CENG 216 – Numerical Computation

Fundamentals

Mustafa Özuysal
mustafaozuysal@iyte.edu.tr

February 27, 2022

İzmir Institute of Technology

Slides are partially based on material from the main textbook:

"Numerical Analysis", The new international edition, 2ed,
by Timothy Sauer

Introduction

Evaluating a Polynomial

$$P(x) = 2x^4 + 3x^3 - 3x^2 + 5x - 1$$

Evaluating a Polynomial

$$P(x) = 2x^4 + 3x^3 - 3x^2 + 5x - 1$$

$$M_1 \rightarrow \quad P(x) = \underbrace{2 \cdot x \cdot x \cdot x \cdot x + 3 \cdot x \cdot x \cdot x - 3 \cdot x \cdot x + 5 \cdot x - 1}_{(\quad \text{mul.,} \quad \text{add.})}$$

Evaluating a Polynomial

$$P(x) = 2x^4 + 3x^3 - 3x^2 + 5x - 1$$

$$M_1 \rightarrow \quad P(x) = \underbrace{2 \cdot x \cdot x \cdot x \cdot x + 3 \cdot x \cdot x \cdot x - 3 \cdot x \cdot x + 5 \cdot x - 1}_{\text{(10 mul., 4 add.)}}$$

$$P(x) = 2x^4 + 3x^3 - 3x^2 + 5x - 1$$

$$M_1 \rightarrow \quad P(x) = \underbrace{2 \cdot x \cdot x \cdot x \cdot x + 3 \cdot x \cdot x \cdot x - 3 \cdot x \cdot x + 5 \cdot x - 1}_{\text{(10 mul., 4 add.)}}$$

$$M_2 \rightarrow \quad y = x \cdot x, z = y \cdot x, w = z \cdot x$$

Evaluating a Polynomial

$$P(x) = 2x^4 + 3x^3 - 3x^2 + 5x - 1$$

$$M_1 \rightarrow \quad P(x) = \underbrace{2 \cdot x \cdot x \cdot x \cdot x + 3 \cdot x \cdot x \cdot x - 3 \cdot x \cdot x + 5 \cdot x - 1}_{\text{(10 mul., 4 add.)}}$$

$$M_2 \rightarrow \quad y = x \cdot x, z = y \cdot x, w = z \cdot x$$

$$P(x) = \underbrace{2 \cdot w + 3 \cdot z - 3 \cdot y + 5 \cdot x - 1}_{\text{(\quad mul., \quad add.)}}$$

Evaluating a Polynomial

$$P(x) = 2x^4 + 3x^3 - 3x^2 + 5x - 1$$

$$M_1 \rightarrow \quad P(x) = \underbrace{2 \cdot x \cdot x \cdot x \cdot x + 3 \cdot x \cdot x \cdot x - 3 \cdot x \cdot x + 5 \cdot x - 1}_{\text{(10 mul., 4 add.)}}$$

$$M_2 \rightarrow \quad y = x \cdot x, z = y \cdot x, w = z \cdot x$$

$$P(x) = \underbrace{2 \cdot w + 3 \cdot z - 3 \cdot y + 5 \cdot x - 1}_{\text{(7 mul., 4 add.)}}$$

Evaluating a Polynomial

$$P(x) = 2x^4 + 3x^3 - 3x^2 + 5x - 1$$

$$M_1 \rightarrow \quad P(x) = \underbrace{2 \cdot x \cdot x \cdot x \cdot x + 3 \cdot x \cdot x \cdot x - 3 \cdot x \cdot x + 5 \cdot x - 1}_{\text{(10 mul., 4 add.)}}$$

$$M_2 \rightarrow \quad y = x \cdot x, z = y \cdot x, w = z \cdot x$$

$$P(x) = \underbrace{2 \cdot w + 3 \cdot z - 3 \cdot y + 5 \cdot x - 1}_{\text{(7 mul., 4 add.)}}$$

$$M_3 \rightarrow \quad P(x) = -1 + x \cdot \left(2x^3 + 3x^2 - 3x + 5\right)$$

Evaluating a Polynomial

$$
\begin{aligned}
& P(x) = 2x^4 + 3x^3 - 3x^2 + 5x - 1 \\
M_1 \rightarrow \quad & P(x) = \underbrace{2 \cdot x \cdot x \cdot x \cdot x + 3 \cdot x \cdot x \cdot x - 3 \cdot x \cdot x + 5 \cdot x - 1}_{\text{(10 mul., 4 add.)}} \\
M_2 \rightarrow \quad & y = x \cdot x, z = y \cdot x, w = z \cdot x \\
& P(x) = \underbrace{2 \cdot w + 3 \cdot z - 3 \cdot y + 5 \cdot x - 1}_{\text{(7 mul., 4 add.)}} \\
M_3 \rightarrow \quad & P(x) = -1 + x \cdot \left(2x^3 + 3x^2 - 3x + 5\right) \\
& \qquad = -1 + x \cdot \left(5 + x \cdot \left(2x^2 + 3x - 3\right)\right)
\end{aligned}
$$

Evaluating a Polynomial

$$
\begin{aligned}
& P(x) = 2x^4 + 3x^3 - 3x^2 + 5x - 1 \\
M_1 \rightarrow \quad & P(x) = \underbrace{2 \cdot x \cdot x \cdot x \cdot x + 3 \cdot x \cdot x \cdot x - 3 \cdot x \cdot x + 5 \cdot x - 1}_{\text{(10 mul., 4 add.)}} \\
M_2 \rightarrow \quad & y = x \cdot x, z = y \cdot x, w = z \cdot x \\
& P(x) = \underbrace{2 \cdot w + 3 \cdot z - 3 \cdot y + 5 \cdot x - 1}_{\text{(7 mul., 4 add.)}} \\
M_3 \rightarrow \quad & P(x) = -1 + x \cdot \left(2x^3 + 3x^2 - 3x + 5\right) \\
& = -1 + x \cdot \left(5 + x \cdot \left(2x^2 + 3x - 3\right)\right) \\
& = -1 + x \cdot (5 + x \cdot (-3 + x \cdot (2x + 3)))
\end{aligned}
$$

