Part 1. Modeling, Computers, and Error Analysis

Ch3. Approximation and Round-Off Errors &
Ch4. Truncation Errors and The Taylor Series

Lecture 2 & 3

Round-Off and Truncation Errors

3.4, 4.1

Homeyra Pourmohammadali

Sources of Numerical Errors

Round-Off

- Created due to approximate representation of numbers

Truncation

- Created by approximating mathematical procedure

Round-Off Error

It is related to how numbers are stored in a computer using binary digit or bit

Computer Representation of Numbers

Number system

- Base (e.g. base-10 system, base-2 system, …)

Integer representation

- Signed magnitude method

Floating-point representation

- Mantissa
- Exponent

Number Systems

- A convention for representing quantities, examples:

Base-2 system (binary)

0
1

Base-8 system (octal)

0
1
2
3
4
5
6
7

Base-10 system (decimal)

0
1
2
3
4
5
6
7
8
9

Example 1. Binary (base-2) number (as 8-bit word) and finding its equivalent decimal number

1	**0**	**1**	**0**	**1**	**1**	**0**	**1**

Integer Representation

Signed Magnitude Method:

- The first bit of a word indicate the sign
- The remaining bits are used to store numbers

1	0	0	0	0	0	0	0	1	0	1	0	1	1	0	1

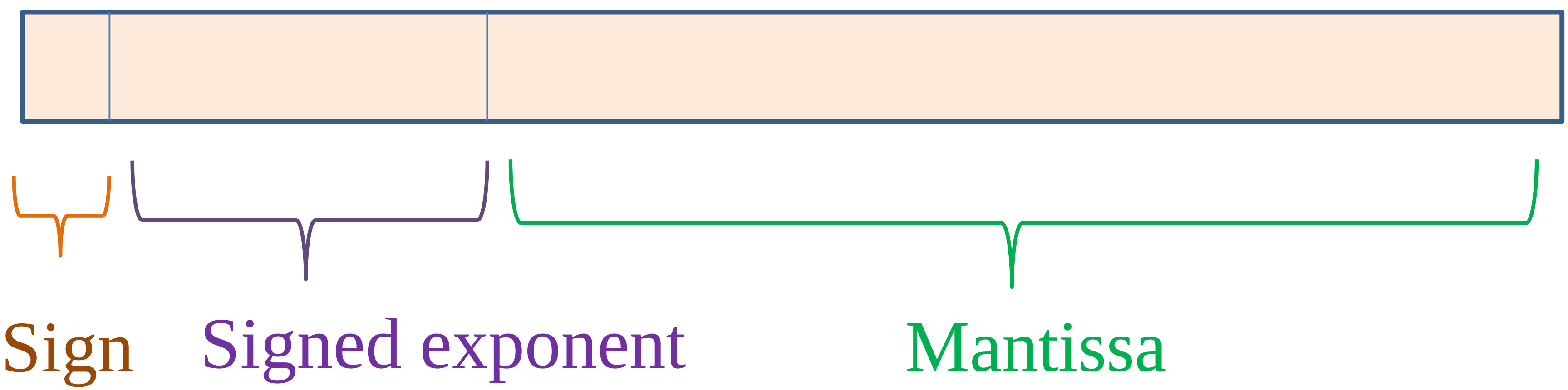
Sign Number

1 is for negative sign **-**
0 is for positive sign **+**

Decimal integer representation in a 16-bit computer (what number is it?)

Floating-Point Representation

Numbers are represented as:
- fractional part (called mantissa or significand) &
- integer part (called exponent or characteristics)



1 is for negative sign **-**
0 is for positive sign **+**

Decimal integer representation in a 16-bit computer (what number is it?)

Floating-Point Representation

$m.\ b^{e}$

m = mantissa
b = base of number system
e = exponent

e.g. 1/ 34 storage   1/34 = 0.029411765….. ?

