

---

## 第2回 $\text{\TeX}$ ゼミ

小椋 信弥, 吉田 昌太, 福永 隆宏

2001年4月14日(土曜4限)

---

# 目次

<b>第 1 章</b>	<b>L<sup>A</sup>T<sub>E</sub>X の基礎</b>	<b>2</b>
1.1	L <sup>A</sup> T <sub>E</sub> X での約束事	2
1.1.1	ファイル名	2
1.1.2	最低限のルール	2
1.2	環境と命令	2
1.3	空白・改行・段落	3
1.3.1	自然な空白	3
1.3.2	段落の区切り	4
1.3.3	書いたまま出力する方法	4
<b>第 2 章</b>	<b>基本的な文書の編集</b>	<b>5</b>
2.1	文字の修飾	5
2.1.1	文字サイズの変更	5
2.1.2	書体の変更	5
2.2	改ページと空白の開け方	6
2.2.1	改ページ	6
2.2.2	空白の開け方	6
2.3	脚注	6
<b>第 3 章</b>	<b>画像の貼り付け</b>	<b>7</b>
3.1	画像の EPS への変換	7
3.2	Ghostscript のインストール	7
3.3	画像の貼り付け	10
<b>第 4 章</b>	<b>DVIOUT のバージョンアップ</b>	<b>12</b>
<b>第 5 章</b>	<b>課題 2</b>	<b>13</b>

# 第1章 L<sup>A</sup>T<sub>E</sub>Xの基礎

## 1.1 L<sup>A</sup>T<sub>E</sub>Xでの約束事

前回のゼミで、L<sup>A</sup>T<sub>E</sub>X環境をインストールする手順を説明しました。今回は実際にL<sup>A</sup>T<sub>E</sub>Xの文章を書くために必要となる知識と、L<sup>A</sup>T<sub>E</sub>Xによる文書作成の基本的な手順について説明します。

### 1.1.1 ファイル名

L<sup>A</sup>T<sub>E</sub>Xに限らず、T<sub>E</sub>Xでは、原稿ファイルの拡張子を”.tex”にするという約束があります。実際は”.tex”以外でも別にかまわないのですが、あえて約束違反をする必要もないでしょうし、ファイルの種類がわかりやすいこともありますので、必ず”.tex”にして下さい。

### 1.1.2 最低限のルール

1. T<sub>E</sub>XやL<sup>A</sup>T<sub>E</sub>Xの組版命令は、原則として” $\%$ ”で始まり、原則として半角空白で終わる。

T<sub>E</sub>XやL<sup>A</sup>T<sub>E</sub>Xの組版命令はそのほとんどが” $\%$ ”で始まります。組版命令に続く文字が、全角の空白や句読点、全角及び半角の括弧や記号(@など)、半角の数字、T<sub>E</sub>XやL<sup>A</sup>T<sub>E</sub>Xの命令であれば、半角の空白は必要ありません。

2. 半角の空白はいくつ続いてても1つとみなされる。

” $\%$  TeX ”と書いた場合も” $\%$  TeX ”と書いた場合も出力上は同じ結果となります。なお、改行直後(行頭)の半角空白は無視されます。

3. 全角文字の直後で改行すると改行は無視され、半角文字の直後で改行すると改行は空白とみなされる。また、2つ以上続いた改行(改行コードのみの行)は、段落の切れ目として扱われる。

「空白」のところで詳しく説明します。

4. 半角の記号文字の中には、そのまま入力しても出力できない特殊な文字がある。次の半角記号は、T<sub>E</sub>XやL<sup>A</sup>T<sub>E</sub>Xではそのまま入力しても出力できません。

#, \$, %, &, -, {, }, <, >, \, |, ^, ~

これらの記号を出力する一番手軽な方法は、全角文字を使用することです。全角文字はT<sub>E</sub>Xの内部で特殊な意味を持っていませんので、自由に使用することができます。

