

# コンピュータ概論 A/B

-- Programming on Mathematica --

数学科 栗野 俊一

2011/10/25 コンピュータ概

# 伝言

---

## 私語は慎むように !!

### □ Mathematica Install

- DVD がありますので、直に申し出てください
  - ▶ 学生証と交換です

### □ 履修確認表を配布します

- 申し出てください

### □ 教室に入ったら

- 直に **Note-PC** の電源を入れておく
  - ▶ Network に接続し、当日の資料に目を通す
  - ▶ skype に Login する
  - ▶ Windows Update をしておこう

### □ やる気のある方へ

- 今日の資料は、すでに上っています
  - ▶ どんどん、先に進んでかまいません

# 追加のお知らせ

---

## □ 講義録画

○ 毎週、講義の内容が録画され Network 経由で公開されます。

○ URL

<http://10.9.74.133/video> (学内からのみ)

<http://edu-gw2.math.cst.nihon-u.ac.jp:10082> (学外からも[暫定])

○ 認証(メモってよい:その内、情報センターの ID/PW にします)

▶ ID: m23b

▶ PW: 3cdfa4747

## □ mathematica

○ 学生証と交換で DVD を貸します

▶ TA に申し出てください (全員の分を用意しました)

▶ DVD と交換で、PW を貸します

▶ PW と交換で学生証を返します

○ 数に限りがあります

▶ 効率良くやってください

○ ポイント

▶ DVD が動いているあいだは何かをやっているので、まつ

▶ 余分な事をすると、時間が無駄になる

# 先週の復習

---

## □ 先週の内容

### ○ 講義

- ▶ Mathematica の結果を TeX で利用する

### ○ 実習

- ▶ [演習 1] Mathematica でのファイル入出力
- ▶ [演習 2] TeX 形式での出力
- ▶ [演習 3] Mathematica の結果の TeX での利用方法

## □ 講義内容

### ○ 複数の tool を組み合わせて利用する

- ▶ tool 間で情報交換をどうするか？
- ▶ ファイルによる情報交換：ファイル形式が問題

### ○ Mathematica & TeX

- ▶ Mathematica が TeX で扱える形式 (TeX, eps) で出力できる
- ▶ TeXForm[式]：式を TeX の形式で画面に出力 (>> でファイルに出力する)
- ▶ Export["ファイル名",図]：図を eps の形式で出力 (TeX で利用できる)

# 本日の予定

---

- 講義

- Mathematica によるプログラミング基礎

- 実習

- [演習 1] Mathematica の変数の利用法
- [演習 2] Mathematica の関数の作成方法
- [演習 3] 課題の作成

# 本日の課題 (2011/10/18)

---

## □ 前回 (2011/10/18) の課題

### ○ 次のファイルを提出しなさい

- ▶ 20111018-QQQQ.pdf (QQQQ は学生番号)
- ▶ 内容 : Mathematica の結果を盛り込んだ PDF ファイル
- ▶ 詳しくは、配布した sample-20111018.tex の内容を参照
- ▶ これから作成された pdf ファイルを提出する

## □ 今週 (2011/10/25) の課題

### ○ 次のファイルを提出しなさい

- ▶ 表題 : ペアノの方法による「有理数の差」の関数 `qsub` を定義しなさい
- ▶ ファイル名 : 20111025-QQQQ.nb (QQQQ は学生番号)
- ▶ 詳しくは、配布した nat.txt の内容を参照

# 構文(表現)と意味

---

## □ 構文規則

- 「文字の並び」を「何の表現か?」と「判断するための規則」
  - ▶ 「 $123 + a$ 」は、「123」、「a」、「+」に分解される
  - ▶ 「123」は、「1」、「2」、「3」という「数字列」なので「数値」
  - ▶ 「a」は、「a」という「英文字から始まる文字列」なので「変数」
  - ▶ 「+」は、「足し算」を表す「演算子」
  - ▶ 「 $123 + a$ 」全体は、「式」になる
- 「どんな文字から構成されるか」という基準で区別される
  - ▶ 「何か」を表す、「表現方法」の規則
  - ▶ 規則に従って、「文字列」をみると、幾つかの「構文要素」に分解される

## □ 意味規則

- 構文規則に対応した「意味」を決める規則
  - ▶ 「123」という「数字列」は、123 という「整数値」
  - ▶ 「a」という「シンボル」は、a という「変数」
  - ▶ 「 $123 + a$ 」は、「二つの数を加える」という「文字式」

## □ 言語

- 構文規則 (いわゆる文法)と、意味規則(「辞書」ならびに、「意味」) から成る
  - ▶ 「 $1 + 2$ 」は「構文的」には、「1」と「2」の「足し算」を表す「式」となる
  - ▶ 「 $1 + 2$ 」は「意味的」には、「3」( 1, 2 は数値を表し、+ は足し算を表すから )

# Mathematica Notebook

---

## □ Mathematica Notebook (\*.nb)

- Mathematica の計算結果を保存する形式
- 結果の保存方法
  - ▶ [ファイル]→[別名で保存] : 好きな名前で保存できる
  - ▶ [ファイル]→[保存] (Ctrl-S) : 今の名前で内容を更新する(古い内容は失われる)
- 結果の利用方法
  - ▶ [ファイル]→[開く] (Ctrl-O) : 以前に保存した内容を読み込む