Evaluating a Polynomial

$$P(x) = 2x^4 + 3x^3 - 3x^2 + 5x - 1$$

$$M_1 \to \quad P(x) = \underbrace{2 \cdot x \cdot x \cdot x \cdot x + 3 \cdot x \cdot x \cdot x - 3 \cdot x \cdot x + 5 \cdot x - 1}_{\text{(10 mul., 4 add.)}}$$

$$M_2 \to \quad y = x \cdot x, z = y \cdot x, w = z \cdot x$$

$$P(x) = \underbrace{2 \cdot w + 3 \cdot z - 3 \cdot y + 5 \cdot x - 1}_{\text{(7 mul., 4 add.)}}$$

$$\begin{aligned} M_3 \to \quad P(x) &= -1 + x \cdot \left(2x^3 + 3x^2 - 3x + 5\right) \\ &= -1 + x \cdot \left(5 + x \cdot \left(2x^2 + 3x - 3\right)\right) \\ &= -1 + x \cdot (5 + x \cdot (-3 + x \cdot (2x + 3)))) \\ &= \underbrace{-1 + x \cdot (5 + x \cdot (-3 + x \cdot (3 + x \cdot 2))))}_{\text{(\quad mul., 4 add.)}} \end{aligned}$$

$$
\begin{aligned}
& & P(x) &= 2x^4 + 3x^3 - 3x^2 + 5x - 1 \\
M_1 &\rightarrow & P(x) &= \underbrace{2 \cdot x \cdot x \cdot x \cdot x + 3 \cdot x \cdot x \cdot x - 3 \cdot x \cdot x + 5 \cdot x - 1}_{\text{(10 mul., 4 add.)}} \\
M_2 &\rightarrow & y &= x \cdot x, z = y \cdot x, w = z \cdot x \\
& & P(x) &= \underbrace{2 \cdot w + 3 \cdot z - 3 \cdot y + 5 \cdot x - 1}_{\text{(7 mul., 4 add.)}} \\
M_3 &\rightarrow & P(x) &= -1 + x \cdot \left(2x^3 + 3x^2 - 3x + 5\right) \\
& & &= -1 + x \cdot \left(5 + x \cdot \left(2x^2 + 3x - 3\right)\right) \\
& & &= -1 + x \cdot (5 + x \cdot (-3 + x \cdot (2x + 3))) \\
& & &= \underbrace{-1 + x \cdot (5 + x \cdot (-3 + x \cdot (3 + x \cdot 2)))}_{\text{(4 mul., 4 add.)}}
\end{aligned}
$$

Evaluating a Polynomial

$$P(x) = 2x^4 + 3x^3 - 3x^2 + 5x - 1$$

$$M_1 \rightarrow \quad P(x) = \underbrace{2 \cdot x \cdot x \cdot x \cdot x + 3 \cdot x \cdot x \cdot x - 3 \cdot x \cdot x + 5 \cdot x - 1}_{\text{(10 mul., 4 add.)}}$$

$$M_2 \rightarrow \quad y = x \cdot x, z = y \cdot x, w = z \cdot x$$

$$P(x) = \underbrace{2 \cdot w + 3 \cdot z - 3 \cdot y + 5 \cdot x - 1}_{\text{(7 mul., 4 add.)}}$$

$$\begin{aligned} M_3 \rightarrow \quad P(x) &= -1 + x \cdot \left(2x^3 + 3x^2 - 3x + 5\right) \\ &= -1 + x \cdot \left(5 + x \cdot \left(2x^2 + 3x - 3\right)\right) \\ &= -1 + x \cdot (5 + x \cdot (-3 + x \cdot (2x + 3)))) \\ &= \underbrace{-1 + x \cdot (5 + x \cdot (-3 + x \cdot (3 + x \cdot 2))))}_{\text{(4 mul., 4 add.)}} \end{aligned}$$

M_3 is called Horner's Method.

Mathematical Function:

$$P(x) = 2x^4 + 3x^3 - 3x^2 + 5x - 1$$

Mathematical Function:

$$P(x) = 2x^4 + 3x^3 - 3x^2 + 5x - 1$$

v.s.

Computation of the Function (Algorithm):

Method M_1, method M_2, or method M_3 (Horner's approach)

Mathematical Function:

$$P(x) = 2x^4 + 3x^3 - 3x^2 + 5x - 1$$

v.s.

Computation of the Function (Algorithm):

Method M_1, method M_2, or method M_3 (Horner's approach)

A central theme in this course:

"The same mathematical function can be computed in possibly many different ways, each with its own characteristics regarding computation time, accuracy, ease of implementation, and so on."

Binary Numbers

Binary to Decimal

$$\ldots b_2 b_1 b_0, \qquad b_i \in \{0, 1\}, i = 0, \infty$$

Binary to Decimal

$$
\begin{aligned}
&\dots b_2 b_1 b_0, \qquad b_i \in \{0, 1\}, i = 0, \infty \\
= &\cdots b_2 \cdot 2^2 + b_1 \cdot 2^1 + b_0 \cdot 2^0
\end{aligned}
$$

Binary to Decimal

$$\ldots b_2 b_1 b_0 . b_{-1} b_{-2} \ldots, \qquad b_i \in \{0, 1\}, i = -\infty, \infty$$
$$= \cdots b_2 \cdot 2^2 + b_1 \cdot 2^1 + b_0 \cdot 2^0 + b_{-1} \cdot 2^{-1} + b_{-2} \cdot 2^{-2} + \cdots$$

$$
\begin{aligned}
&\dots b_2 b_1 b_0 . b_{-1} b_{-2} \dots, \qquad b_i \in \{0, 1\}, i = -\infty, \infty \\
&= \cdots b_2 \cdot 2^2 + b_1 \cdot 2^1 + b_0 \cdot 2^0 + b_{-1} \cdot 2^{-1} + b_{-2} \cdot 2^{-2} + \cdots \\
&= \sum_{i=-\infty}^{\infty} b_i \cdot 2^i
\end{aligned}
$$

$$\begin{aligned} &\dots b_2 b_1 b_0 . b_{-1} b_{-2} \dots, \qquad b_i \in \{0, 1\}, i = -\infty, \infty \\ &= \cdots b_2 \cdot 2^2 + b_1 \cdot 2^1 + b_0 \cdot 2^0 + b_{-1} \cdot 2^{-1} + b_{-2} \cdot 2^{-2} + \cdots \\ &= \sum_{i=-\infty}^{\infty} b_i \cdot 2^i \end{aligned}$$

Examples:

$$(100.0)_2 =$$

$$
\begin{aligned}
&\dots b_2 b_1 b_0 . b_{-1} b_{-2} \dots, \qquad b_i \in \{0, 1\}, i = -\infty, \infty \\
&= \cdots b_2 \cdot 2^2 + b_1 \cdot 2^1 + b_0 \cdot 2^0 + b_{-1} \cdot 2^{-1} + b_{-2} \cdot 2^{-2} + \cdots \\
&= \sum_{i=-\infty}^{\infty} b_i \cdot 2^i
\end{aligned}
$$

