Machine Numbers

When performing arithmetic on a computer (laptop, desktop, mainframe, cell phone, etc.) we will primarily work with two types of numbers:

Integers: whole numbers in a specified range.

Floating point: approximations to real numbers.
2’s Complement Integers

Suppose an integer is represented using $n$ bits (e.g., $n = 32$) bits in 2’s complement format.

- Index the bits from right to left.

<table>
<thead>
<tr>
<th>Bit</th>
<th>31</th>
<th>30</th>
<th>29</th>
<th>...</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$-(2^{31})$</td>
<td>$2^{30}$</td>
<td>$2^{29}$</td>
<td>...</td>
<td>$2^2$</td>
<td>$2^1$</td>
<td>$2^0$</td>
</tr>
</tbody>
</table>

- If the $i$th bit is set, the quantity in the $i$th column is added.
Example (1 of 2)

<table>
<thead>
<tr>
<th>Sign</th>
<th>30</th>
<th>29</th>
<th>...</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0000000000000000000000000100</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ x = 2^5 + 2^1 + 2^0 = 32 + 2 + 1 = 35 \]
Example (2 of 2)

<table>
<thead>
<tr>
<th>Sign</th>
<th>30</th>
<th>29</th>
<th>...</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>00000000000000000000000000100</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\[
x = -(2^{31}) + 2^{29} + 2^5 + 2^1 + 2^0 \\
   = -2147483648 + 536870912 + 32 + 2 + 1 \\
   = -1610612701
\]
Range of Integers

1. What are the smallest and largest integers which can be represented in 32-bit 2’s complement format?
Range of Integers

1. What are the smallest and largest integers which can be represented in 32-bit 2’s complement format?

\[
\begin{align*}
\text{min}_{32} &= -(2^{31}) = -2147483648 \\
\text{max}_{32} &= \sum_{i=0}^{30} 2^i = 2147483647
\end{align*}
\]
Range of Integers

1. What are the smallest and largest integers which can be represented in 32-bit 2’s complement format?

\[
\begin{align*}
\min_{32} &= -(2^{31}) = -2147483648 \\
\max_{32} &= \sum_{i=0}^{30} 2^i = 2147483647
\end{align*}
\]

2. What are the smallest and largest integers which can be represented in 64-bit 2’s complement format?
Range of Integers

1. What are the smallest and largest integers which can be represented in 32-bit 2’s complement format?

\[
\begin{align*}
\text{min}_{32} &= - (2^{31}) = -2147483648 \\
\text{max}_{32} &= \sum_{i=0}^{30} 2^i = 2147483647
\end{align*}
\]

2. What are the smallest and largest integers which can be represented in 64-bit 2’s complement format?

\[
\begin{align*}
\text{min}_{64} &= - (2^{63}) = -9223372036854775808 \\
\text{max}_{64} &= \sum_{i=0}^{62} 2^i = 9223372036854775807
\end{align*}
\]
Floating Point Numbers

Consider representing a real number like \( \pi \) in some binary format.
Floating Point Numbers

Consider representing a real number like π in some binary format.

- π is transcendental (non-repeating, non-terminating decimal number)
Floating Point Numbers

Consider representing a real number like \(\pi\) in some binary format.

- \(\pi\) is transcendental (non-repeating, non-terminating decimal number)
- In any finite number of binary digits, \(\pi\) can only be approximated by some rational number. There will be round-off error.
Floating Point Numbers

Consider representing a real number like $\pi$ in some binary format.

- $\pi$ is transcendental (non-repeating, non-terminating decimal number)
- In any finite number of binary digits, $\pi$ can only be approximated by some rational number. There will be round-off error.
- Round-off error will be present when representing any real number which is not a power of 2.
The Institute for Electrical and Electronic Engineers (IEEE) published the *Binary Floating Point Arithmetic Standard 754-2008* which specified storage and transmission formats for floating point numbers and algorithms for rounding arithmetic operations. Consider the 64-bit representation.

