CS 325
Numerical Computing

OR
Numerical Methods
Numerical Analysis

Introduction to Numerical Computing Methods

- WHAT IS NUMERICAL **COMPUTING ?**
- WHY DO WE NEED THEM?

Numerical Computing :

- is study of Algorithms that are used to obtain numerical (approximate) solutions of a mathematical problem.

- is concerned with how to solve a problem numerically, i.e., how to develop a sequence of numerical calculations to get a satisfactory answer.

Why do we need them?

1. No analytical solution exists,
2. An analytical solution is difficult to obtain

Course Outline:

1- Error analysis:

2- Solution(Root) of equations in one variable:

3-Interpolation and Polynomial approximation:

4-Numerical differentiation and Integration:

5-Differential Equations:

6-Direct Method for solving linear system:

7-Iterative Techniques for solving linear system:

8-Difference Operator analysis:

Text Book: Numerical Analysis , Burden and Faires , 9th Ed

Number Representation and Accuracy

- NUMBER REPRESENTATION
- NORMALIZED FLOATING POINT REPRESENTATION
- SIGNIFICANT DIGITS
- BITS AND BYTE
- ACCURACY AND PRECISION
- SINGLE AND DOUBLE PRECISION
- ALGORITHM AND FLOW CHART

- ROUNDING AND CHOPPING
- ABSOLUTE , RELATIVE AND PERCENTAGE ERROR
- LOSS OF SIGNIFICANCE

READING ASSIGNMENT:

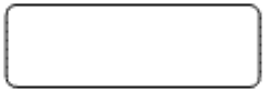
Algorithm:

- To write a logical step-by-step method to solve the problem is called algorithm, in other words,
- An algorithm includes calculations, reasoning and data processing.

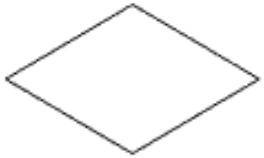
Iteration is the repetition of a process in order to generate a sequence of outcomes.

Flow chart :

A flowchart is the graphical or pictorial representation of an algorithm with the help of different symbols



Start/stop



Decision

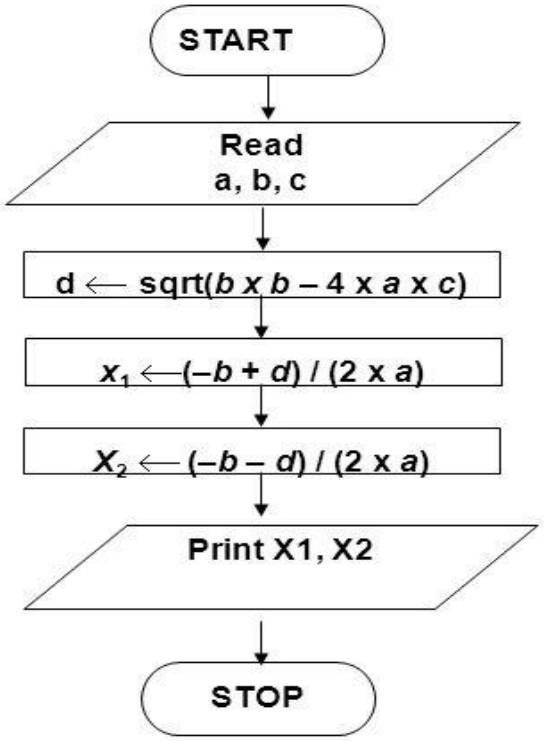
Input or data

Process or action

Example:

Problem: Write Algorithm and Flowchart to find solution of Quadratic equation

- **Algorithm:**
- Step 1: Start
- Step 2: Read a, b, c
- Step 3: $d \leftarrow \text{sqrt}(b \times b - 4 \times a \times c)$
- Step 4: $x1 \leftarrow (-b + d) / (2 \times a)$
- Step 5: $x2 \leftarrow (-b - d) / (2 \times a)$
- Step 6: Print $x1$, $x2$
- Step 7: Stop



Representing Real Numbers

You are familiar with the decimal system:

$$312.45 = 3\times10^{2} + 1\times10^{1} + 2\times10^{0} + 4\times10^{-1} + 5\times10^{-2}$$

Decimal System: Base = 10 , Digits (0,1,…,9)

Standard Representations:

$$\pm \quad 3\ 1\ 2 \quad . \quad 4\ 5$$

sign integral part . fraction part

Normalized Floating Point Representation

$$\pm \quad \underline{d.\ f_1\ f_2\ f_3\ f_4} \times 10^{\pm n}$$

sign mantissa exponent

$d \neq 0, \qquad \pm n$: signed exponent

- Scientific Notation: Exactly one non-zero digit appears before decimal point.
- Advantage: Efficient in representing very small or very large numbers.

