計算機数学II (2018)
1: 数値計算へのガイド

照井 章（筑波大学 数理物質系 数学域）
Akira Terui (Institute of Mathematics, University of Tsukuba)

連絡事項

- 授業サポートページ
  https://researchmap.jp/aterui/compmath2-2018/
  - 授業スライド
  - 講義録画（出来上がり次第掲載）

第1章の内容

- 数値計算の特徴と役割
- 計算機による数値計算の流れ
- 浮動小数の表現
- 微分可能性とTaylor展開
- アルゴリズム，疑似コード，制御構造

1-1 数値計算

数値計算とは

- 数の加減乗除を組み合わせて、何らかの目的をもつ計算を行うこと

数値計算とは

- 数：基本はスカラー（浮動小数）
  - 「配列」を用いることにより、ベクトル、行列、多項式といった数学的構造を表すことも可能

数値計算とは

- ベクトル: $$\begin{pmatrix} a_0 \\ a_1 \end{pmatrix} \leftrightarrow [a_0, a_1]$$
- 行列: $$\begin{pmatrix} a_0 & a_1 \\ b_0 & b_1 \end{pmatrix} \leftrightarrow [[a_0, a_1], [b_0, b_1]]$$
- 多項式: $a_3x^3 + a_2x^2 + a_1x + a_0 \leftrightarrow [a_0, a_1, a_2, a_3]$

数値計算とは

- アルゴリズム (algorithm) :
  ある入力(式や値)に対し，ある出力(式や値)を
  出力するための計算手順(ステップ)

数値計算の役割

- 関数値の計算

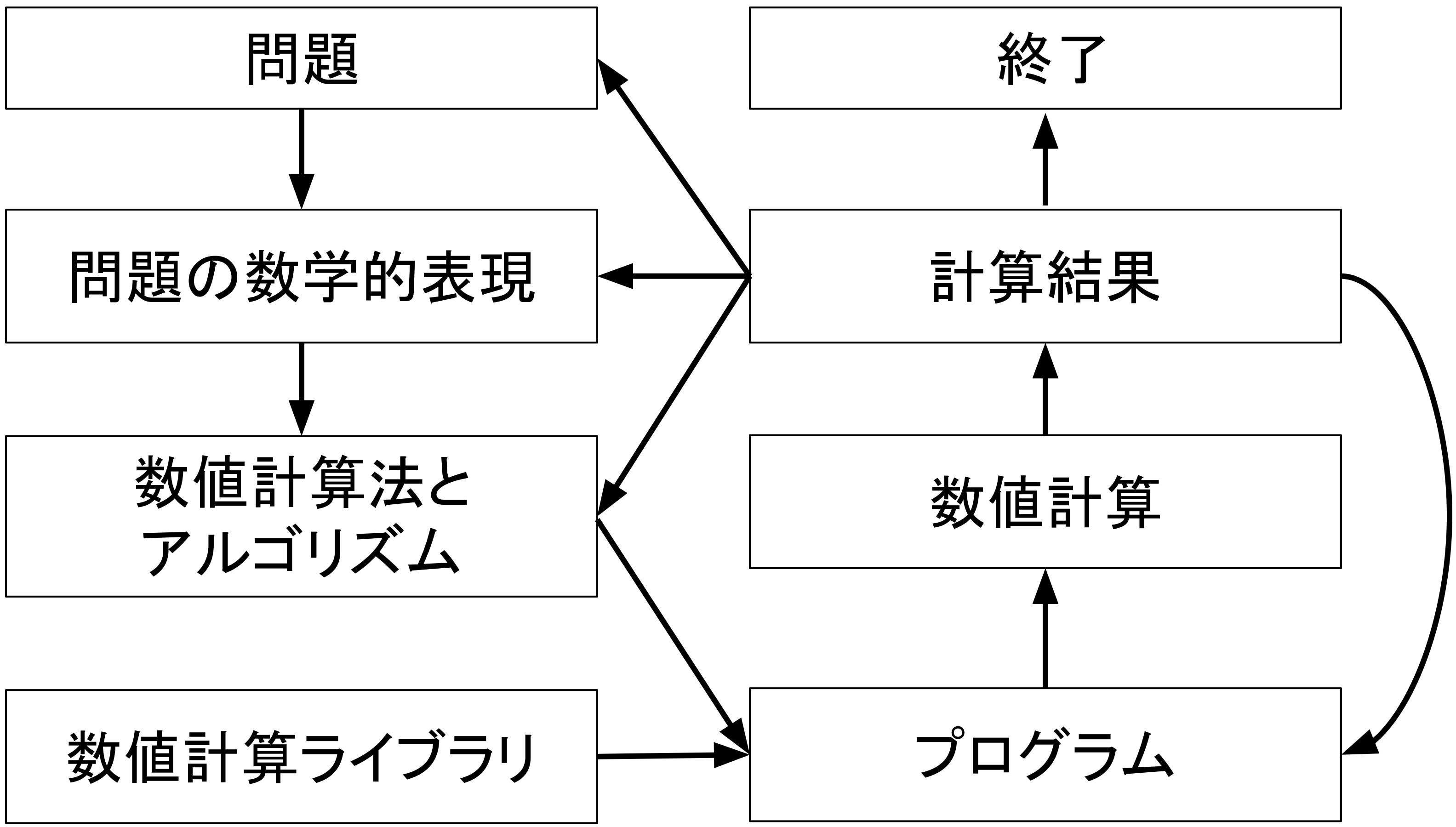
$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \cdots$$

