地球惑星科学実習B
「数値計算と数値」
第2回目（2016年10月14日）

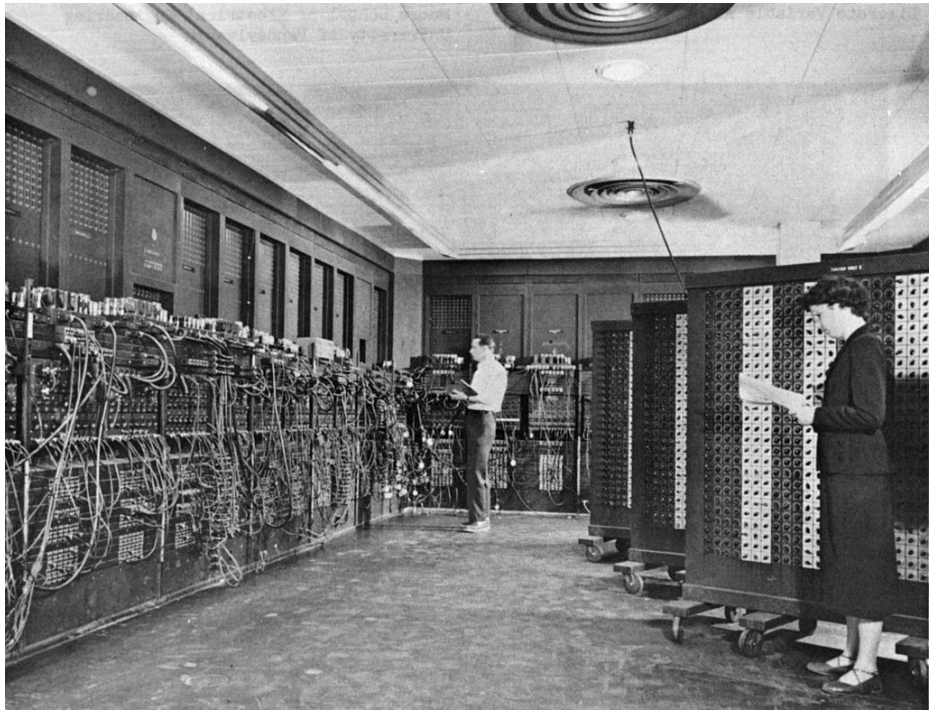
岩山隆寛
神戸大学理学研究科惑星学専攻
iwayama@kobe-u.ac.jp

目次

1. 計算機と数値
   - 計算機の中で数値はどのように表現され，記憶されているのか？
     1. 整数型
     2. 実数型
2. 誤差, 精度
   - 数値計算では誤差はつきものである．どのような誤差がありうるか？
     1. 絶対誤差と相対誤差
     2. 丸め誤差，桁落ち，打切り誤差

はじめに

- 世界初の電子計算機
  - 1946年：ENIAC (**E**lectronic **N**umerical **I**ntegrator **a**nd **C**omputer)
  - 世界初の数値天気予報に利用
- コンピュータの発達
  - 計算量の増大・・・大型計算の要求，高速の数値計算法の開発
- 学問の一分野として
  - 計算数学
  - 計算物理学
  - 数値計算法，数値解析



http://upload.wikimedia.org/wikipedia/commons/thumb/4/4e/Eniac.jpg/1024px-Eniac.jpg

計算機の中では数値はどのように表現され，記憶されているか？

１．計算機と数値

計算機が数値を記憶する形式

- 整数型:

  0, 255, -65535, 2147483647

  – 符号と大きさだけで決まる
- 実数型:

  $0.1, -25.30, 3.1416 \times 10^{2}, -1.2345 \times 10^{-5}$

  – 符号, 因数, 指数で表現
    - 有効数字:因数部の数字の先頭のゼロ以外の数字
  – 浮動小数点

    $1. \times 10^{-1}, -2.530 \times 10^{1}, 3.1416 \times 10^{2}, -1.2345 \times 10^{-5}$

整数型の記憶

- 2進数で表現
- 2進数とは
  - 各桁は0と1だけで表現される. 0から始まって1ずつ加える，各桁が2になると位が1つ上がる
  - 例: 2進数　0_2, 1_2, 10_2, 11_2, 100_2