Binary System:

Binary System: Base = 2, Digits {0,1}

$$\pm \; \underline{1.\; f_1 \;\; f_2 \;\; f_3 \;\; f_4} \;\times\; 2^{\pm n}$$

sign mantissa signed exponent

$$(1.101)_2 = (1 + 1\times 2^{-1} + 0\times 2^{-2} + 1\times 2^{-3})_{10} = (1.625)_{10}$$

$$(1.1)_{10} = (1.0001100110011001100\ldots)_2$$

You can never represent 1.1 exactly in binary system.

Significant digits are those digits that can be used with confidence.

Rules:

- **Non zero numbers are always significant**

  1.23 45.6 6,7263

- **In between zeros are always significant**

  1.005 70206

- **Leading zeros are never significant**

  0.0055 0.0302

- **Trailing zeros are some time significant**

  70,000 70,000. 1,030 1030.0000

FLOATING POINT REPRESENTATION OF REAL NUMBERS

The IEEE standard consists of a set of binary representations of real numbers. A **floating point number** consists of three parts: the **sign** (+ or −), a **mantissa**, which contains the string of significant bits, and an **exponent**. The three parts are stored together in a single computer **word**.

The bits are divided among the parts as follows:

precision	sign	exponent	mantissa
single	1	8	23
double	1	11	52
long double	1	15	64

The Institute of Electrical and Electronics Engineers (IEEE)

IEEE 754 Floating-Point Standard

Single and double precision

Single Precision (32 bit)

23 bits used for significant digits

8 bit used for store exponent

1 bit used for to store sign (+,-)

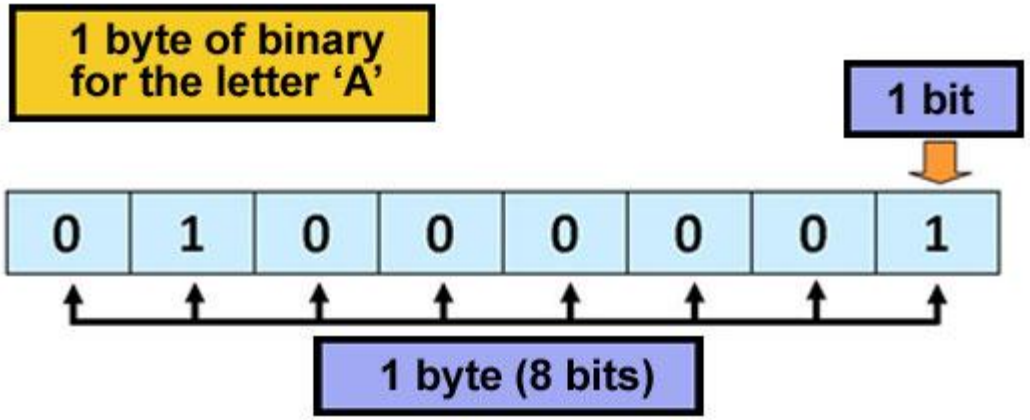
Double precision: (64 bit)

52 for significant digits ,

11 bit for exponent

1 bit for sign

ASCII table :



Bits and byte :

The **byte** is a unit of digital information that most commonly consists of eight **bits**.

bits used to encode a single character of text in a **computer**

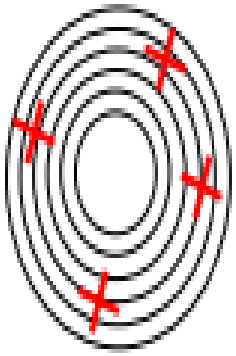
How to Convert Bits and Bytes:

- 8 bits = 1 byte
- 1,024 bytes = 1 kilobyte
- 1,024 kilobytes = 1 megabyte
- 1,024 megabytes = 1 gigabyte
- 1,024 gigabytes = 1 terabyte

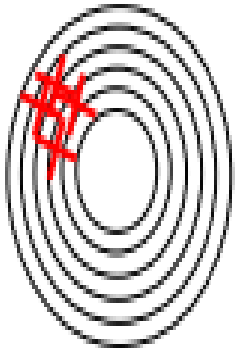
Accuracy and Precision

- Accuracy is related to the closeness to the true value.
- Precision is related to the closeness to other estimated values.

