

A Working Guide To L^AT_EX *

S. C. Phatak †

Institute of Physics
Bhubaneswar 751 005

December 23, 2002

Abstract

An abstract is an essential part of a document which gives a brief description of the contents of a document. So, for this document, the abstract would be: This document is a file that includes basic L^AT_EX commands. Users can use this file to construct a document he wants to make by picking up parts of this document. The user will be able to make functional texts using this documents. For fancier things, the user should refer to standard L^AT_EX guides[1, 2]. This document does not claim to replace these guides.

1 Introduction

L^AT_EX is a very powerful document preparation system based on T_EX document preparation system. Nice thing about this system is its portability. A document prepared in this system can be transported to any machine having any operating system so long as the machine has the L^AT_EX installed. The portability works at two levels. First, one can have the raw document that one types in (usually having a filename ending in .tex) transported to any machine and compiled on it. The second

*Prepared for IoP students

†email : phatak@iopb.res.in

level is, on compilation the machine produces a file with extension .dvi which stands for device-independent file. This file can be transported to any machine and used to print or view the document, provided the \LaTeX is installed on the machine. The second advantage with the \LaTeX system is that it allows one to control the layout of the document as would be done by a type-setter. (It is even more so with the \TeX). So, one can make really professional looking documents. It was more so ten years back when commercial word processors were not that sophisticated. Now the commercial word processors can make professional-looking documents but still, because of portability the \LaTeX is still attractive. For example, a document prepared using Microsoft word package can only be used with a machine having windows with a word package. Sometimes, the document may not work with different versions of word package. The biggest plus point, of course is that it is an integral part of LINUX. So, any machine with LINUX has \LaTeX built-in. The negative point about \LaTeX , in fact the point which turns off large number of users, is one does not see what one is typing immediately. That is, one types the document, compiles it and then sees it. So, for somebody who is not technically inclined, this becomes a stumbling block. Modern word-processors do not have this handicap. However, now with LINUX (or any other) work-stations, one can almost have this feature. For example, one can prepare a document in one window, periodically compile in second window and view in the third window. In fact, xemacs allows one to do that at one stroke.

Now I will very briefly describe how one goes about making a document in \LaTeX . The procedure consist of three steps which one has to repeat several times, especially in the begining when one is not familiar with \LaTeX language. These are

1. Prepare a file following the rules of \LaTeX . The name of the file should be *foo.tex*¹. That is, the extension (in the language of DOS should be tex).
2. Compile the text file by using the latex command
latex foo

Note that .tex need not be specified (although does not hurt if you do). The compilation produces the files foo.dvi, foo.aux and foo.log. The file foo.dvi

¹Here *foo* is a generic name of a file. So, replace *foo* by the name of the file you are dealing with

is the most important file among these. It is used to view the output etc. foo.log contains the details of messages generated by latex which are useful for debugging. foo.aux is the file which contains the details of cross-referencing etc.

3. View or print the document you have prepared by using the file foo.dvi. As mentioned earlier, it is the device-independent file which is used by drivers for display etc. You will need two drivers. One is xdvi which generates an output which can be viewed on the screen. The command for viewing is

xdvi foo

The command will open a new window in which the document will be displayed. For printing, there are a number of commands/drivers which are used. Most commonly used command is

dvips foo

This command is used to make a post-script output.

One has to repeat these steps till one gets the effects one wants. This is what makes the preparation of the document in \LaTeX somewhat cumbersome. But if one uses xemacs, these things are done at a click of the button, so it's not so bad.

The way we shall go about is by actually making a document which includes as many features of \LaTeX as possible. As mentioned earlier, this is not going to replace Lamport's book or the online tutorials of \TeX users group but with the help of this document you will be able to start making reasonably good-looking documents. Once you get a hang of the works, you can go further and learn more tricks if you want. We shall consider a number of possible environments that one will run into. What you have to do is use this document itself and learn about these environments or use these in your document using cut-and-paste method and make your document while you are learning. Things have become easier with xemacs. In xemacs, you can start new environments, change environments, add sections, subsections etc, define macros and so on at a click of the button.

In fact, the way to use this document is to look at the tex file itself and the output of this file together. By comparison, you will be able to learn how to use different environments. The next thing is to use the text of this file in your \LaTeX document and edit appropriate portions to suit your needs.

