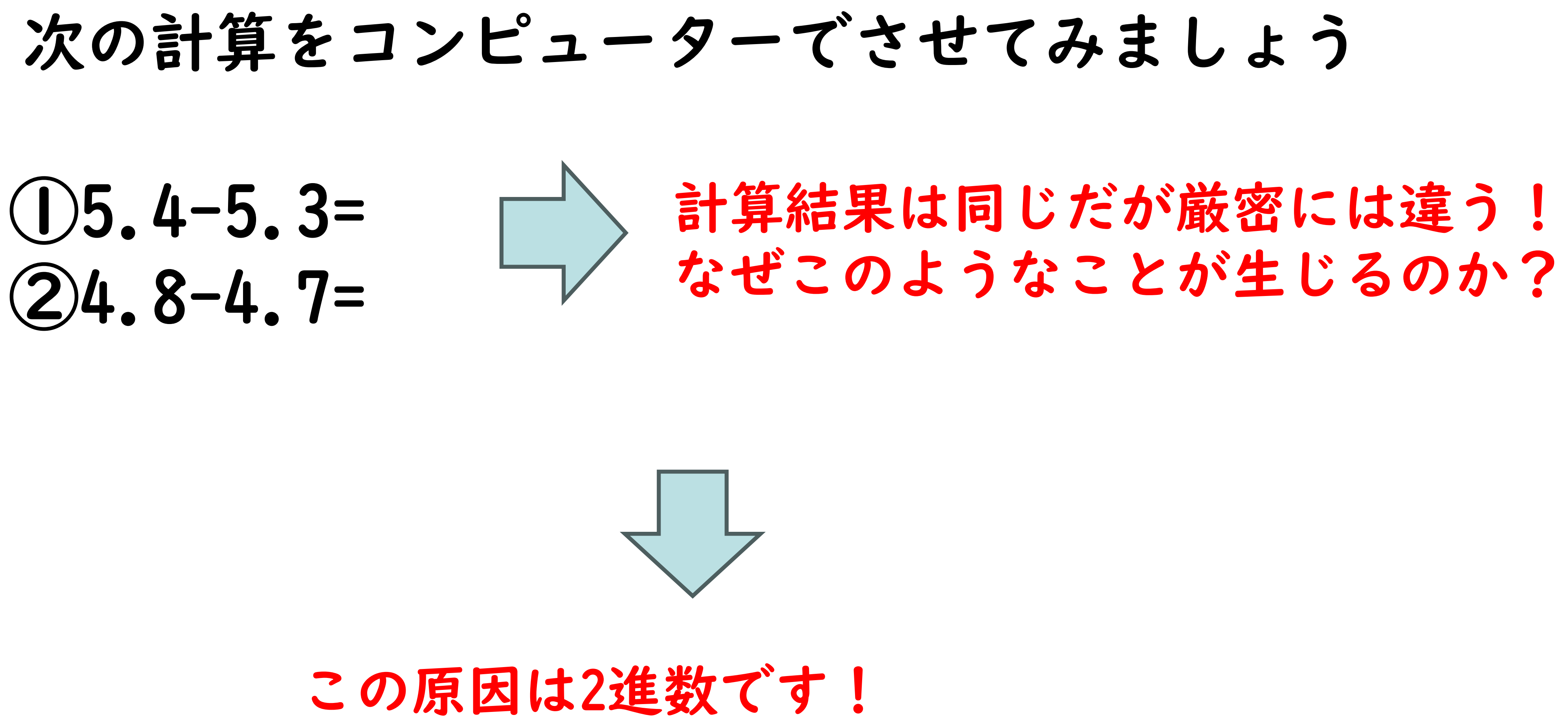
誤差と浮動小数点について

（解説）


次の計算をコンピューターでさせてみましょう
①5.4−5.3=
②4.8−4.7=
計算結果は同じだが厳密には違う！
なぜこのようなことが生じるのか？
この原因は2進数です！

2進数小数を10進法に変換

①11.101

				2^{-1}	2^{-2}	2^{-3}
				$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$
桁の重み	2^1	2^0		(0.5)	(0.25)	(0.125)
	×	×		×	×	×
	1	1	.	1	0	1
	⇩				⇩	
	2 +	1	.	0.5 +	0 +	0.125

答え　3＋0.625=3.625

2進数小数を10進法に変換

②$1\ 0\ 1\ .\ 0\ 1_{(2)}$

$2^2 \quad 2^1 \quad 2^0 \quad 0.5 \quad 0.25$

$2^2 \times 1 + 2^1 \times 0 + 2^0 \times 1 + 0.5 \times 0 + 0.25 \times 1$

$= 4 + 0 + 1 + 0 + 0.25$

$= 5.25$

2進数小数を10進法に変換

③ $1\ 0\ 1\ 1.\ 1\ 0\ 1_{(2)}$

$2^3 \quad 2^2 \quad 2^1 \quad 2^0 \quad 0.5 \quad 0.25 \quad 0.125$

$2^3\times 1+2^2\times 0+2^1\times 1+2^0\times 1 + 0.5\times 1+0.25\times 0+0.125\times 1$

