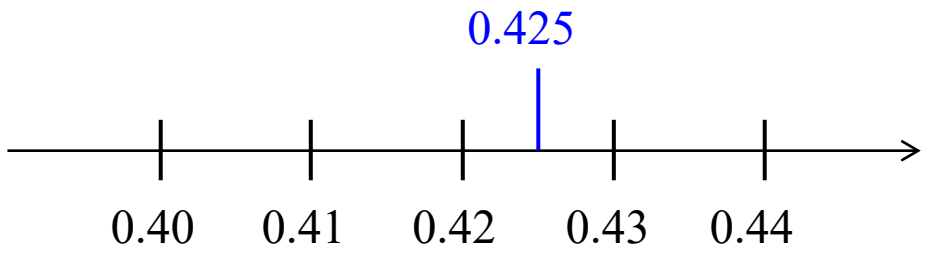
Drifting upward!!

Theorem
Let x and y be FP numbers
$x_0 = x$; $x_1 = (x_0 + y) - y$;
$x_2 = (x_1 + y) - y$;

$x_n = (x_{n-1} + y) - y$.

If addition and subtraction are exactly rounded using round to even then either $x_n = x_0$ or $x_n = x_1$.

Rounding Example

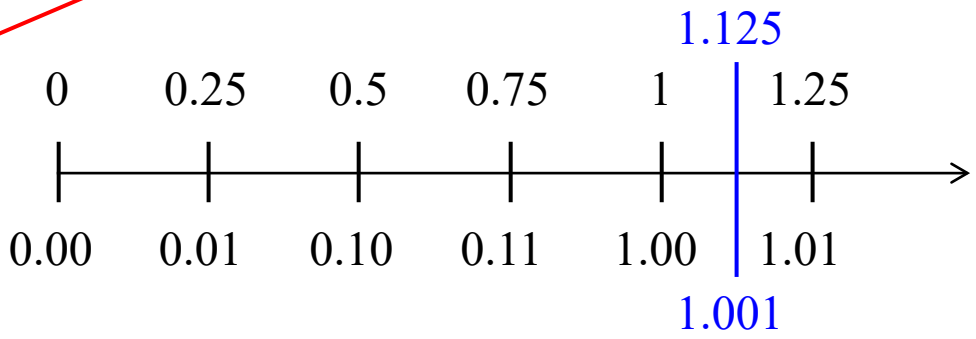


$0.425001 \approx 0.43$ (Nearest)

$0.425 \approx 0.42$ (Even)

Decimal

Binary



$1.001001 \approx 1.01$ (Nearest)

$1.001 \approx 1.00$ (Even)

Defining Machine Epsilon

Widespread: $\varepsilon_M = b^{1-p}$

Definition I. Machine epsilon is the smallest positive number, ε_M, such that $1 + \varepsilon_M \neq 1$

Variant: $\varepsilon_M = b^{1-p}$

Definition II. Machine epsilon is the distance between 1 and the next floating point number.

Scientific: $\varepsilon_M = b^{1-p}/2$

$\varepsilon_M = 2^{-p}$

Definition III. Machine epsilon is the maximum relative error for the chosen rounding procedure.

Machine Epsilons

IEEE does not define the term machine epsilon.

Definition I and II. ISO C standard, C++, Ada, Mathematica, Matlab, Octave, Python.

Definition III. Prof. J. Demmel, LAPACK, Scilab.

Finding Machine Epsilon

Definition. Machine epsilon is the smallest positive number, ε_M, such that $1+\varepsilon_M \neq 1$

The quantity is also called **macheps** or **unit roundoff** and is denoted as Greek epsilon, ε, or bold Roman, **u**.

```
eps = 1;
while 1+eps/2 ~= 1
   eps = eps/2;
end
eps
2.220446e-16
```