数値計算の役割

- 解析的に解けないような問題の解を計算によって推測する（シミュレーション）

1-2 計算機による数値計算

計算機による数値計算の流れ



1-3 計算量と誤差

計算量 (Computational complexity)

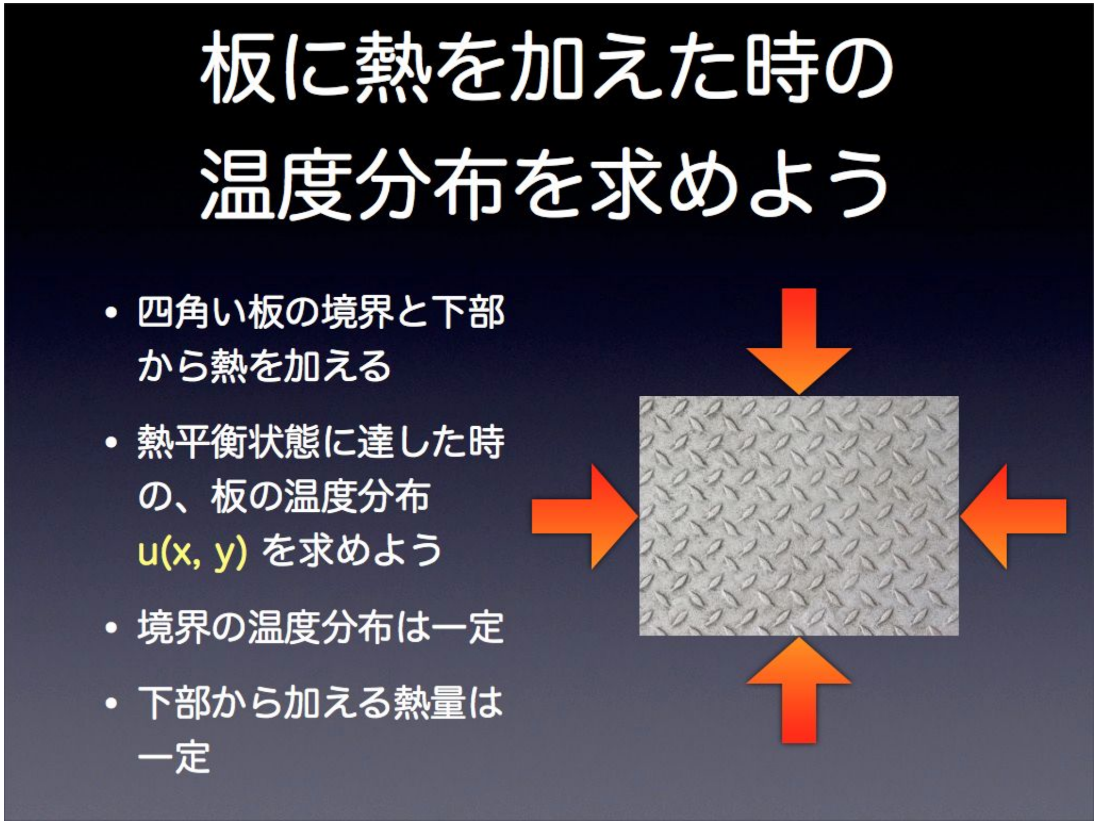
- アルゴリズムの効率を測る上での指標の一つ
  - 時間計算量 (Time complexity)
    必要な計算のステップ数
  - 空間計算量 (Space complexity)
    計算に必要な記憶容量
- 計算機数学 I (2018) 第4回も参照
  https://www.math.tsukuba.ac.jp/~terui/compmath1-2018