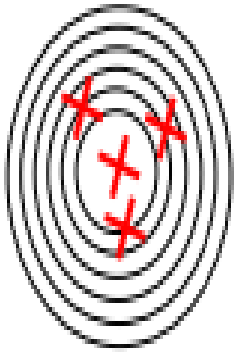
Nether Precise NOR accurate:



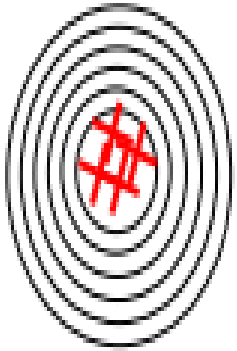
Precise, but NOT accurate:



Accurate but NOT precise:



Precise AND accurate



Rounding and Chopping

Rounding: Replace the number by the nearest

machine number. OR

its impossible to represent all real numbers exactly on machine with finite

Chopping: Throw all or drop the extra digits.

Error: is difference between an approximation of number used in computation and its exact value

OR Error = True value – approximate value

Example: Round vs Chop

√2 =1.41421356237309504880116872

∏ = 3.14159265358979323846264338

∏round =3.1416

∏chop =3.1415

ERROR Analysis:

Truncation Error:

are when an iterative method is terminated

OR mathematical procedure is approximated and

approximate solution differs from exact solution

Discretization Error :

are committed when a solution

of discrete problem does not coincide with solution

of continuous problem

Error in CM – True Error

Can be computed if the true value is known:

$$\text{Absolute Error}:$$

$$AE = \left| \text{true value} - \text{approximation} \right|$$

$$\text{Absolute Relative Error}:$$

$$ARE = \left| \frac{\text{true value} - \text{approximation}}{\text{true value}} \right|$$

Error in CM – Estimated Error

When the true value is not known:

Estimated Absolute Error

$$AE = \left| \text{current estimate} - \text{previous estimate} \right|$$

Estimated Absolute Relative Error

$$ARE = \left| \frac{\text{current estimate} - \text{previous estimate}}{\text{current estimate}} \right|$$

Loss of significance:

occurs in numerical calculations when too many significant digits cancel

$$\begin{array}{r} 123.4567 \\ -\ 123.4566 \\ \hline 000.0001 \end{array}$$

Example

Calculate $\sqrt{9.01} - 3$ on a three-decimal-digit

$$\begin{aligned} \sqrt{9.01} - 3 &= \frac{(\sqrt{9.01} - 3)(\sqrt{9.01} + 3)}{\sqrt{9.01} + 3} \\ &= \frac{9.01 - 3^2}{\sqrt{9.01} + 3} \\ &= \frac{0.01}{3.00 + 3} = \frac{.01}{6} = 0.00167 \approx 1.67 \times 10^{-3}. \end{aligned}$$

Avoiding loss of significance:

i. Rationalizing

Consider the function

$$f(x) = \sqrt{(x^2 + 1)} - 1$$

We see that near zero, there is a potential loss of significance.

However, the function can be rewritten in the form

$$f(x) = (\sqrt{(x^2 + 1)} - 1)\left(\frac{\sqrt{(x^2 + 1)} + 1}{\sqrt{(x^2 + 1)} + 1}\right)$$

$$= \frac{x^2}{\sqrt{(x^2 + 1)} + 1}$$

ii. Using series expansion

Consider the function

$$f(x) = x - \sin x$$

whose values are required near $x = 0$. We can avoid the loss of significance Taylor series for $\sin x$

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} - \dots$$

For x near zero, the series converges quite rapidly.

We can now rewrite the function f as

$$f(x) = x - \left(x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} - \dots \right) = \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} - \dots$$

iii. Using trigonometric identities

As a simple example, consider the function

$$f(x) = \cos^2(x) - \sin^2(x)$$

There will be loss of significance at $x = \pi/4$.

