MATH 3795
Lecture 2. Floating Point Arithmetic

Dmitriy Leykekhman

Fall 2008

Goals

- Basic understanding of computer representation of numbers
- Basic understanding of floating point arithmetic
- Consequences of floating point arithmetic for numerical computation

Representation of Real Numbers

In everyday life we use decimal representation of numbers. For example

$$1234.567$$

for us means

$$1 * 10^4 + 2 * 10^3 + 3 * 10^2 + 4 * 10^0 + 5 * 10^{-1} + 6 * 10^{-2} + 7 * 10^{-3}.$$

Representation of Real Numbers

In everyday life we use decimal representation of numbers. For example

$$1234.567$$

for us means

$$1 * 10^4 + 2 * 10^3 + 3 * 10^2 + 4 * 10^0 + 5 * 10^{-1} + 6 * 10^{-2} + 7 * 10^{-3}.$$

More generally

$$\dots d_j \dots d_1 d_0 . d_{-1} \dots d_{-i} \dots$$

represents

$$\cdots d_j * 10^j + \cdots + d_1 * 10^1 + d_0 * 10^0 + d_{-1} * 10^{-1} + \cdots + d_{-i} * 10^{-i} + \cdots .$$

Representation of Real Numbers

Let $\beta \geq 2$ be an integer. For every $x \in I\!R$ there exist integers e and $d_i \in \{0, \ldots, \beta - 1\}$, $i = 0, 1, \ldots$, such that

$$x = sign(x) \left(\sum_{i=0}^{\infty} d_i \beta^{-i} \right) \beta^e. \tag{1}$$

The representation is unique if one requires that $d_0 > 0$ when $x \neq 0$.

Representation of Real Numbers

Let $\beta \geq 2$ be an integer. For every $x \in I\!R$ there exist integers e and $d_i \in \{0, \ldots, \beta - 1\}$, $i = 0, 1, \ldots,$ such that

$$x = sign(x) \left(\sum_{i=0}^{\infty} d_i \beta^{-i} \right) \beta^e. \qquad (1)$$

The representation is unique if one requires that $d_0 > 0$ when $x \neq 0$.

Example

$$\frac{11}{2} = 5 * 10^0 + 5 * 10^{-1} = (5.5)_{10},$$
$$\frac{11}{2} = 1 * 2^2 + 0 * 2^1 + 1 * 2^0 + 1 * 2^{-1}$$
$$= (1 * 2^0 + 0 * 2^{-1} + 1 * 2^{-2} + 1 * 2^{-3}) * 2^2 = (1.011)_2 * 2^2.$$

Floating Point Numbers

In a computer only a finite subset of all real numbers can be represented. These are the so–called floating point numbers and they are of the form

$$\bar{x} = (-1)^s \left(\sum_{i=0}^{m-1} d_i \beta^{-i} \right) \beta^e$$

with $d_i \in \{0, \ldots, \beta - 1\}$, $i = 0, 1, \ldots, m-1$, and $e \in \{e_{\min}, \ldots, e_{\max}\}$.

Floating Point Numbers

In a computer only a finite subset of all real numbers can be represented. These are the so–called floating point numbers and they are of the form

$$\bar{x} = (-1)^s \left(\sum_{i=0}^{m-1} d_i \beta^{-i} \right) \beta^e$$

with $d_i \in \{0, \dots, \beta - 1\}$, $i = 0, 1, \dots, m-1$, and $e \in \{e_{\min}, \dots, e_{\max}\}$.

- β is called the base,

Floating Point Numbers

In a computer only a finite subset of all real numbers can be represented. These are the so–called floating point numbers and they are of the form

$$\bar{x} = (-1)^s \left(\sum_{i=0}^{m-1} d_i \beta^{-i} \right) \beta^e$$

with $d_i \in \{0, \ldots, \beta - 1\}$, $i = 0, 1, \ldots, m - 1$, and $e \in \{e_{\min}, \ldots, e_{\max}\}$.

- β is called the base,
- $\sum_{i=0}^{m-1} d_i \beta^{-i}$ is the significant or mantissa, m is the *mantissa length*,

Floating Point Numbers

In a computer only a finite subset of all real numbers can be represented. These are the so–called floating point numbers and they are of the form

$$\bar{x} = (-1)^s \left(\sum_{i=0}^{m-1} d_i \beta^{-i} \right) \beta^e$$

with $d_i \in \{0, \dots, \beta - 1\}$, $i = 0, 1, \dots, m-1$, and $e \in \{e_{\min}, \dots, e_{\max}\}$.

- β is called the base,
- $\sum_{i=0}^{m-1} d_i \beta^{-i}$ is the significant or mantissa, m is the *mantissa length*,
- e is the *exponent*, and $\{e_{\min}, \dots, e_{\max}\}$ is the exponent range.

Floating Point Numbers

In a computer only a finite subset of all real numbers can be represented. These are the so–called floating point numbers and they are of the form

$$\bar{x} = (-1)^s \left(\sum_{i=0}^{m-1} d_i \beta^{-i} \right) \beta^e$$

with $d_i \in \{0, \ldots, \beta - 1\}$, $i = 0, 1, \ldots, m-1$, and $e \in \{e_{\min}, \ldots, e_{\max}\}$.

- β is called the base,
- $\sum_{i=0}^{m-1} d_i \beta^{-i}$ is the significant or mantissa, m is the *mantissa length*,
- e is the *exponent*, and $\{e_{\min}, \ldots, e_{\max}\}$ is the exponent range.
- If $\beta = 2$, then we say the floating point number system is a binary system. In this case the d_i's are called bits.

