神戸市立工業高等専門学校
電気工学科／電子工学科
専門科目「数値解析」

2017.4.14

数値解析への案内

山浦 剛（tyamaura@riken.jp）

本講義の到達目標

前期

- 数値を2進数で表す方法、丸め誤差、有効数字について説明できる
  - 数値解析へのガイド
- １変数方程式の数値的解法を説明できる
  - 方程式の根
- 関数の数値的補間法、合成法を説明できる
  - 曲線の推定、関数の近似*

後期

- 関数の数値的積分法を説明できる
  - 積分*
- 関数の数値的微分法を説明できる
  - 常微分方程式、偏微分方程式
- 常微分方程式の数値的解法を説明できる
  - 常微分方程式
- 連立１次方程式の数値的解法を説明できる
  - 連立1次方程式*

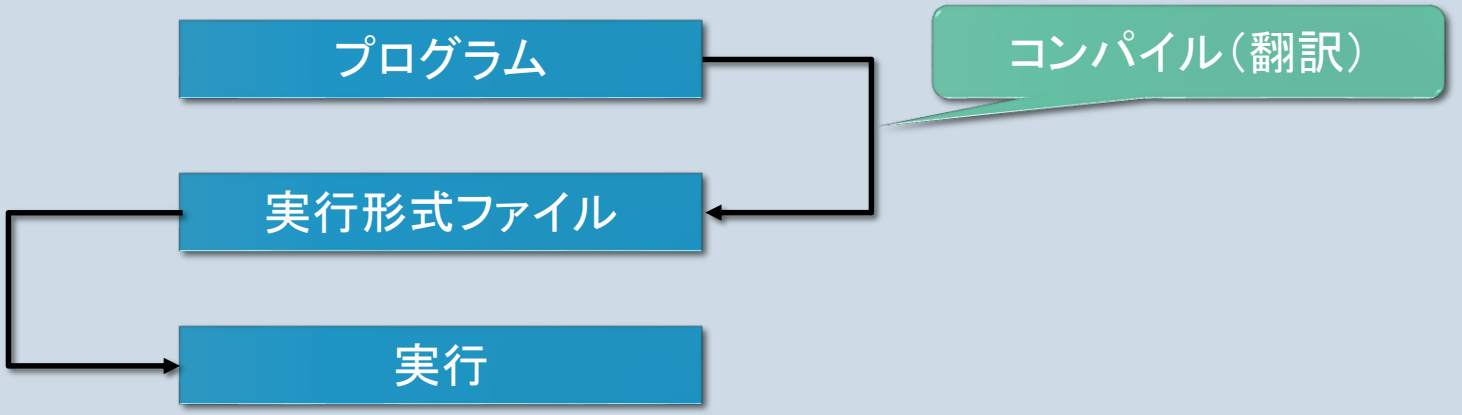
*「数値計算法入門」（松田忠重著・三恵社）の内容を一部利用

数値計算

- 数値
  - 整数: $0, 1, 8, -20, 1050$
  - 実数: $2.37, -5.11, 0.78, 4.00$
  - 複素数: $1.2 + 5.6i, 3.2 - 11.5i$
- 計算
  - 明示された手続き ⇒ アルゴリズム（手順）
  - 例1 テイラー展開による関数の近似
    - $\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \ldots$
  - 例2 流体のシミュレーション（粘性一定、非圧縮流れ）
    - $\frac{D\boldsymbol{v}}{Dt} = \frac{\partial \boldsymbol{v}}{\partial t} + (\boldsymbol{v} \cdot \boldsymbol{\nabla})\boldsymbol{v} = -\frac{1}{\rho}\boldsymbol{\nabla} \mathrm{p} + \frac{\boldsymbol{\mu}}{\boldsymbol{\rho}}\Delta \boldsymbol{v} + \boldsymbol{F}$

計算機による数値計算

- 電子計算機　⇒　単に「計算機」と呼ぶ
  - パソコンや電卓などの小型で個人使用のものから、スーパーコンピュータのような大型計算機まで。
- プログラム：アルゴリズムを計算機が実行可能な命令の並びに翻訳したもの
- プログラム言語：C/C++, C#, Java, Python, Ruby, Fortran, etc.