Examples:

$$(100.0)_2 = 4$$

Binary to Decimal

$$
\begin{aligned}
&\ldots b_2 b_1 b_0 . b_{-1} b_{-2} \ldots, \qquad b_i \in \{0, 1\}, i = -\infty, \infty \\
&= \cdots b_2 \cdot 2^2 + b_1 \cdot 2^1 + b_0 \cdot 2^0 + b_{-1} \cdot 2^{-1} + b_{-2} \cdot 2^{-2} + \cdots \\
&= \sum_{i=-\infty}^{\infty} b_i \cdot 2^i
\end{aligned}
$$

Examples:

$$(100.0)_2 = 4$$

$$(0.11)_2 =$$

$$
\begin{aligned}
&\dots b_2 b_1 b_0 . b_{-1} b_{-2} \dots, \qquad b_i \in \{0, 1\}, i = -\infty, \infty \\
&= \cdots b_2 \cdot 2^2 + b_1 \cdot 2^1 + b_0 \cdot 2^0 + b_{-1} \cdot 2^{-1} + b_{-2} \cdot 2^{-2} + \cdots \\
&= \sum_{i=-\infty}^{\infty} b_i \cdot 2^i
\end{aligned}
$$

Examples:

$$
\begin{aligned}
(100.0)_2 &= 4 \\
(0.11)_2 &= \frac{3}{4}
\end{aligned}
$$

Decimal to Binary

$$53.7 = (?)_2$$

Decimal to Binary

$$53.7 = (?)_2 = 53 + 0.7$$

Decimal to Binary

$$
\begin{aligned}
53.7 = (?)_2 &= 53 + 0.7 \\
&= (\qquad\quad)_2 + (\quad)_2
\end{aligned}
$$

Decimal to Binary

$$53.7 = (?)_2 = 53 + 0.7$$
$$= (\quad\quad 1)_2 + (\quad)_2$$

$53/2 = 26 \quad \text{R} \quad 1$

$$53.7 = (?)_2 = 53 + 0.7$$
$$= (\quad 01)_2 + (\quad)_2$$

$$53/2 = 26 \quad R \quad 1$$
$$26/2 = 13 \quad R \quad 0$$

Decimal to Binary

$$53.7 = (?)_2 = 53 + 0.7$$
$$= (\quad 101)_2 + (\quad)_2$$

$$53/2 = 26 \quad R \quad 1$$
$$26/2 = 13 \quad R \quad 0$$
$$13/2 = 6 \quad R \quad 1$$

Decimal to Binary

$$53.7 = (?)_2 \; = 53 + 0.7$$
$$= (\quad 0101)_2 + (\quad\quad)_2$$

$$53/2 = 26 \quad R \quad 1$$
$$26/2 = 13 \quad R \quad 0$$
$$13/2 = 6 \quad R \quad 1$$
$$6/2 = 3 \quad R \quad 0$$

Decimal to Binary

$$53.7 = (?)_2 = 53 + 0.7$$
$$= (\ 10101)_2 + (\quad)_2$$

$$53/2 = 26 \qquad R \quad 1$$
$$26/2 = 13 \qquad R \quad 0$$
$$13/2 = 6 \qquad R \quad 1$$
$$6/2 = 3 \qquad R \quad 0$$
$$3/2 = 1 \qquad R \quad 1$$

Decimal to Binary

$$
\begin{aligned}
53.7 = (?)_2 &= 53 + 0.7 \\
&= (110101)_2 + (\quad)_2
\end{aligned}
$$

$$
\begin{array}{rcl}
53/2 = 26 & \mathrm{R} & 1 \\
26/2 = 13 & \mathrm{R} & 0 \\
13/2 = 6 & \mathrm{R} & 1 \\
6/2 = 3 & \mathrm{R} & 0 \\
3/2 = 1 & \mathrm{R} & 1 \\
1/2 = 0 & \mathrm{R} & 1
\end{array}
$$

Decimal to Binary

$$53.7 = (?)_2 = 53 + 0.7$$
$$= (110101)_2 + (0.1)_2$$

$$\begin{aligned} 53/2 &= 26 && \text{R} \quad 1 \\ 26/2 &= 13 && \text{R} \quad 0 \\ 13/2 &= 6 && \text{R} \quad 1 \\ 6/2 &= 3 && \text{R} \quad 0 \\ 3/2 &= 1 && \text{R} \quad 1 \\ 1/2 &= 0 && \text{R} \quad 1 \end{aligned}$$

$$0.7 \cdot 2 = .4 \qquad +1$$

Decimal to Binary

$$53.7 = (?)_2 = 53 + 0.7$$
$$= (110101)_2 + (0.10)_2$$

$$53/2 = 26 \quad R \quad 1$$
$$26/2 = 13 \quad R \quad 0$$
$$13/2 = 6 \quad R \quad 1$$
$$6/2 = 3 \quad R \quad 0$$
$$3/2 = 1 \quad R \quad 1$$
$$1/2 = 0 \quad R \quad 1$$

$$0.7 \cdot 2 = .4 \quad +1$$
$$0.4 \cdot 2 = .8 \quad +0$$

Decimal to Binary

$$53.7 = (?)_2 = 53 + 0.7$$
$$= (110101)_2 + (0.101)_2$$

$$53/2 = 26 \quad R \quad 1$$
$$26/2 = 13 \quad R \quad 0$$
$$13/2 = 6 \quad R \quad 1$$
$$6/2 = 3 \quad R \quad 0$$
$$3/2 = 1 \quad R \quad 1$$
$$1/2 = 0 \quad R \quad 1$$

$$0.7 \cdot 2 = .4 \quad +1$$
$$0.4 \cdot 2 = .8 \quad +0$$
$$0.8 \cdot 2 = .6 \quad +1$$

Decimal to Binary

$$53.7 = (?)_2 = 53 + 0.7$$
$$= (110101)_2 + (0.1011)_2$$

$53/2 = 26$	R	1	$0.7 \cdot 2 = .4$	$+1$
$26/2 = 13$	R	0	$0.4 \cdot 2 = .8$	$+0$
$13/2 = 6$	R	1	$0.8 \cdot 2 = .6$	$+1$
$6/2 = 3$	R	0	$0.6 \cdot 2 = .2$	$+1$
$3/2 = 1$	R	1		
$1/2 = 0$	R	1		

