计算方法

对　　象：本科生
主讲人：聂玉峰
2011年8月30日

课程简介

- 教材：数值方法简明教程
  高等教育出版社 2011
- 参考书：
  - Numerical Analysis
    Springer-Verlag R. Kress
  - Introduction to Numerical Analysis
    Springer-Verlag J. Stoer R. Bulirsch
  - 数值分析（第四版），清华大学出版社 李庆杨 等

- 作业：
  - 作业集（**A, B**）
  - 封建湖，车刚明，计算方法典型题分析解集（第三版），西北工业大学出版社，**2001**
  - 封建湖，聂玉峰，王振海，数值分析导教导学导考（第二版），西北工业大学出版社，**2006.7**

课堂要求

- 手机关机；
- 课堂上严禁小声讨论；
- 上课期间，不明白可以随时提问，问问题时不需要站起来，不需要举手，只需要大声些；

第一章　　绪　　论

内容提要

- 误差与算法
  - 误差
    - 误差的产生
      - 舍入误差
      - 截断误差
    - 度量
      - 绝对误差（限）
      - 相对误差（限）
      - 有效数字
    - 传播
      - 一元函数
      - 多元函数
  - 算法
    - 数值方法的收敛性
    - 数值方法的稳定性
    - 算法设计要点

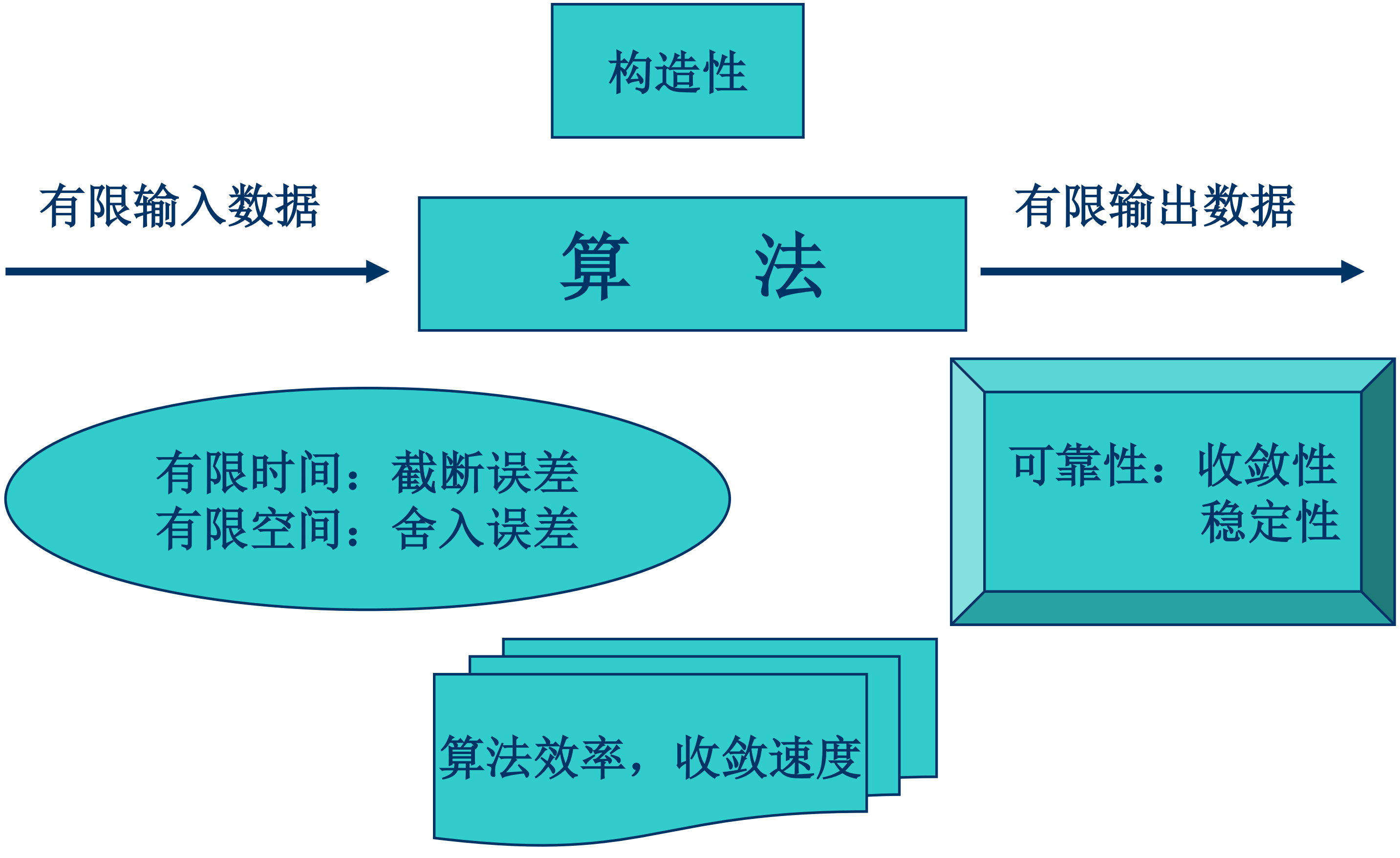
一、 引 言

数学模型

理论分析
（适定性）

非线性方程的近似求解方法；
线性代数方程组的求解方法；
函数的插值近似和数据的拟合近似；
积分和微分的近似计算方法；
常微分方程初值问题的数值解法；