Before we go on, let me mention that there are some special characters in $\text{T}_{\text{E}}\text{X}$ and $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$. Some of these are %, &, \$, {, }, -, ^. Their meanings are defined in the table below. (You can use this to learn about the table environment which will be introduced later.)

%	Comment. Whatever follows % on the line is ignored
&	Character used for alignment of columns (as in this table)
\$	Character to signal begining and ending of math mode. Whatever present between two \$'s or two pairs of \$'s (i. e. \$\$ \cdots \$\$) is in math mode
{ and }	Characters used to define the extent of an environment. { and } are not printed.
_ and ^	characters used to define subscripts and superscripts in math environment

2 Environments

The most elementary environment is the text environment. Normally, the text you want to print should be placed on a page so that it is left as well as right justified. That is, the left-most and right-most parts of the text should be aligned. That is what $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ (actually $\text{T}_{\text{E}}\text{X}$) does. One need not declare the text environment since it is the standard environment. You may want to change fonts within the text. The number of fonts available is somewhat limited but these are enough for a scientific document. The standard text is displayed in roman font. Other often used fonts and how to use them is shown in the table here. The left column shows the type of font used, the middle column shows how the text is entered in the latex file and the right column shows how it appears on the page. Note that the text written in the changed fonts is enclosed in the braces (curly brackets).

emphasize	<code>\emph{emphasize}</code>	<i>emphasize</i>
bold face	<code>\bf{bold face}</code>	bold face
typewriter	<code>\tt{typewriter}</code>	typewriter
small capitals	<code>\sc{small capitals}</code>	SMALL CAPITALS
italic	<code>\it{ italic }</code>	<i>italic</i>
slanted	<code>\sl{slanted}</code>	<i>slanted</i>
roman	<code>\rm{ roman}</code>	roman

In addition to the different fonts, the size of the fonts can also be changed in the document. For example, the text this document is prepared in the fonts of 12 points (declared in the documentclass statement)². One can use different font sizes. One way is to change the font size in the documentclass statement itself. That will change the size of all characters in the text. The other is to scale the size of the letters with respect to the ‘standard’ font size one is using. Different sizes available are `tiny`, `scriptsize`, `footnotesize`, `large`, `Large`, `LARGE`, `huge` and `Huge`. So there is a large variation of font sizes one can have. You may want to have your document justified at left or at right or centered. This is done by `flushleft`, `flushright` and `center` environments. Below is an example of doing it.

This text is left justified. Therefore the right margin is uneven, as would appear when you type the manuscript normally. This is the `flushleft` environment. This effect is produced by inserting this text in between the statements `\begin{flushleft}` and `\end{flushleft}`. If a single line is to be written in this fashion, you can use `\raggedright` declaration and place the line between two braces.

This text is right justified. Therefore the left margin is uneven. This is the `flushright` environment, used in paragraph mode. It is invoked by inserting this text in between the statements `\begin{flushright}` and `\end{flushright}`. As in case of `flushleft`, you can type a single line using `\raggedright` declaration.

²This is the unit in which the typesetter measures length.

Here the paragraph is centered. You need this environment when typing (for example) titles page. Again, this is in paragraph mode and the effect is produced by inserting this text in between the statements `\begin{center}` and `\end{center}`. For a single line to be centered, one can use `\centering` declaration.

2.1 Math Environment

Now we come to the most useful environment for which \LaTeX or \TeX has edge over all word processors. That is making mathematical text. There are various types of mathematical texts. One situation is when you want to insert a formula within the text (appearing on the same line). Next situation is when you want the mathematical expression to appear on a separate line but not as a regular equation (with equation number etc). Finally you may want a regular equation. Usually, you would want to format the equations so that they look esthetically pleasing. You can do all that in math environment. Simple in-line expressions are written between two $\$$ signs. For example $x = y$. Note the change in the font of the mathematical equation. If one wants to write this equation on a separate line, one types

$$x = y.$$

Coming to regular equations, there are two environments possible. These are equation and eqnarray. The equation environment is useful for single, short equations which do not require more than one line. For example,

$$y = e^x \tag{1}$$

Note that the equation environment is quite similar to the earlier example except that one now has equation number appearing on the right. One can also label the equations for reference in the text. For example, if one wants to say that y is a monotonically increasing function of x because of eq(1), one would write ... because of eq(1). Finally, eqnarray environment is a combination of equation and array environments. It is useful when the equations are long and overflow to many lines or when one wants to write several equations together so that they look esthetically pleasing. For example,

one may want to align = signs in one column. Below is an example of doing this.

$$y = \frac{e^{ix} - e^{-ix}}{2i} \tag{2}$$

$$= \sin(x) \tag{3}$$

You have already seen how to write superscripts. The subscripts are written by using ‘underscore’ key `_`. Thus we have $\log_{10} x$. You have also seen how to write a fraction using the macro `\frac`. A number of such macros are defined in `LATEX` and these are quite handy. These are too many to enumerate here. These are included in the appendix. Other handy things are Greek letters which are so often used in equations.