Decimal to Binary

$$53.7 = (?)_2 = 53 + 0.7$$
$$= (110101)_2 + (0.10110)_2$$

$$
\begin{aligned}
53/2 &= 26 \quad \text{R} \quad 1 \\
26/2 &= 13 \quad \text{R} \quad 0 \\
13/2 &= 6 \quad \text{R} \quad 1 \\
6/2 &= 3 \quad \text{R} \quad 0 \\
3/2 &= 1 \quad \text{R} \quad 1 \\
1/2 &= 0 \quad \text{R} \quad 1
\end{aligned}
$$

$$
\begin{aligned}
0.7 \cdot 2 &= .4 \quad +1 \\
0.4 \cdot 2 &= .8 \quad +0 \\
0.8 \cdot 2 &= .6 \quad +1 \\
0.6 \cdot 2 &= .2 \quad +1 \\
0.2 \cdot 2 &= .4 \quad +0
\end{aligned}
$$

Decimal to Binary

$$\begin{aligned} 53.7 = (?)_2 &= 53 + 0.7 \\ &= (110101)_2 + \left(0.1\overline{0110}\right)_2 \end{aligned}$$

$$\begin{array}{rll} 53/2 = 26 & \text{R} & 1 \\ 26/2 = 13 & \text{R} & 0 \\ 13/2 = 6 & \text{R} & 1 \\ 6/2 = 3 & \text{R} & 0 \\ 3/2 = 1 & \text{R} & 1 \\ 1/2 = 0 & \text{R} & 1 \end{array}$$

$$\begin{array}{rl} 0.7 \cdot 2 = .4 & +1 \\ 0.4 \cdot 2 = .8 & +0 \\ 0.8 \cdot 2 = .6 & +1 \\ 0.6 \cdot 2 = .2 & +1 \\ 0.2 \cdot 2 = .4 & +0 \\ 0.4 \cdot 2 = .8 & +0 \end{array}$$

$$53.7 = (?)_2 = 53 + 0.7$$
$$= (110101)_2 + \left(0.1\overline{0110}\right)_2 = \left(110101.1\overline{0110}\right)_2$$

$$\begin{aligned}
53/2 &= 26 \quad \text{R} \quad 1 \\
26/2 &= 13 \quad \text{R} \quad 0 \\
13/2 &= 6 \quad \text{R} \quad 1 \\
6/2 &= 3 \quad \text{R} \quad 0 \\
3/2 &= 1 \quad \text{R} \quad 1 \\
1/2 &= 0 \quad \text{R} \quad 1
\end{aligned}$$

$$\begin{aligned}
0.7 \cdot 2 &= .4 \quad +1 \\
0.4 \cdot 2 &= .8 \quad +0 \\
0.8 \cdot 2 &= .6 \quad +1 \\
0.6 \cdot 2 &= .2 \quad +1 \\
0.2 \cdot 2 &= .4 \quad +0 \\
0.4 \cdot 2 &= .8 \quad +0
\end{aligned}$$

$$(0.\overline{1011})_2 = x$$

$$(0.\overline{1011})_2 = x$$

$$2^4 x = (1011.\overline{1011})_2$$

$$(0.\overline{1011})_2 = x$$

$$2^4 x = (1011.\overline{1011})_2$$
$$x = (0.\overline{1011})_2$$

Repeated Fractions and Hexadecimal Numbers

$$\left(0.\overline{1011}\right)_2 = x$$

$$\begin{aligned} 2^4 x &= \left(1011.\overline{1011}\right)_2 \\ x &= \left(0.\overline{1011}\right)_2 \\ 2^4 x - x = 15x &= (1011)_2 = 8 + 2 + 1 = 11 \end{aligned}$$

$$\left(0.\overline{1011}\right)_2 = x$$

$$\begin{aligned} 2^4 x &= \left(1011.\overline{1011}\right)_2 \\ x &= \left(0.\overline{1011}\right)_2 \\ 2^4 x - x = 15x &= (1011)_2 = 8 + 2 + 1 = 11 \\ x &= \frac{11}{15} \end{aligned}$$

$$(0.\overline{1011})_2 = x$$

$$\begin{aligned} 2^4 x &= (1011.\overline{1011})_2 \\ x &= (0.\overline{1011})_2 \\ 2^4 x - x = 15x &= (1011)_2 = 8 + 2 + 1 = 11 \\ x &= \frac{11}{15} \end{aligned}$$

$$(10110010)_2 = (\mathrm{B}2)_{16}$$

$$(0.\overline{1011})_2 = x$$

$$\begin{aligned} 2^4 x &= (1011.\overline{1011})_2 \\ x &= (0.\overline{1011})_2 \\ 2^4 x - x = 15x &= (1011)_2 = 8 + 2 + 1 = 11 \\ x &= \frac{11}{15} \end{aligned}$$

$$(10110010)_2 = (\mathrm{B2})_{16} = \texttt{0xB2}$$

Number Representations

Representing Integers

On a digital computer, we can represent integers from a finite range exactly using a fixed number of bits and a base of two.

$$23 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \rightarrow 00010111$$

On a digital computer, we can represent integers from a finite range exactly using a fixed number of bits and a base of two.

$$23 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \rightarrow 00010111$$

We still need to consider

- The number of bits to use

On a digital computer, we can represent integers from a finite range exactly using a fixed number of bits and a base of two.

$$23 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \rightarrow 00010111$$

We still need to consider

- The number of bits to use
- Whether the number is signed or unsigned

On a digital computer, we can represent integers from a finite range exactly using a fixed number of bits and a base of two.

$$23 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \rightarrow 00010111$$

We still need to consider

- The number of bits to use
- Whether the number is signed or unsigned
- Whether the operations overflow the available range

Representing Integers

On a digital computer, we can represent integers from a finite range exactly using a fixed number of bits and a base of two.

$$23 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \to 00010111$$

We still need to consider

- The number of bits to use
- Whether the number is signed or unsigned
- Whether the operations **overflow** the available range
    - The result might use more bits than available.

Representing Integers

On a digital computer, we can represent integers from a finite range exactly using a fixed number of bits and a base of two.

$$23 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \rightarrow 00010111$$

We still need to consider

- The number of bits to use
- Whether the number is signed or unsigned
- Whether the operations overflow the available range
  - The result might use more bits than available.
  - For signed numbers, the computation might overflow into the sign bit. For example, adding two positive numbers might yield a negative number.

Fixed-Point Representation

If we want to represent fractional parts, a first approach might be to use a fixed number of binary digits for the fractional part:

$$3.25 = 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} \rightarrow 000011.01$$

Fixed-Point Representation

If we want to represent fractional parts, a first approach might be to use a fixed number of binary digits for the fractional part:

$$3.25 = 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} \rightarrow 000011.01$$

We use eight bits but two of these form the fractional part. We can represent the fractions 0.00, 0.25, 0.5, 0.75 exactly. Every other fraction will require rounding.

