

Understanding Errors in Computing

(Unavoidable)

Storage in any computer / device is finite.

This leads to the following effects:-

- floating point numbers line is only available as opposed to the real line.

- largest number exists
- smallest number exists

→

- machine precision
- Round-off errors.

— need to take care during Computation

— loss of significance

— "close enough"

— cancellation

DURING

+

=

-

Compu

Bit

- introduce Truncation Error when you transfer symbolic calculations to a computer.
- need to track Number of Computations

Computer storage

Bit - a short form for "binary digit"

- smallest unit of storage
- think of it as ON/OFF switch. which can store information.

Byte - 8 bits of storage make one byte.
original interest was to store language
characters - 8 bits was thought to
be enough to assign a unique value

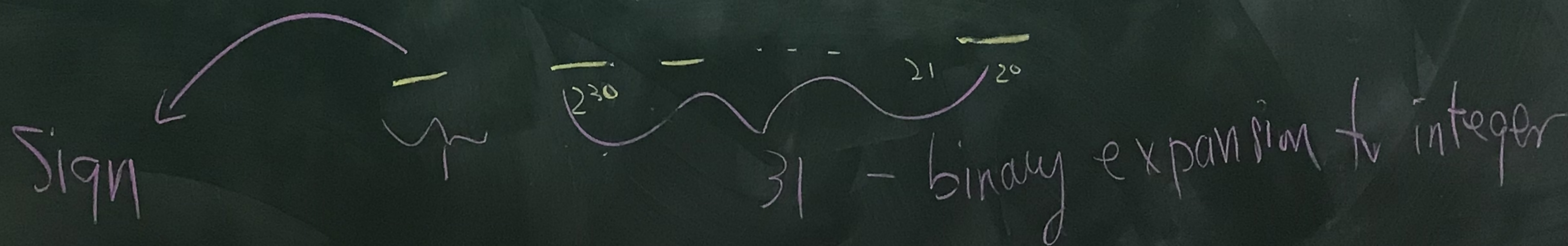
26	A - Z [capitals]
26	a - z [lower case]
32	? punctuation
10	0 - 9
114	~127

American
Standard
Code
Information
Interchange

Word - A machine 64 bit - # of bits computer processor
can handle in one operation

Storing integers in a computer

Integer - R stores integers in 32-bit storage



Largest integer - $2^{31} - 1$

e.g

$(111)_2$

$$= 2^2 + 2^1 + 2^0$$

$$= 2^3 - 1$$

$$= 7$$

← most significant
bit

→ least significant
bit

Floating point number line - default storage in R

I - institution

E - electrical

E - electronics

E - engineering

- double precision format

finite storage will result

Range

largest ^(positive) number = 1.797693×10^{308}

smallest ^(positive) number = $2.225074 \times 10^{-308}$

Precision
[49ps]

$0.8125 \iff$

11010
mantissa exponent

$1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 2^1$

$0.1 \iff$ Binary expansion is not finite

— lose some precision. As approximation have to be made while storing numbers,

— Most real numbers cannot be stored exactly

Round off

— Machine Precision — captures the gaps in the floating point line.

ϵ_M — is the smallest $\epsilon > 0$ such that
(floating point addition) $|1 + \epsilon| > 1$

$$\epsilon_M = 2.20446 \times 10^{-16}$$

Note
- ϵ_M - is not an small on the smallest number
that can be stored. This is a real limitation

x - a floating number - (ϵ_M^x - grows with x)

$$x + \epsilon_M^x > x$$

↓
directly proportional to
 x

Then many numbers cannot be represented by machines
Such numbers have to be approximated in representation
and so introduce errors called **ROUND-OFF ERROR!**

Accuracy

floating point comparison can only do
"close enough" instead of equality

• Absolute difference
 x, y $|x - y|$

Absolute Error

$$\epsilon_{ab} = |\alpha - \hat{\alpha}|$$

α - true
 $\hat{\alpha}$ - calculated

• Relative difference
$$\frac{|x - y|}{|x|}$$

Relative Error

$$\epsilon_{rel} = \frac{|\alpha - \hat{\alpha}|}{|\alpha|} \stackrel{\text{typical choice}}{=} \frac{|\alpha - \hat{\alpha}|}{|\alpha|}$$

Determine convergence

In any calculation with regard to comparison

- $x == y$

X - not to be used

- $abs(x-y) < \epsilon$

✓

↑
tolerance
larger than
machine
precision