矩阵特征值与特征向量的近似计算方法；

1　算法的基本特点



构造性
有限输入数据
算　　法
有限输出数据
有限时间：截断误差
有限空间：舍入误差
可靠性：收敛性
稳定性
算法效率，收敛速度

2 误差举例

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \cdots, \quad e_{\mathrm{n}} = 1 + \frac{1}{1!} + \frac{1}{2!} + \cdots + \frac{1}{\mathrm{n}!}, \quad e_n - e$$

$$e^* - e = (e^* - e_{\mathrm{n}}) + (e_{\mathrm{n}} - e)$$

舍入误差　　　　截断误差

3　算法效率举例

- 秦九韶算法（公元**1202-1261**）

$$a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$$

$$= (((a_4x + a_3)x + a_2)x + a_1)x + a_0$$

$$s = a_n$$

$$t = a_{n-k} \qquad s = sx + t \qquad (k = 1,2,\cdots,n)$$

$$3^{16} = 3^8 * 3^8 = 3^4 * 3^4 * 3^8 = 3^2 * 3^2 * 3^4 * 3^8$$

$$= 3 * 3 * 3^2 * 3^4 * 3^8$$

;

$$s = x$$

$$s = s \cdot s$$

4 算法应用状态

数值分析研究对象以及解决问题方法的广泛适用性，著名流行软件如**Maple**、**Matlab**、**Mathematica**等已将其绝大多数内容设计成函数，简单调用之后便可以得到运行结果。

