

WHAT YOU CAN DO WITH L^AT_EX

KAZuyuki Tamura

Astro Comp-Tool Seminar:
Arizona State University

April 9, 2010
October 1, 2010

Abstract

This is a short document to illustrate the basic use of L^AT_EX. In this example, I explain how to use some L^AT_EX commands, and how to put a text (the main body that appear in ps/pdf files) into a “***.tex” file. Make sure you have both this PDF as well as the `basic_latex.tex` to read along with. Most of the materials in this packet, Section 1–Section 5, will be a summary of “learning L^AT_EX” by David F. Griffiths and Desmond J. Higham (Figure 1). Final section, Section 6, contains some advanced use of L^AT_EX.

You can also find many useful tips and examples of L^AT_EX on Internet. Particularly, a useful web page is the Wikibook for L^AT_EX: <http://en.wikibooks.org/wiki/LaTeX>.

© 2010 Kazuyuki Tamura
School of Earth and Space Exploration
Arizona State University
Tempe, AZ 85287-1404
ktamura@asu.edu

1 Introduction

L^AT_EX is a macro package based on TeX (a low-level markup and programming language). It is a typesetting system that specializes in producing mathematically oriented documents. But L^AT_EX can create a range of documents, including journal papers, class handouts, reports, letters, slides, dissertation/thesis, books, resume/CV, etc., etc.

Why you should learn about L^AT_EX?

- Mathematical formulae can be produced easily (see Section 3). With L^AT_EX, your documents will look polished.
- Chapters, sections, equations, citations, figures, tables, etc. can be labeled, so that cross-referencing is automated.
- L^AT_EX is (usually) installed at many universities and research institutions, and can be run on many different platforms (PCs, Mac OS, workstations, and mainframe computers).
- The `tex` files have the standard ASCII format. Therefore, they can be produced using your favorite text editor.

L^AT_EX is not a WYSIWYG (what you see is what you get) system. Hence, you can't see what the final version will like while typing. Instead, you see the *logical structure* of the document. Therefore, you can concentrate on the content of your writing rather than the final appearance of the document. L^AT_EX takes care of the formatting for you. Changing the format of your document is easy, and you also have some freedom of adjusting the appearance by changing/adding a small number of formatting commands.

2 Basics

In the \LaTeX , the ‘\ (backslash)’ and special characters such as \$, ^, and _ are used to format a document.

2.1 Minimum Requirements

In the “***.tex” file, you need minimum of:

- `\documentclass{article}`
- `\begin{document}`
- `\end{document}`

And the main text are written in between “\begin{document}” and “\end{document}” lines.

For the package selection of “documentclass”, some options are:

- **article.cls** • **report.cls**
- **letter.cls** • **slides.cls**
- **minimal.cls** • **book.cls**
- **memoir.cls** • **beamer.cls**
- **ltnews.cls** • **ltxguide.cls** • **proc.cls** • **ltxdoc.cls**

The options to the “documentclass” are:

- **10pt, 11pt, 12pt** — Sets the size of main font in the document. If no option is specified, 10pt is assumed.
- **a4paper, letterpaper, etc.** — Defines a paper size. Default is “a4paper” (somehow, this does not work with my \LaTeX).
- **fleqn** — Typesets displayed formulas left-aligned.
- **leqno** — Places the numbering of formulae on the left hand side instead of the right.
- **titlepage, notitlepage** — Specifies whether a new page should be started after the document title or not. the **article** class does not start a new page by default, while **report** and **book** do.
- **onecolumn, twocolumn** — Instructs \LaTeX to typeset the document in one column or two columns.
- **landscape** — Changes the layout of the document to print in landscape mode.

2.2 Basics: How to create a PDF file

Figure 1 illustrates how a “***.tex” file is converted into DVI, PDF, and/or a PostScript (ps) files.

