第一章 数值计算的误差

刘可伋

上海财经大学 数学学院

一、误差的来源

观测误差

↓

实际问题 → 数学模型 → 数值计算方法 → 上机求结果

↑

模型误差

算法误差
（截断、迭代初值选择、
参数选取等导致的误差）

↓

舍入误差
（机器字长限制）

↑

误差是用来描述数值计算中近似解的精确程度

引例

例1 为计算地球的表面积，可采用$S = 4\pi r^2$，其中r为地球半径.

那么由此计算会产生以下误差：

① 将地球近似看成球体 $\longrightarrow$ 模型误差

② 取$r \approx 6370$千米 $\longrightarrow$ 观测误差

③ 计算$4\pi r^2$涉及浮点数乘法 $\longrightarrow$ 舍入误差

例2 用泰勒（Taylor）多项式

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

近似代替可导函数 $f(x)$，则**截断误差**为：

$$R_n(x) = f(x) - P_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}x^{n+1}, \qquad \xi \in (0, x).$$

例3 用3.14159近似代替 π ，则舍入误差为：

$$e = \pi - 3.14159 = 0.0000026\cdots$$

例4 近似计算 $\int_0^1 e^{x^2}\,dx.$

解：原式 $= \int_0^1 \left(1 + x^2 + \frac{x^4}{2!} + \frac{x^6}{3!} + \cdots\right) dx$

$$= \underbrace{1 + \frac{1}{3} + \frac{1}{2!}\cdot\frac{1}{5}}_{\text{近似值 } S_4} + \underbrace{\frac{1}{3!}\cdot\frac{1}{7} + \cdots}_{\text{截断误差 } R_4}$$

$$\approx 1 + \underbrace{0.333}_{\text{保留小数点后3位数字（舍入误差：0.0003333）}} + 0.1 = 1.433$$

1.1 绝对误差的定义

定义：设 x 为准确值，x^* 为它的一个近似值，则称

$$e^* = x^* - x$$

为近似值的绝对误差，简称误差.

注记：有量纲！

注记：1、实际应用中，通常 x 未知

2、由于 x 未知，那么绝对误差 e^* 也未知

3、绝对误差可正可负

1.2 绝对误差限的定义

定义：若能根据测量工具或计算情况，有以下估计

$$|x^* - x| \leq \varepsilon^*$$

则称 ε^* 为**绝对误差限**，简称**误差限**. **注记：有量纲！**

由 ε^* 可推测 x 的范围，如：毫米刻度尺测得 x^* 为765mm，若它误差限 ε^* 为0.5mm，则准确值 x 的范围是[764.5mm, 765.5mm].

通常也可记： $x = x^* \pm \varepsilon^*$，如上例可写为：$x = 765 \pm 0.5$ mm.

注记：考察近似值的精确程度不能仅看**误差**或**误差限**！

示例

有两个量 $x = 10 \pm 1$，$y = 1000 \pm 5$，则

$$x^* = 10, \quad \varepsilon_x^* = 1; \quad y^* = 1000, \quad \varepsilon_y^* = 5$$

可见 ε_y^* 比 ε_x^* 大**4倍**，但

$$\frac{\varepsilon_y^*}{y^*} = \frac{5}{1000} = 0.5\%, \quad \frac{\varepsilon_x^*}{x^*} = \frac{1}{10} = 10\%$$

可见 y^*近似 y 的程度比 x^*近似 x 的程度好！

注记：除了考虑误差大小外，还要考虑准确值大小！

1.3 相对误差的定义

定义： 设 x 为**准确值**，x^* 为其**近似值**，称

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

为近似值的**相对误差**.

注记：无量纲！

实际计算中，因 x 未知，常取**相对误差**为：

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

注记：相对误差可正可负

差别：$e_r - e_r^* = \frac{e^*}{x} - \frac{e^*}{x^*} = \frac{e^*(x^* - x)}{x^* x} = \frac{(e^*)^2}{x^*(x^* - e^*)} = \frac{(e^*/x^*)^2}{1 - e^*/x^*} = \frac{(e_r^*)^2}{1 - e_r^*}$

1.4 相对误差限的定义

定义：若 $\exists\ \varepsilon_r^*$ 使

$$\boxed{|e_r^*| = \left|\frac{e^*}{x^*}\right| \le \varepsilon_r^*}$$

成立，则称 ε_r^* 为**相对误差限**. 实际上，$\varepsilon_r^* = \varepsilon^*/|x^*|$.