In `eqnarray` environment, each line is treated as an equation and numbered. If you are writing a long equation, each line should not be numbered. This is done by typing `\nonumber` before `\\`. Note that `\\` ends a line in text as well as math environment.

One also uses script letters (also called caligraphic letters) in mathematical equations. These are invoked by caligraphic declaration `\cal`. Thus we have *CALIGRAPHIC*. Note that caligraphic declaration works in math environment only.

2.2 Array Environment

One needs to write arrays in mathematical equations. One needs array environment for this. Array environment is also required for presenting data in a tabular form. But array environment is more versatile than the tabular environment in certain respects. We shall consider tabular environment later. The array environment is operative within math environment where as the tabular environment is operative in text environment. Let us consider that we want to write an equation $AX = Y$ explicitly. Here A is a 3×3 matrix and X and Y are 3×1 vectors. This equation is written as

$$\left(\begin{array}{ccc} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{array} \right) \left[\begin{array}{c} x_1 \\ x_2 \\ x_3 \end{array} \right] = \left| \begin{array}{c} y_1 \\ y_2 \\ y_3 \end{array} \right|$$

. We have used three different delimiters (symbols which contain an expression) just as an example. Note that the delimiters are defined by means of the declaration `\left` and `\right`. One must have `\left` before `\right`. Not having one of them produces an

error. This is avoided by declaring `\left .` or `\right .` if one does not want to have one of the delimiters.

2.3 Tabular Environment and Tables

There are two environments, `tabbing` and `tabular`, for producing tabular output. I shall discuss `tabular` environment alone since I find it more useful and I can get away with it practically all the time. As mentioned earlier, `tabular` environment is similar to the `array` environment. The `tabular` environment is more suitable for text material. Let us make a table of physical constants to illustrate the making of the table. The latex file is as shown below and the table appears as shown in Table(1)

Table 1: A list of some of the physical constants

Physical constant	name	value	dimension
c	velocity of light	2.998×10^8	m/sec
e	electronic charge	1.602×10^{-19}	coul
N_A	Avogadro number	6.02×10^{22}	per mole
m_e	Electron mass	9.11×10^{-31}	kg

One can note the similarity between the `table` environment and `tabular` and `array` environments. In `table` environment one can give the caption or title and one can refer to it else where in the text (see later). Note that one has `tabular` environment inside the `table` environment and when one declares the `tabular` environment, one declares the number of columns and how the contents of the columns are placed. This is given in the first and second set of braces respectively. Thus, in the table above, we have four items in each row and first two are left justified (`l`), the third is centered (`c`) and the last is right justified (`r`). vertical lines are drawn by inserting `|` at appropriate places. Horizontal lines are drawn by `\hline` declaration.

Notice that center environment (`\begincenter ... \endcenter`) is invoked within `table` environment. This ensures that the table is centered. If we omit the center environment we will get 2.3. Note that the caption is still centered.

Table 2: A list of some of the physical constants

Physical constant	name	value	dimension
c	velocity of light	2.998×10^8	m/sec
e	electronic charge	1.602×10^{-19}	coul
N_A	Avogadro number	6.02×10^{22}	per mole
m_e	Electron mass	9.11×10^{-31}	kg

The table made below is more complex because in this table we have different number of columns for different rows. Essentially, what we have is, some columns in a row are spanning more than one columns. For example, the first row has two columns, the first spanning one column and the second spanning three columns. This is achieved by using `\multicolumn` command (with the span of the wider column shown in the curly brackets).

Table 3: A Complicated Table

one column	three columns		
item1	prop11	prop12	prop13
item2	prop21	prop22	prop23
two columns		two columns	
item3	prop31	prop32	prop33
And So On			

2.4 Enumerate and Itemize Environments

Many times one has to display a list of items together. One may want to enumerate these or simply put one item on each line with some mark in front of it. The first is generated by `enumerate` environment and the second is generated by `itemize` environment. We have encountered an example of `enumerate` environment earlier. The `enumerate` environment is invoked by typing `\begin{enumerate}` in the first line, followed by the items you want to enumerate and closing the environment by typing `\end{enumerate}`. Each item that is to be enumerated is preceded by `\item`.