但由于实际问题的具体特征、复杂性, 以及算法自身的适用范围决定了应用中必须选择、设计适合于自己特定问题的算法，因而掌握数值方法的思想和内容是至关重要的。

二、误差的度量与传播

- 误差度量
  - 绝对误差
  - 相对误差
  - 有效数字
- 误差传播
  - 初值误差传播

1 误差度量：绝对误差

- 绝对误差定义： 近似值——真值

$$x^* - x =: e(x^*)$$

如果正数 $\varepsilon = \varepsilon(x^*)$ 是绝对误差绝对值的上界，即

$$|e^*| = |x^* - x| \le \varepsilon\ ,$$

则称 ε 为 x^* 近似 x 的一个**绝对误差限**。

Remark：实际计算中所要求的误差，是指估计一个尽可能小的绝对误差限。

1 误差度量：相对误差

定义 设 x^* 是对准确值 x（$\neq 0$）的一个近似，称

$$e_r(x^*) = \frac{x^* - x}{x} = \frac{e(x^*)}{x}$$

为 x^* 近似 x 的相对误差。

$$e_r(x^*) - \frac{e(x^*)}{x^*} = O\left(\left(e_r^*\right)^2\right)$$

相对误差限：数值$\left|e(x^*)\right|$的上界，记为 $\varepsilon_r(x^*)$

相对误差限也可以通过$\varepsilon_r(x^*) = \varepsilon(x^*) / \left|x^*\right|$来计算

Remark：当要求计算相对误差，实际上是指估计一个尽可能小的相对误差限。

1 误差度量：有效数字

定义 1.3 设量 x 的近似值 x^* 有如下标准形式

$$x^* = \pm 10^m \times 0.\overline{a_1 a_2 \cdots a_n \cdots a_p}$$
$$= \pm\left(a_1 \times 10^{m-1} + a_2 \times 10^{m-2} + \cdots + a_n \times 10^{m-n} + \cdots + a_p \times 10^{m-p}\right)$$

其中$\{a_i\}_{i=1}^p \subset \{0,1,\cdots,9\}$且$a_1 \neq 0$，$m$为近似值$x^*$的量级．如果使不等式

$$\left|x^* - x\right| \leq \frac{1}{2} \times 10^{m-n}$$

成立的最大整数为 n，则称近似值x^*具有 **n 位有效数字**，它们分别是a_1、a_2、… 和 a_n．特别地，如果有$n = p$，即最后一位数字也是有效数字，则称x^*是**有效数**.

例 1.1 设量 $x=\pi$ 有近似值 $x_1^*=3.141$， $x_2^*=3.142$． 试回答这两个近似值分别有几位有效数字，它们是有效数吗？

$$\left| x-x_1^* \right| = 0.00059\cdots \le 0.005 = \frac{1}{2}\cdot 10^{1-3}$$

3位有效数字，非有效数

$$\left| x-x_2^* \right| = 0.00040\cdots \le 0.0005 = \frac{1}{2}\cdot 10^{1-4}$$

4位有效数字，有效数

注1：有效数的误差限是末位数单位的一半，可见有效数本身就体现了误差界。

注2: 对真值进行四舍五入得到有效数。

注3: 从实验仪器所读的近似数（最后一位是估计位）不是有效数，估计最后一位是为了确保对最后一位进行四舍五入得到有效数。

例 从最小刻度为厘米的标尺读得的数据123.4cm是为了得到有效数123.cm,读得数据156.7cm是为了得到有效数157.cm。

2. 误差传播：初值误差传播

- 概念：近似数参加运算后所得值一般也是近似值，含有误差，将这一现象称为误差传播 。
- 误差传播的表现：
  - 算法本身可能有截断误差；
  - 初始数据在计算机内的浮点表示一般有舍入误差；
  - 每次运算一般又会产生新的舍入误差，并传播以前各步已经引入的误差；
  - 误差有正有负，误差积累的过程一般包含有误差增长和误差相消的过程，并非简单的单调增长；
  - 运算次数非常之，不可能人为地跟踪每一步运算。

初值误差传播（续）

- 初值误差传播：假设每一步都是准确计算，即不考虑截断误差和数据表示引入的舍入误差，仅研究初始数据的误差传播规律。
  - 研究方法：
    - 泰勒（**Taylor**）方法 复习公式
      - **n**元函数
      - 二元函数（算术运算）
      - 一元函数（计算函数值的条件数）

对于函数 $y = f(x_1, x_2, \cdots, x_n)$ 有近似值 $y^* = f(x_1^*, x_2^*, \cdots, x_n^*)$，利用在点 $(x_1^*, x_2^*, \cdots, x_n^*)$ 处的 Taylor(泰勒)公式