Floating Point Numbers

In a computer only a finite subset of all real numbers can be represented. These are the so–called floating point numbers and they are of the form

$$\bar{x} = (-1)^s \left(\sum_{i=0}^{m-1} d_i \beta^{-i} \right) \beta^e$$

with $d_i \in \{0, \dots, \beta - 1\}$, $i = 0, 1, \dots, m-1$, and $e \in \{e_{\min}, \dots, e_{\max}\}$.

- β is called the base,
- $\sum_{i=0}^{m-1} d_i \beta^{-i}$ is the significant or mantissa, m is the *mantissa length*,
- e is the *exponent*, and $\{e_{\min}, \dots, e_{\max}\}$ is the exponent range.
- If $\beta = 2$, then we say the floating point number system is a binary system. In this case the d_i's are called bits.
- If $\beta = 10$, then we say the floating point number system is a decimal system. In this case the d_i's are called digits.

Floating Point Numbers

In a computer only a finite subset of all real numbers can be represented. These are the so–called floating point numbers and they are of the form

$$\bar{x} = (-1)^s \left(\sum_{i=0}^{m-1} d_i \beta^{-i} \right) \beta^e$$

with $d_i \in \{0, \ldots, \beta - 1\}$, $i = 0, 1, \ldots, m - 1$, and $e \in \{e_{\min}, \ldots, e_{\max}\}$.

- β is called the base,
- $\sum_{i=0}^{m-1} d_i \beta^{-i}$ is the significant or mantissa, m is the *mantissa length*,
- e is the *exponent*, and $\{e_{\min}, \ldots, e_{\max}\}$ is the exponent range.
- If $\beta = 2$, then we say the floating point number system is a binary system. In this case the d_i's are called bits.
- If $\beta = 10$, then we say the floating point number system is a decimal system. In this case the d_i's are called digits.
- A floating point number $\bar{x} \neq 0$ is said to be normalized if $d_0 > 0$.

A Toy Floating Point Number System

Consider the floating point number system
$\beta = 2, m = 3, e_{\min} = -1, e_{\max} = 2.$

A Toy Floating Point Number System

Consider the floating point number system
$\beta = 2, m = 3, e_{\min} = -1, e_{\max} = 2$.
The normalized floating point numbers $\bar{x} \neq 0$ are of the form

$$\bar{x} = \pm 1.d_1 d_2 \times 2^e$$

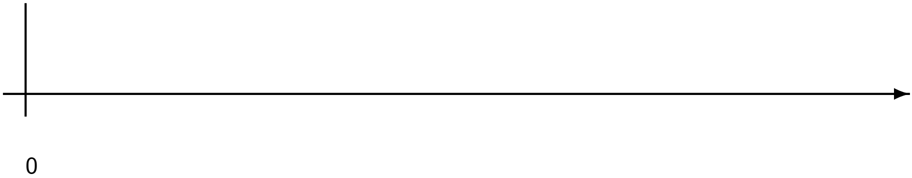
since the normalization condition implies that $d_0 \in \{1, \dots, \beta - 1\} = \{1\}$.

A Toy Floating Point Number System

Consider the floating point number system
$\beta = 2, m = 3, e_{\min} = -1, e_{\max} = 2$.
The normalized floating point numbers $\bar{x} \neq 0$ are of the form

$$\bar{x} = \pm 1.d_1 d_2 \times 2^e$$

since the normalization condition implies that $d_0 \in \{1, \ldots, \beta - 1\} = \{1\}$.

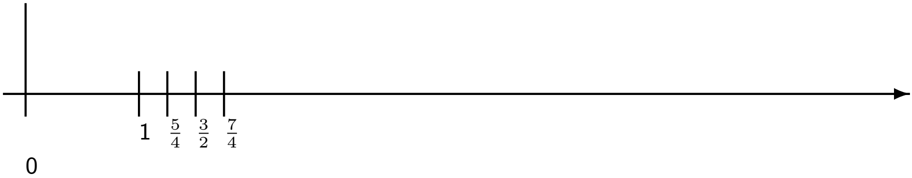


A Toy Floating Point Number System

Consider the floating point number system
$\beta = 2, m = 3, e_{\min} = -1, e_{\max} = 2$.
The normalized floating point numbers $\bar{x} \neq 0$ are of the form

$$\bar{x} = \pm 1.d_1 d_2 \times 2^e$$

since the normalization condition implies that $d_0 \in \{1, \ldots, \beta - 1\} = \{1\}$.



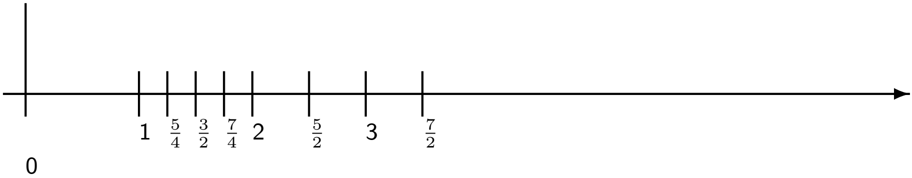
Positive numbers with exponent $e = 0$

A Toy Floating Point Number System

Consider the floating point number system
$\beta = 2, m = 3, e_{\min} = -1, e_{\max} = 2.$
The normalized floating point numbers $\bar{x} \neq 0$ are of the form

$$\bar{x} = \pm 1.d_1 d_2 \times 2^e$$

since the normalization condition implies that $d_0 \in \{1, \ldots, \beta - 1\} = \{1\}$.