Fixed-Point Representation

If we want to represent fractional parts, a first approach might be to use a fixed number of binary digits for the fractional part:

$$3.25 = 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} \rightarrow 000011.01$$

We use eight bits but two of these form the fractional part. We can represent the fractions 0.00, 0.25, 0.5, 0.75 exactly. Every other fraction will require **rounding**.

This is called a **fixed-point** representation. The advantage is that we can use the existing integer arithmetic hardware by simply ignoring the position of the point during the computations.

Fixed-Point Representation

If we want to represent fractional parts, a first approach might be to use a fixed number of binary digits for the fractional part:

$$3.25 = 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} \rightarrow 000011.01$$

We use eight bits but two of these form the fractional part. We can represent the fractions 0.00, 0.25, 0.5, 0.75 exactly. Every other fraction will require **rounding**.

This is called a **fixed-point** representation. The advantage is that we can use the existing integer arithmetic hardware by simply ignoring the position of the point during the computations.

However, we limit precision to multiples of a fixed fraction such as 0.25 and our precision stays the same for small and large numbers such as those around 1 and 10000.

The main idea behind the floating point representation is that we **usually** need more bits for the fractional parts of small numbers than the fractional parts of the large numbers.

The main idea behind the floating point representation is that we usually need more bits for the fractional parts of small numbers than the fractional parts of the large numbers.

Scientific notation has exactly this property so we can use it as a basis for our representation:

$$23143600 = 2.31436 \times 10^{7}$$
$$0.000000231436 = 2.31436 \times 10^{-7}$$

The main idea behind the floating point representation is that we *usually* need more bits for the fractional parts of small numbers than the fractional parts of the large numbers.

Scientific notation has exactly this property so we can use it as a basis for our representation:

$$23143600 = 2.31436 \times 10^{7}$$
$$0.000000231436 = 2.31436 \times 10^{-7}$$

To represent these two numbers, we need the same number of bits, just as many bits as necessary to store 2.31436 and $+7$ or -7.

The floating-point representation

$$(-1)^s c \times b^e$$

has three parameters

The floating-point representation

$$(-1)^s c \times b^e$$

has three parameters

- b, which is called the **base** or **radix** and usually $b = 2$.

The floating-point representation

$$(-1)^s c \times b^e$$

has three parameters

- b, which is called the base or radix and usually $b = 2$.
- The number of bits used to represent c, which is called the significand or mantissa.

The floating-point representation

$$(-1)^s c \times b^e$$

has three parameters

- b, which is called the **base** or **radix** and usually $b = 2$.
- The number of bits used to represent c, which is called the **significand** or **mantissa**.
- The allowed range of integer values $[L_e, U_e]$ for e called the **exponent**.

Let's assume that we store our numbers as

$$1.ab \times 2^e,$$

where a and b are binary digits 0 or 1 and e is in the range $[-1, +1]$.

Visualizing Floating Point Numbers

Let's assume that we store our numbers as

$$1.ab \times 2^e,$$

where a and b are binary digits 0 or 1 and e is in the range $[-1, +1]$. Then we can only represent the following numbers

$$0.5, 0.625, 0.75, 0.875, 1, 1.25, 1.5, 1.75, 2, 2.5, 3, 3.5$$

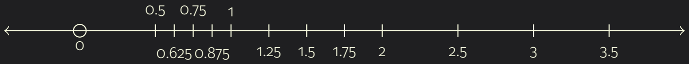
Let's assume that we store our numbers as

$$1.ab \times 2^e,$$

where a and b are binary digits 0 or 1 and e is in the range $[-1, +1]$. Then we can only represent the following numbers

$$0.5, 0.625, 0.75, 0.875, 1, 1.25, 1.5, 1.75, 2, 2.5, 3, 3.5$$

Graphically:



- 0 and negative numbers are not included.

Visualizing Floating Point Numbers

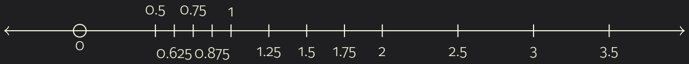
Let's assume that we store our numbers as

$$1.ab \times 2^e,$$

where a and b are binary digits 0 or 1 and e is in the range $[-1, +1]$. Then we can only represent the following numbers

$$0.5, 0.625, 0.75, 0.875, 1, 1.25, 1.5, 1.75, 2, 2.5, 3, 3.5$$

Graphically:



- 0 and negative numbers are not included.
- There is a gap between 0 and the first number 0.5.

Visualizing Floating Point Numbers

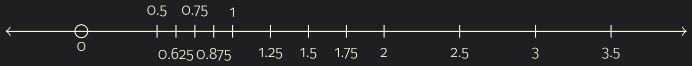
Let's assume that we store our numbers as

$$1.ab \times 2^{e},$$

where a and b are binary digits 0 or 1 and e is in the range $[-1, +1]$. Then we can only represent the following numbers

$$0.5, 0.625, 0.75, 0.875, 1, 1.25, 1.5, 1.75, 2, 2.5, 3, 3.5$$

Graphically:



- 0 and negative numbers are not included.
- There is a gap between 0 and the first number 0.5.
- The precision drops as the numbers get larger.

Floating Point Representation

Normalized form: $\pm 1.\underbrace{bbb\ldots b}_{\text{mantissa}}\times 2^{e}$

IEEE 754 Floating Point Standard

$$\text{Normalized form: } \pm 1.\underbrace{bbb\ldots b}_{\text{mantissa}} \times 2^{e}$$

$$9 = (1001)_2 = +1.001 \times 2^3$$

$$\text{Normalized form: } \pm 1.\underbrace{bbb\ldots b}_{\text{mantissa}} \times 2^e$$

$$9 = (1001)_2 = +1.001 \times 2^3$$

precision	sign	exponent	mantissa	total
single	1	8	23	32
double	1	11	52	64
extended	1	15	64	80

$$1 : +1.\overbrace{000 \ldots\ldots\ldots 000}^{\text{52 bits}} \times 2^0$$

Double Precision Floating Point Numbers

$$1 : +1.\overbrace{000\ldots\ldots\ldots000}^{\text{52 bits}} \times 2^0$$

$$1 + 2^{-52} : +1.\overbrace{000\ldots\ldots\ldots001}^{\text{52 bits}} \times 2^0$$