Below is an example of itemize environment. The method of invoking the itemize environment is similar to the enumerate environment except that the word enumerate is replaced by itemize.

- Item 1
- item 2
- item 3

One can have nested enumerate and itemize environments. For each nested environment, different symbols are used. Usually, one does not need to have more than two nested enumerate or itemize environments but it is possible to go beyond that. The reader is encouraged to explore the effect of having multiply nested enumerate and itemize environments.

2.5 Figure Environment

Figure environment is used for including figures within the text. Postscript (ps) or encapsulated postscript (eps) files can be imported into the text document. \LaTeX can draw some simple figures but one has some very nice drawing and graphics packages so one might as well use these and make nice-looking figures. For example, xfig and xpaint are available with linux and using these one can make good figures. Also, these packages have a facility of giving ps or eps output. For plotting graphs, one can use gnuplot (a package that comes with linux) or xmgr (a free software package) which can plot 2 and 3 dimensional plots, do labeling of axes etc. These also produce ps or eps output. Below I will give an example of including the figures. In order to include the eps figures in a document you must include the line

```
\usepackageeps
```

after the declaration of the documentclass statement.

This is an example for including an eps file. Some comments are in order. h in square brackets in the declaration line of the figure environment means the figure is to be placed here (i.e. the place where the figure environment begins). Other options are t, for the top of the page and b, for the bottom of the page. The second line



Figure 1: Diagrams contributing to nucleon self energy (a) and σ and ω self energies (b).

declares the length of the figure along x axis. One could give y axis length instead by replacing `epsfxsize` by `epsfysize`. But one should not give both the sizes. The reason is, by fitting both the sizes the figure gets distorted. The third line actually places the eps figure at appropriate place. The name of the file containing the figure is declared in braces. This file should be in the same directory where one is working. Note that \LaTeX makes a box for the figure and places it at appropriate place. Finally, the last two lines are figure captions. With this construct, the figure appears *within the text*. Gone are the days when one had the figures at the end of the paper and figure captions appearing on a separate sheet of paper. Surely, this gives a better appearance to the document and it is also easy on the reader. Incidentally, this figure is made by using `xfig`.

3 Cross References

In a document one often refers to other published or unpublished documents. There are different methods of giving these references and these are defined by the document class. For example, in research reports and articles, the works referred to in the document are listed at the end of the document with each work being numbered or having some identifying characters (say first three letters of the author's name followed by last two digits of the year the work was published). Sometimes the references are given at the end of a chapter or at the bottom of a page (although this style of referencing is vanishing). Another use of referencing is when one is referring to an equation, table, section etc of the same document (cross referencing). \LaTeX has a nice way of handling these two cases. The method used by \LaTeX takes care of

the possibility of future modification of the document.

The method of referring to other works is as follows. The place where a work is referred, one types `\cite{xxx}`. Here xxx is a key to the work one is referring to. There will be many such references to different works within the body of the text. All these references are collected and put at the end of the document (before the `\end{document}` line) within the bibliography environment. The bibliography environment is invoked by typing `\begin{thebibliography}{99}` and ended by typing `\end{thebibliography}`. In between these two lines, the references are listed. Each reference is typed on separate line and the line begins with `\bibitem{xxx}` followed by the detailed reference to the referred document. Note that the key xxx here should be same as the key used while referring to the document in the text. This way of handling the references has two advantages. One is that the reference is referred to by means of a key which is easier to remember and type. This helps if same work is referred at many places. Second and more important advantage is that one can add the text and/or references later in the document but that does not require any change in the earlier referencing. But because of this, the text file has to be compiled twice to get the cross references correct.

For cross referencing equations, tables, figures etc the method used is as follows. Any equation, table, figure etc which is to be referred elsewhere in the text is labeled by means of a command `\label{xxx}`. As in the preceding paragraph, xxx is the key with which the object in question will be referred. The place where the object is referred, one types `\ref{xxx}`.

References

- [1] \LaTeX A Document Preparation System; Leslie Lamport (Addison-Wesley, Reading (85))
- [2] On-Line Tutorial On \LaTeX ; url - http://www.river_valley.com/tug (Produced by The Tutorial Team, Indian \TeX Users Group)