Numerical Computation

People

Instructor:

- Prof. Zhanchuan Cai 蔡占川
- Email: zccai@must.edu.mo
- Room: A323
- Tel: 8897-2329

Office Hours:

- Mon 15:00 - 17:30
- TUES 15:00 - 17:30
- Wed 15:00 - 18:00
- THU 15:00 - 17:00

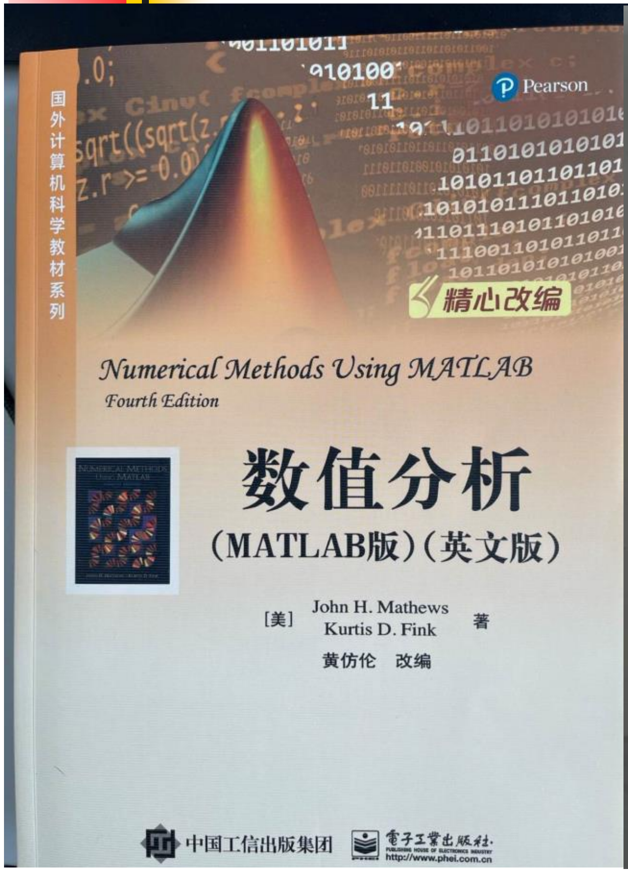
Course Information

- *Lecture Hours and Venues*
  - MON 9:00 – 11:50 B401 D1
  - WED 9:00 – 11:50 C408 D3
- *Course FTP*

Cleve B. Moler

Numerical Computing with MATLAB

SIAM (Society for Industrial and Applied Mathematics)
Philadelphia
2004
ISBN 0-89871-560-1



Appendix B

Some Math Expressions and

一些数学表达式及其读法

Math Expressions	Pronunciations
1/2	A half; one half
1/3	A third; one third
2/3	Two thirds
1/4	A quarter; one quarter
1/1234	One over one thousand two hundred and thirty-four
$2\frac{1}{2}$	Two and a half
$2\frac{7}{8}$	Two and seven over eight
0.1	O point one; zero point one
0.01	O point o one; zero point zero one
60 mi/hr	Sixty miles per hour
20°	Twenty degrees
0°C	Zero degree Centigrade
$6 \times 5 = 30$	Six times (multiplied by) five equals (is equal to; are; makes; make) thirty
$s = vt$	s equals (is equal to) v multiplied by t
1 : 2	The ratio of one to two
$a + b = c$	a plus b is (are; equals; is equal to) c
$c - b = a$	c minus b is (equals; is equal to) a

Appendix C

1000 English-Chinese Math Key Words

1000 英汉数学词汇[1]

A

abacus /'æbəkəs/ 算盘

abscissa /æb'sɪsə/ 横坐标

absolute /'æbsəlu:t/ 绝对

absorb /əb'zɔ:b/ 吸收

abstract /'æbstrækt/ 抽象，摘要

accelerate /ək'seləreɪt/ 加速度

accumulate /ə'kju:mjəleɪt/ 聚积

accuracy /'ækjərəsi/ 准确，精确

acnode /'æknəʊd/ 孤立点

acute /ə'kju:t/ 锐

addend /ə'dend/ 加数

addition /ə'dɪʃn/ 加法

adjacent /ə'dʒeɪsnt/ 邻接的

adjoint /ə'dʒɔɪnt/ 伴随的

affine /ə'faɪn/ 仿射

algebra /'ældʒɪbrə/ 代数（学）

algorithm /'ælgərɪðəm/ 算法

alternate /ɔ:l'tɜ:nət/ 替换，交错

altitude /'æltɪtju:d/ 高度，顶垂

amount /ə'maʊnt/ 总数

amplitude /'æmplɪtju:d/ 幅度，幅角，振幅

analogue /'ænəlɒg/ 模拟

analysis /ə'næləsɪs/ 分析

analytic /ˌænə'lɪtɪk/ 解析的

angle /'æŋgl/ 角

anisotropic /ˌænaɪsə(ʊ)'trɒpɪk/ 各向异性的

antitone /'ænti'təʊn/ 反序

apex /'eɪpeks/ 顶点

apostrophe /ə'pɒstrəfi/ 撇号

appendix /ə'pendɪks/ 附录

application /ˌæplɪ'keɪʃn/ 应用

approximate /ə'prɒksɪmət/ 逼近，近似

arbitrary /'ɑ:bɪtrəri/ 任意的

① 本附录音标主要参考了 www.oxfordlearnersdictionaries.com 和金山词霸给出的英式音标。

Numerical Computation

課程考核 Assessment Method

課堂參與Participation and Test	10%
作業Assignment	15%
期中考試Mid-term Exam	25%
期末考試Final Exam	50%
總計Total	100%

What is numerical computation?

