# Inessential LaTeX*

(Revision: 3.39)

**The Student Information Processing Board**

July 27, 1995

# Contents

# List of Figures

# 1 Introduction

## 1.1 What is LaTeX?

LaTeX is a complex text-processing system often recommended by the SIPB as an alternative to Frame Technology's FrameMaker™ and the Andrew Consortium's EZ. We believe both LaTeX and FrameMaker to be more powerful and more flexible than EZ.[1] Choosing between LaTeX and FrameMaker may be a tougher decision, since each has its strengths and weaknesses. Some people may prefer FrameMaker's graphical user interface to the typesetting commands of LaTeX. FrameMaker makes it easier to arrange your text on the page if you want more than simple one- or two-column layout. However, we believe that LaTeX is simpler to master, and more practical for formatting technical papers. It won't ever erase your document when you aren't looking[2] and leave you stranded. Also, LaTeX's range of document styles and formatting options is currently wider than FrameMaker's. It has a very powerful math typesetting facility, which makes including mathematical formulas in your document easy, and it is capable of generating documents in foreign languages complete with accents and non-English characters, such as the German ß character. Additionally, while FrameMaker is somewhat new to Athena and even EZ is still somewhat unstable on Athena, LaTeX use at Athena was introduced by the SIPB several years ago, and was officially adopted by Athena in the summer of 1989.

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[3] office (11-115) and some available for withdrawal from the MIT Libraries.

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 Graphic Arts, 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 2.09, which is the version currently supported by Athena and SIPB. A newer version (LaTeX $2\varepsilon$) exists, but it is somewhat different

[1] EZ is improving, and may someday be good enough for thesis work—but LaTeX has actually been used to write textbooks, and is a standard system in many parts of the typesetting industry (along with TeX.)

[2] FrameMaker will sometimes "lose" documents due to bugs. EZ *regularly* has problems with large documents. Thesis writers beware...

[3] On-Line Consultants

than the version described here. We recommend that you continue using version 2.09 until SIPB and Athena have gained some familiarity with the new version and can satisfactorily deal with any problems you may encounter using it. If you need to use it, LaTeX $2\varepsilon$ is available in the `newtex` locker under the name `latex2e`.

## 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.

```
\documentstyle[doublespace,fullpage]{article}

\title{A Sample Paper}
\author{Melissa I. Thompson}
\date{\today}

\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 `\documentstyle` 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, and the comma-separated list between square braces (`[` and `]`) are the optional styles. `article` is the general-purpose paper style. The `doublespace` optional style tells LaTeX to doublespace your paper, and the `fullpage` option says to set the margins so you have one inch of blank space on all sides. (These and other styles are discussed later in the document.)

- The `\title`, `\author`, and `\date` commands are used to specify the title, author and creation date of the document.
- 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:

```
\documentstyle{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`.[4] 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.[5] 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

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

[5] Putting `10pt` in the `\documentstyle` will cause an error.

specified, 10 point is assumed. To change to a base size of 11 point or 12 point, change the document style with an optional parameter, placed inside square braces immediately after the command name, like this:

```
\documentstyle[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,

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

  sets the document up as an 11-point, double-spaced, double-sided report. Most of these options cause corresponding LaTeX `style files` to be loaded (twoside.sty, doublespace.sty, etc.), but some of them are built in to the document type (the `11pt` option is done this way). You generally won't care about this unless you mistype one of the options or some optional package isn't available with the current configuration (described later).

- The space between the `\documentstyle` 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}`.

### 3.1.1 Optional Parameters

There are many optional parameters which may be included in the `\documentstyle` line, as in the example above. These options can be combined. Some of the common ones are:[6]

**doublespace** Causes the document to be doublespaced and defines the `\doublespace` and the `\singlespace` commands for changing document spacing.

**fullpage** This is the easiest way to set normal margin widths.[7]

**simplemargins**† Enables commands to simplify modifying the margins. This gives more flexibility than fullpage (see above) but takes a little more effort to use[7].

**twocolumn** Will print text in two vertical columns, side by side.

**times** To format the whole document in Times Roman font.

**palatino** To format the whole document in Palatino font.

**helvetica** To format the whole document in Helvetica font.

**ncs** To format the whole document in New Century Schoolbook font.

[6] Those marked with a † require a special `TEXINPUTS` environment variable discussed in Section 4.

[7] Both fullpage and simplemargins (see below) are explained in detail in Section 6.1.

**avantgarde** To format the whole document in Avant-Garde font.

**bookman** To format the whole document in Bookman font.

**quiet** Will tell LaTeX to run in "quiet" mode, suppressing warning messages.

### 3.2 Title Page

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