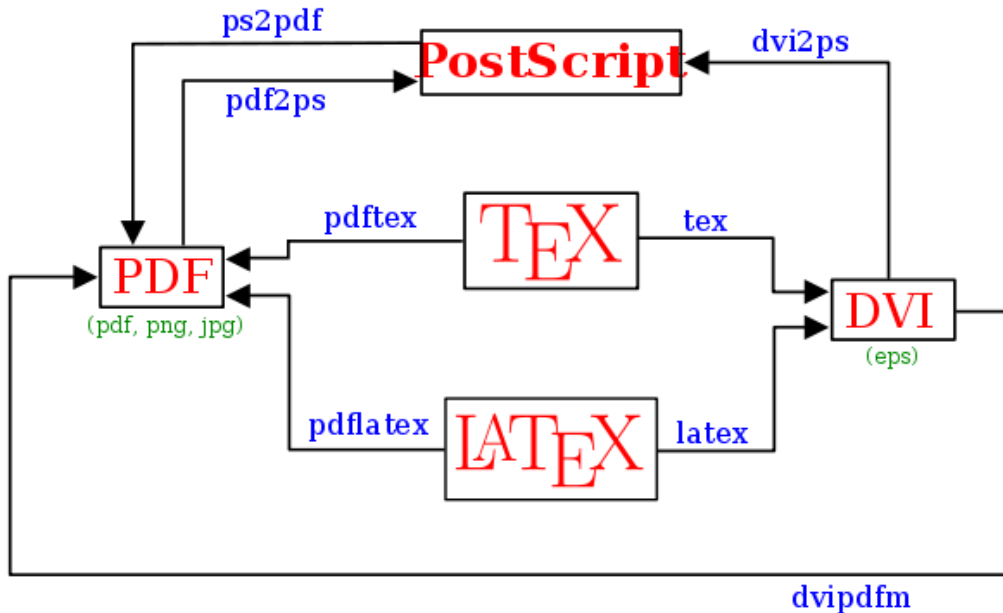


Figure 1: How a \LaTeX file is compiled (image taken from \LaTeX Wikibook). The command to convert a file from DVI to PostScript is “dvips” rather than “dvi2ps”, and the command from PostScript to PDF is “ps2pdf13” instead of “ps2pdf”.

The commands to create a PDF file (with LINUX and Mac X11 environments) for this particular file are (with PostScript figures):

```
latex basic_latex.tex
latex basic_latex.tex
dvips -P amz -P cmz -t letter basic_latex.dvi -o basic_latex.ps
ps2pdf13 basic_latex.ps basic_latex.pdf
```

If you are using JPG, PNG, and other format of figure other than PostScript, then you can simply do:

```
pdflatex basic_latex.tex
```

If you are using Windows or Mac OS X GUI based TeX software, then it might be as easy as simply clicking a button in a menu bar.

2.3 Basics: Paragraphs and line-breaks

In the main body of text, even you make a line-break in the middle of sentence, or even with every single word within the “***.tex” file, L^AT_EX will keep on writing on the same line as long as the text is continuing on the next line. On the other hand, you can also keep writing (as long as your text editor allows) on the same line, and L^AT_EX will automatically adjust the word wrap to its default (or specified) text width of a page.

If you want to make a new paragraph, simply insert an empty line (or any number of multiple empty lines) in between two paragraphs.

Or, if you want to simply break the line, you can also use “\” to make a new line. See lines 243–246 in the “basic_latex.tex” for this example.

When making a new paragraph, and you don’t want any indentation, you can add a L^AT_EX command “\noindent”, so the new paragraph starts from the left-most position. This paragraph is written with this option, so the first word is not indented.

The “%” character in the “basic_latex.tex” indicates that the text following the “%” symbol *in the same line* is commented out. If you want to comment out some lines, simply put “%” at the beginning of the line, and if you want to simply comment out a few words, you can use in the middle of the line as well. See the “basic_latex.tex” to see what is hidden around this sentence.

Commenting out with “%” is useful when you are making a short note that you don’t want to be seen but keep it there to remind you later. Or, when you are re-writing a paragraph, you can keep the old one as comment. See the tex file for example. There were total of 8 lines of comments in the “basic_latex.tex” just before this sentence, but you cannot see it in the ps/PDF file.

You can also use the `footnote`¹ to add some notes at the bottom of the page. One thing is that the numbering has to be done manually as:

```
\footnote[2]{...}
```

¹see the `basic_latex.tex` to see exactly how it works.

2.4 Basics: Fonts

The basic font size can be specified at the beginning of the tex file, as “`\{documentclass}[XXpt]{article}`”. This document is written with a 12-point font size, so the [XXpt] becomes [12pt] (See the very first line in the “`basic_latex.tex`”). If no size is defined, the L^AT_EX uses 10-point font as its default size.

If you want to change a font size within the main body, you can use “`\(fontsize)`” keyword. Those are:

- `\Huge` = Huge Font
- `\huge` = huge Font
- `\LARGE` = LARGE Font
- `\Large` = Large Font
- `\large` = large Font
- `\normalsize` = Normal Size
- `\small` = Small Size
- `\footnotesize` = Footnote Size
- `\scriptsize` = Script Size
- `\tiny` = Tiny

Some other available fonts include:

- `\textup` = Upright Type
- `\textit` = *Italic type*
- `\textsl` = *Slanted type*
- `\textsc` = SMALL CAPS TYPE
- `\texttt` = Typewriter
- `\textsf` = Sans serif
- `\textrm` = Roman
- `\textbf` = **Boldface**

If you want to emphasize a single (or multiple) words within a sentence, you can simply use “`\emph{ }`” to *emphasize* a phrase.

