《数值计算方法》

绪　论

讲授人：杨 程
时　间：2021-2022学年
秋季学期
Email: yang_cheng@nun.edu.cn

课程要求：

1、不准随意缺勤、迟到、早退。有事请假需有学院的假条。

2、上课准守纪律，可以提问、讨论，不准长时间看手机、睡觉。

3、独立完成、及时上交作业。

4、考核：

出勤10%+作业30%+闭卷考试60%，70分及格。

其中，超过（含）查到次数的三分之一不在（包含请假），出勤0分；作业超过（含）三分之一未交，作业0分。

目录

绪　论

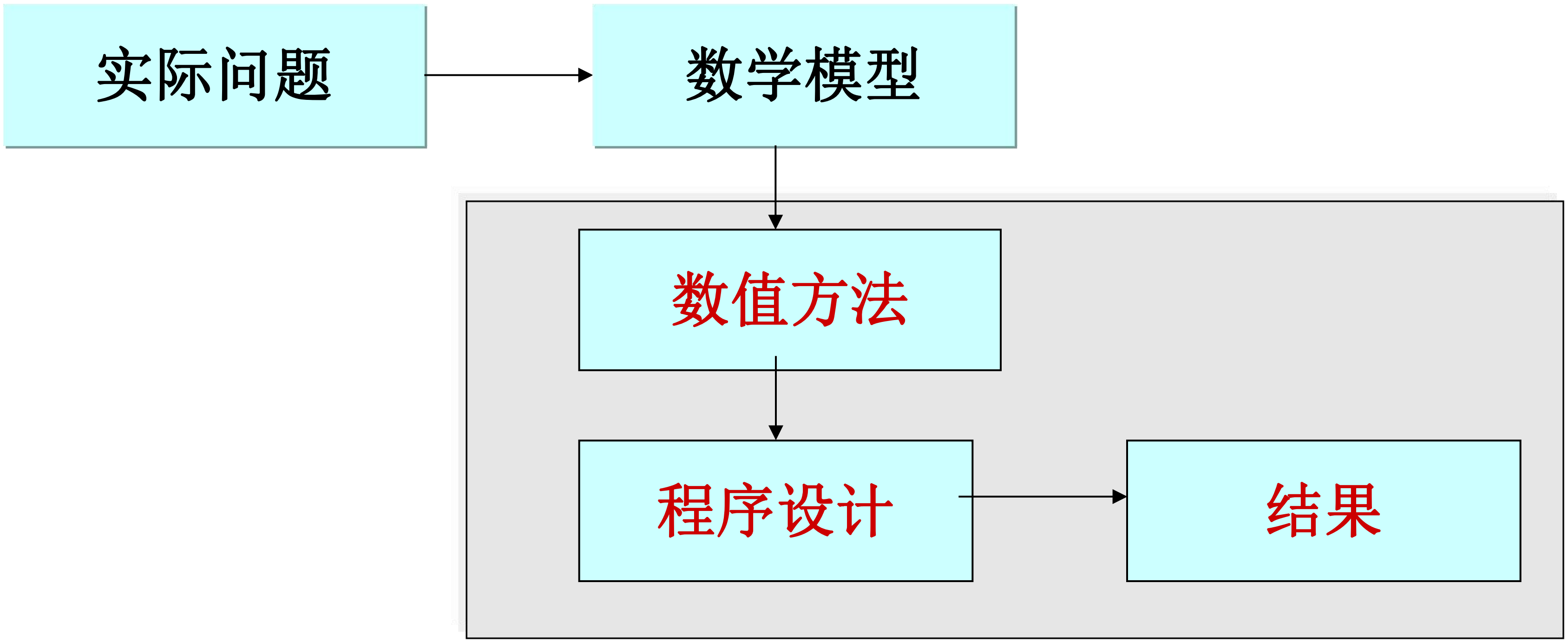
PART 1

数值计算方法的概况

1、研究对象

数值计算方法是研究用计算机解决各种数学问题的数值计算方法及其理论，又称为计算方法、数值分析。

使用计算机解决科学计算问题时大致经历如下几个过程：



2、研究内容

研究解决各种数学问题的数值计算方法及其相应理论，包括以下内容：

- 函数的插值及拟合
- 数值微分与数值积分
- 非线性方程的数值解法
- 线性方程组的直接解法和数值解法
- 微分方程的数值解法
- 矩阵的特征值问题的解法

3、课程特点

- ✓面向计算机

  算法中只涉及计算机可直接处理的一些运算。

- ✓有可靠的理论分析

  收敛性、数值稳定性、误差分析

- ✓有良好的计算复杂性

  时间复杂性和空间复杂性

- ✓有数值实验

  编程计算，检验算法的有效性

4、研究意义

- ✓**科学计算是现代科学发展的重要手段之一，发挥着越来越重要的作用**

随着科学技术的突飞猛进，无论是工农业生产还是国防尖端技术，例如机电产品的设计、建筑工程项目的设计、气象预报和新型尖端武器的研制、火箭的发射等，都有大量复杂的数值计算。

