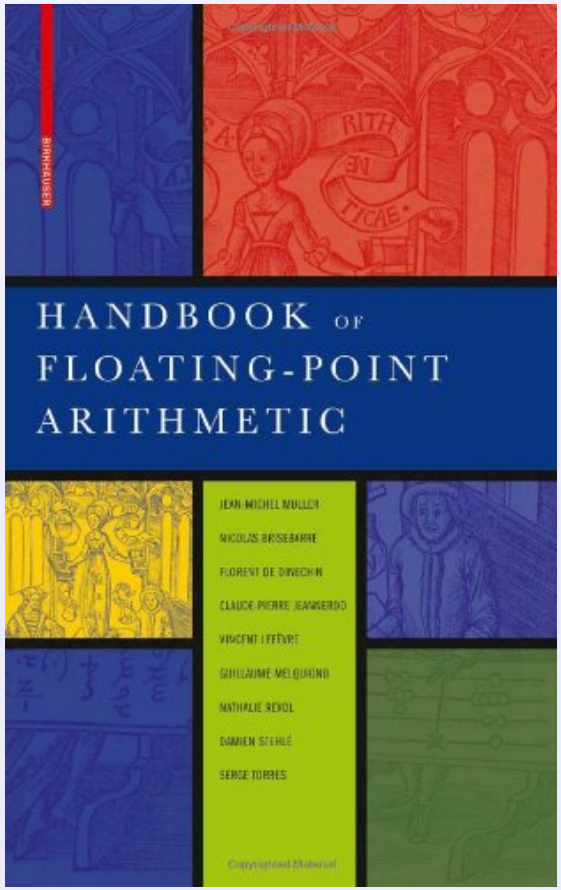
Introduction to floating point arithmetic

Matthias Petschow and Paolo Bientinesi

AICES, RWTH Aachen
petschow@aices.rwth-aachen.de

October 24th, 2013
Aachen, Germany



Muller et al. - Handbook of Floating-Point Arithmetic - **595 pages!**

- **Many topics not covered in this lecture** (e.g., hardware/software implementation of FPA, language support, cleverly using FPA)
- This lecture: **basics** of floating point representation and arithmetic
- Goal: Make you aware of the issues arising in finite precision computations
- See the references below for a more thorough treatment of the topic

References

- Article: "What every computer scientist should know about floating-point arithmetic", by David Goldberg
- Article: "Lecture Notes on the Status of IEEE Standard 754 for Binary Floating-Point Arithmetic", by William Kahan
- Book: "Accuracy and Stability of Numerical Algorithms", by Nick Higham
- Book: "Numerical Computing with IEEE Floating Point Arithmetic", by Michael Overton
- IEEE 754-1985 and IEEE 754-2008: "Standard for Floating-Point Arithmetic"

Numbers

- $\frac{123}{29} =$ (first 40 digits)

  $4.2413793103448275862068965517241379310 34\ldots$

- $\pi =$

  $3.1415926535897932384626433832795028841 97\ldots$

- In general: **Infinite** number of digits

Numbers

- $\frac{123}{29} =$ (first 40 digits)

  $4.2413793103448275862068965517241379310 34\ldots$

- $\pi =$

  $3.1415926535897932384626433832795028841 97\ldots$

- In general: **Infinite** number of digits

Computers

- **Finite** memory $\Rightarrow$ **Approximated numbers**

Floating point system $\mathbb{F}\backslash\{0\}$

$$(-1)^s \times d_0.d_1d_2d_3 \ldots d_{p-1} \times \beta^e$$

with base β, precision p, and

- $s \in \{0, 1\}$
- $d_i \in \{0, \ldots, \beta - 1\}$
- $d_0 \neq 0$
- $e_{min} \leq e \leq e_{max}$

IEEE-754 specifications ($\beta = 2$, $d_0 = 1$)

