# PHYS-4007/5007: Computational Physics
# Course Lecture Notes
# Section III

Dr. Donald G. Luttermoser
East Tennessee State University

**Version 7.1**

**Abstract**

These class notes are designed for use of the instructor and students of the course PHYS-4007/5007: Computational Physics taught by Dr. Donald Luttermoser at East Tennessee State University.

# III. Preparing Scientific Manuscripts: LaTeX

## A. Introduction.

**1.** The LaTeX document preparation system is a special version of Donald Knuth's TeX program written by Leslie Lamport.

**a)** TeX is a sophisticated program designed to produce high-quality typesetting, especially for mathematical text.

**b)** LaTeX adds to TeX a collection of commands that simplify typesetting by letting the user concentrate on the structure of the text rather than on formatting commands $\Longrightarrow$ LaTeX can be thought of as a TeX macro.

**c)** LaTeX has become the standard formatting style in many different scientific disciplines, especially astronomy and mathematics.

**2.** To be more specific, LaTeX processes an input file containing the text of a document with interspersed commands that describe how the text should be formatted. It produces at least three files as output:

**a)** A *Device Independent*, or '.dvi' file. This contains commands that can be translated into commands for a variety of output devices. You can view the output of LaTeX by using a program such as xdvi, which actually uses the '.dvi' file to display the contents of the dvi file onto an X-Window terminal.

**b)** A *transcript* or '.log' file that contains summary information and diagnostic messages for any errors discovered in the input file.

**c)** An *auxiliary* or '.aux' file. This is used by LaTeX itself, for things such as sectioning.

**3.** For a description of what goes on inside TeX, you should consult *The TeX Book* by Donald E. Knuth, ISBN 0-201-13448-9, published jointly by the American Mathematical Society and Addison-Wesley Publishing Company.

**4.** For a description of LaTeX, you should consult:

**a)** *LaTeX: A Document Preparation System*, by Leslie Lamport, ISBN 0-201-15790-X, published jointly by the American Mathematical Society and Addison-Wesley Publishing Company, 1985.

**b)** *LaTeX for Engineers & Scientists*, by David J. Buerger, ISBN 0-07-008845-4, McGraw-Hill, 1990.

**c)** *LaTeX Line by Line: Tips and Techniques for Document Processing* by Antoni Diller, Chichester: John Wiley and Sons, 1992.

**d)** *A Guide to LaTeX, 4th Edition*, by Helmut Kopka and Patrick Daly, Addison-Wesley, 2004.

**5.** Note that LaTeX is periodically updated. In the references above, the first 3 deal with LaTeX Version 2.09. The last reference above deals with LaTeX2e, the latest standard LaTeX.

**a)** The LaTeX compiler knows which version of LaTeX one is using by the first command found in a LaTeX file:

**i)** The '**documentstyle**' command means that it is a LaTeX Version 2.09 file.

**ii)** The '**documentclass**' command means that it is a LaTeX2e file.

**b)** As such, LaTeX2e compilers also will understand a LaTeX 2.09 formatted file.

**c)** An updated Version 3 of LaTeX began development in the early 1990s but has never been released for use.

**d)** Instructions for accessing LaTeX will be given in class.

## B. LaTeX Commands.

**1.** A LaTeX command begins with the command name, which consists of the \ symbol followed by either a string of letters or a single non-letter. In the above mentioned reference books, arguments contained in square brackets, [], are optional while arguments contained in braces, {}, are required.

**2.** LaTeX is case sensitive. Enter all commands in *lower case* unless explicitly directed to do otherwise.

**3.** We don't have time to go through a complete description of all the LaTeX commands and environments. The best way to learn LaTeX is just to get a copy of somebody's LaTeX file or one of your own LaTeX files and make the appropriate modifications relying on any of the reference books mentioned above.

**a)** I have put a file called template.tex (and the resulting hardcopy file template.pdf) located on the LaTeX page on the Course Web Page

http://faculty.etsu.edu/lutter/courses/phys4007/

Download this file and copy it into a new file, named to whatever is appropriate, and edit it, putting in your own

text. Once downloaded, place this file in an editor (say emacs) and take a look at the LaTeX commands. At the same time, look at the compiled output files (either the 'ps' or 'pdf' files) with either ghostview for PostScript files and Acrobat Reader or PDF-XChange Viewer for PDF files to see how the output has been created from the LaTeX commands.

**b)** The instructions for converting a LaTeX file into a compiled output file will be covered later in this section of the notes.

**c)** It should be pointed out here that these course notes are all written in LaTeX.

## C. Helpful LaTeX Hints.

**1.** Some common problems:

**a)** *Why is there no space between my text and a special LaTeX symbol*? For instance, the last part of this question could have appeared by mistake as: "special LaTeXsymbol?" The difference between these two is that the first appears in this LaTeX file as "special \LaTeX\ symbol" and the second as "special \LaTeX symbol."

**i)** LaTeX interprets the *space* after \LaTeX as the ending delimiter to the LaTeX command and putting more spaces in will make no difference since LaTeX ignores spaces.

**ii)** As a result, a *forced* space was put in with "\ " (backslash and a space) after the \LaTeX command.

**b)** *I don't want numbers with my equations.* Use

```
\begin{displaymath}
 some equation
\end{displaymath}
```

(which doesn't produce equation numbers) instead of the more traditional

```
\begin{equation}
 some equation
\end{equation}
```

(which produces equation numbers).

**c)** *I wish to double-space my paper, how do I do that?* Right after the \begin{document} statement, insert the command \baselineskip=22pt if you are using the 11 point font (use 24pt if you are using the 12 point font, see template.tex for an example).

**d)** *I want to undo paragraph indentation and place additional space between my paragraphs.* In the *preamble* (*i.e.*, commands prior to the \begin{document} command), include \setlength{\parindent}{0pt} followed by \setlength{\parskip}{22pt} if your font size was set to 11pt or \setlength{\parskip}{24pt} if your font size was set to 12pt.

**e)** *I have inserted a horizontal space with hspace but LaTeX ignores it.* LaTeX will not put vertical or horizontal spaces in if it thinks that such spaces will make the manuscript *look* sloppy. To force LaTeX to put in a space, use, for instance, \hspace*{5mm} instead of \hspace{5mm}. The '*' forces LaTeX to use the space no matter what.

**2.** Floating figures and tables.

**a)** Whenever you use the \begin{figure} \end{figure} or the \begin{table} \end{table} environment, LaTeX will place such environments in a location that it feels is best for the appearance of the manuscript. See the following subsection III.D for details on including figures.

**b)** Such environments are called *floating* environments.

**c)** Many times LaTeX will place a figure or a table in a location that you do not like. There are ways to override LaTeX. However LaTeX may still ignore your attempt to override it. Then you must reposition your placement of the environment in your text.

**d)** You override LaTeX placement by placing a [t] (for top of page, the default), [b] (for bottom of page), [h] (for current location), and [p] (on a separate page of floats with no text at the end of the manuscript). These location markers are put directly after the *begin* command: \begin{figure}[h].

**e)** Here are the default rules that LaTeX uses when deciding on a location for a float:

**i)** It is printed at the earliest place that does not violate subsequent rules, except that an [h] (here) position takes precedence over a [t] (top) position.

**ii)** It will not be printed on an earlier page than the place in the text where the figure or table environment appears.

**iii)** A figure will not be printed before an earlier figure, and a table printed before an earlier table.

**iv)** It may appear only at a position allowed by the *pos* argument (*e.g.*, t, b, h, p), or if that argument is missing, by the default tbp specifier.

**v)** Placement of the figure cannot produce an overfull page.

**vi)** The page constraints determined by the formatting parameters are not violated.

**f)** Since both figure and table environments are floating arguments, they are said to be *fragile*.

## D. Including Figures in a LaTeX File.

**1.** LaTeX can set up a figure environment where it produces a figure number and a caption, as well as, leave enough space for the figure to reside. However, LaTeX itself does not place a figure into the document itself. Instead, LaTeX communicates *picture* information to the software that generates the final hardcopy version of the paper (*e.g.*, dvips or dvipdf in Unix/Linux) — see below.

### 2. Including Figures on a Unix/Linux Computer.

**a)** This *picture* information is passed to the hardcopy software with the special command. The best way to understand its use is through example:

```
\begin{figure}
\vspace*{3.2in}
\caption{This is the 1st figure caption.}
\special{psfile=figure.eps  hoffset=20  voffset=10  hscale=80
    vscale=80}
\end{figure}
```

**i)** psfile: This is the name of the *encapsulated* PostScript file that is to be inserted in the manuscript. On our Unix/Linux machines, you can make either a PostScript or PDF hardcopy file of your document, which can then be sent to the printer in Brown Hall 264. Note that PDF is the preferred format now-a-days for the final document format.

- **PostScript** is a programming mark-up language that instructs PostScript printers how to generate a printed page (see §III.E).

- Normal PostScript files contain pictures that can be output directly to the printer — they have eject page and page set-up commands inside of them. These files usually have names that end with a .ps suffix. (Note that Microsoft refuses to acknowledge that PostScript exists. As such, one must use a GhostView utility to view and print PostScript files under the Microsoft operating systems.)

- **Encapsulated** PostScript files do not have the eject page and page set-up commands. These pictures are designed to be included in a PDF file or another PostScript file with such printer control commands. These file names end with a .eps suffix.

**ii)** voffset: Tell dvips/dvipdf to move the picture vertically in point (=1/72 inches) units. Positive raises the picture, negative lowers it.

**iii)** hoffset: Tell dvips/dvipdf to move the picture horizontally in point (=1/72 inches) units. Positive is towards the right, negative towards the left from

the default position, which is defined by whatever LaTeX environment you are in.

**iv)** vscale: Same as hscale (see below) except in the vertical direction.

**v)** hscale: Expand or shrink the size of the image horizontally that is stored in the file in units of percentage. Numbers less than 100 shrink the image, greater than 100 expand the image.

**b)** Note that one should create encapsulated PostScript figures in **portrait** format. However, if you mistakenly create it in **landscape** format, one can *rotate* the image with the angle keyword in special. For instance, angle=90 will rotate the image by 90° in the counterclockwise direction. Negative angles rotate in the clockwise direction.

**c)** One should have a separate \begin{figure} \end{figure} environment for each figure that you want to display. Note, however, that one can place 2 or more figures in one figure environment by using 2 or more special commands. Note that the following will place 2 figures in 2 separate encapsulated PostScript files in the same figure environment.

```
\begin{figure}
\vspace*{4.0in}
\caption{(a) Figure on left and (b) figure on the right.}
\special{psfile=figure1a.eps  hoffset=-20  voffset=10  hscale=40
    vscale=40}
\special{psfile=figure1b.eps  hoffset=320  voffset=10  hscale=40
    vscale=40}
\end{figure}
```

## 3. Including Figures in PCTEX on a Windows Machine.

**a)** One of the best software available to run LATEX in the Microsoft world is PCTEX. In this utility the "special" command has a slightly different syntax, and unlike the Unix/Linux version of LATEX, you have the ability to include bitmap (.bmp) images, gif (.gif) images, jpeg (.jpg) images, and a few others, as well as encapsulated PostScript. Here are some examples:

**i)** Include an encapsulated PostScript file:

```
\special{eps:c:/figures/figure1.eps x=2cm y=5cm}
```

Note that one must use a "slash" (/) instead of a "backslash" (\) to delineate directories in the LATEX \special command (even though Microsoft Windows actually uses the backslashes). The reason for this is that the "backslash" is a special control character in both TEX and LATEX. Here, there are no offsets in the special command and the sizes are given in actual lengths (x and/or y) measured in either centimeters (cm) or inches (in).

**ii)** Include a bitmap file and offsetting it by 1 inch:

```
\hskip1in\special{bmp:c:/figures/figure2.bmp
  x=3cm  y=4cm}
```

**iii)** Include a gif file and offsetting it by 1.6 inches:

```
\hskip1.6in\special{bmp:c:/figures/figure3.gif
  x=2.5in  y=5.6in}
```

Note that we still use the "bmp:" keyword for the gif file.

**iv)** Include a jpeg file and offsetting it by 0.5 inches, and include a 0.5 inch space between the previously written text and the beginning of the image, then include and additional 3 inches for the image, and finally put a 0.5 inch vertical space between the end of the image and the next line of text:

```
\vskip.5in
\vskip3in
 \hskip.5in\special{bmp:c:/figures/figure4.jpg
    x=5in  y=3in}
\vskip.5in
```

note that the length in the second '`vskip`' must be the same as the length in the '`y`' keyword in the `\special` command. Once again, use the "bmp:" keyword for the jpeg file in the "special" command.

**b)** A special note is needed here and will be repeated in §III.H of these notes. Besides the use of '/' (slash) instead of '\' (backslash) to delineate directories, PCTEX also insists that there be **no spaces** in either the figure file name and the entire directory path name where the file is located.

**c)** If your figure file is located in the same directory (*i.e.*, folder) as your LATEX file, you do not have to put the full-path name of the file in the special command.

## E. PostScript.

**1.** PostScript is a page description language from *Adobe*. It is actually a programming language that instructs a PostScript printer (or graphics display device through a third-party software such as *GhostScript/GhostView*) how to print a page of text or graphics information. A good web site that describes PostScript in detail can be found at http://www.tailrecursive.org/postscript/ (this

link is also on the Course Web Page). This site also gives a good tutorial in PostScript programming.

**2.** There currently are 3 versions of PostScript. As a result of this, a PostScript file will not necessarily be able to print a PostScript printer.

**a)** PostScript Level 1 is the earliest version. It was introduced by *Adobe* in 1985 and first appeared in the *Apple* LaserWriter.

**b)** PostScript Level 2 came out in the early 1990s. It contained a larger selection of fonts and allowed for color printing.

**c)** PostScript Level 3 came out in the early 2000s. It contains even a larger selection of fonts and allows for direct conversion between PostScript (PS) and PDF (see below) and PDF to PS.

**3.** PostScript is a registered trademark of *Adobe Systems Incorporated*. The copyright to the PostScript language is also held by *Adobe Systems Incorporated*.

**4.** At some point, you may want to include some nice PostScript image into a document. There are a number of problems associated with this, but the main one is that your page layout program needs to know how big the image is, and how to move it to the correct place on the page. Encapsulated PostScript (EPS) is that part of Adobe's Document Structuring Convention that provides this information.

## F. Portable Document Format (PDF).

**1.** PostScript is actually a programming language which instructs a device how to draw the contents of a printed page, whether it be text, figures, and/or images. Since PostScript is simply ASCII text, such a program can be written from a standard text editor.

**2.** PDF is similar to PostScript since it, too, is a format that instructs a device as to how to draw the contents of a page, but one need a *font-end* program from *Adobe* called **Acrobat** to write the PDF program. Acrobat then creates a binary file, called a 'PDF' file that contains the program.

**3.** In PCTEX, one can compile and 'export' their document directly to a PDF file.

**4.** In Unix/Linux, one creates a PDF file from a DVI file with the Unix /Linux dvipdf command (see §III.G).

**5.** Note that if you are putting LATEX compiled documents on the web, it is better to use PDF format over PostScript. This is due to the fact that almost every web-browser has a link to Acrobat Reader which is launched in the browser if somebody double-clicks on the link to your PDF file. A double-click on a PostScript file will cause the browser to download the file onto your local disk (unless you have ghostview installed), hence it would not be viewable from the browser.

## G. Document Preparation on a Unix/Linux Machine.

**1.** Once you have created a LATEX file in an editor (like emacs), one should follow these steps. As mentioned before, the best way to write a LATEX file is to start from another LATEX file that has

already been made, and modify the contents to what you wish to write. For our example here, let's assume our LaTeX file is called template.tex.

**2.** The first step you should take when compiling a LaTeX file is to run the file through the spell checker, *e.g.*,

```
spell template.tex
```

while operating in Unix. You will note that all of the LaTeX commands and special symbols will be displayed as spelling errors. Just ignore those and look closely for real spelling errors.

**3.** Next, compile you file in LaTeX by issuing the command:

```
latex template
```

without the .tex filename suffix. No doubt you will have numerous LaTeX errors in it. When LaTeX finds an error, you can just enter an "e" and LaTeX will send you file into the emacs editor to the location where it found the error. You can then correct it, save the file, and rerun the latex command. If you don't feel like editing the file at present, you can enter "q" to tell LaTeX to quit the current compile. In PCTeX, just use the compile button to compile the current file in the editor box.

**4.** Once your file has successfully compiled, LaTeX creates a dvi file. On Unix machines, you can view this dvi file before making a PDF or a PostScript file. One would do this to check to make sure the pages have been formatted the way you want them (for instance, check to make sure a table does extend off of the page). If you are at a terminal running X-Windows, enter the following:

```
xdvi template &
```

again without the .dvi filename suffix. Include the "&" symbol

to put this process in background. This way, you can edit your file if you need to while investigating the output with xdvi.

**5.** Once you are happy with the way your paper looks, you can make the final hardcopy (*i.e.*, PDF or PostScript) file with the command:

```
dvipdf template
```

or

```
dvips template
```

again without the .dvi filename extension.

**a)** If you have included encapsulated PostScript file figures, the *scale* and *offset* values that you have chosen for the figure might not be correct — the figure might not be centered or in the proper orientation.

**b)** Before wasting paper and printing out this PostScript file, it might be a good idea to view it to check on your figure placements. You can display a PostScript file with either the ghostscript (gs) software package, or the ghostview (gv or ggv) software package, which is a widget-based package that drives ghostscript.

**c)** To run ghostscript from the Linux on an X-Window terminal, enter

```
gs template.ps
```

and you hit <return> every time you wish to view the next page. Enter quit at the GS> prompt to leave ghostscript.

**d)** To run ghostview, enter

```
gv template.ps     or     ggv template.ps
```

and control the page selection with the widget control.

Note that you can print selected pages of a PostScript file from ghostview by *marking* the desired pages and *printing the marked* pages.

**e)** Edit your special keywords and recompile and re-dvips or re-dvipdf your file until your figures are positioned correctly. Note that ghostview can also display PDF files.

**f)** Ghostscript and ghostview also are installed on the MS Windows machines and can be found in the GS folder on the background screen of these machines.

**6.** You can then print out your PostScript file to the default printer by issuing the Unix command:

```
lpr template.ps
```

**7.** Finally note that one can make a PDF file directly from a PostScript file too. All Unix/Linux machines that have LaTeX, dvips, gs, gv, and dvipdf on it will also have the software to convert PostScript to PDF. For this conversion, use

```
ps2pdf template.ps template.pdf
```

— once completed, you will have a new file called template.pdf in addition to your template.ps file. **Note that the PDF file cannot be sent to a printer with the 'lpr' command!** You must send it to the printer through the Acrobat Reader or ghostview utility.

## H. Document Preparation on a Windows Machine Using PCTeX.

**1.** This subsection assumes that PCTeX has been installed on the Microsoft Windows computer that you are using.

**2.** Your file should be created and edited in the PCTEX GUI which is started by clicking on the PXTEX icon on your PC.

**3.** We follow some of the same steps here as we did for Unix. First, run a spell checker. PCTEX has its own spell checker built into the GUI, so on the PCs, just use that.

**4.** Make sure that LATEX is selected in the box next to the Typeset button. Click the Typeset button. This will open up a new window which contain the contents of the DVI file (*i.e.*, the appearance as seen when printed).

**5.** Fix any problems from the edit window that you may discover in the DVI output. Then re-typeset.

**6.** In PCTEX, just use the print button on the GUI to print the document. Note that the HP printer in Brown Hall 264 can print back-to-back. You should be able to control the two-sided printing (if you desire it) from the "print properties" GUI in PCTEX.

**7.** Finally, you can also make a hardcopy of your document in PDF or PostScript format. Just use the Export menu item under the File pulldown icon at the upper right of the GUI.

**a)** Note that for your written assignments requiring the use of LATEX, just follow the instructions above to create a PDF file from your LATEX file.

**b)** One simply then needs to attached the '*.pdf' and '*.tex' files (and any image files used in the LATEX file) to the email message and send it off to **lutter@etsu.edu** by the deadline of the assignment.