**s:** sign bit

**c:** characteristic, 11-bit exponent with base 2, according to IEEE 754-2008, $1 \leq c \leq 2046$ always

**f:** mantissa, 52-bit binary fraction

\[ x = (-1)^s 2^{c-1023} (1 + f) \]
This number is positive since \((-1)^0 = 1\).

\[
\begin{align*}
c &= 2^{10} + 2^2 + 2^0 = 1024 + 4 + 1 = 1029 \\
f &= \left(\frac{1}{2}\right)^1 + \left(\frac{1}{2}\right)^4 + \left(\frac{1}{2}\right)^5 + \left(\frac{1}{2}\right)^6 + \left(\frac{1}{2}\right)^7 + \left(\frac{1}{2}\right)^9 = \frac{317}{512} \\
x &= (-1)^0 \cdot 2^{1029-1023} \left(1 + \frac{317}{512}\right) = \frac{829}{8} \\
  &= 103.625000000000000000000
\end{align*}
\]
Interval of Real Numbers

The previous example actually represents an interval of numbers. Consider the next smaller and next larger floating point numbers.

<table>
<thead>
<tr>
<th>$s$</th>
<th>$c$</th>
<th>$f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10000000101</td>
<td>100111100111111</td>
</tr>
<tr>
<td>0</td>
<td>10000000101</td>
<td>100111101000000</td>
</tr>
<tr>
<td>0</td>
<td>10000000101</td>
<td>100111101000001</td>
</tr>
</tbody>
</table>

$x_s = 103.6249999999998579$

$x = 103.6250000000000000$

$x_l = 103.6250000000001421$
Interval of Real Numbers

The previous example actually represents an interval of numbers. Consider the next smaller and next larger floating point numbers.

<table>
<thead>
<tr>
<th>s</th>
<th>c</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>100000000101</td>
<td>100111100111\cdots111</td>
</tr>
<tr>
<td>0</td>
<td>100000000101</td>
<td>100111101000\cdots000</td>
</tr>
<tr>
<td>0</td>
<td>100000000101</td>
<td>100111101000\cdots001</td>
</tr>
</tbody>
</table>

\[ x_s = 103.624999999999998579 \]
\[ x = 103.625000000000000000 \]
\[ x_I = 103.625000000000001421 \]

Upon rounding \( x \) represents all real numbers in the interval \[ (103.6249999999999289, 103.62500000000000711). \]
Floating Point Limits

- Smallest, non-zero positive floating point number:
  \[ \epsilon = 2^{-1022} \]
  \[ = 1.17 	imes 10^{-38} \]

- Largest floating point number:
  \[ 2^{1023} - 1 \]
  \[ = 1.79 	imes 10^{308} \]
Floating Point Limits

- Smallest, non-zero positive floating point number:
  \[ \epsilon = (-1)^0 \times 2^{1-1023} \times (1 + 0) \approx 0.22251 \times 10^{-307} \]

- Largest floating point number:
Floating Point Limits

- Smallest, non-zero positive floating point number:
  \[ \epsilon = (-1)^0 2^{1-1023} (1 + 0) \approx 0.22251 \times 10^{-307} \]

- Largest floating point number:
  \[ \infty = (-1)^0 2^{2046-1023} (1 + 1 - 2^{-52}) \approx 0.17977 \times 10^{309} \]
Decimal (Base-10) Floating Point Numbers

- We will express floating point numbers in base-10 form for simplicity.
- If $x$ is a non-zero real number, then $x$ can be represented as
  \[ \pm 0.d_1 d_2 \ldots d_{k-1} d_k d_{k+1} \ldots \times 10^n \]
  where $1 \leq d_1 \leq 9$ and $0 \leq d_k \leq 9$ for $k > 1$. 
Decimal (Base-10) Floating Point Numbers

- We will express floating point numbers in base-10 form for simplicity.

- If \( x \) is a non-zero real number, then \( x \) can be represented as

\[
\pm 0.d_1 d_2 \ldots d_{k-1} d_k d_{k+1} \ldots \times 10^n
\]

where 1 \( \leq \) \( d_1 \) \( \leq \) 9 and 0 \( \leq \) \( d_k \) \( \leq \) 9 for \( k > 1 \).

- The \( k \)-digit decimal machine number corresponding to \( x \) will be denoted \( fl (x) \) and is determined by rounding.
  