計算量 (Computational complexity)

- 時間計算量 (Time complexity)
  - 「漸近記法」のところで詳しく説明

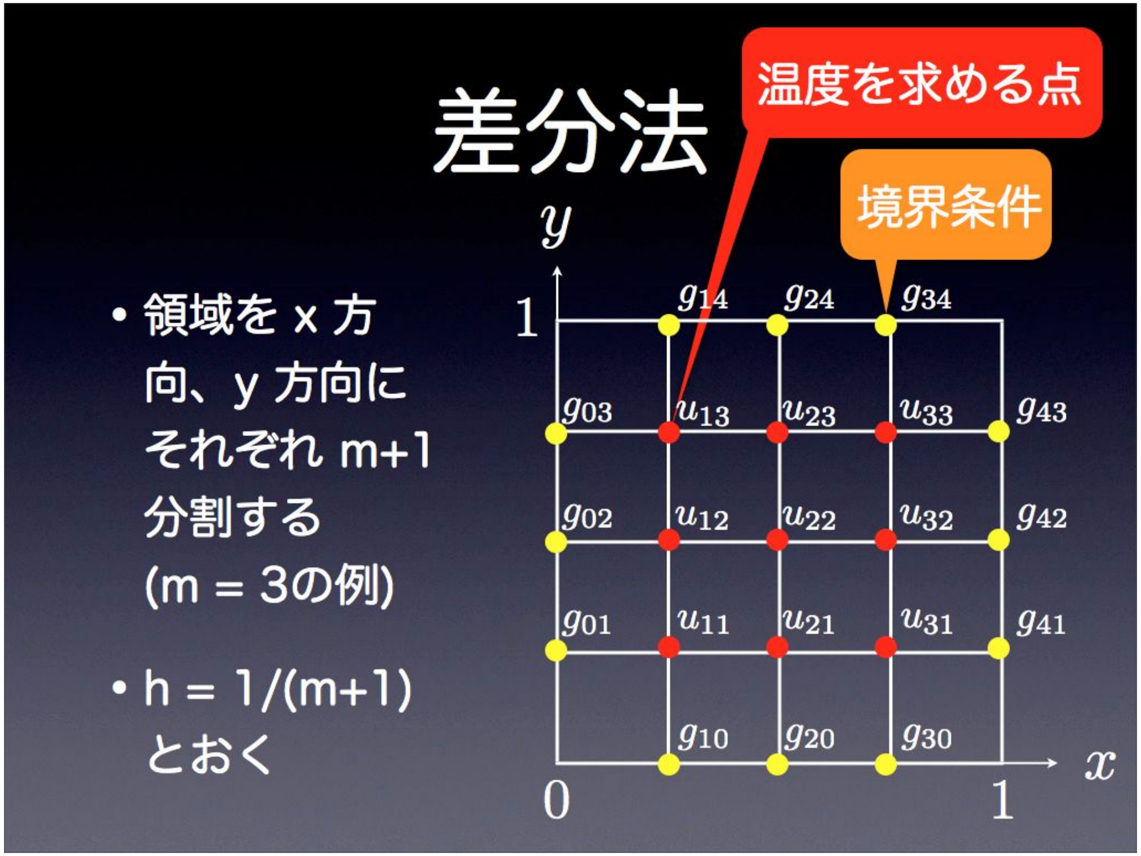
計算量 (Computational complexity)

- 空間計算量 (Space complexity) の例題



計算量 (Computational complexity)

- 空間計算量 (Space complexity) の例題



計算量 (Computational complexity)