注记：无量纲!

当 x 位数较多时，常四舍五入取其近似值，如：

$$x = \pi = 3.1415926\cdots$$

取三位近似：$x_3^* = 3.14$；取五位近似：$x_5^* = 3.1416$.

（3 位有效数字）（5 位有效数字）

判断：从左往右第一个非 0 数开始数，直至最后一个数字为止，如果总共有 n 个数字，就称有 n 位有效数字?

1.5 有效数字的定义

定义：若近似值 x^* 可表示为

$$\boxed{x^* = \pm a_1.a_2a_3\cdots a_n\times10^m}$$

其中 $a_i(i=1,2,\cdots,n)$ 是0到9中的一个数，$a_1\neq0$，且

$$\boxed{|x-x^*|\le0.5\times10^{m-n+1}}$$

注记：m 相同时，
n 越大，即有效位数越多，
则绝对误差限越小，
近似值越精确.

则 x^* 有 n 个有效数字.

推论：若 $|x-x^*|\le0.5\times10^{m-n+1}$，则 x^*至少有 n 个有效数字.

例4 $|x_3^*-\pi|=|3.14\times10^0-3.1415926\cdots|=0.0015\cdots=0.15\cdots\times10^{-2}$

$$0.5\times10^{-3}\le|x_3^*-\pi|\le0.5\times10^{-2},\quad 故\ n=3$$

例5 按照四舍五入原则写出下列各数具有5位有效数字的近似数：

$$187.9325,\qquad 0.03785551,\qquad 8.000033,\qquad 2.7182818$$

解： 按定义，

$$187.93,\qquad 0.037856,\qquad 0000,\ 2.7183$$

例6 重力加速度 $g = 9.80665\cdots\ \mathrm{m/s^2}$，它分别可近似为 $g \approx 9.81\ \mathrm{m/s^2}$和 $g \approx 0.00981\ \mathrm{km/s^2}$，则其有效数字分别是多少？

解： $0.5\times10^{-3} < |g - 9.81\times10^{0}| \le 0.5\times10^{-2} \Rightarrow n = 3$

$0.5\times10^{-6} < |g - 9.81\times10^{-3}| \le 0.5\times10^{-5} \Rightarrow n = 3$

注记：有效数字和小数点后多少位无关！

定理1 设近似值可表示为

$$x^* = \pm a_1.a_2a_3\cdots a_n\times 10^m\ (a_1 \neq 0)$$

若 x^* 有 n 位效数字，则其相对误差限满足

$$\varepsilon_r^* \leq \frac{1}{2a_1}\times 10^{-(n-1)}$$

注记：有效位数越多，相对误差限越小！

反之，若 $\varepsilon_r^* \leq \frac{1}{2(a_1+1)}\times 10^{-(n-1)}$，则 x^* 至少有 n 位效数字.

证明：因 $a_1\times 10^m \leq |x^*| < (a_1+1)\times 10^m$，若 x^* 有 n 位效数字，则

$$\varepsilon_r^* = \frac{|x-x^*|}{|x^*|} \leq \frac{0.5\times 10^{m-n+1}}{a_1\times 10^m} = \frac{1}{2a_1}\times 10^{-(n-1)}，$$ 反之类似可证.

例7 要使 $\sqrt{20}$ 的近似值的相对误差限小于0.1%，需要取几位有效数字？

解： 设取 n 位有效数字，由 $\sqrt{20}=4.4\cdots$ 知 $a_1=4$，则由定理1有

$$\varepsilon_r^* \le \frac{1}{2a_1}\times 10^{-(n-1)} = 0.125\times 10^{1-n} \le 0.1\% = 0.1\times 10^{-2}$$

成立，所以 $1-n=-3$，即 $n=4$.

因此，需要至少取 **4** 位有效数字.