Numerical Analysis is concerned with the design and analysis of algorithms for solving mathematical problems that arise in many fields, especially science and engineering.

----Michael T. Heath

Numerical analysis is the study of algorithms for the problems of continuous mathematics.

----Lloyd N. Trefethen

The construction of numerical algorithms:
calculation formulas and algorithm steps

Theoretical analysis of the algorithm:

Error analysis
Convergence
Stability
etc.

ALGORITHMS AND FLOWCHARTS

- A typical programming task can be divided into two phases:
- ***Problem solving phase***
  - produce an ordered sequence of steps that describe solution of problem
  - this sequence of steps is called an ***algorithm***
- ***Implementation phase***
  - implement the program in some programming language

Steps in Problem Solving

- First produce a general algorithm (one can use ***pseudocode***)
- Refine the algorithm successively to get step by step detailed ***algorithm*** that is very close to a computer language.
- ***Pseudocode*** is an artificial and informal language that helps programmers develop algorithms. Pseudocode is very similar to everyday English.

Pseudocode & Algorithm

- **Example :** Write an algorithm to determine a student's final grade and indicate whether it is passing or failing. The final grade is calculated as the average of four marks.

Pseudocode & Algorithm

Pseudocode:

- *Input a set of 4 marks*
- *Calculate their average by summing and dividing by 4*
- *if average is below 50*

  Print "FAIL"

  else

  Print "PASS"

Pseudocode & Algorithm

- Detailed Algorithm
- Step 1: Input M1,M2,M3,M4

  Step 2: GRADE $\leftarrow$ (M1+M2+M3+M4)/4

  Step 3: if (GRADE < 50) then

  Print "FAIL"

  else

  Print "PASS"

  endif

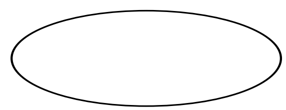
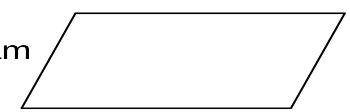
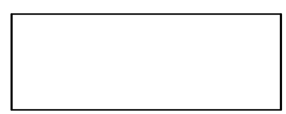
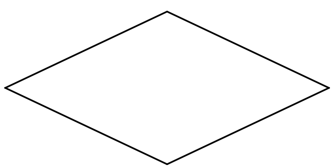
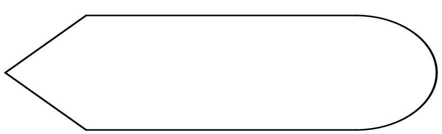
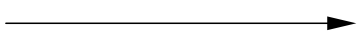
The Flowchart

A Flowchart

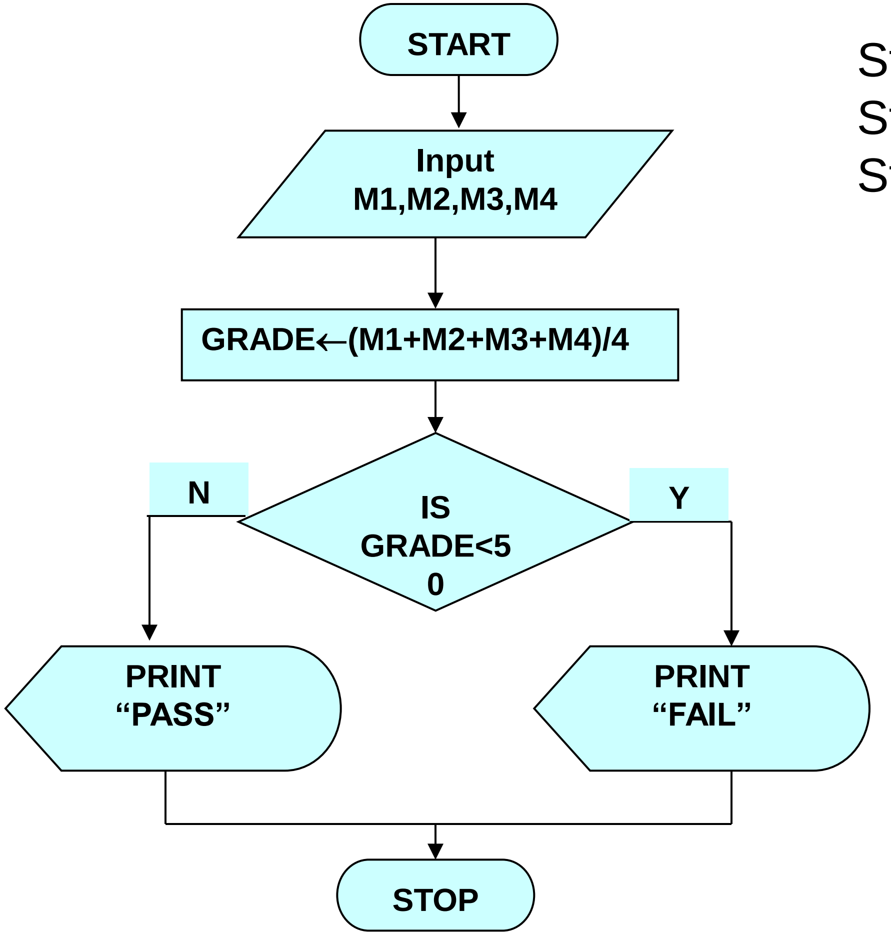
- shows logic of an algorithm
- emphasizes individual steps and their interconnections
- e.g. control flow from one action to the next

