2020数値解析

訂正箇所について
40枚目のスライドに
訂正があります。

学習教育目標と科目との対応について

学習・教育目標(C):

数学、自然科学等の基礎的知識と情報工学に関する専門的な知識を有し、それらを情報社会における諸問題の探求・解決へ自主的・継続的に応用できる人材を育成する。

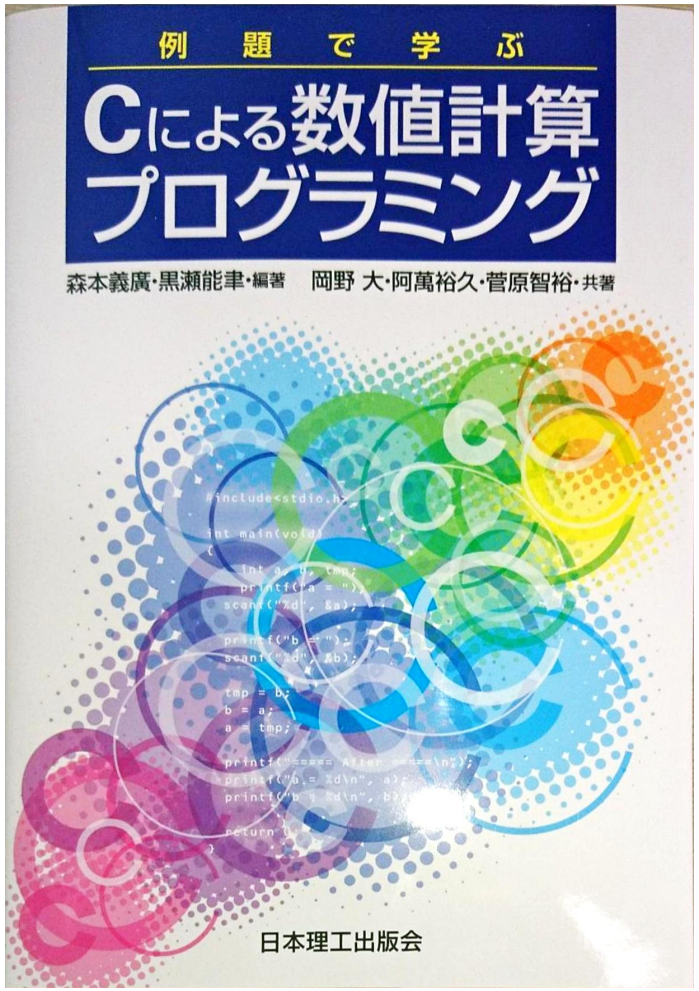
キーワード: 数値計算・記号計算の基礎的知識

1. 浮動小数点数の表現形式と, 浮動小数点数の演算に伴う様々な誤差の種類と性質を説明できる.
2. 典型的な数学的問題に対する代表的な数値解法について, その原理と性質を説明し, アルゴリズムを記述できる.
3. 特に, 連立1次方程式の数値解法について, 不良条件の場合を判定可能な消去法アルゴリズムを記述できる.

教科書

教科書：例題で学ぶCによる数値計算プログラミング
著 森本 義広 他 日本理工出版 2019

ISBN:978-4890195299



この授業のために作成した教科書です。

サンプルプログラムをCで記述しました。

複雑な記述避けてできるだけ分かり易いサンプルになるよう工夫しています。

参考書

参考書:数値解析
著 齋藤宣一 共立出版 2017　ISBN:978-4320111905

参考書:数値計算法の数理(第2版 2003)
著 杉原正顕, 室田一雄 岩波書店 1994
ISBN:978-4000055185

参考書:線形計算の数理
著 杉原正顕, 室田一雄 岩波書店 2009
ISBN:978-4000075565

参考書:数値計算の常識
著 伊理正夫, 藤野和健 共立出版 1985
ISBN:978-4320013438

参考書（昨年度までの教科書）

参考書：数値解析 技術者のための高等数学5
著 E.クライツィグ 培風館 2003 ISBN:978-4563011192

参考書：Advanced Engineering Mathematics
著 Erwin.Kreyszig, Wiley 2011 ISBN:978-0470458365

数値解析

第1回・第2回：浮動小数点数と誤差

数値**解析**とは？
数値計算と何が違うの？

（この授業では）

数値計算を用いた現象やシステムの解析

を「数値解析」と呼ぶことにしましょう

数値解析の例

天気予報

各種エンジン開発

　ロケットエンジンの燃焼制御、ジェットエンジンの流体解析、
　ガソリンエンジンの熱解析…

保険金融商品の開発、投資判断

　価格決定メカニズムの解析、変動予測、価値・リスク計算

システムの動向解析

　市場予測、消費者・有権者の行動予測

…

授業で扱う範囲

数値解析全般を扱うことはできません。
数値計算法の基礎として以下の話題について、その入門にあたる部分を扱います。

- 数値表現
- 関数近似
- 数値微積分
- 計算機による線形計算
- 微分方程式の数値解放

数値表現

数値計算＝**有限桁**・有限回演算による計算

現代の計算機は2進表現を内部表現としている
⇒実際に計算機が扱うのは有限桁の二進数

人の扱う十進数は基数変換して計算すれば良い。

本当に？

消費税込価格を1.08倍の計算で求めましょう。
→1.08の2進表現は？

数値表現

10進数の$1.08 \equiv 1.08_{10}$

$= 1+0\times10^{-1}+8\times10^{-2}$

$10^0 = 1$の位=1、10^{-1}の位=0、10^{-2}の位=8

これを2進数で表現すると

数値表現

1.08_{10} ＝1+0.08

0.08=0.08×2÷2＝0.08×2×2^{-1} なので

＝1+0.16×2^{-1}

これを繰り返して

＝1+0.32×2^{-2}=1+0.64×2^{-3}=1+1.28×2^{-4}

さらに

＝1+1×2^{-4}+0.28×2^{-4}＝1.0001_2+0.28×2^{-4}

として、小数点以下4桁の2進数と余りで表現できる

1.08_{10} ＝

... もっと続けると

$=1.0001_2+1.12\times2^{-6}=1.000101_2+0.12\times2^{-6}$
$=1.000101_2+1.92\times2^{-10}$
$=1.0001010001_2+1.84\times2^{-11}$

$=1.00010100011_2+1.68\times2^{-12}$
$=1.000101000111_2+1.36\times2^{-13}$
$=1.0001010001111_2+0.72\times2^{-14}$
$=1.00010100011110_2+1.44\times2^{-15}$
$=1.000101000111101_2+0.88\times2^{-16}$
$=1.00010100011110101_2+1.76\times2^{-17}$
$=1.000101000111101011_2+1.52\times2^{-18}$
$=1.000101000111101011_2+1.04\times2^{-19}$
$=1.0001010001111010111_2+0.08\times2^{-20}$

$$1.08_{10} = 1 + 0.08$$

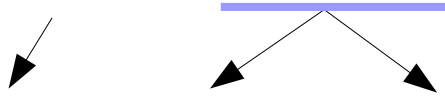
$$= 1.\overset{\cdots}{0}0010100011110101011_{2} + 0.08 \times 2^{-20}$$