The problem can be solved by the simple substitution

$$\cos^2(x) - \sin^2(x) = cos(2x)$$

Example

$$E_1 = \frac{1 - \cos x}{\sin^2 x} \quad \text{and} \quad E_2 = \frac{1}{1 + \cos x}$$

x	E_1	E_2
1.00000000000000	0.64922320520476	0.64922320520476
0.10000000000000	0.50125208628858	0.50125208628857
0.01000000000000	0.50001250020848	0.50001250020834
0.00100000000000	0.50000012499219	0.50000012500002
0.00010000000000	0.49999999862793	0.50000000125000
0.00001000000000	0.50000004138685	0.50000000001250
0.00000100000000	0.50004445029134	0.50000000000013
0.00000010000000	0.49960036108132	0.50000000000000
0.00000001000000	0.00000000000000	0.50000000000000
0.00000000100000	0.00000000000000	0.50000000000000
0.00000000010000	0.00000000000000	0.50000000000000
0.00000000001000	0.00000000000000	0.50000000000000
0.00000000000100	0.00000000000000	0.50000000000000

Example:

Consider using the Taylor series approximation for e^x to evaluate e^{-5}:

$$e^{-5} = 1 + \frac{(-5)}{1!} + \frac{(-5)^2}{2!} + \frac{(-5)^3}{3!} + \frac{(-5)^4}{4!} + \dots$$

Degree	Term	Sum	Degree	Term	Sum
0	1.000	1.000	13	-0.1960	-0.04230
1	-5.000	-4.000	14	0.7001E-1	0.02771
2	12.50	8.500	15	-0.2334E-1	0.004370
3	-20.83	-12.33	16	0.7293E-2	0.01166
4	26.04	13.71	17	-0.2145E-2	0.009518
5	-26.04	-12.33	18	0.5958E-3	0.01011
6	21.70	9.370	19	-0.1568E-3	0.009957
7	-15.50	-6.130	20	0.3920E-4	0.009996
8	9.688	3.558	21	-0.9333E-5	0.009987
9	-5.382	-1.824	22	0.2121E-5	0.009989
10	2.691	0.8670	23	-0.4611 E-6	0.009989
11	-1.223	-0.3560	24	0.9607 E-7	0.009989
12	0.5097	0.1537	25	-0.1921 E-7	0.009989

Table. Calculation of $e^{-5} = 0.006738$ using four-digit decimal arithmetic

There are loss-of-significance errors in the calculation of the sum. To avoid the loss of significance is simple in this case:

$$e^{-5} = \frac{1}{e^5} = \frac{1}{\text{series for } e^5}$$

and form e^5 with a series not involving cancellation of positive and negative terms.

Example 2 Show that $x^5 - 2x^3 + 3x^2 - 1 = 0$ has a solution in the interval $[0, 1]$.

Solution Consider the function defined by $f(x) = x^5 - 2x^3 + 3x^2 - 1$. The function f is continuous on $[0, 1]$. In addition,

$$f(0) = -1 < 0 \qquad \text{and} \qquad 0 < 1 = f(1).$$

The Intermediate Value Theorem implies that a number x exists, with $0 < x < 1$, for which $x^5 - 2x^3 + 3x^2 - 1 = 0$. ■

Example Find both roots of the quadratic equation $x^2 + 9^{12}x = 3$.

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}. \qquad x = \frac{-9^{12} \pm \sqrt{9^{24} + 4(3)}}{2}.$$

$$x_1 = -2.824 \times 10^{11},$$

$$x_2 = \frac{-9^{12} + \sqrt{9^{24} + 4(3)}}{2},$$

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

$$x_2 = 1.062 \times 10^{-11},$$

Page-25,26

The quadratic formula states that the roots of $ax^2 + bx + c = 0$, when $a \neq 0$, are

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \text{and} \quad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}. \tag{1.1}$$

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}\left(\frac{-b - \sqrt{b^2 - 4ac}}{-b - \sqrt{b^2 - 4ac}}\right) = \frac{b^2 - (b^2 - 4ac)}{2a(-b - \sqrt{b^2 - 4ac})},$$

which simplifies to an alternate quadratic formula

$$x_1 = \frac{-2c}{b + \sqrt{b^2 - 4ac}}. \tag{1.2}$$

$$x_2 = \frac{-2c}{b - \sqrt{b^2 - 4ac}}. \tag{1.3}$$

The loss of accuracy due to round-off error can often be avoided by a reformulation