```
\documentstyle{report}

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

\begin{document}
\maketitle
```

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.

### 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 if you want a mathematical $\sim$ you can type `$\sim$` (which stands for *sim*ilar to). More often you will want the $\sim$ within another math expression anyhow; for more detail seesection 8 below.
- `^` can likewise mean circumflex, as in ô which is produced with `\^{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 `\documentstyle` command near the top.

**Divisions** Text can be divided into parts, chapters,[8] 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

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

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 ...
```

**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[9], 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 emphasize some text, *like this*, you should type `{\em like this}`. This limits the *scope* of the emphasize 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,

[9]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 in general, until you're sure that you won't be re-running LaTeX again in the near future.

like<br>these two,

you could use the following lines:

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

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[10] style of editing[11].

### 4.1 Running LaTeX

If you have used any special `.sty` files (the optional `documentstyle` parameters 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:

[10] What You See Is What You Get

[11] 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.

```
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 is an important recent addition. It represents a blank path entry, which LaTeX replaces with the default path entries.

You can process a LaTeX file by typing

```
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 **dev**ice **i**ndependent file), which can be converted and printed on various types of printers.

### 4.1.1 Running newer versions of LaTeX

The supported version of LaTeX on Athena is currently 2.09, but there are usually seperate installations of newer versions available. Currently, (LaTeX $2\varepsilon$) is available in the `newtex` locker. You can access it by:

```
add newtex
latex2e filename.tex
```

at the `athena%` prompt. In general, you can access the new version of LaTeX by default via the commands:

```
add -f -e newtex
latex filename.tex
```

In both cases, the "`add`" line may be placed in the `.environment` file. These versions of LaTeX are sometimes less stable than the default Athena version, and you may find that it takes longer for OLC to respond to questions about these versions, but they are generally useable.

## 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

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[12] and send it to your default printer, use the command:

```
dvips filename.dvi
```

If you have no default printer[13] 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.

It is possible to obtain the intermediate PostScript file, but generally they are huge files. If for some reason, you need that the PostScript file, you can use 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 -p`*first* `-l`*last* `-P`*printername* `filename.dvi`

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

Note that the older `dvi2ps` program no longer exists on Athena.

### 4.3.1 Printing for non-Athena Printers

Currently available in the `outland` locker[14] is a quite of tools to convert DVI files to HP LaserJet or IBM 3812 format, suitable for printing on many home printers. For more information on this suite of tools, you can currently do:

```
add outland
man dvi2xx
```

## 4.4 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

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

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

[14] The `outland` locker is maintained by the SIPB and many other contributors, and generally holds programs that are unsupported or in testing. Please read the file */mit/outland/README* for more information on the support policy.

```
\special{landscape}
```

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

### 4.5 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
```

The difference between duplex and tumble concerns 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 obviosuly 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:

```
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. For more information about processes and how to control them see *More Unix: the C-shell* available from Athena.

## 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.  See LaTeX manual for explanation.
              Type  H <return>  for immediate help.
```

The error message is printed on the next line, followed by the cryptic message

```
\@latexerr ...for immediate help.}\errmessage {#1}
```

This line should be ignored. The next line 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.

```
! 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.

```
!  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.

If you cannot figure out what is wrong, give the SIPB office a call,[15] or try the Athena Consultants.[16] The Zephyr `help` instance is also often useful; see the document "*Inessential Zephyr*" for more details on using zephyr 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.

```
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 return with `\\`). Inserting a blank line or a `\par` command after the offending paragraph should fix it.

[15] Our phone number is 253-7788, or stop by our office in W20-557—just outside the Student Center cluster.

[16] On-line, type `olc` or call 253-4435.

```
(\end occurred inside a group at level 1)
```

This occurs if you open an environment and forget to close it before the `\end{document}`.

```
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. If you are familiar with Scribe™, you will begin to see why LaTeX is easier to use.

## 6.1 Margins

LaTeX has very large default margins—almost two inches wide on each side. Most people don't like them.[17] Fortunately, LaTeX margins are easy to change. An easy way to use more of the page is to use the `fullpage` documentstyle (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.[18] 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:

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

If you've included the *simplemargins* option in your `\documentstyle` line, 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}
```

[17] Except for people writing books, who need them.