$$
\begin{aligned}
1.08_{10} &= 1 + 0.08 \\
&= 1.000_2 + 1.28 \times 2^{-4} \\
&= 1.0001_2 + 1.12 \times 2^{-6} \\
&= 1.000101000_2 + 1.92 \times 2^{-10} \\
&= 1.0001010001_2 + 1.84 \times 2^{-11} \\
&= 1.00010100011_2 + 1.68 \times 2^{-12} \\
&= 1.000101000111_2 + 1.36 \times 2^{-13} \\
&= 1.0001010001111_2 + 1.44 \times 2^{-15} \\
&= 1.00010100011110_2 + 1.76 \times 2^{-17} \\
&= 1.00010100011110101_2 + 1.52 \times 2^{-18} \\
&= 1.00010100011110101011_2 + 1.04 \times 2^{-19} \\
&= 1.00010100011110101011_2 + 0.08 \times 2^{-20}
\end{aligned}
$$

$$
\begin{aligned}
&1.08_{10}\\
&=1+0.08\\
&=1.000_2+(1+0.28)\times 2^{-4}\\
&=1.0001_2+(1+0.12)\times 2^{-6}\\
&=1.000101000_2+(1+0.92)\times 2^{-10}\\
&=1.0001010001_2+(1+0.84)\times 2^{-11}\\
&=1.00010100011_2+(1+0.68)\times 2^{-12}\\
&=1.000101000111_2+(1+0.36)\times 2^{-13}\\
&=1.0001010001111_2+(1+0.44)\times 2^{-15}\\
&=1.00010100011110_2+(1+0.76)\times 2^{-17}\\
&=1.000101000111101_2+(1+0.52)\times 2^{-18}\\
&=1.0001010001111010_2+(1+0.04)\times 2^{-19}\\
&=1.0001010001111010111_2+0.08\times 2^{-20}
\end{aligned}
$$

$$
\begin{aligned}
&1.08_{10} \\
&=1+0.08 \\
&=1.000_2+2^{-4}+0.28\times2^{-4} \\
&=1.0001_2+2^{-6}+0.12\times2^{-6} \\
&=1.000101000_2+2^{-10}+0.92\times2^{-10} \\
&=1.0001010001_2+2^{-11}+0.84\times2^{-11} \\
&=1.0001010001_2+2^{-12}+0.68\times2^{-12} \\
&=1.0001010001_2+2^{-13}+0.36\times2^{-13} \\
&=1.0001010001_2+2^{-15}+0.44\times2^{-15} \\
&=1.0001010001_2+2^{-17}+0.76\times2^{-17} \\
&=1.0001010001_2+2^{-18}+0.52\times2^{-18} \\
&=1.0001010001_2+2^{-19}+0.04\times2^{-19} \\
&=1.0001010001_2+0.08\times2^{-20}
\end{aligned}
$$

$$1.08_{10}$$
$$=1+0.08$$
$$=1.000_2+2^{-4}+0.28\times2^{-4}$$

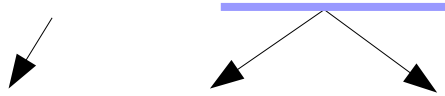


$$=1.0001_2+2^{-6}+0.12\times2^{-6}$$
$$=1.0001010000_2+2^{-10}+0.92\times2^{-10}$$
$$=1.0001010001_2+2^{-11}+0.84\times2^{-11}$$
$$=1.0001010001 1_2+2^{-12}+0.68\times2^{-12}$$
$$=1.0001010001 11_2+2^{-13}+0.36\times2^{-13}$$
$$=1.0001010001 1110_2+2^{-15}+0.44\times2^{-15}$$
$$=1.0001010001 1110 10_2+2^{-17}+0.76\times2^{-17}$$
$$=1.0001010001 1110 101_2+2^{-18}+0.52\times2^{-18}$$
$$=1.0001010001 1110 1011_2+2^{-19}+0.04\times2^{-19}$$
$$=1.0001010001 1110 1011 1_2+0.08\times2^{-20}$$

1.08_{10}

$=1+0.08$

$=1.000_2 \ \ +2^{-4}+0.28\times2^{-4}$



$=1+0.0001_2+2^{-6}+0.12\times2^{-6}$

$=1.000101000_2+2^{-10}+0.92\times2^{-10}$

$=1.0001010001_2+2^{-11}+0.84\times2^{-11}$

$=1.00010100011_2+2^{-12}+0.68\times2^{-12}$

$=1.000101000111_2+2^{-13}+0.36\times2^{-13}$

$=1.0001010001111_2+2^{-15}+0.44\times2^{-15}$

$=1.00010100011110_2+2^{-17}+0.76\times2^{-17}$

$=1.00010100011110101_2+2^{-18}+0.52\times2^{-18}$

$=1.00010100011110101011_2+2^{-19}+0.04\times2^{-19}$

$=1.00010100011110101111_2+0.08\times2^{-20}$

1.08_{10}

$=1+0.08$

$=1.000_2 \quad +2^{-4} \quad +0.28\times2^{-4}$

$=1+0.0001_2 \quad +2^{-6} \quad +0.12\times2^{-6}$

$=1.+0.0001+0.000001_2+2^{-10}+0.92\times2^{-10}$

$=1.0001010001_2+2^{-11}+0.84\times2^{-11}$

$=1.00010100011_2+2^{-12}+0.68\times2^{-12}$

$=1.000101000111_2+2^{-13}+0.36\times2^{-13}$

$=1.0001010001111_0_2+2^{-15}+0.44\times2^{-15}$

$=1.00010100011111010_2+2^{-17}+0.76\times2^{-17}$

$=1.00010100011111010_1_2+2^{-18}+0.52\times2^{-18}$

$=1.00010100011111010_1_1_2+2^{-19}+0.04\times2^{-19}$

$=1.00010100011111010_1_1_1_2+0.08\times2^{-20}$

$$1.08_{10}$$
$$=1+0.08$$

$$=1+(2^{-4}+2^{-6}+2^{-10}+2^{-11}+2^{-12}+2^{-13}$$
$$+2^{-15}+2^{-17}+2^{-18}+2^{-19})+0.08\times2^{-20}$$

$$=1+(2^{16}+2^{14}+2^{10}+2^{9}+2^{8}+2^{7}+2^{5}+$$
$$2^{3}+2^{2}+2^{1})\times2^{-20}+0.08\times2^{-20}$$

$$=1+(2^{16}+2^{14}+2^{10}+2^{9}+2^{8}+2^{7}+2^{5}+$$
$$2^{3}+2^{2}+2^{1}+0.08)\times2^{-20}$$

$$=1+(10100011110101112+0.08)\times2^{-20}$$

$$=1.00010100011110101112+0.08\times2^{-20}$$

1.08_{10}

$=1+0.08$

$=1+(1010001111010111_2+0.08)\times 2^{-20}$

$=1+0.00010100011110101112+0.08\times 2^{-20}$

$=1+0.00010100011110101112$

$+(0.00010100011110101112+0.08\times 2^{-20})\times 2^{-20}$

$=1+0.00010100011110101112$

$+0.00010100011110101112\times 2^{-20}$

$+(0.00010100011110101112+0.08\times 2^{-20})\times 2^{-40}$

$=1.00010100011110101111$

0001010001111010111

$0001010001111010111\ldots$

数値表現

10進数の$1.08 \equiv 1.08_{10} = 1+8\times10^{-2}$
1の位=1、10^{-1}の位=0、10^{-2}の位=8ということだから
2進数で表現すると
$1.08_{10} = 1+0.08 = 1+1.28\times2^{-4}$
$= 1+1\times2^{-4}+0.28\times2^{-4} = 1.0001_2+0.28\times2^{-4}$
$= 1.0001_2+1.12\times2^{-6} = 1.000101_2+0.12\times2^{-6}$
$= 1.000101_2+1.92\times2^{-10} = 1.0001010001_2+\cdots$

