

# コンピュータ概論 A/B

-- Mathematica Programming (1) --

数学科 栗野 俊一

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く  
禁じます

2020/10/13 コンピュータ概

# お知らせ

---

## コンピュータ概論 A/B (2020/10/13)

# お知らせ

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます

# お知らせ

---

- Mathematica のネットワークインストール
  - ISO イメージのファイルが大きい場合
    - ▷ 分割したファイルをダウンロードする
- 次回の微積の対面授業 ( 2020/10/22 [木] )
  - 昼休みにサポートに、微積の教室に行きます
  - 必要な方は、**Note-PC** を持参ください

# 前回の復習

---

コンピュータ概論 A/B (2020/10/13)

## 前回の復習

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます

# 前回の復習

---

## □ 講義内容

### ○ Mathematica を利用してみる

- ▶ 数式を入れ、[SE] ([Shift] + [Enter]) すると、数式の計算をしてくれる(数式処理)
- ▶ 微積：多倍長計算/文字式(展開・因数分解)/微分/積分/極限
- ▶ 代機：ベクトル/行列計算/方程式の解/グラフの描画

### ○ 「高機能」電卓として利用可能(それだけでも十分使える)

- ▶ 「計算」だけなら、人間より優秀

### ○ Mathematica で「何か(数学用語)」をしたければ..

- ▶ 検索エンジンで「Mathematica 数学用語」で見付かる
- ▶ 数学を学ぶ事の効用：「数学用語(概念)」を沢山知る事ができる

## □ 実習

### ○ 色々な Mathematica の機能

# Mathematica の利用

---

## □ Mathematica での計算

- 「式」を入れて [SE] ([Shift]+[Enter]) で計算(評価)が行われる
- 三つの括弧の区別 ( Mathematica では、働きが違うの区別が必要 )
  - ▶ 「(、)」: 丸括弧/パーレン : 計算の優先順位を変更する(先に計算される部分を囲む)
  - ▶ 「[、]»: 角括弧/ブラケット : 関数適用を行う(関数の引数を囲む)
  - ▶ 「{、}»: 波括弧/ブレース、カーリー : 複数の要素を一つにまとめる(ベクトルの要素を囲む)
- Mathematica で「色々な事」をするには (是非、色々試してみよう)
  - ▶ 「(Mathematica の)関数」を適用すれば良い (例: 「微分」は D[] を使う)
  - ▶ 「(凄く強力な)Help」がある / Web を探せば、沢山の解説ページが..
- 「数学の計算は『何でも』 Mathematica で計算できる」と思って良い
  - ▶ 「何」が出来るか : 数学の概念(専門用語)を知っていれば良い(数学、頑張ろう)
  - ▶ 「どうやって(どの関数を利用)」するか : 検索する「Mathematica 数学概念」
  - ▶ 「出来無い」と思うと「出来る事も出来無く」なる (「出来る」、だから「探せ」)

# 本日の予定

---

コンピュータ概論 A/B (2020/10/13)

## 本日の予定

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます

# 本日(2020/10/13)の予定

---

- 本日(2020/10/13)の内容
  - Mathematica によるプログラミング基礎
- 演習
  - [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

---

コンピュータ概論 A/B (2020/10/13)

## 本日の課題

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます

# 本日の課題 (2020/10/13)

---

□ 出席パスワード : 20201013

□ 先週 (2020/10/06) の課題

○ Mathematica で 2 のべき乗を計算する

▷  $N = QQQQ + 100$  とする (QQQQ は学生番号)

▷ 2 の  $N$  乗を答えなさい

□ 今週 (2020/10/13) の課題

○ 次のファイルを Mathematica で作成して CST Portal に提出してください

▷ ファイル名 : 20201013-QQQQ.nb (QQQQ が学籍番号)

▷ 内容 : 1 から  $n$  までの 3 乗和を計算する関数 `cubeSum[n]` の作成

▷ 形式 : nb 形式 ( `sample-20201013.nb` を参照 )

# 『言語』表現

---

コンピュータ概論 A/B (2020/10/13)

## 『言語』表現

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます

# 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』の列なので「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
- 「どんな種類の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」を見ると、幾つかの「構文要素」に分解される

## □ 意味規則

- 「構文要素」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 (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 の使い方

---

コンピュータ概論 A/B (2020/10/13)