Decimal Machine Numbers

normalized *decimal* floating-point form

$$\pm 0.d_1 d_2 \ldots d_k \times 10^n, \quad 1 \leq d_1 \leq 9, \quad \text{and} \quad 0 \leq d_i \leq 9,$$

for each $i = 2, \ldots, k$. Numbers of this form are called k-digit *decimal machine numbers*.

$$y = 0.d_1 d_2 \ldots d_k d_{k+1} d_{k+2} \ldots \times 10^n.$$

$$fl(y) = 0.d_1 d_2 \ldots d_k \times 10^n.$$ Chopping

Rounding

For rounding, when $d_{k+1} \geq 5$, we add 1 to d_k to obtain $fl(y)$;

$d_{k+1} < 5$, we simply chop off all but the first k digits; so we *round down*.

Example 1 Determine the five-digit (a) chopping and (b) rounding values of the irrational number π.

Definition 1.15 Suppose that p^* is an approximation to p. The **absolute error** is $|p - p^*|$, and the **relative error** is $\dfrac{|p - p^*|}{|p|}$, provided that $p \neq 0$. ■

Example 2 Determine the absolute and relative errors when approximating p by p^* when

(a) $p = 0.3000 \times 10^1$ and $p^* = 0.3100 \times 10^1$;

(b) $p = 0.3000 \times 10^{-3}$ and $p^* = 0.3100 \times 10^{-3}$;

(c) $p = 0.3000 \times 10^4$ and $p^* = 0.3100 \times 10^4$.

Finite-Digit Arithmetic

$$x \oplus y = fl(fl(x) + fl(y)),$$
$$x \ominus y = fl(fl(x) - fl(y)),$$

Example 3 Suppose that $x = \frac{5}{7}$ and $y = \frac{1}{3}$.

Use five-digit chopping for calculating $x + y$, $x - y$, $x \times y$, and $x \div y$.

Table 1.2

Operation	Result	Actual value	Absolute error	Relative error
$x \oplus y$	0.10476×10^{1}	22/21	0.190×10^{-4}	0.182×10^{-4}
$x \ominus y$	0.38095×10^{0}	8/21	0.238×10^{-5}	0.625×10^{-5}
$x \otimes y$	0.23809×10^{0}	5/21	0.524×10^{-5}	0.220×10^{-4}
$x \oslash y$	0.21428×10^{1}	15/7	0.571×10^{-4}	0.267×10^{-4}

Display results in tabular form

Example 5

Let $p = 0.54617$ and $q = 0.54601$. Use four-digit arithmetic to approximate $p - q$ and determine the absolute and relative errors using **(a)** rounding and **(b)** chopping.

$$\frac{|r - r^*|}{|r|} = \frac{|0.00016 - 0.0002|}{|0.00016|} = 0.25,$$

$$\frac{|r - r^*|}{|r|} = \frac{|0.00016 - 0.0001|}{|0.00016|} = 0.375,$$

Display results in tabular form

Nested Arithmetic

Example 6 Evaluate $f(x) = x^3 - 6.1x^2 + 3.2x + 1.5$ at $x = 4.71$ using three-digit arithmetic.

Three-digit (chopping): $f(4.71) = ((104. - 134.) + 15.0) + 1.5 = -13.5,$

Three-digit (rounding): $f(4.71) = ((105. - 135.) + 15.1) + 1.5 = -13.4.$

$$\text{Chopping: } \left|\frac{-14.263899 + 13.5}{-14.263899}\right| \approx 0.05, \quad \text{and Rounding: } \left|\frac{-14.263899 + 13.4}{-14.263899}\right| \approx 0.06.$$

Nested Arithmetic

$$f(x) = x^3 - 6.1x^2 + 3.2x + 1.5 = ((x - 6.1)x + 3.2)x + 1.5.$$

$$f(4.71) = ((4.71 - 6.1)4.71 + 3.2)4.71 + 1.5 = ((-1.39)(4.71) + 3.2)4.71 + 1.5$$
$$= (-6.54 + 3.2)4.71 + 1.5 = (-3.34)4.71 + 1.5 = -15.7 + 1.5 = -14.2.$$

Three-digit (chopping): $\left|\dfrac{-14.263899 + 14.2}{-14.263899}\right| \approx 0.0045;$

three-digit rounding answer of -14.3.