$= 1.00010100011110101111$

十進数と二進数では表現できる数値が異なります。
数値の表現方法によって、計算も異なるのです。

休憩

- 消費税が5%のときの計算は2進数で何倍？

数値表現

小数が無ければ問題なし？

整数の計算に問題が無いなら全てを整数と桁合せの計算にしてしまえば良い→仮想小数点法

$3.1415926535=31415926535\times10^{-10}$

乗除算は整数部の計算と指数部の計算にする
加減算は先に桁合せをしてから計算すれば良い

数値表現

小数が無ければ問題なし？

アボガドロ数＝6.02214150×10^{23}
＝
$11111111000011000011000001100000100$
$01011010111111111110001001110000000$
000000000_2
9桁?の10進数が79桁の2進数になった

小数点や「×」と、10^{23}を入れて15文字が
79bit（ASCIIで10～12文字）で表わせたから良い？
位取りは固定情報だから元は11桁では？

数値表現

アボガドロ数の表記の意味

$$6.02214150\times10^{23}$$

国際学術連合会議（ICSU）科学技術データ委員会（CODATA）による推奨値

正確には、

$$(6.02214150\pm0.00000010)\times10^{23}$$

つまり 6.02214140×10^{23}～6.02214160×10^{23} のどこかに正しい値があるということ

これを2進数で表現するにはどうしたら良い？

数値表現

6.02214140×10^{23}～6.02214160×10^{23}

の上限と下限は

11111111000011000011000000011001011111010111101
10001110011001100000000000000000

～

11111111000011000011000010100111100110010100
10011011011110100000000000000000

これを

11111111000011000011000000011001011111010111
10110001110011011×2^{17}

～

11111111000011000011000010100111100110010100
10011011011110100×2^{17}

さらに

(11111111000011000011000000011001011111010111
10110001110011001111±1)×2^{16}

と表現しても良いのだろうか。

数値表現

$(6.02214150 \pm 0.00000010) \times 10^{23}$

の上限と下限の差は20×10^{15}
$= 100011100001101111100100110111111100000100000000000000000_2$

$(11111111000011000011000000011001011111010111011000111001100111 \pm 1) \times 2^{16}$

の上限と下限の差は$2^{17} = 131072 = 13.1072 \times 10^4$

幅が全く違う

数値表現

浮動小数点数

$X.Y \times 10^{Z}$

（例えば$6.02214150 \times 10^{23}$）

X.YのY以下を四捨五入して$X.Y \times 10^{Z}$を得る範囲

$6.02214150 \times 10^{23}$なら

$6.022141495 \times 10^{23}$以上$6.02214155 \times 10^{23}$未満

すなわち、

$6.02214150 \times 10^{23}$と 6.0221415×10^{23}とが

表わす量はその精度が異なることになります。

数値表現

2進n桁の浮動小数点数

$$B_0.B_1B_2\cdots B_{n-2}.B_{n-1}\times 2^{\beta}=(B_0+B_1\times 2^{-1}+\cdots+B_1\times 2^{-n+1})\times 2^{\beta}$$

$B_0,\ldots,B_{n-1}$は整数部の各桁(=0or1)　βは符号付2進数

これを使って表現できる最大の数

$$1.1111\cdots1111_{(2)}\times 2^{\beta\max}=(1+2^{-1}+2^{-2}+\cdots+2^{-n+1})\times 2^{\beta\max}$$

これを使って表現できる最小の数

$$0.0000\cdots0001_{(2)}\times 2^{\beta\min}=(0+\cdots+2^{-n+1})\times 2^{\beta\max}$$

ではなく

$$1.1111\cdots1111_{(2)}\times 2^{\beta\min}=(1+2^{-1}+2^{-2}+\cdots+2^{-n+1})\times 2^{\beta\min}$$

一番細かい数?

数値表現

計算機で採用されている数値表現の細かさ
隣りあった数の間隔＝マシンイプシロン

正確な定義
採用されている数値表現における1に最も近い数を

$$1+\varepsilon_M$$

とするときのε_Mをマシンイプシロンと呼ぶ

2進n桁の浮動小数点数

$$B_0.B_1B_2\cdots B_{n-2}.B_{n-1}\times 2^{\beta}=\left(B_0+B_1\times 2^{-1}+\cdots+B_1\times 2^{-n+1}\right)\times 2^{\beta}$$

のマシンイプシロン　$\varepsilon_M=2^{-n+1}$

数値表現

現代の計算機で採用されている数値表現規格

IEEE754 規格（単精度）

$\pm 1.xxxxxxxxxxxxxxxxxxxxxxx \times 2^{yyyyyyyy}$

符号 1bit、仮数部(xxx...)23bit、指数部(yyy...)8bit
32bitで表現する

最上位桁は1で固定する＝正規化

マシンイプシロンは？

数値表現

現代の計算機で採用されている数値表現規格

IEEE754 規格（単精度）

$\pm 1.\mathrm{xxxxxxxxxxxxxxxxxxxxxxx} \times 2^{\mathrm{yyyyyyyy}}$

符号 1bit、仮数部(xxx...)23bit、指数部(yyy...)8bit
32bitで表現する

最上位桁は1で固定する＝正規化

マシンイプシロンは？

$$2^{-23} = 1.1920928955078 \times 10^{-7}$$

数値表現

IEEE754 規格（倍精度）

$\pm 1.xxxxxxxxxxx \ldots xxxxxxxxxxxx \times 2^{yyy \cdots yyy}$

符号 1bit、仮数部(xxx...)52bit、指数部(yyy...)11bit
64bitで表現する（単精度の2倍のbit数だから倍精度）

IEEE754 規格（4倍精度）

$\pm 1.xxxxxxxxxxx \ldots xxxxxxxxxxxx \times 2^{yyy \cdots yyy}$

符号 1bit、仮数部(xxx...)112bit、指数部(yyy...)15bit
128bitで表現する

数値表現

IEEE754 規格
指数部の表現

$\pm 1.xxxxxxxxxxx \ldots xxxxxxxxxxxx \times 2^{yyy \cdots yyy}$

指数部(yyy...yyy)は符号付整数であれば良いはずだが、規格では補数表現ではなくbias表現を採用している

補数表現:
1の補数 最上位桁＝0の2進整数を正整数とし、
　そのbit反転を対応する負整数とする
2の補数 負整数表現に1の補数に1を加えたもの用いる
　例:-1は1の補数では11...10、2の補数では11...11
　　となり、11...11と00...00が双方0になる無駄を省く
bias表現:
一定のbias値を加えた数を元の数の表現として用いる
　例:bias値を127とした場合0→127、1→128となる

数値表現

IEEE754 規格
正規化数と非正規化数

指数部の最大値と最小値(8bit なら0と255）を特別扱いにして、表現できる数値の範囲を広げる

指数部=00...00のとき、
　仮数部＝　00...00→0を表わす
　仮数部≠　00...00→非正規化数
　　=最上位桁を0とした2進化浮動小数点数
指数部=　00...01～01...11→正規化数
　　=最上位桁を1とした2進化浮動小数点数
指数部=11...11のとき
　仮数部＝　00...00→∞を表わす
　仮数部≠　00...00→非数(NaN)桁溢れに伴う警告

数値表現

もっと大きな数や細かい精度が必要なときは?