    10進数 0,　1,　2,　3,　4
- ビット : bit
  - 2進数の各桁（0または1）をビットという
  - 計算機の場合，整数型の数値は符号を含めて32ビットが限界
    - 絶対値が最小値

      $\pm 0000000000000000000000000000000_2 = \pm 0$
    - 絶対値が最大

      $\pm 1111111111111111111111111111111_2 = \pm 2^{31}-1 = 2147483647$
    - 符号も0(+), 1(-) で表現する

演習１：計算機で扱える整数の最大・最小値

- exp_16の下にerrorというディレクトリを作成しそこに移動する.
  > cd exp_16
  > mkdir error
  > cd error
- Iwayamaのディレクトリからerror_1.fをコピーする
  > cp /home3/iwayama/exp_16/error/error_1.f .
- 上記のファイルの中身を見てみよう
- 上記のファイルを実行してみよう
  > gfortran error_1.f
  > ./a.out
- 結果を解釈してみよう

整数型数値の注意

- 整数型数値は小数点以下の値を持たない
  - 0.1や0.5は整数型としては記憶できない
  - ½の計算を行うとゼロとなる
- 整数型だけでは様々な計算を行うことができない

  ⇒　実数型の記憶方式

演習2

- 実際に下記を確かめてみよう:
  - 0.1や0.5は整数型としては記憶できない
  - ½の計算を整数型で行うとゼロとなる
  1. プログラム名はerror_2.fとして作成する. error_1.fを参考にするとよい
  2. 整数型変数 i, j, k を宣言する
  3. i, j, k にそれぞれ0.1, 0.5, ½ を代入する
  4. i, j, k の値を画面に表示する.

実数型数値の表現法

- 正負の符号・有効数字・指数の形で表現する
  - 例

    $1.234567 \Rightarrow +0.1234567 \times 10^{1}$

    $-1.234567 \Rightarrow -0.1234567 \times 10^{1}$

    $0.01234567 \times 10^{5} \Rightarrow +0.1234567 \times 10^{4}$
- 浮動小数点表示にする
- 正規化する・・・10進数の最初の数字は0とする.

16進数

- 桁数を少なくし，ビットの配列を比較的容易に見る表記法
- 各桁は0～9, A～Fで表現される. 0から始まって1ずつ加える，各桁がG (16) になると位が1つ上がる
- 例：2進数を変換すると簡単

  $0000\ 0001\ 0011\ 0100\ 0110\ 0111\ 1010\ 1111_2$

  $|0000|0001|0011|0100|\ 0110|0111|1010|1111|_2$

  $|\ 0\ |\ 1\ |\ 3\ |\ 4\ |\ 6\ |\ 7\ |\ 10\ |\ 15\ |_{16}$

  $0\ 1\ 3\ 4\ 6\ 7AF_{16}$

実数型数値の記憶

- 16進法で記憶している（単精度（32ビット）の場合）

$$\pm\, b\, a_1\, a_2\, a_3\, a_1\, a_5\, a_6$$

$a_i,\ (i = 1 \sim 6)$: 0～Fまでの数字・・・仮数部

b : 7ビットの2進数（ $-64 \leq b \leq 63$ ）・・・指数部

- 10進数表示への変換

$$\pm\left(\frac{a_1}{16} + \frac{a_2}{16^2} + \frac{a_3}{16^3} + \frac{a_4}{16^4} + \frac{a_5}{16^5} + \frac{a_6}{16^6}\right) \times 16^b$$

- 絶対値が最小の数:

$$a_i = 0 \ \Rightarrow 0$$

- 0に次に小さい数　:

$$a_1 = 1,\ a_i = 0,\ (i = 2 \sim 6), b = -64 \ \Rightarrow 16^{-65} \simeq 5. \times 10^{-79}$$

これより小さいとアンダーフロー状態

- 最大の数　:

$$a_i = \mathrm{F},\ b = 63 \ \Rightarrow \left(1 - \frac{1}{16^6}\right) \times 16^{63} \simeq 7. \times 10^{75}$$