（当 x_1^*、x_2^*、…、x_n^* 很好地近似了相应真值）

$$e(y^*) = y^* - y \approx \sum_{i=1}^{n} f_i(x_1^*, \cdots, x_n^*)(x_i^* - x_i)$$

$$= \sum_{i=1}^{n} f_i(x_1^*, \cdots, x_n^*) e(x_i^*)$$

Taylor formula

相对误差

泰勒方法（n元函数）

$$e_r(y^*) = \frac{e(y^*)}{y^*} \approx \sum_{i=1}^{n} f_i(x_1^*, \cdots, x_n^*) \frac{x_i^*}{y^*} \frac{e(x_i^*)}{x_i^*}$$

$$= \sum_{i=1}^{n} f_i(x_1^*, \cdots, x_n^*) \frac{x_i^*}{y^*} e_r(x_i^*)$$

对于一元函数 $y = f(x)$，有初值误差传播近似计算公式

$$e(y^*) \approx f'(x^*)e(x^*)$$

$$e_r(y^*) \approx f'(x^*)\frac{x^*}{y^*}e_r(x^*)$$

例 1.2 试建立函数 $y = f(x_1, x_2, \cdots, x_n) = x_1 + x_2 + \cdots + x_n$ 的绝对误差(限)、相对误差的近似传播公式，以及 $\left\{x_i^*\right\}_{i=1}^{n}$ 同号时的相对误差限传播公式.

$$e(y^*) \approx \sum_{i=1}^{n} f_i(x_1^*, x_2^*, \cdots, x_n^*) e(x_i^*) = \sum_{i=1}^{n} e(x_i^*)$$

$$e_r(y^*) \approx \sum_{i=1}^{n} f_i(x_1^*, x_2^*, \cdots, x_n^*) \frac{x_i^*}{y^*} e_r(x_i^*) = \sum_{i=1}^{n} \frac{x_i^*}{y^*} e_r(x_i^*)$$

$$\left|e(y^*)\right| \approx \left|\sum_{i=1}^{n} e(x_i^*)\right| \le \sum_{i=1}^{n} \left|e(x_i^*)\right| \le \sum_{i=1}^{n} \varepsilon(x_i^*) \qquad \varepsilon(y^*) \approx \sum_{i=1}^{n} \varepsilon(x_i^*)$$

$$\varepsilon_r(y^*) \approx \frac{\sum_{i=1}^{n} \varepsilon(x_i^*)}{\left|y^*\right|} = \sum_{i=1}^{n} \left|\frac{x_i^*}{y^*}\right| \varepsilon_r(x_i^*) \le \max_{1\le i\le n} \varepsilon_r(x_i^*) \sum_{i=1}^{n} \left|\frac{x_i^*}{y^*}\right|$$

$$= \max_{1\le i\le n} \varepsilon_r(x_i^*) \sum_{i=1}^{n} \frac{x_i^*}{y^*} = \max_{1\le i\le n} \varepsilon_r(x_i^*)$$

例题 1.3

例 1.3 使用足够长且最小刻度为 1mm 的尺子，量得某桌面长的近似值 $a^* = 1304.3\ mm$，宽的近似值 $b^* = 704.8\,mm$ (数据的最后一位均为估计值). 试求桌子面积近似值的绝对误差限和相对误差限.

长和宽的近似值的最后一位都是估计位，尺子的最小刻度是毫米，故有误差限

$$\varepsilon(a^*) = 0.5\,mm，\quad \varepsilon(b^*) = 0.5\,mm$$

面积 $S = ab$，由式(1.7)得到近似值 $S^* = a^* b^*$ 的绝对误差近似为

$$e(S^*) \approx b^* e(a^*) + a^* e(b^*)$$

进而有绝对误差限

$$\varepsilon(S^*) \approx \left|b^*\right| \varepsilon(a^*) + \left|a^*\right| \varepsilon(b^*)$$

$$= 704.8 \times 0.5 + 1304.3 \times 0.5 = 1004.55 \quad mm^2$$

相对误差限

$$\varepsilon_r(S^*) \approx \frac{\varepsilon(S^*)}{S^*} = \frac{1004.55}{1304.3 \times 704.8} \approx 0.0011 = 0.11\%$$

