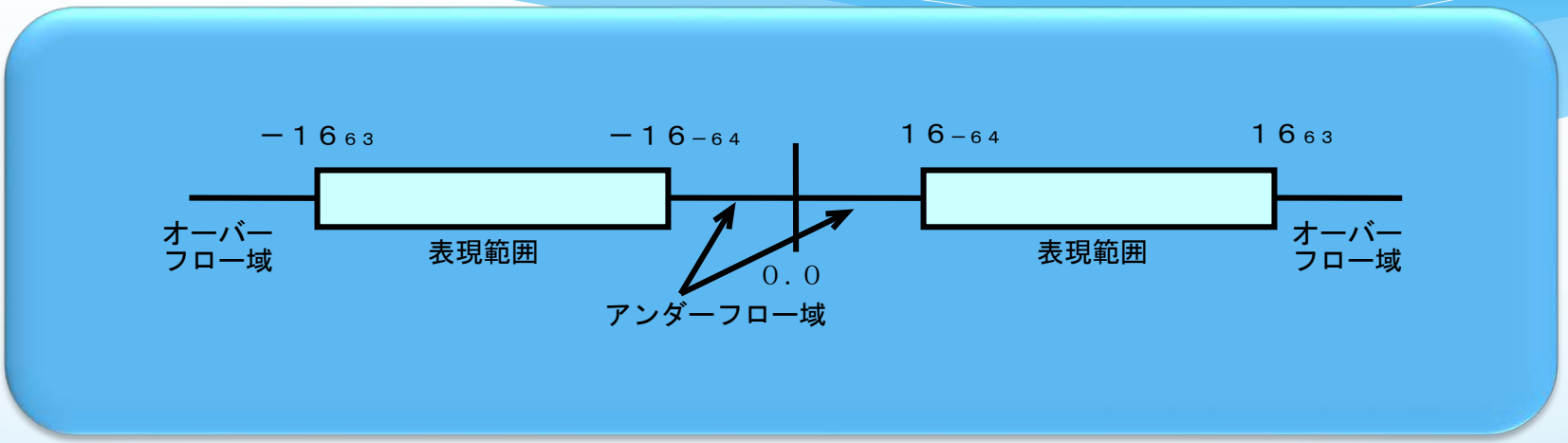
計算機の誤差

基数16の表示範囲



① 0の近傍の値、-16^{-64}～0の範囲、
0～16^{-64}の範囲は表現不能となる。

② -16^{63}より小さい値、16^{63}より大きい値も
表現不能となる。

誤差の発生

① 浮動小数点数の表見範囲が限られているため、
指数調整や演算結果が表現不能範囲になると、
誤差が発生する。

② 表現が有限桁であるため、
桁外れが発生したり、有効桁数が減少すると、
誤差が発生する。

③　計算機の演算回数を
　　有限回に制約することによって誤差が発生する。

④　演算方法、演算順序によって誤差の大きさが異なる。

⑤　誤差の種類には、次のものがある。

❶　情報落ち
❷　桁落ち、桁あふれ
❸　丸め誤差

情報落ち

① 浮動小数点演算において、
次の場合に誤差が発生する。

❶ 絶対値の大きな値に
絶対値の小さな値を加算する場合
❷ 指数部を揃えて計算する場合
❸ 絶対値の小さな値が無視される場合

② 加算演算において、
次の場合に誤差が発生する。

❶ 両者の指数部を合わせるとき、
❷ 小さい方の仮数部が右に大きくシフトし、
有効ビット範囲の下位の桁が欠落する。
❸ この現象を情報落ちという。
❹ 情報落ちの誤差は、
計算を実行する前に発生する誤差である。

具体例

① 2つの2進数A、Bにおいて

A:01000101000100000001100000000000
B:00111111101000000000000000000000

② Bの2進数をAの2進数の指数に合わせる。

B:01000101000000000000000000000000

③ Bの仮数部の1010は
24桁の有効範囲から外れて、情報落ちとなる。

桁落ち(アンダーフロー)

① 桁落ちは、浮動小数点演算において、
次の場合に誤差が発生する。

❶ 非常に小さな指数表記の
浮動小数点数同士の乗算を行う。
❷ 指数が指数部で表現できる範囲を
下回ってしまうことである。

② ほぼ等しい値の浮動小数点同士を減算する。

❶ 結果が非常に小さい値になるため
表現できる範囲からもれる。
❷ 有効桁数が大幅に減ってしまうことである。

③ 浮動小数点表示では、
0近傍の値を表現することが不可能なため
この現象が発生する。

④ 演算結果、発生する誤差である。

指数表記の小さい数 × 指数表記の小さい数 ⇒ 桁落ちの発生

等しい値同士の減算 ⇒ 有効桁数の減少・桁落ちの発生

桁あふれ(オーバフロー)