## □ 電卓としての利用

- 「式」を入れて [Shift]+[Enter] (以下、[SE]) すると計算
  - ▶ 入力した式は In[番号] の形で表示される
  - ▶ 計算した結果は Out[番号] の形で表示される
  - ▶ 番号は、[SE] の順番につけられる
- [SE] の効果
  - ▶ 「式」の評価が行われる
  - ▶ 「番号」が増える
  - ▶ In[番号]/Out[番号]が、それぞれ「定義」される

# 変数とその値

---

## □ シンボル (構文的な要素)

### ○ シンボルとは

- ▶ 「英字」から始まり「英数字」からなる文字列の事 (  $a$ ,  $abc$ ,  $a12z$  )
- ▶ (注意) 「英字」には、「ギリシャ文字」も含まれる (  $\pi$ ,  $\alpha$  )
- ▶ cf. 「数字」から始まる文字列はシンボルではない (  $2a$  は  $2 \times a$  となる )

### ○ Mathematica は、大文字で始まるシンボルの幾つかを事前に定義している

- ▶ (注意) 自分が利用する場合は、小文字で始まるシンボルを利用する事

## □ 変数 (意味的な要素)

### ○ 変数とは

- ▶ 名前として「シンボル」をもち、「値を持つ事ができるもの」
- ▶ 変数名は、変数の名前で、それは「シンボル」である

### ○ (即)値とは

- ▶ (現時点で)変数に対応している「式」の事
- ▶ 「環境」は、「変数名」と「変数の値」の対応表をもっている

### ○ 変数の値の参照

- ▶ 「シンボル」の形を示すを入れると、「その値」が評価され、その結果が表示される
- ▶ 「値」が設定されていない場合は、その「変数自身」が「変数の値」となる
- ▶ ?変数名で、「変数の値」である「式」を見る事ができる

# 変数への代入と定義

---

## □ 代入と定義

- 共に変数に値を「割り当て」る
- 代入：「変数 = 式」
  - ▶ 「式」は「即時評価」される。その評価結果が「変数の値」となる
- 定義：「変数 := 式」
  - ▶ 「式」は「遅延評価」される。その「式」そのものが「変数の値」となる
- 変数の値は何度でも変更できる

## □ 環境

- 変数名とその値の対応表をもっている
  - ▶ 代入も定義も、その対応表を書換えている
  - ▶ 書き換える値が違うだけ

## □ 「式」の評価(変数の場合)

- 式の中に変数名が含まれていた場合に、それを変数の値に書き換える

# 関数の利用

---

## □ 関数

### ○ 関数とは

▶ 「シンボル[引数列]」の形で表現され、値を持つ事ができる「もの」

▶ cf. `Sin[Pi] / f[3] / g[x,y^4]`

### ○ 関数の評価

▶ 「シンボル[引数列]」の形を、その値に置き換える

## □ 引数のパターンマッチ

○ 「`_`」(アンダースコア)を利用して「式」の「抽象パターン」が表現できる

### ○ 例1

▶ `next[0] := 1`

▶ `next[1] := 2`

▶ `next[_] := 0`

### ○ 例2

▶ `fib[1]=1`

▶ `fib[2]=1`

▶ `fib[x_]:=fib[x-1]+fib[x-2]`

# 自然数

---

## □ ペアノの公理

### ○ 自然数を定義する公理

- ▷ 0 は自然数である
- ▷  $n$  が自然数なら  $n+1$  も自然数である
- ▷ 上記の二つ以外に自然数はない

## □ Mathematica でペアノの公理の自然数を考える

### ○ 自然数

- ▷ 0 は 0 で表現
- ▷  $n + 1$  は  $s[n]$  で表現

### ○ 自然数の和

- ▷  $\text{padd}[0, x\_]$  :=  $x$
- ▷  $\text{padd}[s[x\_], y\_]$  :=  $s[\text{padd}[x, y]]$

### ○ これを繰り返すと、分数まで表現できる

# 自然数による数の表現 (nat.txt)

---

## □ 自然数

○ 0 と +1 (サクセッサー) のみで作る

▷  $3 = 0+1+1+1 = s[s[s[0]]]$

## □ 整数

○ 自然数の二つ組  $(m,n)$  で整数  $z$  を表現する

▷  $z = m - n \Rightarrow pp[m,n]$

▷ 同じ整数に対する異なる  $(m,n)$  組があるので、同値類( $\sim$ )を作る

▷ 例  $pp[4,2] \sim pp[3,1] \sim pp[2,0]$

## □ 有理数

○ 自然数と整数の組  $(z,n)$  で整数  $q$  を表現する

▷  $q = z/n \Rightarrow qq[z,n]$

▷ 同じ有理数に対する異なる  $(z,n)$  組があるので、同値類( $\sim$ )を作る

▷ 例  $qq[18,12] \sim pp[9,4] \sim pp[6,2]$

□