Double Precision Floating Point Numbers

$$1 : +1.\overbrace{000\ldots\ldots\ldots000}^{\text{52 bits}}\times 2^0$$

$$1 + 2^{-52} : +1.\overbrace{000\ldots\ldots\ldots001}^{\text{52 bits}}\times 2^0$$

machine epsilon: $\epsilon_{\text{mach}} = 2^{-52}$ (double-precision)

$$9.4 = \left(1001.\overline{0110}\right)_2$$

$$9.4 = \left(1001.\overline{0110}\right)_2$$

$$= +1.\overbrace{00101100110\ldots\ldots\ldots01100}^{\text{52 bits}}\ \underbrace{\overbrace{110}^{b^{53}b^{54}b^{55}}\ \ldots}_{\text{truncate or round}}\ \times 2^3$$

Double Precision Floating Point Numbers

$$9.4 = \left(1001.\overline{0110}\right)_2$$

$$= +1.\overbrace{00101100110\,.\,.\,.\,.\,.\,.\,.\,.\,.\,01100}^{\text{52 bits}}\quad\underbrace{\overbrace{110}^{b^{53}b^{54}b^{55}}\quad\ldots}_{\text{truncate or round}}\quad\times 2^3$$

Round to nearest rule:

- if b^{53} is 0 then round down $\rightarrow$ truncate.

$$9.4 = \left(1001.\overline{0110}\right)_2$$

$$= +1.\overbrace{00101100110\ldots\ldots\ldots01100}^{\text{52 bits}}\ \underbrace{\overbrace{110}^{b^{53}b^{54}b^{55}}\ \ldots}_{\text{truncate or round}}\ \times 2^3$$

Round to nearest rule:

- if b^{53} is 0 then round down $\rightarrow$ truncate.
- if b^{53} is 1 and the rest is non-zero then round up.

$$9.4 = \left(1001.\overline{0110}\right)_2$$

$$= +1.\overbrace{00101100110\ldots\ldots\ldots01100}^{\text{52 bits}}\ \underbrace{\overbrace{110}^{b^{53}b^{54}b^{55}}\ \ldots}_{\text{truncate or round}}\ \times 2^3$$

Round to nearest rule:

- if b^{53} is 0 then round down $\rightarrow$ truncate.
- if b^{53} is 1 and the rest is non-zero then round up.
- if b^{53} is 1 and the rest is all zeros then round such that b^{52} is 0.

$$9.4 = \left(1001.\overline{0110}\right)_2$$

$$= +1.\overbrace{00101100110\ldots\ldots\ldots01100}^{\text{52 bits}}\ \underbrace{\overbrace{110}^{b^{53}b^{54}b^{55}}\ \ldots}_{\text{truncate or round}}\ \times 2^3$$

Round to nearest rule:

- if b^{53} is 0 then round down $\rightarrow$ truncate.
- if b^{53} is 1 and the rest is non-zero then round up.
- if b^{53} is 1 and the rest is all zeros then round such that b^{52} is 0.

$$9.4 = +1.\overbrace{0010110011\ldots\ldots\ldots01101}^{\text{52 bits}}\times 2^3$$

Roundoff Error

$$9.4 = +1.\overbrace{00101100110\ldots\ldots\ldots01100}^{\text{52 bits}}1100110\ldots \times 2^3$$

Roundoff Error

$$9.4 = +1.\overbrace{00101100110\ldots\ldots\ldots01100}^{\text{52 bits}}\,1100110\ldots \times 2^3$$

$$\Big\downarrow \left(\text{subtract } 0.\overline{1100} \cdot 2^{-52} \cdot 2^3\right)$$

$$9.4 = +1.\overbrace{00101100110\ldots\ldots\ldots01100}^{\text{52 bits}} \times 2^3$$

Roundoff Error

$$9.4 = +1.\overbrace{00101100110\ldots\ldots\ldots01100}^{\text{52 bits}}1100110\ldots \times 2^3$$

$$\Big\downarrow \left(\text{subtract } 0.\overline{1100} \cdot 2^{-52} \cdot 2^3\right)$$

$$9.4 = +1.\overbrace{00101100110\ldots\ldots\ldots01100}^{\text{52 bits}} \times 2^3$$

$$\Big\downarrow \left(\text{add } 2^{-52} \cdot 2^3\right)$$

$$9.4 = +1.\overbrace{00101100110\ldots\ldots\ldots01101}^{\text{52 bits}} \times 2^3$$

Roundoff Error

$$\texttt{fl}\,(9.4) = 9.4 - \left(0.\overline{1100} \cdot 2^{-52} \cdot 2^{3}\right) + \left(2^{-52} \cdot 2^{3}\right)$$

$$\begin{aligned}\texttt{fl}\,(9.4) &= 9.4 - \left(0.\overline{1100} \cdot 2^{-52} \cdot 2^{3}\right) + \left(2^{-52} \cdot 2^{3}\right) \\ &= 9.4 - 0.4 \cdot 2^{-48} + 2^{-49}\end{aligned}$$

$$
\begin{aligned}
\texttt{fl}\,(9.4) &= 9.4 - \left(0.\overline{1100} \cdot 2^{-52} \cdot 2^{3}\right) + \left(2^{-52} \cdot 2^{3}\right) \\
&= 9.4 - 0.4 \cdot 2^{-48} + 2^{-49} \\
&= 9.4 + 0.2 \cdot 2^{-49}
\end{aligned}
$$

Roundoff Error

$$\begin{aligned}\texttt{fl}\,(9.4) &= 9.4 - \left(0.\overline{1100} \cdot 2^{-52} \cdot 2^{3}\right) + \left(2^{-52} \cdot 2^{3}\right) \\ &= 9.4 - 0.4 \cdot 2^{-48} + 2^{-49} \\ &= 9.4 + 0.2 \cdot 2^{-49}\end{aligned}$$

$$\text{absolute error} = |x_c - x| = |\texttt{fl}\,(9.4) - 9.4|$$

Roundoff Error

$$\begin{aligned}\texttt{fl}\,(9.4) &= 9.4 - \left(0.\overline{1100} \cdot 2^{-52} \cdot 2^{3}\right) + \left(2^{-52} \cdot 2^{3}\right) \\ &= 9.4 - 0.4 \cdot 2^{-48} + 2^{-49} \\ &= 9.4 + 0.2 \cdot 2^{-49}\end{aligned}$$

$$\begin{aligned}\text{absolute error} &= |x_c - x| = |\texttt{fl}\,(9.4) - 9.4| \\ &= 0.2 \cdot 2^{-49}\end{aligned}$$

$$
\begin{aligned}
\texttt{fl}\,(9.4) &= 9.4 - \left(0.\overline{1100} \cdot 2^{-52} \cdot 2^{3}\right) + \left(2^{-52} \cdot 2^{3}\right) \\
&= 9.4 - 0.4 \cdot 2^{-48} + 2^{-49} \\
&= 9.4 + 0.2 \cdot 2^{-49}
\end{aligned}
$$

$$
\begin{aligned}
\text{absolute error} &= |x_c - x| = |\texttt{fl}\,(9.4) - 9.4| \\
&= 0.2 \cdot 2^{-49}
\end{aligned}
$$

$$
\text{relative error} = \frac{|x_c - x|}{x} = \frac{|\texttt{fl}\,(9.4) - 9.4|}{9.4}
$$

