

コンピュータ概論 A/B

-- TeX --

(様々な数式の表示)

数学科 栗野 俊一 (TA: 佐藤 淳 [院生 1 年])

2015/10/20 コンピュータ概

伝言

私語は慎むように !!

□ 席は自由です (出席パスワード : 20151020)

○ できるだけ前に詰めよう

○ 教室にきたら直ぐにやる事

▶ PC の電源 On / ネットワーク接続 / Web を参照する / skype を起動する

□ 色々なお知らせについて

○ 栗野の Web Page に注意する事

<http://edu-gw2.math.cst.nihon-u.ac.jp/~kurino>

□ 今週は「補習」はありません

前回(2015/10/13)の内容 (1)

□ 講義

○ Mathematica によるプログラミング基礎

▶ 自分で新しい関数を定義することができる

○ 関数定義の二つ形：共に、形式を Mathematica に合わせるだけ

▶ 「 $f(x)=式$ 」の形 (例： $f(x)=x^2$)

▶ 「帰納的定義」の形 (例：階乗、数列の総和の計算)

□ 演習

○ 色々な Mathematica の機能

前回(2015/10/13)の内容 (2)

□ Mathematica での関数の定義 (How to)

○ 「数学形式」から「Mathematica 形式」への変換規則

- ▶ どんな関数を定義するか考える (例: $f(x)=x^2$)
- ▶ 定義する関数名を決める (例: myFunc にする → $\text{myFunc}(x)=x^2$)
- ▶ 関数の引数の括弧をパーレン「()」からブラケット「[]」にする (例: $\text{myFunc}[x]=x^2$)
- ▶ 等号「=」を未評価代入「:=」にする (例: $\text{myFunc}[x]:=x^2$)
- ▶ 関数引数の後ろに「_」を付ける (例: $\text{myFunc}[x_]:=x^2$)

○ 自然数上に帰納的に定義されている関数(例:階乗)の定義は二つの部分からなる

- ▶ 最初($n=0,1$)の時の式: 値を直接指定 (例: $f(0)=1$)
- ▶ 一般の場合の式: $f(n-1)$ を使って、 $f(n)$ を定義 (例: $f(n)=n*f(n-1)$)

前回(2015/10/13)の内容 (3)

□ Mathematica : まとめ

○ Mathematica は便利

- ▶ 数学の計算は「何でもできる」と誤解しておこう
- ▶ 何ができるか → 数学で学んだ内容
- ▶ どうやるか → 検索する : Mathematica+「数学用語」

○ Mathematica での関数定義

- ▶ 既にあるものは ? (知る問題) : 検索して、それをそのまま利用する
- ▶ 丁度のものであれば ? (考える問題) : 自分で関数が定義できる

○ Mathematica の学習

- ▶ コンピュータ概論では導入のみ → とにかく、自分で色々と試してみる

本日(2015/10/20)の予定

- 本日(2015/10/20)の予定
 - (pLa)TeX(2e) とは？
 - 「メタシステム」とは？ (前回の積み残し)
- 本日(2015/10/20)の目標
 - TeX の利用方法を学ぶ
- 演習
 - [演習 1] Ubuntu 環境の構築
 - [演習 2] TeX の利用準備
 - [演習 3] platex による文章の作成
 - [演習 4] TeX でのタイプセット
 - [演習 5] TeX で色々な数式を記述する
 - [演習 6] Mathematica の数式を TeX で利用する

本日の課題 (2015/10/20)

□ 前回 (2015/10/13) 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 : 20151013-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-20151013.nb` を参照)

□ 今回 (2015/10/20) の課題

○ CST Portal に以下のファイを提出しなさい

- ▶ ファイル名 : 20151020-QQQQ.tex (QQQQ は学生番号)
- ▶ 表題 : TeX で色々な数式を記述する
- ▶ 内容 : TeX の色々な数式の記述してみる
- ▶ 条件 : 名前と学生番号は自分のものにする
- ▶ 形式 : テキストファイル (`sample-20151020.tex` 参照)

(pLa)TeX(2e) [テック] とは

□ (pLa)TeX(2e) [テック] とは

○ 文章を整形するツール(組版ソフト)

- ▶ 文章を綺麗に表示して印刷できるようにするツール
- ▶ 特に「数式」を扱う(数学科の)人間には必須(?)

○ (いわゆる..)ワープロとの違い

- ▶ 文章はテキストエディタで作成 : ワープロの場合はワープロで作る
- ▶ 入力したものと印刷結果は異なる : ワープロの場合は見たままに印刷される
- ▶ 特殊効果の指示は「マクロ」で行う
- ▶ 入力 : TeX ファイル (テキストファイル)
- ▶ 出力 : dvi ファイル (この後更に、pdf ファイルにする)

TeX システム

□ TeX システムとは

○ TeX の形で記述されている文章を処理する

- ▶ TeX 文章を綺麗に印刷・表示できる形式 (pdf) に変換する
- ▶ TeX 文章はテキストファイルなので、エディタで編集できる
- ▶ pdf は、観たり、印刷したりするための形式

○ ファイルを変換するツールである事に注意

- ▶ コンピュータのプログラムの典型的な例
- ▶ 形式を色々な形に書き換える (cf. TeX は .tex を .dvi にする)

○ 文の内容に、「形式」を指定すると、整形してくれる

- ▶ 見た目では、結果が解らない (解る必要がない)

TeX の利用法

□ TeX の利用法 (How)

- TeX の表記表現を憶える
- テキストエディタで `FooBar.tex` ファイルを作成 (`FooBar` はファイル名)
 - ▶ 文章の表現を TeX 形式で行う
- 変換しよう (ツールを利用する)
 - ▶ `platex` で、`FooBar.tex` を `FooBar.dvi` に変換
 - ▶ `dvipdfmx` で、`FooBar.dvi` `FooBar.pdf` に変換
 - ▶ `evince` で `FooBar.pdf` を表示

□ TeX の利用所 (When/Why)

- 数式が綺麗に表示される
 - ▶ 数学のプリントは、TeX で作ろう
- 数式の TeX 表現を憶えられる
 - ▶ メールや、チャットでも数式を TeX 表現で使おう

実習 1: Ubuntu 環境の構築

□ Ubuntu 環境の構築

○ フォルダの作成

- ▶ まず、「c:\usr」フォルダを作成
- ▶ 更にその下に、「vmware」、「tex」、「c」フォルダを作成する

○ Ubuntu イメージの入手と展開

- ▶ DVD から ubuntu.zip をコピーする(コピーしたら DVD は次の人へ)
- ▶ ubuntu.zip を右クリックして「すべて(T)」を選ぶ
- ▶ ubuntu フォルダが作られ、更に、その中に ubuntu があるので、それを「c:\usr\vmware」に移動

○ Ubuntu の初回起動 (次回からは、vmware のメニューに入る)

- ▶ 「c:\usr\vmware\ubuntu\Ubuntu.vmx」を開く (自動的に vmware が起動)

○ 共有フォルダの設定 (初回のみ必要)

- ▶ 「Player(P)」→「管理(M)」→「仮想マシンの設定(S)」
- ▶ 「オプション」→「共有フォルダ」→「常に有効」に設定変更

実習 2: TeX の利用 (毎回する準備)

□ TeX を利用するまでの準備 (毎回行う)

- vmware で ubuntu を起動 (ID/PW は共に「soft」が良い)
- 「端末」を開く (`soft@soft-2015010-ubuntu-ja-14:~` と表示される)
 - ▶ [Enter] キーを押すと、同じもの(プロンプト)が何度も表示される
- 作業フォルダ (`TEX_HOME` aka `tex`) に移動
 - ▶ `cd tex` (表示が `soft@soft-2015010-ubuntu-ja-14:~/tex$` に変化)
 - ▶ `ls` (`c:\usr\tex` の内容と同じものが表示される)
- この作業は、「端末」毎に行う
 - ▶ 「端末」を開くを開いたら「`cd ~/tex`」をするって事
- 場合によっては、更に「`cd フォルダ名`」とする必要がある

□ Ubuntu のコマンド

- `cd` 「フォルダ名」: フォルダを移動 (プロンプトが変化)
- `ls`, `dir` : フォルダにあるファイルリストの表示
- `pwd` : 現在の位置を表示

実習 3: TeX を利用してみる (typeset)

□ [実習 3] base.tex のタイプセット

- 「platex base.tex」を実行

- ▶ base.dvi が出来ている事を確認する

- 「dvipdfmx base.dvi」を実行

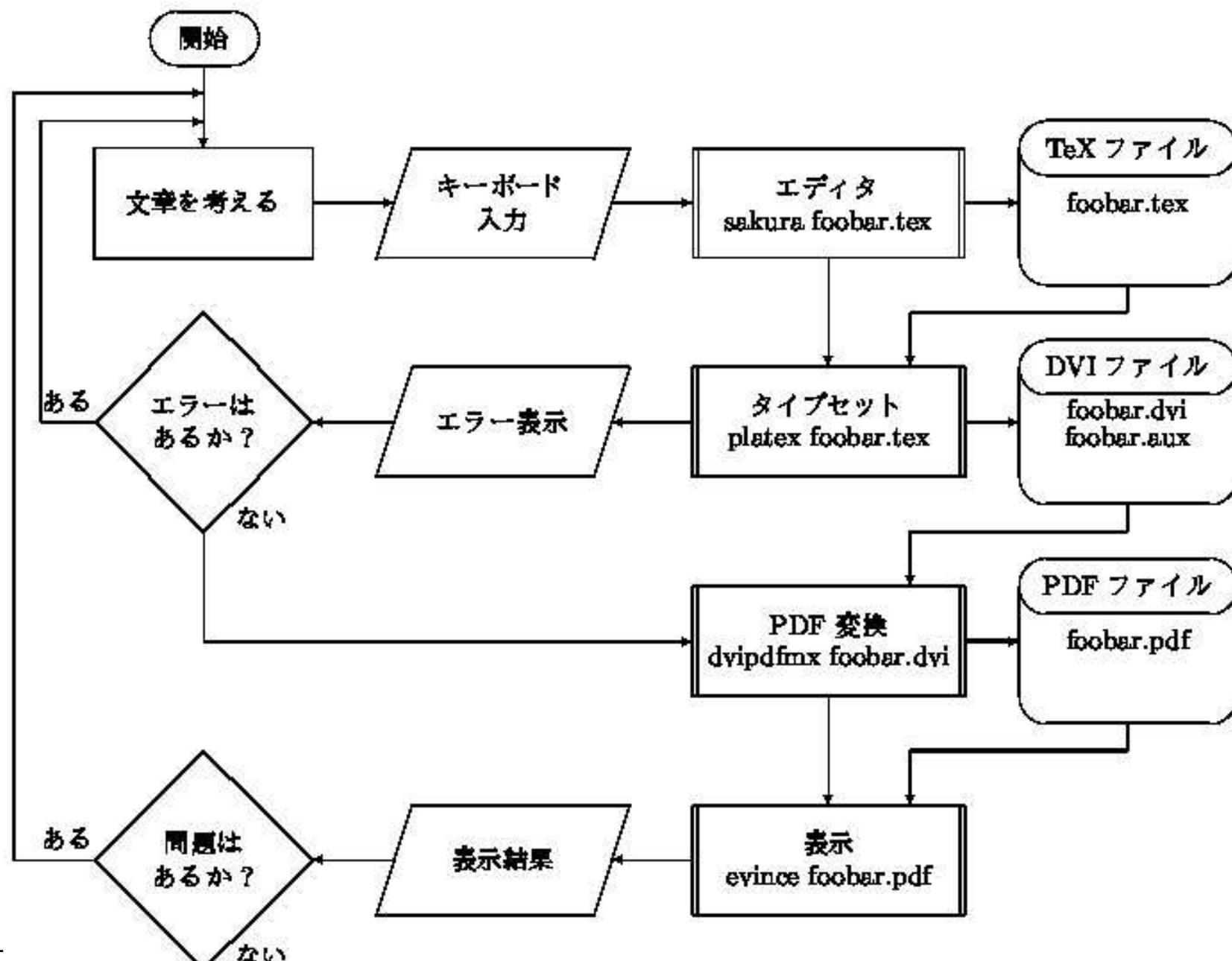
- ▶ base.pdf が出来ている事を確認する

- base.pdf を開いて、内容を確認する

- ▶ 普段は「evince base.pdf」とする

- ▶ Windows から、「開いて」もよい

TeX 文章の作成フロー



実習 4: 提出課題の準備

□ [実習 4-1] exp-001.pdf を読む

- exp-001.tex をダウンロードする (c:\usr\tex\20151020 下へ)
- exp-001.tex をタイプセットする
 - ▶ タイプセットが上手く行かなければ、exp-001.pdf をダウンロードしてもよい
- exp-001.pdf を表示させ、中身を読む

□ [実習 4-2] mybase.tex の作成

- base.tex の中身を変更し mybase.tex を作成する
 - ▶ copy base.tex mybase.tex
 - ▶ sakura mybase.tex
 - ▶ 自分の名前や学生番号が表示されるようにする
- mybase.tex をタイプセット([実習 3] 参照)
 - ▶ mybase.pdf が作られる事を確認する
 - ▶ 表示させて、きちんと自分の変更が反映されている事を確認する
- 「platex mybase.tex」で上手く行かなかつたら..
 - ▶ 基本は、「始めからやり直す」事を試す
 - ▶ 作成したファイルと、表示されたメッセージを skype に貼って質問する

実習 5: Mathematica と TeX

- Mathematica で計算した式を TeX で利用する
 - Mathematica の計算結果：そのままでは TeX で利用できない
 - TeXForm 関数を利用すると、式を TeX で利用できる形になる
- TeX で Mathematica の結果を利用する手順
 - Mathematica で式の計算を行う
 - TeXForm 関数で計算結果を表示させる
 - TeX ファイルに、表示結果を Copy&Past
 - 用途に応じて「 \sim 」(文中)か、「 $\[\sim \]$ 」(独立行)で挟む
 - ▷ 文中で分数を使う場合は、「 \displaystyle 」を先行させる

実習 6: 課題

□ [実習 6-1] sample-20151020.tex の表示

- sample-20151020.tex をダウンロードする
- sample-20151020.tex をタイプセットする

□ [実習 6-2] 課題の作成

- 20151020-QQQQ.tex を作る
 - ▶ mybase.tex を変更して新規に作成する
- 色々な数式を記述して、表示させてみる
 - ▶ exp-001.tex / sample-20151020.tex を参考にする
- この結果を CST Portal に提出する

□ [実習 6-3] 試してみる

- 「TeX 数式」などで検索する
 - ▶ Web ページの内容を参考に様々な数式を試してみる
- 「面白そうな数式」を互いにチャットで交換する
 - ▶ 「TeX でこう入力するとどうなるか？」クイズ

蘊蓄

□ TeX と LaTeX

○ TeX : クヌースが作成したシステム

- ▶ 自分自身が数式が綺麗に出力できるシステムが欲しかった..

○ LaTeX : TeX にランポートが機能を追加

- ▶ 使い易くなっている
- ▶ コンピュータ概論では基本、こちらを学ぶ

□ tex/dvi/pdf

○ tex : TeX/LaTeX で記述された文章ファイル

- ▶ 基本は、テキストファイル : エディターで編集する

○ dvi : ptex/platex コマンドで tex ファイルを変換したもの

- ▶ dviout で表示 / divprt で印刷できる形式
- ▶ 表示できる環境が少ない

○ pdf : dvipdfmx で dvi ファイルから作成

- ▶ 今の PC なら、どの場合でも表示可能
- ▶ 配布や印刷は pdf 形式で行う事が多い

TeX で数独

□ ファイルの入手

○ 次の三つのファイルを Download

▷ `sudokubundle.dtx`, `sudokubundle.ins`, `sudokuc.tex`

□ タイプセット

○ `sudokubundle.ins` を typeset

▷ `platex sudokubundle.ins`

○ `sudokuc.tex` をエディタで編集

▷ `\setsudrandom{1001}` の 1001 の部分を色々変えてみる

▷ `sakura sudokuc.tex`

○ `sudokuc.tex` を typeset

▷ `platex sudokuc.tex`