- ✓**多数数学问题无法求出解析解，只能借助计算机求出其近似解**

如：求定积分的值：

$$\int_0^1 \frac{\sin x}{x}\,dx \qquad \int_0^1 e^{-x^2}\,dx \qquad \int_0^1 \sqrt{1+x^3}\,dx$$

解非线性方程：

$$xe^x-2=0$$

✓数值算法的好坏决定了整个计算过程的效率甚至成败

如：对20阶的线性方程组，用Cramer（克莱默）法则求解，乘除法的运算次数达到9.7×10^{20}，若用每秒钟一亿次的计算机计算也要30万年；而用Gauss消去法求解，则只需乘除法次数为3060次，不需1秒钟就可计算出来。

如何学好数值计算方法

- **注意掌握设计算法的思想和原理、处理技巧，重视误差分析、收敛性、稳定性等理论基础**
- **注重实际问题，练习、作业**
- **注重实践：积极动手上机实践 （Matlab）**

PART 2

算　法

对给定的数学问题，数值计算方法的作用就是对问题在理论分析的基础上，给出有效的、计算机上易于实现的方法–算法。

定义：算法是对一些数据按某种规定的顺序进行数学运算及逻辑运算的一个运行序列。

算法优劣的评价标准：

- 精度
- 工作量（时间花费）
- 占用的存储量（空间花费）
- 逻辑结构是否简单
- 能否充分利用问题本身的特点

说明：不同问题对算法的要求侧重可能不一样，有的问题可能对计算时间要求不高，但要求有很高的精度，数值计算方法研究人员就是要**根据具体问题给出适合该问题的有效的算法**。