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

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

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

Again, however, you must have included the *simplemargins* command 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** (bf), sans serif (sf), *slanted* (sl), SMALL CAPS (sc), `typewriter` (tt), *italic* (it), and the default, roman (rm). 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.** | `{\bf This is bold type.}` |
| This is sans serif type. | `{\sf This is sans serif type.}` |
| *This is slanted type.* | `{\sl This is slanted type.}` |
| THIS IS SMALL CAPS TYPE. | `{\sc This is Small Caps type.}` |
| `This is typewriter type.` | `{\tt This is typewriter type.}` |
| *This is italic type.* | `{\it This is italic type.}` |
| This is normal roman type. | `{\rm 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 `\em` 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[19] were created by typing the following:

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

### 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. Each data field is separated by "`&`" and the end of a row is signalled by "`\\`". 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 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`.

[19] See? It's a footnote.

| 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.[20] 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.

[20]Some languages require unusual fonts. SIPB is experimenting with TeX for other languages, including German, Russian, and Arabic; send email to `sipb-babel` if you are interested in more details.

**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 characers, 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

## 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.

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\displaystyle\sum$ | $\sum$ | `\sum` | $\displaystyle\prod$ | $\prod$ | `\prod` | $\displaystyle\coprod$ | $\coprod$ | `\coprod` | $\displaystyle\int$ | $\int$ | `\int` |
| $\displaystyle\oint$ | $\oint$ | `\oint` | $\displaystyle\bigcap$ | $\bigcap$ | `\bigcap` | $\displaystyle\bigcup$ | $\bigcup$ | `\bigcup` | $\displaystyle\bigsqcup$ | $\bigsqcup$ | `\bigsqcup` |
| $\displaystyle\bigvee$ | $\bigvee$ | `\bigvee` | $\displaystyle\bigwedge$ | $\bigwedge$ | `\bigwedge` | $\displaystyle\bigodot$ | $\bigodot$ | `\bigodot` | $\displaystyle\bigotimes$ | $\bigotimes$ | `\bigotimes` |
| $\displaystyle\bigoplus$ | $\bigoplus$ | `\bigoplus` | $\displaystyle\biguplus$ | $\biguplus$ | `\biguplus` | | | | | | |

Figure 4: Mathematical 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} $`.[21] Anything can go into the brackets, and LaTeX will figure out what to do with it: $x^{35+9^{y_z}+4_z}_{1997}$ 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 $\pm$, $\geq$, $\leadsto$ and $\wp$.

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. To display it, you might want to use

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

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

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

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $\pm$ | `\pm` | $\mp$ | `\mp` | $\times$ | `\times` | $\div$ | `\div` | $\ast$ | `\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` |
| $\lhd$ | `\lhd` | $\rhd$ | `\rhd` | $\unlhd$ | `\unlhd` | $\unrhd$ | `\unrhd` | $\oplus$ | `\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` | $\sqsubset$ | `\sqsubset` | $\sqsubseteq$ | `\sqsubseteq` | $\in$ | `\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` | $\Join$ | `\Join` |
| $\smile$ | `\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` |
| $\leadsto$ | `\leadsto` | $\uparrow$ | `\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` | $\ell$ | `\ell` |
| $\wp$ | `\wp` | $\Re$ | `\Re` | $\Im$ | `\Im` | $\mho$ | `\mho` | $\prime$ | `\prime` |
| $\emptyset$ | `\emptyset` | $\nabla$ | `\nabla` | $\surd$ | `\surd` | $\top$ | `\top` | $\bot$ | `\bot` |
| $\Vert$ | `\Vert` | $\angle$ | `\angle` | $\forall$ | `\forall` | $\exists$ | `\exists` | $\neg$ | `\neg` |
| $\flat$ | `\flat` | $\natural$ | `\natural` | $\sharp$ | `\sharp` | $\backslash$ | `\backslash` | $\partial$ | `\partial` |
| $\infty$ | `\infty` | $\Box$ | `\Box` | $\Diamond$ | `\Diamond` | $\triangle$ | `\triangle` | $\clubsuit$ | `\clubsuit` |
| $\diamondsuit$ | `\diamondsuit` | $\heartsuit$ | `\heartsuit` | $\spadesuit$ | `\spadesuit` | | | | |