  - To round we may perform chopping by ignoring all the decimal digits beyond the \( k \)th,

\[
fl (0.d_1 d_2 \ldots d_{k-1} d_k d_{k+1} \ldots \times 10^n) = 0.d_1 d_2 \ldots d_{k-1} d_k \times 10^n
\]

  - or we may perform rounding in the \( k \)th decimal place.

\[
fl (0.d_1 d_2 \ldots d_{k-1} d_k d_{k+1} \ldots \times 10^n) = 0.\delta_1 \delta_2 \ldots \delta_{k-1} \delta_k \times 10^n
\]
Example

Determine the 5-digit representations for $e$ using

1. chopping,
2. rounding.
Determine the 5-digit representations for $e$ using

$$e \approx 0.2718281828 \times 10^1$$

1. chopping,
2. rounding.
Example

Determine the 5-digit representations for $e$ using

$$ e \approx 0.2718281828 \times 10^1 $$

1. chopping, $fl(e) = 0.27182 \times 10^1$
2. rounding.
Example

Determine the 5-digit representations for $e$ using

$$e \approx 0.2718281828 \times 10^1$$

1. chopping, $fl(e) = 0.27182 \times 10^1$
2. rounding. $fl(e) = 0.27183 \times 10^1$
Definition
Suppose that \( \hat{p} \) is an approximation to \( p \).

- The **actual error** is \( p - \hat{p} \).
- The **absolute error** is \( |p - \hat{p}| \).
- The **relative error** is \( \frac{|p - \hat{p}|}{|p|} \) provided \( p \neq 0 \).
Approximation Errors

Definition
Suppose that $\hat{p}$ is an approximation to $p$.

- The **actual error** is $p - \hat{p}$.
- The **absolute error** is $|p - \hat{p}|$.
- The **relative error** is $\frac{|p - \hat{p}|}{|p|}$ provided $p \neq 0$.

**Remark:** the relative error is generally preferred as a measure of accuracy, since it takes into consideration, the magnitude of the number being approximated.
Example

Determine the absolute and relative error present in each of the 5-digit approximations to $e$. 

- **Chopping** 
  - $\hat{e} = 2.7182$: 
    - $|e - \hat{e}| \approx 0.18285 \times 10^{-5}$ and $|e - \hat{e}|/e \approx 3.0103 \times 10^{-5}$

- **Rounding** 
  - $\hat{e} = 2.7183$: 
    - $|e - \hat{e}| \approx 1.81715 \times 10^{-5}$ and $|e - \hat{e}|/e \approx 6.68494 \times 10^{-6}$
Determine the absolute and relative error present in each of the 5-digit approximations to $e$.

- **Chopping $\hat{e} = 2.7182$:**
  
  $$|e - \hat{e}| \approx 8.18285 \times 10^{-5} \quad \text{and} \quad \frac{|e - \hat{e}|}{e} \approx 3.0103 \times 10^{-5}$$

- **Rounding $\hat{e} = 2.7183$:**
  
  $$|e - \hat{e}| \approx 1.81715 \times 10^{-5} \quad \text{and} \quad \frac{|e - \hat{e}|}{e} \approx 6.68494 \times 10^{-6}$$
Definition
The number \( \hat{p} \) is said to approximate \( p \) to \( t \) **significant digits** if \( t \) is the largest non-negative integer for which

\[
\frac{|p - \hat{p}|}{|p|} \leq 5 \times 10^{-t}.
\]
Example

Suppose that $\hat{p}$ agrees with $p$ to 5 significant digits.

<table>
<thead>
<tr>
<th>$p$</th>
<th>$\hat{p}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>(0.0099995, 0.0100005)</td>
</tr>
<tr>
<td>0.1</td>
<td>(0.099995, 0.100005)</td>
</tr>
<tr>
<td>1</td>
<td>(0.99995, 1.00005)</td>
</tr>
<tr>
<td>10</td>
<td>(9.9995, 10.0005)</td>
</tr>
<tr>
<td>100</td>
<td>(99.995, 100.005)</td>
</tr>
<tr>
<td>1000</td>
<td>(999.95, 1000.05)</td>
</tr>
</tbody>
</table>
Chopping and Relative Error

\[
\left| \frac{x - fl(x)}{x} \right| = \left| \frac{0.d_1 d_2 \ldots d_{k-1} d_k d_{k+1} \ldots \times 10^n - 0.d_1 d_2 \ldots d_{k-1} d_k \times 10^n}{0.d_1 d_2 \ldots d_{k-1} d_k d_{k+1} \ldots \times 10^n} \right|
\]

\[
= \left| \frac{0.d_{k+1} \ldots \times 10^{n-k}}{0.d_1 d_2 \ldots d_{k-1} d_k d_{k+1} \ldots \times 10^n} \right|
\]

\[
= \left| \frac{0.d_{k+1} \ldots}{0.d_1 d_2 \ldots d_{k-1} d_k d_{k+1} \ldots} \right| \times 10^{-k}
\]

\[
\leq 10 \times 10^{-k} = 10^{-k+1}
\]
The performance of basic arithmetic operations on a computing device also results in approximations.