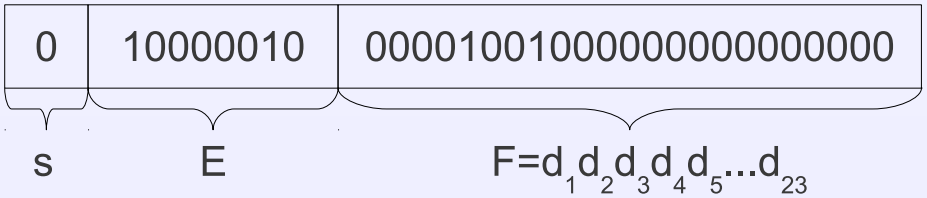
Single: $p = 24$, $e_{min} = -126$, $e_{max} = 127$
Double: $p = 53$, $e_{min} = -1022$, $e_{max} = 1023$
Additionally: ± 0, $\pm\infty$, subnormal numbers, NaNs

IEEE single precision

$\pm$	$a_1 a_2 a_3 \dots a_8$	$b_1 b_2 b_3 \dots b_{23}$

If exponent bitstring $a_1 \dots a_8$ is	Then numerical value represented is
$(00000000)_2 = (0)_{10}$	$\pm(0.b_1 b_2 b_3 \dots b_{23})_2 \times 2^{-126}$
$(00000001)_2 = (1)_{10}$	$\pm(1.b_1 b_2 b_3 \dots b_{23})_2 \times 2^{-126}$
$(00000010)_2 = (2)_{10}$	$\pm(1.b_1 b_2 b_3 \dots b_{23})_2 \times 2^{-125}$
$(00000011)_2 = (3)_{10}$	$\pm(1.b_1 b_2 b_3 \dots b_{23})_2 \times 2^{-124}$
$\downarrow$	$\downarrow$
$(01111111)_2 = (127)_{10}$	$\pm(1.b_1 b_2 b_3 \dots b_{23})_2 \times 2^{0}$
$(10000000)_2 = (128)_{10}$	$\pm(1.b_1 b_2 b_3 \dots b_{23})_2 \times 2^{1}$
$\downarrow$	$\downarrow$
$(11111100)_2 = (252)_{10}$	$\pm(1.b_1 b_2 b_3 \dots b_{23})_2 \times 2^{125}$
$(11111101)_2 = (253)_{10}$	$\pm(1.b_1 b_2 b_3 \dots b_{23})_2 \times 2^{126}$
$(11111110)_2 = (254)_{10}$	$\pm(1.b_1 b_2 b_3 \dots b_{23})_2 \times 2^{127}$
$(11111111)_2 = (255)_{10}$	$\pm\infty$ if $b_1 = \dots = b_{23} = 0$, NaN otherwise

IEEE single precision



$$(-1)^s \times (1 + f) \times 2^{E-127}$$

- $f = d_1 \cdot 2^{-1} + d_2 \cdot 2^{-2} + d_3 \cdot 2^{-3} + \ldots + d_{23} \cdot 2^{-23}$
- $e = E - 127$, biased exponent $1 \leq E \leq 254$
- Special values for $E = 0$:
  $F = 0$: ± 0
  $F \neq 0$: subnormal numbers with $d_0 = 0$ and implicit exponent $e = -126$
- Special values for $E = 255$:
  $F = 0$: $\pm\infty$
  $F \neq 0$: NaNs

