浮点运算与数值运算的误差

北京师范大学物理系　彭芳麟

浮点运算的误差

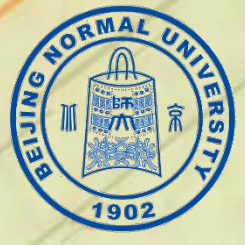
为什么有

$$0.1+0.1+0.1+0.1+0.1+0.1+\ 0.1+0.1+0.1+0.1+0.1+0.1$$
$$=9.99999999999999e-001$$

分析：　MATLAB使用浮点数，而浮点数中没有0.1，只能用二进制中最接近0.1的数来近似表示. 结果必然有误差!

- 数值计算不是使用全部实数，也没有**极限与无限**等实数概念.
- 使用浮点算术体系(有限精度的有限数集合)，计算中会产生：
  - 舍入(roundoff)
  - 下溢出(underflow)
  - 上溢出 (overflow)
  - 机器最小精度(eps)
  - 非数(NaN)

	二进制	十进制
eps	2^{-52}	2.2204e-16
realmin	2^{-1022}	2.2251e-308
realmax	$(2-eps)2^{1023}$	1.7977e+308



双精度浮点数表示形式与存储方式：

浮点数用
64位（bit）=1字节(word)

$$\pm(1+f)\cdot 2^{e}$$

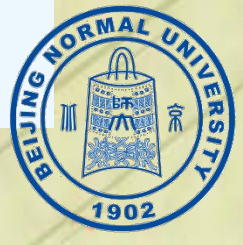
11bit存e

正负号用1bit

52bit存f

浮点数的整个小数部分不是f,而是1+f它占53位.然而首位的1并不需要存储，这样IEEE格式便将65位的信息打包成64位的一个字节（word）.

浮点数表示中整数部份由2^e（二进制）表示，e的取值限制整数的范围.
其中2^{10}=1024，用去10位(bit)，再用1位(bit)表示指数，共用11位.
e的实际取值为-(2^{10}-2) <= e <=(2^{10}-1)；
即 -1022 <= e <=1023, 得最小数2^{-1022}，最大数2^{1023}
e的正负值取值范围不同，正值为1到+1023，所以不必存储e正负号.
e=1024, f=0 表示无穷大（Inf），e=1024, $f\neq 0$ 表示非数（NaN）.
e=-1024时表示最小的规范数，　　e=-1023时表示最小的非规范数.



浮点数的表示法 $x = \pm(1+f)\cdot 2^e$

f 的取值限制数的精度.

$0 \leqslant f \leqslant 1;$ $2^e \leqslant 2^e + f\cdot 2^e < 2^{e+1}$

它表示利用 f 在$[2^e, 2^{e+1}]$之间等间隔插入的2^{52}个数，相邻小数的间隔为2^{e-52}，

全部小数的取值为 $f\cdot 2^e = n\cdot 2^{-52}\cdot 2^e$， $n=(0, 1, 2, 3\cdots, 2^{52}-1)$

例如当整数值为

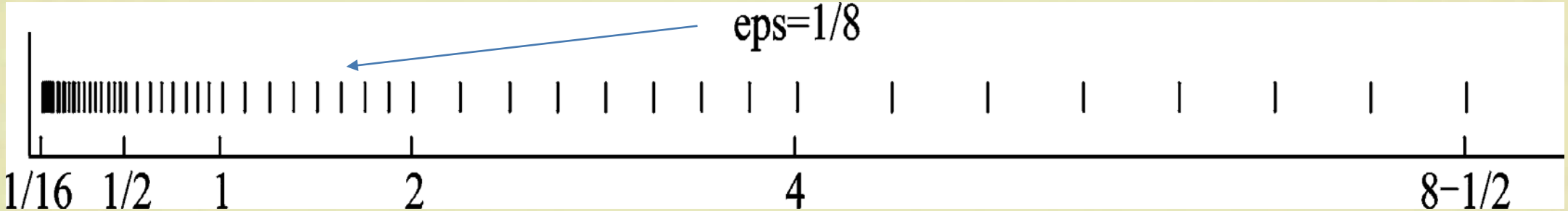
二进制2^e	2^{-1022}		2^{-2}	2^{-1}	2^0	2^1	2^2	2^3		2^{1023}
十进制数	最小实数	...	1/4	1/2	1	2	4	8	...	最大实数

