Introduction and Error Analysis

Objectives

Mathematical/Numerical Preliminaries

- Assess the difference between analytical and numerical solutions
- Understand the concepts of ***accuracy***, ***precision*** and ***significant figures***
- Recognize the difference between a number of error definitions
  - Absolute (True) error
  - Relative error
- Be familiar with the source and types of numerical errors
  - Round-off errors
  - Truncation errors
    - Review of the Taylor series
    - Error estimation

Numerical Analysis

Definition (Trefethen)

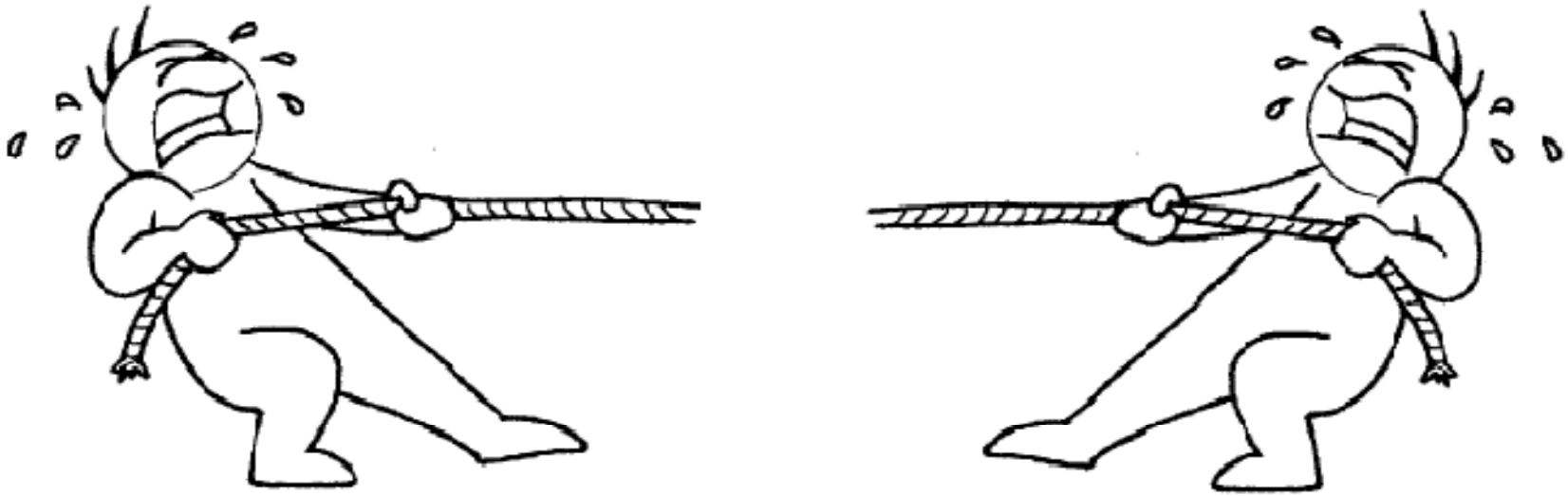
Study of algorithms for the problems of continuous mathematics

Method vs. Algorithm vs Implementation

- Method: a general description of a process
- Algorithm: a detailed description of the method
- Implementation: a particular instantiation of the algorithm

- Is it a “good” method?
- Is it a robust algorithm?
- Is it a fast implementation?

The Big Theme



Accuracy

Cost

Big Questions

- How algorithms work and how they fail
- Why algorithms work and why they fail

- Connects mathematics and computer science
- Need mathematical theory, computer programming, and scientific inquiry

Numerical Analysis

Numerical focus:

Approximation An approximation solution is sought. How close is this to the desired solution?

Efficiency How fast and cheap (memory) can we compute a solution?

Stability Is the solution sensitive to small variations in the problem setup?

Error What is the role of finite precition of our computers?

Unavoidable Errors in Computing

- Hardware problems:

  Example: Pentium™ Chip

Documentation is critical for any code that is not going to be used and immediately discarded. Documentation takes the form of comment statements that describe the input and output parameters of a function as well as the steps performed in the analysis.