Three-digit (rounding): $\left|\dfrac{-14.263899 + 14.3}{-14.263899}\right| \approx 0.0025.$

Nesting has reduced the relative error for the chopping approximation to less than 10% of that obtained initially. For the rounding approximation the improvement has been even more dramatic; the error in this case has been reduced by more than 95%. □

Exercise 1.2

5. Use three-digit rounding arithmetic to perform the following calculations. Compute the absolute error and relative error with the exact value determined to at least five digits.

a. $133 + 0.921$

b. $133 - 0.499$

c. $(121 - 0.327) - 119$

d. $(121 - 119) - 0.327$

e. $\dfrac{\frac{13}{14} - \frac{6}{7}}{2e - 5.4}$

f. $-10\pi + 6e - \dfrac{3}{62}$

13. Use four-digit rounding arithmetic and the formulas (1.1), (1.2), and (1.3) to find the most accurate approximations to the roots of the following quadratic equations. Compute the absolute errors and relative errors.

a. $\frac{1}{3}x^2 - \frac{123}{4}x + \frac{1}{6} = 0$

b. $\frac{1}{3}x^2 + \frac{123}{4}x - \frac{1}{6} = 0$

c. $1.002x^2 - 11.01x + 0.01265 = 0$

d. $1.002x^2 + 11.01x + 0.01265 = 0$

Motivation:

To introduce modern approximation techniques; to explain how, why, and when they

can be expected to work; and to provide a foundation for further study of numerical

analysis and scientific computing.

What is Numerics?

Numerical Mathematics:

- Part of (applied) mathematics.
- Designing computational methods for continuous problems mainly from linear algebra (solving linear equation systems, finding eigenvalues etc.) and calculus (finding roots or extrema etc.).
- Often related to approximations (solving differential equations, computing integrals) and therefore somewhat atypical for mathematics.
- Analysis of numerical algorithms: memory requirements, computing time, if approximations: accuracy of approximation.

Numerical Programming:

- Branch of computer science.
- Efficient implementation of numerical algorithms (memory efficient, fast, considering hardware settings (e.g. cache), parallel).

Application of Numerical Computing

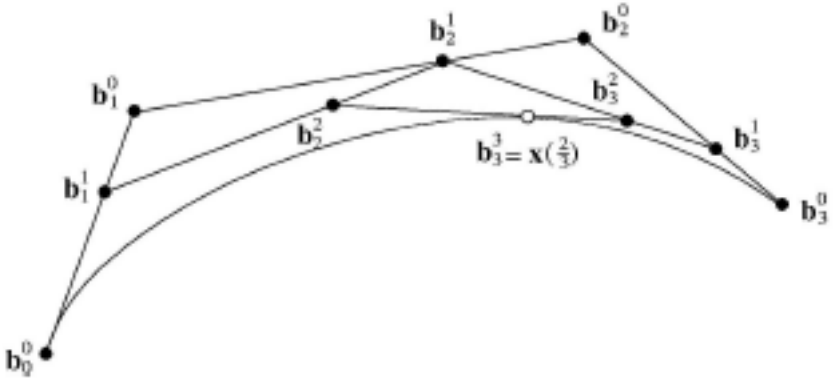
Numerical Computing

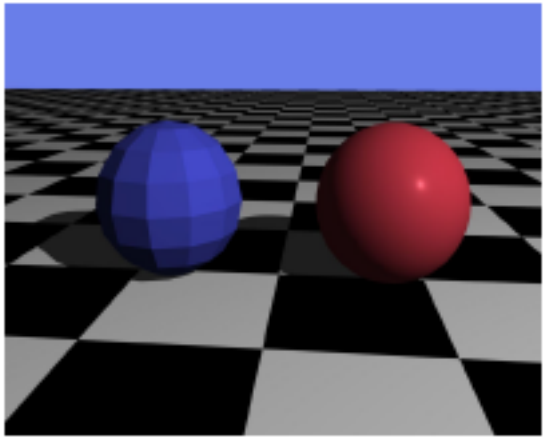
area of mathematics and computer science that creates, analyzes, and implements algorithms for obtaining numerical solutions to problems involving continuous variables. Such problems arise throughout the natural sciences, social sciences, engineering, medicine, and business.