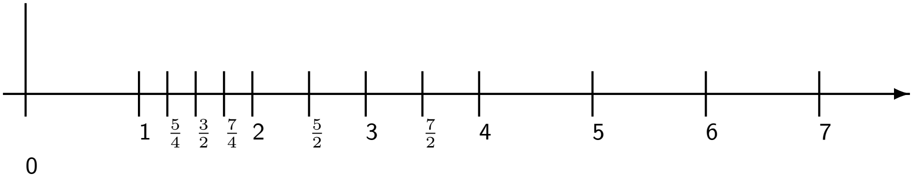
Positive numbers with exponent $e = 0$, 1

A Toy Floating Point Number System

Consider the floating point number system
$\beta = 2, m = 3, e_{\min} = -1, e_{\max} = 2$.
The normalized floating point numbers $\bar{x} \neq 0$ are of the form

$$\bar{x} = \pm 1.d_1 d_2 \times 2^e$$

since the normalization condition implies that $d_0 \in \{1, \dots, \beta - 1\} = \{1\}$.



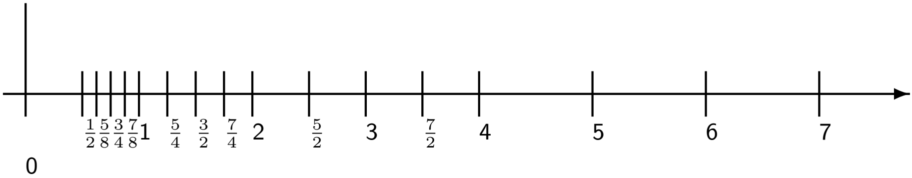
Positive numbers with exponent $e = 0$, 1 , 2

A Toy Floating Point Number System

Consider the floating point number system
$\beta = 2, m = 3, e_{\min} = -1, e_{\max} = 2$.
The normalized floating point numbers $\bar{x} \neq 0$ are of the form

$$\bar{x} = \pm 1.d_1 d_2 \times 2^e$$

since the normalization condition implies that $d_0 \in \{1, \ldots, \beta - 1\} = \{1\}$.



Positive numbers with exponent $e = 0$, 1 , 2 ,-1

Consider the floating point number system

$$\bar{x} = (-1)^s \left(\sum_{i=0}^{m-1} d_i \beta^{-i} \right) \beta^e$$

with $d_i \in \{0, \ldots, \beta - 1\}$, $i = 0, 1, \ldots, m-1$, and $e \in \{e_{\min}, \ldots, e_{\max}\}$.

Consider the floating point number system

$$\bar{x} = (-1)^s \left(\sum_{i=0}^{m-1} d_i \beta^{-i} \right) \beta^e$$

with $d_i \in \{0, \ldots, \beta - 1\}$, $i = 0, 1, \ldots, m - 1$, and $e \in \{e_{\min}, \ldots, e_{\max}\}$.

- The mantissa satisfies

$$\sum_{i=0}^{m-1} d_i \beta^{-i} \leq \sum_{i=0}^{m-1} (\beta - 1) \beta^{-i} = \beta(1 - \beta^{-m}) < \beta.$$

Consider the floating point number system

$$\bar{x} = (-1)^s \left(\sum_{i=0}^{m-1} d_i \beta^{-i} \right) \beta^e$$

with $d_i \in \{0, \ldots, \beta - 1\}$, $i = 0, 1, \ldots, m-1$, and $e \in \{e_{\min}, \ldots, e_{\max}\}$.

- The mantissa satisfies

$$\sum_{i=0}^{m-1} d_i \beta^{-i} \leq \sum_{i=0}^{m-1} (\beta - 1)\beta^{-i} = \beta(1 - \beta^{-m}) < \beta.$$

- The mantissa of a normalized floating point number is always ≥ 1.

Consider the floating point number system

$$\bar{x} = (-1)^s \left(\sum_{i=0}^{m-1} d_i \beta^{-i} \right) \beta^e$$

with $d_i \in \{0, \ldots, \beta - 1\}$, $i = 0, 1, \ldots, m-1$, and $e \in \{e_{\min}, \ldots, e_{\max}\}$.

- The mantissa satisfies

$$\sum_{i=0}^{m-1} d_i \beta^{-i} \leq \sum_{i=0}^{m-1} (\beta - 1)\beta^{-i} = \beta(1 - \beta^{-m}) < \beta.$$

- The mantissa of a normalized floating point number is always ≥ 1.
- The largest floating point number is

$$\bar{x}_{\max} = \left(\sum_{i=0}^{m-1} (\beta - 1)\beta^{-i} \right) \beta^{e_{\max}} = (1 - \beta^{-m})\beta^{e_{\max}+1}.$$

Consider the floating point number system

$$\bar{x} = (-1)^s \left(\sum_{i=0}^{m-1} d_i \beta^{-i} \right) \beta^e$$

with $d_i \in \{0, \ldots, \beta - 1\}$, $i = 0, 1, \ldots, m-1$, and $e \in \{e_{\min}, \ldots, e_{\max}\}$.

- The mantissa satisfies

$$\sum_{i=0}^{m-1} d_i \beta^{-i} \leq \sum_{i=0}^{m-1} (\beta - 1)\beta^{-i} = \beta(1 - \beta^{-m}) < \beta.$$

- The mantissa of a normalized floating point number is always ≥ 1.
- The largest floating point number is

$$\bar{x}_{\max} = \left(\sum_{i=0}^{m-1} (\beta - 1)\beta^{-i} \right) \beta^{e_{\max}} = (1 - \beta^{-m})\beta^{e_{\max}+1}.$$

- The smallest positive normalized floating pt. number is $\bar{x}_{\min} = \beta^{e_{\min}}$.

Consider the floating point number system

$$\bar{x} = (-1)^s \left(\sum_{i=0}^{m-1} d_i \beta^{-i} \right) \beta^e$$

with $d_i \in \{0, \dots, \beta - 1\}$, $i = 0, 1, \dots, m-1$, and $e \in \{e_{\min}, \dots, e_{\max}\}$.