Unavoidable Errors in Computing

- Some software bugs are caused by deterministic errors in the execution of the problem.

Example:

Problems in the built-in functions such as sine or cosine and a series of operational commands.

Matlab Numerical Problems

example

```
>format long e
>2.6 + 0.2
        ans = 2.800000000000e+000
>ans + 0.2
        ans = 3.000000000000e+000
>ans + 0.2
        ans = 3.200000000001e+000
                                ^
```

Note The program has changed the value of ‘ans’

Matlab Numerical Problems

Example

Same method but different results.

```
>format long e
>2.6 + 0.6
        ans = 3.200000000000e+000
>ans + 0.6
        ans = 3.800000000000e+000
>ans + 0.6
        ans = 4.400000000000e+000
>ans + 0.6
        ans = 5
```

Unavoidable Errors in Computing

- Numerical Errors are based on the mathematics of the problem.
  - Round-off Errors
  - Truncation Errors

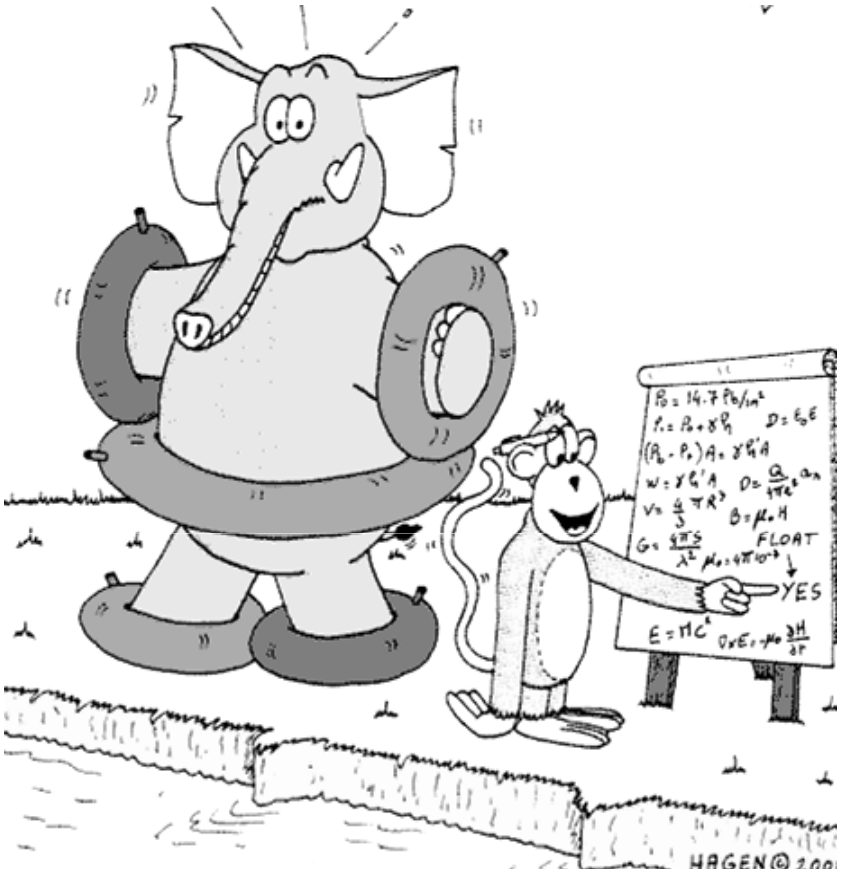
Numerical Errors

Example:

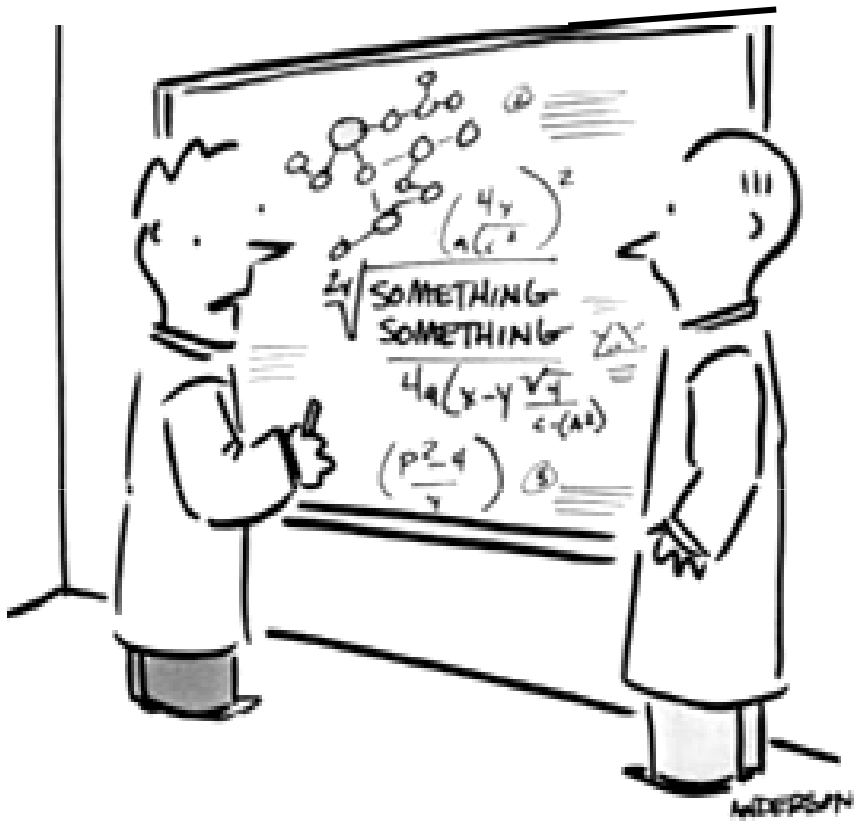
> x = tan(pi/6)
> y = sin(pi/6)/cos(pi/6)

where,

> x - y = 1.1102e-16 *????*

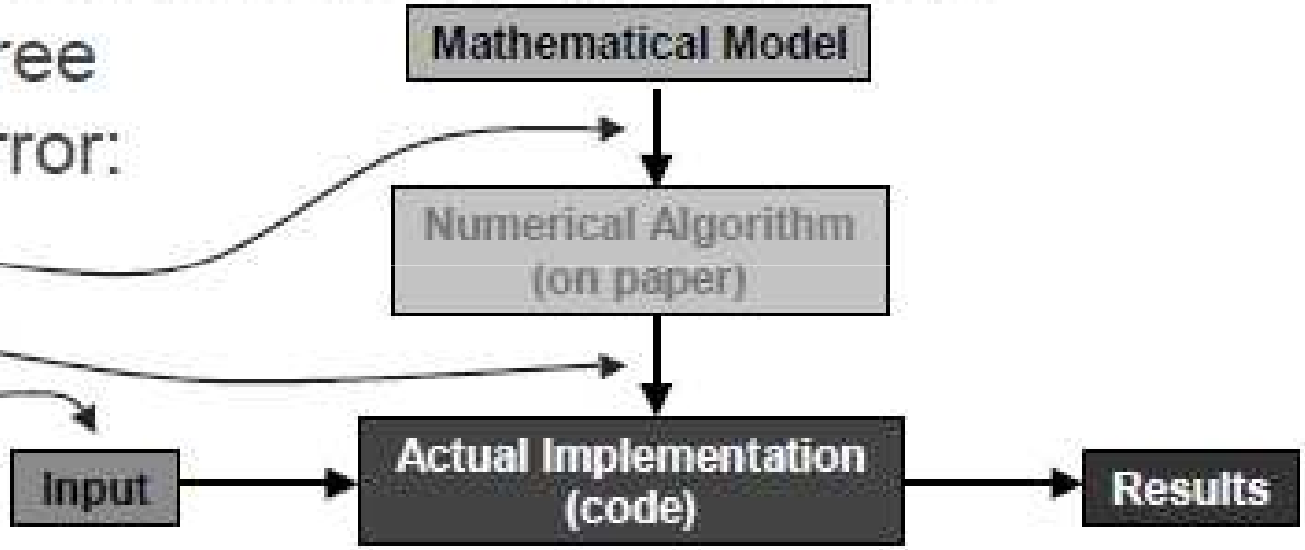


“According to my calculation,
you should float now ... I think ...”