8倍精度や任意精度の数値表現規格があります。
例:指数部と仮数部の境目が変更できる数値表現や
　指数にもう一つ指数を重ねた数値表現

実際には、
　特殊な数値表現は採用し難く、
　また8倍精度の規格は確定していません。

規格化された精度での計算を組み合わせて、桁数の大きい数値の計算を行なう多倍長計算の方法が利用できます。

演習問題1

1、消費税が10%のときの計算は2進数では何倍？

2、符号1bit指数部2bit仮数部2bitの5bitで2進化浮動小数点数表現を作り、IEEE754規格をまねて、

指数部00のとき　仮数部＝　00→0
　　仮数部≠　00→非正規化数
指数部01/10/11のとき　→正規化数

とすると、この数値表現で表わすことのできる数値にはどのようなものがありますか？
全て示してください。

数値表現

練習問題
2、IEEE754規格をまねた5bitの2進符号付浮動小数点数でで表わすことのできる数値は?

符号を除けば4bitなので0000～1111の16通りを全て調べればよい

先頭2桁を指数部、残り2桁を仮数部として示す
0000 指数部00 仮数部00 →0（ゼロ）
0001 指数部00 仮数部01 →$0.01_{(2)} \times 2^0 = 0.25$
0010 指数部00 仮数部10 →$0.10_{(2)} \times 2^0 = 0.5$
0011 指数部00 仮数部11 →$0.11_{(2)} \times 2^0 = 0.75$
これ以降は正規化数

数値表現

練習問題
2、IEEE754規格をまねた5bitの2進符号付浮動小数点数でで表わすことのできる数値は?

先頭2桁を指数部、残り2桁を仮数部として示す

0100 指数部01 仮数部00 $\rightarrow 1.00_{(2)} \times 2^1 = 2$
0101 指数部01 仮数部01 $\rightarrow 1.01_{(2)} \times 2^1 = 2.5$
0110 指数部01 仮数部10 $\rightarrow 1.10_{(2)} \times 2^1 = 3$
0111 指数部01 仮数部11 $\rightarrow 1.11_{(2)} \times 2^1 = 3.5$
1000 指数部10 仮数部00 $\rightarrow 1.00_{(2)} \times 2^2 = 4$
1001 指数部10 仮数部01 $\rightarrow 1.01_{(2)} \times 2^2 = 5$
1010 指数部10 仮数部10 $\rightarrow 1.10_{(2)} \times 2^2 = 6$
1011 指数部10 仮数部11 $\rightarrow 1.11_{(2)} \times 2^2 = 7$

数値表現

訂正箇所について
郵送資料では仮数部が全て00になるという誤りがありました。

練習問題
2、IEEE754規格をまねた5bitの2進符号付浮動小数点数でで表わすことのできる数値は?

先頭2桁を指数部、残り2桁を仮数部として示す
1100 指数部11 仮数部00 $\rightarrow 1.00_{(2)} \times 2^3 = 8$
1101 指数部11 仮数部01 $\rightarrow 1.01_{(2)} \times 2^3 = 10$
1110 指数部11 仮数部10 $\rightarrow 1.10_{(2)} \times 2^3 = 12$
1111 指数部11 仮数部11 $\rightarrow 1.11_{(2)} \times 2^3 = 14$

答え、$0, \pm\{0.25, 0.5, 0.75, 2, 2.5, 3, 3.5, 4, 5, 6, 8, 10, 12, 14\}$
$11111 = \infty$を含めた32種としてもOKです。

第2回：浮動小数点数と誤差

数値表現

練習問題で使った5bit浮動小数点数について考える

実際のIEEE754規格は指数部にbias表現による符号付整数を用いる
2bit整数であれば bias値=$10_{(2)}$=2 が考えられる
指数部01→$\times 2^{-1}$ 10→$\times 2^{0}$ 11→$\times 2^{1}$ となるので、

$0100 \rightarrow 1.00_{(2)} \times 2^{-1} = 0.5$
$0101 \rightarrow 1.01_{(2)} \times 2^{-1} = 0.625$
$0110 \rightarrow 1.10_{(2)} \times 2^{-1} = 0.75$
$0111 \rightarrow 1.11_{(2)} \times 2^{-1} = 0.875$
$1000 \rightarrow 1.00_{(2)} \times 2^{0} = 1$
$1001 \rightarrow 1.01_{(2)} \times 2^{0} = 1.25$
$1010 \rightarrow 1.10_{(2)} \times 2^{0} = 1.5$
$1011 \rightarrow 1.11_{(2)} \times 2^{0} = 1.75$
$1100 \rightarrow 1.00_{(2)} \times 2^{1} = 2$
$1101 \rightarrow 1.01_{(2)} \times 2^{1} = 2.5$
$1110 \rightarrow 1.10_{(2)} \times 2^{1} = 3$
$1111 \rightarrow 1.11_{(2)} \times 2^{1} = 3.5$

