Inessential LaTeX *

(Revision: 4.4)

The Student Information Processing Board

July 7, 2004

Contents

1 Introduction

1.1 What is LaTeX ?

LaTeX is a complex text-processing system often recommended by the SIPB and used for scholarly publishing all over academia, and by many of your professors.

There are LaTeX manuals available for use in the SIPB office, and you're welcome to drop by the office with a question. There are also manuals available for short-time loan from the OLC (On-Line Consultants) office (N42) and some available for withdrawal from the MIT Libraries. There is also plenty of documentation on the World Wide Web.

If you have any comments, additions or corrections to this manual, please send them to `sipb@mit.edu` or drop by the SIPB office (W20-557) and let us know.

1.2 LaTeX on Athena

You can run LaTeX on any Athena-ized workstation. Instructions for setting up your environment and running LaTeX on these platforms are given in section 4 of this document. To get more information on using other software available on the Athena workstations, see the *Getting Started on Athena* document available from MIT Copy Tech, and the *Inessential Guide to Athena* available from the SIPB office (W20-557).

Versions of LaTeX can be obtained many other types of machines. However, many LaTeX distributions for popular platforms are commercial packages, and their features and usage may differ. Hopefully, your LaTeX package will come with some instructions for setting it up and running it.

The version of LaTeX described in this document is version LaTeX 2_ε, which is the version currently supported by Athena and SIPB. Version 2.09 is still available in the `newtex` locker under the name `latex209`. Section 11 of this document describes the differences between the two versions.

2 Getting Started

This section describes how to typeset a simple paper in the standard way for processing by LaTeX. All of the information here is also presented elsewhere in this document, but with a lot more detail; think of this section as "the short form" to using LaTeX.

The first thing you need to do is to create your LaTeX source file. This file contains all the text for your paper as well as LaTeX commands (to be described shortly). You can use any editor you wish to create the file; documentation on how to use text editors is available from Athena. Emacs includes special code set up to make using LaTeX easier, and is a good first choice. You can name your LaTeX file anything you wish; the only requirement is that it end with the four characters `.tex`.

The general format of a LaTeX file is shown in the following example.

```
\documentclass{article}
\usepackage{doublespace}
\usepackage{fullpage}

\title{A Sample Paper}
\author{Melissa I. Thompson}
\date{June Sixteenth, 1904}
% \date{\today} is a useful construct to know about too.

\begin{document}

\maketitle

It was a dark and stormy night, and all of the Athena fileservers had
crashed.  Poor me, I had to write a paper for one of my HASS-D
classes.

Luckily, I was able to use a wonderful text formatting program called
\LaTeX{}, and everything worked out fine.
\end{document}
```

Here is a description of each part.

- The first line of every LaTeX file contains a `\documentclass` command. This command tells LaTeX the general style of your document as well as the style "options" you wish to use. The word in between curly braces (`{` and `}`) is the main style. You can specify options for the style in square brackets before the main style. For example, `\documentclass[12pt]{article}` would use article document class in 12 point type.
- The `usepackage` commands specifies additional packages or styles to use. For example, the `doublespace` package tells LaTeX to doublespace your document, and the `fullpage` option says to set the margins so you have one inch of blank space on all sides. You can supply options to these packages as well in the same way that you did to the main package (ie, in square brackets).
- The `\title`, `\author`, and `\date` commands are used to specify the title, author and creation date of the document. The default date is `\today` (meaning that you don't need to give the `\date` command if you just want to use the current date).
- The `\begin{document}` line marks the beginning of the main part of the document. The section before this line is called the *preamble*. More on this later.
- The `\maketitle` line tells LaTeX to insert a title page in the beginning of your document. The data from each of the `\title`, `author`, and `\date` commands is used here.
- `\LaTeX{}` is a special command in LaTeX, used to represent the appropriate magic so that you see "LaTeX" instead of "LaTeX."

- The `\end{document}` line marks the end of the main part of the document. In LaTeX, something between `\begin{`*foo*`}` and `\end{`*foo*`}` is said to be in the *foo* environment (more on environments later).

3 Creating your Document

For LaTeX to know that you are writing a document, you have to give it some information first. This section describes what you need to have in a file to run it through LaTeX.

3.1 In the Beginning...

A typical file, such as a term paper, will begin like this:

```
\documentclass{article}

\begin{document}
```

This tells LaTeX the following things:

- This example is an `article`. The article style may be used for any kind of standard paper. An `article` is shorter than a `report`.[1] The `book` style also exists. It is meant for real books. These options control section headings and tables of contents and other things.
- The main text is printed in a 10 point font by default. There are several text sizes available: 10, 11, and 12 point. This document is printed in 11 point. The various *type sizes* within a document are relative to this base size, as described in subsection 6.3. If no base size is specified, 10 point is assumed. To change to a base size of 11 point or 12 point, change the document class with an optional parameter, placed inside square braces immediately after the command name, like this:

  ```
  \documentclass[11pt]{article}
  ```

- There are many other optional parameters which can be used, some of which will be mentioned later. They are all entered inside the square brackets and are separated by commas, with no spaces in between. For example,

  ```
  \documentclass[twoside,11pt]{report}
  \usepackage{doublespace}
  ```

  sets the document up as an 11-point, double-spaced, double-sided report.

[1] This LaTeX guide is a `report`.

- The space between the `\documentclass` and the `\begin{document}` commands is called the *header area* or *preamble*. Various control parameters, such as the layout of the title page (if you want one) go here. The actual text of your document begins after the `\begin{document}`.

There are many packages and options which may be included with the `\usepackage` line or as an option to `documentclass` . Packages can also take options, which are specified in brackets before the package name. You can include multiple packages with a single `\usepackage` command. For example, `\usepackage[red,blue]{foo,bar}` would include the packages `foo` and `bar`, giving the `red` and `blue` options to both. Some of the common packages and options are:

doublespace (package) Causes the document to be doublespaced and defines the `\doublespace` and the `\singlespace` commands for changing document spacing.
fullpage (package) This is the easiest way to set normal margin widths.[2]
simplemargins (package) Enables commands to simplify modifying the margins. This gives more flexibility than fullpage (see above) but takes a little more effort to use[7].[3]
twocolumn (option) Will print text in two vertical columns, side by side.
graphicx (package) Allows you to include Encapsulated PostScript in your document.
times (package) To format the whole document in Times Roman font.
palatino (package) To format the whole document in Palatino font.
helvetica (package) To format the whole document in Helvetica font.
ncs (package) To format the whole document in New Century Schoolbook font.
avantgarde (package) To format the whole document in Avant-Garde font.
bookman (package) To format the whole document in Bookman font.
quiet (package) Will tell LaTeX to run in "quiet" mode, suppressing warning messages.
latexsym (package) Allows you to use symbols that are no longer defined by default in LaTeX 2_ε. See Figure 16 for a list of these symbols.

3.2 Title Page

To put a title page into a document, you could type the following:

```
\documentclass{report}

\title{Inessential \LaTeX{}}
\author{The Student Information Processing Board}
\date{\today}

\begin{document}
\maketitle
```

[2]Both fullpage and simplemargins (see below) are explained in detail in Section 6.1.
[3]Requires a special `TEXINPUTS` environment variable discussed in Section 4.

Creating a title page is a two step process in LaTeX. First you give it the information you want it to put on the title page and then you tell it to make the title page. The `\title`, `\author` and `\date` commands do not generate text—they merely specify what `\maketitle` will place on the title page. If you use `\maketitle`, you must specify a `\title`, but any or all of the others may be omitted. The `\maketitle` command generates the title page. You must place it after the `\begin{document}`. In the `article` document style, the title "page" is at the top of the first page of the document. To put the title "page" on a page by itself place a `\newpage` command after the `\maketitle`. In the `report` and `book` styles, it is on a separate page at the beginning.

LaTeX will automatically number pages, but often you don't want a page number on the title page. To eliminate the page number on the first page, put `\thispagestyle{empty}` before the `\maketitle`.

It is possible to number pages alphabetically or with roman numerals. It is also possible to change the current page number. See the section of the LaTeX manual on page styles.

You can also generate a title page using the `titlepage` environment. Anything placed in this environment will be placed on the title page (which will be a separate page even if you are using the `article` document class). Another good way to create a title page is to do it yourself, probably using commands like `\begin{center}`, `\large{}`, and `\newpage`. Using this method, LaTeX won't know the difference between your title page and any other page, but it may be easier to get what you want.

3.3 Body

The body of the document can be normal text. Paragraphs are separated by completely blank lines or by the `\par` command.

Special characters When entering normal text, you must watch out for a few things. Some characters are reserved for commands to LaTeX, and are treated as such unless you tell LaTeX to treat them as normal characters. To use the characters

```
# $ % & _ { }
```

outside of a `verbatim` environment (see Section 6 for more information on environments), you must precede them with a backslash (\), like this:

```
\# \$ \% \& \_ \{ \}
```

These characters are also reserved:

```
~ ^ \ < >
```

Often, if you think you want one of these characters, you probably want to do something else.

- ~ can be produced in several ways. If you want a tilde over a character, like ñ, you can type `\~{n}` or `\~n`. If you want a mathematical ∼ you can type `$\sim$` (which stands for *sim*ilar to). More often you will want the ∼ within another math expression anyhow; for more detail see Section 8 below.
- ^ can likewise mean circumflex, as in ô which is produced with `\^{o}` or `\^o`. In math mode you would use `\hat{o}` to indicate a unit vector.
- \ is not used very often in normal writing, but you can type `$\backslash$` if you want to use it.
- < and > can be written as `$<$` and `$>$`. If you insert these symbols in your document without quoting them as described here, they will appear as upside-down exclamation points (¡) and question marks (¿).

Quotation Marks Ever notice how typeset material always has nice quotation marks, for example, “this” instead of "this"? LaTeX can generate these fancy quotes with just a little effort on your part. The “ character, used to start a quotation, is made by typing ``. Note that these are reverse quote, or “back-tick” characters, usually found on the upper-left hand side of PC-style keyboards (however, the Sun Type 5 keyboard puts it in the upper right). The " character is made by typing ''. In Emacs, the default mode for LaTeX files (called `tex-mode`) will generate the proper characters when you press the " key. LaTeX mode is automatically loaded when you load a file ending in `.tex` that has a `\documentclass` command near the top.

Divisions Text can be divided into parts, chapters,[4] sections, subsections, subsubsections, paragraphs, and subparagraphs. Each division has its own LaTeX command (`\section`, `\subsection`, etc.). These commands produce the appropriate headings and entries in the table of contents, as well as numbering the divisions. For normal use, you probably want to use `section`, `subsection`, and `paragraph`. The section `paragraph` is different from the command `\par`. A section command places the title you give it in bold at the beginning of the paragraph. The `\par` command simply marks the beginning of the paragraph, as would a blank line.

For example, in this document you are currently reading:

```
\section{Creating Your Document}
... things that are in the section ...
\subsection{Body}
... things that are in the subsection ...
\paragraph{Divisions}
... things that are in the paragraph ...
```

If you don't want the division to be numbered in the document, include an asterisk (`*`) after the division name (e.g. `\subsection*{Body}`).

[4] Chapters are available within `book` and `report` but not in `article`, so that an article can be included as a chapter.

Table of Contents Since LaTeX already has all of this information about the structure of the document, it can easily collect it and build a table of contents. It will only do this if you explicitly tell it to. LaTeX will put the table of contents wherever you include the line

```
\tableofcontents
```

in your document. To put the table of contents at the beginning (where people expect to find it) just put it in after the `\maketitle` command. You should run LaTeX at least twice after you add the table of contents, since it takes two or three tries to get the table of contents right—once to write all the page numbers out to a file[5], and a second time to include those numbers at the beginning of the document, and possibly a third time if the number of pages the table of contents occupies changes.

When you make major changes, LaTeX may tell you that you need to run it twice. It is always a good idea to do this before you print the final version of a document, just to make sure everything is correct.

3.4 Environments

If you want to change how LaTeX treats a certain section of text (for instance changing the type style), you have to specify what text. In LaTeX lingo, you need to indicate the *scope* of the command you are using to change the *environment*. For small sections of text, this is done by starting the *environment* with an open brace (`{`) character, followed by the command to change the defaults, and ending with the close brace `}` character.

For example, to make appear in a larger font, you could type `{\Large larger font}`. This limits the *s*cope of the large command to the environment within the braces.

An environment can also be delimited with a *begin–end* pair. This is quite useful for larger sections of text. For example, to center multiple lines of text,

like
these two,

you could use the following lines:

```
\begin{center}
like\\
these two,
\end{center}
```

[5]This information is stored in the file ending in `.aux`, which some people remove between invocations of LaTeX. Doing this may save space, but it causes LaTeX to have to repeat a lot of work, so you should avoid doing this until you're sure that you won't be re-running LaTeX again in the near future.

Note the \\ following the first line. This is a command which tells LaTeX to break the line at that point, rather than filling the text completely between the two margins like it normally does. This allows you to force lines to end when you want them to, rather than when LaTeX wants them to.

3.5 Ending the Document

The last thing to put in the document is the line

```
\end{document}
```

LaTeX ignores anything which follows the `\end{document}` command. The only text after this should be comments, since you have now finished your document.

4 Processing your Document

Once you have written your file and entered all the necessary LaTeX commands, you have to process the file before previewing or printing it. This separation of editing and processing is true for almost any text formatter, and is the major difference between text formatters and word processing systems. FrameMaker and EZ both lean more towards the word processing, WYSIWYG[6] style of editing[7]. A graphical frontend to LaTeX named LyX is also available in the `outland` locker.[8]

4.1 Running LaTeX

If you have used any special `.sty` files (the optional packages or options mentioned previously) which are only available from SIPB, you must first set an environment variable to let LaTeX know where to go to find these files. The command is:

```
attach sipb
setenv TEXINPUTS .:/mit/sipb/lib/tex/macros/:
```

These lines can be placed in your `.environment` file. (See the Athena document "*Essential Dotfiles.*" for more details). The trailing colon represents a blank path entry, which LaTeX replaces with the default path entries.

You can process a LaTeX file by typing

[6] What You See Is What You Get

[7] The problem with WYSIWYG systems is that often what you see *isn't* really what you get, and the system usually doesn't have any good way to deal with these problems. We prefer to think of it as WYSIRN (What You See Is Really Neat) vs. WYSIWYAF (What You See Is What you Asked For) systems.

[8] The `outland` locker is maintained by the SIPB and many other contributors and generally holds programs that are unsupported or in testing. This means that you probably don't want to be using LyX (or anything else from `outland` for important work. Please read the file `/mit/outland/README` for more information on the support policy.

```
latex filename.tex
```

where `filename.tex` is the name of your file that you want to process. LaTeX produces an output file named `filename.dvi` (a **d**e**v**ice **i**ndependent file), which can be converted and printed on various types of printers.

Running Previous Versions of LaTeX The supported version of LaTeX on Athena is currently LaTeX 2_ε, but there are usually separate installations of older versions available. Currently, LaTeX 2.09 is available in the `newtex` locker. You can access it by typing:

```
add newtex
latex209 filename.tex
```

at the `athena%` prompt.

See Section 12 for information on converting LaTeX 2.09 documents to LaTeX 2_ε.

4.2 Previewing the Final Output

It is usually convenient to use the graphic display of your workstation to preview your document before printing it. The command `xdvi filename &` creates a window which shows you what the file will look like when it is printed on the printer. After you position the `xdvi` window on your screen, it should show the first page of the document. You can use this like the `more` command: hit space to go forward a page, and `b` to go back a page. This way you can go back and forth between the `xdvi` previewer and the LaTeX source, rerunning LaTeX and looking at the `xdvi` display each time. (`xdvi` automatically notices that you've changed something, and displays the newest output.)

4.3 Printing the File

Printing on Athena When you are ready to print out the file, you must convert from dvi format to the format of the printer you are using. To translate `filename.dvi` to PostScript[9] and send it to your default printer, use the command:

```
dvips filename.dvi
```

If you have no default printer[10] or you want to send the printout to a specific printer use the command:

`dvips -P`*`printername`* `filename.dvi`

replacing *printername* with the name of the printer to which you want to send your document.

[9] PostScript is the "language" understood by Athena printers, and many others.

[10] You can check this by typing `lpq` and noting what printer it lists.

It is possible to obtain the intermediate PostScript file with the command

```
dvips -o filename.ps filename.dvi
```

It is possible to select particular pages out of a document, if you are making changes and want to see what they will look like on paper (note that `xdvi` gives a very accurate rendition, and should be enough for most purposes). The command

```
dvips -pfirst -llast -Pprintername filename.dvi
```

will print from the page numbered "first" to the page numbered "last."

To get more information on `dvips`, consult the manual page (typing `man dvips` at your athena prompt). A more detailed document can be found by typing `texdoc dvips`. Note that the older `dvi2ps` program no longer exists on Athena.

4.4 Producing PDFs and Web Pages

The Adobe Portable Document Format is now the lingua franca of much of the world, and LaTeX provides you with a way to produce them. When you are ready to print a document, you are also ready to make a PDF of it. Typing `pdflatex filename.tex` will go through a similar process and produce a file named *filename.pdf* with the same appearance and typesetting as the PostScript file the section above showed you how to make. If, however, you run into a bug in `pdflatex`, you can bypass it by typing `latex filename.tex` to produce a DVI file, and then making a PDF file out of the DVI file. In the `ghostscript` locker there is the `dvipdf` utility and among the regular Athena utilities there is `dvipdfm`. Type `dvipdf filename./dvi` or `dvipdfm filename.dvi` to produce *filename.pdf*. If the resulting PDF gets a mathematical character wrong (as it may), a more circuitous route is still available: from the DVI, make a PostScript file using `dvips -P pdf -G0 file.dvi`, and then `distill file.ps` to make the PDF. (This last utility is available in the `acro` locker.)

Since the PDF format allows for things that PostScript does not, like active Web links, and forms, you may not be able to use the DVI to PDF routes. So here are some more notes on using `pdflatex`. Unlike `latex`, `pdflatex` cannot import EPS files for images. To put figures in your document, you can make them available to `pdflatex` as their own PDFs, using `epstopdf`. More on that in Section 9.1. To encode hypertext references in your PDF, you can use the `hyperref` package and use the macro `\href{URL}{text}` in the same way you write the equivalent HTML prose. And finally, to put in your own PDF bookmarks, use `\pdfbookmark[n]{title}{internal_label}`, with `internal_label` serving as a label for any `\ref` references you want to put in.

The `latex2html` program, available on the Web and on Athena in the `infoagents` locker, is a Perl script that generates a set of HTML pages from a Latex document. It can make a single Web page, or a set of them, with a page for each section, subsection or so on and so forth. To generate a Web page set based in the directory `dstdir`, from a document `file.tex`, type `latex2html -d dstdir file.tex`. This will generate a whole tree of HTML files for each subsection. The control how deep the tree should go, there is the `-split` option. For a single HTML page, set it as `-split 0`. For more information, type `latex2html -help`.

4.5 Landscape (or Sideways) Mode

If you want to print your document "sideways," with the text running parallel to the long edge of the paper, simply include

```
\special{landscape}
```

in the preamble of your document. You also need to set the margins to correspond to this new paper size (11x8.5″ instead of 8.5x11″). See Section 6.1 for details on changing them.

4.6 Using Both Sides of the Paper: Duplex and Tumble Modes

With `dvips`, you can print in *duplex* or *tumble* mode by using the `-h` option:

```
dvips -h duplex filename.dvi
```

or

```
dvips -h tumble filename.dvi
```

If you always want your document to be printed two-sided, you can include one of the following commands in your document preamble:

```
\special{header=duplex}
\special{header=tumble}
```

The difference between duplex and tumble is whether pages are printed with opposite sides sharing a "top" of the page (duplex), so that turning the page side-to-side leaves both sides correctly oriented (as in a book), or whether these pages are flipped vertically (tumbled), so that turning the page up leaves both sides correctly oriented (as in a legal notepad). The orientation will obviously be somewhat different if the printout is landscape mode instead of portrait mode. For most uses, the duplex mode is the more fitting.

5 Error Messages

The error messages produced by LaTeX are the most confusing part of using it. Most people have trouble understanding what LaTeX is telling them. This section describes some of the more common errors, what they mean, and how you can fix the problem.

In general, when an error occurs you should hit return to see if LaTeX will keep on going, or enter `x` to exit. If LaTeX is asking for a filename, `Control-D` will help exit LaTeX. If `x`, `^C`, and `^D` do not return you to your `athena%` prompt, type `^Z`. In order to kill the *stopped* process you need to do the following (assuming that you use `csh` or one of its descendants):

```
athena% kill %latex
athena%
[1]  + Terminated           latex
athena%
```

By hitting <return> when you get your prompt back you should get the message that LaTeX was indeed killed.

5.1 LaTeX errors

There are two kinds of errors: LaTeX errors and TeX errors. TeX is the language in which LaTeX is written, and some of the basic errors come from there. All errors have an error message which starts with an exclamation point. A LaTeX error can be distinguished from a TeX error by the following:

```
! LaTeX error: <error message here>

See the LaTeX manual or LaTeX Companion for explanation.
Type  H <return>  for immediate help.
 ...
```

The error message is printed on the next line, which starts with a number and then the text of the line that caused the error. The part of the number after the decimal point is the line number where the error occurred. If you cannot figure out what the problem is by looking at the error message, you can go to the offending line in the original document and look at it. The problem may lie a few lines above the one which LaTeX complained about. The error message should make more sense after you have looked at the offending text.

Some of the more common error messages are:

```
! \begin{...} ended by \end{...}.
```

This is usually caused by forgetting to end an environment which was started by a `\begin` command.

```
!  Can be used only in preamble.
```

Certain commands can only be entered before the `\begin{document}` statement. This part of the file is called the "preamble." Move the offending statement to the preamble and try again.

```
!  Environment ... undefined.
```

An environment defined in a `\begin` or `\end` statement is not recognized. This is usually caused by a spelling error.

```
! Missing \begin{document}.
```

All files must have a `\begin{document}` before any text which is intended to be printed.

5.2 TeX errors

```
! I can't find file `...'.
```

This often indicates that the file name was mistyped or that you are not in the correct directory. TeX will wait for you to type in a file name. If a mistake was made, and you want to abort the command, you will need to hit `^D` which will cause an "Emergency stop" when TeX is waiting for you to type something.

```
! Missing { inserted.
! Missing } inserted.
```

```
! Missing $ inserted.
```

This usually occurs if a command was typed which can only be used in math mode. LaTeX can be allowed to continue, but the output will look strange. The offending command should be put into math mode.

```
! Undefined control sequence.
l.7 \auhtor{The Student Information Processing Board}
```

The command `\auhtor` on line seven is misspelled and so LaTeX does not recognize it. If LaTeX does not recognize a command you used, but you think it exists, check the documentation again to make sure you are using the command correctly.

If you cannot figure out what is wrong, give the SIPB office a call,[11] or try the Athena Consultants.[12] The Zephyr `help` class is also often useful; see the document *Inessential Zephyr* for more details on using zephyr classes and instances.

5.3 Warnings

Warnings do not cause LaTeX to stop processing the document, but preview the output before printing your final version to make sure the output is acceptable. Below are some common warnings, followed by explanations. In general, TeX is probably more of a perfectionist about wanting to get your spacing right than you are, so you shouldn't worry if it complains about overfull/underfull when it looks fine to you.

[11] Our phone number is 253-7788, or stop by our office in W20-557—just outside the Student Center cluster.
[12] On-line, type `olc` or call 253-4435.

```
Overfull \hbox (24.30751pt too wide) in paragraph at lines 6--9
```

This occurs whenever a line is too wide to fit between the margins. LaTeX will produce a line that is too wide when it cannot find an acceptable place to insert a line break. Use `xdvi` to look at the output, then change the file if necessary.

```
Underfull \hbox (badness 10000) in paragraph at lines 6--18
```

This error is similar to `Overfull \hbox`. It occurs when LaTeX is forced to put lots of space between words to justify the text. This usually happens when a paragraph is not properly ended before certain kinds of environments (like forcing a carriage return with `\\`). Inserting a blank line or a `\par` command after the offending paragraph should fix it.

```
LaTeX Warning: Label(s) may have changed. Rerun to get cross-references right.
```

This occurs when page numbers change from one running of LaTeX to another. Simply re-run LaTeX to resolve the problem. Occasionally, LaTeX will have to be run twice in a row to clear up the problem.

6 Other useful things to know about

LaTeX is very powerful. There are commands to do almost anything. This guide is an attempt to explain many of the commonly used commands. Below you will find information that should help you produce papers with LaTeX.

6.1 Margins

LaTeX has very large default margins—almost two inches wide on each side. Most people don't like them.[13] Fortunately, LaTeX margins are easy to change. An easy way to use more of the page is to use the `fullpage` option to `documentclass` (see Section 3.2).

While most text formatting systems use left and right margins, LaTeX keeps track of the left margin and the text width. In fact, since LaTeX lets you specify different margins for even and odd numbered pages, it keeps track of two left-hand margins.[14] So, to reduce the left hand margin by half an inch you would use the commands:

```
\addtolength{\oddsidemargin}{-.5in}
\addtolength{\evensidemargin}{-.5in}
```

Then to reduce the "right-hand margin" by the same amount, you would increase text width:

[13] Except for people writing books and theses, who need them.

[14] While this is very useful for double-sided documents, it can make changing the margins a little confusing.

```
\addtolength{\textwidth}{1in}
```

If you've included the `simplemargins` package, you can use a set of commands to simplify setting margins. These commands are perhaps more intuitive:

```
\setleftmargin{1in}
\setrightmargin{1in}
\settopmargin{1in}
\setbottommargin{1in}
```

If you merely want to set all margins to a uniform size, use:

```
\setallmargins{1in}
```

Again, however, you must have included the `simplemargins` packages and you must thus also set the `TEXINPUTS` variable mentioned in Section 4.

6.2 Type Styles

There are several different type styles available in LaTeX: **bold** (textbf), sans serif (textsf), *slanted* (textsl), SMALL CAPS (textsc), `typewriter` (texttt), *italic* (textit), and the default, roman (textrm). To use one of these type styles simply, use the appropriate command as shown in the following example.

The following lines:	Are produced by:
This is bold type.	`\textbf{This is bold type.}`
This is sans serif type.	`\textsf{This is sans serif type.}`
This is slanted type.	`\textsl{This is slanted type.}`
THIS IS SMALL CAPS TYPE.	`\textsc{This is Small Caps type.}`
`This is typewriter type.`	`\texttt{This is typewriter type.}`
This is italic type.	`\textit{This is italic type.}`
This is normal roman type.	`\textrm{This is normal roman type.}`

You will notice that there is no underline style. This is intentional, since underlining is not normally used in a typeset document. Titles (and other text normally underlined in hand-written documents) are properly emphasized with the `\emph` command. If you really want to get underlining, type `\underline{`*text to be underlined*`}`, but note that underlining is "fragile" (which makes it difficult to use in section titles and a few other places—see the LaTeX manual for details).

6.3 Type Sizes

The type size can be controlled as well. If you want to change the size as well as the style, you must set the size first, and then the style; to get a large bold type style, type `{\large\bf` *text*`}`.

The following sizes:	Are produced by:
This is tiny type.	`{\tiny This is tiny type.}`
This is scriptsize type.	`{\scriptsize This is scriptsize type.}`
This is footnotesize type.	`{\footnotesize This is footnotesize type.}`
This is small type.	`{\small This is small type.}`
This is normalsize type.	`{\normalsize This is normalsize type.}`
This is large type.	`{\large This is large type.}`
This is Large type.	`{\Large This is Large type.}`
This is LARGE type.	`{\LARGE This is LARGE type.}`
This is huge type.	`{\huge This is huge type.}`
This is Huge type.	`{\Huge This is Huge type.}`

6.4 Footnotes

Footnotes are very easy in LaTeX. You simply place the footnote text where you want the reference to appear and LaTeX takes care of everything else. This sentence and footnote[15] were created by typing the following:

```
This sentence and footnote\footnote{See? It's a footnote.} were created
by typing the following.
```

6.5 Tables

LaTeX has a simple method for generating all sorts of tables. You give it one command to tell it how to set up the table and then give it the data. You specify how many columns the table is going to have, what to do to the data within each column as well as how to separate each column. When you enter the data you give one row at a time. The character "`&`" comes after each data field and "`\\`" marks the end of a row. Here are several examples of what is possible and how to create them.

a	b	c
aa	bb	cccccc
aaa	bbb	ccc

This is a table with three columns. In the first column the data are centered, in the second they are left justified and in the third they are right justified. This was done by entering `{clr}`. Changing

[15] See? It's a footnote.

what is in these curly braces changes the number of columns and how each one is formatted. The previous example was created with the following text. Notice that the line breaks in the typed text make no difference to LaTeX. Rows are separated by the \\ character and the columns within the rows are separated by the & character.

```
\begin{tabular}{clr}
a& b& c\\ aa& bb& cccccc\\
aaa&
bbb& ccc\\
\end{tabular}
```

You can also have boxes around the table and lines separating the columns if you like. LaTeX will put vertical lines wherever you put a | in the column specifications. LaTeX has two commands for creating horizontal lines in tables. `\hline` creates a horizontal line across the whole table. `\cline{m-n}` creates a horizontal line from the beginning of column `m` to the end of column `n`.

a	b	c
aa	bb	cc
aaa	bbb	ccc

This was created using the following commands:

```
\begin{center}
\begin{tabular}{||c|l||r||}\hline
a& b& c\\ \cline{2-3} aa& bb& cc\\ \hline
aaa&
bbb& ccc\\ \hline\hline
\end{tabular}
\end{center}
```

6.6 Lists

There are four different types of lists in LaTeX. The following is an example containing all four.

Enumerate will give you a numbered list.

1. You can nest lists in LaTeX
   (a) You can even sub-nest lists
   (b) and it changes how they are counted
2. up to four levels deep

Itemize produces a list with bullets

- Some people like bullets,
- while others do not.

Description is what you use when you want to describe a list of items. This list of `list` environments is an example of a `description`. Notice that it will make the items you are describing appear in bold faced type.

List If you do not like any of these possibilities you can modify the `list` environment to create your own. See the LaTeX manual for more information.

The following text was used to create the previous example. We have broken it up to add comments and explanations of some of the more obscure commands.

```
% The percent sign is the comment character in LaTeX.  LaTeX ignores
% everything on a line following a % character.
\begin{description}

% Notice the brackets and braces; {} are very different from [] in LaTeX

\item [Enumerate]  will give you a numbered list.

\begin{enumerate}
\item You can nest lists in \LaTeX{}
\begin{enumerate}
\item You can even sub-nest lists
\item and it changes how they are counted
\end{enumerate}
\item up to four levels deep
\end{enumerate}

\item [Itemize] produces a list with bullets

\begin{itemize}
\item Some people like bullets,
\item while others do not.
\end{itemize}

\item [Description] is what you use when you want to describe a
list of items.  This list of \verb+list+ environments is an example of
a \verb+description+.

\item [List] If you do not like any of these possibilities you can
modify the \verb+list+ environment to create your own.  See the
\LaTeX{}  manual for more information.
\end{description}
```

7 Documents in a Foreign Language

LaTeX is very useful for writing documents in a foreign language. LaTeX can theoretically support any language. However, in reality, there is hyphenation support for only a few dozen. The standard version of LaTeX supports the English language, and it can hyphenate most English words correctly. SIPB has a version of LaTeX which can hyphenate a document written in German, and we are working on getting French and Spanish versions. The English version can be used to produce foreign language documents, but be warned that hyphenation will be according to rules of the English language.

Foreign Symbols Many Western languages use symbols and accents that do not appear in normal English. LaTeX provides a convenient method of generating accents and other symbols. Figure 1 shows accents that can be placed on any existing character—simply replace the `o` in the curly braces with the letter you wish to accent.

ó	`\'{o}`	ò	`\`{o}`	ô	`\^{o}`	ö	`\"{o}`	õ	`\~{o}`
ō	`\={o}`	ȯ	`\.{o}`	ŏ	`\u{o}`	ǒ	`\v{o}`	ő	`\H{o}`
o͡o	`\t{oo}`	o̧	`\c{o}`	ọ	`\d{o}`	o̲	`\b{o}`		

Figure 1: Accents

Some other foreign characters are encoded directly, when they cannot be built as combinations. Figure 2 shows foreign-language symbols which are available, and Figure 3 shows some special non-English symbols. Note that these characters cannot immediately be followed by other characters, so you must follow them with a space or with empty curly braces, as in:

Łukasiewicz invented prefix (Polish) notation.

which is produced by

```
\L{}ukasiewicz invented prefix (Polish) notation.
```

œ	`\oe`	Œ	`\OE`	æ	`\ae`	Æ	`\AE`	å	`\aa`
Å	`\AA`	ø	`\o`	Ø	`\O`	ł	`\l`	Ł	`\L`
ß	`\ss`	¿	`?‘`	¡	`!‘`				

Figure 2: Foreign language symbols

†	`\dag`	‡	`\ddag`	§	`\S`	¶	`\P`	©	`\copyright`
£	`\pounds`								

Figure 3: Special Language Symbols

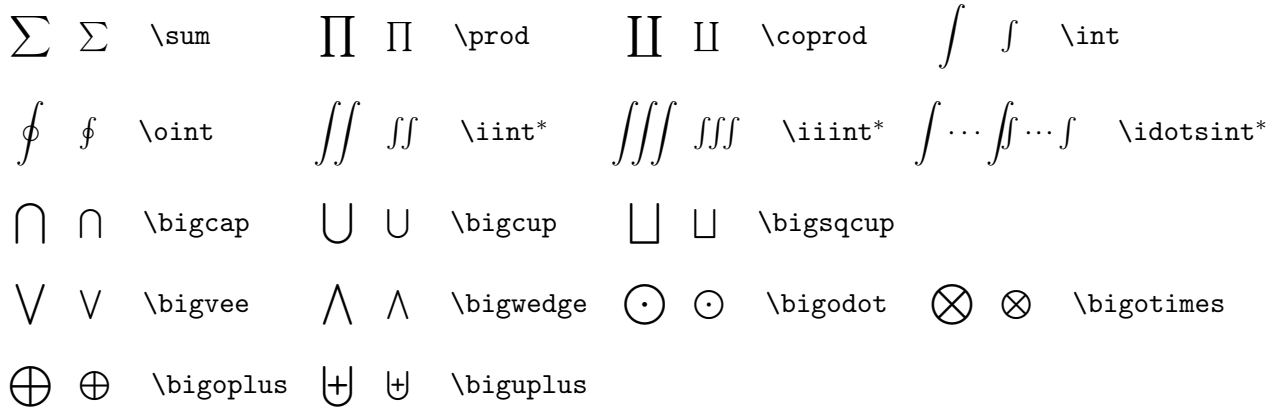


Figure 4: Mathematical Symbols (∗ Requires `amsmath` package)

8 Mathematical Text and other Special Symbols

Equations can be inserted into your text in several ways. Placing `$...$` or `\(...\)` or `\begin{math}...\end{math}` around the mathematical expression or special symbol allows you to place it in a line of normal text. This will use the smaller symbols (see Figure 4), and adjust the size of the formula as necessary. It will also adjust other things. For example, to place $\sum_{i=1}^{n}$ in a line of text, LaTeX puts the "$i = 1$" to the right of the $\sum$ instead of underneath. Placing `\[...\]` or `\begin{displaymath}...\end{displaymath}` around the expression will center it and set it off from the rest of the text, using the larger symbols.

To create a simple equation, for example $F = ma$, you need only type `$ F=ma $`. For more advanced formulæ, like $E = mc^2$, you need to tell LaTeX to superscript the 2. The `^` character instructs LaTeX to superscript, and the `_` character instructs LaTeX to subscript.

For example, `$ E=mc^2 $` is used to produce $E = mc^2$. Likewise `$ x_{k+1} $` is used to produce x_{k+1}.

If you want to put more than one character in the superscript or subscript, you need to group them, using `{` and `}` as in the examples above. To print something with both superscripts and subscripts, like ${}_{238}U^{+4}$, just type `$ {}_{238}U^{+4} $`.[16] Anything can go into the brackets, and LaTeX will figure out what to do with it: $x_{1997}^{35+9^{y_z}+4_z}$ requires the hideous expression `$x^{35+9^{y_{z}}+4_{z}}_{1997}$`, but if you look carefully it should be clear what is happening. To get the prime ($'$) symbol in math mode, type an apostrophe (`'`). Try some complex expressions of your own, and see how easy they are to create. There are many symbols not found on the keyboard to help you in formatting just about any kind of formula. See Figures 5, 6, 7 and 8 for symbols not found on the keyboard, like $\pm$, $\geq$, and $\wp$.

[16] Notice that although nothing in particular was subscripted, LaTeX was told to subscript and it did the "right thing".

$\pm$	\pm	$\mp$	\mp	$\times$	\times	$\div$	\div
$*$	\ast	$\star$	\star	$\circ$	\circ	$\bullet$	\bullet
$\cdot$	\cdot	$\cap$	\cap	$\cup$	\cup	$\uplus$	\uplus
$\sqcap$	\sqcap	$\sqcup$	\sqcup	$\vee$	\vee	$\wedge$	\wedge
$\setminus$	\setminus	$\wr$	\wr	$\diamond$	\diamond	$\amalg$	\amalg
$\oplus$	\oplus	$\ominus$	\ominus	$\otimes$	\otimes	$\oslash$	\oslash
$\odot$	\odot	$\bigcirc$	\bigcirc	$\dagger$	\dagger	$\ddagger$	\ddagger
$\bigtriangleup$	\bigtriangleup	$\bigtriangledown$	\bigtriangledown	$\triangleleft$	\triangleleft	$\triangleright$	\triangleright

Figure 5: Binary Operations

$\leq$	\leq	$\prec$	\prec	$\preceq$	\preceq	$\ll$	\ll
$\subset$	\subset	$\subseteq$	\subseteq	$\in$	\in	$\vdash$	\vdash
$\geq$	\geq	$\succ$	\succ	$\succeq$	\succeq	$\gg$	\gg
$\supset$	\supset	$\supseteq$	\supseteq	$\sqsupset$	\sqsupset	$\sqsupseteq$	\sqsupseteq
$\ni$	\ni	$\dashv$	\dashv	$\equiv$	\equiv	$\sim$	\sim
$\simeq$	\simeq	$\asymp$	\asymp	$\approx$	\approx	$\cong$	\cong
$\neq$	\neq	$\doteq$	\doteq	$\propto$	\propto	$\models$	\models
$\perp$	\perp	$\mid$	\mid	$\parallel$	\parallel	$\bowtie$	\bowtie
$\smile$	\smile	$\frown$	\frown				

Figure 6: Relations

$\leftarrow$	\leftarrow	$\Leftarrow$	\Leftarrow	$\rightarrow$	\rightarrow
$\Rightarrow$	\Rightarrow	$\leftrightarrow$	\leftrightarrow	$\Leftrightarrow$	\Leftrightarrow
$\mapsto$	\mapsto	$\hookleftarrow$	\hookleftarrow	$\leftharpoonup$	\leftharpoonup
$\leftharpoondown$	\leftharpoondown	$\rightleftharpoons$	\rightleftharpoons	$\longleftarrow$	\longleftarrow
$\Longleftarrow$	\Longleftarrow	$\longrightarrow$	\longrightarrow	$\Longrightarrow$	\Longrightarrow
$\longleftrightarrow$	\longleftrightarrow	$\Longleftrightarrow$	\Longleftrightarrow	$\longmapsto$	\longmapsto
$\hookrightarrow$	\hookrightarrow	$\rightharpoonup$	\rightharpoonup	$\rightharpoondown$	\rightharpoondown
$\uparrow$	\uparrow	$\Uparrow$	\Uparrow	$\downarrow$	\downarrow
$\Downarrow$	\Downarrow	$\updownarrow$	\updownarrow	$\Updownarrow$	\Updownarrow
$\nearrow$	\nearrow	$\searrow$	\searrow	$\swarrow$	\swarrow
$\nwarrow$	\nwarrow				

Figure 7: Arrows

$\aleph$	\aleph	$\hbar$	\hbar	$\imath$	\imath	$\jmath$	\jmath
ℓ	\ell	$\wp$	\wp	$\Re$	\Re	$\Im$	\Im
∞	\infty	$\prime$	\prime	$\emptyset$	\emptyset	∇	\nabla
$\surd$	\surd	$\top$	\top	$\bot$	\bot	$\|$	|
$\angle$	\angle	$\forall$	\forall	$\exists$	\exists	$\neg$	\neg
$\flat$	\flat	$\natural$	\natural	$\sharp$	\sharp	$\backslash$	\backslash
∂	\partial	$\triangle$	\triangle	$\clubsuit$	\clubsuit	$\diamondsuit$	\diamondsuit
$\heartsuit$	\heartsuit	$\spadesuit$	\spadesuit				

Figure 8: Miscellaneous Symbols

Of course, not all expressions are polynomials. LaTeX uses two special cases, `\sqrt` and `\frac` to create square roots (radical sign surrounding the entire expression) and fractions (one expression over another with a dividing bar between them) respectively. The term $\sqrt{1-\frac{v^2}{c^2}}$, commonly used in Lorentz transformations, is expressed as `$\sqrt{1-\frac{v^2}{c^2}}$` in line. You might want it displayed as

$$\sqrt{1-\frac{v^2}{c^2}}$$

instead, which is generated with `\[\sqrt{1-\frac{v^2}{c^2}} \]`.

If the equation is more interesting, or if you want to refer to it later on, you might want to assign it a number by using the `equation` environment. The quadratic formula

$$x=\frac{-b\pm\sqrt{b^2-4ac}}{2a} \tag{1}$$

is displayed by

```
\begin{equation}
x=\frac{-b\pm\sqrt{b^2-4ac}}{2a}
\end{equation}
```

You can supply an optional argument to `\sqrt`, specifying a different root than 2. For example, the fourth root, $\sqrt[4]{1024}$, is expressed as `\sqrt[4]{1024}`.

Often in a mathematical equation you will want to use symbols instead of the standard italicized letters for variable names. You can add Greek letters by typing the command associated with the particular letter—see Figures 9 and 10.

Note that if you want to use Greek letters in normal text mode, you must place them in a math environment by surrounding them with `$...$`. You can also create function names with the $\mathcal{CALLIGRAPHY}$ font. This is treated like any type style change such as `\bf`, but like the Greek letters, you can use it only in math mode. In addition, you only have the 26 upper-case letters available (see Figure 11).

α	`\alpha`	β	`\beta`	γ	`\gamma`	δ	`\delta`	ϵ	`\epsilon`
ε	`\varepsilon`	ζ	`\zeta`	η	`\eta`	θ	`\theta`	ϑ	`\vartheta`
ι	`\iota`	κ	`\kappa`	λ	`\lambda`	μ	`\mu`	ν	`\nu`
ξ	`\xi`	o	`o`	π	`\pi`	ϖ	`\varpi`	ρ	`\rho`
ϱ	`\varrho`	σ	`\sigma`	ς	`\varsigma`	τ	`\tau`	υ	`\upsilon`
ϕ	`\phi`	φ	`\varphi`	χ	`\chi`	ψ	`\psi`	ω	`\omega`

Figure 9: Lowercase Greek

Γ	`\Gamma`	Δ	`\Delta`	Θ	`\Theta`	Λ	`\Lambda`	Ξ	`\Xi`
Π	`\Pi`	Σ	`\Sigma`	Υ	`\Upsilon`	Φ	`\Phi`	Ψ	`\Psi`
Ω	`\Omega`								

Figure 10: Uppercase Greek

$\mathcal{A}$	`\mathcal{A}`	$\mathcal{B}$	`\mathcal{B}`	$\mathcal{C}$	`\mathcal{C}`	$\mathcal{D}$	`\mathcal{D}`
$\mathcal{E}$	`\mathcal{E}`	$\mathcal{F}$	`\mathcal{F}`	$\mathcal{G}$	`\mathcal{G}`	$\mathcal{H}$	`\mathcal{H}`
$\mathcal{I}$	`\mathcal{I}`	$\mathcal{J}$	`\mathcal{J}`	$\mathcal{K}$	`\mathcal{K}`	$\mathcal{L}$	`\mathcal{L}`
$\mathcal{M}$	`\mathcal{M}`	$\mathcal{N}$	`\mathcal{N}`	$\mathcal{O}$	`\mathcal{O}`	$\mathcal{P}$	`\mathcal{P}`
$\mathcal{Q}$	`\mathcal{Q}`	$\mathcal{R}$	`\mathcal{R}`	$\mathcal{S}$	`\mathcal{S}`	$\mathcal{T}$	`\mathcal{T}`
$\mathcal{U}$	`\mathcal{U}`	$\mathcal{V}$	`\mathcal{V}`	$\mathcal{W}$	`\mathcal{W}`	$\mathcal{X}$	`\mathcal{X}`
$\mathcal{Y}$	`\mathcal{Y}`	$\mathcal{Z}$	`\mathcal{Z}`				

Figure 11: Calligraphic Symbols

Sometimes, a formula contains text that should be set in roman type. For example, in the formula "$\log xy = \log x + \log y$," the word "log" is not italicized like the variables x and y. It would be inconvenient to have to exit math mode to type the text in normal roman type, so the most common function names are defined as special commands. If the desired command isn't defined, you can use something like `$x = \textrm{foo}$` to produce $x = \textrm{foo}$ rather than $x = foo$. A formula such as $\cos^2 x + \sin^2 x = 1$ would be entered as `$\cos^2 x + \sin^2 x = 1$`. See Figure 12 for a list of all the special commands which are typeset like this.

The modulo function is slightly different from the other special commands—it has two forms, a binary and a parenthesized one. See Figure 13 for an example of their usage. There are also various types of brackets and other delimiters available in math mode, as shown in figure 14. Certain accents and other diacritical marks are available in math mode. Some of these *math mode accents* are only available in math mode, while others are modified to work better in math mode. See Figure 15 for examples.

```
\arccos  \cos   \csc  \exp  \ker     \limsup  \min  \sinh
\arcsin  \cosh  \deg  \gcd  \lg      \ln      \Pr   \sup
\arctan  \cot   \det  \hom  \lim     \log     \sec  \tan
\arg     \coth  \dim  \inf  \liminf  \max     \sin  \tanh
```

Figure 12: Log-like symbols

$a \bmod b$	`a \bmod b`
$\pmod{a+b}$	`\pmod{a+b}`

Figure 13: Modulus functions

```
(  (       )  )          [  [       ]  ]
{  \{      }  \}         ⌊  \lfloor ⌋  \rfloor
⌈  \lceil  ⌉  \rceil     ⟨  \langle ⟩  \rangle
/  /       \  \backslash |  |       ‖  \|
```

Figure 14: Delimiters

```
â  \hat{a}    ă  \check{a}  ǎ  \breve{a}  á  \acute{a}  à  \grave{a}
ã  \tilde{a}  ā  \bar{a}    a⃗  \vec{a}    ȧ  \dot{a}    ä  \ddot{a}
```

Figure 15: Math-mode accents

The symbols in Figure 16 were provided by LaTeX 2.09, but not by LaTeX 2_ε. You can access them by using the `latexsym` package.

℧	\mho	⋈	\Join	□	\Box	◇	\Diamond	⇝	\leadsto
⊏	\sqsubset	⊐	\sqsupset	◁	\lhd	⊴	\unlhd	▷	\rhd
⊵	\unrhd								

Figure 16: Symbols available in the `latexsym` package

9 Pretty Pictures

9.1 Including PostScript and PDF Files

To import picture files into LaTeX, you need to generate files in the Encapsulated PostScript format. Many programs, *e.g.* Matlab will create EPS files for you. Otherwise, the utility `ps2epsi` will convert any PS file into an EPS one for you (just type `ps2ps filename.ps filename.eps`).

Having done so, you have many packages at your disposal to import the file into your LaTeX document. We recommend the `graphicx` package. An example of how it is used is below:

```
\use{graphicx}
... ... ...
\begin{figure}
\label{ndas}
\includegraphics[scale=0.5]{ndas}
\caption{A caption goes here.}
\end{figure}
```

The `scale` directive is an example of one of the many things the `includegraphics` directive lets you do. The `trim` argument lets you make a picture of only a cropped portion of the imported file, and the `angle` and `origin` arguments let that cropped portion be rotated.

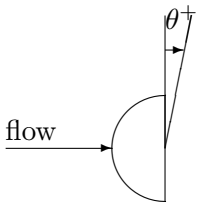
When `latex` looks at this section, it will go and find the file `ndas.eps` and put it in the image. For documents created with `pdflatex`, you can include other PDF files as figures. Conveniently enough, this same piece of LaTeX code will prompt `pdflatex` to find the file `ndas.pdf` for making the image. The utility `epstopdf` will come in handy. Also, Matlab figures, can be exported into PDF.

9.2 LaTeX `picture` Environment

You can create pictures within LaTeX using a limited set of picture symbols. These include `vector`, `line`, `oval`, and others. They are fairly difficult to use, and you have to set them up by hand. Fortunatly, the `xfig` drawing utility (`add graphics`) and the `dia` diagram program (`add dia`) let you create such figures with your mouse and generate the LaTeX code that will create them. Just use the export button and copy the resulting file into your LaTeX document. For more information you should refer to the LaTeX manual, and also SIPB's *Inessential Graphics on Athena*.

This picture was made with the following LaTeX code.

```
\begin{picture}(80,80)(0,0)
\put(60,77){\vector(1,0){7}}
\put(60,85){$\theta^+$}
\put(60,40){\line(1,5){10}}
\put(60,40){\line(0,1){50}}
\put(60,40){\oval(40,40)[l]}
\put(60,20){\line(0,1){40}}
\put(0,40){\vector(1,0){40}}
\put(0,43){flow}
\end{picture}
```



Another way to get pictures in your LaTeX document is to use plain TeX commands. See *The TeXbook* and/or ask for more information.

10 Citations and References

Your document may require references from one section to another, references to figures, tables, or references to other documents in a bibliography. Once you begin to edit and re-edit your document, LaTeX can keep track of those references better than you can. For internal references, you start by using the macro `\label{foo22}` anywhere you will later want to put a reference. In the source code for this guide, this section is marked `\label{sec:cite}`. The label must be unique. The conventional label for a section is the mark 'sec', then the semicolon, then a relevant keyword. Once the section is labeled, you can put "Section `\ref{sec:cite}`" anywhere and it will always refer to Section 10 no matter how things get rearranged.

External references call for using BibTeX, a related program for arranging a proper bibliography, and one which involves a slight learning curve. To use it, open a file called something.bib in `emacs`. The editor will oblige by going into `bibtex-mode`, and offer a menu of the common types of bibliographic entries. Choose one of them and a blank version of a BibTeX entry will pop up. After you've filled them out, it will look something like this:

```
@Article{giggle,
  author =   {Alan Sokal},
  title =   {Transgressing the Boundaries, towards a
transformative hermeneutics of quantum gravity},
  journal =   {Lingua Franca},
  year =   {1996},
  OPTkey =   {},
  OPTvolume =   {},
  OPTnumber =   {},
  OPTpages =   {},
  OPTmonth =   {},
  OPTnote =   {},
```

```
  OPTannote =   {}
}
```

The first four entries are the details that every standard for scholarly publishing requires you to fill. The others are labelled with "OPT" since they are optional. If you fill them in, delete the characters OPT. There are some subtleties and conveniences here. One one hand, BibTeX will take care of proper capitalization for any of these entries. (Acronyms need to go in curly braces to keep their capitalization as-is.) On the other hand, the author entry must all the others listed in the form "Ringo Starr and Paul McCartney and John Lennon and George Harrison" with no commas. Complicated surnames are preserved by grouping in curly braces.

Building a BibTeX file is a relatively easy task compared to anything else you do when writing a paper. You can do it even when you are exhausted. So keep at it. Now, notice the word "giggle" at the top of the example entry? It serves the same purpose as the contents of a label, and thus must be kept unique. At any point in the document where you want a reference to a citation of Sokal's article, you put `\cite{giggle}`. Let's say you have a `doc.tex` document pointing to a `cites.bib` file. Then you put the commands

```
\bibliography{cites}
\bibliographystyle{plain}
```

at the right spot in the document and you are ready to generate it.

Now comes the time to generate the bibliography. With the incantation `bibtex doc`, the `bibtex` utility uses the cites.bib file to generate a doc.blb file, in which your entries (only the ones that are cited) get translated to a properly organized and TeX formatted file that is then included by `latex`. The proper incantation to get a full bibliography assembled is `latex doc; bibtex doc; bibtex doc; latex doc; latex doc`. This makes sure all of the auxiliary files are properly written. After that, each time you add or remove a citation from your document, `bibtex doc; latex doc;` will get the final document properly updated.

11 Setting Page Layout Parameters

LaTeX uses many parameters to layout a page. These parameters set the distance between the main text and margin notes, headers, and footers, among other things. To change the value of one of these parameters, you can use the `\addtolength` commands described in subsection 6.1. Figure 17 shows the layout for a printed page reduced to 65% of full size.

12 Converting from LaTeX 2.09 to LaTeX 2$_\varepsilon$

Converting documents from LaTeX 2.09 format to LaTeX 2$_\varepsilon$ format is fairly simple. The primary differences are that the `\documentstyle` command has been replaced with the `\documentclass` command and that outside packages are included with the `usepackage` command. Another change

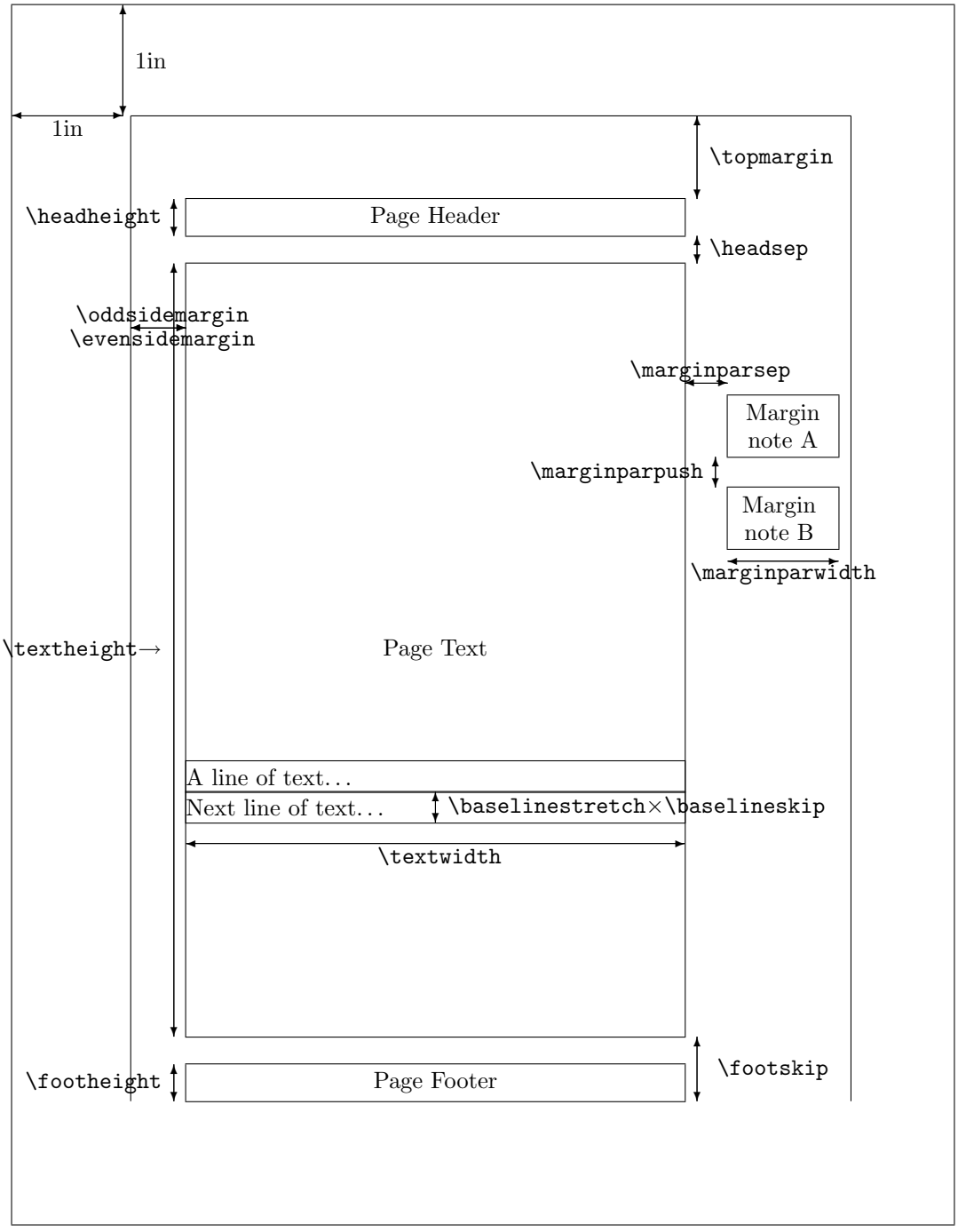


Figure 17: Page layout parameters.

is that the commands such as `{\tt}` have been superseded by the commands of the form `\texttt{}`. LaTeX 2$_\varepsilon$ also added the `\textsuperscript` command for use in normal text as an alternative to using math mode to generate superscripted text. See Section 13 for several additional sources of information which are useful in converting documents.

13 Where to Get More Information

For quick lookups of this LaTeX function or that, there is Athena's `texdoc` command.

The *LaTeX User's Guide & Reference Manual* is a very useful book which goes into much more detail than we have attempted here, for those who wish to delve deeper into the intricacies of LaTeX. It is highly recommended, and not very expensive, at least as far as reference books go. Since Athena officially supports LaTeX, you can ask questions about it using `olc`, and look through the OLC Stock Answers for the answers to other commonly asked questions. You may also wish to look at the online version of the LaTeX user's guide, which is particularly useful when converting documents to LaTeX 2$_\varepsilon$ format. It is found at
`/usr/athena/share/texmf/doc/latex/base/usrguide.dvi`.

Other information is found through the `texdoc` command. It does not offer an index of the files it examines, but poking around the directories under `/usr/athena/share/texmf/doc/latex/` will tell you what is available.

You can also come by the SIPB office (in W20-557, just outside the Student Center cluster), or call us on the phone (617-253-7788), particularly if you have a more obscure or complicated question. As a last resort, *The TeXbook* describes the TeX typesetting system, which forms the basis for LaTeX. It is significantly harder to understand than the LaTeX manual.

CTAN, the Comprehensive TeX Archive Network, centered at `http://www.ctan.org`, is another good resource for LaTeX information, and for such things as the LaTeX templates required by various publishers and academic organizations.

14 Acknowledgments

The original version of this document was written by Mark Eichin to give people a handy reference to the many symbols that LaTeX supports, especially the math mode symbols. Most of his text and all of his tables were included in the guide that you are holding now.

The rest of the first edition of the guide was written and edited by Nancy Gilman, David Jedlinsky and Mary Vogt, with lots of comments (flamage) from SIPB members and other people who experimented with the early versions of the guide. Robert French perfected this document to produce Revision 1.

Later, Chee Chew, added more changes (with more input/flamage from SIPB members) to produce Revision 2.

Revision 3 was updated by Mark Eichin to cover the replacement of `dvi2ps` by the superior `dvips`, to acknowledge Athena's support of LaTeX, and to include further SIPB member comments. Subsequent changes have been made by Richard Barbalace, bert Dvornik, and Chad Brown, with the cheerful support of many SIPB members and prospectives.

Revision 4 was updated by Alex Rolfe to reflect the adoption of LaTeX 2_ε as the default version on Athena.

Thanks to John Kohl and Jean Marie Diaz for getting most of the original TeX and LaTeX system running on Athena in the first place; without their efforts, this guide would be useless.

And, of course, thanks to Don Knuth, the author of TeX, and Leslie Lamport, the author of LaTeX; they have done an incredible job.

15 List of Figures