记点$(x_1^*, x_2^*, \cdots, x_n^*)$为 p*，点$(x_1, x_2, \cdots, x_n)$为 p，n 元泰勒公式

$$f(p) = f(p^*) + \frac{1}{1!}\left[f_1(p^*)(x_1 - x_1^*) + \cdots + f_n(p^*)(x_n - x_n^*)\right] +$$

$$\frac{1}{2!}\left[f_{11}(p^*)(x_1 - x_1^*)^2 + \cdots + f_{1n}(p^*)(x_1 - x_1^*)(x_n - x_n^*) +\right.$$

$$+ f_{21}(p^*)(x_2 - x_2^*)(x_1 - x_1^*) + \cdots + f_{2n}(p^*)(x_2 - x_2^*)(x_n - x_n^*) +$$

$$+ \cdots$$

$$\left.+ f_{n1}(p^*)(x_n - x_n^*)(x_1 - x_1^*) + \cdots + f_{nn}(p^*)(x_n - x_n^*)^2\right]$$

$$+ \cdots$$

建立公式　　Return

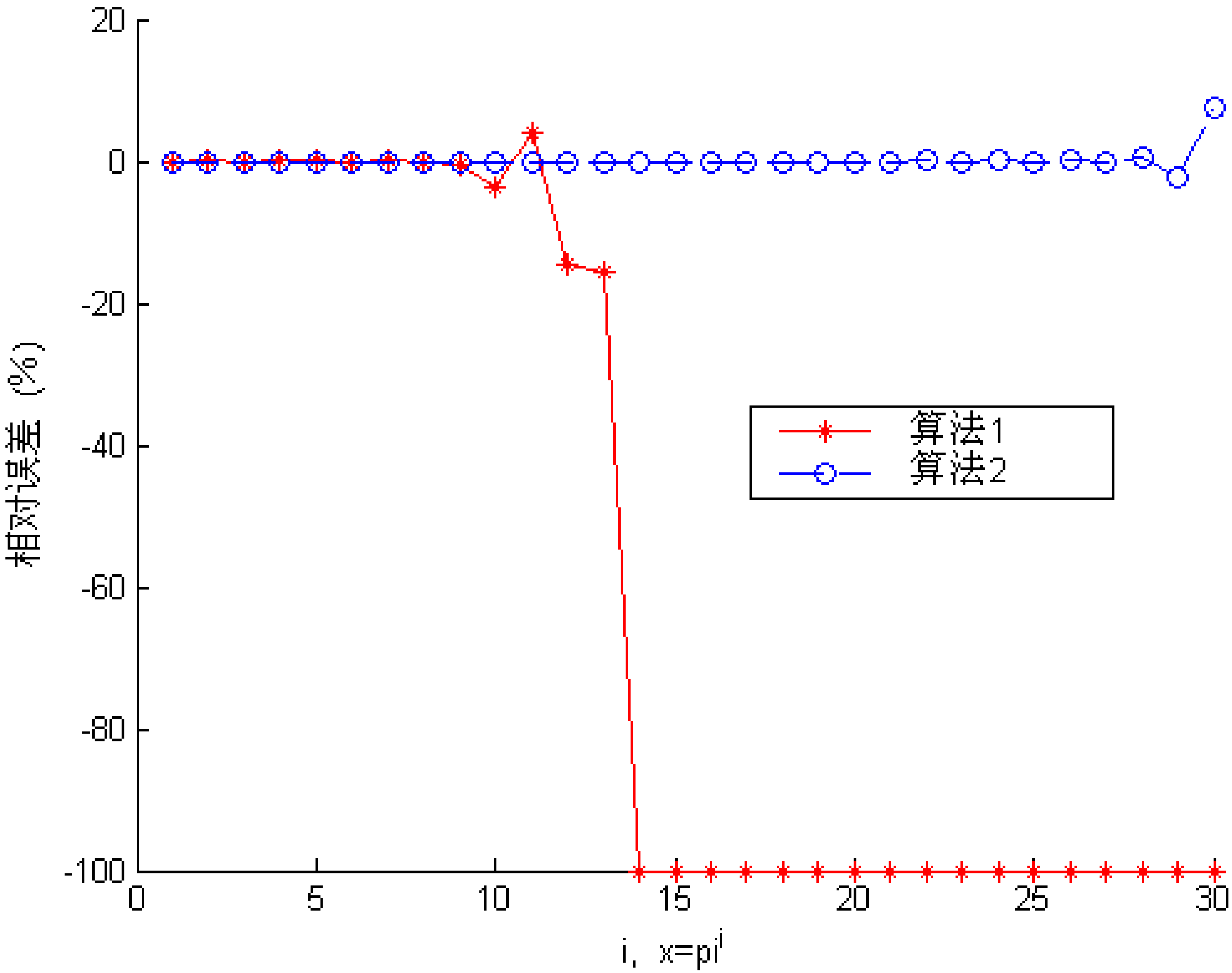
三、数值试验与算法性能比较

例 1.4 对于表达式 $\frac{1}{x}-\frac{1}{x+1}$ 和 $\frac{1}{x(x+1)}$，在计算过程中保留 7 位有效数字，研究对不同的 x，两种计算公式的计算精度的差异.

算法 1： $y_1(x)=\frac{1}{x}-\frac{1}{x+1}$

算法 2： $y_2(x)=\frac{1}{x(x+1)}$

选取点集 $\{\pi^i\}_{i=1}^{30}$ 中的点作为 x



算法设计时，避免相近数相减

•当|x|>>1时

$$\sqrt{x+1}-\sqrt{x}=\frac{1}{\sqrt{x+1}+\sqrt{x}}$$

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

$$\ln(x+1)-\ln x=\ln\frac{x+1}{x}$$

•当|x|<<1时

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

$$\arctan x-x=-\frac{x^3}{3}+\frac{x^5}{5}-\cdots$$