これより大きいとオーバーフロー状態

実数型数値の精度

- 単精度（32ビット）の場合

$$\pm\, b\, a_1\, a_2\, a_3\, a_4\, a_5\, a_6$$

符号：1ビット（0・・・+, 1・・・－）

b ：7ビットの２進数（ $-64 \leq b \leq 63$ ）

$a_i,\ (i = 1 \sim 6)$：0~F・・・各4ビット

- 仮数部の桁数10進数で6ケタ　　$16^6 = 2^{24} > 2^{24-3} = 2 \times 10^6$

- 倍精度（64ビット）の場合

$$\pm\, b\, a_1\, a_2\, a_3\, a_4\, a_5\, a_6 \ldots a_{14}$$

符号：1ビット（0・・・+, 1・・・－）

b ：7ビットの２進数（ $-64 \leq b \leq 63$ ）

$a_i,\ (i = 1 \sim 14)$：0~F・・・各4ビット

- 仮数部の桁数10進15ケタ　　$16^{14} = 2^{56} > 2^{56-3} = 9 \times 10^{15}$

実数型数値の注意

- オーバーフロー，アンダーフローに注意

$$10^{-50} \times 10^{-50} = 0$$

- 演算の順序で値が変わる

$$\left(10^{-50} \times 10^{-40}\right) \times 10^{60} = 0$$

$$10^{-50} \times \left(10^{-40} \times 10^{60}\right) = 10^{-30}$$

  - 演算の順序に気を付ける必要がある

数値計算では誤差はつきものである．どのような誤差がありうるか？

２．誤差，精度

2.1　絶対誤差と相対誤差

絶対誤差

- ある物理量の真の値: a
- その近似値 : x
- 絶対誤差 : $|e|=|x-a|$
- ある小さな数 : ε
  - 次のような ε を誤差の限界という

$$|e| = |x - a| \leq \epsilon$$

  - 同じ物理量 a を表現しているときに，ε の小さい表現法がより精度が高い.

$$x - \epsilon \leq a \leq x + \epsilon$$

相対誤差

- 異なる物理量の測定，表現で誤差を比べるときには絶対誤差での比較は不適当.
  - 例
    - $a = 1.0$ のときの $\epsilon = 1.0 \times 10^{-7}$
    - $a = 1.0 \times 10^{-6}$ のときの $\epsilon = 1.0 \times 10^{-7}$
- 相対誤差

$$e_R = \frac{e}{a} \quad \text{または} \quad e_R = \frac{e}{x}$$

- 一般には精度の表現には相対誤差のほうがよく使われる. a がゼロに近いときには絶対誤差を使用する.

許容誤差

- 計算や測定を行う前に誤差の限界を設定しておくことがおおい.
- 設定された誤差の限界を許容誤差という
  - 許容絶対誤差　ε_A
  - 許容相対誤差　ε_R
- 許容誤差としてどのくらいの大きさをとったらいいか？
  - 例：マシンイプシロン
    - 現在使用している計算機で，0以外で一番小さな絶対値を持つ数値

演習3:マシンイプシロンを求めてみよう

- 手順
  1. プログラム名をerror_3.fとする
  2. epsilonという倍精度の変数を宣言する
  3. ε に1を代入する
  4. εと$1+\varepsilon$ を画面に表示する
  5. εを2で割る
  6. 再び4，5を実行する（doループを使う）
  7. 4の答えが1となってしまう一つ手前のεがマシンイプシロン
  8. プログラムが完成したら，結果の出力先を画面ではなく，ファイルに書き出してみる.

```
open (10, file='erro_r3.dat', status='new')
write(10.*) 'epsilon=', epsilon, '1+epsilon=', 1+epsilon
close(10)
```

数値計算では誤差はつきものである．どのような誤差がありうるか？

2．誤差

2.2　丸目誤差，桁落ち，打切り誤差

丸め誤差

- 計算機の中での実数型数値の記憶方法
  - 符号，指数部，仮数部
- 数値を上記の形式に変換することを「丸める」という.
- 丸めるときに生じる誤差を「丸め誤差」という
  - 無理数はもちろん丸め誤差を持つ
  - 10進数で有限桁で表現可能な数値も丸め誤差を持つ.