2.5 Basics: Making a list

If you are reading the “`basic_latex.tex`” file, you have noticed that several lists are made using `\begin{itemize}` and `\end{itemize}`, where each line/item is made with “`\item`” keyword at the beginning of line. Without any specific option, each item will start with a “•”. But you can also change it to anything (including special characters) by simply adding [] right after the “`\item`” as “`\item[***]`”. For example:

- X is made with “`\item[X]`”
- ◊ is made with “`\item[\diamond]`”
- 3) is made with “`\item[3]`”, and
- Item 1. is made with “`\item[item 1.]`”

Above are made with “`itemize`”, but you can also make it with “`enumerate`” with a simple “`\item`” if you simply want numbers at the beginning of each item as:

1. This is item No. 1
2. This is item No. 2, and
3. This is item No. 3

And if you want sub-items, you can combine two loops of `enumerate` or `itemize` as:

1. This is item No. 1
 - (a) This is item No. 1(a)
 - (b) This is item No. 1(b), and
2. This is item No. 2

2.6 Basics: Spacing

When you are writing a text, sometimes you want to shift around some words or lines or even figures and tables up/down/left/right. If you are reading the “`basic_latex.tex`”, you should have noticed that “`\vspace{*mm}`” have been used a lot. `\vspace{*mm}` shifts what follows up/down for `*mm` (millimeter). Other units of lengths such as `cm`, `in`, `em` (the width of the letter “M”—the widest character), `ex` (the height of the letter “x”), and `pt` (points) are also available.

Other than “`\vspace`”, there is also “`\hspace`” to shift anything horizontally. For example, “`\hspace{1in}`” shift the following word to left for 1 inch.

Or, you can start a line (or shift figures) to left for 2-cm by using “`\hspace{-2cm}`” (as in this one sentence paragraph).

If you use “`\hspace{-2cm}`” after a line-break by “`\`”, it usually does not shift the following line (omitted by L^AT_EX).

Like this line here (See the tex file).

But instead, if you add “`*`” as “`\hspace*{-2cm}`”, then it works. Adding the “`*`” character before `{ }` override the default setting of the L^AT_EX. So, when you use “`\vspace`” or “`\hspace`” but don’t get the effect you wanted, try adding a “`*`”.

If you want to align the text to the center of the page, you can use “`\begin{center}`” and “`\end{center}`”.

These sentences are aligned to the center of the page.

All other L^AT_EX commands works as described before.

`\hspace*{4cm}` works as well (on this line).

You can also separate the words to left/right by using “`\hspace{\fill}`”, or simply “`\hfill`”.

This line is separated to left and right by using one “`\hfill`”.

This line is separated into three sections using two “`\hfill`”s.

You can use as many “`\hfill`” as you want in a single line.

There is also “`\vfill`” for vertical separation, but I have never used it.

This last line is written at the bottom of page by using “`\vfill`” in front of it.

2.7 Basics: Cross-referencing

One reason why you want to use the \LaTeX is its automated “cross-reference” feature, as mentioned in Section 1. In \LaTeX , you can label sections, figures, tables, citations, etc. by using “ $\backslash\text{label}\{ \}$ ” command. Within the $\{ \}$, write a unique name of each section, figure, or equation that you want to cross-reference later, i.e, this section is labeled as “crossref” and the Figure 1 is labeled as “fig1”. For example, this section heading in `basic_latex.tex` is written as:

```
 $\backslash\text{subsection}\{\text{Basics: Cross-referencing}\}\backslash\text{label}\{\text{crossref}\}$ 
```

Once a label is created, you can reference it using a “ $\backslash\text{ref}$ ” command. Since labeling are automatically separated and numbered by \LaTeX into different categories (figures, sections, tables, etc.), if I type “ $\backslash\text{ref}\{\text{fig1}\}$ ”, it will appear as “1”, and if I type “ $\backslash\text{ref}\{\text{crossref}\}$ ”, it will appear as “2.7” in the text. Therefore by adding “Figure” or “Section” before using “ $\backslash\text{ref}$ ” command, will create the appropriate cross-reference as: “Section 2.7”

An advantage of \LaTeX is that since all the references are automatically managed, you do not have to fix the reference number when adding or removing any “ $\backslash\text{label}$ ” in the tex file. For example, even though you suddenly have to add a new figure in Chapter 1 of your dissertation (say already have XX figures), you do not have to go and fix the other figure numbers. \LaTeX will automatically adjust *all* the numbering instantly.

To simplify the cross-referencing, you can always make a macro for figures, tables, and sections. See Section 6.1 and the beginning of `basic_latex.tex` file for more details.

