Some Examples of Using LaTeX (First Draft)

Samuel R. Buss*
Department of Mathematics
University of California, San Diego

May 15, 1996

Abstract

We discuss some of the more useful features of LaTeX, especially ones which differ from other versions of TeX. We are not trying to give definitive or in-depth documentation, but just to show some of the more useful elementary features of LaTeX.

1 Numbering sections

The previous is an example of how section are numbered.

1.1 Numbering subsections

Now we have an example of how subsections are numbered. One can also number subsubsections.

2 Typesetting Mathematics

LaTeX supports the usual features of TeX for typesetting mathematics; in addition, it has some macros which are useful for formating mathematics.

2.1 Automatic numbering of theorems

Using commands such as `\begin{theorem}...\end{theorem}`, one can have theorems automatically numbered. For example,

*Supported in part by an NSF grant DMS-3.1415926535 and DMS-2.718281828459045 .

Theorem 1 *Let $m \geq 0$. Then*

$$\sum_{i=0}^{m} i = \frac{m(m+1)}{2}.$$

Corollary 2

$$\sum_{i=0}^{m} = \Omega(m)$$

Lemma 3

$$\sum_{i=0}^{m} 1 = m+1$$

One can also readily define environments for definitions. For example, see the definition of the `definition` environment above.

Definition *The numeral 1 denotes the least integer greater than zero.*

2.2 Numbered equations

One can use the commands `\begin{equation}...\end{equation}` to typeset a numbered equation. An example of this the equation

$$\sum_{i=0}^{m} i = \frac{m(m+1)}{2}. \tag{1}$$

Please note the use of `\label{...}` in the above equation, which allows one to refer to equation (1) by its number using `\ref{...}`.

2.3 Multiline equations

Multiline equations are always difficult to typeset, but LaTeX supplies a few tools that help somewhat. Nonetheless, good judgement is needed to make multiline equations attractive.

The basic tool for multiline equations is

`\begin{eqnarray}...\end{eqnarray}`.

This command generates equation numbers on every line of the equation: the equations numbers can be turned off selectively by using `\nonumber`, or they can all be omitted, but using `eqnarray*` in place of `eqnarray`.

An example of a multiline equation is:

$$\begin{aligned}\frac{m(m+1)}{2} &= 1+2+\cdots+m \qquad (2)\\ &= 0+1+2+\cdots+m\\ &= \sum_{i=0}^{m} i.\end{aligned}$$

Another example, with a long first line, is:

$$\begin{aligned}
\lefteqn{1+2+3+4+5+6+7+8+9+\cdots+(m-1)+m} \\
&= 0+1+2+3+4+5+6+7+8+9+10+11 \\
&\qquad +12+13+\cdots+(m-1)+m \\
&= \frac{m(m+1)}{2} \\
&= \sum_{i=0}^{m} i.
\end{aligned}$$

2.4 Tables and arrays in mathematics

Arrays or tables are often used in math mode. Two common examples are shown next:

$$I = \left(\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array} \right)$$

and

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

2.5 Tables in non-math mode

One can also build tables in non-math mode, using the `tabular` environment. Here is a small example:

Title of table		
Animal	Feet	Eyes
Sheep	Four	Two
Fish	Zero	Two
Fly	Six	Many

Note that the above example enclosed the `tabular` environment in a `center` environment to make the table be centered.

By the way, entries in tables can be almost anything, including mathematical formulas, or paragraphs of text, or even tables.

2.6 Tabbing

Finally, the `tabbing` environment is a another simply and useful way to align your data; it's not as flexible as tables, but it is a lot easier. For example,

New Years Day: January 1
Cinco de Mayo: May 5

Halloween:	October 30
Christmas:	December 25

3 Lists and Itemizations

LaTeX has several similar commands for enumerating or itemizing lists. These are controlled with the `enumerate` environment, the `itemize` environment, and the `description` environment. The first two automatically generate numbers or bullets, but these can be overridden with the `\item[...]` command.

Example of the `enumerate` environment:

1. This is the first item in the list.

   Of course, multiple line and multiple paragraphs are permitted in the body of a single item in the enumeration.

2. The second item in the list.

3. The last item in the list. The numeric labels on the items in the `enumerate` environment can be overridden with `\item[...]`.

Example of the `itemize` environment:

- This is the first item in the list.

  Of course, multiple line and multiple paragraphs are permitted in the body of a single item in the itemization.

- The second item in the list.

- The last item in the list. The bullets on the items in the `itemize` environment can be overridden with `\item[...]`.

Example of the `description` environment:

First This is the first item in the list.