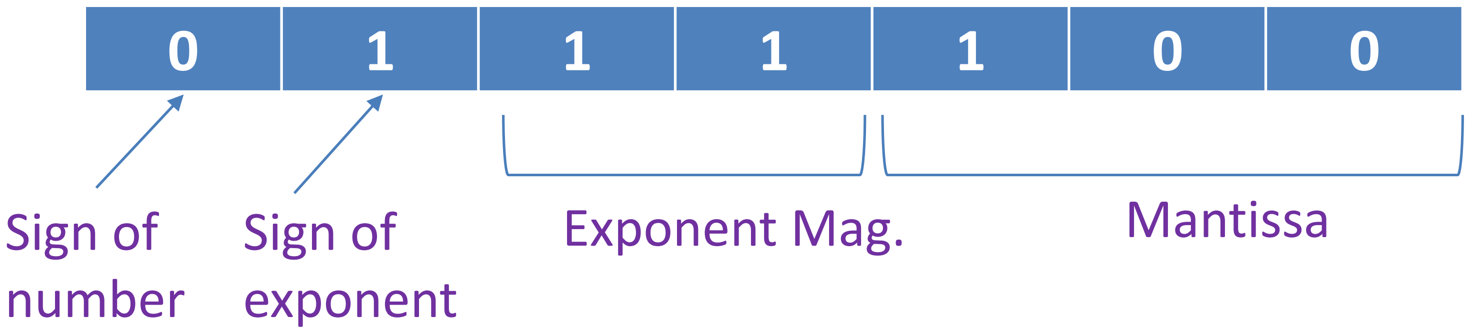
Issue of leading zero

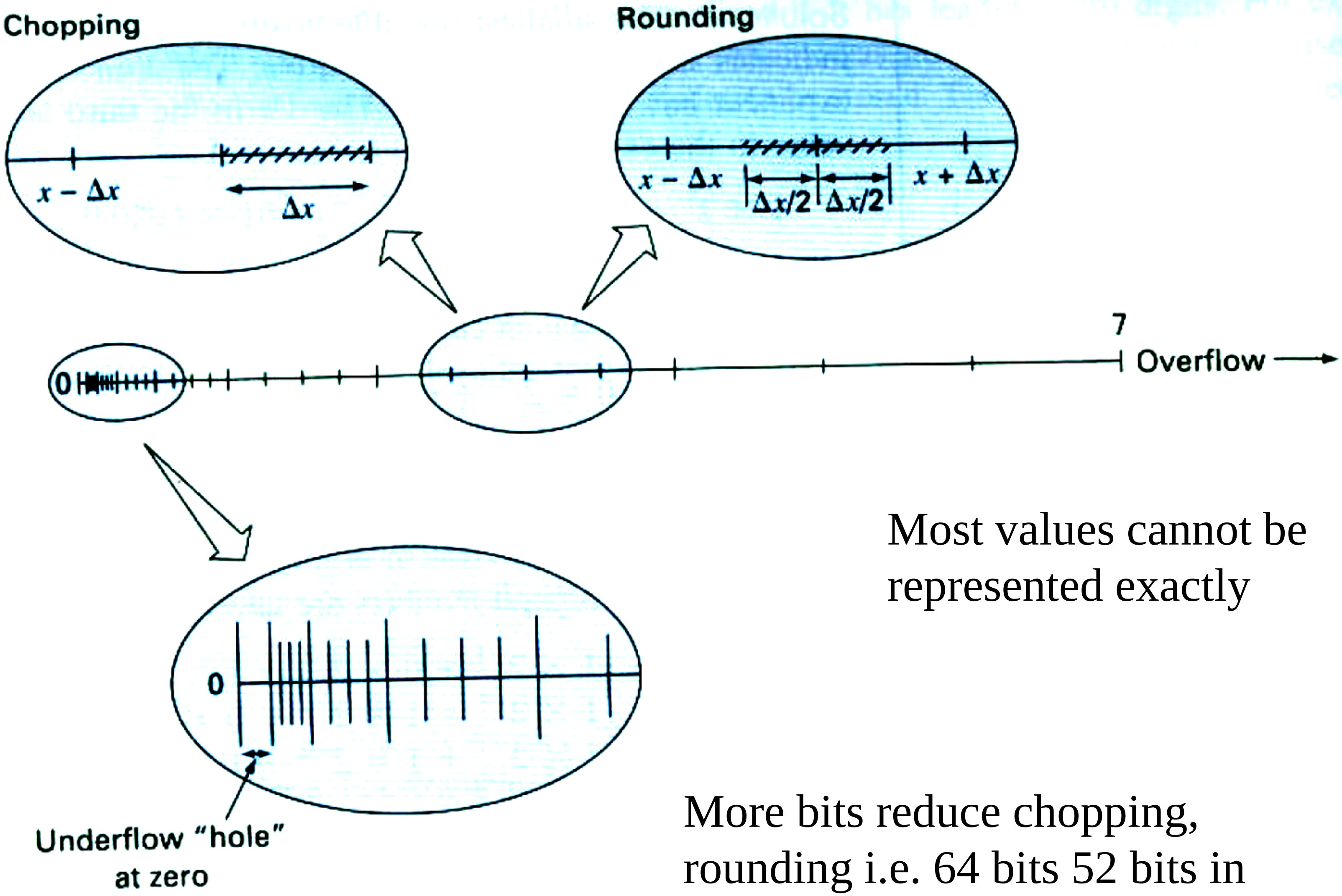
Normalization & its limitations

$$1/b \leq m < 1$$

Example 2. Consider 7-bit floating-point system

Example 2. Consider 7-bit floating-point system





Most values cannot be represented exactly

More bits reduce chopping, rounding i.e. 64 bits 52 bits in mantissa

Truncation Error

- Created by approximating mathematical procedure

Example 3. Truncation error due to approximation of mathematical procedure for differentiation

Taylor Series

- Predict a function value at one point in terms of the function value and its derivatives at another point

- The smooth function can be approximated as a polynomial

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + f''(x_i)\frac{h^2}{2!} + f'''(x_i)\frac{h^3}{3!} + \ldots$$

$$h = x_{i+1} - x_i$$

- zero-order approximation (considering only the first term)
- Higher-order approximation (considering higher terms)

Taylor Series Example

If f(a) , f'(a) , f'' (a) and f''' (a) is known and the higher order derivatives are zero, find f (a+2) using Taylor series.

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + f''(x_i)\frac{h^2}{2!} + f'''(x_i)\frac{h^3}{3!} + \ldots$$

$$h = x_{i+1} - x_i$$

Control of numerical errors, Overall approach

- True error (based on exact solution) not usually available, use approximate (estimated) error
- No systematic, general approach for error estimation for all problems – specific methods use different approaches
- A few guidelines
  - Avoid subtracting two nearly equal numbers
  - Don't add very small and very large numbers
- Error control methods
  - Sensitivity analysis, such as grid refinement study
  - Examine limiting cases