例1、n阶线性方程组求解

- **Cramer法则：求解n+1个n阶行列式的值。**

其计算量是$n!(n-1)(n+1)$乘法，20阶需要10^{21}次乘法如用每秒10^{10}亿次乘法的计算机，计算的需要 3080 年。

不适用

- **Gauss消元法：$n^3/3+n^2$,需 3000 次乘法运算**

❖ 算法的特点

1.构造性的数值方法，通过数值演算过程来完成解的构造

- 准确定义解题方案的每个细节
- 完整描述解题过程

算法：指解题方案的准确而完整的描述
(不仅仅指计算公式)。 核心

2. 递推化

- 递推化是将一个复杂的计算过程归结为简单过程的多次重复。 这种重复在算法上表现为循环，

3. 近似替代

在误差允许的范围内，无限次的计算能用有限次计算替代

例2、求解二次方程 $x^2+2bx+c=0$ 的实根。

解法：公式法

$$x_{1,2}=-b\pm\sqrt{b^2-c}$$

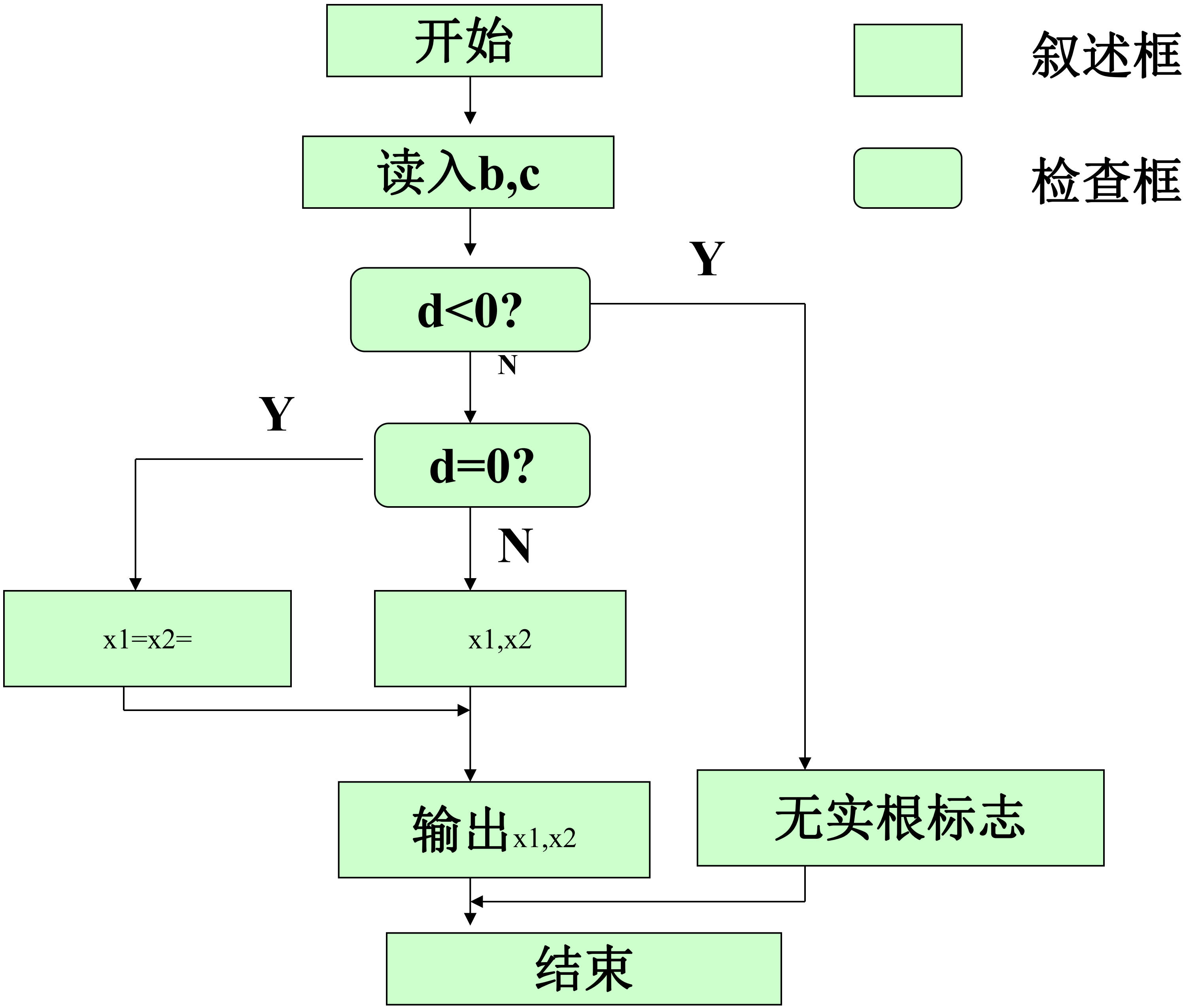
判别式的情形：

1) $d=b^2-c<0$, 无实根

2) $d=0$, 有重根 $x_1=x_2=-b$

3) $d>0$，有两个互异实根, x_1, x_2

- **算法的描述**
  - **框图**


开始
叙述框
读入b,c
检查框
Y
d<0?
N
Y
d=0?
N
x1=x2=
x1,x2
输出x1,x2
无实根标志
结束

例3、多项式求值

对于给定的x求多项式的值

$$p_n(x)=a_0+a_1x+a_2x^2+\cdots+a_nx^n$$

方法一：直接利用公式。 $\frac{n(n+1)}{2}$ 次乘法

方法二： 令$t_0=1, p_0=a_0$

$for \quad k=0:n-1$

$$t_{k+1}=t_kx_0$$

$$p_{k+1}=p_k+a_{k+1}t_{k+1}$$

end

t_k,p_k,在计算完t_{k+1},p_{k+1}后不起作用，不需要继续存储

$p_n=p_n(x_0)$ 2n次乘法

方法三：秦九昭算法（多项式改写）

$$p_n\left(x_0\right)=\left[(\cdots\left(\left(a_n x_0+a_{n-1}\right)x_0+a_{n-2}\right)x_0+\cdots+a_1)x_0+a_0\right]$$

设$t_n = a_n$

$t_{n-1} = t_n x_0 + a_{n-1}$

n次乘法

$\cdots$

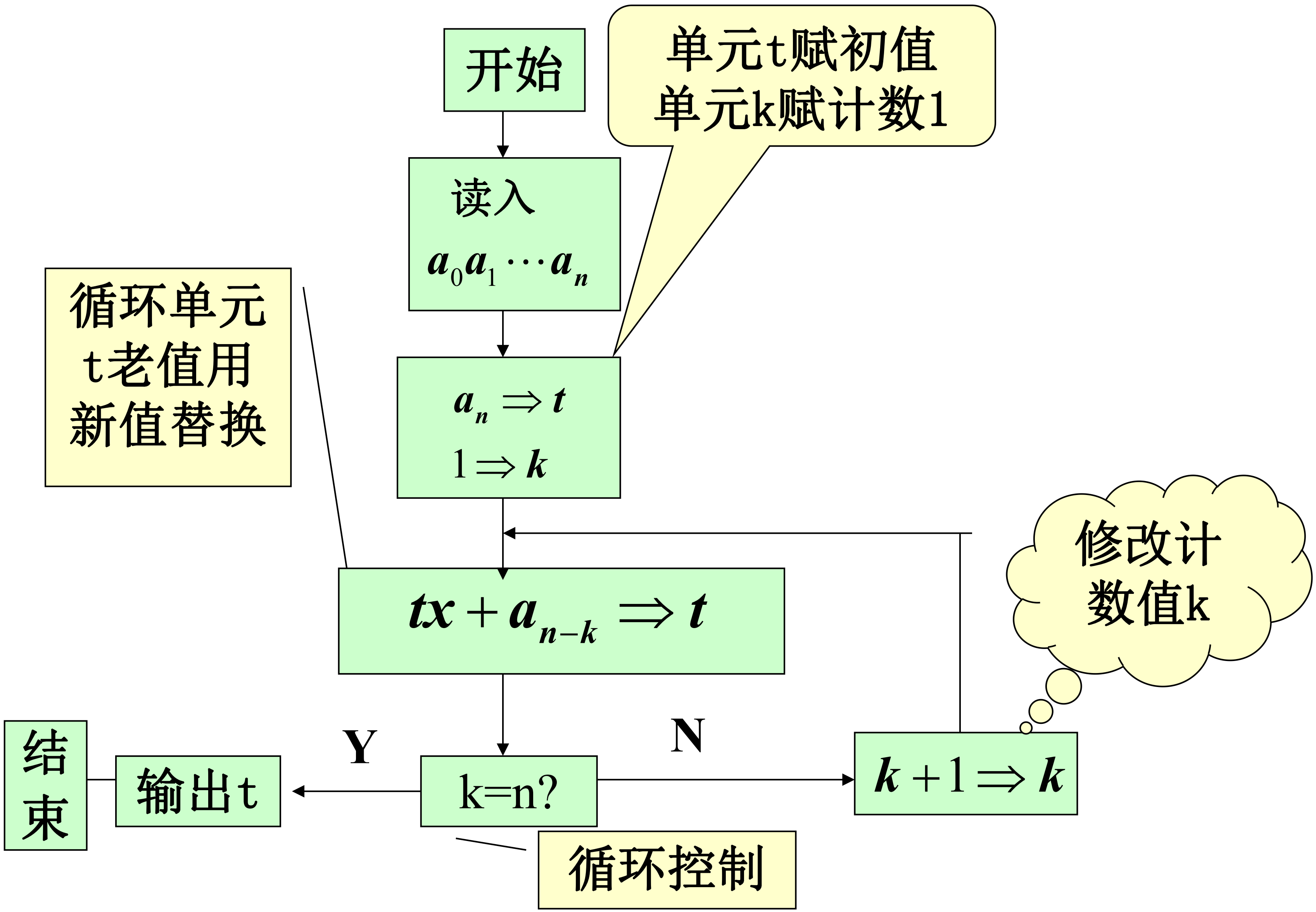
$t_k = t_{k+1} x_0 + a_k, k = n-2, \cdots, 1, 0$