Flowchart Symbols

Basic

Name	Symbol	Use in Flowchart
Oval		Denotes the beginning or end of the program
Parallelogram		Denotes an input operation
Rectangle		Denotes a process to be carried out e.g. addition, subtraction, division etc.
Diamond		Denotes a decision (or branch) to be made. The program should continue along one of two routes. (e.g. IF/THEN/ELSE)
Hybrid		Denotes an output operation
Flow line		Denotes the direction of logic flow in the program

Example



Step 1: Input M1,M2,M3,M4
Step 2: GRADE ← (M1+M2+M3+M4)/4
Step 3: if (GRADE <50) then
Print “FAIL”
else
Print “PASS”
endif

Content

1 Matlab Language

2 Solutions of Nonlinear Equations

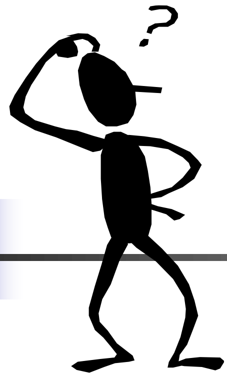
3 Methods for Solving Linear Systems

4 Interpolation and Polynomial Approximation

5 Curve Fitting

6 Numerical Integration

7 Methods for Solving Initial-Value Problems of Ordinary Differential Equations



What is the purpose of Numerical Computation?

Input

$\sqrt{x}$, a^x, $\ln x$, $A\vec{x}=\vec{b}$,

$\int_a^b f(x)dx$, $\frac{d}{dx}f(x)$, ……

→ Numerical Analysis → $+ \ - \ \times \ \div$ → **Computer** → Approximate Solution

The basic idea of constructing numerical algorithm

Approximate substitution**近似替代**

Discretization**离散化**

Recurrence**递推化**

Course features

Practicality, Theory and Practice

实用性　理论性　实践性

1. Facing the computer, provide feasible and effective algorithms according to the characteristics of the computer;

- Only provide addition, subtraction, multiplication and division and logical operations
- Serial machine and parallel machine

2. Reliable theoretical analysis: the convergence, stability and error analysis of the algorithm;

3. Good computational complexity: time and space complexity;

4. There are sufficient numerical experiments to prove the effectiveness of the algorithm.

Approximate Substitution近似替代

Approximate substitute近似替代

Example 1： To approximate the value of the irrational number e.

Solve:

$$\because e^x = 1 + x + \frac{x^2}{2!} + \ldots + \frac{x^n}{n!} + $$

$$\therefore e = 1 + 1 + \frac{1}{2!} + \ldots + \frac{1}{n!} + $$

The problems cannot be solved using a finite number of operations into problems solvable with a finite number of operations.

This is an infinite process that computers cannot achieve. In general, they can only calculate the sum of a finite number of terms and approximate its value.

$$e \approx 1 + 1 + \frac{1}{2!} + \ldots + \frac{1}{n!}$$

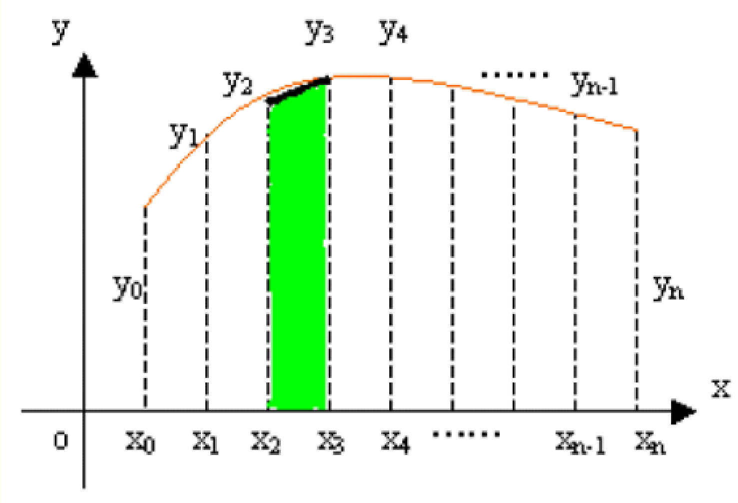
The error generated by the Taylor series $\left|R_n\right| < \frac{e}{(n+1)!} < \frac{3}{(n+1)!}$

$$\because e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \ldots + \frac{x^n}{n!} + \frac{e^{\theta x}}{(n+1)!} x^{n+1}, (0 < \theta < 1)$$

Discretization离散化--Transforming a problem involving continuous variables into a discrete variable problem.

Example 2： Calculate definite integral $$I = \int_a^b f(x)dx$$

The area of the curvilinear trapezoid shown in the figure cannot be computed continuously on a computer.



Generally, it can be calculated as follows:

1. Divide the interval [a, b] into n equal parts,

$$a = x_0 < x_1 < \ldots < x_n = b, y_i = f(x_i), i = 0,1,\ldots,n.$$

2. The sum of the areas of n trapezoids approximately replaces the area of a curvilinear trapezoid.

$$\int_a^b f(x)dx \approx \frac{b-a}{n}\left[\frac{1}{2}(y_0 + y_n) + y_1 + \ldots + y_{n-1}\right]$$

Recurrence递推化

Recurrence递推化--Complex calculations reduced to simple processes can be easily implemented using loop structures (iterative method).

Example 3：Calculate the polynomial $P_n(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0$

Constructing recursive algorithm:

$$\because P_n(x) = \left(a_n x + a_{n-1}\right)x^{n-1} + a_{n-2}x^{n-2} + \ldots + a_1 x + a_0$$

$$let, u_0 = a_n, u_1 = a_n x + a_{n-1} = u_0 x + a_{n-1}$$

$$\begin{aligned} \therefore P_n(x) &= u_1 x^{n-1} + a_{n-2}x^{n-2} + \ldots + a_1 x + a_0 \\ &= \left(u_1 x + a_{n-2}\right)x^{n-2} + \ldots + a_1 x + a_0 \\ &= u_{12} x^{n-2} + \ldots + a_1 x + a_0 \end{aligned}$$

$$\cdots\cdots \begin{cases} u_0 = a_n \\ u_k = u_{k-1} + a_{n-k} \end{cases} (1.3) \Rightarrow P_n(x) = u_n$$

It is known as the Qin Jiushao algorithm first proposed by Chinese mathematician Qin Jiushao during the Song Dynasty.