① 桁あふれは、非常に大きい値同士を掛けると、
指数部で表現できる最大の範囲を
超えてしまう現象である。

② 2進数のビット数を増加させることで、
誤差の発生を少なくすることができるが、
誤差の発生を回避することはできない。

丸め誤差

① 丸め誤差の定義

❶ 入力データを有限桁の数値で使用したり、
❷ 数値計算の結果

四捨五入、切り上げ、切り捨てによって
意味ある有効桁に丸めるときに
発生する誤差である。

② 10進数の値を2進数に変換する場合、

❶ 2進数を限られたビット数の数値で表示するときに
誤差が発生する場合がある。

❷ 10進数の次の小数を、
2進数の小数に変換すると循環小数となり、
有限けた数で表現すると、丸め誤差が発生する。

0.1、0.2、0.3、0.4、0.6、0.7、0.8、0.9

具体例

① 0.2～0.9の10進数の小数を2進数に変換すると、
0.5を除いて、循環数になる。

② 次ページの具体例の下線の部分が循環する。

③ この循環する2進数を8ビットの2進数で表現すると、
8ビット以降の循環部分が切り捨てられるため、
丸め誤差が発生する。

$(0.2)_{10}=(0.00110011\underline{0011}\cdots)_2\fallingdotseq(0.00110011)_2$
$(0.3)_{10}=(0.01001100\underline{1001}\cdots)_2\fallingdotseq(0.01001100)_2$
$(0.4)_{10}=(0.01100110\underline{0110}\cdots)_2\fallingdotseq(0.01100110)_2$

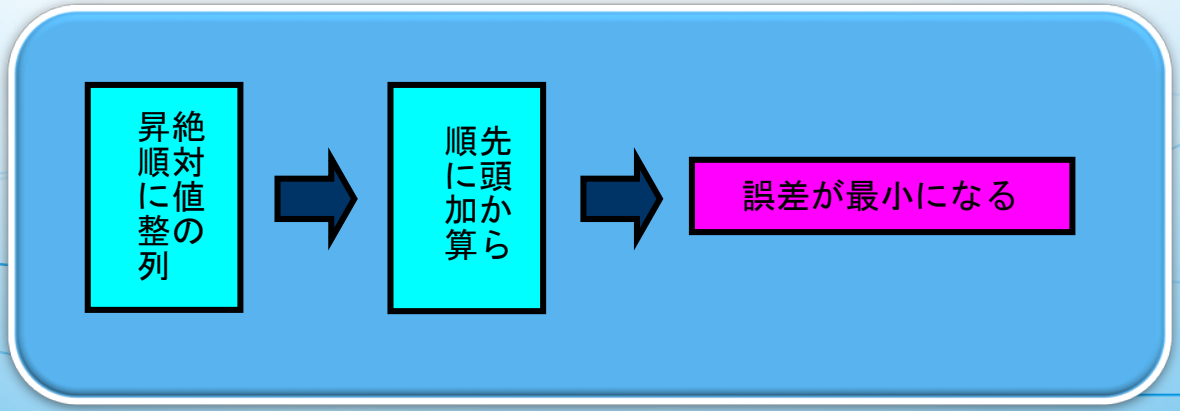
$(0.5)_{10}=(0.1)_2$

$(0.6)_{10}=(0.10011001\underline{1001}\cdots)_2\fallingdotseq(0.10011001)_2$
$(0.7)_{10}=(0.10110011\underline{0110}\cdots)_2\fallingdotseq(0.10110011)_2$
$(0.8)_{10}=(0.11001100\underline{1100}\cdots)_2\fallingdotseq(0.11001100)_2$
$(0.9)_{10}=(0.11100110\underline{1100}\cdots)_2\fallingdotseq(0.11100110)_2$

誤差を少なくする演算法

① 全てのデータを絶対値の昇順に並べ替え、
先頭から順に加える。

② 正数は加算し、負数は減算する計算法で
誤差の発生が最も小さくなる。



③　情報落ちによる誤差を少なくする方法

絶対値の差が小さい値同士の演算を行えば
指数調整による誤差を少なくすることができる。

④　複数の演算を連続して実行する場合の
誤差を少なくする方法

❶　加算結果の値と次に加算する値の差を小さくすると、
誤差が小さくなる。
❷　情報落ちが発生しないため誤差が小さくなる。