说明：三个算法：理论上等价但是算法上不等价

$t_0 = p_n(x_0)$

n次多项式求值 $\longrightarrow$ n个1次式$t_k = t_{k-1} x + a_{n-k}$

- 程序实现



开始
读入
$a_0a_1\cdots a_n$
单元t赋初值
单元k赋计数1
$a_n \Rightarrow t$
$1 \Rightarrow k$
循环单元
t老值用
新值替换
$tx + a_{n-k} \Rightarrow t$
修改计
数值k
Y
N
k=n?
$k+1 \Rightarrow k$
结
束
输出t
循环控制

算法举例—不同算法对计算结果的影响

例、计算

$$x=\left(\frac{\sqrt{2}-1}{\sqrt{2}+1}\right)^3$$

可用四种算式算出：

$$x=\left(\sqrt{2}-1\right)^6,\quad x=99-70\sqrt{2}$$

$$x=\left(\frac{1}{\sqrt{2}+1}\right)^6,\quad x=\frac{1}{99+70\sqrt{2}}$$

如果分别用近似值 $\sqrt{2}\approx 7/5=1.4$ 和 $\sqrt{2}\approx 17/12=1.4166\cdots$

按上列四种算法计算 x 值，其结果如下**表**所示。

表1（x=0.0050506）

算式	计算结果	
	$\sqrt{2}\approx 7/5$	$\sqrt{2}\approx 17/12$
$\left(\sqrt{2}-1\right)^6$	$\left(\frac{2}{5}\right)^6=0.004096$	$\left(\frac{5}{12}\right)^6=0.005233$
$99-70\sqrt{2}$	1	$-\frac{1}{6}=-0.166667$
$\left(\frac{1}{\sqrt{2}+1}\right)^6$	$\left(\frac{5}{12}\right)^6=0.005233$	$\left(\frac{12}{29}\right)^6=0.005020$
$\frac{1}{99+70\sqrt{2}}$	$\frac{1}{197}=0.005076$	$\frac{12}{2378}=0.005046$

练习

1.证明：方程$f(x)=e^x+10x-2=0$在$[0,1]$内有唯一实根，用二分法求此实根，要求误差不超过$\frac{1}{2}\times 10^{-2}$

2.用秦九昭算法求多项式$p(x)=x^5-3x^4+4x^2-x+1$在$x=3$时的值

PART 3

误　差

1. 误差的来源与分类

（1）模型误差

（2）观测误差

（3）截断误差—方法误差

在不少数值运算中常遇到超越计算，如微分、积分和无穷级数求和等，它们须用**极限或无穷过程**来求得。然而**计算机却只能完成有限次**算术运算和逻辑运算，因此需将解题过程化为一系列有限的算术运算和逻辑运算。**这样就要对某种无穷过程进行“截断”**，即仅保留无穷过程的前段有限序列而舍弃它的后段。

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

（4）舍入误差

在**数值计算过程中**，由于受计算机机器字长的限制，它所能表示的数据只能是有限位数，这时就需把数据舍入成一定位数的近似的有理数来代替。如

$$\pi = 3.14159265\cdots,\quad \pi^* = 3.14$$

由此引起的误差称为**“舍入误差”**。

数学模型一旦建立，**进入具体计算时所要考虑和分析的就是截断误差和舍入误差了**。在计算机上经过千百次运算后所积累起来的总误差不容忽视，有时可能会大得惊人，甚至到达“淹没”所欲求解的真值的地步，而使计算结果失去根本的意义。**因此，在讨论算法时，有必要对其截断误差的估算和舍入误差的控制作适当的分析。**