数値表現

指数部00→$\times 2^{-1}$となる非正規化数と併せて

$0000 \rightarrow 0$（ゼロ）

$0001 \rightarrow 0.01_{(2)} \times 2^{-1} = 0.125$

$0010 \rightarrow 0.10_{(2)} \times 2^{-1} = 0.25$

$0011 \rightarrow 0.11_{(2)} \times 2^{-1} = 0.375$

$0100 \rightarrow 1.00_{(2)} \times 2^{-1} = 0.5$

$0101 \rightarrow 1.01_{(2)} \times 2^{-1} = 0.625$

$0110 \rightarrow 1.10_{(2)} \times 2^{-1} = 0.75$

$0111 \rightarrow 1.11_{(2)} \times 2^{-1} = 0.875$

$1000 \rightarrow 1.00_{(2)} \times 2^{0} = 1$

$1001 \rightarrow 1.01_{(2)} \times 2^{0} = 1.25$

$1010 \rightarrow 1.10_{(2)} \times 2^{0} = 1.5$

$1011 \rightarrow 1.11_{(2)} \times 2^{0} = 1.75$

$1100 \rightarrow 1.00_{(2)} \times 2^{1} = 2$

$1101 \rightarrow 1.01_{(2)} \times 2^{1} = 2.5$

$1110 \rightarrow 1.10_{(2)} \times 2^{1} = 3$

$1111 \rightarrow 1.11_{(2)} \times 2^{1} = 3.5$

2進数と浮動小数点数との大小の順序が一致する

数値表現

bias表現をそのまま指数部00→$\times 2^{-2}$とすると

$0000 \rightarrow 0$（ゼロ）

$0001 \rightarrow 0.01_{(2)} \times 2^{-2} = 0.0625$

$0010 \rightarrow 0.10_{(2)} \times 2^{-2} = 0.125$

$0011 \rightarrow 0.11_{(2)} \times 2^{-2} = 0.1875$

$0100 \rightarrow 1.00_{(2)} \times 2^{-1} = 0.5$

$0101 \rightarrow 1.01_{(2)} \times 2^{-1} = 0.625$

$0110 \rightarrow 1.10_{(2)} \times 2^{-1} = 0.75$

$0111 \rightarrow 1.11_{(2)} \times 2^{-1} = 0.875$

$1000 \rightarrow 1.00_{(2)} \times 2^{0} = 1$

$1001 \rightarrow 1.01_{(2)} \times 2^{0} = 1.25$

$1010 \rightarrow 1.10_{(2)} \times 2^{0} = 1.5$

$1011 \rightarrow 1.11_{(2)} \times 2^{0} = 1.75$

$1100 \rightarrow 1.00_{(2)} \times 2^{1} = 2$

$1101 \rightarrow 1.01_{(2)} \times 2^{1} = 2.5$

$1110 \rightarrow 1.10_{(2)} \times 2^{1} = 3$

$1111 \rightarrow 1.11_{(2)} \times 2^{1} = 3.5$

2進数と浮動小数点数との大小の順序が一致する

数値表現

指数部00→$\times 2^{-2}$に正規化数を用いた場合

$0000 \rightarrow 0$（ゼロ）

$0001 \rightarrow \underline{1}.01_{(2)} \times 2^{-2} = 0.3125$

$0010 \rightarrow \underline{1}.10_{(2)} \times 2^{-2} = 0.375$

$0011 \rightarrow \underline{1}.11_{(2)} \times 2^{-2} = 0.4375$

$0100 \rightarrow 1.00_{(2)} \times 2^{-1} = 0.5$

$0101 \rightarrow 1.01_{(2)} \times 2^{-1} = 0.625$

$0110 \rightarrow 1.10_{(2)} \times 2^{-1} = 0.75$

$0111 \rightarrow 1.11_{(2)} \times 2^{-1} = 0.875$

$1000 \rightarrow 1.00_{(2)} \times 2^{0} = 1$

$1001 \rightarrow 1.01_{(2)} \times 2^{0} = 1.25$

$1010 \rightarrow 1.10_{(2)} \times 2^{0} = 1.5$

$1011 \rightarrow 1.11_{(2)} \times 2^{0} = 1.75$

$1100 \rightarrow 1.00_{(2)} \times 2^{1} = 2$

$1101 \rightarrow 1.01_{(2)} \times 2^{1} = 2.5$

$1110 \rightarrow 1.10_{(2)} \times 2^{1} = 3$

$1111 \rightarrow 1.11_{(2)} \times 2^{1} = 3.5$

2進数と浮動小数点数との大小の順序は一致する

数値表現

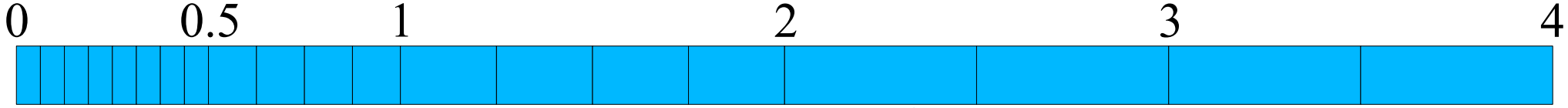
指数部を2の補数表現とした場合

$0000 \rightarrow 1.00_{(2)} \times 2^{0} = 1$
$0001 \rightarrow 1.01_{(2)} \times 2^{0} = 1.25$
$0010 \rightarrow 1.10_{(2)} \times 2^{0} = 1.5$
$0011 \rightarrow 1.11_{(2)} \times 2^{0} = 1.75$
$0100 \rightarrow 1.00_{(2)} \times 2^{1} = 2$
$0101 \rightarrow 1.01_{(2)} \times 2^{1} = 2.5$
$0110 \rightarrow 1.10_{(2)} \times 2^{1} = 3$
$0111 \rightarrow 1.11_{(2)} \times 2^{1} = 3.5$

$1000 \rightarrow 1.00_{(2)} \times 2^{-2} = 0.25$
$1001 \rightarrow 1.01_{(2)} \times 2^{-2} = 0.3125$
$1010 \rightarrow 1.10_{(2)} \times 2^{-2} = 0.375$
$1011 \rightarrow 1.11_{(2)} \times 2^{-2} = 0.4375$
$1100 \rightarrow 1.00_{(2)} \times 2^{-1} = 0.5$
$1101 \rightarrow 1.01_{(2)} \times 2^{-1} = 0.625$
$1110 \rightarrow 1.10_{(2)} \times 2^{-1} = 0.75$
$1111 \rightarrow 1.11_{(2)} \times 2^{-1} = 0.875$

2進数と浮動小数点数との大小の順序が一致しない

数値表現



最小目盛を0.0625とした数直線上に表現してみる

指数部00→×2^{-1}の非正規化数とした場合

指数部00→×2^{-2}に正規化数を用いた場合
指数部を2の補数表現とした場合

指数部00→×2^{-2}の非正規化数とした場合

数値表現（復習）

計算機で採用されている数値表現の細かさ
隣りあった数の間隔＝マシンイプシロン

正確な定義
採用されている数値表現における1に最も近い数を

$$1+\varepsilon_M$$

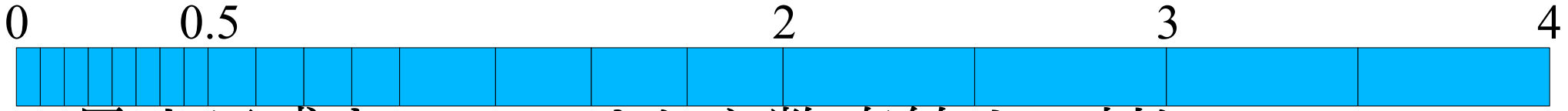
とするときのε_Mをマシンイプシロンと呼ぶ

2進n桁の浮動小数点数

$$B_0.B_1B_2\cdots B_{n-2}.B_{n-1}\times 2^{\beta}=\left(B_0+B_1\times 2^{-1}+\cdots+B_1\times 2^{-n+1}\right)\times 2^{\beta}$$