Figure 8: Miscellaneous Symbols

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.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $\alpha$ | `\alpha` | $\beta$ | `\beta` | $\gamma$ | `\gamma` | $\delta$ | `\delta` | $\epsilon$ | `\epsilon` |
| $\varepsilon$ | `\varepsilon` | $\zeta$ | `\zeta` | $\eta$ | `\eta` | $\theta$ | `\theta` | $\vartheta$ | `\vartheta` |
| $\iota$ | `\iota` | $\kappa$ | `\kappa` | $\lambda$ | `\lambda` | $\mu$ | `\mu` | $\nu$ | `\nu` |
| $\xi$ | `\xi` | $o$ | `o` | $\pi$ | `\pi` | $\varpi$ | `\varpi` | $\rho$ | `\rho` |
| $\varrho$ | `\varrho` | $\sigma$ | `\sigma` | $\varsigma$ | `\varsigma` | $\tau$ | `\tau` | $\upsilon$ | `\upsilon` |
| $\phi$ | `\phi` | $\varphi$ | `\varphi` | $\chi$ | `\chi` | $\psi$ | `\psi` | $\omega$ | `\omega` |

Figure 9: Lowercase Greek

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).

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $\Gamma$ | `\Gamma` | $\Delta$ | `\Delta` | $\Theta$ | `\Theta` | $\Lambda$ | `\Lambda` | $\Xi$ | `\Xi` |
| $\Pi$ | `\Pi` | $\Sigma$ | `\Sigma` | $\Upsilon$ | `\Upsilon` | $\Phi$ | `\Phi` | $\Psi$ | `\Psi` |
| $\Omega$ | `\Omega` | | | | | | | | |

Figure 10: Uppercase Greek

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

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $($ | `(` | $)$ | `)` | $[$ | `[` | $]$ | `]` |
| $\{$ | `\{` | $\}$ | `\}` | $\lfloor$ | `\lfloor` | $\rfloor$ | `\rfloor` |
| $\lceil$ | `\lceil` | $\rceil$ | `\rceil` | $\langle$ | `\langle` | $\rangle$ | `\rangle` |
| $/$ | `/` | $\backslash$ | `\backslash` | $\vert$ | `\|` | $\Vert$ | `\\|` |

Figure 14: Delimiters

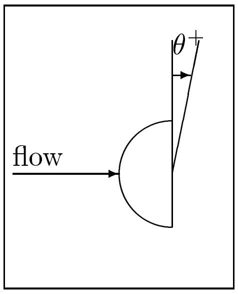
| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $\hat{a}$ | `\hat{a}` | $\check{a}$ | `\check{a}` | $\breve{a}$ | `\breve{a}` | $\acute{a}$ | `\acute{a}` | $\grave{a}$ | `\grave{a}` |
| $\tilde{a}$ | `\tilde{a}` | $\bar{a}$ | `\bar{a}` | $\vec{a}$ | `\vec{a}` | $\dot{a}$ | `\dot{a}` | $\ddot{a}$ | `\ddot{a}` |

Figure 15: Math-mode accents

# 9 Pretty Pictures

## 9.1 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. For more information you should refer to the LaTeX manual.

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}
```

## 9.2 Including PostScript Files

In general you are better off using a drawing program designed for the device you will be printing on, and then using the `\special` command (or the `\PSbox` macro below) to include a file.[22]

If you are using a program which generates Encapsulated PostScript Files, you can simply use the `epsf` documentstyle, and then say

```
\epsfbox{filename.ps}
```

at the place where you want the figure. This automatically reserves space for the PostScript picture based on the size stored in it. See the `dvips` man page for more detail.

[22] The programs *idraw* and *xfig*, in the `sipb` locker, will allow you to draw pictures and save them as PostScript files. Also, the program *xv*, in the `graphics` locker, can convert files from many popular graphics formats to PostScript format (with some loss of quality). See the SIPB document *An Inessential Guide to Athena* for more information.

To include a non-Encapsulated PostScript file (*i.e.* one written by hand or by certain drawing programs):

1. Include the following line before the `\begin{document}`. Most of the magic is needed so that LaTeX knows how big the document is, so that it can reserve space for the PostScript figure for proper placement and spacing.

```
\newcommand{\PSbox}[3]{\mbox{\special{psfile=#1}\hspace{#2}\rule{0pt}{#3}}}
```

2. If possible, generate the PostScript file such that the picture is flush against the bottom-left corner. (For example, if `idraw` is being used, group the entire picture and move it to the bottom-left corner of the page. If it is a raw PostScript file, add an appropriate translate command, if you know how.) When you do this, the left-most $\frac{1}{4}$ inch of the picture may not be printed, since although (0,0) is at the lower-left hand corner of the page, on some printers the left most $\frac{1}{4}$ inch part of the page isn't part of the image area. Don't worry about it—the missing part should be visible when $\LaTeX$ moves it back onto the page.

3. Print out the PostScript file. There are two reasons for doing this. The first is to make sure that there aren't any errors in the PostScript. The second reason is to get the dimensions of the picture and where it is located on the page. Measure the height and the width of the picture. Also, if the picture is not located in the lower-left-hand corner, measure the $(x, y)$ distance from the lower-left-hand corner of the page to lower-left-hand corner of the graphic in inches.

4. To do the actual inclusion, do step (a) if the image is in the lower left corner of the page. Otherwise do step (b).

   (a) If the PostScript file has the image in the lower-left-hand corner of the page, it's really easy. At the point where you want the picture to be included, just use the following command:

   ```
   \PSbox{/mit/username/filename.PS}{1.7in}{0.25in}
   ```

   Where the first argument is the filename of the PostScript file, the second is the width of the graphic, and the third argument is the height of the graphic. The dimensions can be specified in any format that LaTeX accepts. So, for example, a dimension such as "`4cm`" will also work.
   This "PSbox" can be included in a figure, or anything else you want!

   (b) If you were unfortunate enough as to be unable to get the graphic in the lower-left-hand corner, you will need to do a bit more work. Take the distance from the left-hand side of the page (which you measured in inches, right?) and multiply by -72. Call this number the "hoffset". Take the distance from the bottom of the page to the bottom of the graphic (also measured inches), multiply it by -72, and call it the "voffset". Then the PSbox command should be modified to be:
   `\PSbox{/mit/username/filename hoffset=`hoffset `voffset=`voffset`}{`width`}{`length`}`
   For example, to include a file which is 4 inches high and 5 inches wide, which is located 3 inches from the bottom of the page and 2 inches from the left side of the page, the PSbox would be:

```
\PSbox{/mit/username/filename.PS hoffset=-144 voffset=-216}{5in}{4in}
```

NOTE: `xdvi` cannot show included PostScript figures; instead, it leaves a blank space of the correct size. `Xdvi` will also print a warning that says '`xdvi: special ''filename'' not implemented`' every time you preview a page with a PostScript figure on it. This is nothing to worry about—it's just reminding you that it can't preview the PostScript figure—it should still print out without any problems.

NOTE NOTE: the version of `xdvi` currently installed in the `sipb` locker *can* preview PostScript figures, either using the native Display PostScriptfeatures (default), or by rendering the image through another program first (the "`-nodps`" option). This feature still has some problems, and in some circumstances X server bugs could caused you to be logged out without warning. As always, we recomend that you save your files early and often.

If you want to see where the PostScript picture will appear, you can change the `\mbox` in the definition of `\PSbox` to `\framebox`. In other words, you'd change the line

```
\newcommand{\PSbox}[3]{\mbox{\special{psfile=#1}\hspace{#2}\rule{0pt}{#3}}}
```

in your document to be:

```
\newcommand{\PSbox}[3]{\fbox{\special{psfile=#1}\hspace{#2}\rule{0pt}{#3}}}
```

Assuming that you have specified the arguments to `\PSbox` correctly, this will result in a box being drawn around the area where the PostScript image will appear. However, when you print your LaTeX document, the box will appear on the printed output as well. You will need to change the `\fbox` back to `\mbox` and run LaTeX again on your file to make the box go away when you want to print your final copy.

### 9.2.1 Other Tricks

You can also shrink and scale your PostScript figures. For example, if you wanted to include your figure shrunk by half in each direction, you would use:

```
\PSbox{/mit/username/filename.PS hscale=50 vscale=50}{1in}{1in}
```

(with the dimensions being the size of the figure once it has been scaled to half size in each direction). Note that it's possible to shrink your figure in one direction and not the other; in most cases, though, you want to scale it equally both horizontally and vertically. The scale factors are percentages, which is why the example uses 50 for one half.

## 10 Setting Page Layout Parameters

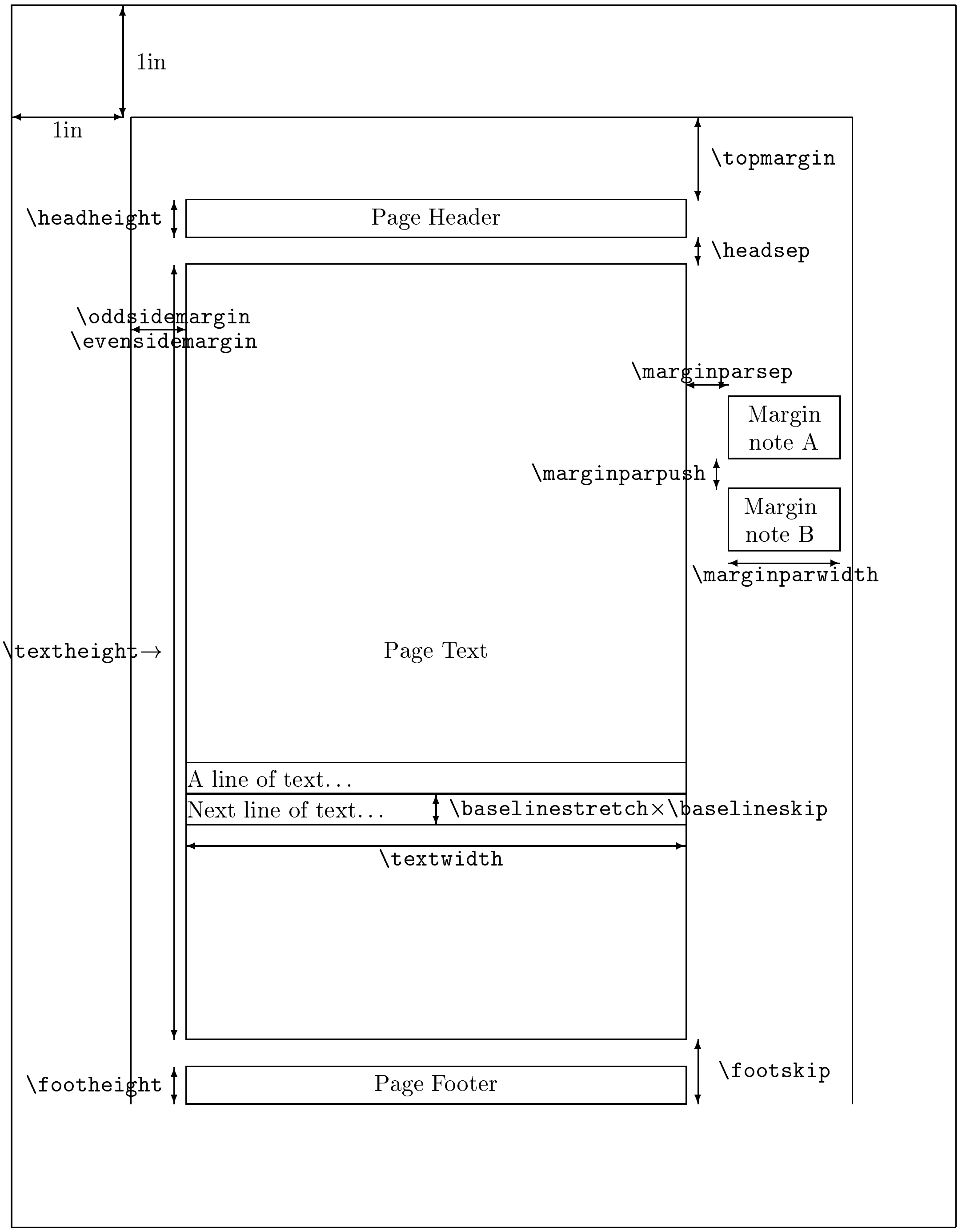



Figure 16: Page layout parameters.

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 16 shows the layout for a printed page reduced to 65% of full size.

## 11 Similarities between Scribe™ and LaTeX

Many Scribe™ and LaTeX commands are quite similar. If you are unfamiliar with Scribe™, don't bother to learn about it; this section is intended to help those unfortunate souls who already have Scribe™ experience and are trying to learn LaTeX and convert their old documents.

| **Scribe™** | LaTeX |
|---|---|
| `@make(article)` | `\documentstyle{article}` |
| `@make(report)` | `\documentstyle{report}` |
| `@style(size=11)` | `\documentstyle[11pt]{article}` |
| `@style(font` *fontname*`)` | `\documentstyle[`*fontabbrev*`]{article}` |
| `NewCenturySchoolBook` | `ncs` |
| `TimesRoman` | `times` |
| `@begin(mode)` | `\begin{mode}` |
| `@end(mode)` | `\end{mode}` |
| `@i(italics)` | `{\em italics}` |
| `@verbatim(xyz)` | `\verb+xyz+` |
| `@section(...)` | `\section{...}` |

## 12 Where to Get More Information

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 can also come by the SIPB office (in W20-557, just outside the Student Center cluster), or call us on the phone (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.

## 13 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 (which you are now reading) 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.

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.