☆☆☆数值计算方法中主要考虑分析的是：截断误差和舍入误差

2. 绝对误差和相对误差

（1） 绝对误差和绝对误差限

定义 设某一个准确值（称为真值）为 x，其近似值为 x^*，则 x 与 x^* 的差：

$$|\varepsilon(x)| = x - x^*$$

称为近似值 x 的“**绝对误差**”，简称“**误差**”。

由于真值往往是未知或无法知道的，因此，$\varepsilon(x)$ 的准确值（真值）也就无法求出。但一般可估计此绝对误差的上限，也即可以求出一个正值 η，使

$$|\varepsilon(x)| = |x - x^*| \le \eta$$

$$|\varepsilon(x)| = |x - x^*| \leq \eta$$

此 η 称为近似值 x^* 的“**绝对误差限**”，简称“**误差限**”，或称“**精度**”。有时也用

$$x = x^* \pm \eta$$

这时等式右端的两个数值 $x^* + \eta$ 和 $x^* - \eta$ 代表了x所在范围的上、下限。η **越小，表示该近似值 x^* 的精度越高。**

(2) 相对误差和相对误差限

定义绝对误差与真值之比，即

$$\varepsilon_r(x)=\frac{\varepsilon(x)}{x}=\frac{x-x^*}{x}$$

称为近似值 x^* 的“相对误差”。在估计近似值运算结果的误差时，它比绝对误差更能反映出误差的特性。因此在误差分析中，相对误差比绝对误差更为重要。

和绝对误差一样，可以估计它的大小范围，即可以找到一个正数 δ，使

$$\left|\varepsilon_r(x)\right|\le\delta$$

δ 称为近似值 x^* 的“相对误差限”

在实际计算中，由于真值 x 总是无法知道的，因此往往取

$$\varepsilon_r^*(x) = \frac{\varepsilon(x)}{x^*}$$

作为相对误差的另一定义。

下面比较 $\varepsilon_r^*(x)$ 与 $\varepsilon_r(x)$ 之间的相差究竟有多大：

$$\varepsilon_r(x) - \varepsilon_r^*(x) = \varepsilon(x)\left(\frac{1}{x} - \frac{1}{x^*}\right) = -\frac{1}{x \bullet x^*}\left[\varepsilon(x)\right]^2$$

$$= -\frac{1}{xx^*}(x\varepsilon_r(x))^2 = -\frac{x}{x^*}(\varepsilon_r(x))^2 = -\left\{\frac{1}{1-\varepsilon_r(x)}\right\}\left[\varepsilon_r(x)\right]^2$$

3 有效数字

（1） 有效数字

在表示一个近似值的准确程度时，常用到“有效数字”的概念。例如 $\pi = 3.14159265\cdots$，若按四舍五入取四位小数，则得 π 的近似值为 3.1416；若取五位小数则得其近似值位 3.14159。这种近似值取法的特点是**误差限为其末位的半个单位**，即

$$\left|\pi - 3.1416\right| \le \frac{1}{2} \times 10^{-4}$$

$$\left|\pi - 3.14159\right| \le \frac{1}{2} \times 10^{-5}$$

$$|x-\tilde{x}| \le 0.\underbrace{0 0 \cdots 0 5}_{\text{n个}} = 0.5 \times 10^{-n} = \frac{1}{2} \times 10^{-n}$$

- 称近似值 $\tilde{x}$ 准确到n位小数。
- 将从该位起直到最左端的非零数字之间的一切数字称为近似值 $\tilde{x}$ 的准确数字（有效数字）。

数1.23和1.230是有区别的，前者准确到两位小数，后者准确到三位小数，故不能随意给数的末尾添零。

对于一般形式，设有一个数 x ，其近似值 x^* 的规格化形式：

$$x^* = \pm 0.\alpha_1\alpha_2\cdots\alpha_n \times 10^m$$

式中：$\alpha_1,\alpha_2\cdots\alpha_n$ 都是 $0,1,2,\cdots,9$ 中的一个数字，$\alpha_1 \neq 0;$

n是**正**整数，m是整数。

若 x^* 的误差限为：

$$|\varepsilon(x)| = |x - x^*| \leq \frac{1}{2} \times 10^{m-n}$$

则称 x^* 为具有n位有效数字的有效数，或称它精确到 10^{m-n} 。其中每一位数字 $\alpha_1,\alpha_2,\cdots,\alpha_n$ 都是 x^* 的有效数字。

$$\left|\pi - 3.1416\right| \le \frac{1}{2} \times 10^{-4}$$

若上式中的 x^* 是 x 经四舍五入得到的近似数，则 x^*具有 n 位有效数字。例如，3.1416是 π 的具有五位有效数字的近似值，精确到0.0001。首先按照准确数字的定义分析：

$$|\pi - 3.1416| = 0.0000074\cdots \le 0.00005 = 0.5 \times 10^{-4}$$
$$= \frac{1}{2} \times 10^{-4}$$

x^* 准确到四位小数，具有五位有效数字。

由一般形式的结论分析，此时 $x^* = 3.1416 = 0.31416 \times 10$

则，m=1。 $|\varepsilon(x)| = |x - x^*| \le 0.5 \times 10^{-4}$ **可推出n=5**

因此，x^* 的有效数字是五位。

例、 x = 0.1524，x* = 0.154 ， 判断x* 的有效数字位数。

$$|\varepsilon(x)| = |x - x^*| = |0.1524 - 0.154| = 0.0016$$

$$\leq 0.5 \times 10^{-2}$$

x^* 准确到2位小数，具有2位有效数字。

对于 $\boldsymbol{x} = \boldsymbol{\pi} = 3.1415926\cdots,$

思考 $\boldsymbol{x}_1^* = 3.14,\ \boldsymbol{x}_2^* = 3.1416,\ \boldsymbol{x}_3^* = 3.1415$