We will define the following machine arithmetic operations:

\[
\begin{align*}
x \oplus y &= \text{fl}(\text{fl}(x) + \text{fl}(y)) \\
x \ominus y &= \text{fl}(\text{fl}(x) - \text{fl}(y)) \\
x \otimes y &= \text{fl}(\text{fl}(x) \times \text{fl}(y)) \\
x \odot y &= \text{fl}(\text{fl}(x) \div \text{fl}(y))
\end{align*}
\]
Example

Using 5-digit chopping arithmetic with \( x = \frac{2}{3} \) and \( y = \frac{3}{7} \), compute the following quantities and the errors involved in their calculation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( x \oplus y )</td>
<td>0.10952 \times 10^1</td>
<td>0.23</td>
<td>5.283 \times 10^{-1}</td>
<td>0.023</td>
</tr>
<tr>
<td>( x \ominus y )</td>
<td>0.80952 \times 10^0</td>
<td>0.23</td>
<td>5.283 \times 10^{-1}</td>
<td>0.023</td>
</tr>
<tr>
<td>( x \otimes y )</td>
<td>0.47826 \times 10^1</td>
<td>0.3</td>
<td>4.783 \times 10^{-1}</td>
<td>0.014</td>
</tr>
<tr>
<td>( x \oslash y )</td>
<td>0.23809 \times 10^0</td>
<td>0.5</td>
<td>1.371 \times 10^{-1}</td>
<td>0.026</td>
</tr>
<tr>
<td>( x \cratch y )</td>
<td>0.28571 \times 10^1</td>
<td>0.5</td>
<td>1.371 \times 10^{-1}</td>
<td>0.026</td>
</tr>
<tr>
<td>( x \div y )</td>
<td>0.15555 \times 10^1</td>
<td>1.14</td>
<td>1.144 \times 10^{-1}</td>
<td>0.096</td>
</tr>
</tbody>
</table>

Remark: the relative errors are small, so the machines results can be trusted in these cases.
Example

Using 5-digit chopping arithmetic with \( x = \frac{2}{3} \) and \( y = \frac{3}{7} \), compute the following quantities and the errors involved in their calculation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( x \oplus y )</td>
<td>( 0.10952 \times 10^{1} )</td>
<td>( \frac{23}{21} )</td>
<td>( 3.80952 \times 10^{-5} )</td>
<td>( 3.47826 \times 10^{-5} )</td>
</tr>
<tr>
<td>( x \ominus y )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( x \otimes y )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( x \oslash y )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Remark: the relative errors are small, so the machines results can be trusted in these cases.
Using 5-digit chopping arithmetic with $x = 2/3$ and $y = 3/7$, compute the following quantities and the errors involved in their calculation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$x \oplus y$</td>
<td>$0.10952 \times 10^1$</td>
<td>$\frac{23}{21}$</td>
<td>$3.80952 \times 10^{-5}$</td>
<td>$3.47826 \times 10^{-5}$</td>
</tr>
<tr>
<td>$x \ominus y$</td>
<td>$0.23809 \times 10^0$</td>
<td>$\frac{5}{21}$</td>
<td>$5.2381 \times 10^{-6}$</td>
<td>$2.2 \times 10^{-5}$</td>
</tr>
<tr>
<td>$x \otimes y$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x \oslash y$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

Using 5-digit chopping arithmetic with $x = 2/3$ and $y = 3/7$, compute the following quantities and the errors involved in their calculation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$x \oplus y$</td>
<td>$0.10952 \times 10^1$</td>
<td>$\frac{23}{21}$</td>
<td>$3.80952 \times 10^{-5}$</td>
<td>$3.47826 \times 10^{-5}$</td>
</tr>
<tr>
<td>$x \ominus y$</td>
<td>$0.23809 \times 10^0$</td>
<td>$\frac{5}{21}$</td>
<td>$5.2381 \times 10^{-6}$</td>
<td>$2.2 \times 10^{-5}$</td>
</tr>
<tr>
<td>$x \otimes y$</td>
<td>$0.28571 \times 10^0$</td>
<td>$\frac{4}{7}$</td>
<td>$4.28571 \times 10^{-6}$</td>
<td>$1.5 \times 10^{-5}$</td>
</tr>
<tr>
<td>$x \oslash y$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Remark: the relative errors are small, so the machines results can be trusted in these cases.
Example