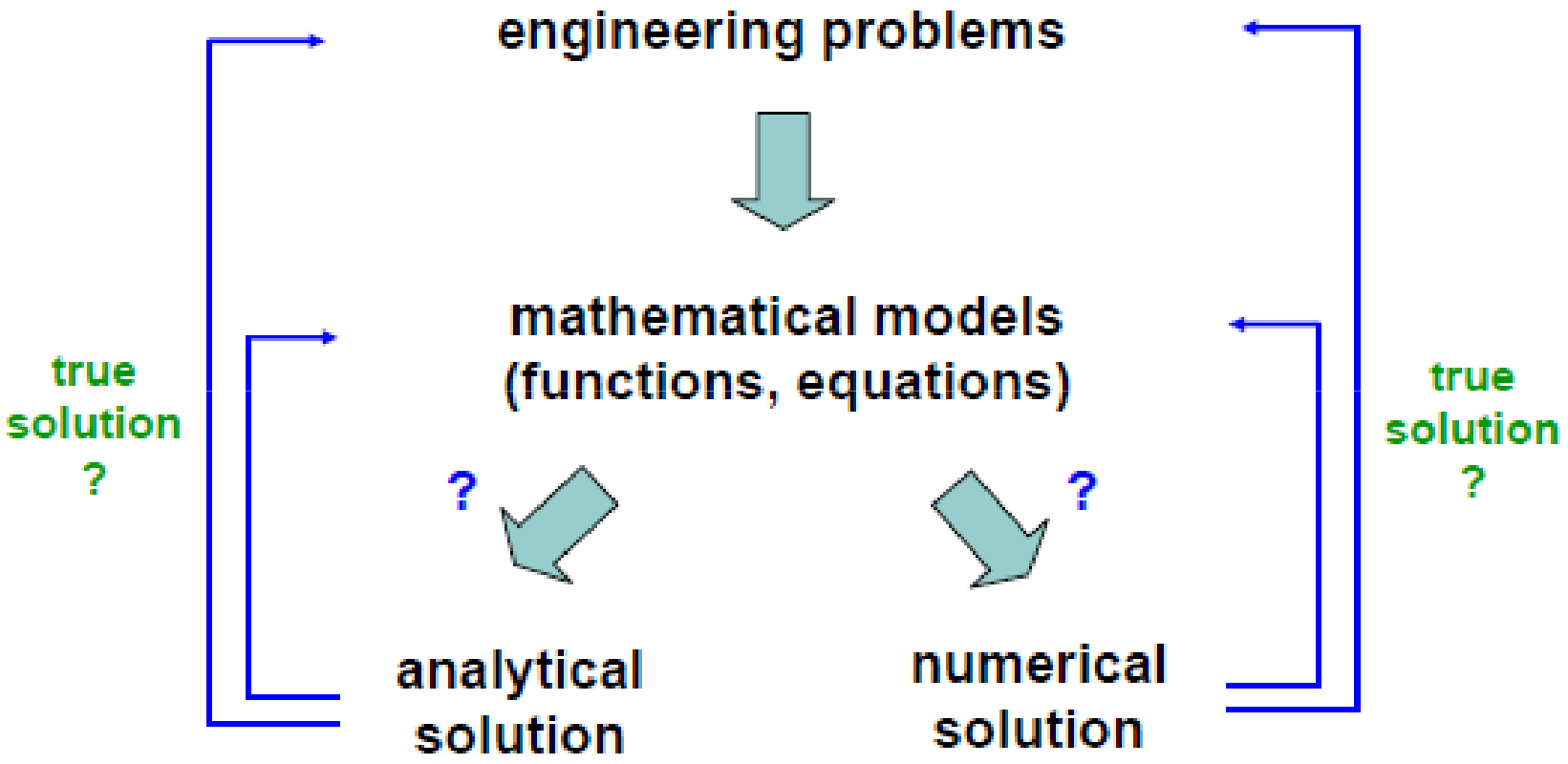


“It’s an inexact science.”

