

---



---

## 第2回 L<sup>A</sup>T<sub>E</sub>X ゼミ

---



---

ゼミ担当者 : 降幡 建太郎, 澤田 淳二, 谷口 義樹  
 指導院生 : 水田 伯典, 小椋 信弥  
 開催日 : 2002年4月15日

---

ゼミ内容: 本ゼミでは, L<sup>A</sup>T<sub>E</sub>X で文書を作成する際に, 基本となる事項について説明する. また, EPS 画像の貼り付け方法について説明し, 何故, EPS 形式を使うのかについても説明する.

### 1 L<sup>A</sup>T<sub>E</sub>X の基礎

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する.

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

##### 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る. 実際は “.tex” 以外でもかまわないが, あえて約束違反をする必要もないし, ファイルの種類がわかりやすいこともあるので, 必ず “.tex” にするように.

##### 1.1.2 最低限のルール

T<sub>E</sub>X では原稿ファイルを記述するときに注意しなければいけないルールがいくつかある. ここではそれを解説する.

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, 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 “ $\%$ ” で始まる. 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, 半角の空白は必要ない.

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る. なお, 改行直後 (行頭) の半角空白は無視される.
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」のところでも詳しく説明する.
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ない.

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

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ある. 全角文字は T<sub>E</sub>X の内部で特殊な意味を持っていないので, 自由に使用することができる.

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らないが, それを出力できる DVI ドライバがない.

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る. このため, 実際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う.

#### 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ある. 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らないが, 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.

$\%$ begin{何々}... $\%$ end{何々} のように対になった命令を環境という. たとえば,  $\%$ begin{quote}... $\%$ end{quote} なら quote 環境という. 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る. 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る. 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る.

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 で挟まれた部分がその環境になる．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る．

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る．命令は，\begin{何々}...\end{何々} という形をとらず，

```
\命令 { 引数 }
```

という書式で記述する．たとえば，節を定義する \section 命令などがそれにあたる．\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ている．

### 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る．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する．

#### 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る．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る．文の区切りでは，ピリオドを検知し，ある処理を行って余分な空白が空けられる．

また，先に述べたように，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らない．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かない．たとえば，“\TeX for Windows” と記述した場合の出力は，“TeXfor Windows” となってしまう．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する．具体的には，Table 1 に示した 5 つの方法で対処することができる．

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する．

Table 1 単語間の空白を空ける命令

	入力	出力
1	\TeX for Windows	TeX for Windows
2	\TeX for Windows	TeX for Windows
3	\TeX@ for Windows	TeX for Windows
4	\TeX{} for Windows	TeX for Windows
5	{\TeX} for Windows	TeX for Windows

```

\documentclass[a4paper]{jarticle}
\begin{document}
This is a
pen.
これは、ペン
です。
\end{document}

```

この文章を T<sub>E</sub>X で処理すると，次のように出力される．

```
This is a pen. これは、ペンです。
```

まず，3 行目末～4 行目の a pen. の部分と，4 行目末～5 行目のペンです．に相当する部分に注目すると，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ている．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ていない．

#### 1.3.2 段落の区切り

T<sub>E</sub>X は，連続する改行記号を段落の区切りとしている．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ている．段落の始まりでは，T<sub>E</sub>X が自動的に英文の場合には適量な分量だけ，和文の場合には全角一文字分の字下げをしてくれる．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ない．縦方向の空白を空ける場合にはそのための命令を使用する．(空白を空ける命令については，2.2.2 節を参照．)

#### 1.4 書いたまま出力する方法

先に述べた通り，次に示す文字は T<sub>E</sub>X や L<sup>A</sup>T<sub>E</sub>X で特別な役割を果たすので，原稿中にそのまま記述することはできない．