$= 8 + 0 + 2 + 1 + 0.5 + 0 + 0.125$

$=11.625$

10進数小数を2進法に変換

①０．７５

↓

小数ではない場合はひたすら２で割りましたよね！

小数の場合は(0.0) 1.0になるまでひたすら２をかけます

10進数小数を2進法に変換

① 0.75

$$
\begin{array}{rl}
0.75 & \cdots\ 0 \\
\times\ \ 2 & \\
\hline
0\ \cancel{1}.5 & \cdots\ 1 \\
\times\ \ 2 & \\
\hline
0\ \cancel{1}.0 & \cdots\ 1
\end{array}
$$

=0.11

10進数小数を2進法に変換

① 0．７５

	$\frac{1}{2}$ (0.5)	$\frac{1}{4}$ (0.25)	$\frac{1}{16}$ (0.125)	$\frac{1}{32}$ (0.03125)
⇨ 0.	1	1		

10進数小数を2進法に変換

② 0．5

$$\begin{array}{r} 0.5 \\ \times \quad 2 \\ \hline 0\cancel{1}.0 \end{array}$$

・・・ 0

・・・ 1

＝0.1

10進数小数を2進法に変換

③ 0．１２５

$$
\begin{array}{r l}
0.125 & \cdots 0 \\
\times \quad 2 & \\
\hline
0.25 & \cdots 0 \\
\times \quad 2 & \\
\hline
0.5 & \cdots 0 \\
\times \quad 2 & \\
\hline
0\ \cancel{1}.0 & \cdots 1
\end{array}
$$

＝0.００１

なぜ同じ数字なのに異なる結果がでたか？

				A
5.4	-	5.3	=	0.1
				B
4.8	-	4.7	=	0.1
		AとBは		異なる

10進数の0.1では計算ができる。
しかし2進数の0.1は無限小数となり計算できない場合がある（コンピュータは2進数で計算）

10進数小数を2進法に変換

① 2.75

①左の整数部分は2を2進数に直す

＝10

②右の少数部分は0.75を2進数に直す

＝0.11

③2つを合わせると　＝10.11

10進数小数を2進法に変換

① 4.25

⇨ ＝100.01

誤差について

数学の正解でも1÷3=0.33333・・・
と計算が続くものがありますが、
コンピュータでも無限に計算が続いたときにビット数に
限りがあるためにどこかで打ち切らないといけません！
そのために誤差が生じます

誤差①

- ８ビットで表現できるのは0~255にあたる256通り$2^8=256$

0～255は<u>$0～2^8-1$</u>とも表現する

- ２の補数表現（符号ビット）をつかうと−も表現できるので-128~127の256通りになる。

-128～127は<u>$-2^7～2^7-1$</u>とも表現する

誤差①

・２の補数表現（符号ビット）をつかうと－も表現できるので-128~127の256通りになる。

⇩

-128～127は-2^7～2^7-1とも表現する

⇩

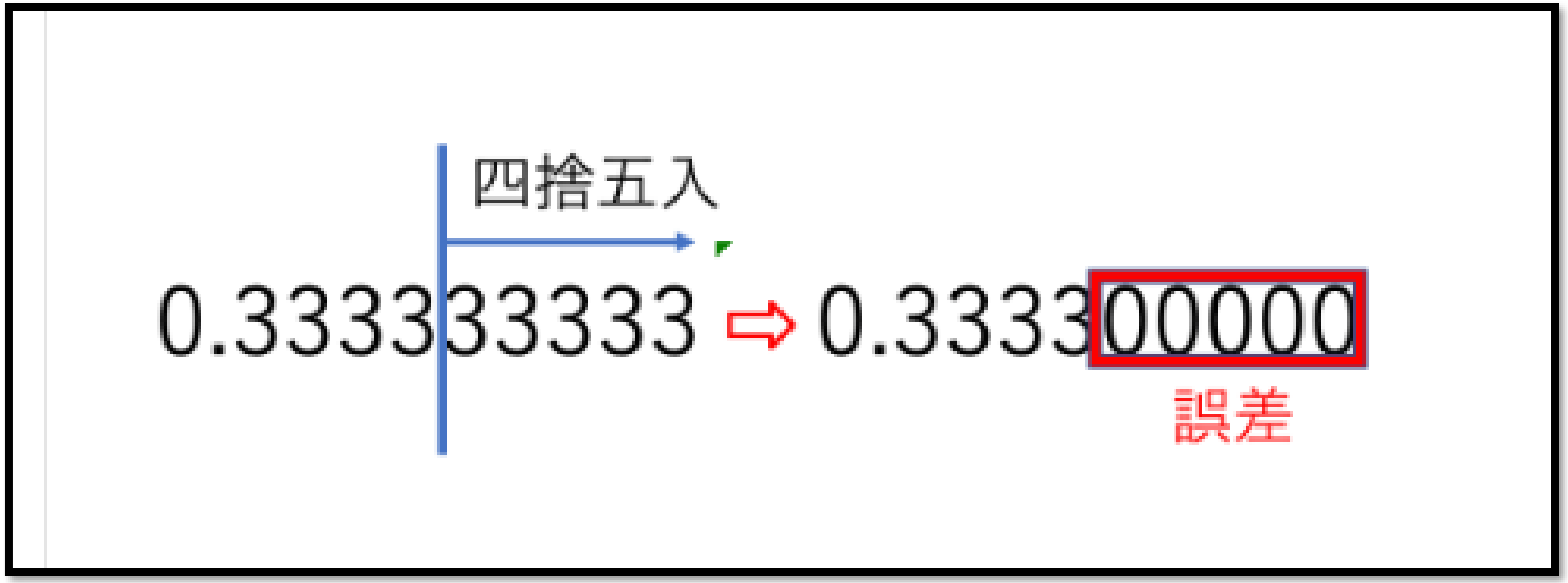
正の整数で表現できるのは127までなので$2^7=128$を表現しようとすると表示できる範囲を超える