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

例1.5　试用不同位数的浮点数系统求解如下线性方程组

$$\begin{cases} 0.00001x_1 + 2x_2 = 1 \\ 2x_1 + 3x_2 = 2 \end{cases}$$

算法 1　首先用第一个方程乘以适当的系数加至第二个方程，使得第二个方程的 x_1 的系数为零，这时可解出 x_2；其次将 x_2 带入第一个方程，进而求得 x_1 (高斯消元法)．当用 4 位和 7 位尾数的浮点运算实现该算法，分别记之为算法 1a 和算法 1b．

算法 2　首先交换两个方程的位置，其次按算法 1 计算未知数 (选主元的高斯消元法)．当用 4 位和 7 位尾数的浮点运算实现该算法，分别记之为算法 2a 和算法 2b．

精确解为 $x_1 = 0.25000187\ldots$， $x_2 = 0.49999874\ldots$

对例 1.5 用不同算法的计算结果比较

例 1.5	x_1^*	$\varepsilon_r(x_1^*)$	x_2^*	$\varepsilon_r(x_2^*)$
算法 1a	**0.0000**	0.10×10^{1}	**0.5000**	0.25×10^{-7}
算法 2a	**0.2500**	0.75×10^{-7}	**0.5000**	0.25×10^{-7}
算法 1b	**0.2600000**	0.40×10^{-1}	**0.4999987**	0.10×10^{-6}
算法 2b	**0.2500020**	0.50×10^{-8}	**0.5000000**	0.25×10^{-7}

$$\frac{-2}{0.00001}\times2+3=-0.4\times10^{6}+0.3\times10^{1}=-0.4\times10^{6}+0.000003\times10^{6}$$

$$\frac{-2}{0.00001}\times1+2=-0.2\times10^{6}+0.2\times10^{1}=-0.2\times10^{6}+0.000002\times10^{6}$$

$$\begin{cases}0.1\times10^{4}x_1+0.2\times10^{1}x_2=0.1\times10^{1}\\ \qquad\qquad -0.4\times10^{6}x_2=-0.2\times10^{6}\end{cases}$$