Roundoff Error

$$
\begin{aligned}
\texttt{fl}\,(9.4) &= 9.4 - \left(0.\overline{1100} \cdot 2^{-52} \cdot 2^{3}\right) + \left(2^{-52} \cdot 2^{3}\right) \\
&= 9.4 - 0.4 \cdot 2^{-48} + 2^{-49} \\
&= 9.4 + 0.2 \cdot 2^{-49}
\end{aligned}
$$

$$
\begin{aligned}
\text{absolute error} &= |x_c - x| = |\texttt{fl}\,(9.4) - 9.4| \\
&= 0.2 \cdot 2^{-49}
\end{aligned}
$$

$$
\begin{aligned}
\text{relative error} &= \frac{|x_c - x|}{x} = \frac{|\texttt{fl}\,(9.4) - 9.4|}{9.4} \\
&= \frac{8}{47} \cdot 2^{-52}
\end{aligned}
$$

$$\begin{aligned}
\texttt{fl}\,(9.4) &= 9.4 - \left(0.\overline{1100} \cdot 2^{-52} \cdot 2^{3}\right) + \left(2^{-52} \cdot 2^{3}\right) \\
&= 9.4 - 0.4 \cdot 2^{-48} + 2^{-49} \\
&= 9.4 + 0.2 \cdot 2^{-49}
\end{aligned}$$

$$\begin{aligned}
\text{absolute error} &= |x_c - x| = |\texttt{fl}\,(9.4) - 9.4| \\
&= 0.2 \cdot 2^{-49}
\end{aligned}$$

$$\begin{aligned}
\text{relative error} &= \frac{|x_c - x|}{x} = \frac{|\texttt{fl}\,(9.4) - 9.4|}{9.4} \\
&= \frac{8}{47} \cdot 2^{-52} \\
&\leq \frac{1}{2}\epsilon_{\text{mach}}
\end{aligned}$$

Addition of Floating Point Numbers

$$\texttt{fl}(9.4) = 9.4 + 0.2 \cdot 2^{-49}$$

$$\texttt{fl}(9.0) =$$

$$\texttt{fl}(0.4) =$$

Addition of Floating Point Numbers

$$\texttt{fl}\,(9.4) = 9.4 + 0.2 \cdot 2^{-49}$$

$$\texttt{fl}\,(9.0) = 9.0$$

$$\texttt{fl}\,(0.4) =$$

Addition of Floating Point Numbers

$$\texttt{fl}\,(9.4) = 9.4 + 0.2 \cdot 2^{-49}$$
$$\texttt{fl}\,(9.0) = 9.0$$
$$\texttt{fl}\,(0.4) = 0.4 + 0.1 \cdot 2^{-52}$$

Addition of Floating Point Numbers

$$\texttt{fl}(9.4) = 9.4 + 0.2 \cdot 2^{-49}$$
$$\texttt{fl}(9.0) = 9.0$$
$$\texttt{fl}(0.4) = 0.4 + 0.1 \cdot 2^{-52}$$

$$\texttt{9.4 - 9.0 - 0.4} = \texttt{fl}\left(\texttt{fl}\left(\texttt{fl}(9.4) - \texttt{fl}(9.0)\right) - \texttt{fl}(0.4)\right)$$

Addition of Floating Point Numbers

$$\texttt{fl}\,(9.4) = 9.4 + 0.2 \cdot 2^{-49}$$
$$\texttt{fl}\,(9.0) = 9.0$$
$$\texttt{fl}\,(0.4) = 0.4 + 0.1 \cdot 2^{-52}$$

$$\begin{aligned}\texttt{9.4 - 9.0 - 0.4} &= \texttt{fl}\,(\texttt{fl}\,(\texttt{fl}\,(9.4) - \texttt{fl}\,(9.0)) - \texttt{fl}\,(0.4)) \\ &= \texttt{fl}\left(9.4 + 0.2 \cdot 2^{-49} - 9.0\right) - 0.4 - 0.1 \cdot 2^{-52}\end{aligned}$$

$$\begin{aligned} \texttt{fl}(9.4) &= 9.4 + 0.2 \cdot 2^{-49} \\ \texttt{fl}(9.0) &= 9.0 \\ \texttt{fl}(0.4) &= 0.4 + 0.1 \cdot 2^{-52} \end{aligned}$$

$$\begin{aligned} \texttt{9.4 - 9.0 - 0.4} &= \texttt{fl}(\texttt{fl}(\texttt{fl}(9.4) - \texttt{fl}(9.0)) - \texttt{fl}(0.4)) \\ &= \texttt{fl}\left(9.4 + 0.2 \cdot 2^{-49} - 9.0\right) - 0.4 - 0.1 \cdot 2^{-52} \\ &= 0.4 + 0.2 \cdot 2^{-49} - 0.4 - 0.1 \cdot 2^{-52} \end{aligned}$$

Addition of Floating Point Numbers

$$
\begin{aligned}
\texttt{fl}\,(9.4) &= 9.4 + 0.2 \cdot 2^{-49} \\
\texttt{fl}\,(9.0) &= 9.0 \\
\texttt{fl}\,(0.4) &= 0.4 + 0.1 \cdot 2^{-52}
\end{aligned}
$$

$$
\begin{aligned}
\texttt{9.4 - 9.0 - 0.4} &= \texttt{fl}\,(\texttt{fl}\,(\texttt{fl}\,(9.4) - \texttt{fl}\,(9.0)) - \texttt{fl}\,(0.4)) \\
&= \texttt{fl}\left(9.4 + 0.2 \cdot 2^{-49} - 9.0\right) - 0.4 - 0.1 \cdot 2^{-52} \\
&= 0.4 + 0.2 \cdot 2^{-49} - 0.4 - 0.1 \cdot 2^{-52} \\
&= (1.6 - 0.1) \cdot 2^{-52} = 1.5 \cdot 2^{-52} = 3 \cdot 2^{-53}
\end{aligned}
$$

