Lecture 2 - Outline

Monday, August 26, 2019 5:00 PM

HW 1 Overview

Matlab Coding Updates

Numerical Analysis: (notes adapted from P. Bauman & D. Salac)

- Objectives of Numerical Analysis
- Accuracy versus Precision
- Numerical Errors
- Significant Figures
- Number Systems
- Finite Precision Arithmetic
- Integer Representations
- Floating Point Representations
  - Single Precision
  - Double Precision $\longleftrightarrow$ Matlab

Numerical Analysis

Monday, August 26, 2019 5:52 PM

Objectives

Predict the behavior of some process or system (physical systems, social networks, finacial markets, political systems, ...), for which exact analytical results are not available.

Need to have confidence in these predictions

Need to know about potential errors in these models or the data

---

Two things to keep in mind:

"The purpose of computing is insight, not numbers," Richard Hamming, Numerical Methods for Scientists and Engineers, McGraw-Hill, 1962

"Prediction is hard, especially about the future," paraphrased from Niels Bohr, ..., Yogi Berra, old Danish proverb

Regarding errors in numerical analysis

- Rarely is input exact
- Algorithms introduce errors

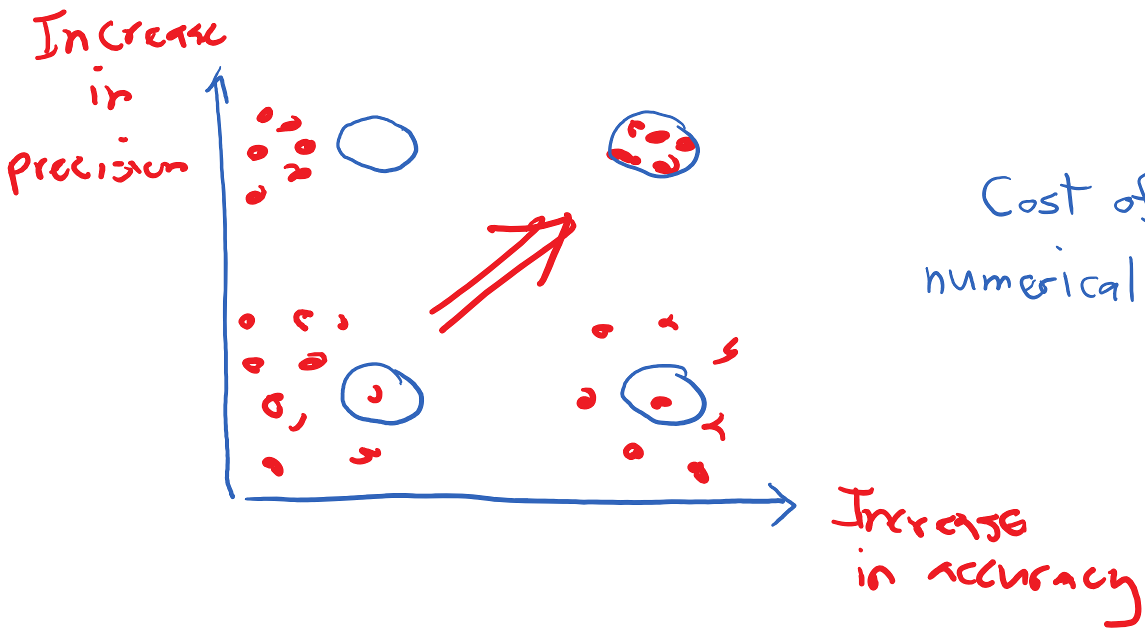
Results depend on both sources of error

How much error is present in our calculation and is that error tolerable?

This requires identification, quantification and minimization of error

Accuracy & Precision

Are these the same?



Cost of numerical analysis?

Numerical Errors

Due to model choice, numerical approximations, data

Depends upon complexity (or sensitivity) of system

Let x^* represent some true value

Then

$$x^* = x + e$$

x: approximate value

e: error

or $e = x^* - x$

Relative error

$$e_{rel} = \frac{x^* - x}{x^*} \qquad e_{rel\%} = e_{rel} \cdot 100\%$$

In practice, can we compute this error?

Iteration: x^{old}, x^{new}

$$e_{approx} = \frac{x^{old} - x^{new}}{x^{old}}$$

Can anything go wrong here?

Significant Figures (or Digits)

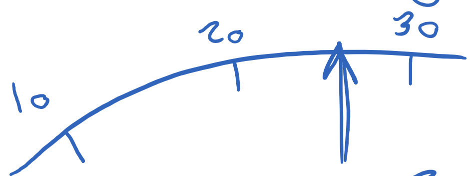
Simple concept, but ...

How much information do we really have?

Number of birds in a wildlife preserve

↳ coming & going all the time

Pressure in one of your tires



26?

26.5?

26.53?

Gauge is quite crude

Example:

Value		Significant Digits
23,500 →	2.35×10^4	3
	2.350×10^4	4
	2.3500×10^4	5

Example:

	Significant Digits
0.746	3
$0.0746 = 0.746 \times 10^{-1}$	3
0.00746	3

Leading zeros; following zeros?

Significant digits → how many digits you can use with confidence

Example: Division

$$\frac{6.72}{3.45} = ?$$

$$\frac{6.72}{3.45} = \underline{1.94}78 2608\ldots$$ in Matlab

What should be reported?