Remarks :

1. To master the principles and ideas of the algorithm

2. To master the processing skills, steps and calculation formulas of the algorithm

3. Pay attention to error analysis, understand the theory of convergence and stability analysis

4. Do certain theoretical analysis and calculation exercises

5. Practice on the computer

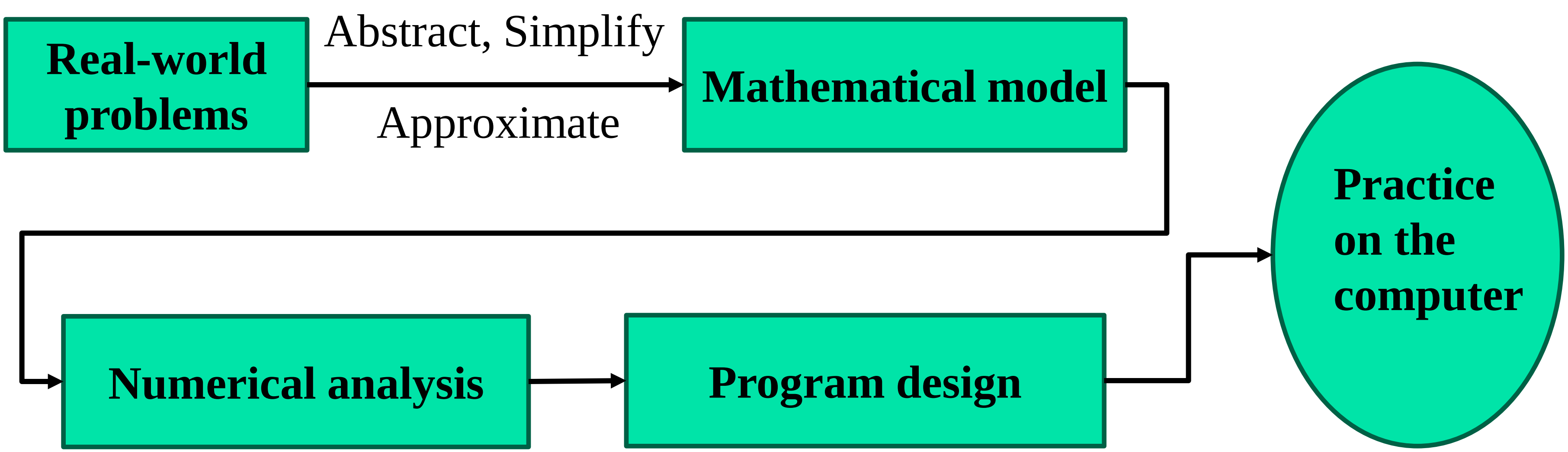
备注：

1. 掌握算法的原理和思想；
2. 掌握算法的处理技巧、步骤和计算公式；
3. 注重错误分析，了解收敛性和稳定性分析理论；
4. 进行一定的理论分析和计算练习；
5. 在计算机上进行实践。

§ 2 Introduction of error 误差介绍

1. When performing numerical computations for real-world problems using a computer, the obtained solutions are often approximations that contain errors.
2. Errors are inevitable, therefore it is important to allow for errors while also controlling them. Emphasis should be placed on error analysis, analyzing the sources of errors, the propagation of errors, and making estimations of the errors.

The process of computer solving scientific and engineering computational problems.



The main sources of error include the following:

1. modeling error, 2.truncation error,

3.observation error, 4.roundoff error.

1. 误差的来源与分类 /* Source & Classification */

➢ **To abstract (simplify) a mathematical model from practical problems, there exists an error between the model and the actual problem.—— 模型误差 /* Modeling Error */**

Example The population model proposed by the British economist Malthus.

$$\begin{cases} \dfrac{dp}{dt} = \alpha p & (1) \\ p(t_0) = p_0 \end{cases}$$,where $\alpha = 0.029$ is ecological coefficient.

$$\begin{cases} \dfrac{dp}{dt} = \alpha p - \beta p^2 & (2) \\ p(t_0) = p_0 \end{cases}$$,where $\beta > 0$ is social friction coefficient.

Measurement Error观测误差

- 通过测量得到模型中参数的值，观测产生误差

The values of the parameters in the model are obtained through measurement, and the observations find errors.

—— **观测误差** /* Measurement Error */

Truncation Error截断错误

➢ **Using numerical methods to obtain the approximate solution of the model, there exists an error between the approximate solution and the exact solution.—— 方法误差 (截断误差 /* Truncation Error */**

Example

$$e^x = 1 + x + \frac{x^2}{2!} + \ldots + \frac{x^n}{n!} + \ldots$$

$$e^x \approx S_n(x) = 1 + x + \frac{x^2}{2!} + \ldots + \frac{x^n}{n!}$$

The truncation error at this moment is

$$R_n(x) = e^x - S_n(x) = \frac{e^{\theta x}}{(n+1)!} x^{n+1}, (0 < \theta < 1)$$

Roundoff Error舍入误差

- **Machine word length is limited, resulting in errors in representing data in computers.—— 舍入误差** /* Roundoff Error */

Due to the limited word length of computers, calculations can only be performed on a finite number of digits, and any excess digits are rounded according to certain rules, resulting in 'rounding errors'.