Of course, multiple line and multiple paragraphs are permitted in the body of a single item in the description list.

SECOND The second item in the list.

Last The last item in the list.

4 Using Bibtex

Bibtex provides a convenient way of including citations to papers in a bibliography file. It allows you to create a single bibliography file that can be shared among many LaTeX papers. In addition, many sources of bibliographic information on the Web now provide citations in Bibtex format, and these can be readily included in your own Bibtex files.

For example, one can reference books [3], or one can reference Ph.D. theses [4], or one can reference journal articles [1, 2], or one can reference articles in conferences [5]. And many other types of references are also possible.

To prepare bibliographies, one must do the following:

(1) Create a `.bib` file, such as the file `samplebibl.bib` used for this document,

(2) Add the lines

```
\bibliography{samplebibl}
\bibliographystyle{siam}
```

near the end of the LaTeX document, where you want the bibliography to appear,

(3) Write citations in the form `\cite{AJ}` which cite the article with the name "AJ".

One has to run latex and bibtex alternately to create all the appropriate auxiliary files that generate the bibliography citations. For this file, I had to run:

```
% latex samplelatex
% bibtex samplelatex
% latex samplelatex
% bibtex samplelatex
% latex samplelatex
```

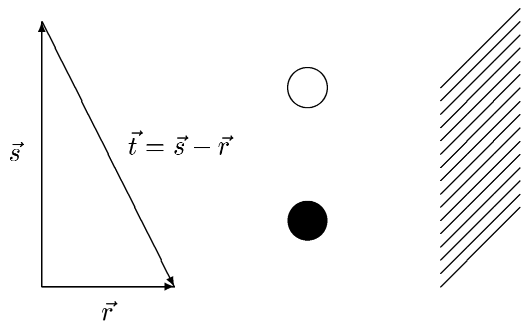
5 Graphics in LaTeX

There are two ways to create pictures in LaTeX. The first is its built-in picture environment, which allows a limited amount of functionality forming pictures. The second is to include postscript pictures in your document.

5.1 The picture environment

The `picture` environment allows pictures to be formed from lines, vectors, circles, disks, psuedo-ovals, boxes, and any LaTeX object. Unfortunately, the picture environments has some serious limitations; lines can have slopes only from a limited set of slope values, lines must have a certain minimum length, circles can be no larger than 40 points (about 1/2 inch) in diameter. However, if a picture environment suffices from your application, it is often the best choice.

Here is an example, showing some of the features:



5.2 The pstricks environment

The pstricks macro package allows much sophisticated graphics commands than are allowed by the built-in `picture` environment. Documentation for this is not included in this document. (You can get it from the Web from the CTAN - Comprehensive TeX Achive Network - web site.)

Pstricks is the only package that I know of that allows the full power of postscript, plus allows you to include TeX formatted mathematical text in the same figure.

5.3 Including postscript files

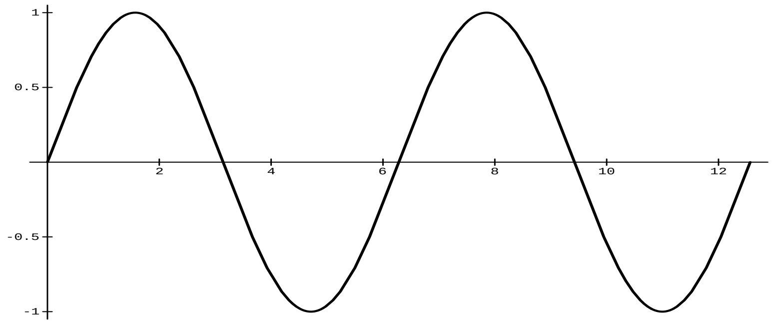
To include a postscript figure in your document, one uses the `psfig` package. First, near the beginning of the LaTeX document include the command `\input{psfig}`. Then use the `\psfig` command to include encapsulated postscript files wherever you wish. The general format of the `\psfig` command:

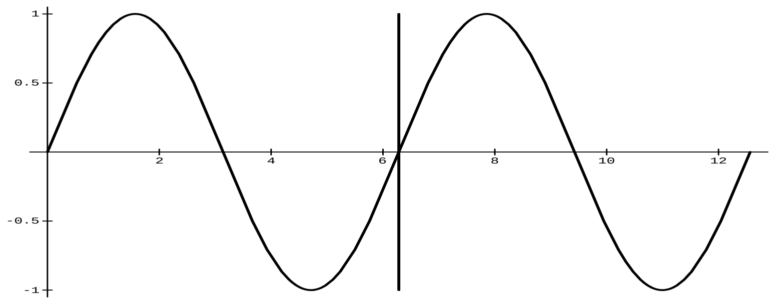
`\psfig{figure=`*filename*`,height=2.7in,width=4.5in}`

where the figure will be stretched or compressed to fit the specified dimensions. (Important: there should be no extra spaces in the `\psfig` command!)