- The mantissa satisfies

$$\sum_{i=0}^{m-1} d_i \beta^{-i} \le \sum_{i=0}^{m-1} (\beta - 1)\beta^{-i} = \beta(1 - \beta^{-m}) < \beta.$$

- The mantissa of a normalized floating point number is always ≥ 1.
- The largest floating point number is

$$\bar{x}_{\max} = \left(\sum_{i=0}^{m-1} (\beta - 1)\beta^{-i} \right) \beta^{e_{\max}} = (1 - \beta^{-m})\beta^{e_{\max}+1}.$$

- The smallest positive normalized floating pt. number is $\bar{x}_{\min} = \beta^{e_{\min}}$.
- The distance between 1 and the next largest floating pt. number is β^{1-m}. Half this number, $\epsilon_{\text{mach}} = \frac{1}{2}\beta^{1-m}$, is called machine precision or unit roundoff. (We will see later why).
  The spacing between the floating pt. numbers in $[1, \beta]$ is $\beta^{-(m-1)}$.
  The spacing between the floating pt. numbers in $[\beta^e, \beta\beta^e]$ is $\beta^{-(m-1)}\beta^e$.

IEEE Floating Point Numbers

- Almost all every modern computer implements the IEEE binary ($\beta = 2$) floating point standard.
- IEEE single precision floating point numbers are stored in 32 bits.
- IEEE double precision floating point numbers are stored in 64 bits.
- How these numbers are stored is quite interesting (clever), but a little too involved to get into here. See the references [G91,O01,SUN] given at the end of this lecture.
- Here are some important numbers.

Common Name	(Approximate) Equivalent Value	
	Single Precision	Double Precision
Unit roundoff	$2^{-24} \approx 6.e-8$	$2^{-53} \approx 1.1e-16$
Maximum normal number	$3.4e+38$	$1.7e+308$
Minimum positive normal number	$1.2e-38$	$2.3e-308$
Maximum subnormal number	$1.1e-38$	$2.2e-308$
Minimum positive subnormal number	$1.5e-45$	$5.0e-324$

Rounding

Given a real number x we define

$$\mathrm{fl}(x) = \text{ normalized floating point number closest to } x.$$

A floating point number $\bar{x}$ closest to x is obtained by rounding. If

$$x = sign(x)\left(\sum_{i=0}^{\infty} d_i\beta^{-i}\right)\beta^e,$$

then

$$\mathrm{fl}(x) = \begin{cases} sign(x)\left(\sum_{i=0}^{m-1} d_i\beta^{-i}\right)\beta^e, & \text{if } d_m < \frac{1}{2}\beta, \\ sign(x)\left(\sum_{i=0}^{m-1} d_i\beta^{-i} \;+\beta^{-(m-1)}\right)\beta^e, & \text{if } d_m \geq \frac{1}{2}\beta. \end{cases}$$

Rounding

Given a real number x we define

$$\text{fl}(x) = \text{ normalized floating point number closest to } x.$$

A floating point number $\bar{x}$ closest to x is obtained by rounding. If

$$x = sign(x) \left(\sum_{i=0}^{\infty} d_i \beta^{-i} \right) \beta^e,$$

then

$$\text{fl}(x) = \begin{cases} sign(x) \left(\sum_{i=0}^{m-1} d_i \beta^{-i} \right) \beta^e, & \text{if } d_m < \frac{1}{2}\beta, \\ sign(x) \left(\sum_{i=0}^{m-1} d_i \beta^{-i} \; + \beta^{-(m-1)} \right) \beta^e, & \text{if } d_m \geq \frac{1}{2}\beta. \end{cases}$$

Example Let $\beta = 10$, $m = 3$. Then

$$\begin{aligned} \text{fl}(1.234 * 10^{-1}) &= 1.23 * 10^{-1}, \\ \text{fl}(1.235 * 10^{-1}) &= 1.24 * 10^{-1}, \\ \text{fl}(1.295 * 10^{-1}) &= 1.30 * 10^{-1}. \end{aligned}$$

Rounding

Given a real number x we define

$$\text{fl}(x) = \text{ normalized floating point number closest to } x.$$

A floating point number $\bar{x}$ closest to x is obtained by rounding. If

$$x = sign(x) \left(\sum_{i=0}^{\infty} d_i \beta^{-i} \right) \beta^e,$$

then

$$\text{fl}(x) = \begin{cases} sign(x) \left(\sum_{i=0}^{m-1} d_i \beta^{-i} \right) \beta^e, & \text{if } d_m < \frac{1}{2}\beta, \\ sign(x) \left(\sum_{i=0}^{m-1} d_i \beta^{-i} + \beta^{-(m-1)} \right) \beta^e, & \text{if } d_m \geq \frac{1}{2}\beta. \end{cases}$$

Example Let $\beta = 10$, $m = 3$. Then

$$\begin{aligned} \text{fl}(1.234 * 10^{-1}) &= 1.23 * 10^{-1}, \\ \text{fl}(1.235 * 10^{-1}) &= 1.24 * 10^{-1}, \\ \text{fl}(1.295 * 10^{-1}) &= 1.30 * 10^{-1}. \end{aligned}$$

Note, there may be two floating point numbers closest to x. $\text{fl}(x)$ picks one of them. For example, let $\beta = 10$, $m = 3$. Then $1.235 - 1.24 = 0.005$, but also $1.235 - 1.23 = 0.005$. See [G91,O01,SUN] for more details on 'breaking' ties.