Like, $\pi = 3.1415926\ldots, \sqrt{2} = 1.41421356\ldots$

When performing computations on a computer, only a finite number of digits can be taken into account.

The rounding error is obtained by keeping the digits after the decimal point to four decimal places

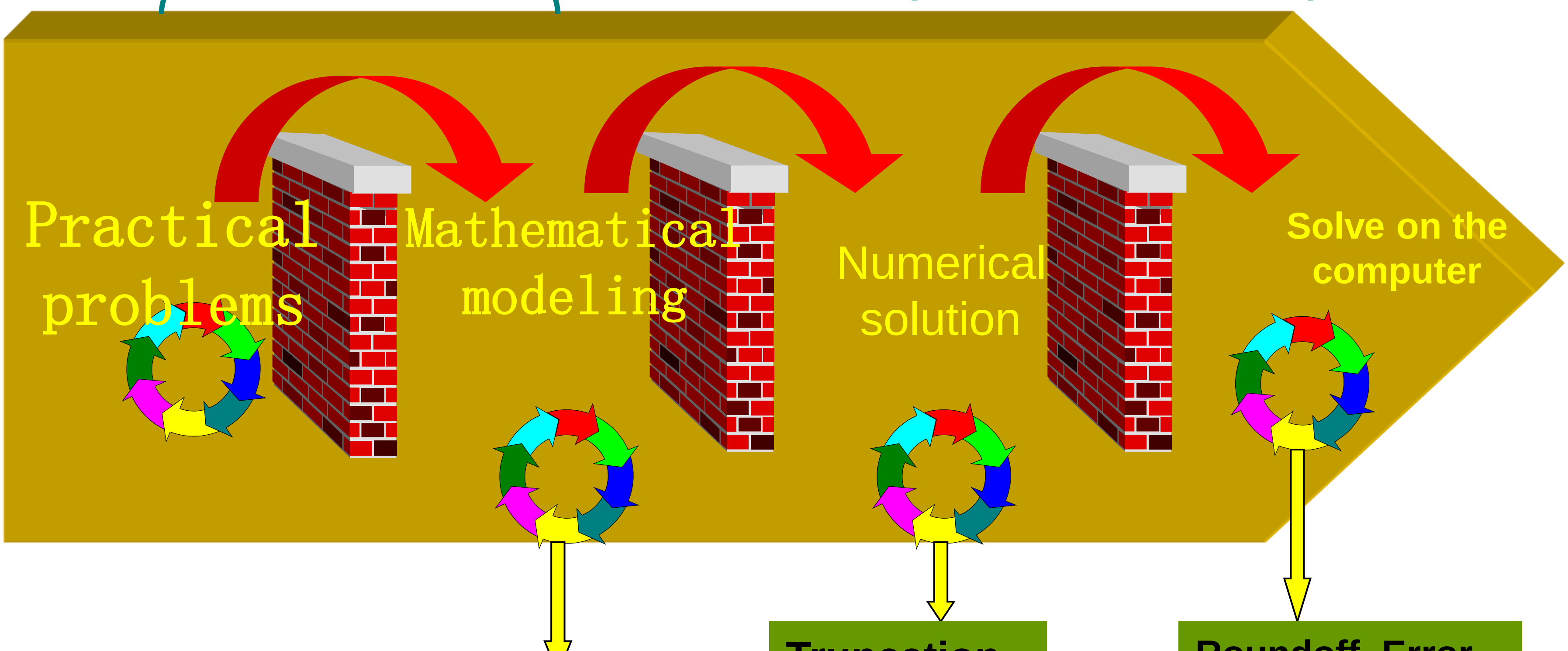
$$3.1416 - \pi = 0.0000074\ldots, 0.3333 - \frac{1}{3} = -0.000033\ldots$$

Summary: Model error and observational error are not the subject of discussion in numerical analysis. Computational methods primarily investigate the propagation of truncation error and rounding error during the calculation process, aiming to improve the accuracy of computations.

The general process of solving practical problems with a computer

Problems Solved by Applied Mathematics

Problems Solved by Numerical Computation



Modeling Error 、 Measurement Error

Truncation Error

Roundoff Error

The two kinds of errors are mainly studied here

Example 5. The approximate calculation $\int_0^1 e^{-x^2}\,dx = 0.747\ldots\ldots$

解法之一：将 e^{-x^2} 作Taylor展开后再积分

Integrate after Taylor expansion.

$$\int_0^1 e^{-x^2}\,dx = \int_0^1 \left(1 - x^2 + \frac{x^4}{2!} - \frac{x^6}{3!} + \frac{x^8}{4!} - \cdots\right) dx$$

$$= \underbrace{1 - \frac{1}{3} + \frac{1}{2!}\times\frac{1}{5} - \frac{1}{3!}\times\frac{1}{7}}_{S_4} + \underbrace{\frac{1}{4!}\times\frac{1}{9} - \cdots}_{R_4}$$