IEEE 754 Standard: The Details

IEEE 754 Standard

$$(-1)^s \times 1.b_0 \ldots b_{p-1} \times 2^{e-\text{bias}}$$

IEEE 754 Standard

$$(-1)^s \times 1.b_0 \ldots b_{p-1} \times 2^{e-\text{bias}}$$

Precision	Bits in				Bias	L_e	U_e
	Sign	Exp.	Mantissa	Total			
Single	1	8	23+1	32	127	-126	127
Double	1	11	52+1	64	1023	-1022	1023

IEEE 754 Standard

$$(-1)^s \times 1.b_0 \ldots b_{p-1} \times 2^{e-\text{bias}}$$

Precision	Bits in				Bias	L_e	U_e
	Sign	Exp.	Mantissa	Total			
Single	1	8	23+1	32	127	-126	127
Double	1	11	52+1	64	1023	-1022	1023

- e is stored as a positive number, the real exponent value is $e -$ bias.
- For normalized numbers e can not be all zeros or all ones in binary, these are reserved for special cases.
- L_e/U_e is the lowest/highest possible exponent value.

For normalized numbers the bits in the exponent can not be all zeros or all ones. These two cases are reserved for special numbers as follows:

	e	$b_0 \ldots b_{p-1}$	sign bit
Positive Zero	All Zeros	All Zeros	0
Negative Zero	All Zeros	All Zeros	1
Subnormal	All Zeros	Non-Zero	0 or 1
$+\infty$	All Ones	All Zeros	0
$-\infty$	All Ones	All Zeros	1
NaN	All Ones	Non-Zero	0 or 1

Subnormal Numbers

For an exponent of all zeros and a non-zero significand we change the representation to

$$(-1)^s \times 0.b_0 \ldots b_{p-1} \times 2^{L_e}$$

For an exponent of all zeros and a non-zero significand we change the representation to

$$(-1)^s \times 0.b_0 \ldots b_{p-1} \times 2^{L_e}$$

These are called **subnormal** or **denormal** numbers. They fill the gap around zero and they are separated from each other by the constant factor of 2^{L_e}.

For an exponent of all zeros and a non-zero significand we change the representation to

$$(-1)^s \times 0.b_0 \ldots b_{p-1} \times 2^{L_e}$$

These are called **subnormal** or **denormal** numbers. They fill the gap around zero and they are separated from each other by the constant factor of 2^{L_e}.

The largest positive subnormal number is $0.111 \ldots 111 \times 2^{L_e}$ and the smallest positive normalized number is $1.000 \ldots 000 \times 2^{L_e}$. They are also separated from each other by a factor of 2^{L_e}.

Artificial Example of Floating-Point Representation

$$\overbrace{\boxed{1}}^{\text{sign}}\overbrace{\boxed{1}\boxed{1}\boxed{1}}^{\text{exponent}}\overbrace{\boxed{1}\boxed{1}\boxed{1}\boxed{1}}^{\text{significand}}$$

$$\overbrace{\boxed{1}}^{\text{sign}}\ \overbrace{\boxed{1}\boxed{1}\boxed{1}}^{\text{exponent}}\ \overbrace{\boxed{1}\boxed{1}\boxed{1}\boxed{1}}^{\text{significand}}$$

Bias is 3, $L_e = -2, U_e = +3.$

Artificial Example of Floating-Point Representation

$$\overbrace{1}^{\text{sign}}\;\overbrace{1\;1\;1}^{\text{exponent}}\;\overbrace{1\;1\;1\;1}^{\text{significand}}$$

Bias is 3, $L_e = -2, U_e = +3$.

sign	exp	frac	value	comment
0	000	0000	$+0.0$	Positive Zero
0	000	0001	$\frac{1}{16} \times 2^{-2}$	Smallest Subnormal
0	000	0010	$\frac{2}{16} \times 2^{-2}$	
			$\cdots$	
0	000	1111	$\frac{15}{16} \times 2^{-2}$	Largest Subnormal
0	001	0000	$\frac{16}{16} \times 2^{-2}$	Smallest Normalized
0	001	0001	$\frac{17}{16} \times 2^{-2}$	
			$\cdots$	
0	110	1111	$\frac{31}{16} \times 2^{+3}$	Largest Normalized
0	111	0000	$+\infty$	Plus Infinity
0	111	0001	NaN	Not a Number

Loss of Significance due to Catastrophic Cancellation

When we subtract two large numbers that are close to each other, the result will be much smaller. Since the precision is low for large numbers the result of the computation may have large relative error.

This is called catastrophic cancellation and is a source of major loss in precision for certain formulas. Whenever the scale of the numbers change in a computation we need to be extra careful to ensure a small relative error.

Example: Loss of Significance

The formula

$$\sqrt{x+1} - \sqrt{x}$$

involves catastrophic cancellation when x is large relative to 1.

The formula

$$\sqrt{x+1} - \sqrt{x}$$

involves catastrophic cancellation when x is large relative to 1. We can manipulate it to

$$(\sqrt{x+1} - \sqrt{x}) \left(\frac{\sqrt{x+1} + \sqrt{x}}{\sqrt{x+1} + \sqrt{x}} \right) = \frac{1}{\sqrt{x+1} + \sqrt{x}},$$

which is safe.

Example: Solving Quadratic Equations

The quadratic equation $ax^2 + bx + c = 0$ has the solutions

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \text{ and } x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

Example: Solving Quadratic Equations

The quadratic equation $ax^2 + bx + c = 0$ has the solutions

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \text{ and } x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

When $|4ac| \ll b^2$, one of the formulas

$$-b - \sqrt{b^2 - 4ac}, \text{ for } b < 0$$

$$-b + \sqrt{b^2 - 4ac}, \text{ for } b > 0$$

leads to catastrophic cancellation.

Example: Solving Quadratic Equations

The quadratic equation $ax^2 + bx + c = 0$ has the solutions

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \text{ and } x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

When $|4ac| \ll b^2$, one of the formulas

$$-b - \sqrt{b^2 - 4ac}, \ \text{ for } b < 0$$
$$-b + \sqrt{b^2 - 4ac}, \ \text{ for } b > 0$$

leads to catastrophic cancellation. One solution is to use the fact that

$$x_1 x_2 = \frac{(-b + \sqrt{b^2 - 4ac})(-b - \sqrt{b^2 - 4ac})}{4a^2} = \frac{c}{a}$$

and depending on the sign of b use this formula to calculate the problematic root.