2.8 Basics: Tables

This section describes how a basic tables can be created. The first and the simplest is with `tabular`. The `tabular` environment has the form:

```
\begin{tabular}{format}
```

```
...
```

```
\end{tabular}
```

where, you specify how many columns and whether they are left justified (`l`), centered (`c`), or right justified (`r`) in the *format*.

Example 1

```
\begin{tabular}{lcr}
```

```
Friday Meetings & Time & Room \\
```

```
\hline
```

```
Astro Journal Club & 12:15 pm & PSF 226 \\
```

```
AstroComp Tool Seminar & 1:30 pm & PSF 226 \\
```

```
Another meeting & 3:00 pm & PSF 566 \\
```

```
Yet another meeting? & 4:00 pm & PSF 226 \\
```

```
Happy Hour!! & 5:00 pm & Four Peaks \\
```

```
\hline
```

```
\end{tabular}
```

Friday Meetings	Time	Room
Astro Journal Club	12:15 pm	PSF 226
AstroComp Tool Seminar	1:30 pm	PSF 226
Another meeting	3:00 pm	PSF 566
Yet another meeting?	4:00 pm	PSF 226
Happy Hour!!	5:00 pm	Four Peaks

Within the table entries,

- The entries across each row are separated by `&`.
- A horizontal line is created with “`\hline`”.
- If you want a vertical line, simply add “`|`” in the *format*, (i.e., `{l|cr}` draws vertical line between “Friday Meetings” and “Time” columns.)

If you want a heading for a multicolumn, you can use “`\multicolumn{X}`” for a heading across ‘X’ columns.

Example 2

```

\begin{tabular}{|l|c|c|}
\hline
& \multicolumn{2}{c}{Time} \\
\cline{2-3}
Friday Meetings & Feb 26 & Mar 5 \\
\hline \hline
Astro Journal Club & 12:15 pm & 12:15 pm \\
AstroComp Tool Seminar & 1:30 pm & 1:30 pm \\
Another meeting & 3:00 pm & 3:30 pm \\
Yet another meeting? & 4:00 pm & Canceled \\
Happy Hour!! & 5:00 pm & 5:00 pm \\
\hline
\end{tabular}

```

Friday Meetings	Time	
	Feb 26	Mar 5
Astro Journal Club	12:15 pm	12:15 pm
AstroComp Tool Seminar	1:30 pm	1:30 pm
Another meeting	3:00 pm	Canceled
Yet another meeting?	4:00 pm	3:30 pm
Happy Hour!!	5:00 pm	5:00 pm

- ◇ `\multicolumn` takes three arguments, (1) the number of columns, (2) format of the column, and (3) the content (text).
- ◇ `\cline{2-3}` draws the horizontal line for columns 2 and 3 (underneath “Time”).

Another environment to make a table, and probably with more useful than `tabular`, is `table`, which is added outside the `tabular` environment. Adding the following before and after the previous page’s `tabular` environment changes the look of the table as below.

before `tabular`:

```
\begin{table}[h]
\begin{center}
\footnotesize
\caption{Meeting Times For Friday Meetings}\label{table1}
```

after `tabular`:

```
\end{center}
\end{table}
```

Table 1: Scheduled Times For Friday Meetings

Friday Meetings	Time	
	Feb 26	Mar 5
Astro Journal Club	12:15 pm	12:15 pm
AstroComp Tool Seminar	1:30 pm	1:30 pm
Another meeting	3:00 pm	Canceled
Yet another meeting?	4:00 pm	3:30 pm
Happy Hour!!	5:00 pm	5:00 pm

- The `[h]` at the very beginning a option for the placement of the table within the page, that puts the table *here*, or where it is typed in the `basic_latex.tex` file. `[t]` puts the table at the top, and `[b]` puts it at the bottom of the page.

However, the factors influencing L^AT_EX table locations are so many, that sometimes tables appear other than specified location. In that case, adding “!” can/may overwrite the L^AT_EX preference (i.e., use as `[!b]`).

- The `\footnotesize` changes the font size of the table itself to a smaller fonts.
- The `\caption` can be moved to after the `tabular` environment if you want the caption of the table to appear *below* the table itself.
- The `\label{table1}` will be used in the main text when cross-referencing the table as “Table 1”. (See following sections for details about cross-referencing.)

Depending on the classfile you are using (e.g., for some pre-defined class file for journals), you can also use the “`longtable`” and/or “`deluxtable`” packages/environment to create tables. These have some additional features that you can use, but still follow the same rules to create a table.

2.9 Verbatim

Verbatim is an extremely useful environment for displaying the “raw” \LaTeX , and computer codes. All the \LaTeX significance (i.e., the use of special characters “`\& $\% _ \{ \} \# \^`”) is ignored within this environment, and whatever you type will displayed directly.