小数值为（以1与2之间插入的小数为例）

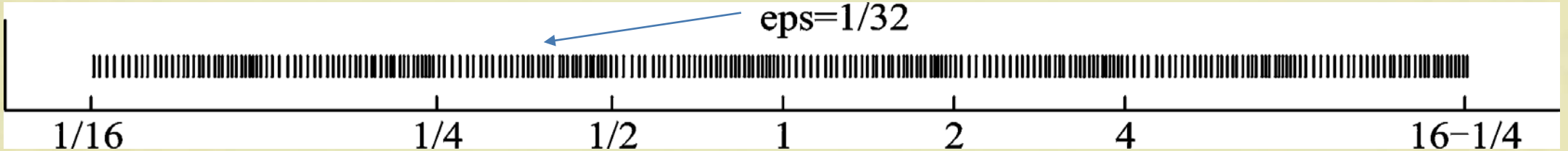
$2^0(1+f)$	$2^0=(1+0)$	$2^0(1+2^{-52})$	$2^0(1+2\cdot 2^{-52})$	...	$2^0(1+2^{51}\cdot 2^{-52})$	$2^1(1+0)=2$
十进制数	1	$1+2^{-52} = 1\frac{1}{4503599627370496}$	$1+2\cdot 2^{-52}$	...	$1+(2^{52}-1)*2^{-52}$	2

例 订箝$x=\pm(1+t)2^{e}$ ，t变$[0,1,2,3,4,5,6,7\,]*2^{-3}$,

$e_{\min}=-4,\quad e_{\max}=3,$



在对数显示下浮点数是等间隔的



在每个二进制区间，$2^{e}\le x\le 2^{e+1}$, 数按间隔2^{e-t} 等距排列,

$$\text{eps}=2^{-52}\approx 2.2204\bullet 10^{-16}$$

是两个浮点数的最大相对间距，eps/2是计算结果的最大相对误差.

要得到t=0.1就必须进行舍入，因为用二进制表示十进制的分数1/10需要一个无穷级数，所以存储于t的数值并不精确地等于0.1.事实上，

$$\frac{1}{10}=\frac{1}{2^4}+\frac{1}{2^5}+\frac{0}{2^6}+\frac{0}{2^7}+\frac{1}{2^8}+\frac{1}{2^9}+\frac{0}{2^{10}}+\frac{0}{2^{11}}+\frac{1}{2^{12}}+\cdots$$

在第一项之后，后续项的系数按1，0，0，1重复出现，根据这个规律以4项为一组进行合并后，可得到一个基为16，或十六进制的序列.

$$\frac{1}{10}=2^{-4}\left(1+\frac{9}{16}+\frac{9}{16^2}+\frac{9}{16^3}+\frac{9}{16^4}+\cdots\right)$$

需要在二进制表达式的第52项或十六进制表达式的第13项截断这个无穷级数的小数部分，然后进行向上或向下舍入，才能得到1/10的浮点数近似值.因此 $t_1<\frac{1}{10}<t_2$

其中

$$t_1=2^{-4}\left(1+\frac{9}{16}+\frac{9}{16^2}+\frac{9}{16^3}+\cdots+\frac{9}{16^{12}}+\frac{9}{16^{13}}\right)$$

$$t_2=2^{-4}\left(1+\frac{9}{16}+\frac{9}{16^2}+\frac{9}{16^3}+\cdots+\frac{9}{16^{12}}+\frac{10}{16^{13}}\right)$$

下面使用符号计算证明1/10更接近于t_2

浮点数0.1 更接近于t2的证明(用符号计算)

- >> syms k,a=symsum(1/16^k,1,13)
- a = 300239975158033/450359627370496
- >> t1=2^(-4)*(1+a*9)
- t1 = 7205759403792793/7205759403792936
- >> t2=t1+2^(-4)*1/16^(13)
- t2 = 3602879701896397/3602879701896396
- >> eval((1/10-t1)>(t2-1/10))
- ans = 1