二、数值计算的误差限（相对误差限）估计

设近似值 x_1^* 与 x_2^* 的误差限分别为 $\varepsilon(x_1^*)$ 与 $\varepsilon(x_2^*)$，那么其四则运算满足：

(1) $\varepsilon(x_1^* \pm x_2^*) \le \varepsilon(x_1^*) + \varepsilon(x_2^*)$

(2) $\varepsilon(x_1^* x_2^*) \le |x_1^*|\varepsilon(x_2^*) + |x_2^*|\varepsilon(x_1^*)$

$$|x_1^* x_2^* - x_1 x_2| = |x_1^*(x_2^* - x_2) + x_2^*(x_1^* - x_1) - (x_1^* - x_1)(x_2^* - x_2)|$$

(3) $\varepsilon\left(\dfrac{x_1^*}{x_2^*}\right) \le \dfrac{|x_1^*|\varepsilon(x_2^*) + |x_2^*|\varepsilon(x_1^*)}{|x_2^*|^2}$， $x_2^* \ne 0$

2.1 一元函数误差估计

设 x^* 为 x 的近似值，那么

$$f(x)-f(x^*)=f'(x^*)(x-x^*)+\frac{f''(\xi)}{2}(x-x^*)^2, \qquad \xi \text{介于} x^*, x \text{之间}$$

则有

$$|f(x)-f(x^*)| \le |f'(x^*)|\varepsilon(x^*)+\frac{|f''(\xi)|}{2}\varepsilon^2(x^*)$$

若 $f''(x^*)$ 不是很大且 x^* 与 x 很接近，则

$$\varepsilon\big(f(x^*)\big) \approx |f'(x^*)|\varepsilon(x^*)$$

2.2 多元函数误差估计

设 $x^* = (x_1^*, x_2^*, \cdots, x_n^*)$ 为 x 的近似值，则多元可微函数 $f(x)$ 的误差限为：

$$\varepsilon\big(f(x^*)\big) \approx \sum_{k=1}^{n} \left|\frac{\partial f(x^*)}{\partial x_k}\right| \varepsilon(x_k^*)$$

其相对误差限为

$$\varepsilon_r\big(f(x^*)\big) \approx \sum_{k=1}^{n} \left|\frac{\partial f(x^*)}{\partial x_k}\right| \frac{\varepsilon(x_k^*)}{|f(x^*)|}$$

例8 测得某场地的长 x 和宽 y 分别为： $x^* = 110\ \mathrm{m}$，$y^* = 80\ \mathrm{m}$，它们的测量误差限分别为 $\varepsilon(x^*) = 0.2\ \mathrm{m}$ 和 $\varepsilon(y^*) = 0.1\ \mathrm{m}$. 试求面积 S 的误差限和相对误差限.

解： 由题可知

$$\varepsilon(S^*) \approx \left|\frac{\partial S(x^*, y^*)}{\partial x}\right| \varepsilon(x^*) + \left|\frac{\partial S(x^*, y^*)}{\partial y}\right| \varepsilon(y^*)$$

$$= |y^*|\varepsilon(x^*) + |x^*|\varepsilon(y^*) = (80\times0.2 + 110\times0.1)\ \mathrm{m}^2 = 27\ \mathrm{m}^2$$

$$\varepsilon_r(S^*) = \frac{\varepsilon(S^*)}{|S^*|} \approx \frac{27}{110\times80} \approx 0.31\%\,.$$

三、误差定性分析

1、误差会传播、积累、对消

2、误差分析的核心：

研究原始数据的误差和计算中产生的误差对最终计算结果的影响

注记：算法有“优劣”之分，即使不能定量地估计最终误差，但能判别计算过程中误差不会被任意放大，从而能实施数值计算，这是误差分析的目的！

3.1 算法稳定性

$$\frac{1}{7(n+1)} = \int_0^1 \frac{x^n}{7} dx \le I_n \le \int_0^1 \frac{x^n}{6} dx = \frac{1}{6(n+1)}$$

例9 近似计算

$$I_n = \int_0^1 \frac{x^n}{x+6} dx \quad (n = 1,2,\cdots)$$

解： 由于

$$I_n + 6I_{n-1} = \int_0^1 \frac{x^n + 6x^{n-1}}{x+6} dx = \int_0^1 x^{n-1}\, dx = \frac{1}{n},$$

因此，有递推式：$I_n = \frac{1}{n} - 6I_{n-1}$，由于 $I_0 = \ln 7 - \ln 6 \approx 0.154$，则

$I_1^* = 0.0751,\quad I_2^* = 0.0494,\quad I_3^* = 0.0369,\quad I_4^* = 0.0286,\quad I_5^* = 0.0284,$