- 尽可能避免用绝对值较大的数除以绝对值较小的数

- 尽可能避免大数吃小数

合理安排量级相差很大的数之间的运算次序，尽可能避免大数“吃掉”小数。

➢例 $$987654321. + \sum_{k=1}^{1000000} \delta_k \qquad (0 < \delta_k \le 1)$$

方法1: $987654321. + \delta_1 + \delta_2 + \cdots + \delta_{1000000}$

方法2: $987654321. + (\delta_1 + \delta_2 + \cdots + \delta_{1000000})$

方法3: $987654321. + \left(\sum_{i=1}^{1000} \delta_i + \cdots + \sum_{i=999000}^{1000000} \delta_i \right)$

例 1.6 计算积分 $I_n = \int_0^1 \frac{x^n}{x+5} dx$ 有递推公式

$I_n = \frac{1}{n} - 5I_{n-1}\ (n = 1,2,\cdots)$，已知 $I_0 = \ln\frac{6}{5}$. 采用 IEEE 双精度浮点数，分别用如下两种算法计算 I_{30} 的近似值.

算法 1 取 I_0 的近似值为 $I_0^* = 0.18232155679395$

按递推公式 $I_n^* = \dfrac{1}{n} - 5I_{n-1}^*$ 计算 I_{30}^*

算法 2 $\dfrac{1}{6\times(39+1)} = \int_0^1 \dfrac{x^{39}}{6}dx < I_{39} < \int_0^1 \dfrac{x^{39}}{5}dx = \dfrac{1}{5\times(39+1)}$

取 I_{39} 的近似值为

$$I_{39}^* = \frac{1}{2}\left(\frac{1}{240} + \frac{1}{200}\right) \approx 0.0045833333333$$

按递推公式 $I_{n-1}^* = \dfrac{1}{5}\left(\dfrac{1}{n} - I_n^*\right)$ 计算 I_{30}^*

表 1.2 例 1.6 的计算结果

n	算法 1		n	算法 2	
	I_n^*	$\lvert I_n^* - I_n \rvert$		I_n^*	$\lvert I_n^* - I_n \rvert$
1	8.8392e-002	1.9429e-016	39	4.5833e-003	3.9959e-004
2	5.8039e-002	9.8532e-016	38	4.2115e-003	7.9919e-005
3	4.3139e-002	4.9197e-015	37	4.4209e-003	1.5984e-005
4	3.4306e-002	2.4605e-014	36	4.5212e-003	3.1967e-006
5	2.8468e-002	1.2304e-013	35	4.6513e-003	6.3935e-007
6	2.4325e-002	6.1520e-013	34	4.7840e-003	1.2787e-007
…	…	…	33	4.9255e-003	2.5574e-008
25	1.1740e+001	1.1734e+001	32	5.0755e-003	5.1148e-009
26	-5.8664e+001	5.8670e+001	31	5.2349e-003	1.0230e-009
27	2.9336e+002	2.9335e+002	30	5.4046e-003	2.0459e-010
28	-1.4667e+003	1.4668e+003			
29	7.3338e+003	7.3338e+003			
30	-3.6669e+004	3.6669e+004			