/* Remainder */

取 $\int_0^1 e^{-x^2}\,dx \approx S_4$，

则 $R_4 =$ … 为**截断误差** /* Truncation Error */

由留下部分
/* included terms */
引起

这里 $|R_4$ …

$S_4 = 1 - \frac{1}{3} + \frac{1}{10} - \frac{1}{42} \approx 1 - 0.333 + 0.1$ …

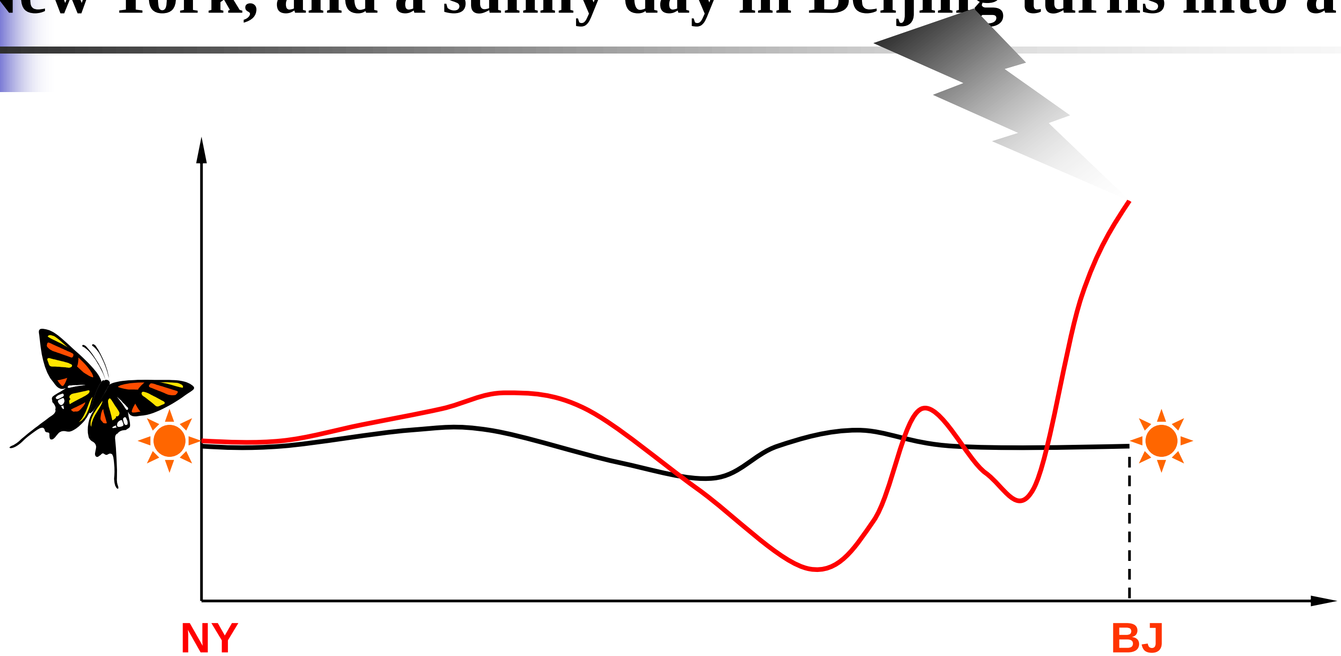
由截去部分
/* excluded terms */
引起

| **舍入误差** /* Roundoff Error */ | $< 0.0005 \times 2 = 0.001$

计算 $\int_0^1 e^{-x^2}\,dx$ 的总体误差 $< 0.005 + 0.001 = 0.006$

2. 传播与积累 /* Spread & Accumulation */

Example： Butterfly Effect — A flap of a butterfly's wings in New York, and a sunny day in Beijing turns into a typhoon?!



以上是一个病态问题 /* ill-posed problem*/

As for the problem of being pathological in itself, let's leave it to mathematicians to worry about!

例6：计算 $I_n = \frac{1}{e}\int_0^1 x^n e^x dx\ ,\quad n = 0, 1, 2, \ldots\ldots$

公式一：$I_n = 1 - n\, I_{n-1}$

注意此公式精确成立

Please note that this formula holds true.

$$I_0 = \frac{1}{e}\int_0^1 e^x dx = 1 - \frac{1}{e} \approx 0.63212056 \overset{\text{记为}}{=\!=} I_0^*$$

则初始误差 $|E_0| = |I_0 - I_0^*| < 0.5 \times 10^{-8}$

$$\frac{1}{e}\int_0^1 x^n \cdot e^0\, dx < I_n < \frac{1}{e}\int_0^1 x^n \cdot e^1\, dx \qquad \therefore \frac{1}{e(n+1)} < I_n < \frac{1}{n+1}$$

$$I_1^* = 1 - 1 \cdot I_0^* = 0.36787944$$

$$\cdots\cdots\cdots\cdots$$

$$I_{10}^* = 1 - 10 \cdot I_9^* = 0.08812800$$

$$I_{11}^* = 1 - 11 \cdot I_{10}^* = 0.03059200$$

$$I_{12}^* = 1 - 12 \cdot I_{11}^* = 0.63289600 \ ?$$

$$I_{13}^* = 1 - 13 \cdot I_{12}^* = -7.2276480 \ ??$$

$$I_{14}^* = 1 - 14 \cdot I_{13}^* = 94.959424 \ ?\,!$$

$$I_{15}^* = 1 - 15 \cdot I_{14}^* = -1423.3914 \ !\,!$$

考察第n步的误差$|E_n|$

$$|E_n| = |I_n - I_n^*| = |(1 - nI_{n-1}) - (1 - nI_{n-1}^*)| = n|E_{n-1}| = \cdots = n!|E_0|$$

可见初始的小扰动 $|E_0| < 0.5\times 10^{-8}$ 迅速积累，误差呈递增走势。

造成这种情况的是**不稳定的算法** /* unstable algorithm */

我们有责任改变。

公式二： $I_n = 1 - n\,I_{n-1} \quad \Rightarrow \quad I_{n-1} = \frac{1}{n}(1 - I_n)$

方法：先估计一个I_N，再反推要求的I_n ($n << N$)。

注意此公式与公式一在理论上**等价**。

$$\frac{1}{e(N+1)} < I_N < \frac{1}{N+1}$$

可取 $I_N^* = \frac{1}{2}\left[\frac{1}{e(N+1)} + \frac{1}{N+1}\right] \approx I_N$