計算量と誤差

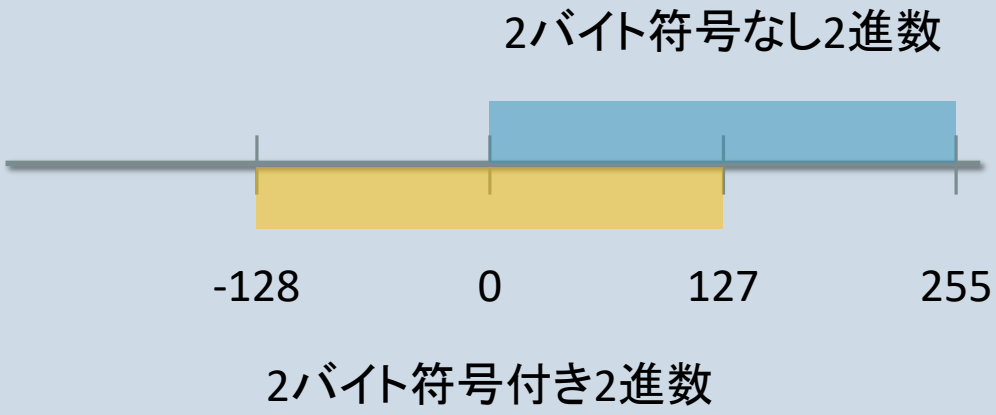
- 計算量
  - 計算にかかる時間を見積もる
    - 四則演算、代入、条件判定等々、演算の内容は様々
    - スーパーコンピュータ「京」: 1秒間に1京回（＝10^{16}回）の演算を行うことができる
    - 地球上にいる70億人が1秒間に1回演算できるとすると、1京回の演算にはおよそ16.5日程度必要
  - 計算に必要なメモリ（記憶素子）のサイズを見積もる
    - 100万個のデータを保存できるメモリ
    - 縦横100個の板の内部温度変化を考える（$100 \times 100 = 10{,}000$）
    - さらに高さ100層ある板の1秒間ごとの時間変化を100秒分考える
      - このメモリでは全て保存しながら計算することは困難
      - 工夫: 途中の時間経過のデータは上書きする、別の保存領域に移動させる

計算量と誤差

- 数値表現
  - 2進数　⇒　2個の数字（0と1）で数値を表現。2進数の各桁を“bit”と呼ぶ。
    - 例:（2進数） $101 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 5$ （10進数）
  - ビットが多いほど、表現できる数が多い。
    - 単精度　⇒　32 bit (4 byte)
    - 倍精度　⇒　64 bit (8 byte)
    - 4倍精度　⇒　128 bit (16 bit)

計算量と誤差

- 数値表現
  - 符号付き2進数　⇒　最上位ビットを“符号ビット”と呼ぶ。0なら正、1なら負。
    - 例: $1111 = -1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$
      $= -8 + 7 = -1$



2バイト符号なし2進整数		2バイト符号付き2進整数	
2進数表記	10進数表記	2進数表記	10進数表記
00000000	0	00000000	0
00000001	1	00000001	1
00000010	2	00000010	2
	……		
01111101	125	01111101	125
01111110	126	01111110	126
01111111	127	01111111	127
10000000	128	10000000	-128
10000001	129	10000001	-127
10000010	130	10000010	-126
	……		
11111101	253	11111101	-3
11111110	254	11111110	-2
11111111	255	11111111	-1

計算量と誤差

- 数値表現
  - 浮動小数点数　⇒　実数を計算機上で表現するのに都合がよい方式。
    - 例: $10.78 = 0.1078 \times 10^2$
  - 整数と同様、単精度や倍精度などの数値表現の大きさを示す型がある。
    - 仮数部ビット:　小数点以下の数値を表現するためのビット
    - 指数部ビット:　基数（10進数なら10, 2進数なら2）の何乗かを表現するためのビット
    - 符号ビット:　数値の正負を決定するためのビット

	10進数換算の仮数部の桁数N	指数部の範囲p
16bit（半精度）	3	-4 ～ +4
32bit（単精度）	6	-37 ～ +38
64bit（倍精度）	15	-307 ～ +308

計算量と誤差

- 誤差
  - 真の値からの“ずれ”
  - $\epsilon = x - \alpha$
  - 真の値: $\alpha = 12.34567$, 近似値: $x = 12.34$
  - 誤差 : $\epsilon = 12.34 - 12.34567 = -0.00567$
- 絶対誤差
  - 単に“誤差”という場合、こちらを指すことが多い。
  - $|\epsilon| = 0.00567$
- 相対誤差
  - $\epsilon_R = \frac{\epsilon}{\alpha}$

計算量と誤差

- 有効桁数
  - ある数値の先頭から数えてn桁目までが意味がある数字であることを「有効数字n桁」という。
  - 例: $0.12345678 \times 10^2 \Rightarrow 0.123457 \times 10^2$
  - 小数点7桁目を四捨五入 ⇒ 有効数字6桁
- 丸め誤差
  - 四捨五入、切り上げ、切り捨てなど、数値をある桁数に収める操作によって生じる誤差。
  - 丸め誤差によって、有効桁数が減る現象を“桁落ち”という。
- 打ち切り誤差
  - 無限回や無限小といった数学的操作を有限の操作に置き換えた近似計算によって生じる誤差。