Using 5-digit chopping arithmetic with $x = \frac{2}{3}$ and $y = \frac{3}{7}$, compute the following quantities and the errors involved in their calculation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$x \oplus y$</td>
<td>$0.10952 \times 10^1$</td>
<td>$\frac{23}{21}$</td>
<td>$3.80952 \times 10^{-5}$</td>
<td>$3.47826 \times 10^{-5}$</td>
</tr>
<tr>
<td>$x \ominus y$</td>
<td>$0.23809 \times 10^0$</td>
<td>$\frac{5}{21}$</td>
<td>$5.2381 \times 10^{-6}$</td>
<td>$2.2 \times 10^{-5}$</td>
</tr>
<tr>
<td>$x \otimes y$</td>
<td>$0.28571 \times 10^0$</td>
<td>$\frac{2}{7}$</td>
<td>$4.28571 \times 10^{-6}$</td>
<td>$1.5 \times 10^{-5}$</td>
</tr>
<tr>
<td>$x \oslash y$</td>
<td>$0.15555 \times 10^1$</td>
<td>$\frac{14}{9}$</td>
<td>$5.55556 \times 10^{-5}$</td>
<td>$3.57143 \times 10^{-5}$</td>
</tr>
</tbody>
</table>

Remark: the relative errors are small, so the machines results can be trusted in these cases.
Example

Using 5-digit chopping arithmetic with $x = 2/3$ and $y = 3/7$, compute the following quantities and the errors involved in their calculation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$x \oplus y$</td>
<td>$0.10952 \times 10^1$</td>
<td>$\frac{23}{21}$</td>
<td>$3.80952 \times 10^{-5}$</td>
<td>$3.47826 \times 10^{-5}$</td>
</tr>
<tr>
<td>$x \ominus y$</td>
<td>$0.23809 \times 10^0$</td>
<td>$\frac{5}{21}$</td>
<td>$5.2381 \times 10^{-6}$</td>
<td>$2.2 \times 10^{-5}$</td>
</tr>
<tr>
<td>$x \otimes y$</td>
<td>$0.28571 \times 10^0$</td>
<td>$\frac{7}{4}$</td>
<td>$4.28571 \times 10^{-6}$</td>
<td>$1.5 \times 10^{-5}$</td>
</tr>
<tr>
<td>$x \oslash y$</td>
<td>$0.15555 \times 10^1$</td>
<td>$\frac{14}{9}$</td>
<td>$5.55556 \times 10^{-5}$</td>
<td>$3.57143 \times 10^{-5}$</td>
</tr>
</tbody>
</table>

Remark: the relative errors are small, so the machines results can be trusted in these cases.
Example

Suppose $y = 3/7$, $v = 0.428551$, and $w = 0.123 \times 10^{-4}$. Using 5-digit chopping arithmetic compute the following results and the associated errors.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$y \ominus v$</td>
<td>0.204286</td>
<td>0.2042851</td>
<td>0</td>
<td>0.000004</td>
</tr>
<tr>
<td>$(y \ominus v) \ominus w$</td>
<td>0.16261</td>
<td>0.1626086</td>
<td>0.000004</td>
<td>0.000024</td>
</tr>
</tbody>
</table>

Remark: in this example the subtraction of nearly equal quantities leads to larger relative errors.
Example

Suppose $y = 3/7$, $v = 0.428551$, and $w = 0.123 \times 10^{-4}$. Using 5-digit chopping arithmetic compute the following results and the associated errors.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$y \ominus v$</td>
<td>$0.2 \times 10^{-4}$</td>
<td>$0.204286 \times 10^{-4}$</td>
<td>$4.286 \times 10^{-7}$</td>
<td>$2.09804 \times 10^{-2}$</td>
</tr>
<tr>
<td>$(y \ominus v) \ominus w$</td>
<td>$0.1626 \times 10^{-4}$</td>
<td>$0.166086 \times 10^{-4}$</td>
<td>$0.486 \times 10^{-7}$</td>
<td>$2.09891 \times 10^{-2}$</td>
</tr>
</tbody>
</table>