2^{-52}

Deducing Machine Epsilon

$1 + 2^{-53}$

$1.\boxed{00000000000000000000000000000000000000000000000000}\times 2^{0}$

$0.\boxed{00000000000000000000000000000000000000000000000000}1\times 2^{0}$

$1.\boxed{00000000000000000000000000000000000000000000000000}1\times 2^{0}$

$1 + 2^{-53} = 1 \quad \varepsilon_M = 2^{-52}$

Standard Machine Epsilons

IEEE-754	Common name	b	p	$\varepsilon_M = b^{1-p}$
Binary 16	Half precision	2	11	$2^{-10} = 9.77\times10^{-4}$
Binary 32	Single precision	2	24	$2^{-23} = 1.19\times10^{-7}$
Binary 64	Double precision	2	53	$2^{-52} = 2.22\times10^{-16}$
Binary 80	Extended precision	2	64	$2^{-63} = 1.08\times10^{-19}$
Binary 128	Quadruple precision	2	113	$2^{-112} = 1.93\times10^{-34}$
Decimal 32	Single precision dec.	10	7	10^{-6}
Decimal 64	Double precision dec.	10	16	10^{-15}
Decimal 128	Quadruple precision d.	10	34	10^{-33}

Representation Error

$$fl(x) = x(1+\delta) \qquad |\delta| \le \begin{cases} b^{1-p}/2 = \frac{1}{2}\varepsilon_M & \text{rounding} \\ b^{1-p} = \varepsilon_M & \text{truncating} \end{cases}$$

$$\left|\frac{fl(x)-x}{x}\right| \le \frac{1}{2}\varepsilon_M = \frac{1}{2}b^{1-p} = 2^{-p} \quad \text{(rounding in binary case)}$$

Numerical Differentiation

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \approx \frac{f(x+h) - f(x)}{h}$$

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2} f''(x) + O(h^3)$$

$$f'(x) \approx \frac{f(x+h) - f(x)}{h} = f'(x) + \frac{1}{2} f''(x)h + O\left(h^2\right)$$

$$Err\left(x,h\right) = \frac{1}{2} f''(x)h + O\left(h^2\right) \xrightarrow[h \to 0]{} 0$$

More Accurate Analysis

$$f'(x) \approx \frac{f(x+h)-f(x)}{h} \longrightarrow f'(x) \approx \frac{f^M(x+h)-f^M(x)}{h}$$

$$f^M(x) = f(x)\cdot\big(1+\delta(x)\big); \qquad \big|\delta(x)\big| \le \frac{1}{2}\varepsilon_M \ll 1$$

$$\frac{f^M(x+h)-f^M(x)}{h} = \frac{f(x+h)-f(x)}{h} + \frac{f(x+h)\delta(x+h)-f(x)\delta(x)}{h}$$