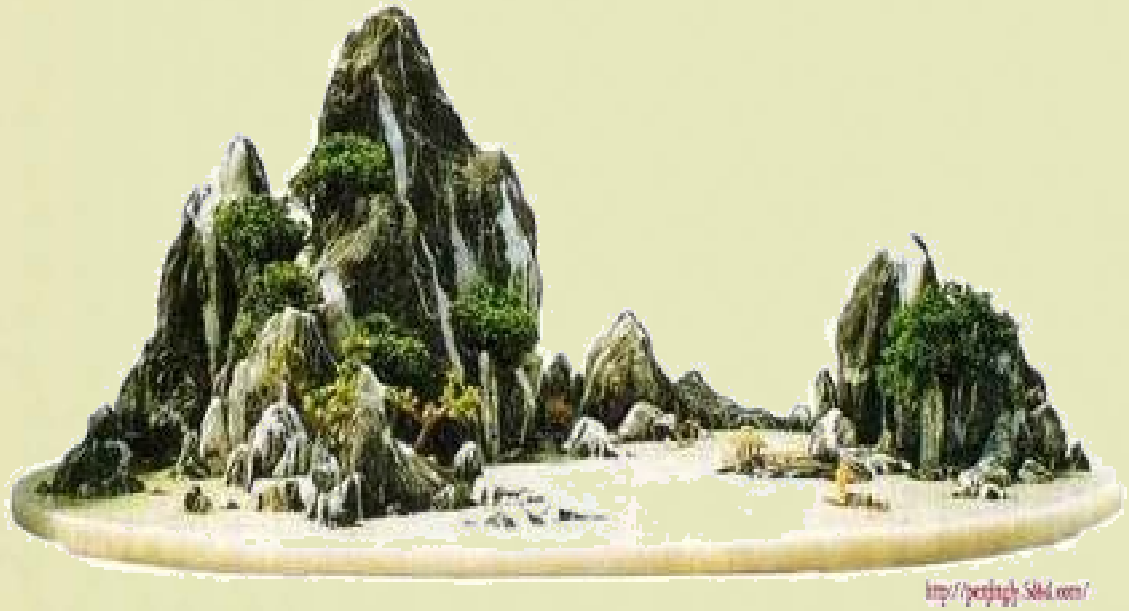
误差的起源

1. •模型误差
   忽略次要因素建立的模型与实际有偏差.
2. •观测误差
   模型中使用的观测参数有误差.
3. •计算方法带来的误差
   数值计算方法是近似方法，会产生误差.
4. •计算过程中的舍入误差
   如浮点数计算中所产生的误差.

思考题

双精度浮点数存储方式是什么?

谢谢！



浮点数的表示法　　$x = \pm(1 + f) \cdot 2^e$

其中畝嗽鄙伢2^e + 互迟劫 - 寿庚　$-1022 \le e \le 102$，　e的取值限制整数的范围

小数部分取决于f，表示在$[2^e,\ 2^{e+1}]$之间插入的小数，其取值限制数的精度.

$0 \le f \le 1$;　　$f=[0,\ 2^n]$,　　$n=[-52,\ -51\ldots,\ -2,\ -1]$,

二进制2^e	2^{-1022}		2^{-2}	2^{-1}	2^0	2^1	2^2	2^3		2^{1023}
十进制数	最小实数	…	1/4	1/2	1	2	4	8	…	最大实数

两个数之间插入的数计算:　　$0 <= f <= 1;\ 1 <= (1+f) <= 2;\ \ 2^e <= (1+f)\ 2^e <= 2^{e+1}$

例如要计算1与2之间插入的数值

$2^0\ (1+f)$	$2^0=(1+0)$	$2^0(1+2^{-52})$	$2^0(1+22^{-52})$	…	$2^0(1+2^{-2})$	$2^0(1+2^{-1})$	$2^1(1+0)=2$
十进制数	1	$1+2^{-52} = 1+\frac{1}{4503599627370496}$	$1+2/2^{-52}$	…	$1+3/2^{-52}$	$1+2/2^{-52}$	2