Rounding Error

Theorem

If x is a number within the range of floating point numbers and $|x| \in [\beta^e, \beta^{e+1})$, then the absolute error between x and the floating point number $\mathsf{fl}(x)$ closest to x is given by

$$|\mathsf{fl}(x) - x| \leq \frac{1}{2}\beta^{e(1-m)}$$

and, provided $x \neq 0$, the relative error is given by

$$\frac{|\mathsf{fl}(x) - x|}{|x|} \leq \frac{1}{2}\beta^{1-m}. \tag{2}$$

The number

$$\epsilon_{\mathsf{mach}} \stackrel{\text{def}}{=} \frac{1}{2}\beta^{1-m}$$

is called machine precision or unit roundoff.

Proof of the theorem:
If $x = 0$, then $\text{fl}(x) = x$ and the assertion follows immediately.

Proof of the theorem:
If $x = 0$, then $\text{fl}(x) = x$ and the assertion follows immediately.
Consider $x > 0$. (The case $x < 0$ can be treated in the same manner.)
Recall that the spacing between the floating point numbers

$$\bar{x} = \left(\sum_{i=0}^{m-1} d_i \beta^{-i} \right) \beta^e \in [\beta^e, \beta^{e+1})$$

is $\beta^{-(m-1)}\beta^e$. Hence if $x \in [\beta^e, \beta^{e+1})$, then the floating point number $\bar{x}$ closest to x satisfies $|\bar{x} - x| \leq \frac{1}{2}\beta^{-(m-1)}\beta^e$. Since $x \geq \beta^e$,

$$\frac{|\bar{x} - x|}{|x|} \leq \frac{1}{2}\beta^{-(m-1)}.$$

Proof of the theorem:
If $x = 0$, then $\mathrm{fl}(x) = x$ and the assertion follows immediately.
Consider $x > 0$. (The case $x < 0$ can be treated in the same manner.)
Recall that the spacing between the floating point numbers

$$\bar{x} = \left(\sum_{i=0}^{m-1} d_i \beta^{-i} \right) \beta^e \in [\beta^e, \beta^{e+1})$$

is $\beta^{-(m-1)}\beta^e$. Hence if $x \in [\beta^e, \beta^{e+1})$, then the floating point number $\bar{x}$ closest to x satisfies $|\bar{x} - x| \leq \frac{1}{2}\beta^{-(m-1)}\beta^e$. Since $x \geq \beta^e$,

$$\frac{|\bar{x} - x|}{|x|} \leq \frac{1}{2}\beta^{-(m-1)}.$$

$\mathrm{fl}(x)$ is a floating point number closest to $x = \left(\sum_{i=0}^{\infty} d_i \beta^{-i} \right) \beta^e$, $d_0 > 0$?

Examples

Let $\beta = 10$, $m = 3$, thus $\epsilon_{\text{mach}} = 5 * 10^{-3}$.

$$
\begin{aligned}
|\text{fl}(1.234 * 10^{-1}) - 1.234 * 10^{-1}| &= 0.0004, \\
\frac{|\text{fl}(1.234 * 10^{-1}) - 1.234 * 10^{-1}|}{1.234 * 10^{-1}} &= \frac{0.0004}{1.234 * 10^{-1}} \approx 3.2 * 10^{-3}, \\
|\text{fl}(1.295 * 10^{-1}) - 1.295 * 10^{-1}| &= 0.0005, \\
\frac{|\text{fl}(1.295 * 10^{-1}) - 1.295 * 10^{-1}|}{1.295 * 10^{-1}} &= \frac{0.0005}{1.295 * 10^{-1}} \approx 3.9 * 10^{-3}.
\end{aligned}
$$

Floating Point Arithmetic

- Let $\Box$ represent one of the elementary operations $+, -, *, /$. If $\bar{x}$ and $\bar{y}$ are floating point numbers, then $\bar{x}\Box\bar{y}$ may not be a floating point number.
  Example: $\beta = 10$, $m = 4$: $1.234 + 2.751 * 10^{-1} = 1.5091$.
  What is the computed value for $\bar{x}\Box\bar{y}$?

Floating Point Arithmetic

- Let $\Box$ represent one of the elementary operations $+, -, *, /$. If $\bar{x}$ and $\bar{y}$ are floating point numbers, then $\bar{x}\Box\bar{y}$ may not be a floating point number.
  Example: $\beta = 10$, $m = 4$: $1.234 + 2.751 * 10^{-1} = 1.5091$.
  What is the computed value for $\bar{x}\Box\bar{y}$?
- In IEEE floating point arithmetic the result of the computation $\bar{x}\Box\bar{y}$ is equal to the floating point number that is nearest to the exact result $\bar{x}\Box\bar{y}$. Therefore we use $\mathrm{fl}(\bar{x}\Box\bar{y})$ to denote the result of the computation $\bar{x}\Box\bar{y}$

Floating Point Arithmetic

- Let $\Box$ represent one of the elementary operations $+, -, *, /$. If $\bar{x}$ and $\bar{y}$ are floating point numbers, then $\bar{x}\Box\bar{y}$ may not be a floating point number.
  Example: $\beta = 10$, $m = 4$: $1.234 + 2.751 * 10^{-1} = 1.5091$.
  What is the computed value for $\bar{x}\Box\bar{y}$?
- In IEEE floating point arithmetic the result of the computation $\bar{x}\Box\bar{y}$ is equal to the floating point number that is nearest to the exact result $\bar{x}\Box\bar{y}$. Therefore we use fl$(\bar{x}\Box\bar{y})$ to denote the result of the computation $\bar{x}\Box\bar{y}$
- Model for the computation of $\bar{x}\Box\bar{y}$, where $\Box$ is one of the elementary operations $+, -, *, /$.
  1. Given *floating point* numbers $\bar{x}$ and $\bar{y}$.
  2. Compute $\bar{x}\Box\bar{y}$ exactly.
  3. Round the exact result $\bar{x}\Box\bar{y}$ to the nearest floating point number and normalize the result.