5. 半角のカタカナは使用できない

T<sub>E</sub>XやL<sup>A</sup>T<sub>E</sub>Xでは半角のカタカナを入力してもエラーにはなりませんが、それを出力できるDVIドライバがありません。

6. ”% ”以降は、改行コードを含めてコメントとみなされる。

行中の”% ”記号以下は改行コードも含めてコメントとして扱われます。このため、実際の実原稿ファイルでは改行しているのに、改行していないのと同様に処理したい場合などには”% ”記号を利用して改行を行います。

## 1.2 環境と命令

L<sup>A</sup>T<sub>E</sub>Xの最大の特徴は、論理デザインが可能であるということです。論理デザインでは、中央揃えであるとか、段落であるとか、箇条書きであるとか、表であるといった、文章の論理的な構成を、本文とは別にソースの中に書き加えなくてはなりませんが、それによって、視覚的なデザインを意識せずに文書を作成することができるのです。

$\%$ begin{何々}... $\%$ end{何々}のように対になった命令を環境(environment)といいます。たとえば、 $\%$ begin{quote}... $\%$ end{quote}ならquote環境といいます。環境の内側は一種の別天地で、いろいろな設定が環境の外側と異なります。たとえばquote環境は、引用を入力するための環境なので、左余白が周囲よりも自動的に広がります。これがL<sup>A</sup>T<sub>E</sub>Xの最大の特徴である論理デザインの利点であるといえます。

環境は具体的には、

```
¥begin{ (環境名) }
¥end{ (環境名) }
```

という書式で記述し、begin と end で挟まれた部分はその環境になります。このほかにも、中央揃えにする center 環境や、図版を張り込むための figure 環境などが用意されています。

L<sup>A</sup>T<sub>E</sub>X では環境の他にも命令と呼ばれるものが用意されています。命令は、¥begin{ 何々 }...¥end{ 何々 } という形をとらず、

```
¥命令 { 引数 }
```

という書式で記述します。たとえば、節を定義する ¥section 命令などがそれにあたります。¥section{ はじめに } という命令の場合、section という命令に「はじめに」が引数として渡されることで、「はじめに」という節があるということを宣言します。

### 1.3 空白・改行・段落

T<sub>E</sub>X では、たいいていの場合において、単語間の空白や段落の最初の空白を自動的に空けてくれます。ここでは T<sub>E</sub>X が自動的にあけてくれる空白の規則について説明します。

#### 1.3.1 自然な空白

英文の場合

T<sub>E</sub>X では英文の単語や文の区切りで自動的に適量の空白を空けるようになっています。このとき単語の区切りとしての空白より、文の区切りとしての空白のほうが幅広く確保されます。文の区切りでは、ピリオドを検知し、ある処理（詳しくは参考文献を参照）を行って余分な空白が空けられます。しかし、この規則では”Dr.Knuth”は、間違っ文の区切りとして扱われてしまいます。

また、先に述べましたように、T<sub>E</sub>X や L<sup>A</sup>T<sub>E</sub>X の命令のあとには、原則的に命令の終わりを示すために半角の空白を挿入しておかなければなりません。したがって、T<sub>E</sub>X や L<sup>A</sup>T<sub>E</sub>X の命令の直後に半角空白をあけても、単語間の自然な空白は空きません。たとえば、”¥TeX for Windows”と記述した場合の出力は、”T<sub>E</sub>Xfor Windows”となってしまいます。このような場合、T<sub>E</sub>X に半角空白以外の方法で命令の終端を認識させるか、すでに述べた単語間の空白を空ける命令を使用して対処します。具体的には、表 1.1 に示した 5 つの方法で対処することができます。

Table 1.1 単語間の空白を空ける命令

	入力	出力
1	¥TeX¥ for Windows	T <sub>E</sub> X for Windows
2	¥TeX for Windows	T <sub>E</sub> X for Windows
3	¥TeX¥@ for Windows	T <sub>E</sub> X for Windows
4	¥TeX{ } for Windows	T <sub>E</sub> X for Windows
5	{¥TeX} for Windows	T <sub>E</sub> X for Windows