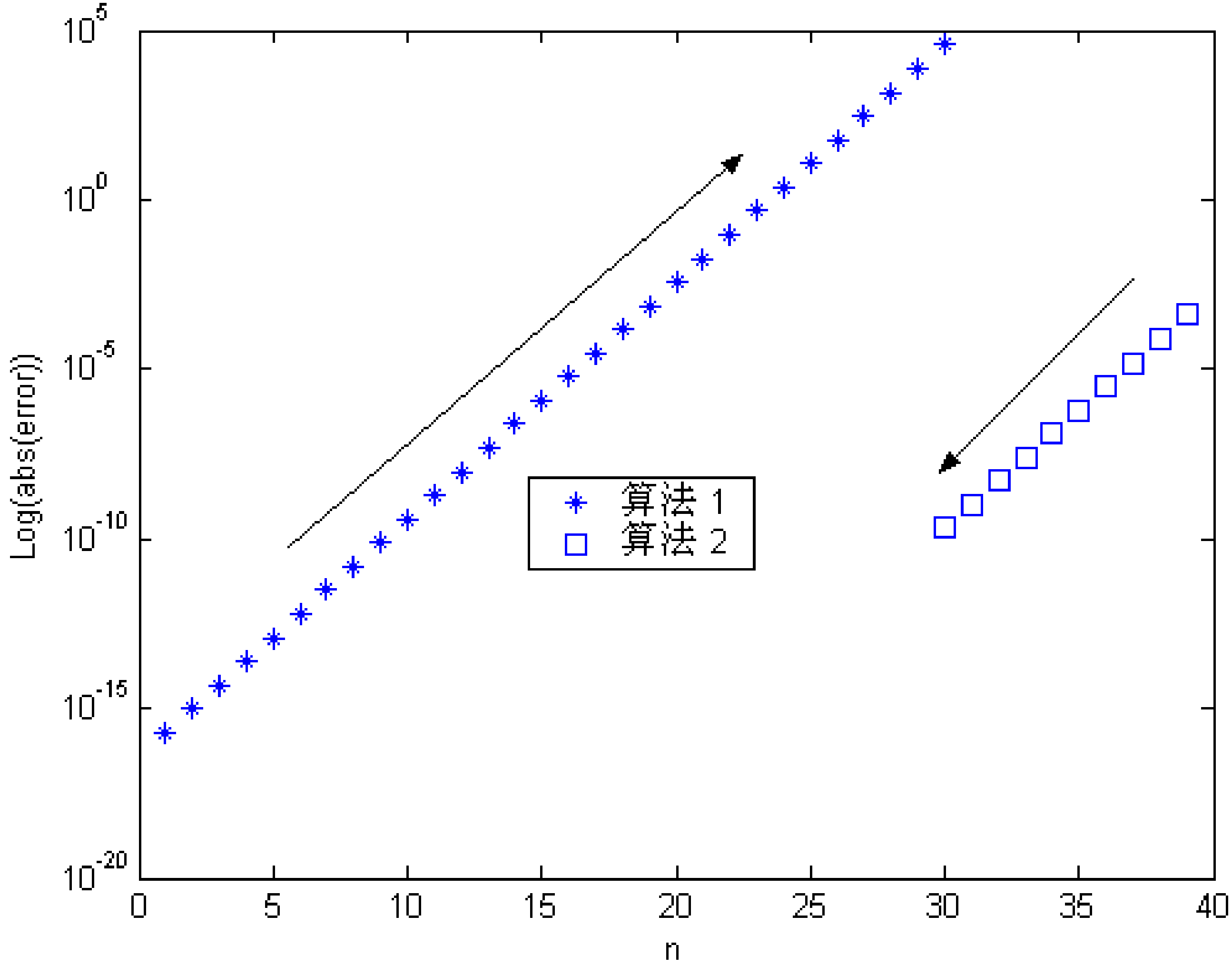


图 1.3　例 1.6 用不同算法计算结果的误差绝对值的对数图

选用数值稳定的算法

$$I_n^* = \frac{1}{n} - 5I_{n-1}^*$$

$$\left|I_n^* - I_n\right| \approx 5\left|I_{n-1}^* - I_{n-1}\right| \approx 5^2\left|I_{n-2}^* - I_{n-2}\right| \approx \cdots \approx 5^n\left|I_0^* - I_0\right|$$

$$I_{n-1}^* = \frac{1}{5}\left(\frac{1}{n} - I_n^*\right)$$

$$\left|I_n^* - I_n\right| \approx \left|I_{n+1}^* - I_{n+1}\right|/5 \approx \left|I_{n+2}^* - I_{n+2}\right|/5^2 \approx \cdots \approx \left|I_{n+m}^* - I_{n+m}\right|/5^m$$