  Example cont.: $1.234 + 2.751 * 10^{-1} = 1.5091$. Comp. result: 1.509
  The actual implementation of the elementary operations is more sophisticated. For more details see [DG91,O01].

Floating Point Arithmetic (Cont.)

Given two numbers $\bar{x}, \bar{y}$ in *floating point format*, the computed result satisfies

$$\frac{|\text{fl}(\bar{x}\square\bar{y}) - (\bar{x}\square\bar{y})|}{\bar{x}\square\bar{y}} \leq \epsilon_{\text{mach}}.$$

Examples

Consider the floating point system $\beta = 10$ and $m = 4$.

i. $\bar{x} = 2.552 * 10^3$ and $\bar{y} = 2.551 * 10^3$.
$\bar{x} - \bar{y} = 0.001 * 10^3 = 1.000 * 10^0$. In this case $\bar{x} - \bar{y}$ is a floating point number and nothing needs to done; no error occurs in the subtraction of $\bar{x}$, $\bar{y}$.

ii. $\bar{x} = 2.552 * 10^3$ and $\bar{y} = 2.551 * 10^2$.
$\bar{x} - \bar{y} = 2.2969 * 10^3$. This is not a floating point number. The floating point number nearest to $\bar{x} - \bar{y}$ is $\text{fl}(\bar{x} - \bar{y}) = 2.297 * 10^3$.

$$\frac{|\text{fl}(\bar{x} - \bar{y}) - (\bar{x} - \bar{y})|}{|\bar{x} - \bar{y}|} = \frac{|2.297 * 10^3 - 2.2969 * 10^3|}{2.2969 * 10^3} \approx 4.4 * 10^{-5}$$

$$< \epsilon_{\text{mach}} = 5 * 10^{-4}.$$

Floating Point Arithmetic: Cancellation

For the previous result on the error between $\bar{x}\Box\bar{y}$ and the computed fl$(\bar{x}\Box\bar{y})$ only holds if $\bar{x}, \bar{y}$ in floating point format. What happens when we operate with numbers that are not in floating point format?

Example

Consider the floating point system $\beta = 10$ and $m = 4$.
Subtract the numbers $x = 2.5515052 * 10^3$ and $y = 2.5514911 * 10^3$.

1. Compute the floating point numbers $\bar{x}$ and $\bar{y}$ nearest to x and y, respectively: $\bar{x} = 2.552 * 10^3$ and $\bar{y} = 2.551 * 10^3$.
2. Compute $\bar{x} - \bar{y}$ exactly: $\bar{x} - \bar{y} = 0.001 * 10^3$.
3. Round the exact result $\bar{x} - \bar{y}$ to the nearest floating point number: fl$(0.001 * 10^3) = 0.001 * 10^3$. Normalize the number: fl$(0.001 * 10^3) = 1.000$. The last digits are filled with (spurious) zeros.

The exact result is $2.5515052 * 10^3 - 2.5514911 * 10^3 = 1.410 * 10^{-2}$. The relative error between exact and computed solution is

$$\frac{|1.000 - 1.410 * 10^{-2}|}{1.410 * 10^{-2}} \approx 70 \gg \epsilon_{\text{mach}} = 5 * 10^{-4}.$$

Note that this large error is not due the computation of fl$(\bar{x} - \bar{y})$. The large error is caused by the rounding of x and y at the beginning.

Floating Point Arithmetic: Cancellation (cont.)

- To analyze the analyze the error incurred by the subtraction of two numbers, the following representation is useful:
  For every $x \in I\!R$, there exists ϵ with $|\epsilon| \leq \epsilon_{\text{mach}}$ such that

$$\text{fl}(x) = x(1 + \epsilon).$$

  Note that if $x \neq 0$, then the previous identity is satisfied for $\epsilon \stackrel{\text{def}}{=} (\text{fl}(x) - x)/x$. The bound $|\epsilon| \leq \epsilon_{\text{mach}}$ follows from (2).

Floating Point Arithmetic: Cancellation (cont.)

- To analyze the analyze the error incurred by the subtraction of two numbers, the following representation is useful:
  For every $x \in I\!R$, there exists ϵ with $|\epsilon| \leq \epsilon_{\text{mach}}$ such that

  $$\text{fl}(x) = x(1+\epsilon).$$

  Note that if $x \neq 0$, then the previous identity is satisfied for $\epsilon \stackrel{\text{def}}{=} (\text{fl}(x) - x)/x$. The bound $|\epsilon| \leq \epsilon_{\text{mach}}$ follows from (2).
- For $x, y \in I\!R$ we have ϵ_1, ϵ_2 with $|\epsilon_1|, |\epsilon_2| \leq \epsilon_{\text{mach}}$ such that

  $$\text{fl}(x) = x(1+\epsilon_1), \quad \text{fl}(y) = y(1+\epsilon_2).$$

  Moreover $\text{fl}(\text{fl}(x) - \text{fl}(y)) = (\text{fl}(x) - \text{fl}(y))(1+\epsilon_3)$, with $|\epsilon_3| \leq \epsilon_{\text{mach}}$.

Floating Point Arithmetic: Cancellation (cont.)