Remark: in this example the subtraction of nearly equal quantities leads to larger relative errors.
Example

Suppose $y = 3/7$, $v = 0.428551$, and $w = 0.123 \times 10^{-4}$. Using 5-digit chopping arithmetic compute the following results and the associated errors.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$y \oplus v$</td>
<td>$0.2 \times 10^{-4}$</td>
<td>$0.204286 \times 10^{-4}$</td>
<td>$4.286 \times 10^{-7}$</td>
<td>$2.09804 \times 10^{-2}$</td>
</tr>
<tr>
<td>$(y \oplus v) \ominus w$</td>
<td>$0.1626 \times 10^1$</td>
<td>$0.166086 \times 10^1$</td>
<td>$3.486 \times 10^{-2}$</td>
<td>$2.09891 \times 10^{-2}$</td>
</tr>
</tbody>
</table>

Remark: In this example the subtraction of nearly equal quantities leads to larger relative errors.
Example

Suppose $y = 3/7$, $v = 0.428551$, and $w = 0.123 \times 10^{-4}$. Using 5-digit chopping arithmetic compute the following results and the associated errors.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$y \ominus v$</td>
<td>$0.2 \times 10^{-4}$</td>
<td>$0.204286 \times 10^{-4}$</td>
<td>$4.286 \times 10^{-7}$</td>
<td>$2.09804 \times 10^{-2}$</td>
</tr>
<tr>
<td>$(y \ominus v) \ominus w$</td>
<td>$0.1626 \times 10^1$</td>
<td>$0.166086 \times 10^1$</td>
<td>$3.486 \times 10^{-2}$</td>
<td>$2.09891 \times 10^{-2}$</td>
</tr>
</tbody>
</table>

**Remark:** in this example the subtraction of nearly equal quantities leads to larger relative errors.
Subtraction of Nearly Equal Quantities

Suppose \( x > y \) and the \( k \)-digit representations of \( x \) and \( y \) are respectively

\[
fl(x) = 0.d_1d_2\ldots d_0a_{p+1}a_{p+2}\ldots a_k \times 10^n \\
fl(y) = 0.d_1d_2\ldots d_0b_{p+1}b_{p+2}\ldots b_k \times 10^n.
\]
Subtraction of Nearly Equal Quantities

Suppose \( x > y \) and the \( k \)-digit representations of \( x \) and \( y \) are respectively

\[
\text{fl}(x) = 0.d_1d_2\ldots d_p a_{p+1}a_{p+2}\ldots a_k \times 10^n
\]
\[
\text{fl}(y) = 0.d_1d_2\ldots d_p b_{p+1}b_{p+2}\ldots b_k \times 10^n.
\]

Since the first \( p \) decimal digits of \( x \) and \( y \) are the same, then

\[
x \ominus y = 0.c_{p+1}c_{p+2}\ldots c_k \times 10^{n-p}
\]

where

\[
0.c_{p+1}c_{p+2}\ldots c_k = 0.a_{p+1}a_{p+2}\ldots a_k - 0.b_{p+1}b_{p+2}\ldots b_k.
\]
Subtraction of Nearly Equal Quantities

Suppose \( x > y \) and the \( k \)-digit representations of \( x \) and \( y \) are respectively

\[
fl(x) = 0.d_1d_2 \ldots d_pa_{p+1}a_{p+2} \ldots a_k \times 10^n
\]
\[
fl(y) = 0.d_1d_2 \ldots d_pb_{p+1}b_{p+2} \ldots b_k \times 10^n.
\]

Since the first \( p \) decimal digits of \( x \) and \( y \) are the same, then

\[
x \ominus y = 0.c_{p+1}c_{p+2} \ldots c_k \times 10^{n-p}
\]

where

\[
0.c_{p+1}c_{p+2} \ldots c_k = 0.a_{p+1}a_{p+2} \ldots a_k - 0.b_{p+1}b_{p+2} \ldots b_k.
\]

**Remark:** the result \( x \ominus y \) has at most \( k \) digits of significance.
Subtraction of Nearly Equal Quantities