当$N \to +\infty$时，$|E_N| = |I_N - I_N^*| \to 0$

取 $$I_{15}^* = \frac{1}{2}\left[\frac{1}{e\cdot 16}+\frac{1}{16}\right] \approx 0.042746233$$

$$\Rightarrow I_{14}^* = \frac{1}{15}(1-I_{15}^*) \approx 0.063816918$$

$$I_{13}^* = \frac{1}{14}(1-I_{14}^*) \approx 0.066870220$$

$$I_{12}^* = \frac{1}{13}(1-I_{13}^*) \approx 0.071779214$$

$$I_{11}^* = \frac{1}{12}(1-I_{12}^*) \approx 0.077351732$$

$$I_{10}^* = \frac{1}{11}(1-I_{11}^*) \approx 0.083877115$$

$$\vdots$$

$$I_{1}^* = \frac{1}{2}(1-I_{2}^*) \approx 0.36787944$$

$$I_{0}^* = \frac{1}{1}(1-I_{1}^*) \approx 0.63212056$$

考察反推一步的误差：/* The error of the reverse step*/

$$|E_{N-1}| = \left| \frac{1}{N}(1 - I_N) - \frac{1}{N}(1 - I_N^*) \right| = \frac{1}{N} |E_N|$$

以此类推/* and so on*/，对 $n < N$ 有：

$$|E_n| = \frac{1}{N(N-1)\dots(n+1)} |E_N|$$

误差逐步递减, 这样的算法称为稳定的算法 /* stable algorithm */
/* Gradually decreasing error */

In our future discussions, errors will not be avoided, and the stability of algorithms will be a very important topic.

§ 3 误差/* Error */

误差、误差限 /* Error, accuracy*/

➢ 绝对误差/* absolute error*/

Definition 1.1 Let x is an exact value, and x^* is an approximation of x, then there is

$\varepsilon(x) = |x - x^*|$ is the absolute error of x^* ;

$e = x^* - x$ is the error of x^* ;

(1)Errors have dimensions and can be positive or negative.

(2)The magnitude of $\varepsilon(x)$ indicates the precision of x^*, but calculating its exact value is challenging. However, an upper bound ε can be estimated, $\varepsilon(x) = |x^* - x| \le \varepsilon$.

And **the absolute error accuracy**, for ε is represented as x^*.

$$x^* - \varepsilon \le x \le x^* + \varepsilon \text{或} x \in [x^* - \varepsilon, x^* + \varepsilon] \text{或} x = x^* \pm \varepsilon$$

(3) Let $\varepsilon = 0.5 \times 10^p$, and the minimum integer satisfies the given condition p.

Example 7. The length of the table, measured using a ruler with millimeter scale, is recorded as x*=1235mm.

The approximate value of the actual length x of the table, based on the precision of the meter ruler, indicates that the error of this approximation will not exceed 0.5mm (i.e., the absolute error limit is 1/2mm).

$$|x^* - x| = |1235 - x| \leq \frac{1}{2} mm$$

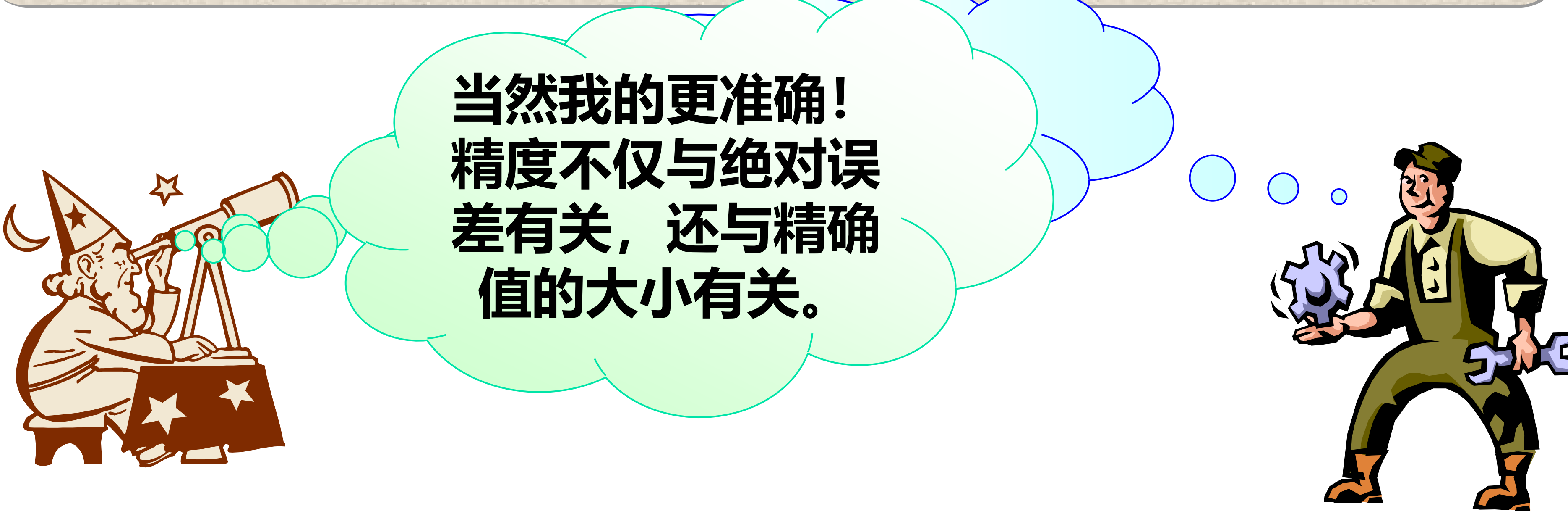
$$1234.5 \leq x \leq 1235.5$$

so $$x \in [1234.5, 1235.5]$$

or $$x = 1235 \pm 0.5 mm$$

Example 5: $\int_0^1 e^{-x^2}dx = 0.743 \pm 0.006$

Remarks： *In theory, e is uniquely determined and can take on either positive or negative values. The statement "e > 0" is not unique; however, smaller values of e are more valuable as references.*