のマシンイプシロン　$\varepsilon_M=2^{-n+1}$

休憩



最小目盛を0.0625とした数直線上に対して、
以下の表現におけるマシンイプシロンは？

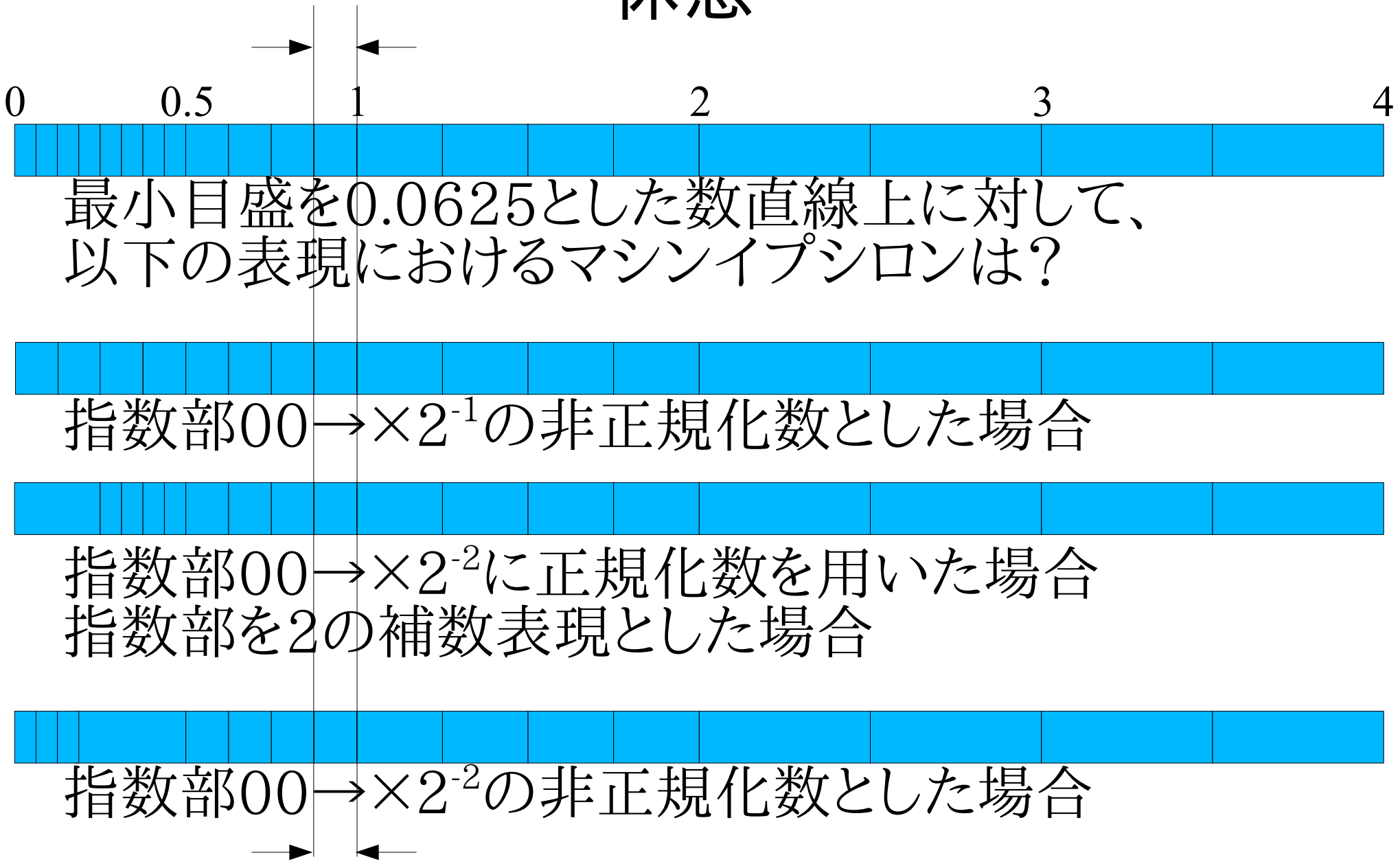
指数部00→×2^{-1}の非正規化数とした場合

指数部00→×2^{-2}に正規化数を用いた場合
指数部を2の補数表現とした場合

指数部00→×2^{-2}の非正規化数とした場合

マシンイプシロンε_Mはいくつ？

休憩



最小目盛を0.0625とした数直線上に対して、
以下の表現におけるマシンイプシロンは？

指数部00→$\times 2^{-1}$の非正規化数とした場合

指数部00→$\times 2^{-2}$に正規化数を用いた場合
指数部を2の補数表現とした場合

指数部00→$\times 2^{-2}$の非正規化数とした場合

1に一番近い値と1の差なので、全てε_M=0.125

数値表現(復習)

現代の計算機で採用されている数値表現規格

IEEE754 規格(単精度)

$\pm 1.\mathrm{xxxxxxxxxxxxxxxxxxxxxxx} \times 2^{\mathrm{yyyyyyyy}}$

符号 1bit、仮数部(xxx...)23bit、指数部(yyy...)8bit
32bitで表現する

最上位桁は1で固定する＝正規化

マシンイプシロンは

$$2^{-23} = 1.1920928955078 \times 10^{-7}$$

数値表現(復習)

IEEE754 規格(倍精度)

$\pm 1.xxxxxxxxxxx \ldots xxxxxxxxxxxx \times 2^{yyy \cdots yyy}$

符号 1bit、仮数部(xxx...)52bit、指数部(yyy...)11bit
64bitで表現する(単精度の2倍のbit数だから倍精度)

IEEE754 規格(4倍精度)

$\pm 1.xxxxxxxxxxx \ldots xxxxxxxxxxxx \times 2^{yyy \cdots yyy}$

符号 1bit、仮数部(xxx...)112bit、指数部(yyy...)15bit
128bitで表現する

数値表現(復習)

IEEE754 規格
指数部の表現

$\pm 1.xxxxxxxxxxx \ldots xxxxxxxxxxxx \times 2^{yyy \cdots yyy}$

指数部(yyy...yyy)は符号付整数であれば良いはずだが、規格では補数表現ではなくbias表現を採用している

補数表現:
1の補数 最上位桁＝0の2進整数を正整数とし、
　そのbit反転を対応する負整数とする
2の補数 負整数表現に1の補数に1を加えたもの用いる
　例:-1は1の補数では11...10、2の補数では11...11
　　となり、11...11と00...00が双方0になる無駄を省く
bias表現:
一定のbias値を加えた数を元の数の表現として用いる
　例:bias値を127とした場合0→127、1→128となる

数値表現（復習）

IEEE754 規格
正規化数と非正規化数

指数部の最大値と最小値(8bit なら0と255）を特別扱いにして、表現できる数値の範囲を広げる

指数部=00...00のとき、
　仮数部＝　00...00→0を表わす
　仮数部≠　00...00→非正規化数
　　=最上位桁を0とした2進化浮動小数点数
指数部=　00...01～01...11→正規化数
　　=最上位桁を1とした2進化浮動小数点数
指数部=11...11のとき
　仮数部＝　00...00→∞を表わす
　仮数部≠　00...00→非数(NaN)桁溢れに伴う警告

誤差

誤差

x の測定結果が X であるとき X と x の差を誤差と呼ぶ

本当？

本当であるとしたら、誤差はどうやって測る？

誤差

x の測定結果が X であるとき X と x の差を誤差と呼ぶ

「測定結果Xの誤差はd」と言うとき、正しい値xについて

$$X-d<x<X+d$$

としか判らないのが普通。

上式が成立するとき $x=X\pm d$ と表現する。

ただし、ここで $d>0$ とする。
dは一般に誤差、あるいは絶対誤差と呼ばれる。

相対誤差

Xの xに対する絶対誤差が

$$x=X\pm d$$

と表現されるとき、正しい値xに対するdの比

$$|d|/|x|$$

を、Xの xに対する相対誤差と呼ぶ

丸め誤差

実数 x を浮動小数点数 X で表現した場合

$$x=\pm 1.x_1x_2x_3\ldots x_n\ldots{}_{(2)}\times 2^{Z}$$

に対してXは

$$X=\pm 1.x_1x_2x_3\ldots x_n{}_{(2)}\times 2^{Z}$$

であり、絶対誤差は

$$X=x\pm x_{n+1}.x_{n+2}x_{n+3}\ldots{}_{(2)}\times 2^{Z-n-1}$$

相対誤差は

$$|X-x|/|x|=x_{n+1}.x_{n+2}x_{n+3}\ldots{}_{(2)}\times 2^{-n-1}$$

と見積ることができる

実際にxがXで表現されているとき、x_{n+1}以降を知ることはできず、相対誤差をε_M以下とだけ見積ることができる

$$X=x\pm\varepsilon_M\times 2^{Z},\qquad |X-x|/|x|=\varepsilon_M$$

計算誤差

xとyの2つの量について

$$x=X\pm a,\ \ y=Y\pm b,$$

としか判らないとき、

$$x+y=?$$

誤差を含む数値の足し算で起こること

計算誤差

xとyの2つの量について

$$x=X\pm a,\ \ y=Y\pm b,$$

としか判らないとき、$x+y=?$

$$X-a<x<X+a,\ \ Y-b<y<Y+b$$

なのだから

$$(X-a)+(Y-b)<x+y<(X+a)+(Y+b)$$

すなわち

$$x+y=X+Y\pm(a+b)$$

計算をしたら誤差が大きくなった⇒**誤差の伝搬**

計算誤差

xとyが数値表現X,Yに等しく丸め誤差が無い場合

$$x=X,\ y=Y,$$

には誤差は含まれていない

このとき、加算 $x+y$ の計算誤差はゼロ？

計算誤差

xとyが数値表現X,Yに等しく丸め誤差が無い場合

$$x=X,\ y=Y,$$

加算 $x+y$ 誤差はゼロ？

符号1bit指数部bias値2の2bit仮数部2bitの5bit表現で

$$1.5+0.875+0.625=3$$

確かめる（全ての項、答は丸め誤差なしで表現可能）

浮動小数点で式を示す

$$1.10_{(2)}\times 2^{0}+1.11_{(2)}\times 2^{-1}+1.01_{(2)}\times 2^{-1}=1.10_{(2)}\times 2^{1}=3$$