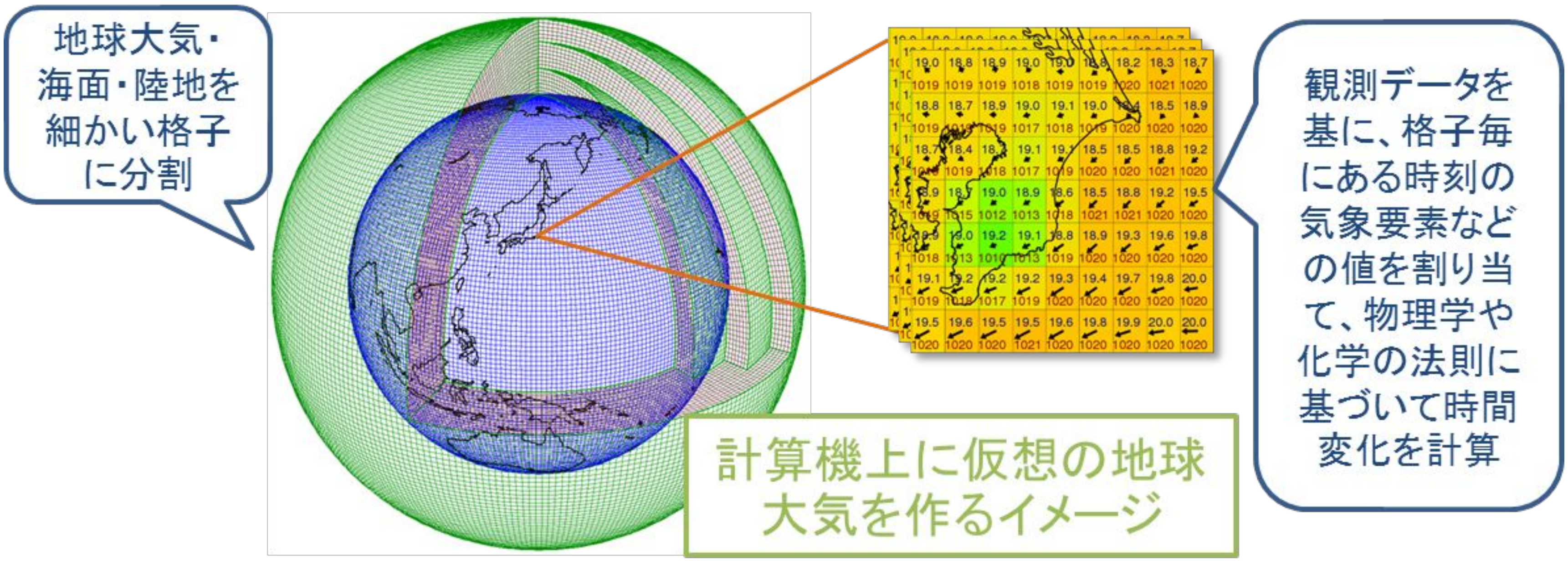
- 空間計算量 (Space complexity) の見積もり
  - x軸方向、y軸方向にそれぞれ100分割する場合:

    100 × 100 = 10000 点分
  - x軸方向、y軸方向、z軸方向にそれぞれ100分割する場合:

    100 × 100× 100 = 1000000 点分

計算量 (Computational complexity)

- 気象庁の数値予報における全球モデル



気象庁:こんにちは！気象庁です！平成30年2月号
https://www.jma.go.jp/jma/kishou/jma-magazine/1802/index.html

数値の表現

- 浮動小数点数（浮動小数）
- 計算機数学 I (2018) 第2回も参照

浮動小数による実数の表現

$$v = (-1)^s \times M \times B^E$$

s: 符号 (sign) ∈ {0,1},　*M*: 仮数 (mantissa),

B: 基数 (base), *E*: 指数 (exponent)

浮動小数の正規化

$$
\begin{aligned}
v &\simeq 2.9979 \times 10^8 \\
&= 29.979 \times 10^7 \\
&= 0.29979 \times 10^9 \\
&= \cdots
\end{aligned}
$$