相对误差 /* Relative error*/

- **相对误差 /* Relative error*/**

$e_r = \frac{e}{x} = \frac{x^* - x}{x}$ is the relative error of the approximate value x^*

Remarks:

(1) When $|e_r|$ is small, define $e_r = \frac{e}{x} = \frac{x^* - x}{x^*}$

(2) The upper bound of the absolute value of relative error is called **the relative accuracy**, denoted as ε_r

$$|e_r| = \left|\frac{x^* - x}{x^*}\right| \le \frac{\varepsilon}{|x^*|} = \varepsilon_r$$

where ε is the error limit of x^*

光速/* speed of light */ $c = (2.997925 \pm 0.000001) \times 10^{10}$ cm/s

$c^* = 2.997925 \times 10^{10}$ cm/s

绝对误差限/* the absolute accuracy*/

$$\varepsilon = 0.000001 \times 10^{10} \text{ cm/s}$$

相对误差限/* the relative accuracy*/

$$\varepsilon_r = \frac{\varepsilon}{|c^*|} = \frac{0.000001}{2.997925} \approx 0.0000003.$$

Remarks: The difference between absolute error and relative error.

1. Absolute error is dimensioned, while relative error is dimensionless.
2. For two different approximate values of the same quantity, their precision can be determined by their absolute errors. In contrast, when comparing the accuracy level of two different quantities' approximate values, only their relative errors are considered.

§5 几点注意事项 /* Remarks */

1. 避免相近二数相减

/* Avoid subtracting two similar numbers */

例：$a_1 = 0.12345$，$a_2 = 0.12346$，各有5位有效数字。

而 $a_2 - a_1 = 0.00001$，只剩下1位有效数字。

几种经验性避免方法/* Several empirical avoidance methods */ :

$$\sqrt{x+\varepsilon}-\sqrt{x}=\frac{\varepsilon}{\sqrt{x+\varepsilon}+\sqrt{x}};\qquad \ln(x+\varepsilon)-\ln x=\ln\left(1+\frac{\varepsilon}{x}\right);$$

当 $|x| << 1$ 时：

$$1-\cos x = 2\sin^2\frac{x}{2};$$

$$e^x-1=x\left(1+\frac{1}{2}x+\frac{1}{6}x^2+\ldots\right)$$

2.To avoid the absolute value of the divisor being significantly smaller than the absolute value of the dividend.

Example: $$\frac{2.718}{0.001} = 2718.2$$

When the denominator Y undergoes a small change 0.0001, then,

$$\frac{2.7182}{0.0011} = 2471.1$$

The computer results are sensitive to perturbations in Y, which is usually an approximate value, making the calculation results unreliable. Additionally, dividing by very small numbers can cause a computer to overflow and shut down.

3. Avoiding decimals in large numbers

例：Calculate the root with single precision $x^2-(10^9+1)x+10^9=0$ 。

The exact solution is $x_1=10^9,\ x_2=1$

✍ **Algorithm1：** **Using the Rooting Formula** $x=\dfrac{-b\pm\sqrt{b^2-4ac}}{2a}$

在计算机内，10^9存为0.1×10^{10}，1存为0.1×10^{1}。When performing addition, align the exponents of the two addends first according to the larger exponent, and then add the fractional parts.即1 的指数部分须变为10^{10}，则：$1 = 0.0000000001\times10^{10}$，取单精度时就成为：

$$10^9+1=0.10000000\times10^{10}+0.00000000\times10^{10}=0.10000000\times10^{10}$$

大数吃小数

$$\Rightarrow x_1=\frac{-b+\sqrt{b^2-4ac}}{2a}=10^9,\quad x_2=\frac{-b-\sqrt{b^2-4ac}}{2a}=0$$

✍算法2：先解出 $x_1 = \dfrac{-b - sign(b)\cdot\sqrt{b^2-4ac}}{2a} = 10^9$

再利用 $x_1 \cdot x_2 = \dfrac{c}{a} \Rightarrow x_2 = \dfrac{c}{a \cdot x_1} = \dfrac{10^9}{10^9} = 1$

Remark: To minimize the error when summing, **add numbers in ascending order.**

Example: Calculate in ascending and descending order.

$$1 + 2 + 3 + \ldots + 40 + 10^9$$

4. Simplify first and then calculate to reduce steps and avoid accumulation of errors.

In general, the speed at which computers process the following operations is: $(+\,,-) > (\times\,,\div) > (\exp)$

5. Choose a stable algorithm.

Criteria for evaluating algorithms: complexity, accuracy, stability.

Criteria for evaluating algorithms:
评估算法的标准：

Complexity复杂度

accuracy准确性

stability稳定性

Assignment 作业

If the first 6 terms of Taylor expansion are used to calculate
使用泰勒展开式的前6项来计算

$$\int_0^1 e^{-x^2} dx$$

, find its error, including truncation error and roundoff error.
并且计算其误差，包括截断误差和舍入误差。

Criteria for evaluating algorithms:

Complexity

accuracy

stability

Assignment

If the first 6 terms of Taylor expansion are used to calculate

$$\int_0^1 e^{-x^2}\,dx=$$

, find its error, including truncation error and roundoff error.