$I_6^* = -0.00371,\quad I_7^* = 0.165,\quad I_8^* = -0.866,\quad I_9^* = 5.305,\quad I_{10}^* = -31.7,$

考察第 n 步误差：

$$I_n^* - I_n \approx \left(\frac{1}{n} - 6I_{n-1}^*\right) - \left(\frac{1}{n} - 6I_{n-1}\right) = -6(I_{n-1}^* - I_{n-1})$$

所以

$$|I_n^* - I_n| \approx 6|I_{n-1}^* - I_{n-1}| \approx 6^2|I_{n-2}^* - I_{n-2}| \approx \cdots \approx 6^n|I_0^* - I_0|$$

误差以6倍速度增长！

该算法的误差无法有效控制，需更换算法计算！

思考：如何编程计算？

算法二：$I_{n-1} = \frac{1}{6n} - \frac{1}{6} I_n$，先预估 $I_{10}^* = \frac{1}{2}\left(\frac{1}{6(10+1)} + \frac{1}{7(10+1)}\right) \approx 0.0141$

然后由递推式可得

$$I_9^* = 0.0143, \quad I_8^* = 0.0161, \quad I_7^* = 0.0181, \quad I_6^* = 0.0208, \quad I_5^* = 0.0243$$

$$I_4^* = 0.0293, \quad I_3^* = 0.0368, \quad I_2^* = 0.0494, \quad I_1^* = 0.0751$$

可见此时算法**误差可控**，误差没有递增. 思考：分析此算法误差

定义：在计算过程中，若误差不增长或能有效控制，则称该算法是**稳定的**，否则为**不稳定的**。

（既然误差不可避免，那么算法的稳定性十分重要！）

Ch1_ex9.m

```
clc,clear; % 清屏，清缓存

%% 初始化
I = zeros(10,1); % 初始化I(n)为十行一列的0向量
I(10) = 1/2*(1/66+1/77); % I(10)赋值

%% 迭代过程
fprintf('n \t I(n)\n'); % 屏幕输出: n   I(n)
for n = 9:-1:1

    I(n) = 1/6/(n+1) - 1/6*I(n+1); % 迭代计算 I(n)

    fprintf('%d\t%g\n',n,I(n)); % 输出当前迭代次数和迭代值

end
```

```
n          I(n)
9          0.0143218
8          0.0161316
7          0.0181447
6          0.0207854
5          0.0243135
4          0.0292811
3          0.0367865
2          0.0494245
1          0.0750959
```

3.2 问题的性态

定义： 如果输入数据的微小扰动（误差）会引起计算结果的很大变化，

则称该问题是病态的，否则就是良态的。

例10 求解线性方程组 $\begin{cases} x+ay=1 \\ ax+y=0 \end{cases}$

解： 当 $a=1$时，无解. 当 $a\neq 1$时，解为 $x=\frac{1}{1-a^2}, y=-\frac{a}{1-a^2}$.

当 $a\approx 0.9990$时，$x\approx 500.25$，

若扰动 $\tilde{a}\approx 0.9991$，$\tilde{x}\approx 555.81=500.25+\mathbf{55.56}$

故当 $a\approx 1$时，a 的微小扰动可能会引起解的很大变化（病态）

思考：若 a 远离1呢？

3.3 一元函数的条件数

若一元函数 $f(x)$ 可微，则

$$\frac{\varepsilon_r\left(f(x^*)\right)}{\varepsilon_r(x^*)} = \left|\frac{f(x)-f(x^*)}{f(x)}\right| \div \left|\frac{x-x^*}{x}\right| \approx \left|\frac{xf'(x)}{f(x)}\right| \triangleq C_p,$$

称 C_p 为 $f(x)$ 的**条件数**，它是判断 $f(x)$ **是否病态的重要依据**.

C_p **越大**，将引起函数值的**相对误差越大**.

如： $f(x)=x^n$， $C_p=\left|\frac{xnx^{n-1}}{x^n}\right|=n$，若 $n=10$，取 $x=1$，$x^*=1.02$，

则 $f(1)=1$，$f(1.02)\approx 1.24$，**自变量**相对误差：2%，**函数值**相对误差：24%

病态！

注记：1. 病态与否是问题本身固有的性质，与数值算法无关！

2. 一般 $C_p \geq 10$ 就是病态问题， C_p 越大病态越严重！

3. 对于病态问题，算法选择尤为重要！

四、避免误差危害的若干原则

1、避免两个相近的数相减

2、避免绝对值很小的数做分母

3、避免大数“吃掉”小数

4、减少误差积累

4.1 避免两个相近的数相减

例11 求 $x^2-16x+1=0$ 的最小根.