結合則のもとで計算する

$$\left(1.10_{(2)}\times 2^{0}+1.11_{(2)}\times 2^{-1}\right)+1.01_{(2)}\times 2^{-1}$$
$$=\left(1.10_{(2)}\times 2^{0}+0.111_{(2)}\times 2^{0}\right)+1.01_{(2)}\times 2^{-1}$$
$$=10.011_{(2)}\times 2^{0}+1.01_{(2)}\times 2^{-1}=10.011_{(2)}\times 2^{0}+0.101_{(2)}\times 2^{0}$$
$$=11.00_{(2)}\times 2^{0}=1.10_{(2)}\times 2^{1}=3$$

計算誤差

符号1bit指数部bias値2の2bit仮数部2bitの5bit表現で

$$1.5+0.875+0.625=3$$

確かめる（全ての項、答は丸め誤差なしで表現可能）
浮動小数点で式を示す

$$1.10_{(2)}\times 2^{0}+1.11_{(2)}\times 2^{-1}+1.01_{(2)}\times 2^{-1}=1.10_{(2)}\times 2^{1}=3$$

実際の加算は順番に行なう（一度には足せない）

結合則のもとで計算する

$$\left(1.10_{(2)}\times 2^{0}+1.11_{(2)}\times 2^{-1}\right)+1.01_{(2)}\times 2^{-1}$$
$$=\left(1.10_{(2)}\times 2^{0}+0.111_{(2)}\times 2^{0}\right)+1.01_{(2)}\times 2^{-1}$$
$$=10.011_{(2)}\times 2^{0}+1.01_{(2)}\times 2^{-1}=10.011_{(2)}\times 2^{0}+0.101_{(2)}\times 2^{0}$$
$$=11.00_{(2)}\times 2^{0}=1.10_{(2)}\times 2^{1}=3$$

計算誤差

符号1bit指数部bias値2の2bit仮数部2bitの5bit表現で

$$1.5+0.875+0.625=3$$

確かめる（全ての項、答は丸め誤差なしで表現可能）

浮動小数点で式を示す

$$1.10_{(2)}\times 2^{0}+1.11_{(2)}\times 2^{-1}+1.01_{(2)}\times 2^{-1}=1.10_{(2)}\times 2^{1}=3$$

実際の加算は順番に行なう（一度には足せない）

結合則のもとで計算する

$$\left(1.10_{(2)}\times 2^{0}+1.11_{(2)}\times 2^{-1}\right)+1.01_{(2)}\times 2^{-1}$$
$$=\left(1.10_{(2)}\times 2^{0}+0.111_{(2)}\times 2^{0}\right)+1.01_{(2)}\times 2^{-1}$$
$$=\underline{10.011}_{(2)}\times 2^{0}+1.01_{(2)}\times 2^{-1}=\underline{10.011}_{(2)}\times 2^{0}+0.101_{(2)}\times 2^{0}$$
$$=11.00_{(2)}\times 2^{0}=1.10_{(2)}\times 2^{1}=3$$

こんな値は表現でできません

計算誤差

符号1bit指数部bias値2の2bit仮数部2bitの5bit表現で

$$1.5+0.875+0.625=3$$

確かめる（全ての項、答は丸め誤差なしで表現可能）

浮動小数点で式を示す

$$1.10_{(2)}\times 2^{0}+1.11_{(2)}\times 2^{-1}+1.01_{(2)}\times 2^{-1}=1.10_{(2)}\times 2^{1}=3$$

実際の加算は順番に行なう（一度には足せない）

結合則のもとで計算する

$$\left(1.10_{(2)}\times 2^{0}+1.11_{(2)}\times 2^{-1}\right)+1.01_{(2)}\times 2^{-1}$$
$$=\left(1.10_{(2)}\times 2^{0}+0.111_{(2)}\times 2^{0}\right)+1.01_{(2)}\times 2^{-1}$$
$$=\underline{10.011}_{(2)}\times 2^{0}+1.01_{(2)}\times 2^{-1}=\underline{10.011}_{(2)}\times 2^{0}+0.101_{(2)}\times 2^{0}$$
$$=10.101_{(2)}\times 2^{0}=1.01_{(2)}\times 2^{1}=2.5$$

演算結果が切り捨てられた場合

計算誤差

符号1bit指数部bias値2の2bit仮数部2bitの5bit表現で

$$1.5+0.875+0.625=3$$

確かめる（全ての項、答は丸め誤差なしで表現可能）
浮動小数点で式を示す

$$1.10_{(2)}\times 2^{0}+1.11_{(2)}\times 2^{-1}+1.01_{(2)}\times 2^{-1}=1.10_{(2)}\times 2^{1}=3$$

実際の加算は順番に行なう（一度には足せない）

結合則のもとで計算する

$$\left(1.10_{(2)}\times 2^{0}+1.11_{(2)}\times 2^{-1}\right)+1.01_{(2)}\times 2^{-1}$$
$$=\left(1.10_{(2)}\times 2^{0}+0.111_{(2)}\times 2^{0}\right)+1.01_{(2)}\times 2^{-1}$$
$$=\underline{10.011}_{(2)}\times 2^{0}+1.01_{(2)}\times 2^{-1}=\underline{10.1}_{(2)}\times 2^{0}+0.101_{(2)}\times 2^{0}$$
$$=11.0\underline{01}_{(2)}\times 2^{0}=1.10_{(2)}\times 2^{1}=3$$

演算結果が0捨1入された場合

注意：規格上の0捨1入は、まるめ後に偶数になるよう調整するので切り捨てと同じ結果になる。

計算誤差

xとyが数値表現X,Yに等しく丸め誤差が無い場合

$$x=X,\ y=Y,$$

加算 $x+y$ 誤差はゼロ？

符号1bit指数部bias値2の2bit仮数部2bitの5bit表現で

$$1.5+0.875+0.625=3$$

を計算すると結果は

$$1.01_{(2)}\times 2^{1}=2.5$$

となる、誤差は

$$|3\text{-}2.5|=0.5 \text{ or } |0.5|/|3|\sim 17\%$$

演習問題2

1.符号1bit指数部bias値2の2bit仮数部2bitの5bit表現で

$$1.5+0.875+0.625=3$$

を計算せよ、ただし次の順で加算を実行すること

$$1.5+(0.875+0.625)$$

$$=1.10_{(2)}\times 2^{0}+\left(1.11_{(2)}\times 2^{-1}+1.01_{(2)}\times 2^{-1}\right)$$

2.何故、加算の順番が異なると、計算結果も異なるのか説明せよ

計算誤差（復習）

先頭から計算した場合

$$(1.10_{(2)}\times 2^{0}+1.11_{(2)}\times 2^{-1})+1.01_{(2)}\times 2^{-1}$$
$$=(1.10_{(2)}\times 2^{0}+0.111_{(2)}\times 2^{0})+1.01_{(2)}\times 2^{-1}$$
$$=\underline{10.011}_{(2)}\times 2^{0}+1.01_{(2)}\times 2^{-1}=\underline{10.011}_{(2)}\times 2^{0}+0.101_{(2)}\times 2^{0}$$
$$=10.101_{(2)}\times 2^{0}=1.01_{(2)}\times 2^{1}=2.5$$