- To analyze the analyze the error incurred by the subtraction of two numbers, the following representation is useful:
  For every $x \in I\!R$, there exists ϵ with $|\epsilon| \leq \epsilon_{\text{mach}}$ such that
  $$\text{fl}(x) = x(1+\epsilon).$$
  Note that if $x \neq 0$, then the previous identity is satisfied for $\epsilon \stackrel{\text{def}}{=} (\text{fl}(x) - x)/x$. The bound $|\epsilon| \leq \epsilon_{\text{mach}}$ follows from (2).
- For $x, y \in I\!R$ we have ϵ_1, ϵ_2 with $|\epsilon_1|, |\epsilon_2| \leq \epsilon_{\text{mach}}$ such that
  $$\text{fl}(x) = x(1+\epsilon_1), \quad \text{fl}(y) = y(1+\epsilon_2).$$
  Moreover $\text{fl}(\text{fl}(x) - \text{fl}(y)) = (\text{fl}(x) - \text{fl}(y))(1+\epsilon_3)$, with $|\epsilon_3| \leq \epsilon_{\text{mach}}$.
- Thus,
  $$\begin{aligned} \text{fl}(\text{fl}(x) - \text{fl}(y)) &= (\text{fl}(x) - \text{fl}(y))(1+\epsilon_3) = [x(1+\epsilon_1) - y(1+\epsilon_2)](1+\epsilon_3) \\ &= (x-y)(1+\epsilon_3) + (x\epsilon_1 - y\epsilon_2)(1+\epsilon_3) \end{aligned}$$
  and, if $x - y \neq 0$, then the relative error is given by
  $$\frac{|\text{fl}(\text{fl}(x) - \text{fl}(y)) - (x-y)|}{|x-y|} = \left|\epsilon_3 + \frac{x\epsilon_1 - y\epsilon_2}{x-y}(1+\epsilon_3)\right| \qquad (3)$$
  If $\epsilon_1\epsilon_2 \neq 0$ and $x - y$ is small, the quantity on the rhs could be $\gg \epsilon_{\text{mach}}$.

Floating Point Arithmetic: Cancelation (cont.)

- Similar analysis can be carried out for $+, -, *, /$.
- Catastrophic cancelation can only occur with $+, -$.
- Catastrophic cancelation can only occur if one subtracts two numbers which are not both in floating point format and which have the same sign and are of approximately the same size, see (3), or if one adds two numbers which are not both in floating point format, which have opposite sign and their absolute values of approximately the same size.

Floating Point Arithmetic: Cancelation Example 1

Evaluation of $1 - \cos(x)$ near $x = 0$. (All computations were done using single precision Fortran.)

x	$1 - \cos$
0.500000	$0.122417E+00$
0.125000	$0.780231E-02$
$0.312500E-01$	$0.488222E-03$
$0.781250E-02$	$0.305176E-04$
$0.195312E-02$	$0.190735E-05$
$0.488281E-03$	$0.119209E-06$
$0.122070E-03$	$0.$
$0.305176E-04$	$0.$
$0.762939E-05$	$0.$
$0.190735E-05$	$0.$

Since $\cos(0) = 1$ we expect catastrophic cancelation. If $x = 0.122070E-03$, then

$$\begin{aligned} 1 - \cos(x) &\approx 1.0000000000 - 0.99999999254\ldots\ldots \\ &= 0.00000000745\ldots\ldots = 7.45054\ldots\ldots e-09 \\ 1 - \mathrm{fl}(\cos(x)) &\approx 1.000000 - \mathrm{fl}(\underbrace{9.999999}_{7\text{ digits}}9254\ldots\ldots * 10^{-1}) \\ &= 1.000000 - 1.000000 = 0. \end{aligned}$$

Floating Point Arithmetic: Cancelation Example 1 (cont.)

Two alternatives for small $|x|$.

Floating Point Arithmetic: Cancelation Example 1 (cont.)

Two alternatives for small $|x|$.

- Since $\cos^2(x) + \sin^2(x) = 1$ it holds that
  $1 - \cos(x) = \sin^2(x)/(1 + \cos(x))$.
  The formula $\sin^2(x)/(1 + \cos(x))$ avoids subtraction of two number that are not in floating point format and are almost the same (recall that we consider the case $|x|$ small).

Floating Point Arithmetic: Cancelation Example 1 (cont.)

Two alternatives for small $|x|$.

- Since $\cos^2(x) + \sin^2(x) = 1$ it holds that $1 - \cos(x) = \sin^2(x)/(1 + \cos(x))$.
  The formula $\sin^2(x)/(1 + \cos(x))$ avoids subtraction of two number that are not in floating point format and are almost the same (recall that we consider the case $|x|$ small).
- The Leibnitz criterion says that if the series $S = \sum_{i=1}^{\infty} (-1)^i c_i$, $c_i \geq 0$, converges, then $\left|S - \sum_{i=1}^{n} (-1)^i c_i\right| < c_{n+1}$.
  If we apply this to the Taylor expansion of $cos(x)$,

$$\cos(x) = 1 - \frac{x^2}{2} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} \pm \ldots,$$

  then

$$\left|\cos(x) - \left(1 - \frac{x^2}{2} + \frac{x^4}{4!} - \frac{x^6}{6!}\right)\right| < \frac{x^8}{8!}.$$

  After some rearrangements we can use the approximation

$$1 - \cos(x) \approx \frac{x^2}{2}\left(1 - \frac{x^2}{12} + \frac{x^4}{360}\right)$$

  and we know that the difference is less than $x^8/(8!)$ which allows us to control the error of the approximation.

Floating Point Arithmetic: Cancelation Example 1 (cont.)

x	$1 - \cos$	$\sin^2/(1+\cos)$	Taylor
0.500000	0.122417	0.122417	0.122418
0.125000	$0.780231E-02$	$0.780233E-02$	$0.780233E-02$
$0.312500E-01$	$0.488222E-03$	$0.488241E-03$	$0.488242E-03$
$0.781250E-02$	$0.305176E-04$	$0.305174E-04$	$0.305174E-04$
$0.195312E-02$	$0.190735E-05$	$0.190735E-05$	$0.190735E-05$
$0.488281E-03$	$0.119209E-06$	$0.119209E-06$	$0.119209E-06$
$0.122070E-03$	$0.$	$0.745058E-08$	$0.745058E-08$
$0.305176E-04$	$0.$	$0.465661E-09$	$0.465661E-09$
$0.762939E-05$	$0.$	$0.291038E-10$	$0.291038E-10$
$0.190735E-05$	$0.$	$0.181899E-11$	$0.181899E-11$
$0.476837E-06$	$0.$	$0.113687E-12$	$0.113687E-12$
$0.119209E-06$	$0.$	$0.710543E-14$	$0.710543E-14$
$0.298023E-07$	$0.$	$0.444089E-15$	$0.444089E-15$

Computations were performed using single precision Fortran.