$$|\pi - 3.14| = 0.00159265\cdots \leq 0.005 = 0.5 \times 10^{-2} = \frac{1}{2} \times 10^{-2}$$

x_1^*准确到两位小数，具有三位有效数字。

$$|\pi - 3.1415| = 0.0000926\cdots \leq 0.0005 = 0.5 \times 10^{-3} = \frac{1}{2} \times 10^{-3}$$

x_3^*准确到三位小数，具有四位有效数字。

注意

(1)有效数字的位数与小数点的位置无关;

(2)有效数位越多,相对误差越小.

4　数据误差的传播与估计

（1）　误差估计的一般公式

在实际的数值计算中，参与运算的数据往往**都是近似值**，**带有误差**，这些数据误差在多次运算过程中会进行传播，**导致计算结果产生误差**，而**确定计算结果所能达到的精度显然是十分重要的，但往往很困难**。

对计算误差作出一定的定量估计还是可以做到的。下面利用函数泰勒（Taylor）展开式推出误差估计的一般公式。

考虑二元函数$y=f(x_1,x_2)$，设x^*_1和 x^*_2分别是x_1和 x_2的近似值，y^* 是函数值y的近似值，且

$$y^* = f(x^*_1, x^*_2)$$

函数$f(x_1,x_2)$在点(x^*_1,x^*_2)处的泰勒展开式为：

$$
\begin{aligned}
f(x_1,x_2) = f(x_1^*,x_2^*) &+ [(\frac{\partial f}{\partial x_1})^*(x_1 - x_1^*) + (\frac{\partial f}{\partial x_2})^*(x_2 - x_2^*)\ ] \\
&+ \frac{1}{2!}[(\frac{\partial^2 f}{\partial x_1^2})^* \cdot (x_1 - x_1^*)^2 + 2(\frac{\partial f}{\partial x_1 \partial x_2})^* \cdot (x_1 - x_1^*)(x_2 - x_2^*) \\
&+ (\frac{\partial^2 f}{\partial x_2^2})^* \cdot (x_1 - x_2^*)^2] + \cdots
\end{aligned}
$$

式中，$(x_1 - x_1^*) = \varepsilon(x_1)$和$(x_2 - x_2^*) = \varepsilon(x_2)$一般都是小量值，如**忽略高阶小量**，则上式可简化为：

$$
f(x_1,x_2) \approx f(x_1^*,x_2^*) + (\frac{\partial f}{\partial x_1})^* \cdot \varepsilon(x_1) + (\frac{\partial f}{\partial x_2})^* \cdot \varepsilon(x_2)
$$

$$f(x_1,x_2) \approx f(x_1^*,x_2^*) + \left(\frac{\partial f}{\partial x_1}\right)^* \cdot \varepsilon(x_1) + \left(\frac{\partial f}{\partial x_2}\right)^* \cdot \varepsilon(x_2)$$

因此，y^* 的绝对误差和相对误差分别为：

$$\begin{aligned}\varepsilon(y) &= y - y^* = f(x_1,x_2) - f(x_1^*,x_2^*) \\ &\approx \left(\frac{\partial f}{\partial x_1}\right)^* \cdot \varepsilon(x_1) + \left(\frac{\partial f}{\partial x_2}\right)^* \cdot \varepsilon(x_2)\end{aligned} \qquad (1)$$

$$\begin{aligned}\varepsilon_r^*(y) &= \frac{\varepsilon(y)}{y^*} \approx \left(\frac{\partial f}{\partial x_1}\right)^* \frac{\varepsilon(x_1)}{y^*} + \left(\frac{\partial f}{\partial x_2}\right)^* \frac{\varepsilon(x_2)}{y^*} \\ &= \frac{x_1^*}{y^*}\left(\frac{\partial f}{\partial x_1}\right)^* \cdot \varepsilon_r^*(x_1) + \frac{x_2^*}{y^*}\left(\frac{\partial f}{\partial x_2}\right)^* \cdot \varepsilon_r^*(x_2)\end{aligned} \qquad (2)$$

（2）误差在算术运算中的传播

可以利用公式（1）和（2）对算术运算中数据误差传播规律作一具体分析。

第一步：加，减运算

由（1）和（2）有

$$\varepsilon(x_1 + x_2) \approx \varepsilon(x_1) + \varepsilon(x_2) \tag{3}$$

$$\varepsilon_r^*(x_1 + x_2) \approx \frac{x_1^*}{x_1^* + x_2^*}\varepsilon_r^*(x_1) + \frac{x_2^*}{x_1^* + x_2^*}\varepsilon_r^*(x_2) \tag{4}$$