- We want fast, accurate and stable algorithms.
  - But all numerical algorithms produce inaccurate results.
- There are three sources of error:
  - Truncation
  - Rounding
  - Data



- And there is always a trade-off between speed & accuracy.


engineering problems
mathematical models
(functions, equations)
?
?
analytical
solution
numerical
solution
true
solution
?
true
solution
?

- Poor data.
  - It is very difficult to write code that will compensate for errors in your input data.
- Badly stored data:
  - The simplest failure can occur on the loading and saving of data.

  - Input too fine?
  - Output too coarse?

The concept of errors

True Value = Numerical Approximation + Error

Often not known

- Effective way to assess quantitatively the numerical methods
  - How close the numerical result with the exact analytical solution

Important part of numerical analysis

- Need to estimate the errors and develop criteria to determine if the approximation of the solution is acceptable

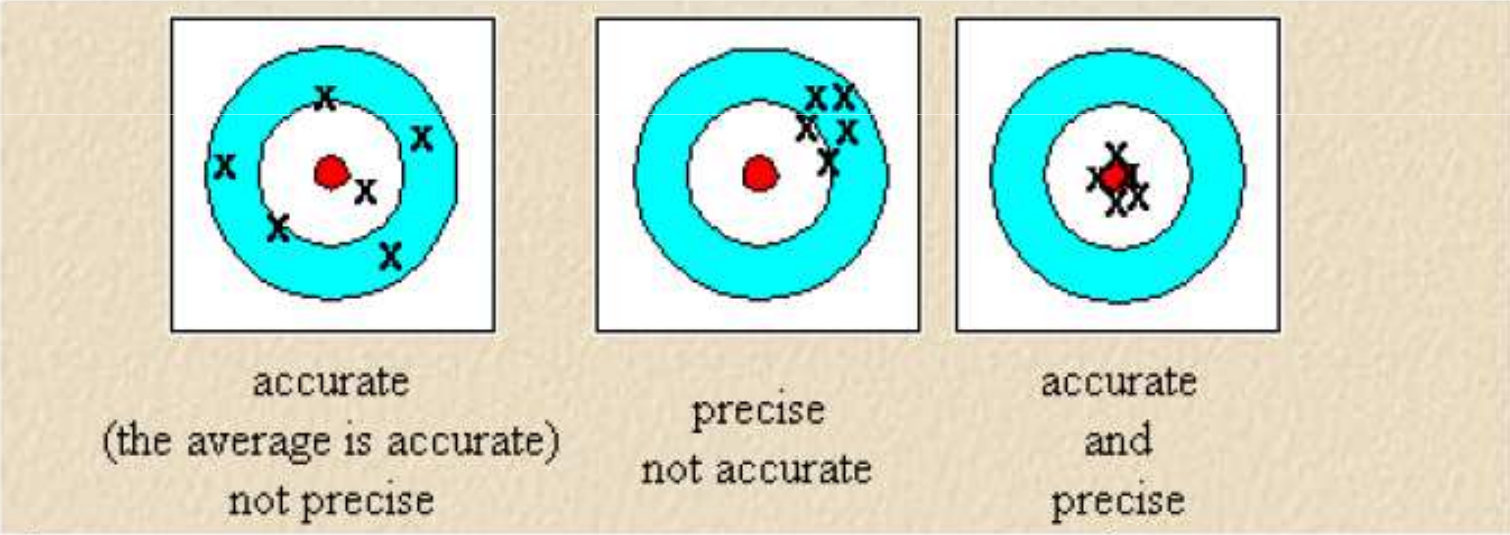
- Formulate the numerical methods to minimize the errors

Characterizing the errors

PRECISION* VERSUS *ACCURACY

Accuracy refers to how closely a measured value agrees with the correct value
Precision (or *reproducibility*) refers to how closely individual measurements agree with each other

Inaccuracy (or *bias*). A systematic deviation from the actual value.
Imprecision (or *uncertainty*). Magnitude of scatter.