Floating Point Arithmetic: Cancelation Example 2

- The roots of the quadratic equation $ax^2 + bx + c = 0$ are given by

$$x_{\pm} = \left(-b \pm \sqrt{b^2 - 4ac}\right) / (2a).$$

Floating Point Arithmetic: Cancelation Example 2

- The roots of the quadratic equation $ax^2 + bx + c = 0$ are given by

$$x_{\pm} = \left(-b \pm \sqrt{b^2 - 4ac}\right)/(2a).$$

- When $a = 5 * 10^{-4}, b = 100$, and $c = 5 * 10^{-3}$ the computed (using single precision Fortran) first root is

$$x_{+} = 0.$$

  Cannot be exact, since $x = 0$ is a solution of the quadratic equation if and only if $c = 0$.

Floating Point Arithmetic: Cancelation Example 2

- The roots of the quadratic equation $ax^2 + bx + c = 0$ are given by
$$x_\pm = \left(-b \pm \sqrt{b^2 - 4ac}\right)/(2a).$$
- When $a = 5 * 10^{-4}, b = 100$, and $c = 5 * 10^{-3}$ the computed (using single precision Fortran) first root is
$$x_+ = 0.$$
Cannot be exact, since $x = 0$ is a solution of the quadratic equation if and only if $c = 0$.
- Since $\mathrm{fl}(b^2 - 4ac) = \mathrm{fl}(b^2)$ for the data given above, we suffer from catastrophic cancellation.

Floating Point Arithmetic: Cancelation Example 2

- The roots of the quadratic equation $ax^2 + bx + c = 0$ are given by

$$x_\pm = \left(-b \pm \sqrt{b^2 - 4ac}\right)/(2a).$$

- When $a = 5 * 10^{-4}, b = 100$, and $c = 5 * 10^{-3}$ the computed (using single precision Fortran) first root is

$$x_+ = 0.$$

  Cannot be exact, since $x = 0$ is a solution of the quadratic equation if and only if $c = 0$.

- Since $\mathrm{fl}(b^2 - 4ac) = \mathrm{fl}(b^2)$ for the data given above, we suffer from catastrophic cancellation.
- A remedy is the following reformulation of the formula for x_+:

$$\frac{-b + \sqrt{b^2 - 4ac}}{2a} = \frac{1}{2a} \frac{\left(-b + \sqrt{b^2 - 4ac}\right)\left(-b - \sqrt{b^2 - 4ac}\right)}{-b - \sqrt{b^2 - 4ac}} = \frac{2c}{-b - \sqrt{b^2 - 4ac}}$$

  Here the subtraction of two almost equal numbers is avoided and the computation using this formula gives $x_+ = -0.5E - 04$.

Floating Point Arithmetic: Cancelation Example 2

- The roots of the quadratic equation $ax^2 + bx + c = 0$ are given by
$$x_\pm = \left(-b \pm \sqrt{b^2 - 4ac}\right)/(2a).$$
- When $a = 5 * 10^{-4}, b = 100$, and $c = 5 * 10^{-3}$ the computed (using single precision Fortran) first root is
$$x_+ = 0.$$
Cannot be exact, since $x = 0$ is a solution of the quadratic equation if and only if $c = 0$.
- Since $\mathrm{fl}(b^2 - 4ac) = \mathrm{fl}(b^2)$ for the data given above, we suffer from catastrophic cancellation.
- A remedy is the following reformulation of the formula for x_+:
$$\frac{-b+\sqrt{b^2-4ac}}{2a} = \frac{1}{2a}\frac{\left(-b+\sqrt{b^2-4ac}\right)\left(-b-\sqrt{b^2-4ac}\right)}{-b-\sqrt{b^2-4ac}} = \frac{2c}{-b-\sqrt{b^2-4ac}}$$
Here the subtraction of two almost equal numbers is avoided and the computation using this formula gives $x_+ = -0.5E-04$.
- A 'stable' (see later for a description of stability) formula for both roots
$$x_1 = \left(-b - \mathrm{sign}(b)\sqrt{b^2-4ac}\right)/(2a), \quad x_2 = c/(ax_1).$$

Summary

- Introduced how numbers are represented on a computer.
- Only a small set of numbers can be represented on the computer.
- The relative error between $x \neq 0$ and its nearest floating point number $\mathrm{fl}(x)$ is

$$\frac{|\mathrm{fl}(x) - x|}{|x|} \leq \epsilon_{\mathrm{mach}} \stackrel{\mathrm{def}}{=} \frac{1}{2}\beta^{1-m}.$$

- Introduced basic properties of floating point arithmetic.
- Catastrophic cancellation can occur if one subtracts [adds] two numbers which are not both in floating point format and which have the same [opposite] sign and [their absolute values] are of approximately the same size.

Additional Reading

G91 David Goldberg. What every computer scientist should know about floating-point arithmetic, *ACM Comput. Surv.*, Vol. 23 (1), 1991, pp. 5 - 48.
http://docs.sun.com/source/806-3568/ncg_goldberg.html

O01 Michael L. Overton. Numerical Computing with IEEE Floating Point Arithmetic, SIAM, Philadelphia, 2001.

SUN SUN Microsystems Numerical Computation Guide
http://docs.sun.com/source/806-3568/