If I write as following:

```
\begin{verbatim}
%% This '%' sign is usually used to comment out words following
%% in a same line.
```

```
$ % & ^
```

```
y = a*x + b + cos(theta)
\end{verbatim}
```

It will appear as:

```
%% This '%' sign is usually used to comment out words following
%% in a same line.
```

```
$ % & ^
```

```
y = a*x + b + cos(theta)
```

This is useful when one have to show a (portion of) programming code, which is usually filled with symbols and characters that \LaTeX treat as “special” characters.

3 Typesetting Mathematics

3.1 Basics

One of the strong-point of \LaTeX is mathematical typesetting. There are several ways to write this in \LaTeX . Simplest way is to put in between '\$' symbols, which is usually used in a sentence. For example, $\$x\$$ and $\$y=2.5x\$$ will appear in *italic-like* fonts as, '*x*' and '*y = 2.5x*'.

To display equations (unnumbered) in a separate line, use the delimiters $\backslash[$ and $\backslash]$.

$$y = \sin(x)$$

where, "sin" is typed as $\backslash\sin$ and NOT as $\$sin$. Mathematical functions such as "sin", "log", and "cosh" are, by convention, typeset in standard roman type. Or, if you need a certain situation that math fonts to be in roman type, then you can use $\backslash\mathrm$. For example,

$\backslash\mathrm\{y = 2.5x\}$

appears as: $y = 2.5x$.

Other equation environments that you can use to write numbered equations are:

1. $\backslash\begin\{equation\}...\backslash\end\{equation\}$

This only gives a single line equation and a simple number.

$$A_N = \sum_{i=1}^N a_i \tag{1}$$

2. $\backslash\begin\{eqnarray\}...\backslash\end\{eqnarray\}$

This let you format sets of equations.

$$\int_{x=0}^{\infty} e^{-x^2} dx = \frac{\sqrt{\pi}}{2} \tag{2}$$

$$\int_{x=0}^{\infty} x e^{-x^2} dx = \frac{1}{2} \tag{3}$$

The \eqnarray can also be used to align equations, or in other words, it can be used as a table (in math mode) with precisely three columns within

a row. (Separated with `&`.)

$$\begin{aligned} y &= x^4 + 4 \\ &= (x^2 + 2)^2 - 4x^2 \\ &\leq (x^2 + 2)^2 \end{aligned} \tag{4}$$

where, if you don't want any number at the end of each line, you have to add "`\nonumber`" before the "`\\`". Or if you don't want your equations to be numbered at all, you can simply use `\begin{eqnarray*}... \end{eqnarray*}`, with "*" marks at added.

3.2 If you need more than basics

3.2.1 Fonts, Hats, and Underlining

In math mode, you can type equations, but you can also use other fonts, and markings (hats, underlining, etc.). One of the most important, or frequently used commands are for spacing. For example, if you want to add a space between a and b in a single line within math environment, simply adding a space (' ') does not change the spacing between a and b .

ab – typed as `'ab'`

$a b$ – typed as `'$a b$'`

$a b$ – typed as `'$a\!b$'` = Negative thin space

$a b$ – typed as `'$a\,b$'` = Thin space

$a b$ – typed as `'$a\:b$'` = Medium space

$a b$ – typed as `'$a\;b$'` = Thick space

These become important in situations when you have, e.g., double integrals:

1) without space:

$$\iint f(x,y) dx dy = \int \int f(x,y) dx dy$$

2) with spaces:

$$\iint f(x,y) dx dy = \int \! \int \! \int f(x,y) \, dx \, dy$$

You can also use markings such as:

\hat{a} = `'\hat{a}'`

\dot{x} = `'\dot{x}'`

$\widehat{\alpha + \beta}$ = `'$\widehat{\alpha + \beta}$'`

$\underline{\Gamma - \Omega}$ = `'$\underline{\Gamma - \Omega}$'`

and,

$\overline{\dot{x} + v_0}$ = `'$\overline{\dot{x} + v_0}$'`

3.2.2 Braces

You saw curly braces `{` and `}` have a special meaning in \LaTeX . It is same in equations. So, if you physically want `{` and `}` to be typed, then you have to use `\{` and `\}`. However, `{` and `}`, as well as `'()'` and `'[]'`, have a fixed height of a single line in math mode if typed directly, and it *won't* adjust its height even if `\frac` command is used to write some equations.

$$p(x) = 4 \times [2x + (\frac{3x}{5} + \frac{1}{\{x-1\}})]$$

To match the height of braces, you have to use `'\left{'` ... `'\right}'`, `'\left('` ... `'\right)'`, and `'\left['` ... `'\right]'`.

$$p(x) = 4 \times \left[2x + \left(\frac{3x}{5} + \frac{1}{\{x-1\}} \right) \right]$$

3.2.3 Arrays and Matrices

Arrays and Matrices can be written as well *within mathematical environment*. The format used in `array` environment is similar to that of tables. See the `basic_latex.tex` for examples how the followings are written.