- 10進数で有限桁で表現可能な数値も丸め誤差を持つ
  - 例: $0.1 = 0.00011001\ldots_2$
    $= 2^{-4} + 2^{-5} + 2^{-8} + 2^{-9} + 2^{-12} + 2^{-13} + \ldots$
    $= 0.199999\ldots_{16}$
- 整数もしくは2進数, 16進数に直した時に仮数部の桁数内に入りきる数値だけが丸め誤差を持たない

計算でも丸め誤差が生じる

- 数値を記憶するとき以外にも丸め誤差を生じることがある.
- 桁合わせ
  - 例:16進数6桁(単精度)の2つの数の足し算

$$123.456_{16} + 0.\mathrm{ABF987}_{16}$$

    - 計算機内での記憶の仕方

$$0.123456_{16} \times 16^3, \qquad 0.\mathrm{ABF987}_{16} \times 16^0$$

    - 足し算に際して, 指数部の3に桁が揃えられる.

$$0.123456_{16} \times 16^3, \qquad 0.000\mathrm{ABF}_{16} \times 16^3$$

  - 引き算でも桁合わせが行われる
  - 乗除算では桁合わせは行われない
    - ただし, 結果は丸められるので丸め誤差を伴う.

$$0.123456 \times 456.789 = 0.56393342784 \times 10^1$$
$$= 0.563933 \times 10^1$$

桁落ち

- 絶対値のほぼ等しい2つの数の加減算で有効数字が失われることを「桁落ち」という
  - 丸め誤差はほとんどすべての演算の結果生じる. ただし，誤差は最終結果の最後の桁に狂いが生じる程度.
  - 桁落ちは，大きな誤差を生じる.
  - 例 : $3.14159 - 3.14158 = 0.00001$

    $= 0.1 \times 10^{-4}$

    - 元の数は6ケタの精度を持つが，最終結果は1ケタの精度.

打切り誤差

- 無限小，無限大の極限として理論的に与えられた量を数値計算するために，有限で打ち切ったことにより生じた誤差を「打切り誤差」という
  - 例：関数は級数で表現されている．

$$e^x = 1 + \frac{1}{1!}x + \frac{1}{2!}x^2 + \frac{1}{3!}x^3 + \dots$$

    - 無限級数は，有限項で打ち切って関数の値を求めている．
  - 例：微分は差分で表現

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \lim_{\Delta x \to 0} \frac{\Delta f}{\Delta x}$$

    積分は区分求積法で計算

$$\int_a^b f(x)\,\mathrm{d}x = \lim_{\Delta x \to 0} \sum_i f(x_i)\Delta x$$

演習問題4：関数の多項式による表現

三角関数は次のような無限級数和で表現できる．無限級数の上限を1から10まで変えた時の部分和と三角関数の値を比較し，三角関数が計算機内部でどのような級数で表現されているかを調べなさい．

$$\sin x = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!}$$

上記の関数についてはサンプルプログラムerror_4.fがある

$$\cos x = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!}$$

error_4.fを参考にして上記の等式について確かめなさい

まとめ

1. 計算機と数値
   - 整数型（32ビット），実数型（単精度：32ビット，倍精度：64ビット）
   - 計算機の中で扱える，最大，最小の数値がある
2. 誤差の種類
   - 絶対誤差と相対誤差
     - 真値が0に近いときには，絶対誤差をそうでないときには相対誤差を使用
   - 丸め誤差：数値を計算機内で記憶するときに生じる
   - 桁落ち：引き算の際に生じる
   - 打切り誤差：関数の級数表現，微分，積分の際などに生じる
3. Fortranの復習
   - 変数の宣言
   - 代入文
   - write文：画面への書き出し．ファイルへの書き出し
   - do文
   - function 文

参考文献

- 川上一郎:「数値計算」. 岩波書店, 1989年.
- 伊理正夫, 藤野和健:「数値計算の常識」. 共立出版株式会社, 1985年.