近似值之和的绝对误差等于各近似值绝对误差的代数和。

两数x_1和x_2相减，由式**(4)**有

$$\varepsilon_r^*(x_1-x_2)\approx\frac{x_1^*}{x_1^*-x_2^*}\varepsilon_r^*(x_1)-\frac{x_2^*}{x_1^*-x_2^*}\varepsilon_r^*(x_2)$$

即

$$\left|\varepsilon_r^*(x_1-x_2)\right|\le\left|\frac{x_1^*}{x_1^*-x_2^*}\right|\cdot\left|\varepsilon_r^*(x_1)\right|+\left|\frac{x_2^*}{x_1^*-x_2^*}\right|\cdot\left|\varepsilon_r^*(x_2)\right|$$

当 $x_1^*\approx x_2^*$，即大小接近的两个同号近似值相减时，由上式可知，这时 $\left|\varepsilon_r^*(x_1-x_2)\right|$ 可能会很大，说明计算结果的有效数字将严重丢失，计算精度很低。因此在**实际计算中，应尽量设法避开相近数的相减**。

第二步：乘法运算

由（1）和（2）有

$$\varepsilon(x_1x_2) \approx x_1\varepsilon(x_2) + x_2\varepsilon(x_1) \qquad （5）$$

和

$$\varepsilon_r^*(x_1x_2) \approx \varepsilon_r^*(x_1) + \varepsilon_r^*(x_2) \qquad （6）$$

因此，近似值之积的相对误差等于相乘各因子的相对误差的代数和。

当乘数 x_i^* 的绝对值很大时，乘积的绝对值误差

$$\left|\varepsilon\left(\prod_{i=1}^{n} x_i\right)\right|$$

可能会很大，因此也应设法避免。

第三步：除法运算

由（1）和（2）有

$$\varepsilon\left(\frac{x_1}{x_2}\right) \approx \frac{1}{x_2^*}\varepsilon(x_1) - \frac{x_1^*}{(x_2^*)^2}\varepsilon(x_2)$$

$$= \frac{x_1^*}{x_2^*}[\varepsilon_r^*(x_1) - \varepsilon_r^*(x_2)] \qquad (7)$$

和

$$\varepsilon_r^*\left(\frac{x_1}{x_2}\right) \approx \varepsilon_r^*(x_1) - \varepsilon_r^*(x_2) \qquad (8)$$

由式（8）可知：$$\varepsilon_r^*\left(\frac{x_1}{x_2}\right)\approx\varepsilon_r^*(x_1)-\varepsilon_r^*(x_2)$$

两近似值之商的相对误差等于被除数的相对误差与除数的相对误差之差。

由式（7）可知：$$\varepsilon\left(\frac{x_1}{x_2}\right)=\frac{x_1^*}{x_2^*}[\varepsilon_r^*(x_1)-\varepsilon_r^*(x_2)]$$

当除数 x_2^* 的绝对值很小，接近于零时，商的绝对误差

$$\left|\varepsilon\left(\frac{x_1}{x_2}\right)\right|$$

可能会很大，甚至造成计算机的“溢出”错误，故应设法避免让绝对值太小的数作为除数。

通过以上关于数据误差的传播和估计的分析可知：在实际的计算时，误差的累计可能对计算结果产生很大的影响，甚至导致计算结果不可靠。

下一节我们将对算法的“数值稳定性”进行研究。

小结

- 算法
  - 递推实现
  - 秦九昭算法求解思想
  - 二分法求根
- 误差
  - 绝对误差限
  - 相对误差限
  - 有效数字（会求近似数的有效数字位数）

PART 4

算法的数值稳定性

一般使用“数值稳定性”来评价一个算法舍入误差对计算结果的影响。

定义：如果一个算法在运算过程中舍入误差能得到控制，或者舍入误差的积累不影响产生可靠的计算结果，则称该算法是数值稳定的；否则，称其是数值不稳定的。

思路：为简单起见，往往只考虑某一步计算产生的舍入误差的影响。如运算开始时舍入误差的影响，若一步的误差逐步增大，则多步的误差将更大；反之，若一步的误差逐步减少，则多步误差势必逐步减小。

例：计算积分

$$I_k = e^{-1}\int_0^1 x^k e^x dx\,, k = 0,1,\cdots 7$$

解：利用分部积分公式，得到 I_k 的递推关系

$$I_k = 1 - kI_{k-1}$$

算法1：

$$I_0 = e^{-1}\int_0^1 e^x dx = 1 - \frac{1}{e}$$

$$I_k = 1 - kI_{k-1}, k = 1,2,\cdots 7$$

算法2：若用某种方法（如数值积分法）算得 I_7 的近似值 $\tilde{I}_7$ ，则有：

$$I_{k-1} = \frac{1 - I_k}{k}, k = 7,6,\cdots 1$$

下面分析两种算法的数值稳定性。

对于算法1，能够得到 I_0 的近似值 $\tilde{I}_0$ 。现在只考虑这一步产生误差的影响，假定以后各步计算无误差。

记 $\tilde{I}_k$ 为 I_k 的近似值，则有：

$$I_k = 1 - kI_{k-1}, k = 1,2,\cdots 7$$

$$\tilde{I}_k = 1 - k\tilde{I}_{k-1}, k = 1,2,\cdots 7$$

$$\Longrightarrow \quad I_k - \tilde{I}_k = (-k)(I_{k-1} - \tilde{I}_{k-1}), k = 1,2,\cdots 7$$

$$\Longrightarrow \quad |I_7 - \tilde{I}_7| = 7!\,|I_0 - \tilde{I}_0| = 5040|I_0 - \tilde{I}_0|$$