（真空中の光速 m/s）

表現を一意にするために

仮数部の整数部を1以上*B*未満の整数で表す

計算機上での浮動小数の例

$$v = (-1)^s \times M \times B^E$$

$s \in \{0,1\}$, M: n ビット, $1 \leqq M < 2$,

$B = 2$,　E: l ビット

$$M = m_0.m_1 m_2 \cdots m_{n-1},$$

$$m_0 = 1, \quad m_j \in \{0,1\} \quad (1 < j < n)$$

$$\text{s.t.} \quad M = \sum_{j=0}^{n-1} \left(\frac{m_j}{2^j} \right)$$

IEEE 754 倍精度 (64ビット)

（m_0=1 とおいた正規化数の表現）

s	e_{10}	…..	e_0	m_1	……..	m_{51}	m_{52}

符号ビット

指数ビット11ビット

m_0=1

仮数ビット（正規化された暗黙の仮数ビットを含め53ビット）

$$v = (-1)^s \times M \times 2^E,$$

$$M = 1 + \sum_{i=1}^{52} \left(\frac{m_i}{2^i} \right), \quad E = -1023 + \sum_{i=0}^{11} e_i 2^i$$

(ただし -1022 ≦ E ≦ 1022)

IEEE 754規格で規定されている主な浮動小数

精度名	ビット数	仮数部 ビット数	指数部ビット数
単精度	32	23	8
倍精度	64	52	11
4倍精度	128	112	15

絶対誤差と相対誤差

- x: 真の値, x': 近似値
  - 絶対誤差: $|x' - x|$
  - 相対誤差: $|x' - x|/|x|$

有効数字と有効桁数

- 円周率の近似値

$$\underline{3.141592}324\ldots$$

- 先頭の非ゼロの数から p 桁までが正しい
  - p: 有効桁数(「有効精度 p 桁」ともいう)
  - 先頭から p 桁までの数値: 有効数字

丸め誤差

- 丸め (rounding, round-off):
  与えられた真の値 x を有限桁の数（浮動小数）x' で近似すること
- 丸め誤差: 実数を浮動小数に丸める際に生じる際の誤差

桁落ち (cancelling, cancellation)

- 絶対値がほぼ等しい数値同士の減算を行うことで、計算結果の有効桁数が減る現象
- 例:

```
    31.6386
-   31.6228
-------------
     0.0158
```

打ち切り誤差

- 級数や関数の極限値など、本来無限個の数値から計算される値を、有限個で打ち切ることで生じる誤差
- 例1（自然対数の底 e の近似）

$$e \simeq 1 + \frac{1}{1!} + \frac{2}{2!} + \cdots + \frac{5}{5!} = \frac{163}{60} \simeq 2.71667\cdots$$

打ち切り誤差

- 例2（微分係数の近似）

$$f'(a) = \lim_{h \to 0} \frac{f(a+h) - f(a)}{h} \simeq \frac{f(a+\Delta x) - f(a)}{\Delta x}$$

1-4 数学的予備知識

微分可能性

- 本授業で扱う関数は、基本的に n 階連続微分可能(C^n級)とする

Taylor展開

- $f(x)$: R → R, a ∈ R において k 階微分可能のとき

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2}(x-a)^2 + \cdots + \frac{f^{(k-1)}(a)}{(k-1)!}(x-a)^{k-1} + \frac{f^{(k)}(\xi)}{k!}(x-a)^k$$

- $\frac{f^{(k)}(\xi)}{k!}(x-a)^k$ Lagrangeの剰余項, ξ ∈ (a, x)

cos *x* のTaylor展開の例

$$\cos x = 1 - \frac{1}{2}x^2 + \frac{\sin(\xi)}{6}x^3$$

($\xi \in (0, x)$) より

$$\left|\cos x - \left(1 - \frac{1}{2}x^2\right)\right| = \left|\frac{\sin\xi}{6}x^3\right| \leq \frac{1}{6}|x^3|$$

$x \ll 1$ のとき、$1-x^2/2$ は $\cos x$ の近似式になり得る

cos *x* のTaylor展開の例

$$\left|\cos x - \left(1 - \frac{1}{2}x^2\right)\right| = \left|\frac{\sin\xi}{6}x^3\right| \leq \frac{1}{6}|x^3|$$