IEEE double precision

$\pm$	$a_1 a_2 a_3 \dots a_{11}$	$b_1 b_2 b_3 \dots b_{52}$

If exponent bitstring is $a_1 \dots a_{11}$	Then numerical value represented is
$(00000000000)_2 = (0)_{10}$	$\pm(0.b_1 b_2 b_3 \dots b_{52})_2 \times 2^{-1022}$
$(00000000001)_2 = (1)_{10}$	$\pm(1.b_1 b_2 b_3 \dots b_{52})_2 \times 2^{-1022}$
$(00000000010)_2 = (2)_{10}$	$\pm(1.b_1 b_2 b_3 \dots b_{52})_2 \times 2^{-1021}$
$(00000000011)_2 = (3)_{10}$	$\pm(1.b_1 b_2 b_3 \dots b_{52})_2 \times 2^{-1020}$
$\downarrow$	$\downarrow$
$(01111111111)_2 = (1023)_{10}$	$\pm(1.b_1 b_2 b_3 \dots b_{52})_2 \times 2^{0}$
$(10000000000)_2 = (1024)_{10}$	$\pm(1.b_1 b_2 b_3 \dots b_{52})_2 \times 2^{1}$
$\downarrow$	$\downarrow$
$(11111111100)_2 = (2044)_{10}$	$\pm(1.b_1 b_2 b_3 \dots b_{52})_2 \times 2^{1021}$
$(11111111101)_2 = (2045)_{10}$	$\pm(1.b_1 b_2 b_3 \dots b_{52})_2 \times 2^{1022}$
$(11111111110)_2 = (2046)_{10}$	$\pm(1.b_1 b_2 b_3 \dots b_{52})_2 \times 2^{1023}$
$(11111111111)_2 = (2047)_{10}$	$\pm\infty$ if $b_1 = \dots = b_{52} = 0$, NaN otherwise

Example questions

Question 1

What is the largest finite floating point number is IEEE single precision?

Question 2

What is the smallest positive normalized floating point number is IEEE single precision? What is the smallest positive number?

Question 3

How many normalized IEEE single precision numbers and how many subnormal numbers are there?

Question 4

Given IEEE single/double precision ($p = 24, 53$), what is the gap between 1 and the next larger number?

Question 5

Between an adjacent pair of nonzero IEEE single precision real numbers, how many IEEE double precision numbers are there?

Question 6

What is the largest integer u such that all integer in the interval $[-u, u]$ are exactly representable in IEEE single precision format? What is the corresponding u for double precision?

Relative rounding error

Let $x \in \mathbb{R}$ and $|x| \in [\omega_{\min}, \omega_{\max}]$, then

$$\bar{x} = x(1+\delta) \quad \text{where} \quad |\delta| \leq u$$

and

$$\bar{x} = x/(1+\tilde{\delta}) \quad \text{where} \quad |\tilde{\delta}| \leq u$$

with u denoting the unit roundoff.

- $\omega_{\min}$ = smallest normalized positive floating point number
- $\omega_{\max}$ = largest normalized positive floating point number
- $\bar{x} = [x]$ = floating point representation of x
- If rounded to nearest float, $\bar{x} = RN(x)$ and $u = 2^{-p}$
- For other rounding modes, $u = 2^{-p+1}$

Task

Given a base 2 floating point format with precision p, show that if $x \in \mathbb{R}$ lies in the normalized range than
$RN(x) = x(1+\delta)$, with $|\delta| \leq 2^{-p}$,
where $RN()$ rounds to the nearest floating point number.

Question

What if $RN(x)$ is subnormal? Find an example where the above is not true.

Question

What if we truncate – that is round to zero $RZ()$ – instead of rounding to the nearest number $RN()$?

Inexact Arithmetic

$$\begin{array}{rl} 123.4 & + \\ .5678 & = \\ \hline 123.9678 & = \end{array}$$

Inexact Arithmetic

$$\begin{array}{rl} 123.4 & + \\ .5678 & = \\ \hline 123.9678 & = \end{array}$$

Inexact Arithmetic

	123.4	+
	.5678	=
	123.9678	=
Truncated	123.9	
Rounded	124.0	

Inexact Arithmetic

	123.4	+
	.5678	=
	123.9678	=
Truncated	123.9	
Rounded	124.0	

Associativity?