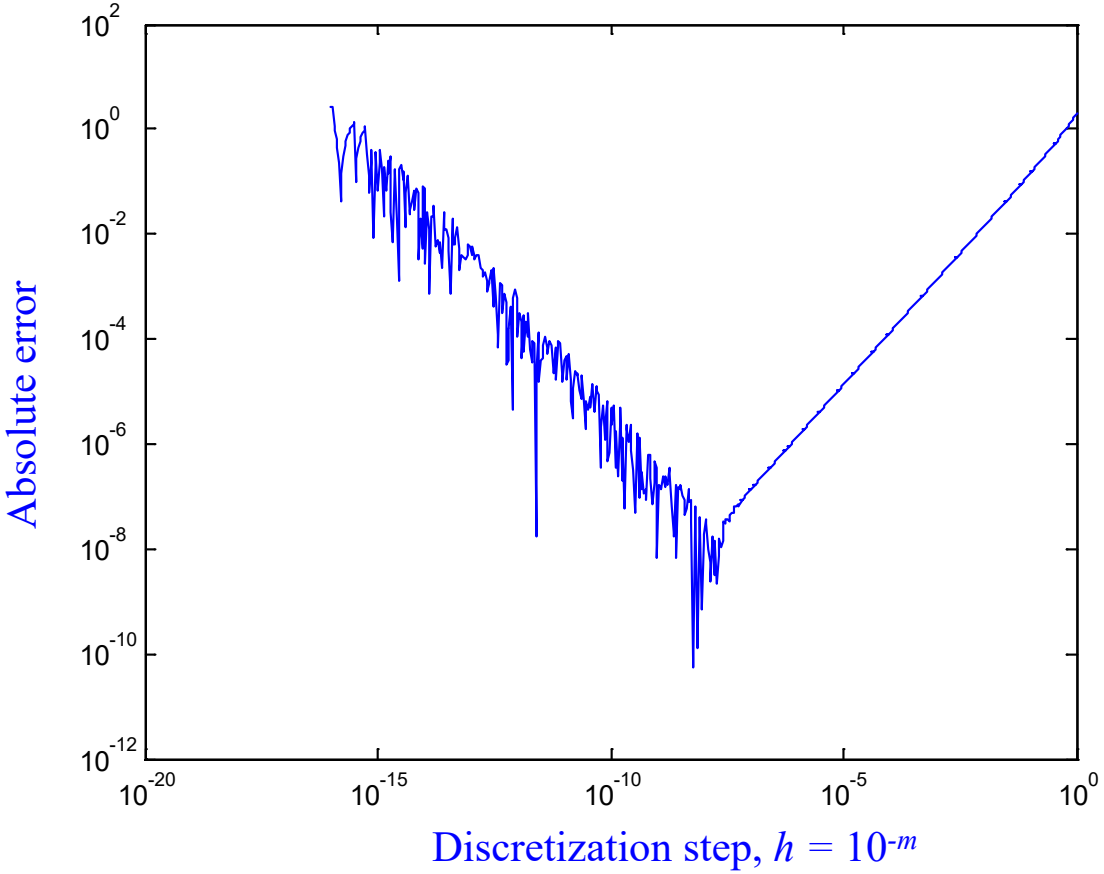
Finite-Difference Error

$$Err(x,h) = \frac{1}{2} f''(x)h + O\left(h^2\right) + \frac{f(x+h)\delta(x+h) - f(x)\delta(x)}{h}$$

$$\left|Err(x,h)\right| \le \frac{1}{2}\left|f''(x)\right|h + O\left(h^2\right) + \frac{\varepsilon_M \max\left|f\right|}{h}$$