$x = 0.1$ のとき |cos 0.1 - 0.995| ≦ 1/6000

$x = 0.01$ のとき|cos 0.01 - 0.99995| ≦ $1/(6\text{x}10^6)$

漸近記法（ランダウの記号）

- 計算機数学 I (2018) 第4回も参照

漸近記法（ランダウの記号）

$$T : \mathbb{N} \longrightarrow \mathbb{R}_{\geq 0}$$
$$n \mapsto T(n)$$

n: 入力の「サイズ」
（数値の大きさ、桁数、ワード数、多項式の次数など）

漸近記法(ランダウの記号)

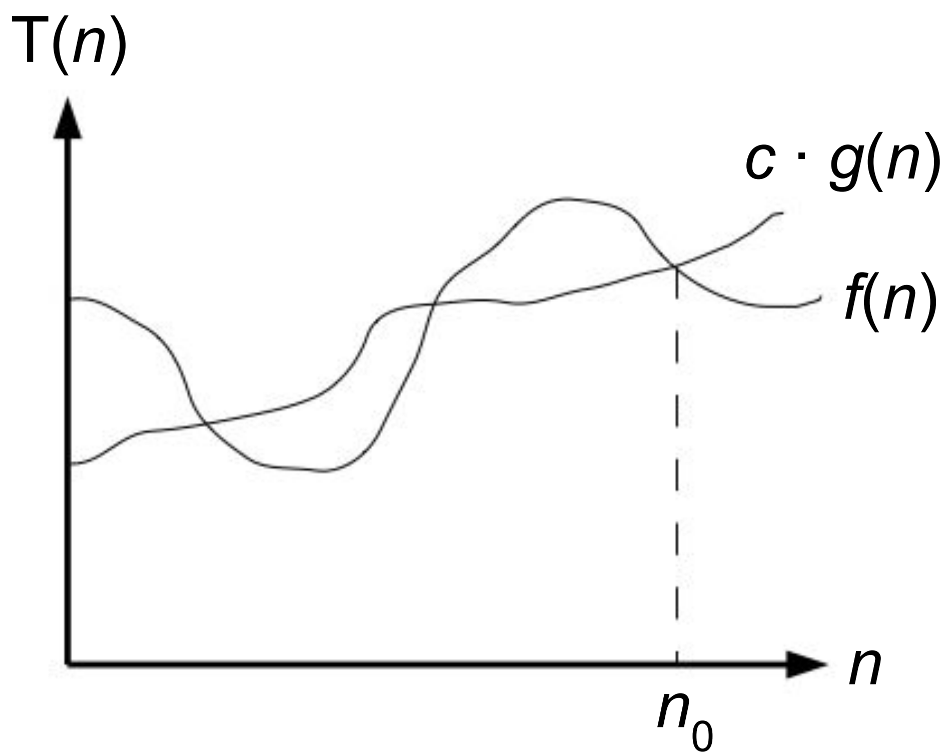
- 例: アルゴリズム 1 と 2 の時間計算量の比較
  - T_1: アルゴリズム1の時間計算量
  - T_2: アルゴリズム2の時間計算量
  - $T_1(n) = a\,n\ (a > 0)$, $T_2(n) = b\,n^2\ (b > 0)$
    ⇒ あるnの時点で $T_2(n)$ が $T_1(n)$ を追い越す
  - 他の例:
    - $T_3(n) = c^n\ (c > 0)$, $T_4(n) = d\log(n)\ (d > 0)$

漸近記法(ランダウの記号)

- O-記法: 漸近的上界⇦本授業ではこれを扱う
- o-記法
- Ω-記法: 漸近的下界
- Θ-記法
- ω-記法

O-記法（漸近的上界）

$$f(n) = O(g(n)) \overset{\text{def}}{\Longleftrightarrow} \exists c > 0,\ \exists n_0 > 0$$
$$\text{s.t. } n \geq n_0 \Longrightarrow 0 \leq f(n) \leq c \cdot g(n)$$



1-5 アルゴリズムの書式

アルゴリズムの書式

- 計算機数学 I (2018) 第3回も参照

疑似コード (Pseudo code)

Algorithm (アルゴリズムの名前など)

- 入力:
- 出力:
  - 1. (Step 1)
  - 2. (Step 2)
  - …
  - n. (Step n)
    // (コメント)