- Exact arithmetic:
  $(123.4 + .5678) + .5432 =$
  $123.4 + (.5678 + .5432)$

Inexact Arithmetic

	123.4	+
	.5678	=
	123.9678	=
Truncated	123.9	
Rounded	124.0	

Associativity?

- Exact arithmetic:
  $(123.4 + .5678) + .5432 =$
  $123.4 + (.5678 + .5432)$
- Inexact arithmetic:
  $(123.4 + .5678) + .5432 = 124.4$

Inexact Arithmetic

	123.4	+
	.5678	=
	123.9678	=
Truncated	123.9	
Rounded	124.0	

Associativity? No!

- Exact arithmetic:
  $(123.4 + .5678) + .5432 =$
  $123.4 + (.5678 + .5432)$
- Inexact arithmetic:
  $(123.4 + .5678) + .5432 = 124.4$
  $123.4 + (.5678 + .5432) = 124.5$

The standard floating point model

Given operation $\diamond \in \{+, -, \times, /\}$ and $x, y \in \mathbb{F}_0$,

$$[x \diamond y] = RN(x \diamond y) = (x \diamond y)(1 + \delta)$$

with $|\delta| \leq u$, provided no underflow/overflow occurred.

- $RN()$ can be replaced by other rounding modes.
- For $RN()$ we have seen that $u = 2^{-p}$.
- Also, $(x \diamond y)(1 + \delta)$ can be replaced by $(x \diamond y)/(1 + \delta)$.
- Similarly, $\sqrt{x} = RN(x)$.
- What about $\sin(x)$, $\cos(x)$, $\exp(x)$, $\log(x)$, ...?
- Notation: Assuming a left-to-right evaluation, it holds

$$[x + y + z/w] = \Big[\big[[x] + [y]\big] + \big[[z]/[w]\big]\Big]$$

Question 1

If x is a floating point number, is the floating point product $[1 \times x]$ equal to x?

Question 2

If $x \neq 0$ is a (finite) floating point number, is the floating point quotient $[x/x]$ equal to 1?

Question 3

If x is a floating point number, is the floating point product $[0.5 \times x]$ equal to floating point quotient $[x/2]$?

Question 4

Is it true that for all $a, b \in \mathbb{R}$ we have $[a + b] = [b + a]$ and $[a \times b] = [b \times a]$?

Question 5

Let $a = -1$, $b = 1$, and $c = 2^{-25}$. In IEEE single precision arithmetic, what are the results of $[a + [b + c]]$ and $[[a + b] + c]$?

Question 6

Let $a = b = 2^{513}$, and $c = 2^{-1022}$. In IEEE double precision arithmetic, what are the results of $[a \times [b \times c]]$ and $[[a \times b] \times c]$?

Question 7

What is the result calling the function **fun(2.0,0.0)** defined below (using IEEE floating point values)?

$$
\begin{aligned}
&fun(a, b)\ \{ \\
&\quad res = 1/(1/a + 1/b) \\
&\quad return(res)\ \}
\end{aligned}
$$

Question 8

What is a potential problem of the following program? How can it be fixed?

$$
\begin{aligned}
&hypot(a, b)\ \{ \\
&\quad c = a^2 + b^2 \\
&\quad c = sqrt(c) \\
&\quad return(c)\ \}
\end{aligned}
$$

Task

Calculating the roots of a quadratic polynomial with $roots()$ as below can cause inaccurate results (e.g. $a = c = 1$, $b = -10^8$ for double precision). Write a code that avoids cancellation as much as possible, e.g. by using $r_1 r_2 = c/a$. Are there other potential problems, such as overflow etc.?