In engineering problems, we try to minimize both imprecision and inaccuracy

Significant Figures

• Number of significant figures indicates precision. Significant digits of a number are those that can be *used* with *confidence*, e.g., the number of certain digits plus one estimated digit.

Rules For Significant Digits

- **Digits from 1-9 are always significant.**
- **Zeros between two other significant digits are always significant**
- **One or more additional zeros to the right of both the decimal place and another significant digit are significant.**

Error Definitions

True Value = Approximation + Error

E_t = True value (if known) – Approximation (+/-)

True error

$$\text{True fractional relative error} = \frac{\text{true error}}{\text{true value}}$$

$$\text{True percent relative error, } \varepsilon_t = \frac{\text{true error}}{\text{true value}} \times 100\%$$

Error Definitions

- For numerical methods, the true value will be known only when we deal with functions that can be solved analytically (simple systems). In real world applications, we usually not know the answer a priori. Then

$$\varepsilon_a = \frac{\text{Approximate error}}{\text{Approximation}} \times 100\%$$

- *Iterative approach:*

$$\varepsilon_a = \frac{\text{Current approximation - Previous approximation}}{\text{Current approximation}} \times 100\%$$

(+ / -)

Error Definitions

- Use absolute value
- Computations are repeated until stopping criterion (tolerance) is satisfied.

$$|\varepsilon_a| < \varepsilon_s$$

Pre-specified % tolerance based on the knowledge of your solution

- If the following criterion is met

$$\varepsilon_s = (0.5 \times 10^{(2-n)})\%$$

you can be sure that the result is correct to at least n significant figures. *(Scarborough 1966)*

Source of errors

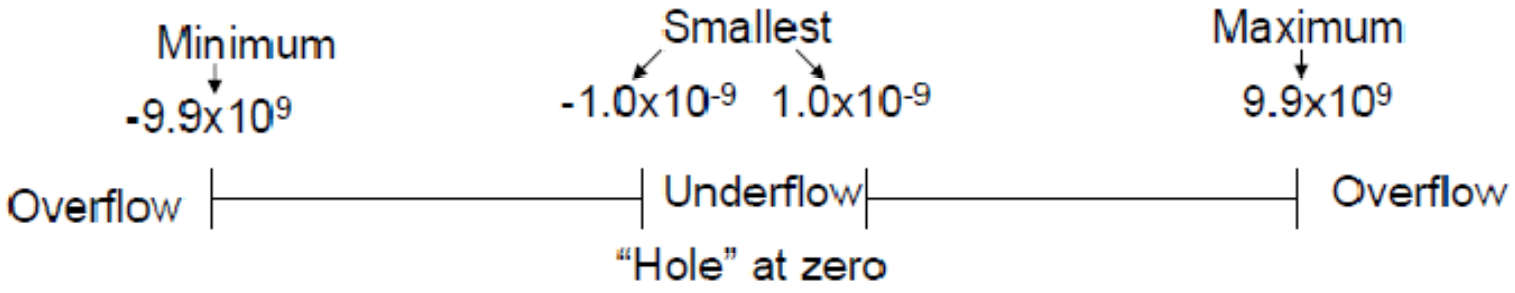
- Errors in mathematical modeling
- Blunders (bugs)
- Errors in inputs
- Round-off errors
- Truncation errors

"Human Error"

Specific to numerical methods

Round-off Errors

- Originate from the fact that computers retain only a fixed number of significant figures during calculation
  - *directly related to the manner in which numbers are stored in a computer*
- Instead of using decimal number system (or base-10) as we do, a computer uses a binary system (or base-2). Why?
  - this corresponds to the on/off positions of electronic components
- As the number of bits (binary digit) is limited some very large or very small numbers can not be represented. If you try to store a number outside this range you will generate an overflow error.



Round-off Errors

Limited range of quantities that may be represented

- Irrational numbers (π, e, or $\sqrt{7}$ cannot be represented exactly.
- Degree of precision is limited.

Example: How to deal the problem of π?