解：方法一：$x=8-\sqrt{63}\approx 8-7.94=0.06$（**1**位有效数字）

方法二：$x=8-\sqrt{63}=\frac{1}{8+\sqrt{63}}\approx\frac{1}{15.94}\approx 0.0627$（**3**位有效数字）

例12 计算 $A=10^7(1-\cos 2^{\circ})$.

解：方法一：$A\approx 10^7(1-0.9994)=6\times 10^3$（**1**位有效数字）

方法二：$A=10^7\cdot 2\cdot(\sin 1^{\circ})^2\approx 6.13\times 10^3$（**3**位有效数字）

注记：相近两数相减会损失有效数字！

常用避免技巧

1. $\sqrt{x+\varepsilon}-\sqrt{x}=\dfrac{\varepsilon}{\sqrt{x+\varepsilon}+\sqrt{x}}$

2. $1-\cos x=2\sin^2\dfrac{x}{2}$ （当 x 靠近 0时）

3. $f(x)-f(x^*)=f'(x^*)(x-x^*)+\dfrac{f''(x^*)}{2!}(x-x^*)^2+\cdots$ （当 $f(x)\approx f(x^*)$时）

注记：若无法改变算式，则计算机上采用双倍字长运算！

4.2 避免绝对值很小的数做分母

1、可能出现溢出情形 (超出计算机所能表示的范围)

2、可能放大误差

3、避免技巧

$$\frac{x}{y} = e^{\ln x - \ln y}$$ （当$|y|$很小时）

4.3 避免大数“吃掉”小数

例13 计算 $\dfrac{10^9+10^{-9}-10^9}{10^{-9}}$.

更改：$\dfrac{10^9-10^9+10^{-9}}{10^{-9}}$

答案：1　　输出：0

例14 计算 $10^8+1+2+3+\cdots+10$.

更改：$1+2+3+\cdots+10+10^8$

答案：100000055　　输出：10^8

注记：先同数量级相加，再从小到大相加！

4.4 减少误差积累

例15 计算 $\ln 2$.

解： 方法一：利用 $\ln(1+x)=x-\frac{1}{2}x^2+\frac{1}{3}x^3-\frac{1}{4}x^4+\cdots+\frac{(-1)^{n+1}}{n}x^n+\cdots$

将 $x=1$ 代入，并分别取前 n 项求和，可得

n	**1**	**5**	**10**	**20**	**50**	**100**
S_n	1	0.783	0.646	0.669	0.683	0.688
$\lvert S_n-\ln 2\rvert$	3.1e-1	9.0e-2	4.8e-2	2.4e-2	9.9e-3	5.0e-3

思考：如何编程计算？

Ch1_ex15.m

```matlab
clc,clear % 清屏，清缓存

%% 初始化
S = zeros(100,1);    % 初始化前n项和S
Err = zeros(100,1); % 初始化前n项和的误差
S(1) = 1;
x = 1;

%% 迭代过程
for n = 2:100 % 注意从n=2开始，为什么？

    S(n) = S(n-1)+(-1)^(n+1)/n*x^n; % 迭代计算 S(n)

    Err(n)  = abs( S(n) - log(2) ); % 迭代计算 Err(n)=|S(n)-ln(2)|

end

%% 输出结果
fprintf('\n前100项和为：%g, 误差为：%g\n\n',S(100),Err(100));
```

```
前100项和为：0.688172, 误差为：0.004975
```

方法二：利用 $\ln\left(\frac{1+x}{1-x}\right) = 2\left(x + \frac{1}{3}x^3 + \cdots + \frac{1}{2n-1}x^{2n-1} + \cdots\right)$

将 $x = \frac{1}{3}$ 代入，并分别取前 n 项求和，可得

n	**1**	**3**	**5**	**10**
S_n	0.667	0.693	0.693	0.693
$\lvert S_n - \ln 2 \rvert$	2.6e-2	1.4e-4	1.1e-6	1.0e-11

注记：选择合适算法，减少四则运算次数！

例16 已知 $p(x)=a_0+a_1x+a_2x^2+\cdots a_nx^n$，计算 $p(2)$.