2項目と3項目の和を先に求めた場合

$$1.10_{(2)}\times 2^{0}+(1.11_{(2)}\times 2^{-1}+1.01_{(2)}\times 2^{-1})$$
$$=1.10_{(2)}\times 2^{0}+(1.11_{(2)}\times 2^{-1}+1.01_{(2)}\times 2^{-1})$$
$$=1.10_{(2)}\times 2^{0}+11.00_{(2)}\times 2^{-1}=1.10_{(2)}\times 2^{0}+1.10_{(2)}\times 2^{0}$$
$$=11.00_{(2)}\times 2^{0}=11.0_{(2)}\times 2^{1}=3$$

3項の和の計算結果が計算順によって異なる
→和の結合則が成立しない。

計算誤差（復習）

誤りはどこで発生したのか？

$$(1.10_{(2)}\times 2^{0}+1.11_{(2)}\times 2^{-1})+1.01_{(2)}\times 2^{-1}$$
$$=(1.10_{(2)}\times 2^{0}+0.111_{(2)}\times 2^{0})+1.01_{(2)}\times 2^{-1}$$
$$=\underline{10.011}_{(2)}\times 2^{0}+1.01_{(2)}\times 2^{-1}=\underline{10.011}_{(2)}\times 2^{0}+0.101_{(2)}\times 2^{0}$$
$$=10.101_{(2)}\times 2^{0}=1.01_{(2)}\times 2^{1}=2.5$$

1項目と2項目の和を5bit浮動小数点数にまるめたところ
$10.011_{(2)}\times 2^{0}$（正）→$1.00_{(2)}\times 2^{1}$（仮数部3桁の表現）

計算結果が$0.011_{(2)}\times 2^{0}$分だけ少なくなってしまった。
（まるめ誤差＝$1.1_{(2)}\times 2^{-2}$が発生）

~~四捨五入~~0捨1入

演算結果が0捨1入された場合

符号1bit指数部bias値2の2bit仮数部2bitの5bit表現で

$$1.5+0.875+0.625=3$$

確かめる（全ての項、答は丸め誤差なしで表現可能）

浮動小数点で式を示す

$$1.10_{(2)}\times 2^{0}+1.11_{(2)}\times 2^{-1}+1.01_{(2)}\times 2^{-1}=1.10_{(2)}\times 2^{1}=3$$

結合則のもとで計算する

$$\left(1.10_{(2)}\times 2^{0}+1.11_{(2)}\times 2^{-1}\right)+1.01_{(2)}\times 2^{-1}$$

$$=\left(1.10_{(2)}\times 2^{0}+0.111_{(2)}\times 2^{0}\right)+1.01_{(2)}\times 2^{-1}$$

$$=\underline{10.011}_{(2)}\times 2^{0}+1.01_{(2)}\times 2^{-1}=\underline{10.1}_{(2)}\times 2^{0}+0.101_{(2)}\times 2^{0}$$

$$=11.001_{(2)}\times 2^{0}=1.10_{(2)}\times 2^{1}=3$$

注意：規格上の0捨1入は、まるめ後に偶数になるよう調整するので切り捨てと同じ結果になる。

規格に準拠した計算

個々の演算における実際の誤差のふるまいはもう少し複雑です。
- 非正規数による効果
- ガードビットの利用
- 内部演算方式の違い
- 関数演算アルゴリズムの影響

IEEE954は（規格の範囲で）計算結果が正しいことを要求しています。

計算誤差（復習）

$1.10_{(2)} \times 2^{0} + 1.11_{(2)} \times 2^{-1} + 1.01_{(2)} \times 2^{-1}$

$= 11.0\mathrm{x}_{(2)} \times 2^{-1} + 1.11_{(2)} \times 2^{-1} + 1.01_{(2)} \times 2^{-1}$ (指数部を揃えた表現)

1項目は他の項と桁が異なるので、演算結果にまるめが必要になることが多い。
2項目と3項目、またその和と1項目は桁が揃うので、演算結果にまるめ誤差が発生しない。

$$
\begin{array}{lll}
x.xxxxxxx \times 2^{a} & \boxed{xxxxx.xxx} \times 2^{b} & \boxed{xxxxx.xxx} \times \\
+\underline{y.yyyyyyy \times 2^{a}} & + \underline{\boxed{}y.yyyyyyy \times 2^{b}} & + \underline{\boxed{}y.}
\end{array}
$$

実際の数値計算

2次方程式の解法

$ax^2+bx+c=0$ の解は？

判別式　　$D=b^2-4ac$

$D>0$ なら　　$x=(-b\pm\sqrt{D})/2a$

$a=2.718282$
$b=-684.4566$　　10進7桁で計算してみよう
$c=0.3161592$

例題：伊理正夫・藤野和健 著「数値計算の常識」共立出版より

実際の数値計算

2次方程式

$$2.718282x^2-684.4566x+0.3161592=0$$

の解を10進7桁（単精度）の数値計算で求める

全ての計算が単精度分は正しいものとする

この4は4.000000

判別式 $D=(-684.4566)^2-4\times2.718282\times0.316592$

$=468480.8\sout{3728356}-3.437639\sout{4499776}$

$=468477.4>0$

$\therefore x=(684.4566\pm\sqrt{D})/2a$

$\sqrt{D}=\sqrt{468477.4}=684.454089\ldots=684.4541$ なので

$(-b\pm\sqrt{D})/2a=(684.4566\pm684.4541)/5.436564$

実際の数値計算

2次方程式

$$2.718282x^2-684.4566x+0.3161592=0$$

の解を10進7桁（単精度）の数値計算で求める

$$x = (684.4566 \pm 684.4541)/5.436564$$

$$(+) \quad =1368.9107/5.436564=1368.911/5.436564$$

$$=251.7970\underline{09}\ldots=251.7971$$

$$(-) \quad =0.0025/5.436564=0.00045984\underline{9272444875}10$$

$$=0.0004598493$$

この結果は全ての数字の有効桁が7の場合のもの

0.0025≠2.500000×10-3なので

実際の数値計算

2次方程式

$$2.718282x^2-684.4566x+0.3161592=0$$

の解を **できるだけ高い精度で求める**

$$x = \frac{684.4566 \pm \sqrt{684.4566^2 - 4\times 2.718282\times 0.3161592}}{2\times 2.718282}$$

$$= \frac{684.4566 \pm \sqrt{468477.39493822022 4}}{5.436564}$$

$$\approx \frac{684.4566 \pm 684.4540853397108798 6464}{5.436564}$$

$$\begin{cases} = 251.797033078192564249 15 & (+) \\ \quad 251.7971 & \text{(有効桁数7の結果)} \\ = 0.000462545881759165 41 & (-) \\ \quad 0.0004598493 & \text{(有効桁数7の結果)} \end{cases}$$

実際の数値計算

実際の数値＝コンピュータで扱っている数値には、
誤差が含まれる。

実際の計算＝コンピュータができる計算には、
誤差が含まれる。

例：桁落ち、情報落ち、桁あふれ…
　代数的性質（結合則）も確かではない。

$\mathrm{A}+(\mathrm{B}+\mathrm{C})=(\mathrm{A}+\mathrm{B})+\mathrm{C}$ かどうか$\mathrm{A},\mathrm{B},\mathrm{C}$に依る
計算結果が全て正しいかどうかも同じ