§1.2　预备知识和误差

§1.2.1　误差的来源

实际问题　建立数学模型　研究计算方法　编程上机计算　解结果

a)　模型误差　:
在建立数学模型过程中，不可能将所有因素均考虑，必然要进行必要的简化，这就带来了与实际问题的误差。
b)　测量误差　:　测量已知参数时，数据带来的误差。
c)　截断误差　:　在设计算法时，必然要近似处理，寻求一些简化。
d)　舍入误差　:　计算机的字长是有限的，每一步运算均需四舍五入，由此产出的误差称舍入误差。如：π、1/3，……取小数点8位、16位。

[截断误差的实例]

例　1.4：$\cos x = 1 - \frac{x^2}{2} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots + \frac{(-1)^n x^{2n}}{(2n)!} + \cdots$

当$|x|$很小时，可用$1-\frac{x^2}{2}$作为$\cos x$的近似值，其截断误差小于$\frac{x^4}{24}$。

例　1.5：　已知：$e^x = 1 + x + \frac{1}{2!}x^2 + \frac{1}{3!}x^3 + \cdots + \frac{1}{n!}x^n + \cdots$,

求e^{-1}的近似值，并估计误差。

分析：对函数$f(x)$用 Taylor展开，用多项式

$$P_n(x) = f(0) + \frac{f'(0)}{1!}x + \frac{f''(0)}{2!}x^2 + \cdots + \frac{f^{(n)}(0)}{n!}x^n$$

近似代替，则数值方法的截断误差为

$$R_n(x) = f(x) - P_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} x^{n+1}$$ 。

解： 利用展开式的前三项，取 n=2，

$$e^{-1} \approx 1 + (-1) + \frac{1}{2}(-1)^2 = 0.5$$

由Taylor公式

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \cdots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n + \frac{f^{(n+1)}(\xi)}{(n+1)!}(x - x_0)^{n+1}$$

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

截断误差为：

$$|R_2| = |e^{-1} - 0.5| \approx \frac{1}{3!} < 1.7 * 10^{-1}$$

数值计算方法主要讨论截断误差和舍入误差的影响，不讨论模型误差和测量误差。

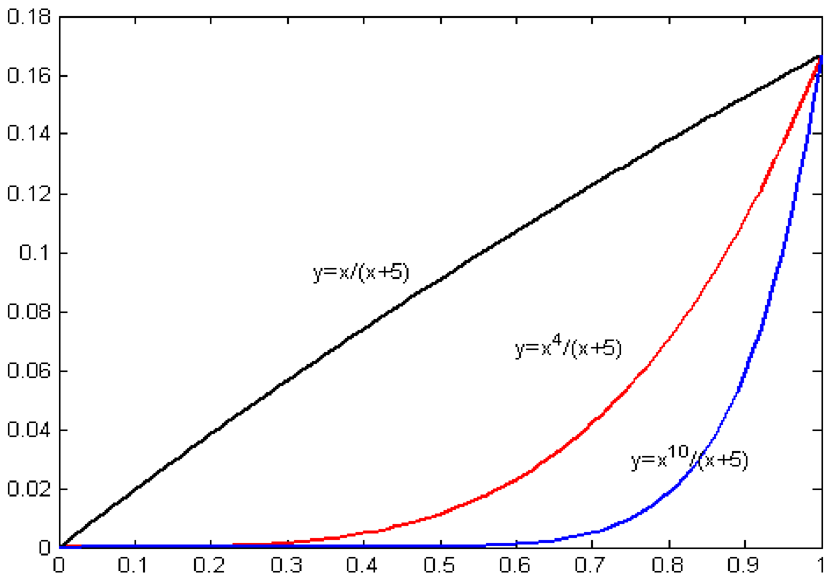
§1.2.2 误差分析的重要性以及数值稳定性

一个数值方法进行计算时，由于原始数据有误差，在计算中这些误差会传播，有时误差增长很快使计算结果误差很大，影响了结果不可靠.

定义 一个算法如果原始数据有扰动(即误差)，而计算过程舍入误差不增长，则称此算法是数值稳定的. 否则，若误差增长则称算法不稳定.

例如：**计算并分析误差**

$$I_n = \int_0^1 \frac{x^n}{x+5}dx \quad n=0,1,2,\cdots$$



解：

由积分估值

$$\begin{aligned} I_n &= \int_0^1 \frac{(x^n + 5x^{n-1}) - 5x^{n-1}}{x+5}dx \\ &= \int_0^1 x^{n-1}dx - 5\int_0^1 \frac{x^{n-1}}{x+5}dx \\ &= \frac{1}{n} - 5I_{n-1} \end{aligned}$$

由积分性质知 $$\frac{1}{6(n+1)}=\min_{0\le x\le 1}\left(\frac{1}{x+5}\right)\int_0^1 x^n dx < I_n < \max_{0\le x\le 1}\left(\frac{1}{x+5}\right)\int_0^1 x^n dx=\frac{1}{5(n+1)}$$

设计如下两种算法:

[算法 1]:

取 $I_0=\int_0^1\frac{1}{x+5}dx=\ln 1.2$，按公式

$$I_n=\frac{1}{n}-5I_{n-1}\quad（n=0，1，2……）$$

依次计算 $I_1,I_2\cdots$的近似值。

设 $e_0=I_0-I_0^*$。

假设计算过程中不产生新的舍入误差，则有

$$e_n=I_n-I_n^*=-5I_{n-1}+5I_{n-1}^*=-5e_{n-1}\quad（n=0，1，2……）$$

$\Rightarrow e_n=(-5)^n e_0$ 误差扩散。

[算法 2]:

从 I_k 计算 I_{k-1}，应有

$e_{k-1}=-\frac{1}{5}e_k \Rightarrow e_0=(-\frac{1}{5})^n e_n$ 。

在运算过程中，舍入误差不增大,数值稳定。

1 $I_n = \frac{1}{n} - 5I_{n-1}$, $I_0 = \ln\frac{6}{5}$ $\tilde{I}_n$

2 $I_{n-1} = \frac{1}{5}\left(\frac{1}{n} - I_n\right)$, $I_8 = 0.019$ $\bar{I}_n$

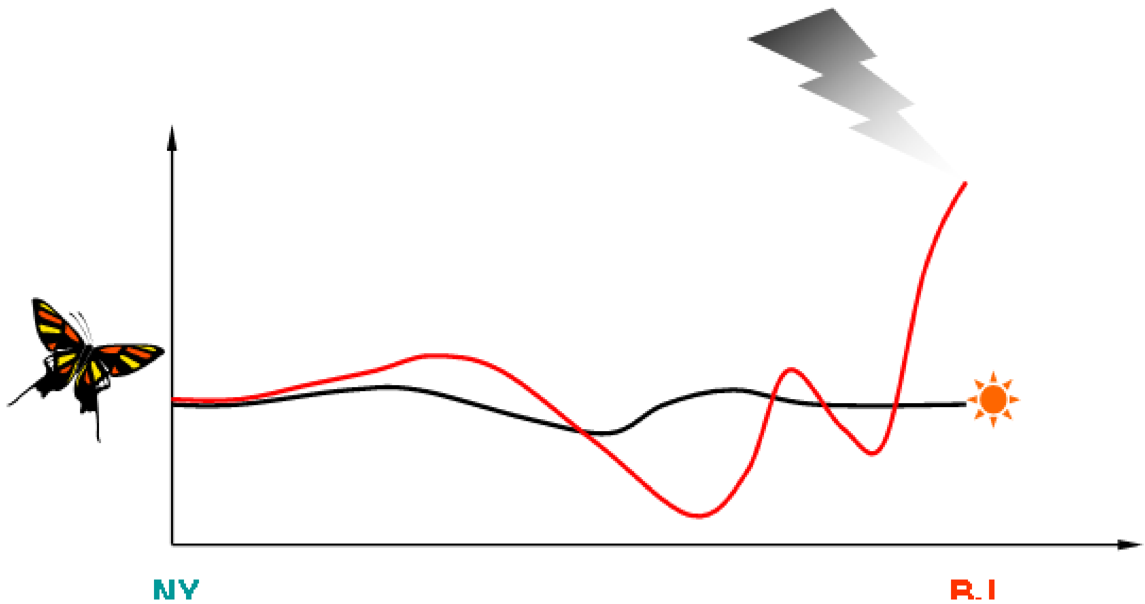
n	I_n	$\tilde{I}_n$	$\bar{I}_n$
0	0.182	0.182	0.182
1	0.088	0.090	0.088
2	0.058	0.050	0.058
3	0.0431	0.083	0.0431
4	0.0343	-0.165	0.0343
5	0.0284	1.025	0.0284
6	0.024	-4.958	0.024
7	0.021	24.933	0.021
8	0.019	-124.540	0.019

[关于数值稳定性的算法]

- 误差的传播与积累

例：蝴蝶效应 —— 纽约的一只蝴蝶翅膀一拍，风和日丽的北京就刮起台风来了？！

- 以上是一个病态问题



例 5：$I_n = e^{-1}\int_0^1 x^n e^x dx, \qquad n = 0,1,2,\cdots$，

解：用分部积分公式得递推式：

$$I_n = 1 - nI_{n-1}, I_0 = 1 - e^{-1}$$。

用四位有效数字计算：$I_0 = 0.6321$，

$I_1 = 1 - I_0 = 0.3679$，$I_2 = 1 - 2I_1 = 0.2642$，

$I_3 = 1 - 3I_2 = 0.2074$, $I_4 = 1 - 4I_3 = 0.1704$ ，

$I_5 = 1 - 5I_4 = 0.1480$, $I_6 = 1 - 6I_5 = 0.1120$ ，

$I_7 = 1 - 7I_6 = 0.2160$, $I_8 = 1 - 8I_7 = -0.7280$.

分析 1：

可以估计出 $0 < \dfrac{e^{-1}}{n+1} < I_n < \dfrac{1}{n+1}$ 故

$0{,}0460 < I_7 < 0.1250$, $0.0409 < I_8 < 0.1111$。

于是 I_7, I_8 与精确值已经面目全非,一位有效数字也没有。

这是由于如果 I_0 有误差 $e = 0.5\times10^{-4}$,不计中间再产生的舍入误差,该误差随着计算过程分别乘以 $2,3,\cdots,7,8$,到 I_8 时已经变成了 $8!\,e = 40320\,e$,误差扩大了 4 万倍。因而该算法不是稳定的。

分析 2：

如果递推式改为,由

$$I_{n-1} = \frac{1}{n}\left(1 - I_n\right),\ I_7 = 0.1124\ ,$$

逐步计算 $I_6, I_5, \cdots$,直到 $I_0 = 0.6321$。计算结果有四位有效数字,如果 I_7 有误差 e,其传播到 I_0 所引起的误差仅为 $\dfrac{1}{7!}e = \dfrac{1}{5040}e$。故该算法是稳定