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

したがって，“(^-^)” と書くとエラーになる．これらの特殊文字を出力する最も簡単な方法は \verb という命令を使用することである．

```
\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字でそのまま出力する働きを持っている。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わないが、両端の区切り記号と同じ文字だけは使用できない。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ない。これは、`\verb*` という別の命令があるからである。

## 2 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 $\text{\LaTeX}$ にも当然ある。ここでは、書体や文字サイズの変更方法について簡単に説明する。

#### 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 Table 2 の 10 種類の命令を使用する。出力した文字の右側に、その出力方法は以下の Table 2 のとおりである。

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>{\huge sample}</code>
sample	<code>{\Huge sample}</code>

#### 2.1.2 書体の変更

書体の変更方法は Table 3 の通りである。

### 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う。しかし、ユーザが自ら指定することも可能である。次に改ページに関するコマンドをいくつか記す。

`\newpage`

強制的に改ページしたい場合には、その場所で `\newpage` 命令を使用する。二段組みをしている場合で、現在左の段であれば、`\newpage` 命令によって右側の段に移動する。強制的に改ページしたい場合には、その場所で `\clearpage` 命令を使用する。`\newpage` 命令との違いは、`\clearpage` 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ある。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す。

`\clearpage`

`\cleardoublepage`

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 `\cleardoublepage` を使用する。

#### 2.2.2 空白の開け方

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

`\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る。

`\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ある。

`\hspace{30mm}` という命令では、と  
 というように文字と文字の間隔を設定することができる。

また、段落間に適当な大きさの空白をあげたい場合には、次のような命令を使用する。

`\smallskip` は小さな空白を作る。

`\medskip` は中くらいの空白を作る。

`\bigskip` は大きな空白を作る。

Table 3 フォントの変更

フォント名	出力	ソース
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>

このように、段落間の改行を操作することができる。

### 2.3 脚注

脚注を使用したい場合には、`\footnote{…}` 命令を使用する。`\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する。<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る。傍注を出力したい場合には、`\footnote` 命令の代わりに、`\marginpar` 命令を使用する。この時、傍注欄の横幅を指定することもできる。

## 3 PostScript

パソコンで印刷物を作成する際には、画面上できれいに表示されるだけでなく、最終的に印刷物になったときにきれいな仕上がりになっていなければならない。そういった要求に応えたのが Adobe System 社が開発した PostScript である。

### 3.1 PostScript とは

PostScript は、強力なグラフィック機能を持ったプログラミング言語である。これはページ記述言語の一種で、PostScript 言語を用いると、印刷したいページの中身を記述することができる。1つのページは必ず1つのグラフィックとして扱われる。PostScript は文字や画像を美しく正確に印刷できるため、広告デザインや編集デザインなどにおいて広く使われている。

ポストスクリプトでは、文字や図形や画像を扱うことができ、それらの属性やページ内での位置情報を指定できる。文字にはフォントや文字の大きさ、字飾りなどを指定することができ、図形は直線や円のほか、自由曲線を表現することが可能になっている。文字や図形は、ベジェ曲線を利用したベクトルデータ（図形中の主要な点の座標とそれらを結ぶ曲線の方程式のパラメータからなるデータ形式）として表現されるため、出力装置の最大

解像度での精細な出力が可能となっている。

PS 言語の命令の列を直接プリンタに送る代わりにファイルの形で保存したものを、PS ファイルという。通常はテキストファイルなので、テキストエディタでの編集も行える。dvi も PS も元々は、プリンタや画面の解像度とは無関係に組版結果を記述するためにつくられたもので、dvi は文字情報に重点を置き、PS はあらゆる種類の画像を扱える点が異なる。

### 3.2 EPS (Encapsulated PostScript) とは

PS ファイルは、ページを記述したものであるため、中身が小さなグラフィックスの場合でもそのままでは別の PS ファイルに挿入できない。すなわち、ページの中にページを挿入できないということである。そこで、ページ内に挿入できるように、PS ファイルの内容にいくつかの制約事項を設ける方式が提案された。これが EPS 形式である。EPS ファイルは、一般の PS ファイルから「ページ」の概念をのぞいた限定的な PS ファイルであるといえる。EPS 形式のファイルには、PS 言語によるグラフィックス記述に加え、プレビュー用のビットマップ画像を収めることができる。EPS ファイルをエディタで開くとわかるが、ファイル内には