Suppose $x > y$ and the $k$-digit representations of $x$ and $y$ are respectively

$$fl(x) = 0.d_1d_2 \ldots d_pa_{p+1}a_{p+2} \ldots a_k \times 10^n$$

$$fl(y) = 0.d_1d_2 \ldots d_pb_{p+1}b_{p+2} \ldots b_k \times 10^n.$$ 

Since the first $p$ decimal digits of $x$ and $y$ are the same, then

$$x \ominus y = 0.c_{p+1}c_{p+2} \ldots c_k \times 10^{n-p}$$

where

$$0.c_{p+1}c_{p+2} \ldots c_k = 0.a_{p+1}a_{p+2} \ldots a_k - 0.b_{p+1}b_{p+2} \ldots b_k.$$ 

**Remark:** the result $x \ominus y$ has at most $k - p$ digits of significance.
Subtraction of Nearly Equal Quantities

Suppose \( x > y \) and the \( k \)-digit representations of \( x \) and \( y \) are respectively

\[
fl(x) = 0.d_1d_2 \ldots d_p a_{p+1} a_{p+2} \ldots a_k \times 10^n
\]
\[
fl(y) = 0.d_1d_2 \ldots d_p b_{p+1} b_{p+2} \ldots b_k \times 10^n.
\]

Since the first \( p \) decimal digits of \( x \) and \( y \) are the same, then

\[
x \ominus y = 0.c_{p+1} c_{p+2} \ldots c_k \times 10^{n-p}
\]

where

\[
0.c_{p+1} c_{p+2} \ldots c_k = 0.a_{p+1} a_{p+2} \ldots a_k - 0.b_{p+1} b_{p+2} \ldots b_k.
\]

**Remark:** the result \( x \ominus y \) has at most \( k - p \) digits of significance.

Any further calculations with \( x \ominus y \) will inherit only \( k - p \) digits of significance.
Use 4-digit rounding arithmetic to solve the following quadratic equation.

\[ x^2 - 64.2x + 1 = 0 \]
Use 4-digit rounding arithmetic to solve the following quadratic equation.

\[ x^2 - 64.2x + 1 = 0 \]

If we solve the equation exactly we see that

\[ x_1 \approx 0.0155801 \quad \text{and} \quad x_2 \approx 64.1844. \]
Extended Example (2 of 5)

\[ x_1 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \]

\[ = \frac{-b - \sqrt{(0.6420 \times 10^2)^2 - (0.4000 \times 10^1)(0.1000 \times 10^1)(0.1000 \times 10^1)}}{2a} \]

\[ = \frac{-b - \sqrt{0.4122 \times 10^4 - 0.4000 \times 10^1}}{2a} \]

\[ = \frac{-b - \sqrt{0.4118 \times 10^4}}{2a} \]

\[ = \frac{0.6420 \times 10^2 - 0.6417 \times 10^2}{2a} \]

\[ = 0.3000 \times 10^{-1} \]

\[ = \frac{0.2000 \times 10^1}{0.1500 \times 10^{-1}} \]
Absolute Error:

\[ |0.0155801 - 0.015| = 0.0005801 \]

Relative Error:

\[ \frac{|0.0155801 - 0.015|}{|0.0155801|} = 0.0372334 \]
Extended Example (4 of 5)

Suppose now we rationalize the quadratic formula so as to avoid the subtraction of nearly equal quantities.

\[ x_1 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \left( \frac{-b + \sqrt{b^2 - 4ac}}{-b + \sqrt{b^2 - 4ac}} \right) \]

\[ = \frac{2c}{-b + \sqrt{b^2 - 4ac}} \]

\[ = \frac{0.2000 \times 10^1}{0.6420 \times 10^2 + 0.6417 \times 10^2} \]

\[ = \frac{0.2000 \times 10^1}{0.1284 \times 10^3} \]

\[ = 0.1558 \times 10^{-1} \]
Extended Example (5 of 5)

Absolute Error:

\[ |0.0155801 - 0.01558| = 10^{-7} \]

Relative Error:

\[ \frac{|0.0155801 - 0.01558|}{|0.0155801|} = 6.41844 \times 10^{-6} \]
Re-arrangement of Calculations

In addition to avoiding the subtraction of nearly equal results in calculations, it is generally a good idea to reduce the number of operations performed to obtain a desired result.

Example
Evaluate the polynomial $p = x^3 - 5x^2 + 3x - 2.7$ at $x = 7.14$ using 3-digit arithmetic.