解： 方法一：直接计算. 需要 $\frac{n(n+1)}{2}$ 次乘法和 n 次加法.

方法二：

思考：如何编程实现？

$$p(x)=a_0+x(a_1+x(a_2+x(a_3+x(\cdots x(a_{n-1}+a_nx)\cdots))))$$

仅需要 n 次乘法和 n 次加法. **运算量和误差累计均减少！**

（公元1208—1268）

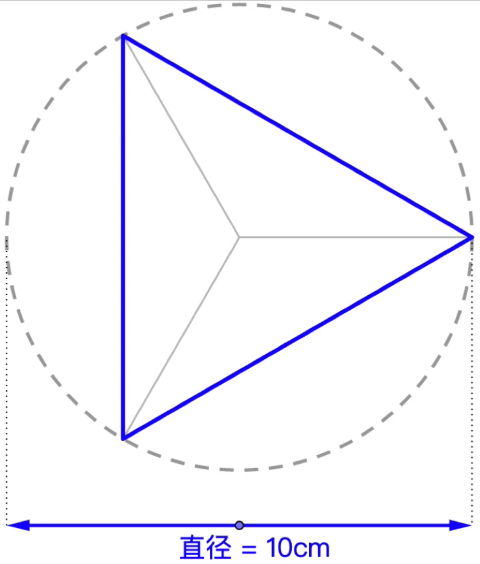
注记：选择合适算法，减少四则运算次数！

五、算法设计引入：以直代曲与化整为零

计算圆周率 π

“割之弥细，所失弥少，
割之又割，以至于不可割，
则与圆周合体而无所失矣”

——刘徽（魏晋）

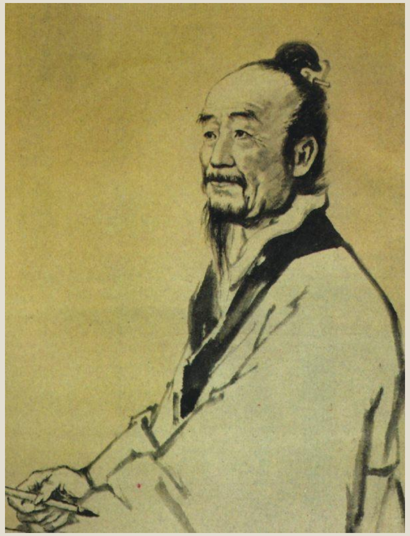


正 3 边形

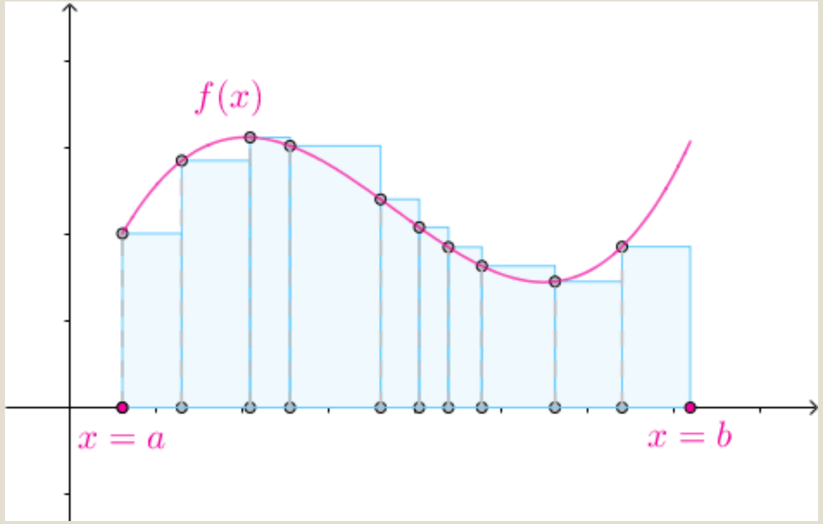
周长 $= 8.66025 \times 3 = 25.98076\text{cm}$