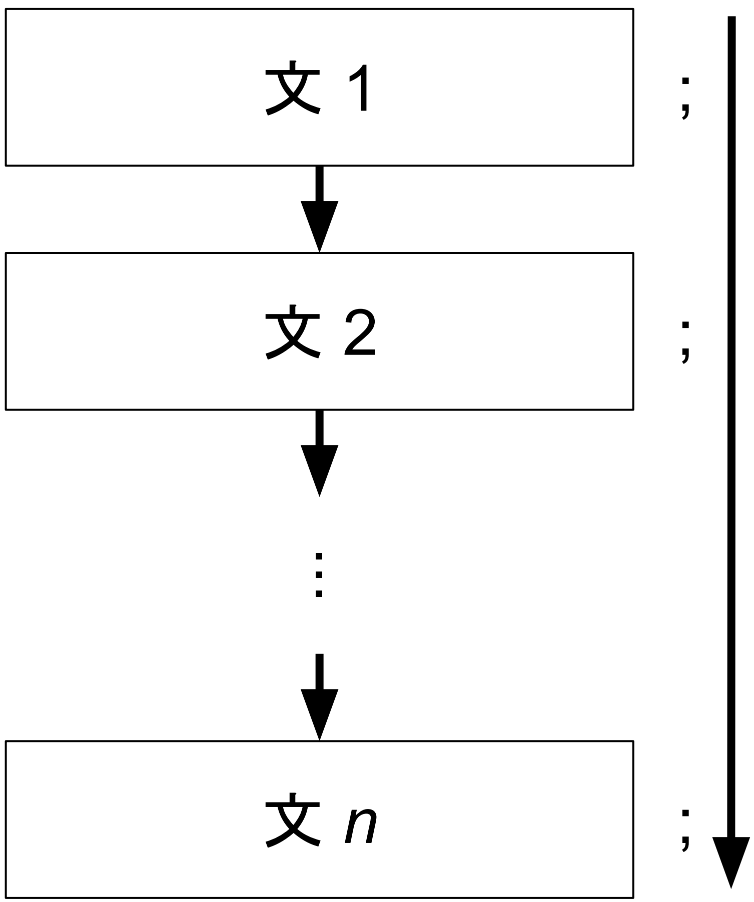
アルゴリズム特有の記法

- 変数: 数や式の値を保持する
  - 例: $p, q, u, v, w, \ldots$
- 代入: 変数に数値や式を代入する
  - 例: u ← 0; v ← p(x); …
- 論理式: 真偽値 (true or false) を返す
  - 例: u = 0; p and q; not w
- Return: アルゴリズムの値を返し，その時点でアルゴリズムを終了する
  - 例: return p;

アルゴリズムの制御構造 / Control structures

- 逐次
  - α; β; …
- 選択
  - if α then β;
  - if α then β else γ;
- 反復
  - for α do β;
  - while α do β;
  - repeat β until α;

逐次



- 上から順番に実行する
- 各文はセミコロンで区切る

選択: if 文

if α then

β: 文（のブロック）

else

γ: 文（のブロック）

;

この部分はなくてもかまわない

- α: 論理式
- αが真ならばβを実行する
- αが偽ならばγを実行する（この部分はなくてもかまわない）

反復: for 文

for α do

β: 文（のブロック） ;

α = “$i \in [m..n]$” のとき

- $m \leqq n$: i を m から1ずつ増やして n になるまでβを繰り返す
- $m \geqq n$: i を m から1ずつ減らして n になるまでβを繰り返す

反復: while 文

while α do

β: 文（のブロック）

;

- α: 論理式
- α が真の場合
  - βを実行して(論理式) の評価を繰り返す
- それ以外の場合
  - ループを脱出して次の文の処理に移る

反復: repeat ... until 文

repeat

β: 文（のブロック）

until α ;

- α: 論理式
- まずβを実行する
- αが**偽**の場合
  - 再びβを実行して (論理式) を評価する
- それ以外の場合
  - ループを脱出して次の文の処理に移る

第1章のまとめ

- 数値計算の特徴と役割
- 計算機による数値計算の流れ
- 浮動小数の表現
- 微分可能性とTaylor展開
- アルゴリズム，疑似コード，制御構造

次回の内容

- 第7章: 連立1次方程式
  - 7-1: 連立1次方程式
  - 7-2: ガウスの消去法
  - 7-3: LU分解