$$
\begin{aligned}
&roots(a, b, c)\ \{ \\
&\quad r_1 = \frac{-b+\sqrt{b^2-4ac}}{2a} \\
&\quad r_2 = \frac{-b-\sqrt{b^2-4ac}}{2a} \\
&\quad return(r_1, r_2)\ \}
\end{aligned}
$$

$$f : A \to B, \quad y = f(x)$$

Es.: $f(x) = x^2 + \sin(2 * x)$

$$x = \frac{\pi}{123}, \qquad f(x) = ?$$

$$f : A \to B, \quad y = f(x)$$

Es.: $f(x) = x^2 + \sin(2 * x)$

$$x = \frac{\pi}{123}, \qquad f(x) = ?$$

- Exact arithmetic:

$$\left(\frac{\pi}{123}\right)^2 + \sin\left(2 * \frac{\pi}{123}\right) = \ldots$$

$$f : A \to B, \quad y = f(x)$$

Es.: $f(x) = x^2 + \sin(2 * x)$

$$x = \frac{\pi}{123}, \qquad f(x) = ?$$

- Exact arithmetic:

$$\left(\frac{\pi}{123}\right)^2 + \sin\left(2 * \frac{\pi}{123}\right) = \ldots$$

- Inexact arithmetic:
  $x \rightsquigarrow \hat{x}, \quad f \rightsquigarrow \hat{f}$ $\qquad \hat{f}(\hat{x})$ instead of $f(x)$

Example: Dot Product

$$x, y \in \mathbb{R}^n; \quad \kappa := x^T y$$

$$\kappa := \Big(\big((\chi_0 \psi_0 + \chi_1 \psi_1) + \cdots \big) + \chi_{n-2} \psi_{n-2} \Big) + \chi_{n-1} \psi_{n-1}$$

$$\begin{aligned} \check{\kappa} &= \Big(\Big(\big(\chi_0 \psi_0 (1 + \epsilon_*^{(0)}) + \chi_1 \psi_1 (1 + \epsilon_*^{(1)}) \big) (1 + \epsilon_+^{(1)}) + \cdots \Big) (1 + \epsilon_+^{(n-2)}) \\ &\qquad + \chi_{n-1} \psi_{n-1} (1 + \epsilon_*^{(n-1)}) \Big) (1 + \epsilon_+^{(n-1)}) \\ &= \sum_{i=0}^{n-1} \left(\chi_i \psi_i (1 + \epsilon_*^{(i)}) \prod_{j=i}^{n-1} (1 + \epsilon_+^{(j)}) \right) \end{aligned}$$

where $\epsilon_+^{(0)} = 0$ and $|\epsilon_*^{(0)}|, |\epsilon_*^{(j)}|, |\epsilon_+^{(j)}| \leq \mathsf{u}$ for $j = 1, \ldots, n-1$

Backward Stability

Let $f : \mathcal{D} \to \mathcal{R}$ be a map from the domain $\mathcal{D}$ to the range $\mathcal{R}$.
Let $\hat{f} : \mathcal{D} \to \mathcal{R}$ represent the execution in floating point arithmetic of a given algorithm $\mathcal{A}$ that computes f.

$\mathcal{A}$ is said to be **backward stable** if
for all $x \in \mathcal{D}$ there exists a perturbed input $\bar{x} \in \mathcal{D}$, close to x, such that $\hat{f}(x) = f(\bar{x})$.

Backward Stability

Let $f : \mathcal{D} \to \mathcal{R}$ be a map from the domain $\mathcal{D}$ to the range $\mathcal{R}$.
Let $\hat{f} : \mathcal{D} \to \mathcal{R}$ represent the execution in floating point arithmetic of a given algorithm $\mathcal{A}$ that computes f.

$\mathcal{A}$ is said to be **backward stable** if
for all $x \in \mathcal{D}$ there exists a perturbed input $\bar{x} \in \mathcal{D}$, close to x, such that $\hat{f}(x) = f(\bar{x})$.

I.e., the result computed in floating point arithmetic $(\hat{f}(x))$ equals the result obtained when the mathematically exact function (f) is applied to slightly perturbed data $(\bar{x})$.

The difference between $\bar{x}$ and x, is the perturbation to the original input x.