5.3.1 Including MATHEMATICA drawings

One can include Mathematica graphics in your LaTeX file; for example, here are two graphs of $sin(x)$, the second having an extra vertical line at $x = 2\pi$.





To generate the encapsulated postscript files from Mathematics, one outputs a graphics object to a file using the `Display` command. The `Display` command does not include the postscript encapsulation commands, so one must build an encapsulated postscript file using the `psfix` command which is supplied with Mathematica. As an illustration, the commands that generated the two encapsulated postscript files which contain the above two plots are:

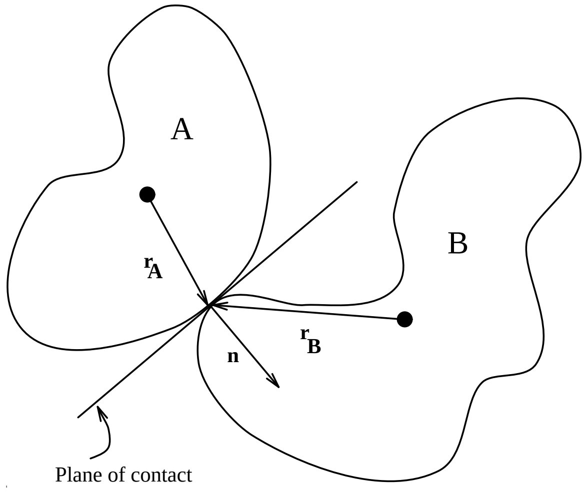
```
euclid.ucsd.edu% math
Mathematica (sun4) 1.2 (November 6, 1989) [With pre-loaded data]
by S. Wolfram, D. Grayson, R. Maeder, H. Cejtin,
   S. Omohundro, D. Ballman and J. Keiper
with I. Rivin and D. Withoff
Copyright 1988,1989 Wolfram Research Inc.
 -- X11 windows graphics initialized --

In[1]:= graphSin = Plot[ Sin [x], {x, 0, 4 Pi}]

Out[1]= -Graphics-
```

```
In[2]:= Display [ "SinePlotFile", graphSin ]

Out[2]= -Graphics-

In[3]:= myLine = Graphics[ Line[ {{2Pi, -1}, {2Pi, 1}}] ]

Out[3]= -Graphics-

In[4]:= Show [ graphSin, myLine ]

Out[4]= -Graphics-

In[5]:= Display [ "SineWithLineFile", % ]

Out[5]= -Graphics-

In[6]:= Quit
euclid.ucsd.edu% psfix SinePlotFile > SinePlotFile.eps
euclid.ucsd.edu% psfix SineWithLineFile > SineWithLineFile.eps
```

5.3.2 Using xfig to generate postscript files

sc Unix has a command `xfig` that can be used to handdraw figures in the `fig` style. The native file format for xfig file are files which the `.fig` extension. However, xfig can export files in encapsulated postscript format.

Here is a very simple example of a drawing generated with `xfig`:



To run xfig, you must use OpenWin or another X-windows environment and give the command

```
% xfig
```

If you have used CAD-style programs before, xfig is fairly self-explanatory. Some things to note especially are that the function of the mouse button is dynamically displayed in the upper right corner of the window. Choosing the correct grid resolution will simplify the drawing process considerably. Also note that the `Edit` feature allows you to modify the attributes of picture objects after they have been created. You should always save you drawing as a `.fig` file and, then export it to postscript afterwards. (Since xfig can read only `.fig` format files, you will need the `.fig` version to make subsequent changes.)

References

[1] S. Abramsky and R. Jagadeesan, *Games and full completeness for multiplicative linear logic*, Journal of Symbolic Logic, 59 (1994), pp. 543–574.

[2] W. Ackermann, *Zur Widerspruchsfreiheit der Zahlentheorie*, Mathematische Annallen, 117 (1941), pp. 162–194.

[3] J. Barwise, *Handbook of Mathematical Logic*, North-Holland, Amsterdam, 1977.

[4] J. H. Bennett, *On Spectra*, PhD thesis, Princeton University, 1962.

[5] A. Cobham, *The intrinsic computational difficulty of functions*, in Logic, Methodology and Philosophy of Science, proceedings of the second International Congress, held in Jerusalem, 1964, Y. Bar-Hillel, ed., Amsterdam, 1965, North-Holland.