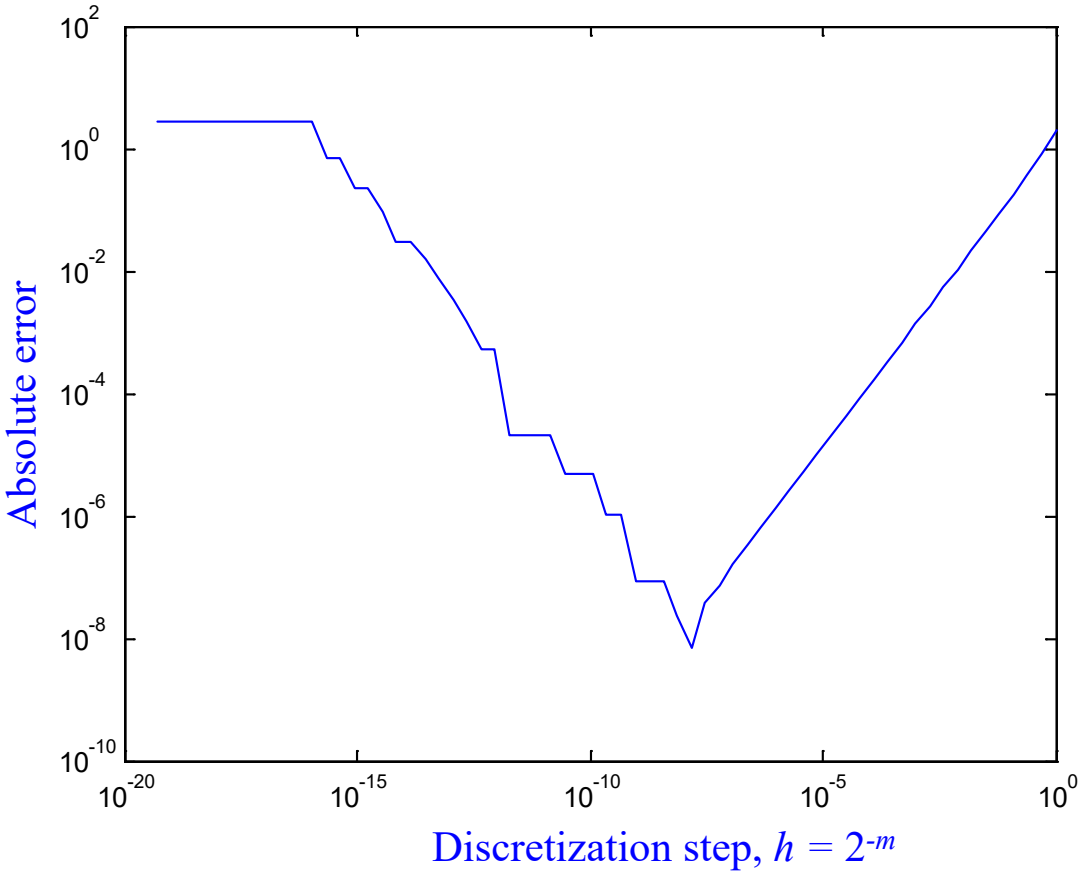
Round-off Errors

$$y(x) = e^x \qquad y'(x) \approx \frac{y(x+h) - y(x)}{h} \qquad x = 1$$



Representation Errors

$$y(x) = e^x \qquad y'(x) \approx \frac{y(x+h) - y(x)}{h} \qquad x = 1$$



Representation Error

$$0.1_{10} = 0.0\overline{0011}_2 = 0.0\ 0011\ 0011\ 0011\ \ldots$$

$$fl(0.1) = 0.1000000000000000055511151231257827021181583404541$$

```
>> format long
>> x = 9.4
x =
9.40000000000000
>> y = x-9
y =
0.40000000000000
>> z = y-0.4
z =
3.3306690738754 70e-16
>> 3*2^(-53)
ans =
     3.3306690738754 70e-16
```

Coding Sums and Products

$$S_n = x_1 + x_2 + \ldots + x_n; \qquad P_n = x_1 x_2 \cdots x_n$$

```
S=0;
for i=1:n
   S=S+x(i);
end
```

```
P=1;
for i=1:n
   P=P*x(i);
end
```

Computing Products

$$P_1 = x_1; \quad P_2 = fl\left(P_1 x_2\right) = P_1 x_2 \left(1+\delta_1\right) = x_1 x_2 \left(1+\delta_1\right)$$

$$P_3 = fl\left(P_2 x_3\right) = P_2 x_3 \left(1+\delta_2\right) = x_1 x_2 x_3 \left(1+\delta_1\right)\left(1+\delta_2\right)$$

$$P_n = fl\left(P_{n-1} x_n\right) = x_1 \cdots x_n \left(1+\delta_1\right)\cdots\left(1+\delta_{n-1}\right)$$

$$\delta \equiv \frac{P_n - x_1 \cdots x_n}{x_1 \cdots x_n} = \left(1+\delta_1\right)\cdots\left(1+\delta_{n-1}\right) - 1$$

$$\left|\delta\right| \le \left(1+\varepsilon_M/2\right)^{n-1} - 1 \approx n\,\varepsilon_M/2$$

Computing Sums

For positive machine numbers, $x_1, x_2, \ldots, x_n$

$$S_1 = x_1; \quad S_2 = fl(S_1 + x_2) = (x_1 + x_2)(1 + \delta_1)$$

$$S_3 = fl(S_2 + x_3) = \left[(x_1 + x_2)(1 + \delta_1) + x_3\right](1 + \delta_2)$$

$$S_n = fl(S_{n-1} x_n) = (x_1 + \ldots + x_n)(1 + \delta)$$

$$|\delta| \le (1 + \varepsilon_M / 2)^{n-1} - 1 \approx n\,\varepsilon_M / 2$$

Loss of Significance

In addition to round-off errors, there are other types of errors.

$$x = 0.3721478693 \qquad fl(x) = 0.37215$$
$$y = 0.3720230572 \qquad fl(y) = 0.37202$$
$$x - y = 0.0001248121 \qquad fl(x) - f(y) = 0.00013$$

$$\left| \frac{x - y - [fl(x) - fl(y)]}{x - y} \right| = \left| \frac{0.0001248121 - 0.00013}{0.0001248121} \right| \approx 0.04$$

Theorem on Loss of Significance

Let x and y be positive normalized floating point machine numbers such that $x > y$ and $2^{-q} \leq 1 - y/x \leq 2^{-p}$ then at most q and at least p significant binary bits are lost in the subtraction $x - y$.