Since the mid 20th century, the growth in power and availability of digital computers has led to an increasing use of realistic mathematical models in science and engineering, and numerical analysis of increasing sophistication is needed to solve these more detailed models of the world. The formal academic area of numerical analysis ranges from quite theoretical mathematical studies to computer science issues.

Geometric Modeling

- **Geometric modeling** or **CAGD (Computer-Aided Geometric Design)** deals with the modeling of geometric objects on a computer (car bodies, dinosaurs for Jurassic Park, . . .).
- Especially for **nonlinear curves and surfaces** there are a number of numerical methods including efficient algorithms for their generation and modification:
  - **Bézier curves and surfaces**
  - **B-spline curves and surfaces**
  - **NURBS** (Non-Uniform Rational B-Splines)
- Such methods are based on the methods of interpolation from Chapter 2.



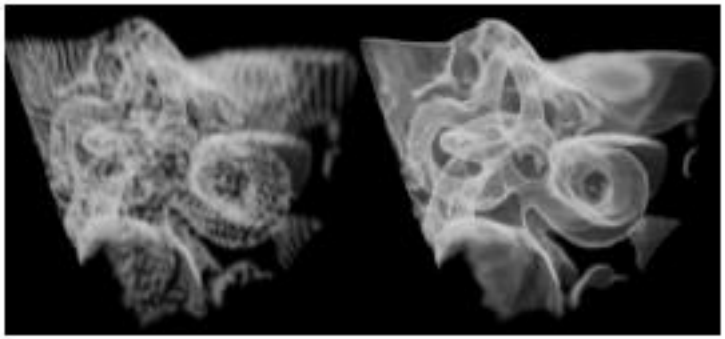


Computer Graphics

- **Computer graphics** is a very computationally intensive branch of computer science:
  - In the context of **ray tracing**, to compute highlight and reflection effects, a huge number of intersection points of rays with objects of the scenery have to be computed – which leads to the problem of solving a system of linear or nonlinear equations (see Chapters 4 and 6).
  - In the context of the **radiosity method** for computing diffuse illumination, a large linear system of equations is constructed which usually has to be solved iteratively – this is covered in Chapter 6.
  - All computer games or flight simulations require very powerful numerical algorithms due to their real time characteristics.

Visualization

- **Visualization** has developed from a branch of computer graphics to an independent domain. In visualization, numerical computations are carried out in numerous places:
  - **Particle Tracing** is a possibility to visualize numerically simulated flows. Here, many virtual particles are brought into the computed flow field to make it visible on the basis of their paths (vertices etc.). To compute the paths of the particles, ordinary differential equations have to be solved. We will learn more about methods to accomplish this in Chapter 5.
  - **Volume visualization** deals with the visualization of three-dimensional data, for example from the area of medicine. To make far away details visible the intensities along the rays are integrated – through numerical methods such as those described in Chapter 3.



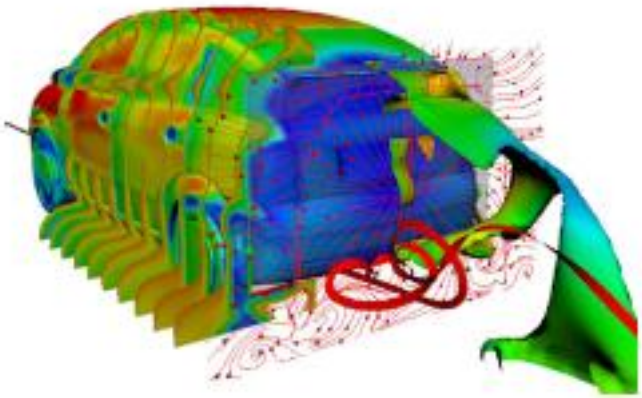
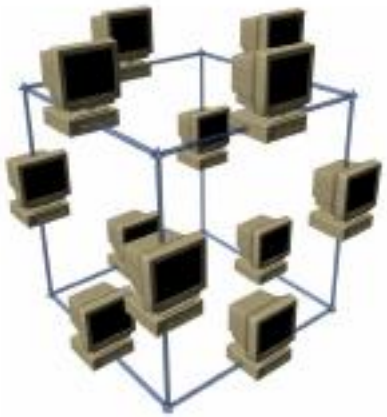