For the sake of comparison, the exact value is $p = 90.1907$. 
Re-arrangement of Calculations

In addition to avoiding the subtraction of nearly equal results in calculations, it is generally a good idea to reduce the number of operations performed to obtain a desired result.

Example
Evaluate the polynomial

\[ p(x) = x^3 - 5.9x^2 + 3.4x + 2.7 \]

at \( x = 7.14 \) using 3-digit arithmetic.
Re-arrangement of Calculations

In addition to avoiding the subtraction of nearly equal results in calculations, it is generally a good idea to reduce the number of operations performed to obtain a desired result.

Example

Evaluate the polynomial

\[ p(x) = x^3 - 5.9x^2 + 3.4x + 2.7 \]

at \( x = 7.14 \) using 3-digit arithmetic.

For the sake of comparison, the exact value is \( p(7.14) = 90.1907 \).
Consider $p(x) = x^3 - 5.9x^2 + 3.4x + 2.7$ and intermediate results obtained using 3-digit chopping and 3-digit rounding arithmetic.
Consider \( p(x) = x^3 - 5.9x^2 + 3.4x + 2.7 \) and intermediate results obtained using 3-digit chopping and 3-digit rounding arithmetic.

<table>
<thead>
<tr>
<th>(x)</th>
<th>(x^2)</th>
<th>(x^3)</th>
<th>(5.9x^2)</th>
<th>(3.4x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chopping</td>
<td>0.714E01</td>
<td>0.509E02</td>
<td>0.363E03</td>
<td>0.300E03</td>
</tr>
<tr>
<td>Rounding</td>
<td>0.714E01</td>
<td>0.510E02</td>
<td>0.364E03</td>
<td>0.301E03</td>
</tr>
</tbody>
</table>
Evaluation (2 of 2)

Chopping:

\[
p(7.14) = 0.899 \times 10^2
\]

Abs. Err. = \[|90.1907 - 89.9| = 0.2907\]

Rel. Err. = \[\frac{|90.1907 - 89.9|}{|90.1907|} = 3.22317 \times 10^{-3}\]

Rounding:

\[
p(7.14) = 0.900 \times 10^2
\]

Abs. Err. = \[|90.1907 - 90.0| = 0.1907\]

Rel. Err. = \[\frac{|90.1907 - 90.0|}{|90.1907|} = 2.11441 \times 10^{-3}\]
Nested Polynomial Form

We may reduce the number of arithmetic operations performed by re-writing the polynomial as

\[ p(x) = x^3 - 5.9x^2 + 3.4x + 2.7 \]
\[ = x(x(x - 5.9) + 3.4) + 2.7. \]
Nested Polynomial Form

We may reduce the number of arithmetic operations performed by re-writing the polynomial as

\[
p(x) = x^3 - 5.9x^2 + 3.4x + 2.7
\]

\[
= x(x(x - 5.9) + 3.4) + 2.7.
\]

Evaluate \( p(7.14) \) using 3-digit chopping and rounding arithmetic.
Evaluation (Chopping)

\[ p(7.14) = 7.14(7.14(7.14 - 5.9) + 3.4) + 2.7 \]
\[ = 7.14(7.14(1.24) + 3.4) + 2.7 \]
\[ = 7.14(8.85 + 3.4) + 2.7 \]
\[ = 7.14(12.2) + 2.7 \]
\[ = 87.1 + 2.7 \]
\[ = 89.8 \]

Abs. Err. \[ = |90.1907 - 89.8| = 0.3907 \]

Rel. Err. \[ = \frac{|90.1907 - 89.8|}{90.1907} = 4.33193 \times 10^{-3} \]
Evaluation (Rounding)

\[ p(7.14) = 7.14 (7.14(7.14 - 5.9) + 3.4) + 2.7 \]
\[ = 7.14 (7.14(1.24) + 3.4) + 2.7 \]
\[ = 7.14(8.85 + 3.4) + 2.7 \]
\[ = 7.14(12.3) + 2.7 \]
\[ = 87.8 + 2.7 \]
\[ = 90.5 \]

Abs. Err. \[ = |90.1907 - 90.5| = 0.3093 \]

Rel. Err. \[ = \frac{|90.1907 - 90.5|}{90.1907} = 3.4294 \times 10^{-3} \]
Homework

- Read Section 1.2.
- Exercises: 1, 3, 5, 7, 11, 13, 19, 21, 25, 28