π= 3.141592653558

To be stored on a base-10 system carrying seven significant figures:
we can omit the figures after the seventh: π = 3.141592; this is called **chopping**.
This will generate an error of 0.0000065

Or

we can round the eighth figure: π = 3.141593
This will generate an error of -0.0000035
Therefore, **rounding** reduces the error

Roundoff errors

- Precision of representation of numbers is finite
  - errors accumulate
- a real number x can be represented as

$fl(x) = x \cdot (1+\varepsilon)$: floating point computer representation

$|fl(x)-x| = \varepsilon x$ absolute error (often also Δx)

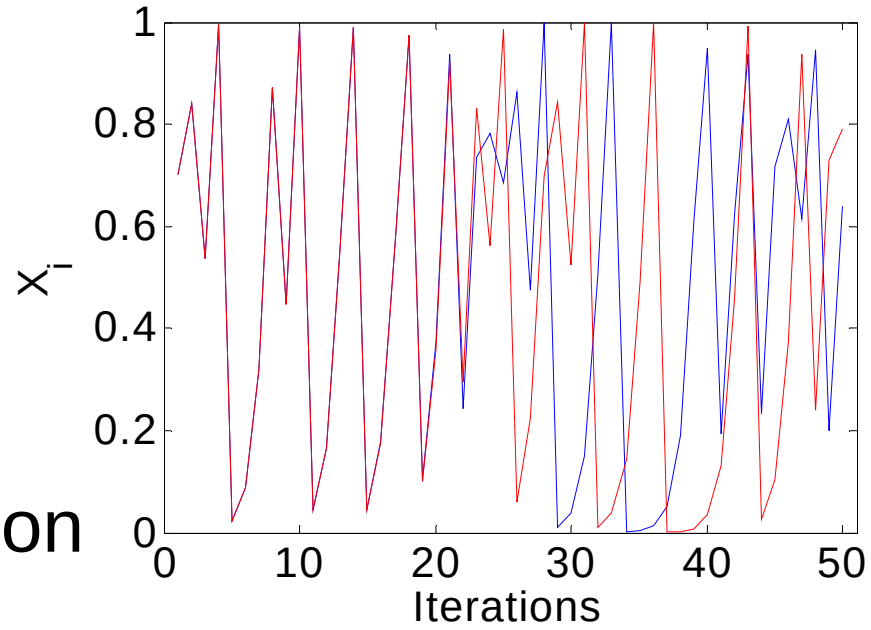
$|fl(x)-x|/x = \varepsilon$ relative error

Example: Logistic Map

$x_{i+1} = r * x_i * (1-x_i)$

$x_0 = 0.7$; $r = 4$

single and double precision



Uncertainty: well- or ill-conditioned?

Errors in input data can cause *uncertain* results

- input data can be experimental or rounded. leads to a certain variation in the results
- *well-conditioned*: numerical results are insensitive to small variations in the input
- *ill-conditioned*: small variations lead to drastically different numerical calculations (a.k.a. poorly conditioned)

What can we solve

- Suppose we want to evaluate *f(x) with perfect algorithm*
- we have FP number $x+\Delta x$ with error Δx

$\Delta f(x) = f(x+\Delta x)-f(x) \approx f'(x)\Delta x$ (if *f* differentiable)

relative error:

$$\frac{\Delta f(x)}{f(x)} \approx \frac{xf'(x)}{f(x)}\frac{\Delta x}{x}$$

Definition: condition number

$$\gamma(x) = \left|\frac{xf'(x)}{f(x)}\right|$$

- γ >>1 : problem ill-conditioned
- γ small: problem well-conditioned

well- and ill-conditioned methods

- Let's try a simple calculation:
  `99-70*sqrt(2)` (≈0.00505)
- suppose we have 1.4 as approximation for $\sqrt{2}$
- We have 2 mathematically equivalent methods:

  f_1: *99-70*$\sqrt{2}$ $f_1(1.4) = 1$

  f_2: *1/(99+70*$\sqrt{2}$)* $f_2(1.4) \approx 0.0051$

Condition numbers:

$f_1(x) =$ *99-70* x $\gamma_1(\sqrt{2}) \approx 20000$

$f_1(x) =$ *1/(99+70* x*)* $\gamma_2(\sqrt{2}) \approx 0.5$

what happened?