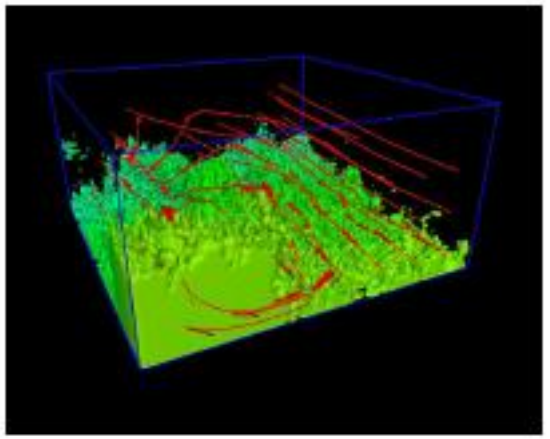
Image Processing

- **Image processing** without numerical methods is also unthinkable. Almost all filters and transformations are numerical algorithms, most times related to the **fast Fourier transformation (FFT)**.
- In addition, most methods for **image compression** (such as JPEG) rely on numerical transformations (discrete cosine transformation, wavelet transformation)
- We will have a quick look at those transformations in Chapter 4.

Numerical Simulation & High Performance Computing

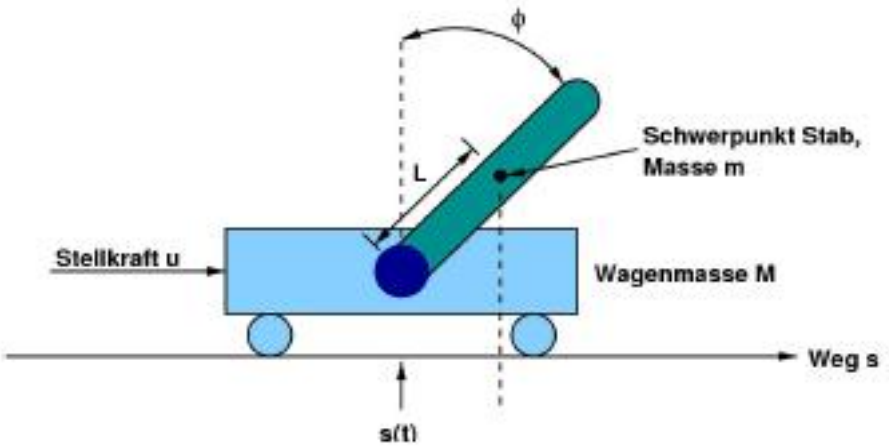
- The links to numerics are nowhere as strong as in **High-Performance Scientific Computing**, i.e. the numerical simulation on high-performance computers.
- Supercomputers spend a major part of their lives with numerical calculations, that's why they are trimmed especially on floating point performance – a core topic for us in CSE!
- Here, efficient methods to solve differential equations numerically are needed – a first foretaste of this will be given in Chapter 8.

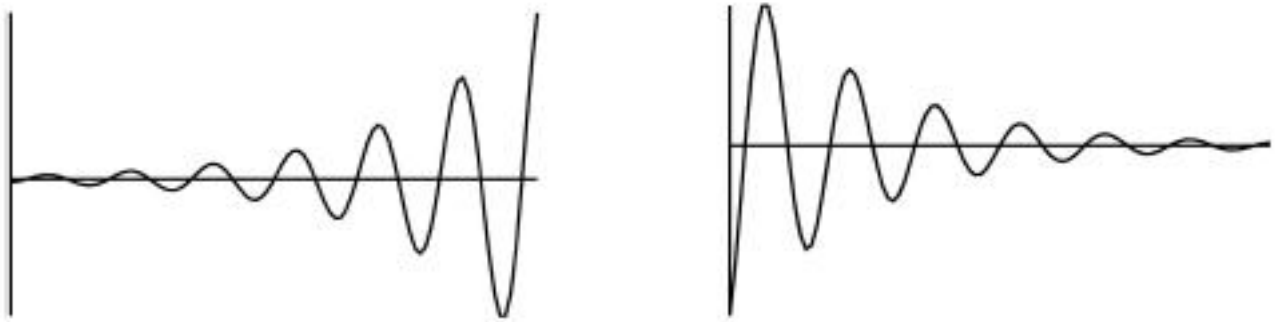




Control

- Process computers in particular have to deal with **control**.
- One possible mathematical description of control processes uses ordinary differential equations whose numerical solution will be discussed in Chapter 5.





building up

damped oscillation