Examples

$$y = \sqrt{x^2+1} - 1 \longrightarrow y = \frac{x^2}{\sqrt{x^2+1}+1}$$

$$y = x - \sin x = x - \left(x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots \right) \approx \frac{x^3}{6}\left\{1 - \frac{x^2}{20}\left[1 - \frac{x^2}{42}\left(1 - \frac{x^2}{72}\right)\right]\right\}$$

$$2^{-1} \le 1 - \frac{\sin x}{x} \xrightarrow[(\sin x > 0)]{} |x| \ge 1.9$$

Exponential Function

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots \qquad b = 10; \quad p = 5; \quad x = -5.5$$

$e^{-5.5}$ =

1.0000
−5.5000
+15.125
−27.730
.
.
+1.5997

0.0026363

13 terms

(Precise)
0.00408677

$e^{5.5}$ =

1.0000
+5.5000
+15.125
+27.730
.
.
+1.5997

0.0040865

Trigonometric Functions

$$E_1 = \frac{1-\cos x}{\sin^2 x} \qquad E_2 = \frac{1}{1+\cos x}$$

x	E_1	E_2
1.000000000000000	0.64922320520476	0.64922320520476
0.100000000000000	0.50125208628858	0.50125208628857
0.010000000000000	0.50001250020848	0.50001250020834
0.001000000000000	0.50000012499219	0.50000012500002
0.000100000000000	0.49999999862793	0.50000000125000
0.000010000000000	0.50000004138685	0.50000000001250
0.000001000000000	0.50004445029134	0.50000000000013
0.000000100000000	0.49960036108132	0.50000000000000
0.000000010000000	0.00000000000000	0.50000000000000
0.000000001000000	0.00000000000000	0.50000000000000
0.000000000100000	0.00000000000000	0.50000000000000
0.000000000010000	0.00000000000000	0.50000000000000
0.000000000001000	0.00000000000000	0.50000000000000
0.000000000000100	0.00000000000000	0.50000000000000

Quadratic Equations

$$ax^2 + bx + c = 0 \quad \Rightarrow \quad x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}, \quad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

b = 10, p = 8, emin = -50, emax = 50

$a = 1$; $b = -10^5$; $c = 1$.

exact: $x_1 = 99999.999990$; $x_2 = 0.0000100000$

comp: $x_1 = 100000.0$; $x_2 = 0$

Quadratic Solver

$$ax^2 + bx + c = a(x - x_1)(x - x_2) \longrightarrow c = ax_1x_2$$

$$x_1 = -\frac{b + \mathrm{sgn}(b)\sqrt{b^2 - 4ac}}{2a}$$

$$x_2 = \frac{c}{ax_1}$$

Overflow

$\beta = 10;\ p = 3;\ emax = 98.$

$x = 3 \times 10^{70};\ y = 4 \times 10^{70};$

$\sqrt{x^2 + y^2} = 5 \times 10^{70}$ (exact)

$x^2 \to x^2 = 9.99 \times 10^{98}$

$y^2 \to y^2 = 9.99 \times 10^{98}$

$x^2 + y^2 \to x^2 + y^2 = 9.99 \times 10^{98}$

$\sqrt{x^2 + y^2} = 3.16 \times 10^{49}$

IEEE requires that $x^2 = +\infty$

$$\sqrt{x^2 + y^2} = |y| \sqrt{1 + (x/y)^2}$$

Important

- FP Number System
- Special Quantities
- Machine Epsilon
- Behaviour of Discretisation Error
- Rounding
- Loss of Significance

Approximation Error

a true value

$\tilde{a}$ approximate value $\longrightarrow a \approx \tilde{a}$

Approximation Error $\quad \Delta a \equiv \tilde{a} - a$

Absolute Error is $|\Delta a|$

An upper bound is any (known) number Δ_a such that $|\Delta a| \leq \Delta_a$

$$\tilde{a} - \Delta_a \leq a \leq \tilde{a} + \Delta_a \longrightarrow a = \tilde{a} \pm \Delta_a$$