数值不稳定

说明计算I_7产生的误差是第一步计算I_0的5040倍。

同理，对算法2可得：

$$|I_0 - \tilde{I}_0| = \frac{|I_7 - \tilde{I}_7|}{5040}$$

说明计算I_0产生的误差是计算I_7的5040分之一。算法2是数值稳定的。

表 两种积分方法计算结果的比较

I_k	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
准确值	**0.6321**	**0.3679**	**0.2642**	**0.2073**	**0.1709**	**0.1455**	**0.1268**	**0.1124**
算法1	0.6321	0.3679	0.2642	**0.2074**	**0.1704**	**0.1480**	**0.1120**	**0.2160**
算法2	0.6321	0.3679	0.2642	0.2073	0.1709	0.1455	0.1268	0.1124

- **不稳定的算法不能用，实际计算时应采用数值稳定性好的算法。**
- **实际计算中，应尽量设法避开相近数的相减。**

当实在无法避免时，可用变换计算公式的办法来解决。例如，计算$\sqrt{3.01}-\sqrt{3}$ ，至少取到8位有效数字：

$$\sqrt{3.01}=1.7349352 \qquad \sqrt{3}=1.7320508$$

这样

$$\sqrt{3.01}-\sqrt{3}=2.8844\times10^{-3}$$

才能达到具有四位有效数字的要求。如果变换算式：

$$\sqrt{3.01}-\sqrt{3}=\frac{3.01-3}{\sqrt{3.01}+\sqrt{3}}=\frac{0.01}{1.7349+1.7321}=2.8843\times10^{-3}$$

也能达到结果具有五位有效数字的要求，而这时$\sqrt{3.01}$和$\sqrt{3}$所需的有效位数只要五位，远比直接相减所需有效位数（八位）为少。

例：当x很小时，$\cos x \to 1$，如要求$1-\cos x$的值，可利用三角恒等式

$$1-\cos x = 2\sin^2\left(\frac{x}{2}\right)$$

进行公式变换后再来计算。同理，也可把 $\cos x$ 展开成泰勒级数后，按

$$1-\cos x = \frac{x^2}{2!} - \frac{x^4}{4!} + \cdots$$

来进行计算。这两种算法都避开了两个相近数相减的不利情况。

➢ 实际计算时应避免大数吃掉小数。

当两个绝对值相差很大的数进行加法或减法运算时，绝对值小的数有可能被绝对值大的数“吃掉”从而引起计算结果很不可靠。

例 求一元二次方程$x^2-(10^5+1)x+10^5=0$ 的实数根。

解：采用因式分解法,很容易得到两个根为$x_1=10^5$, $x_2=1$.

若用求根公式，则

$$x=\frac{10^5+1\pm\sqrt{(10^5+1)^2-4\times10^5}}{2}$$

若用5位小数计算机运算，则$10^5\pm1\approx10^5$

求得结果$x_1=10^5$， $x_2=0$是错误的。可改为

$$x_2=\frac{10^5+1-\sqrt{(10^5+1)^2-4\times10^5}}{2}$$

$$=\frac{2\times10^5}{10^5+1+\sqrt{(10^5+1)^2-4\times10^5}}\approx\frac{2\times10^5}{10^5+10^5}=1$$

两者结果不同，因为计算机计算时做加减法要“对阶”，“对阶”的结果使大数吃掉了小数，产生了误差。为了避免由于上述原因引起的计算结果严重失真，可以根据一些具体情况，存在需要把某些算式改写成另一种等价的形式。

- 防止接近零的数做除数。
- 注意简化计算步骤，减少运算次数。

秦九韶算法计算多项式的值

同样一个计算问题,如果能减少运算次数,不但可节省计算机的计算时间,还能减少舍入误差.这是数值计算必须遵从的原则.

避免误差危害的若干原则

- 应选用数值稳定性的计算方法;
- 避免两个相近的数相减;
- 要防止大数“吃掉”小数;
- 要避免除数绝对值远小于被除数绝对值的除法;
- 简化计算步骤和公式，设法减少运算次数。

知识结构图一

内容小结

- 误差及算法
  - 误差
    - 分类
      - 舍入误差的产生及定义
      - 截断误差的产生及定义
    - 度量
      - 绝对误差（限）
      - 相对误差（限）
      - 有效数字
      - 三者的联系
    - 传播
      - 一元函数
      - 二元算术运算
      - n元函数
      - 计算函数值问题的条件数
  - 算法
    - 数值稳定性概念
    - 算法设计注意要点

算法设计遵循的条件：

（1）应选用数值稳定性的计算方法；

（2）简化计算步骤和公式，设法减少运算次数；

（3）合理安排运算顺序，防止大数淹没小数；

（4）避免两相近数相减；

（5）绝对值太小的数不宜作为除数。

20XXPOWERPOINT

感谢观看 THANGKS!

单击此处添加您的副标题文字