和文の場合

和文のみの文書、あるいは英文と和文が混在している場合には、空間規則が若干変化します。

```
¥documentclass[a4paper]{jarticle}
¥begin{document}
コンピュータ社会で技術的な貢献をした人には ACM
Turing 賞が贈られます。特に、コンピュータ中心
分野に、著しい影響を及ぼすような貢献が対象とされています。
¥end{document}
```

この文章を  $\text{T}_\text{E}\text{X}$  で処理すると、次のように出力されます。

コンピュータ社会で技術的な貢献をした人には ACM Turing 賞が贈られます。特に、コンピュータ中心分野に、著しい影響を及ぼすような貢献が対象とされています。

まず、3行目末～4行目の ACM Turing 賞の部分と、4行目末～5行目の中心分野に相当する部分に注目してください。行末が半角文字の場合には、改行が単語間の空白として処理されます。これに対して、行末が全角文字の場合には、特に何の処理もされません。

### 1.3.2 段落の区切り

$\text{T}_\text{E}\text{X}$  は、連続する改行記号を段落の区切りとします。簡単に言えば、何も記述されていない改行記号だけの行があれば、それを段落の区切りとみなします。段落の始まりでは、 $\text{T}_\text{E}\text{X}$  が自動的に英文の場合には適当な分量だけ、和文の場合には全角一文字分の字下げをしてくれます。また、ワープロの場合と異なり、改行コードだけの行を続けても、縦方向の空白を空けることはできません。縦方向の空白を空ける場合にはそのための命令を使用します。(空白を空ける命令については、2.2.2 節を参照してください。)

### 1.3.3 書いたまま出力する方法

先に述べました通り、次に示す文字は、 $\text{T}_\text{E}\text{X}$  や  $\text{L}^{\text{A}}\text{T}_\text{E}\text{X}$  で特別な役割を果たしますので、原稿中にそのまま出力することはできません。

#, \$, %, &, -, {, }, <, >, \, |, ^, ~

したがって、”(^~)”と書くとエラーになります。これらの特殊文字を出力する最も簡単な方法は” $\text{\verb}$ ”という命令を使用することです。

$\text{\verb} | (^~) |$

この命令は、引数の始まりと終わりに同じ区切り記号を（上の場合は |）を指定することで、区切り記号の間に記述された原稿をタイプライタ文書でそのまま出力する働きを持っています。区切り記号の間に記述する文字列は基本的にどのようなものでもかまいませんが、両端の区切り記号と同じ文字だけは使用できません。また、\*記号を両端の区切り文字として使用することもできません。これは、 $\text{\verb}^*$  という別の命令があるからです。

## 第2章 基本的な文書の編集

### 2.1 文字の修飾

文章を書くときに必要になってくる文字の修飾方法は L<sup>A</sup>T<sub>E</sub>X にも当然あります。ここでは、書体や文字サイズの変更方法について簡単にお話します。

#### 2.1.1 文字サイズの変更

文字の大きさを変えるには、通常は表 2.1 の 10 種類の命令を使用します。出力した文字の右側に、その出力方法を記述しておきます

Table 2.1 文字サイズの変更

出力	ソース
sample	<code>{\tiny sample}</code>
sample	<code>{\scriptsize sample}</code>
sample	<code>{\footnotesize sample}</code>
sample	<code>{\small sample}</code>
sample	<code>{\normalsize sample}</code>
sample	<code>{\large sample}</code>
sample	<code>{\Large sample}</code>
sample	<code>{\LARGE sample}</code>
sample	<code>{\huge sample}</code>
sample	<code>{\Huge sample}</code>

#### 2.1.2 書体の変更

書体の変更方法は表 2.2 通りです。

Table 2.2 フォントの変更

フォント名	ソース	出力
ローマン体	This is Roman.	<code>\textrm{This is Roman.}</code>
サンセリフ体	This is Sans Serif.	<code>\textsf{This is Sans Serif.}</code>
タイプライタ体	This is Typewriter.	<code>\texttt{This is Typewriter.}</code>
ボールド体	<b>This is Boldface.</b>	<code>\textbf{This is Boldface.}</code>
イタリック体	<i>This is Italic.</i>	<code>\textit{This is Italic.}</code>
スラント体	<i>This is Slanted.</i>	<code>\textsl{This is Slanted.}</code>
スモールキャピタル体	THIS IS SMALL CAPS.	<code>\textsc{This is Small Caps.}</code>
明朝体	明朝体です。	<code>\textmc{明朝体です。}</code>
ゴシック体	ゴシック体です。	<code>\textgt{ゴシック体です。}</code>

## 2.2 改ページと空白の開け方

### 2.2.1 改ページ

改ページは、通常  $\text{\LaTeX}$  が自動的に位置を決めて行います。しかし、ユーザが自ら指定することも可能です。次に改ページに関するコマンドをいくつか記します。

<code>\newpage</code>	強制的に改ページしたい場合には、その場所で <code>\newpage</code> 命令を使用します。二段組みをしている場合で、現在左の段であれば、 <code>\newpage</code> 命令によって右側の段に移動します。
<code>\clearpage</code>	強制的に改ページしたい場合には、その場所で <code>\clearpage</code> 命令を使用します。 <code>\newpage</code> 命令との違いは、 <code>\clearpage</code> 命令の使用時に配置が決定されていない図表があれば、それらを全て出力してから改ページされることです。また、2 段組であるか無いかに関わらず、常に新しいページを起こします。
<code>\cleardoublepage</code>	次のページが右ページから始まるように、必要に応じて白紙のページを挿入して改ページしたい場合には、その場所で <code>\cleardoublepage</code> を使用します。

### 2.2.2 空白の開け方

先ほど説明しましたように、 $\text{\TeX}$  では空白がいくつ続いても 1 つの空白とみなされます。1 つ以上の空白を空けるには、次の命令を使用します。

```
\vspace{height}  
\hspace{weight}
```

`\vspace{height}` 命令は "height" に記述された値だけ行間を空け、`\hspace{weight}` 命令は "weight" に記述された値だけ文字と文字との間隔をあけます。

```
\vspace{10mm}
```

この命令によって、

というように、行間を設定することが可能です。

```
\hspace{30mm}
```

という命令では、というように文字と文字の間隔を設定することができます。  
また、段落間に適当な大きさの空白をあけたい場合には、次のような命令を使用します。  
`\smallskip` は小さな空白を作ります。  
`\medskip` は中くらいの空白を作ります。  
`\bigskip` は大きな空白を作ります。

このように、段落間の改行を操作することができます。

## 2.3 脚注

脚注を使用したい場合には、`\footnote{ …… }` 命令を使用します。`\footnote` 命令を使用すると  $\text{\LaTeX}$  は組版時に自動的に番号を付け、引数に指定された文字列をページの下部に脚注として出力します。<sup>1</sup>また、脚注でなく、本文の隣に傍注を出力することもできます。傍注を出力したい場合には、`\footnote` 命令の代わりに、`\marginpar` 命令を使用します。この時、傍注欄の横幅を指定することもできます。

---

<sup>1</sup>脚注はこのように出力されます。

## 第3章 画像の貼り付け

この章では、 $\text{T}_{\text{E}}\text{X}$  へ画像を貼りこむ方法を説明します。今まで Microsoft Word などでは、画像は主に JPG を用いていたと思いますが、ここ三木研では画像はすべて EPS 形式で貼り付けます。BMP や JPG と、EPS との違いはまた次の機会に説明するので、ここでは単に EPS で貼り付けたほうがきれいに出力できるからだと考えておいてください。ここでは、BMP や JPG の画像を EPS に変換する方法を説明します

### 3.1 画像の EPS への変換

BMP の画像を EPS ファイルとして保存する方法はいくつかのソフトで実行可能ですが、ここでは Adobe 社の Illustrator9.0J を例にとって解説していきます。まず Illustrator を起動すると、図 3.1 のような画面になるので適当な画像ファイルを開きます。

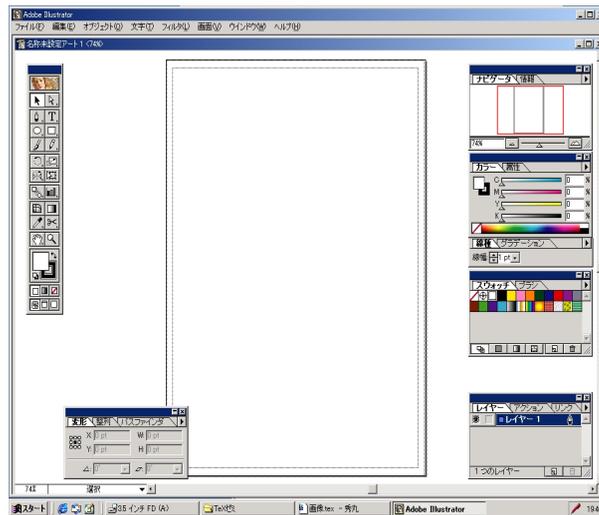


Fig. 3.1 イラストレーターの起動

ただし画像が中央に表示されている枠からはみ出すと、 $\text{T}_{\text{E}}\text{X}$  に取り込んだときに画像が切れてしまうので、貼り付けた画像がはみ出さないように調整してください(図 3.2)。EPS へ変換するには、この調整した画像を EPS ファイルとして保存します。まず、「メニュー」の「ファイル」から「別名で保存」を選んでください(図 3.3)すると、図 3.4 のような画面が表示されます。ここではファイルの種類が「Illustrator」でファイル名の部分の拡張子が「.ai」となっているはずですので、ファイルの種類から「Illustrator EPS」を選んでください。ファイルの拡張子が「.eps」になるので(図 3.5)、ここで保存ボタンを押してください。次は EPS の形式の設定を行います。先ほどの保存ボタンを押すと図 3.6 のような画面が表示されますので、プレビューの部分のなし、オプションのフォントデータを含むにチェックを入れ、オプションのサムネールを作成の部分のチェックをはずして「OK」を押してください。また最後に図 3.8 のような画面が表示されますが、気にせず「続行」を押してください。これで EPS ファイルとして画像を保存する方法は終わりです。また Illustrator を用いて描いた画像も同じようにして EPS ファイルとして保存することが可能です。

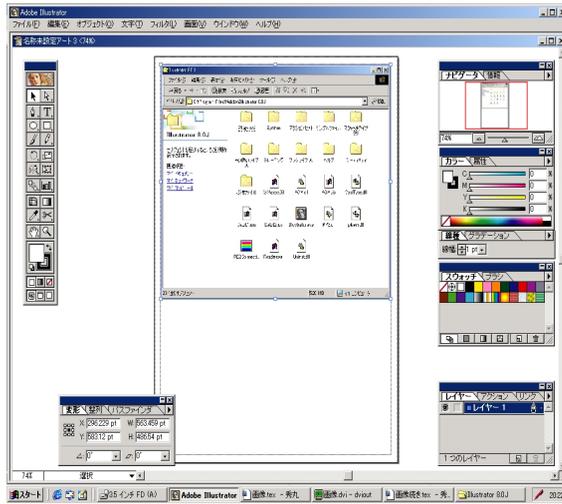


Fig. 3.2 画像調整

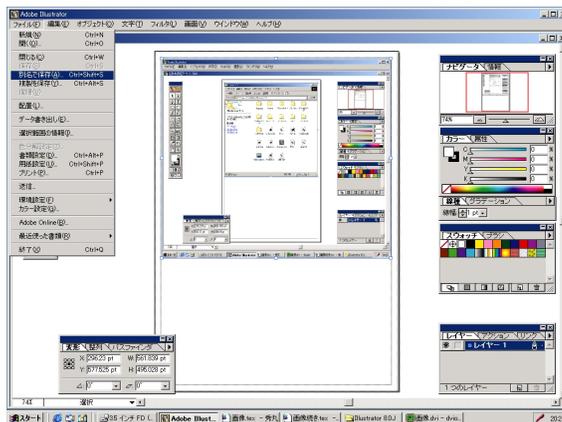


Fig. 3.3 画像の保存その1

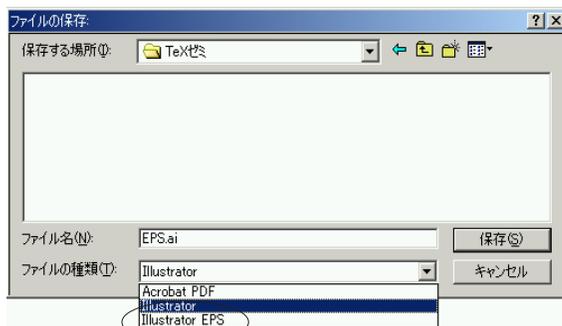


Fig. 3.4

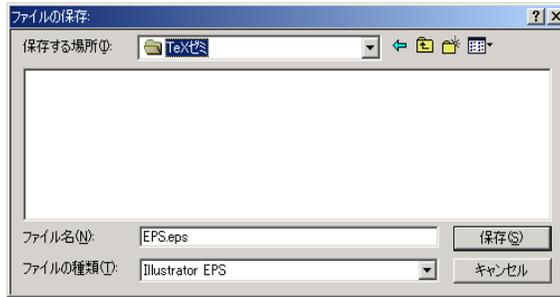


Fig. 3.5 画像の保存その 3

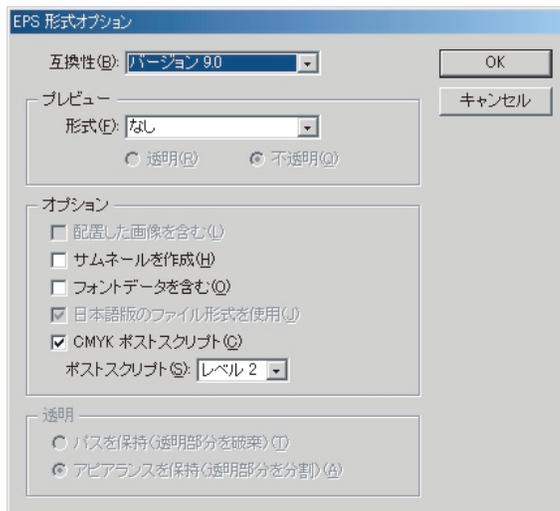


Fig. 3.6 EPS の設定前

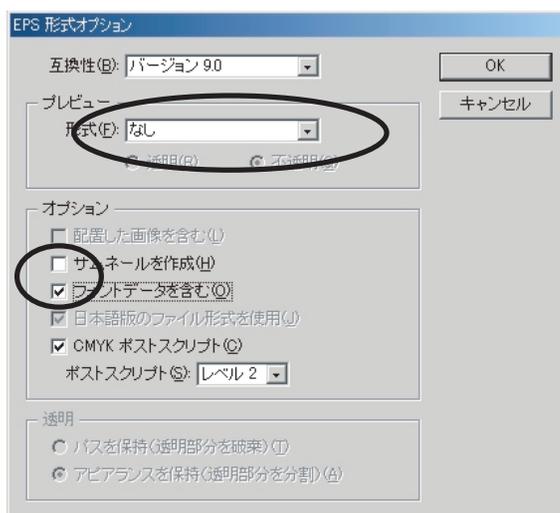


Fig. 3.7 EPS の設定後

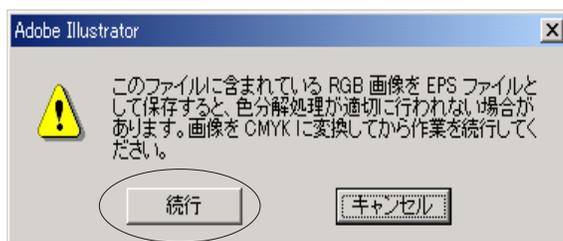


Fig. 3.8 注意

### 3.2 Ghostscript のインストール

ここでは  $\text{T}_{\text{E}}\text{X}$  の文書に EPS 形式の図を挿入する方法を解説します。EPS 画像を  $\text{T}_{\text{E}}\text{X}$  で利用するためには、Ghostscript という、PostScript 言語<sup>1</sup>を解釈（インタープリト）してくれるプログラムを用いる必要があります。Ghostscript を用いると、PostScript 形式や EPS 形式の画像ファイルや文書ファイルを Windows 上で利用可能な画像形式に変換して画面上に表示させたり、本来は PostScript 言語を解釈できないプリンタに出力させたりすることができるようになります。

DVIOUT もまた、EPS ファイルを貼り込むように指定されている DVI ファイルを出力するとき、Ghostscript の手助けを必要とします。したがって、まず始めに Ghostscript をインストールする必要があります。

Ghostscript 本体や、インストールの方法および各種設定の方法は、

```
\\Taka\software\TeX\gs6.5
```

に置いてあるので、各自でインストールおよび設定を行ってください。ここから先は、インストールおよび設定が完了しているものとして説明を行います。

### 3.3 画像の貼り付け

$\text{T}_{\text{E}}\text{X}$  に図を挿入するには graphics またはそれを拡張した graphicx というパッケージ<sup>2</sup>を使います。graphicx の方が graphics より高性能であり、DVIOUT とも相性がよいので以下では graphicx を使います。このパッケージを用いるには、 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  のプリアンブルに

```
\usepackage[dviout]{graphicx}
```

と書いておきます。そして、図 EPS.eps を入れたいところに

```
\includegraphics[width=7.5cm,height=5cm]{img2/EPS.eps}
```

のように書いておきます。これは図 EPS.eps を幅が 7.5cm、高さが 5cm になるように拡大縮小してその場所に出力することを意味します。また

```
\includegraphics[width=7.5cm,clip]{img2/EPS.eps}
```

のようにすると、縦横比 (aspect ratio) を保ったまま、7.5cm×5cm の長方形に収まるように拡大・縮小します。一般的にはこちらの方を用いるとよいでしょう。

また、通常論文などを書く場合には

```
\begin{figure}[htbp]
\begin{center}
\includegraphics[width=7.5cm,clip]{img2/EPS.eps}
\caption{図の解説をここに書く}
\label{図の参照名をここに書く}
\end{center}
\end{figure}
```

<sup>1</sup>プリンタなどの出力装置を制御するためのプログラム言語

<sup>2</sup> $\text{T}_{\text{E}}\text{X}$  の機能を拡張してくれるいわゆるインクルードファイルのようなもの

のように figure 環境に入れて用います。figure 環境は、図版を貼り込む領域を確保するための環境です。

\caption の引数には、図の説明を入れ、\label の引数には、図の参照名を入力します。本文中で図の参照を行うときにこの参照名を用いると、図の番号を自動的に付けてくれます。たとえば、図 3.8 には、参照名として EPS7 が付けられていますが、この図を本文中で参照するには、たとえば

図\ref{EPS7}を参照のこと

と入力します。この出力結果は、

図 3.8 を参照のこと

となります。

## 第4章 DVIOUTのバージョンアップ

この章では、DVI ファイルを出力するためのプログラムである DVIOUT のバージョンアップを行います。配布する CD-ROM を開き、中にある VUP.BAT というバッチファイルを実行します（図 4.1）。

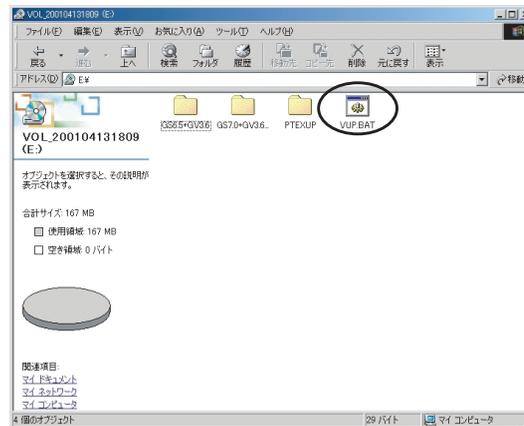


Fig. 4.1 バージョンアップのためのバッチファイル

実行すると、図 4.2 のようなウィンドウが開きますので、a と打って Enter を押してください。するとファイルの上書きが始まりますので、終了するまで待ちます。終了すると、バージョンアップは完了しているはずで。

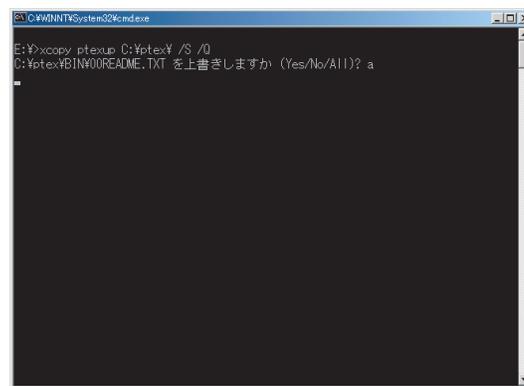


Fig. 4.2 バッチファイルの実行画面

バッチファイルの実行が終了したら、DVIOUT を起動し、[Help]→[About dviout] で表示されるウィンドウが図 4.3 のようになっていたらバージョンアップは成功です。

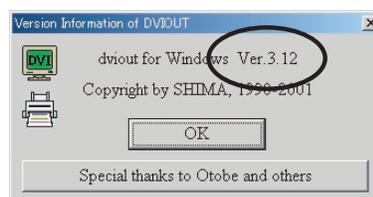


Fig. 4.3 バージョンアップの確認

## 第5章 課題 2

各自, Ghostscript のインストールおよび各種設定を行い, 適当な図を EPS に変換して DVIOUT で表示させてください.

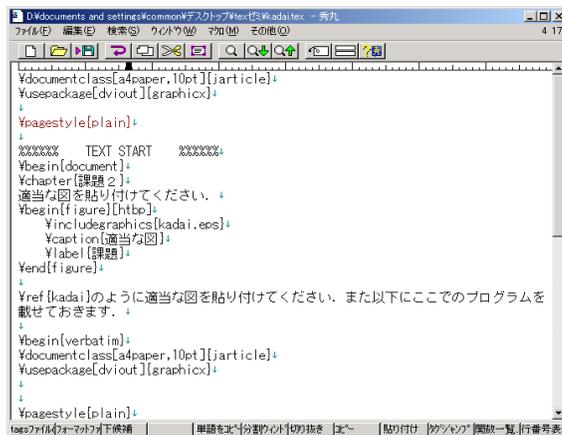


Fig. 5.1 適当な図

図 5.1 のように適当な図を貼り付けてください. また参考のために, 以下にここでのソースを載せておきます.

```
\documentclass[a4paper,10pt]{jarticle}
\usepackage[dvips]{graphicx}
\pagestyle{plain}
```

```
%%%%%%%% TEXT START %%%%%%%%%
\begin{document}
\chapter{課題 2}
```

各自, Ghostscript のインストールおよび各種設定を行い, 適当な図を EPS に変換して DVIOUT で表示させてください.

```
\begin{figure}[htbp]
\begin{center}
\includegraphics[width=7.5cm,clip]{img2/kadai.eps}
\caption{適当な図}
\label{kadai}
\end{center}
\end{figure}
```

図\ref{kadai}のように適当な図を貼り付けてください. また参考のために, 以下にここでのプログラムを載せておきます.

```
\end{document}
```