1. Simple array:

$$\begin{array}{cccc} a & 0 & \sin(2x) & c \\ a + b & 16 & \cos x & b + c \end{array}$$

2. Matrix:

$$A = \begin{bmatrix} 1 & 1 & 2 \\ x & y & z \\ (2 \tan y) & xy & x^3 \end{bmatrix} \neq B \quad (5)$$

3. Some useful way of using `array`:

$$\delta_{ij} = \begin{cases} 1 & \text{when } i = j, \\ 0 & \text{when } i \neq j. \end{cases} \quad (6)$$

In `basic_latex.tex`, for the last example, the `"\right."`, which is called "dummy" right brace, is required to match the `"\left{"` we used at the beginning of the array. Otherwise, \LaTeX would complain that `\left` did not have a matching partner. Also, the `"\mbox{...}"` are used to temporarily leave mathematical typesetting mode.

4 Including Figures

In \LaTeX , if you are using *'latex'* and *'dvips'* to create a PostScript document, you can only input PostScript (****.ps*) files, or Encapsulated PostScript (****.eps*) files only. On the other hand, if you are using *'pdflatex'*, *TeXShop*, or other PDF-oriented compiler to create the final document, you can only use PDF, PNG, and JPEG (and GIF?) images.

Here, I explain the case for creating a PostScript document with PostScript (*ps*) files.

To display a pictures, you have to use a command such as “`\includegraphics`” or “`\psfig`”. To use these commands, the `graphicx` or `psfig` package has to be loaded at the beginning of `basic_latex.tex` document, between the lines `\documentclass[...]{...}` and `\begin{document}`.

```
\documentclass[12pt]{article}

\usepackage{...}
\usepackage{graphicx}
\usepackage{psfig}

\newcommand{\figref}[1]{Figure~\ref{#1}}

\begin{document}
```

The following example will load the figure as in the following page:

```
\begin{figure}[h]
\centerline{
\includegraphics[height=0.8\textheight]{tmp.ps}
}
\caption[Figure Title Comes Here]{\footnotesize One of the best
introductory level  $\LaTeX$  textbook, ‘learning  $\LaTeX$ ’ by David
F.\ Griffiths and Desmond J.\ Higham. Published by saim (ISBN:
0-89871-383-3). Sold at Amazon.com for  $\$26.39$  (new).
}\label{fig2}
\end{figure}
```

As in tables, options “[!h/t/b/p]” can be used to specify the location of the figure to put in, and adding “*” (as `{figure*}`) overwrite the basic \LaTeX settings.

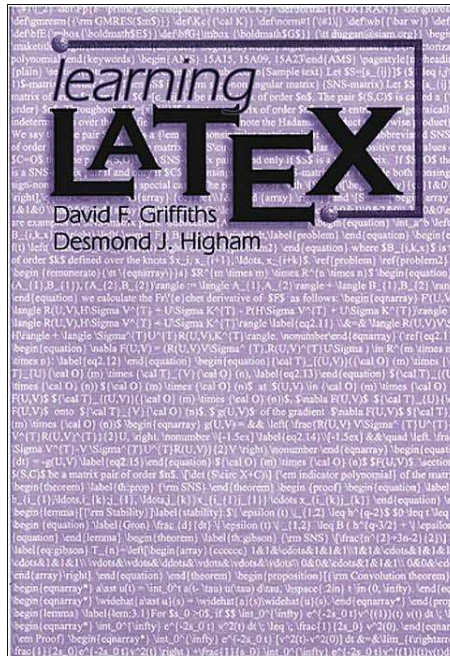


Figure 2: One of the best introductory level L^AT_EX textbook, “learning L^AT_EX” by David F. Griffiths and Desmond J. Higham. Published by saim (ISBN: 0-89871-383-3). Sold at Amazon.com for \$26.39 (new).

To convert the format of images into a PostScript (***.ps), I usually use “gimp (<http://www.gimp.org/>)”, but other image processing tools/software, should be able to convert the image format as well.

If the L^AT_EX is having trouble when creating pdf file with a converted PostScript file, try to modify the PostScript file as following. The **red** are originally in the converted file, and the **blue** are how you have to change.