Chopping → 1.94 (3 significant digits)

$$\frac{6.729}{3.450} = \underline{1.95}0434\ldots$$

$$\frac{6.720}{3.459} = \underline{1.94}2758\ldots$$

bounds

Rounding → 1.95 (3 significant digits)

$$\frac{6.724}{3.445} = \underline{1.95}1814\ldots$$

$$\frac{6.715}{3.454} = \underline{1.94}4122\ldots$$

bounds

Number Systems

Base 10 (Decimal) $0, 1, 2, \ldots, 9$

60711

1×10^0	1
1×10^1	10
7×10^2	700
0×10^3	0000
6×10^4	60000
Base 10 $\leftarrow$	60711

Base 2 (Binary) $0, 1$

11101001

1×2^0	1
0×2^1	0
0×2^2	0
1×2^3	8
0×2^4	0
1×2^5	32
1×2^6	64
1×2^7	128
Base 10 $\leftarrow$	233

Base 8 (Octal) $0, 1, 2, \dots, 7$

Base 16 (Hexadecimal) $0, 1, 2, \dots, 9, \underset{10}{A}, B, \dots, \underset{15}{F}$

Representing numbers < 1 (mantissa)

Example: Base 10

$$0.9613 = 9\times10^{-1} + 6\times10^{-2} + 1\times10^{-3} + 3\times10^{-4}$$

Example: Base 2

$$0.1101 = 1\times2^{-1} + 1\times2^{-2} + 0\times2^{-3} + 1\times2^{-4}$$

$$= \frac{1}{2} + \frac{1}{4} + \frac{0}{8} + \frac{1}{16}$$

$$= \frac{8+4+1}{16} = \frac{13}{16}$$

Finite Precision Arithmetic

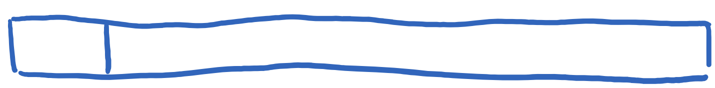
Computer limited to use a finite number of digits

Operations between these finite representations introduce round-off errors

How are numbers stored?

Integers

For simplicity, let's assume a single 8-bit word representation, having 8 pieces of base 2 data (i.e., 0 or 1 at each position)



sign bit

7 bits to provide the integer in base 2

$0 \rightarrow +$

$1 \rightarrow -$

$$\begin{array}{ccccccc} 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ & \downarrow & \downarrow & & \downarrow & \downarrow & \downarrow \\ & 2^5 & 2^4 & & 2^2 & 2^1 & 2^0 \end{array} = 32 + 16 + 4 + 2 + 1$$

$$= 55$$

Smallest integer $0000000_{\text{base } 2} = 0_{\text{base } 10}$

Largest integer $1111111_{\text{base } 2} = 127_{\text{base } 10}$

Along with sign bit → -127 to $+127$

but $+0 = -0$ Assign -0 to -128

∴ Overall range -128 to $+127$

for 8-bit representation

Note: This is $2^8 = 256$ distinct pieces of data, which is very limited! Early computers had 2 byte integers with $256^2 = 65536$ distinct pieces of data

Integer arithmetic is <u>exact</u>, except for division with a non-zero remainder

$\frac{12}{3} = 4$ exact $\frac{9}{2} = 4$ non-exact

Also, there is a possibility for underflow or overflow

$$-81 * 3 = -243 < -128$$

$$+44 + 113 = +157 > +127$$

Modern computers use either 32-bit or 64-bit representations for integers, where

32 bit: $2^{32} = 4,294,967,296$

64 bit: $2^{64} = (2^{32})^2 \cong 1.8446 \times 10^{19}$

distinct pieces of data

Same ideas for representation as above with 8-bits

Floating Point Representations (General)

Real (or floating point) numbers are stored using the IEEE 754 specification

sign	signed exponent	mantissa
$\pm$	$\underbrace{\pm\, \varepsilon_1 \varepsilon_2 \text{ ——— } \varepsilon_m}_{\varepsilon}$	$\underbrace{d_1 d_2 \text{ ———————— } d_p}_{M}$

e.g. 0.1101011

$$\therefore r = \pm M B^{\varepsilon}$$

B = base of number system

However, exponent can be shifted so that first digit is always non-zero

e.g. $0.05476 \times 10^2 = 0.5476 \times 10^1$

For binary system, this means first digit (or bit) is always 1, which can be assumed implicitly (Very clever!)

Single Precision Floating Point

Base 2 system

4 bytes, 8 bits per byte $\longrightarrow$ 32 bits

Sign 1 bit

Signed exponent 8 bits

Mantissa 23 (+ 1 implicit) bits

Signed exponent $\pm 2^7 \to \pm 128$

Mantissa $2^{24} = 10^{\alpha}$

$$24 \log_2 = \alpha \log_{10} 10$$

$$\therefore \alpha = 24 \log_{10} 2 = 7.22$$

Approximately 7 digits of precision

Precision $2^{-24} \cong 5.96 \times 10^{-8}$

Double Precision Floating Point

Base 2 system

8 bytes, 8 bits per byte $\to$ 64 bits

Sign 1 bit

Signed exponent 11 bits

Mantissa 52(+1 implicit) bits

Signed exponent $\pm 2^{10} = \pm 1024$

Mantissa $2^{53} = 10^{\beta}$

$$53 \log_{10} 2 = \beta \log_{10} 10$$

$$\therefore \beta \approx 53 \log_{10} 2 = 15.95$$

More than 15 digits of precision

Precision $2^{-52} \cong 2.22 \times 10^{-16}$

Note: In both single and double precision, space also is provided for $\pm\infty$ and NaN

Range Comparison

	Smallest	Largest
Single Precision	$\pm 1.17 \times 10^{-38}$	$\pm 3.40 \times 10^{38}$
Double Precision	$\pm 2.22 \times 10^{-308}$	$\pm 1.79 \times 10^{308}$