⇨　このことを（① オーバーフロー　　）という

誤差②

（②　丸め誤差　　　　　　）・・・

切り捨て・切り上げ・四捨五入することで発生する誤差



誤差②

（③ **打ち切り誤差**　）・・・

計算結果を打ち切ることで発生する誤差

誤差③

（④　桁落ち誤差　　　）・・・

有効桁数が減少することで発生する誤差
（有効数字の桁数が変わってしまう）

例）10.0001-10.0000=0.001になるがもし小数点以下が2桁しか扱えない計算機だと10.00-10.00になり答えが0になる。そのため本当は0.001の答えが0と出てしまう。

b　小数点を含む数の内部表現

有効数字の部分を仮数という

例）10進法の　12.34　⇨　①$1.234 \times 10^1$

有効数字の部分　　10の何乗かの部分

例）２進法の　$101.11_{(2)}$　⇨　②$1.0111 \times 2^2{}_{(2)}$

仮数は、整数部分を１桁とし、
そこに０以外の一番上の位の数をおく

このような数は、小数点の位置を移動させて表現するため、

浮動小数点数　という

図12　浮動小数点数

次の10進数で表現された数を浮動小数点で表せ

①11.23

答え 1.123×10^1

②123.245

答え 1.23245×10^2

③-240.234

答え -2.40234×10^2

④123

答え 1.23×10^2

次の2進数で表現された数を浮動小数点で表せ

①111.11

答え 1.1111×2^2

②10.11

答え 1.011×2^1

③-101.01

答え -1.0101×2^2

④1001

答え 1.001×2^3

C　浮動小数点数のデータの形式

浮動小数点数を表すデータは・・・

ビットの列を３つの部分に分けて表現する

- 正か負かを表す部分（**符号ビット**）
- 「2 の何乗か」を表す部分（**指数部**）
- 仮数を表す部分（**仮数部**）

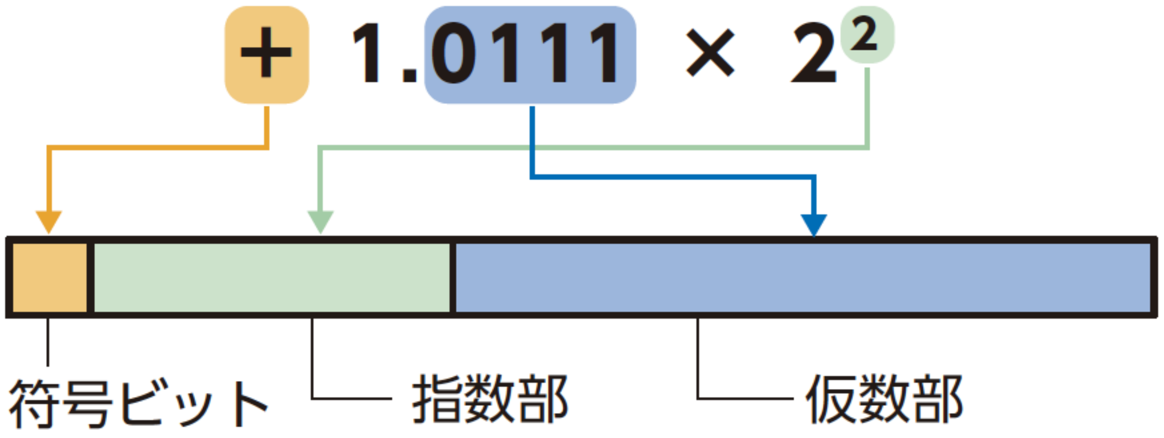


図14　浮動小数点数とビット列