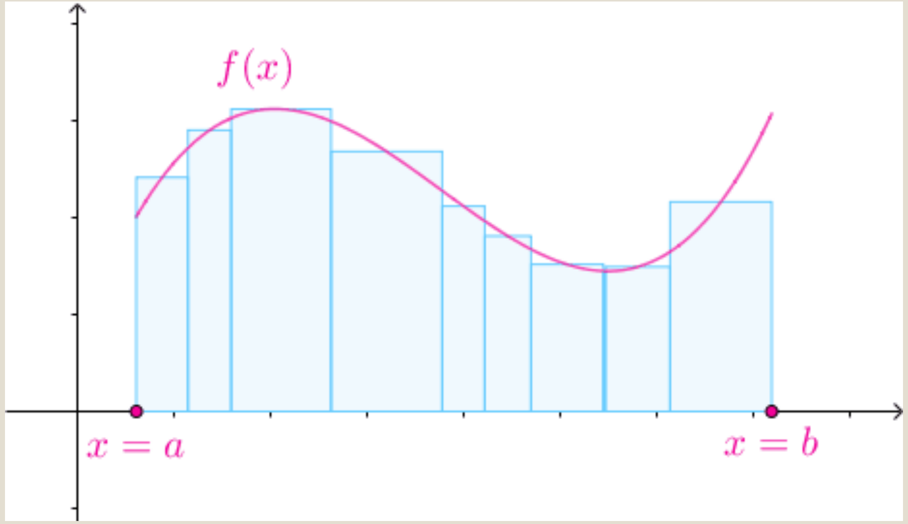
圆面积(78.53982) $\approx 12.99038 \times 2.5 = 32.47595\ \text{cm}^2$

$$\pi \approx \frac{\text{周长}}{\text{直径}} = \frac{25.98076\ \text{cm}}{10\ \text{cm}} = 2.59808$$



曲边梯形面积和 $\Longrightarrow I = \int_a^b f(x)\,dx$





思考：如何编程实现？

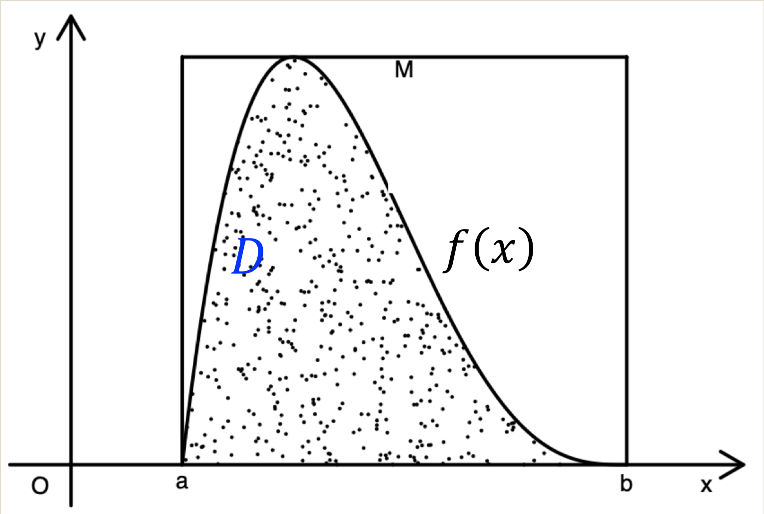
统计的方法计算

（1）计算 $I=\int_a^b f(x)\,dx$，$x\in[a,b]$，$f(x)\in[0,M]$

（2）在 $G=[a,b]\times[0,M]$ 上**均匀**投点 $Z_i(i=1,2,\cdots,N)$，

用 N 个点中**落入 D 的百分比 $\hat{p}$** 估计 p，从而

$$I\approx\hat{I}=\hat{p}\,S(G)$$



小结

避免误差危害的原则

误差

绝对误差（限）

相对误差（限）

有效数字

算法稳定性

问题病态性

条件数

习题与资料见Canvas课程网页

1、参考书目

[1]《第三种科学方法：计算机时代的科学计算》，石忠慈著，清华大学出版社，2000

[2]《数值分析》，张平文等著，北京大学出版社，2007

[3]《数值分析》，冯烟利等译，高等教育出版社，2005

2、网络学习资源

(1) https://www.icourse163.org/course/NEU-1002089009?from=searchPage《数值分析》精品课程

(2) https://uk.mathworks.com/help/pdf_doc/matlab/getstart.pdf（MATLAB入门）

(3) https://www.icourse163.org/course/CSU-1002475002?from=searchPage《科学计算与MATLAB语言》精品课程

3、拓展性参考文献

[1] 陈志明，科学计算：科技创新的第三种方法[J]，科学发展，2012(27): 161-166.