- at the very first line of the “***.ps” file:
%!PS-Adobe-3.0
%!PS-Adobe-3.0 EPSF-3.0
- For lines with “Page/Pages”:
%%Page(s): 1
%%Page(s): 1

5 Making Bibliography

A bibliography can be created with the `thebibliography` environment. As in the list-making environment, you start a new input “`\bibitem`”, but then you can refer/cite the item anywhere in the document with “`\cite`”, “`\citet`” or “`\citep`” commands (the latter two require more specific ‘class-file’ and/or ‘style-file’).

For example, to create a reference list, you can simply type:

```
\begin{thebibliography}{99}

\bibitem{short_name} Authours, Title, etc.

\bibitem{griffith97} Griffiths, D.F., \&
  Higham, D.J., \emph{learning \LaTeX}, siam, Philadelphia, 1997.

\end{thebibliography}
```

and this will create the following:

References

- [1] Authours, Title, etc.
- [2] Griffiths, D.F., & Higham, D.J., *learning L^AT_EX*, siam, Philadelphia, 1997.

This will appear as [GH97], when “`\cite{griffith97}`” is typed in a sentence. Or, if you rather want name/short notation to be shown, you can add an option as:

```
\bibitem[GH97]{griffith97} Griffiths, D.F., \&
  Higham, D.J., \emph{learning \LaTeX}, siam, Philadelphia, 1997.
```

which is cited as [GH97], and appears in the “References” as:

References

- [GH97] Griffiths, D.F., & Higham, D.J., *learning L^AT_EX*, siam, Philadelphia, 1997.

6 Advanced Use of L^AT_EX

6.1 Macros

In L^AT_EX, you can define frequently used phrases and commands with shorter notation using “\newcommand”. Best if you all write together at the beginning (*before* \begin{document}), but you can use it anywhere in the tex file.

```
\newcommand{cmd}[args]{def}%
```

cmd — A command name beginning with a \. For \newcommand, it must not be already defined, and must not begin with \end.

args — An integer from 1 to 9 denoting the number of arguments of the command being defined. The default is for the command to have no arguments.

def — The text to be substituted for every occurrence of **cmd**. A parameter of the form ‘#n’ in **cmd** is replaced by the text of the nth argument when this substitution takes place.

% — Add this to the end to make sure the defined macro ends here. If you forget the closing ‘}’, it can cause some errors.

Examples:

```
\newcommand{\thistex}{\texttt{basic\_latex.tex}}%
\newcommand{\figref}[1]{Figure~\ref{#1}}%
\newcommand{\figreftwo}[2]{Figures~\ref{#1} and \ref{#2}}%
\newcommand{\HII}{\mbox{H\, ,\textsc{ii}}}%
```

When you use, simply type “\thistex”, “\figref{fig1}”, and “\figreftwo{fig1}{fig3}”, etc. When a word/phrase is defined with a \newcommand, you have to manually add an extra space *after* the macro by ending it with “\”. For example, if you don’t add “\”, simply typing “\HII region” will appear as “HIIregion”. But if you type “\HII\ region”, then it will appear as “H II region”.

6.2 Adjusting the Numbering of Figures and Tables

When you have one figure that spans a multiple pages, or use different format of figures/tables in a single document, the numbering can be shifted. In such a situation, you can use “`\addtocounter`” command to adjust the numbering.



Figure 3: Second image in the document *without* using `addtocounter`. Two images are scaled with `'height'` (rather than `'width'`) for better size-matching.



Figure 6: Third image in the document appearing as “Figure 6” (instead of Figure 4) by adding `\addtocounter{figure}{+2}` before the `\begin{figure}` command.

6.3 Adjusting the Location of Figure Caption

6.3.1 SCfigure

Sometimes, you might want to save space by moving a caption to a side of a figure, instead of writing it under the figure. In that case, you can use a command “SCfigure”. You also need to load the “sidecap” package by adding a line “\usepackage{sidecap}” at the beginning of tex file.

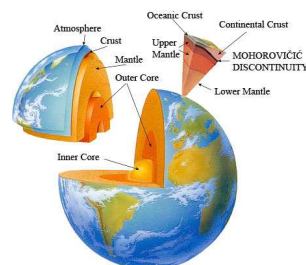


Figure 7: This figure caption is written on the right side of the figure by using “SCfigure” command.

6.3.2 wrapfigure

Another way of manually writing caption or main text around the figure is with “wrapfigure”. As for the ‘SCfigure’, to use the “wrapfigure”, you have to load “wrapfig” package at the beginning of the tex file. This also allows you to save some space. The text written *below* the “\endwrapfigure” will appear next to the figure, as in this case.

Figure 8: You can write a figure caption here (on a side of a figure) *manually*.