f_1: $99-70*\sqrt{2}$

f_2: $1/(99+70*\sqrt{2})$

- Condition number of subtraction, addition:

f(x)=x-a $\gamma_- = |-x/(x-a)|$ ill-conditioned for x-a≈0

f(x)=x+a $\gamma_+ = |x/(x+a)|$ ill-conditioned for x+a≈0

- Condition number for multiplication, division:

f(x)=ax $\gamma = |xa/(ax)| = 1$

f(x)=1/x $\gamma = |xx^{-2}/(x^{-1})| = 1$ well-conditioned

Loss of significance

- Computers stores only a finite number of digits. The consequence is that you may lose completely the significance of digits during calculations (arithmetic manipulations)

- Example, solve: $x^2 + 9^{12}x - 3 = 0$

- Solution given by:

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Matlab computes:
0
-2.8243 *1.0e+011

=> Is this correct ? NO

Why is this so ??

- Because in our example:

$$b^2 - 4ac \approx b^2$$

- Consequently we lose all significant digits when computing:

$$-b - \sqrt{b^2 - 4ac}$$

➢Can we avoid that by reformulating the quadratic formula?

- **Example:**
- Two solutions of quadratic equation

$$ax^2 + bx + c = 0$$

given by

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

- Naive use of formula can suffer overflow, or underflow, or severe cancellation
- Rescaling coefficients can help avoid overflow and harmful underflow
- Cancellation between -b and square root can be avoided by computing one root using alternative formula

$$x = \frac{2c}{-b \mp \sqrt{b^2 - 4ac}}$$

- Cancellation inside square root cannot be easily avoided without using higher precision

- **Example:**

$a = 0.05010$
$b = -98.78$
$c = 5.015$

(exact sol.= 1971.605916 and 0.05077069387)

$b^2 - 4ac = 9757 - 1.005 = 9756$
$\sqrt{b^2 - 4ac} = 98.77$

以四個有效位數計算

(1) $x = \dfrac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ sol.= $\dfrac{98.78 \pm 98.77}{0.1002} = 1972$ and 0.09980

(2) $x = \dfrac{2c}{-b \mp \sqrt{b^2 - 4ac}}$ sol.= $\dfrac{10.03}{98.78 \mp 98.77} = 1003$ and 0.05077

Arithmetic and Error

Summary:

-- how does error arise

-- how machines do arithmetic

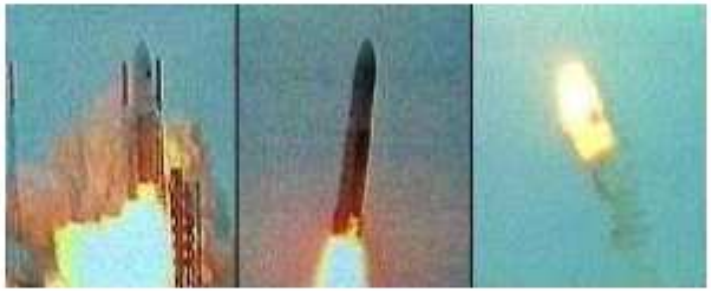
    -- fixed point arithmetic
    -- floating point arithmetic

-- how errors are propagated in calculations.

-- how to measure error

Additional Information

On June 4, 1996 an unmanned Ariane 5 rocket launched by the European Space Agency exploded just forty seconds after its lift-off from Kourou, French Guiana. The rocket was on its first voyage, after a decade of development costing $7 billion. The destroyed rocket and its cargo were valued at $500 million. A board of inquiry investigated the causes of the explosion and in two weeks issued a report. It turned out that the cause of the failure was a software error in the inertial reference system. Specifically a 64 bit floating point number relating to the horizontal velocity of the rocket with respect to the platform was converted to a 16 bit signed integer. The number was larger than 32,767, the largest integer storable in a 16 bit signed integer, and thus the conversion failed.



Banking problem

Foreign money conversion

For example: CAD → US

After multiplying the exchange rate, show only the last two decimal digit in the bankbook. → chopping off to the penny

Accumulating $0.00999999 for million of million of transaction

-> Banker becomes a millionaire