```
%BoundingBox: 0 0 413 139
```

のようなコメントが書かれている。これが外枠の座標を表している。このサイズにかかわらず、読み込む際は自由に、拡大・縮小できる。

### 3.3 EPS と BMP の違い

BMP の画像フォーマットは、基本的に画像を色の付いた点の集まりとして扱う。このような方式の画像は、通常何らかのアルゴリズムで画像データ自身が圧縮されており、ファイルの大きさを小さくするような工夫がされている。また、一般的に、表示にかかる計算量も少なくすむ。しかし、コンピュータのモニタは、比較的解像度の低いピクセルを1つずつ表示するので、これらの画像ファイルを拡大または縮小しようとするのは困難と

<sup>1</sup>脚注はこのように出力される。

なる．この様に画像フォーマットでは，画像を点の集まりとして考えているので解像度の変化に伴って品質が悪化することは否めない．

EPS は，このような問題点を解決するために，画像中に存在する各種要素を点に分解して保存するのではなく，出力の直前までは要素の属性を保持しておく方式である．すなわち，さきほども述べたとおり，ベクトル形式である．線分や円でも同様に，出力の直前までは線分であれば端点を，円であれば中心と半径をという具合に，図形の情報として保管しておく．もちろん，最終的にディスプレイやプリンタで出力する段階では，要素を点の集まりに分解しなければならないが，分解はデバイスの処理能力に応じて実行されるため，描画イメージが粗くなるという事態は避けられる (Fig. 1) ．

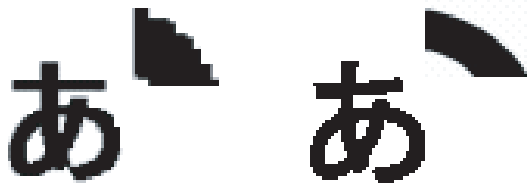


Fig. 1 BMP と EPS の違い

## 4 画像の貼り付け

ここでは， $\text{T}_{\text{E}}\text{X}$  へ画像を貼りこむ方法を説明する．今まで Microsoft Word などでは，画像は主に JPG を用いていたと思うが，ここ三木研では画像はすべて EPS 形式で貼り付ける．

### 4.1 画像の EPS への変換

まず，BMP や JPG の画像を EPS に変換する方法を説明する．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あるが，ここでは Adobe 社の Illustrator10J を例にとって解説する．まず Illustrator を起動する (Fig. 2) ．

次に，適当な画像ファイルを開く．ただし画像が中央に表示されている枠 (バウンディング・ボックス) 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する (Fig. 3) ．

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する．ちなみに，EPS はベクトル形式なので，画像のサイズをいくら縮小，拡大しようと，画像データが損なわれることはない．まず，「メニュー」の「ファイル」から「別名で保存」を選ぶ．デフォルト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」となっているはずなので，ファイルの種

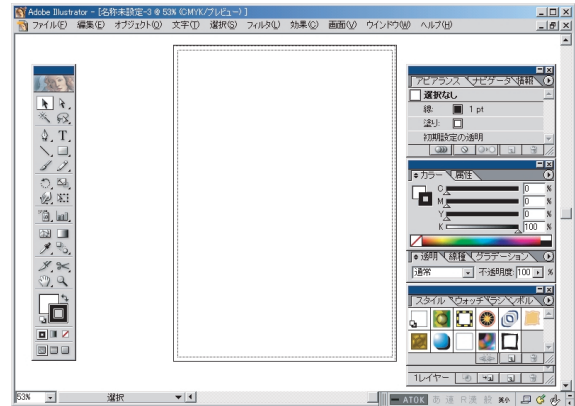


Fig. 2 Illustrator の起動画面

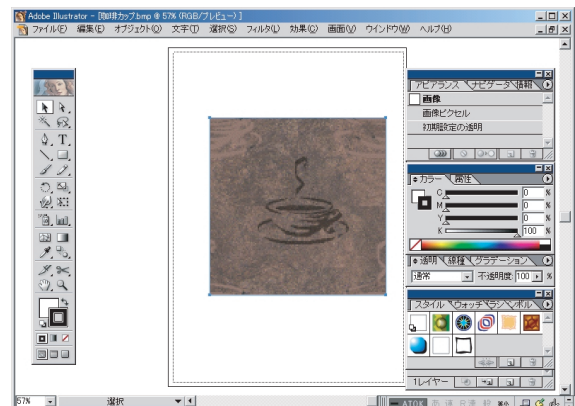


Fig. 3 バウンディング・ボックス内の画像

類から「Illustrator EPS」を選ぶ。ファイルの拡張子が「.eps」になるので、ここで「保存」ボタンを押す（Fig. 4）。

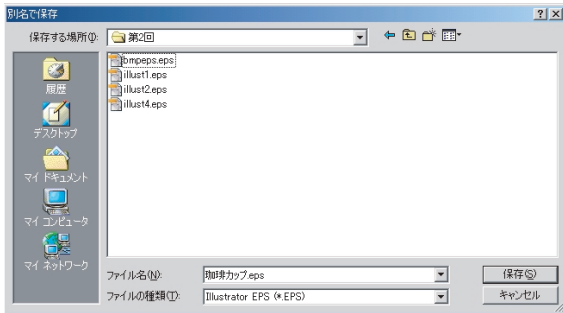


Fig. 4 画像の保存

次に、EPS の形式の設定を行う。先ほどの保存ボタンを押すと（Fig. 5）のような画面が表示されるので、そのまま「OK」ボタンを押す。ただし、プレビュー画像は、EPS ファイルを直接表示できないアプリケーションで表示される。必要なければ、プレビューの「形式」で「なし」を選択する。

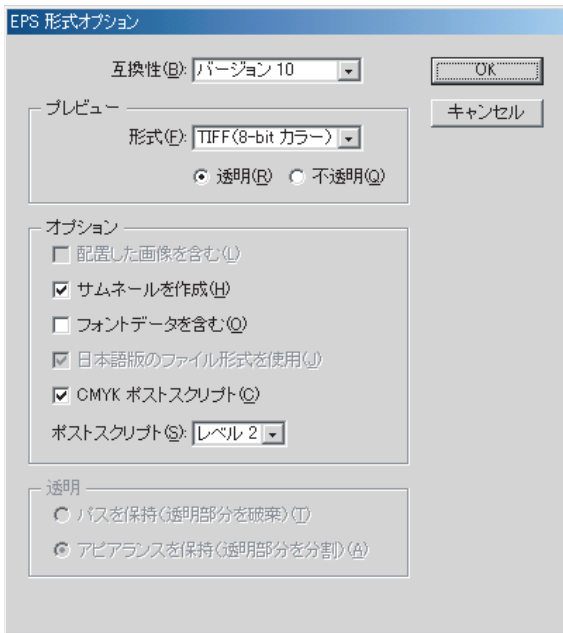


Fig. 5 EPS 形式オプション

最後に警告画面が表示されるが、気にせず「続行」を押す（Fig. 6）。

以上が、EPS ファイルとして画像を保存する方法である。

#### 4.2 TeX への図の挿入

TeX に図を挿入するには graphics またはそれを拡張した graphicx というパッケージを使う。graphicx の方が graphics より高性能であり、DVIOOUT とも相性がよ

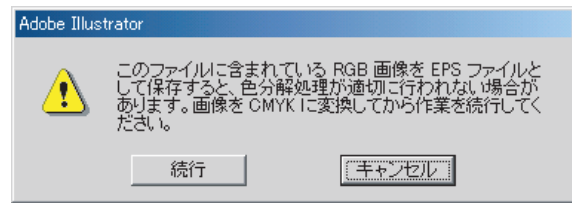


Fig. 6 警告画面

いので以下では graphicx を使う。このパッケージを用いるには、LaTeX のプリアンプルに

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

と書いておく。そして、図 EPS.eps を入れたいところに

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

のように書く。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へ出力することを意味する。また

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

のようにすると、縦横比 (aspect ratio) を保ったまま、幅 7.5cm に収まるように拡大・縮小する。一般的にはこちらの方を用いるとよい。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]{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}
```

のように、figure 環境に入れて用いる。figure 環境は、図版を貼り込む領域を確保するための環境である。caption の引数には、図の説明を入れ、label の引数には、図の参照名を入力する。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る。たとえば、参照名が "EPS7" である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

と入力する。この出力結果は、

図 3.8 を参照のこと

のようになる。

以上が、TeX への基本的な図の貼り付け方である。

#### 5 課題

各自、適当な図を EPS に変換して DVIOOUT で表示させなさい。

Fig. 7 のように適当な図を貼り付けなさい。また参考のために、以下にここでのソースを載せる。

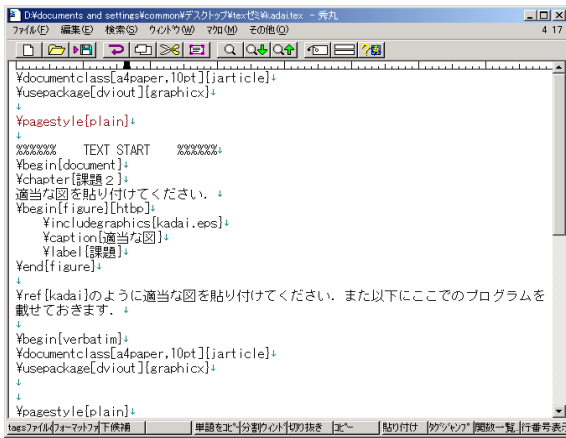


Fig. 7 適当な図

```

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

%%%%%%%% TEXT START %%%%%%%%%
\begin{document}
\section{課題}
各自、適当な図を EPS に変換して DVIOUT で表示させな
さい。

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

\fgref{kadai}のように適当な図を貼り付けなさい。ま
た参考のために、以下にここでのソースを載せる。

\end{document}

```