You can use this to manually add some figure caption (typed in blue above) as a part of main text. This will allow figure caption to take wider space than the image width (compare to “SCfigure” above). As well as you have a choice of putting your figure on the *right* or *left* of the wrapped text. The space above the figure and below the figure caption can be adjusted by ‘\vspace’ command.

Figure 8: Or, you can write caption here.

6.3.3 minipage

Another way to create a wrapped text or side caption is with “minipage”.



Figure 9: Depending on the size of the image, you can adjust the size of left/right column widths.

IMPORTANT!!: With this method (without using `\begin{figure}`), the numbering of figures might change after this. Make sure to adjust the figure numbers with `addtocounter{figure}` on the following figure.

This is also useful when writing the “References” section in two (or more) columns. Following is an example.

References:

Abraham, R.G., et al. 1999, MNRAS
308, 569
Anders, P., & U. Fritze-von Alvensleben
2003, A&A, 401, 1063

Heller, C.H., et al. 2007, ApJ, 657, L65
Jogee, S., et al. 2004, ApJ 615, L105
Sheth, K., et al. 2008, ApJ 675, 1141
Trumpler, R. J. 1930, PASP, 42, 214

6.4 Use of “\input” and “\include” Commands

Sometimes—especially if you are writing long documents such as your dissertation or a book—it is easier to write each chapter as a separate \LaTeX file, and then put all together at once using a master tex file. (i.e., I crated my dissertation (total 243 pages, but “kaz_thesis.tex” is less than 100 lines) in this way.)

`\input ***.tex` — Simply insert another tex file into the position specified.

`\include{***}` — *WITHOUT* the “.tex” file extension. This put the another tex file *starting from a new page*.

These are useful when you are including tables, references, chapters, etc.

6.5 Predefined (Journal) Style files and Class files

Some journals and organizations, require their original format in documents prepared with \LaTeX . If you are writing papers for astronomy journals (AJ/ApJ) you can use “AASTeX”. If you are writing papers for Elsevier, they have their own class-files and style-files. And if you are writing one to Nature, the DO NOT accept LaTeX files, but *they accept the PDF files created with \LaTeX* .

For example, if you download “AASTeX” from:
<http://ucpjournals.uchicago.edu/AAS/AASTeX/>
The tar file (zip file) include:

- `aastex.cls` = class file, which format the paper,
 - `natbib.sty` = bibliography style file,
- and other associated files (README, support document, template tex file, etc.).

For Elsevier, who publishes various journals with many different formats, you can download the specific class-file and style-file form:
<http://www.elsevier.com/wps/find/authorsview.authors/elsarticle>

And for Nature, you can google it to find the class-files that are modified by different individuals. Since they are NOT OFFICIAL class-files, make sure the created PDF satisfies the Nature requirements.

6.6 ASU Dissertation with L^AT_EX: 2007-2010 Format

IMPORTANT!!

Different printers have somehow slightly different margin set-ups. Make sure to adjust the margins to your printer before printing the final version of your thesis/dissertation.

To obtain the necessary style files and class files, contact Kaz Tamura or recently graduated people. **These style files are modified to match the format of *Astronomical Journal* (and *Astrophysical Journal*) *after* 2009 (IOP version of Journals).**

For 2009/2010 format of dissertation, the necessary files are:

- **report4.cls** — Enabled the use of “\subsubsection” (modified by Melissa Morris in 2009).
- **body_only.sty** — For printing main body of thesis.
- **toc_only.sty** — For pages *before* the actual table of contents (i.e., title page, approval page, abstract, and acknowledgment).
- **toc_p1.sty** — *ONLY* for page 1 of the “Table of Contents”.
- **toc_p2on.sty** — For pages 2–X (second page on) of “Table of Contents”, and for pages 2–X of “List of Tables/Figures”.
- **toc_tab_fig_p1.sty** — *ONLY* for page 1’s of “List of Tables/Figures”

You also need:

- **longtable_asu.sty** — Table style file to match the table layout with *Astronomical/Astrophysical Journal*. Modified by Kaz Tamura.

The original style file is:

- **asthesis4.sty** — Original Dissertation/Thesis format style-file originally created by John Jones in 1998, modified by various people (e.g., Kevin Healy, Melissa Morris, Nimish Hathi, and Kaz Tamura).

For example, the beginning of your thesis tex file should look like:

```
\documentclass[12pt]{report4}
\usepackage{body_only,natbib,amsmath,rotating,amssymb}
\usepackage{graphicx,epsfig,lscap,psfig}
\usepackage{longtable_asu}
```

where, the “**body_only**” should be changed to another classfile based on which page(s) of thesis/dissertation you are preparing for.

You can find the ASU thesis template (formatted for 2009–2010 Astronomical Journal style) at:

<http://www.public.asu.edu/~ktamura/>
under “Presentations/Talks”.