## Mathematica の使い方

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます

# Mathematica(復習)

---

## □ Mathematica とは

### ○ 数式処理言語システム

▶ 「数式」を計算したり、数式の計算を行うプログラムが作れる

### ○ 数式電卓

▶ (文字を含む)数式の計算を行う

▶ cf. 電卓 : 「数」の計算が出来る(数の式を入れると数の計算を行う)

## □ Mathematica の使い方

### ○ ノートブックを開く

### ○ 式を入力して [Shift]+[Enter] (以下 [SE] と表現)で評価開始

▶ 計算に時間がかかりそうなら.. [Alt]+[,] で中断できる

### ○ この講義では「Mathematica の導入」のみを扱う

▶ 自分で色々調べて、試してみる (Help/チュートリアル)

# 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

# Mathematica の変数と代入

---

コンピュータ概論 A/B (2020/10/13)

## Mathematica の変数と代入

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます



# 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 (  $a$ ,  $abc$ ,  $a1$ ,  $a12z$  )
- ▶ (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利用

- ▶ (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

### ○ 変数の(即)値とは

- ▶ (現時点で)変数に対応している「式(の評価結果)」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」の形を示す物を入れると、「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

---

## □ 評価代入(=)と未評価代入(:=)

- 共に変数に「値」を「割り当て」る命令

- 評価代入 : 「変数 = 式」

  - ▶ 「式」は「評価」される。その「評価結果」が「変数の値」となる

  - ▶ 「値」は、何度でも参照できる (「値」に「名前」が付けられる)

- 未評価代入 : 「変数 := 式」

  - ▶ 「式」は「評価」されない。その「式(表現)」そのものが「変数の値」となる

- 変数の値は何度でも変更(代入)できる

  - ▶ 変数に記録されているのは、最後の代入の結果(その前の値は失われる)

## □ 環境

- 変数名とその値の対応表を持っている

  - ▶ 「代入」も「定義」も、その対応表を「書き換えて」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

# Mathematica の関数

---

コンピュータ概論 A/B (2020/10/13)

## Mathematica の関数

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます

# 関数の利用

---

## □ Mathematica の関数

### ○ Mathematica の関数とは

- ▶ 「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]`
- ▶ <注意> `fib[3]` とすると、`x` に一時的に `3` が代入され、右辺(`fib[x-1]+fib[x-2]`)の評価の時に利用される

# ペアノの公理

---

コンピュータ概論 A/B (2020/10/13)

## ペアノの公理

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます

# 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 でペアノの公理の形の「自然数」を考える

### ○ 自然数 (これを「ペアノ形式の表現」と呼ぶ事にする)

- ▷ 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(有理数)」まで表現できる

- ▷ 「実数」を表現するには、「収束概念」が必要になる

# 自然数による数の表現 (nat.txt)

---

## □ 自然数

○ 0 と +1 (サクセッサ) 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  $qq[18,12] \sim pp[9,4] \sim pp[6,2]$

# 「ペアノ形式」と普通の表現の関係

表 3: 「S 表現」と普通の表現の関係

概念	普通の表現 (例)	ペアノ形式 (例)	コメント
0	0	0	0 は自然数
自然数 (0 以外)	1,2,3,..	s[0], s[s[0]], s[s[s[0]]], ..	x が自然数なら s[x] も自然数
n の次	n + 1	s[n]	s はサクセッサ (後継) 関数
和	1 + 2	padd[s[0],s[s[0]]]	
差	1 - 2	psub[s[0],s[s[0]]]	引く数の方が大きい場合は 0 にな
積	1 * 2	pmul[s[0],s[s[0]]]	
商	1 / 2	pdiv[s[0],s[s[0]]]	整数割り算 (小数点以下は切り捨て
最大公約数	gcd(1,2)	pgcd[s[0],s[s[0]]]	
整数	1, -1	pp[s[0],0], s[0,s[0]]	自然数の対を同値類で割った物が
整数の正規化		zbar	ペアのどちらか一方を 0 にする
整数の四則	+, -, *, /	zadd,zsub,zmul,zdiv	
有理数	1/2	qq[pp[s[0],0],s[s[0]]]	分子は整数で、分母が自然数の対
有理数の正規化		qbar	約分する
有理数の四則	+, -, *, /	qadd,qsub,qmul,qdiv	

## □ 可換性(well defined)

- 通常 of 自然数とペアノ形式 of 自然数は ntop, pton で同型になっている
- 個々のペアノ形式 of 関数は ntop, pton に関して可換になっている

▷ pton[padd[ntop[1],ntop[2]]]=3=1+2



おしまい

---

コンピュータ概論 A/B (2020/10/13)

おしまい

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます