情報処理 2 資料

TEX メモ

桂田 祐史 (かつらだ まさし)

2007 年 6 月 19 日

この授業用の WWW ページは http://www.math.meiji.ac.jp/~mk/syori2-2007/

システム管理課の方達のご尽力により、0508 号室のパソコンの Windows XP 環境に、TEX 関係のソフトウェア一式がインストールされました。学期中のため相当の手間がかかったはずです。どうもありがとうございます。

1 TEX とは？

(駆け足で説明する。)

TEX は組版ソフトである

TEX は著名なコンピューター科学者であるドナルド・クヌース (Donald Knuth[1], ウィキペディアにも載っています) の開発した文書整形システム (組版[2]システム) です。TEX は日本では「てっく」または「てふ」と呼ばれています[3]。

当初、数式を含む英文を清書することを目的に、従来の組版技術の歴史を入念に調べた上で、それをコンピューター上で実現することを目標に開発されたそうです。

(ワープロとの違いを説明すべきだけど...)

TEX はフリーソフトである

Knuth 自身は TEX に関する情報を完全に公開していて (書籍になっています)、ソフトウェアを無償で利用することができます。また、多くのボランティアの活動により、TEX を補助、発展させるためのソフトウェア、データもほとんどは無償で利用可能です。例えば、TEX 本

[1] http://www-cs-faculty.stanford.edu/~knuth/

[2] 組版とは、文字や図版などの要素を配置して、紙面を構成することで、もともとは活版印刷において、活字を組み上げることから来ている。

[3] 「てっくす」とは読みません。ちなみに TEX の解説書に “Joy of TEX” という本があって、それは英語圏の国では有名な本のパロディだったそうです。昔、テレビで深夜映画を見ていたら、元ネタの本が出て来て、思わず見入ってしまいました。

体や周辺ソフトウェアの C 言語への変換、表示用ドライバー、印刷用ドライバー、PDF への変換ソフトウェア、日本語対応、日本語フォント (やそれを利用する仕組み)、Windows 環境への移植、インストーラーなど。これら成果物はすべてインターネットから無償で入手できます。

TEX は数学の世界では標準である

数学者村では標準の文書作成ソフトウェアです。理工系の多くの分野で利用されていますが、それだけでなく文系の研究者が利用した例もあります (発音記号や、ややマイナーな言語などを扱う場合)。

TEX で高品位の文書が作成できる

組版技術をしっかり研究した上で作られたものであるため、高品質な仕上がりが得られます。異なる環境下での再現性も抜群です。英語圏ではもちろん、日本でも理工系の多くの書籍 (中学高校の教科書や問題集なども含む) で採用されています。

TEX で作った文書は PDF にして配布が楽々

それ自身は文書の配布フォーマットとして適当ではありませんが (表示、印刷に専用のソフトウェアが必要なためです)、簡単に **PDF** (portable document format) に変換できるので、配布に際して問題はありません。

この授業では、TEX の一種である **LATEX** (正確にはその日本語対応版 pLATEX) を使ってもらうことにします。

2 Windows 環境での利用法 (1) 基本

“コマンドプロンプト” の中でキーボードからコマンドを打ち込むという、最も基本的な (原始的な？) 使い方を説明します。

(TEX は複数のプログラムを利用して目的を達成するソフトウェアです。完全なブラックボックスにしてしまうと「わけが分からなく」なるので、個々のプログラムが目に見える「コマンドによる操作」に慣れてもらいます。後でより便利な方法を紹介しますが、基本的な方法は、他の環境 (Linux, Mac, ...) に行ったときも応用が効きますし、トラブルが発生したときに自分で解決する力をつけられる可能性も高くなります。)

2.1 コマンド・プロンプトの準備

1. まずコマンドプロンプトのショートカットを用意しましょう。情報処理 2 用の作業フォルダ (マイドキュメントの中に “syori2” のような名前で作成しました) を開いてから、

[スタート] → [すべてのプログラム (P)] を開き[4]、[コマンドプロンプト] をマウスの右ボタンを押しながら作業フォルダまでドラッグしてコピーします[5]。

2. 試しにショートカットのアイコンをダブルクリックして起動すると、Z:¥> のようなコマンドプロンプトが現われるはずです。試しに

```
(以下の “syori2” は自分で選んだ名前に置き換えて下さい。なお、コマンドプロンプトで日本語を入力するには、Alt-漢字 とします。)

Z:\>dir
  (色々表示される)
Z:\>cd .windows2000
  (マイドキュメントに移動する)
Z:\.windows2000>dir
  (マイドキュメント内のファイルの名前が表示されるはず)
Z:\.windows2000>cd syori2
  (マイドキュメント内のフォルダ syori2 に移動する)
Z:\.windows2000\syori2>dir
  (syori2 内のファイルの名前が表示されるはず)
Z:\.windows2000\syori2>type first.BAS
  (BASIC プログラム first.BAS があったとして、それを表示してみる。)
```

3. 今後の TEX に関する作業は、マイドキュメント内の作業フォルダで行うのがよいので、コマンドプロンプトを実行した直後に Z:¥.windows2000¥syori2 が作業フォルダになるように設定します。具体的には、[コマンドプロンプト] のアイコンを右クリックして現われるメニューの [プロパティ] を選んで、「作業フォルダ」を “%HOME%%HOMEPATH%” から “%HOME%%HOMEPATH%¥.windows2000¥syori2” に変更します。もちろん “syori2” の部分は自分で選んだ名前で置き換えてください。OK としてから、アイコンをダブルクリックして最初から

```
Z:\.windows2000\syori2>
```

となったら成功です。

2.2 秀丸で .tex ファイルを作成する

1. 秀丸エディタを起動して、次のような内容を入力してください。

[4] [すべてのプログラム (P)] に [コマンドプロンプト] があるのは、情報処理教室のパソコンの特殊事情です。普通は [すべてのプログラム (P)] → [アクセサリ] → [コマンドプロンプト] と辿ります。

[5] [コマンドプロンプト] を右クリックして、[送る] で [マイドキュメント] を選択すると、マイドキュメント] にショートカットが作れるので、それを移動しても良いでしょう。

tamago.tex

```
\documentclass[12pt,leqno]{jarticle}
\begin{document}
\end{document}
```

これを tamago.tex という名前で保存しましょう。そのためには、[ファイル] → [名前をつけて保存 (A)] で、

- 「保存する場所 (I)」をマイドキュメント内の作業フォルダ (syori2) にして、
- 「ファイル名 (N)」を “tamago.tex” にして、
- 「ファイルの種類 (T)」を「すべてのファイル (*.*)」に

します。

注意「ファイルの種類 (T)」をデフォールトの「テキストファイル (*.txt)」のままで保存すると、環境によっては、“tamago.tex.txt” というファイルが出来てしまいます。これは使えません。

2. 一度秀丸エディタを終了して、作業フォルダ (syori2) 内の tamago.tex のアイコンをダブルクリックして起動してみましょう。少し書き足してみます。

first.tex

```
\documentclass[12pt,leqno]{jarticle}
\begin{document}
こんにちは。

\[
 \int_{-\infty}^\infty e^{-x^2}\;d x=\sqrt{\pi}.
\]
\end{document}
```

これを first.tex という名前をつけて保存しましょう。[ファイル] → [名前をつけて保存 (A)] とすると、「ファイルの種類 (T)」は「その他 (*.TEX)」となっているはずです。「ファイル名 (N)」のところだけ first.tex として OK です。

3. 今後は、手順 2 の部分、つまり tamago.tex を開いてから、「名前をつけて保存 (A)」で名前を決める、というやり方で作業をするのが私のお勧めです。

2.3 .tex ファイルから .dvi ファイルを作る

1. 作業フォルダ内のコマンドプロンプトのアイコンをダブルクリックして、コマンドプロンプトを開きます。

2. キーボードから platex first.tex と入力し、最後に Enter を打ちます (最後の .tex は省略できるかも知れません)。

platex first.tex Enter

うまく行けば次のように次のコマンド待ちの状態になるはずです。

```
Z:\.windows2000\syori2>platex first.tex
  (色々表示が出る)
Z:\.windows2000\syori2>
```

first.dvi というファイルが出来ているはずです。

3. first.tex の内容に打ち間違いがあった場合、エラーになり、platex コマンドはどうすればよいか、? というプロンプトを出して尋ねてきます。それに対しては、x Enter (即終了) あるいは q Enter (バッチモードでとにかく最後まで処理) として、終了させてください (それぞれ eXit, Quit ということでしょう)。エラーメッセージを頼りに first.tex を修正してから (上書き保存 C-s を忘れずに)、2 に戻ります (platex first.tex とする)。

2.4 .dvi ファイルを表示する/印刷する

1. コマンドプロンプトから

```
Z:\.windows2000\syori2>dviout first.dvi
```

と打ちます (最後の .dvi は省略できるかも知れません)。新しいウィンドウが現われ、.dvi ファイルが表示されるはずです。

2. 作業中は dviout を終了する必要はあまりないでしょう。first.tex を書き換えて、platex で処理し直しせば、dviout は新しい内容を表示してくれます。

3. 印刷したい場合は dviout のメニューから行えばよいでしょう。

2.5 .dvi ファイルから .pdf ファイルを作る

課題のレポートとして .dvi ファイルを送ってもらうので十分なこともあるのですが、.dvi ファイルには図のデータを含めることが出来ないことと、.dvi ファイルが読めない人も多い (dviout はどこにでもインストールされているものではない) ので、人に渡すときは、PDF (portable document format) に変換したものを渡すのがお勧めです。

それには dvipdfm あるいは dvipdfmx というコマンドを用います。

```
Z:\.windows2000\syori2>dvipdfm first.dvi
```

または

```
Z:\.windows2000\syori2>dvipdfmx first.dvi
```

で first.dvi から first.pdf が作成されます。

出来上がった first.pdf は Adobe Reader 等で表示・印刷が可能です。
なお、Adobe Reader で読んでいる間は、first.pdf を変更することができません。dvipdfm (dvipdfmx) を実行する時は、一旦 Adobe Reader を終了して下さい。

2.6 まとめ

2.6.1 TEX のための準備作業

- マイドキュメントの下の授業用のフォルダ (“soyri2” とか “処理 2” という名前の人が多い— 以下 syori2 として説明する) の中にコマンド・プロンプトへのショートカットを作り、その「作業フォルダ」をその syori2 にした。
→ 参考http://www.math.meiji.ac.jp/~mk/syori2-2007/jouhousyori2-2007-06/node9.html

- LATEX 文書の雛形となる、tamago.tex という名前のファイルを syori2 の中に作成した。
→ 参考http://www.math.meiji.ac.jp/~mk/syori2-2007/jouhousyori2-2007-06/node10.html

2.6.2 こうやって LATEX で文書を作る ~ 作業の流れ

1. tamago.tex を (秀丸などのテキスト・エディターで) 開いて、[名前をつけて保存 (A)] で、適当な名前に変えて保存してから、編集 (執筆？) を始める。

2. なんとか.tex を dvi ファイルに変換 (コンパイル?) するには、コマンド・プロンプトで、

   platex なんとか.tex Enter

   とする。

3. なんとか.dvi を表示 (プリビュー) するには、コマンド・プロンプトで

   dvioout なんとか.dvi Enter

   とする。印刷は dviout のメニューから行う。

4. 人に渡す場合は pdf ファイルにするとよい。なんとか.dvi を pdf ファイルに変換するには、コマンド・プロンプトで

   dvipdfm なんとか.dvi Enter

   または

```
dvipdfmx なんとか.dvi Enter
```

とする。

3 Windows 環境での利用法 (2) 祝鳥

祝鳥[6] [7]という、LaTeX の統合環境を提供する秀丸マクロを情報処理教室用に用意しました。試しに使ってみましょう。

3.1 準備作業 (一度だけすれば良い)

1. http://www.math.meiji.ac.jp/~mk/syori2-2007/syori2texenv.zip を適当な場所 (例えばデスクトップ) に保存する。
2. マイドキュメント・フォルダを開く。
3. syori2texenv.zip を (ダブルクリックして) 開き、その中にある macro というフォルダーをマイドキュメント・フォルダに (ドラッグして) コピーする。また hidemarutex.bat を LaTeX 作業をするのに便利な場所、例えばマイドキュメント・フォルダの下の syori2 フォルダに (ドラッグして) コピーして、一度実行しておく。(おまけ: verbatimfiles.sty, wjpeg2ps.exe も syori2 にコピーするとよい。)
4. (初期設定をする) 何か LaTeX 文書 (例えば tamago.tex) を秀丸で開いて、C-T を入力し、後は Enter を (約 10 回) 打つ。

3.2 使い方

- ログオンする毎に一度だけ hidemarutex.bat を実行する[8]。
- LaTeX 文書を秀丸で編集中に C-] を入力すると

DVI に変換 (T)
DVI に変換して表示 (P)
PS に変換して表示
PDF に変換して表示
PS 経由で PDF に変換して表示
MakeIndex
BibTeX
設定 (S)

[6] http://www.ms.u-tokyo.ac.jp/~abenori/mycreate/fortex.html
[7] 作者は東京大学数理科学研究科の阿部紀行氏です。
[8] 出来ればこれを省略できるとよいのですが、情報処理教室の Windows 環境ではできないようです (レジストリが毎回クリアされてしまうので)。

というメニューが現われる。例えば 'P' と入力すると platex を呼び出して、.dvi ファイルを作った上で dviout を起動して表示してくれる。

- 祝鳥の使い方は「祝鳥ヘルプ」[9] で調べられる。
- 「マクロ」メニューをチェックしておくことを勧める。特に C-T で出て来るメニューや、コマンドを途中まで打ってから C-H で補完する、環境を閉じる C-T, C-E など...

4 LaTeX 文書 .tex の書き方 — 入門

- 円記号 ¥ もバックスラッシュ \ も、LaTeX のコマンドにとっては同じもの。
- パーセント記号 % から行末までは注釈になる。
- いつでも書く必要がありそうな次の内容は tamago.tex に書いておいた。

tamago.tex

```
\documentclass[12pt,leqno]{jarticle}
\begin{document}
\end{document}
```

- \begin{document} から \end{document} までの間に、ローマ字、数字などの "フツーの字" で書くとそのまま文書に入力される。いわゆる記号は注意が必要である。
- タイトルをつけるには、

```
\title{はじめての \TeX}% タイトル
\author{桂田祐史}       % 著者名
\date{2007 年 5 月 22 日}    % 日付（省略すると組版した日になる）
\maketitle              % これでタイトルを表示する
```

[9] http://www.ms.u-tokyo.ac.jp/~abenori/mycreate/chm/fortex/fortex.html

簡単なサンプル

```
\documentclass[12pt]{jarticle}% スタイルの指定
\begin{document}
% この行は注釈。次の 4 行でタイトル、著者名、日付を表示する
\title{レポート課題 X}
\author{2 年 16 組 99 番  桂田  祐史}
\date{2007 年 4 月 26 日}
\maketitle

  ここにフツーの文字で書いたものは出力される。

\[
 \int_{-\infty}^\infty e^{-x^2}\;d x=\sqrt{\pi}.
\]

\end{document}
```

細かい話: (ローマ字数字以外の) 記号について

まず、そのまま入力＆表示できるものとして、

! " ’ ( ) - = ‘ @ [ ] + * ; : ? , .

がある (マイナス - は、1 文字の場合、2 文字連続の場合、3 文字連続の場合で - – — となるので、そのまま入力できるものと考えるべきではないかもしれない。もっとも通常、マイナスは数式中に現われるものだから$–$ と書くものかも。)。

一方、シャープ # などは、そのまま入力したのではダメで、これはバックスラッシュ \ を前につけて \# と入力する必要がある (\ でエスケープする、という)。同様にする必要がある文字としては、

$ % & _ { }

$ と $ で囲んで、数式モードで扱うべき文字としては

| < > −

がある。

難しいのは次の 3 文字で、これは専用のコマンドが用意されています。

~ \textasciitilde
\ \textbackslash
^ \textasciicircum

日本人専用の応急処置として、難しい文字の入力には漢字を使うというのがある (やや幅広になってしまうが)。 # $ % & _ { }| < > ~ \ という感じで簡単。

改行と空白 (最低限)

意外と難しいので[10]、ある程度 TEX の説明が進んでから詳しく説明する。ここではごく基本的なことと、「予告」に止める。

- .tex の中にいくら空白を続けても、一つ空白を入れたのと同じで、小さな空きができるだけ。\quad などのコマンドがある。数式モードでは微調整用のコマンドがたくさんある。
- 連続した改行は「空行」と呼び、パラグラフ (段落？) を変更するという意味になり、改行されて、次の文の先頭に空きができる。連続した空行は 1 つの空行と同じことになる。
- 英文中の一つの改行 (連続していない) は、一つの空白と同等。日本語文中の一つの改行 (連続していない) は、無視される。(不正確な言い方だが...)
- 強制的な改行は \\ だが、初心者が使いたくなるケースの 90% は誤用である。

5 簡単な数式

5.1 数式モード

数式は「数式モード」の中で書く。数式モードには次の二つがある。

1. 文中の数式 (インライン数式) は、ドル記号 $ ではさんでかく。

   ピタゴラスの定理から `$a^2+b^2=c^2$` が成り立つ。

   ピタゴラスの定理から $a^2+b^2=c^2$ が成り立つ。

2. 数式のみの行 (ディスプレイ数式) を作るには、色々な命令があるが、もっとも基本的なものは、\[と \] ではさむもので、例えば

   ```
   ピタゴラスの定理から
   \[
     a^2+b^2=c^2
   \]
   がなりたつ。
   ```

   のようにすると

[10]少し大げさなようだが、TEX の設計思想に係わることなので。

ピタゴラスの定理から

$$a^2 + b^2 = c^2$$

がなりたつ。

となる。式番号をつけるには equation 環境というものを用いて、

ピタゴラスの定理から
```
\begin{equation}
  a^2+b^2=c^2
\end{equation}
```
がなりたつ。

のように書く。

5.2 かっこ

丸い括弧 $(,)$ とカギ括弧 $[,]$ は普通に入力できる。$\{,\}$ は前に $\backslash$ をつける。

```
\[
  \{[(a+b)+c]+d\}
\]
```

とすると

$$\{[(a+b)+c]+d\}$$

となる。

5.3 空白

数式モード中は、たくさんの空白用コマンドがある[11]。

```
\[
 a\,a\;a\ a\quad a\qquad a
\]
```

[11] quad (=quadrat) 印刷用語で空白の込め物 (広辞苑によると、「組版の際に、印刷する必要のない余白部を埋めるために組み込むもの。」だそうである。字と字の間に入れるのが「スペース」、大きな余白にいれるのが「クワタ」。) の一種。日本でも「クワタ」と呼ばれるそうである。個人的には、焼き鳥の「ハツ」 (heart) を思い出してしまう...

$$a\,a\,a\,a\quad a\qquad a$$

空白を詰めることも必要になる。\! で詰まる。

```
\[
  \int\int f(x,y)dxdy=\int\!\!\int f(x,y)dxdy
\]
```

とすると

$$\int\int f(x,y)dxdy=\int\!\!\int f(x,y)dxdy$$

となる。

(もっとも重積分、三重積分には、\dint, \tint というコマンドが用意されているので、出番は少なくなった ??)

5.4 色々な記号

5.4.1 ギリシャ文字

バックスラッシュ \ の後に、ローマ字で読み[12]を書くことでギリシャ文字が書ける。

```
\[
  \alpha\beta\gamma\delta\epsilon\zeta\eta\theta\iota\kappa\lambda\mu\nu\xi
  % omicron は o と字の形が同じなのでない
  \pi\rho\sigma\tau\upsilon\phi\chi\psi\omega
\]
```

とすると

$$\alpha\beta\gamma\delta\epsilon\zeta\eta\theta\iota\kappa\lambda\mu\nu\xi\pi\rho\sigma\tau\upsilon\phi\chi\psi\omega$$

となる。なお、

```
\[
  \varepsilon\vartheta\varpi\varrho\varsigma\varphi
\]
```

とすると、

[12]使っているうちに、よく出てくるものは覚えてしまうと思います。なお、祝鳥では、C-T, j, ? とすると一覧が出ます。

$$\varepsilon\vartheta\varpi\varrho\varsigma\varphi$$

大文字のギリシャ文字は、先頭のローマ字を大文字にすればよい。例えば

```
\[
  \Gamma  \Delta  \Theta  \Lambda  \Xi  \Pi  \Sigma  \Upsilon  \Phi
  \Psi  \Omega
\]
```

とすると

$$\Gamma\Delta\Theta\Lambda\Xi\Pi\Sigma\Upsilon\Phi\Psi\Omega$$

となる。イタリック $\mathit{\Gamma}$ にしたければ[13]、\mathit{\Gamma} のようにする。

$$\mathit{\Gamma\Delta\Theta\Lambda\Xi\Pi\Sigma\Upsilon\Phi\Psi\Omega}$$

5.4.2 矢印

```
\[
  \to \quad \mapsto \quad \leftarrow \quad \Leftarrow\quad
 \longleftarrow \quad \Longleftarrow
 \quad \leftrightarrow \quad \Leftrightarrow
 \quad \longleftrightarrow \quad \Longleftrightarrow
\]
```

とすると

$$\to \quad \mapsto \quad \leftarrow \quad \Leftarrow \quad \longleftarrow \quad \Longleftarrow \quad \leftrightarrow \quad \Leftrightarrow \quad \longleftrightarrow \quad \Longleftrightarrow$$

上下、斜めの矢印については、

[13] ギリシア文字の大文字をイタリックにするのは、もしかすると変なのかも知れないけれど、結構使われています。

```
\[
  \uparrow \quad \downarrow \quad \Uparrow \quad \Downarrow \quad \Updownarrow \quad \Downarrow \quad
  \updownarrow \quad \Updownarrow \quad
  \nearrow \quad \nwarrow \quad \searrow \quad \swarrow
\]
```

とすると

$$\uparrow \quad \downarrow \quad \Uparrow \quad \Downarrow \quad \Updownarrow \quad \Downarrow \quad \nearrow \quad \nwarrow \quad \searrow \quad \swarrow$$

5.4.3 点

```
\[
 \cdot \quad \cdots \quad \ldots \quad \ddots \quad \vdots
\]
```

は順に、真ん中に一つの点、真ん中に 3 つの点、下に 3 つの点、斜めに 3 つの点、垂直方向に 3 つの点

$$\cdot \quad \cdots \quad \ldots \quad \ddots \quad \vdots$$

5.4.4 集合

```
\[
  a\in A\subset B,\quad
  C\supset D,\quad
  a\not\in A,\quad
  C\not\supset D,\quad
  A\cup B, A\cap B, A\setminus B=\emptyset,\quad
  \bigcup_{i=1}^\infty A_i=\bigcap_{i=1}^\infty B_i
\]
```

$$a\in A\subset B,\quad C\supset D,\quad a\not\in A,\quad C\not\supset D,\quad A\cup B, A\cap B, A\setminus B=\emptyset,\quad \bigcup_{i=1}^\infty A_i=\bigcap_{i=1}^\infty B_i$$

5.4.5　不等式

等号のつかないものはそのまま $<, >$ を使うとよい。$\leq$ は \le とし、$\geq$ は \ge とする[14]。

```
\[
 a<b\le c\ge d
\]
```

$$a < b \leq c \geq d$$

なお \ll, \gg で $\ll, \gg$ となる。また、等号 $=$ の否定 $\neq$ は \ne と入力する。

5.4.6　その他

$$\| \quad \pm \quad \mp \quad \times \quad \div \quad \sim \quad \simeq \quad \fallingdotseq \quad \nabla \quad \triangle \quad \partial \quad \forall \quad \exists \quad \infty \quad \propto \quad \angle \quad \langle \quad \rangle$$

を表示するには、次のようにします (\fallingdotseq $\fallingdotseq$ のような AMS (アメリカ数学会) 由来のフォントには、プリアンブルに \usepackage{amssymb} と書くことが必要です)。

```
...
\usepackage{amssymb}% AMS で用意したシンボルのフォント
...
\begin{document}
...
\[
  \| \quad \pm\quad \mp \quad \times \quad \div \quad
 \sim \quad \simeq \quad \fallingdotseq \quad \nabla \quad \triangle \quad
 \partial \quad
 \forall \quad \exists \quad
 \infty \quad \propto\quad
 \angle\quad \langle\quad \rangle
\]
...
```

5.5　上つき添字、下つき添字

a^2 は a^2 とする。a_n は a_n とする。2^{2^n} は 2^{2^n} とする。
積分やシグマなどもこの応用で、

[14] 多分、“less than or equal to” から le, “greater than or equal to” から ge となったのであろう。

```
\[
  \lim_{R\to\infty}\int_a^R f(x)dx=\sum_{n=1}^\infty a_n
\]
```

とすると

$$\lim_{R\to\infty}\int_a^R f(x)dx=\sum_{n=1}^\infty a_n$$

5.6 分数

```
\[
  \frac{a+b}{c}=\frac12
\]
```

は

$$\frac{a+b}{c}=\frac12$$

となる。

分数や積分、和の記号など、インライン数式では小さく組版されるが、ディスプレイ数式と同じように大きく組版するには、\displaystle コマンドを用いる。

```
$\frac{a+b}{c}=\frac12$ は小さいので、
$\displaystyle\frac{a+b}{c}=\frac12$ とすると大きくなる。
```

は

$\frac{a+b}{c}=\frac12$ は小さいので、$\displaystyle\frac{a+b}{c}=\frac12$ とすると大きくなる。

実は \dfrac という命令もあります。

5.7 sin などの「作用素」

単に log x のように書くと $logx$ となってしまう (これでは l, o, g, x の積にしか見えない)。専用のコマンド \log が用意されている (l,o,g を立体 (roman face) で表示し、小さな空白を作ることに注目)。

```
\[
 \sin x,\log x,\max A
\]
```

$$\sin x, \log x, \max A$$

(細かい注意) なお、後で説明するマクロ機能を使って、自分で定義することもできる。

```
\newcommand{\grad}{\mathop{\rm grad}\nolimits}
```

をプリアンブル (\begin{document} の前) に書いておくと、\grad というコマンドが使えるようになる。

5.8 行列、ベクトル

行列や (縦) ベクトルでは、式 (成分) を「きれいに並べる」必要がある。このためには、array 環境や matrix 環境を用いる (縦ベクトルは、列の個数が 1 である行列とみなす)。また括弧 (,) は \left と \right を使って拡大する。

$$\left(\begin{array}{cc} a & b \\ c & d \end{array} \right) \left(\begin{array}{c} x \\ y \end{array} \right)$$

は

array 環境を用いて行列を書く

```
\[
 \left(
  \begin{array}{cc}
     a & b \\
     c & d
  \end{array}
 \right)
 \left(
  \begin{array}{c}
     x \\
     y
  \end{array}
 \right)
\]
```

または

matrix 環境を用いて行列を書く

```
\usepackage{amsmath}
...
 \left(
  \begin{matrix}
     a & b \\
     c & d
  \end{matrix}
 \right)
 \left(
  \begin{matrix}
     x \\
     y
  \end{matrix}
 \right)
\]
```

matrix 環境の方が使い方は簡単だが[15]、array 環境は左寄せ (l)、中央揃え (c)、右寄せ (r) など細かい制御ができる。

なお

[15]なお、括弧つきの行列を作る pmatrix, bmatrix, Bmatrix 環境等もある。

$$|x| = \begin{cases} x & (x \geq 0 \text{ のとき}) \\ -x & (x < 0 \text{ のとき}) \end{cases}$$

も似た感じで出力できる。

```
\[
 |x|=
 \left\{
 \begin{array}{rl}
   x & \mbox{($x\ge 0$ のとき)}\\
  -x & \mbox{($x<0$ のとき)}
 \end{array}
 \right. % 右側は括弧なし
\]
```

6　外部からのファイルの取り込み

例えば十進 BASIC の課題のレポートを作る場合など、ソースプログラムや画像イメージなどを取り込みたくなります。

6.1　ソースプログラム等テキストファイルの取り込み

短いものは

verbatim 環境の利用

```
\begin{verbatim}
PRINT "HELLO"
END
\end{verbatim}
```

のように、.tex ファイルの中の、verbatim (言葉通りに、逐語的に、という意味) 環境の中に入れてしまえばよいですが、長いものや頻繁に変更を加えるものを扱うは面倒です。verbatimfiles パッケージを組み込むと有効になる \verbatimfile コマンドや \verbatimlisting コマンド (行番号つき) を使うとよいでしょう。

KADAI7.BAS を取り込む

```
\documentclass[12pt,leqno]{jarticle}
\usepackage{verbatimfiles}% パッケージを組み込む（複数形の s がついている）

\begin{document}
...
\verbatimfile{KADAI7.BAS}% KADAI7.BAS は別途用意してあるとして
...
\end{document}
```

6.2 画像の取り込み

LaTeX には EPS 形式で取り込むのが便利

画像ファイルには色々なフォーマットがありますが、LaTeX に取り込むには、カプセル化 **PostScript** 形式 (Encapsulated PostScript, 長いので EPS 形式と呼ぶことにします, 通常は、ファイル名に ".eps" という拡張子をつけます) に変換してから、\includegraphics 命令で取り込むのが簡単で問題が生じにくいです。

十進 BASIC の画像を EPS 形式で作る

十進 BASIC のグラフィックスの場合、「名前をつけて保存 (A)」から JPEG 形式で (ファイル名拡張子は ".JPG") 保存しておき、**jpeg2ps**[16] コマンドで EPS 形式に変換するのが便利です。今回は Windows XP の GUI で使える wjpeg2ps[17] を紹介します。前回、指示した通りに操作していれば、syori2 フォルダにアイコンがあるはずです。使い方は簡単で、JPEG ファイルを Wjpeg2ps のアイコンにドラッグして、Convert ボタンを押すだけです。それで EPS 形式のファイルが出来ます。

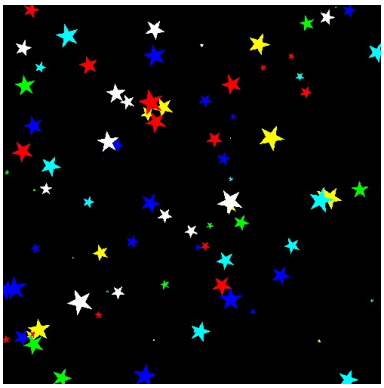
仕組みについて、もう少し説明が読みたければ、「イメージデータの TeX への取り込み — jpeg2ps のすすめ」[18] を見てください。

kamehoshi2.eps を取り込む

```
\documentclass[12pt,leqno]{jarticle}
\usepackage[dviout]{graphicx}% graphicx パッケージを用いる

\begin{document}
\begin{center}
 \includegraphics[width=5cm]{kamehoshi2.eps}% データは各自用意してください。
\end{center}
\end{document}
```

[16] http://www.pdflib.com/
[17] http://www.vector.co.jp/soft/dl/win95/art/se248407.html
[18] http://www.math.meiji.ac.jp/~mk/labo/howto/jpeg2ps.html



dviout でカラー表示・印刷をするには

カラーで表示・印刷するには、dviout で Option → Setup Parameters → Graphic で、GIF の取り扱いの設定で BMP(full-color) を選択します。dviout 起動時に -GIF=5 としても良いです。本当は、情報処理教室のデフォールトの設定にしておくべきだったかも知れません (すみません、今年度はもう直せません)。

余談: ウィンドウの画像を取り込む

Windows XP のウィンドウの画像をファイルに保存したければ、マウスカーソルを取り込みたいウィンドウに置いて、キーボードから Alt+Print Screen (Print Screen は、場合によっては Fn キーと一緒に押す必要があり、その場合は Alt+Fn+PrintScreen となる) を入力し、ペイント[19]のようなソフトに貼付けてから、適当に編集した後で、保存すると良いでしょう (もちろん JPEG 形式に出来ます)。

A 参考: Linux 環境での LaTeX の使い方

TeX は UNIX の世界での利用実績が長いので、UNIX 環境では使いやすくなっている、と言えるでしょう。情報処理教室の Linux 環境ではどうすれば使えるか、簡単に説明します。

まず emacs のようなエディターでファイル名末尾が ".tex" であるファイルを作ります。例えば "first.tex" であれば、

```
icr3-1008% emacs first.tex &
```

とようにすれば OK です。

YaTeX (野鳥) のような便利なソフトもありますが、それはまたの機会に紹介するとして、ここでは基本的な TeX 関係のコマンドの使い方を紹介します。

.tex ファイルを .dvi に変換するために platex コマンドを使うのは、Windows の場合と同じです[20]。

[19] スタート → すべてのプログラム (P) → アクセサリ → ペイント として起動できます。

[20] 現在、日本語が使える LaTeX としては、pLaTeX が標準的で、コマンドの名前は普通 platex である。

.tex を .dvi に変換 (「コンパイル」)

```
icr3-1008% platex first.tex
```

“first.tex” に (LaTeX の文法上の) 誤りがなければ、“first.dvi” という名前のファイルができます。

画面に (組版結果を) 表示する (「プリヴューする」という) には pxdvi コマンドを用います[21]。

.dvi ファイルをプリビュー (画面表示)

```
icr3-1008% pxdvi first.dvi &
```

印刷するには、情報処理教室の場合、例えば次のようにします[22]。

.dvi ファイルを印刷

```
icr3-1008% pdvips -f first.dvi | lp -d プリンター名
```

(両面印刷機能を持ったプリンターでは、途中で適当なフィルターを挟むと両面で印刷されるようになります。例えば double[23] を用いて、pdvips -f first.dvi | double | lpr で両面印刷できます。)

dvi ファイルを PDF に変換する方法については後日説明します (dvipdfm がありますが、そのままでは日本語うまく行かないみたいです)。

参考情報

[1] 奥村晴彦, 日本語 TeX 情報, http://oku.edu.mie-u.ac.jp/~okumura/texfaq/
最も有名な情報源。ここの掲示板は最後の頼みの綱。

[2] 奥村晴彦, LaTeX 2_ε 美文書作成入門 改訂第 4 版, 技術評論社 (2007).

[3] 阿部紀行, (Windows 環境向けの TeX 関係のソフトウェアのインストーラー), http://www.ms.u-tokyo.ac.jp/~abenori/mycreate/index.html

[4] 阿部紀行, 祝鳥 (のりてふ) — 秀丸エディタ用の TeX 統合環境, http://www.ms.u-tokyo.ac.jp/~abenori/mycreate/fortex.html

[5] 広瀬 雄二, 野鳥 (やてふ) — Emacs 用の TeX 統合環境, http://www.yatex.org/

[21] UNIX 環境ではプリヴューアーとして xdvi が定番で、pxdvi はその派生物らしい。なお、Windows 環境では、dviout というソフトが定番である。

[22] dvi ファイルを PostScript に変換するためのソフトウェアとしては dvips が定番で、pdvips はその派生物であるらしい。

[23] http://www.math.meiji.ac.jp/~mk/syori2-2007/double