A LaTeX 2ε Sampler

Jim Callahan

Revised: 6 August 2002

Introduction

New features

LaTeX 2ε is the current version of LaTeX, released in June 1994. In the decade after the release of the original LaTeXin the 1980s, many incompatible "dialects" had evolved at different sites—$\mathcal{AMS}$-LaTeX among them. One of the main goals of the developers of LaTeX 2ε was to pull these different dialects together—so that, for example, we could use all the special macros of $\mathcal{AMS}$-TeX in an ordinary LaTeX document. Another goal was to improve the handling of typefonts, and to make a wide selection of typefonts available and easy to access. Finally, the new version also adds some useful commands.

Documentation

The first two items listed below are the basic references. The next two are brief "local guides" for working on our machines. Documents covering many topics are available on paper in Burton 303 and also on the department Web site (`http://math.smith.edu/Local/localtex.html`).

- *LaTeX: A Document Preparation System*, by Leslie Lamport. Lamport developed the original LaTeX (which is now called LaTeX2.09 when the distinction needs to be made). His book is a very friendly and well-written guide. With it you can be producing complex documents within a day. The second edition (published by Addison-Wesley in 1994; ISBN 0-201-52983-1) covers the new version.
- *The LaTeX Companion*, by Michel Goosens, Frank Mittleback, and Alexander Samarin (also Addison-Wesley 1994; ISBN 0-201-54199-8). This book gives the details about LaTeX 2ε itself. It tells you how to handle fonts and it describes the features of about 100 optional packages that make the new version so flexible.
- *Running TeX and LaTeX on Mathematics UNIX Machines.*
- *Putting Mathematica Graphics into TeX Files on Mathematics UNIX Machines.*

Audience and topics

This *Sampler* is intended to illustrate some of what TeX and LaTeX users can find in *The LaTeX Companion*. It describes how a LaTeX document is put together and how the new version differs from the earlier. It describes fonts and how to use them, and it looks at a few of the many packages that are available—concentrating on the mathematical ones.

Structure of a LaTeX document

An example

LaTeX is a macro package that "sits on top of" TeX; the main purpose of the macros is to define a particular *style* for the document that is being produced with TeX commands. The information about the style is put into a **preamble** at the beginning of the document. Here is an example of a complete LaTeX 2_ε document.

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

preamble

```
\begin{document}

The differential equations
\begin{align*}
x' &= v
v' &= -k\sin{x}
\end{align*}
describe the motion of a frictionless
pendulum that makes an angle of $x(t)$
radians with the vertical at time $t$.

\end{document}
```

body

Here is the output of that document.

> The differential equations
>
> $$x' = v$$
> $$v' = -\sin x$$
>
> describe the motion of a frictionless pendulum that makes an angle of $x(t)$ radians with the vertical at time t.

The document class

The first line should be familiar to LaTeX users (except that the declaration `\documentclass` was written `\documentstyle` in LaTeX2.09). This document uses the article class. The other common classes are report, book, letter, and slides (for overhead transparencies). Some journals and publishers define their own document classes and distribute files that define those classes. For example, we have the AMS classes amsabs (abstract), amsart (article), amsppt (preprint), amsproc (proceedings), and amsbook.

Class options

The `\documentclass` declaration has an optional argument (set in square brackets) that modifies the class. The main class options are 11pt, 12pt (which switch the text size from the 10pt default) twocolumn, and twoside.

Packages

The `\usepackage` declaration allows you to introduce special features into your document in a standard way. The rest of this *Sampler* describes some of those packages.

Input files

Similar to package files are input files. They contain macros and typically have the extension `.tex`. You can even put your own macros in an input file. To access the macros in `mymacs.tex`, for example, put the command

```
\input{mymacs}
```

in the preamble of your document. All our packages and input files are found in subdirectories of the standard RedHat Linux distribution `/usr/share/texmf/` or our local supplement `/usr/local/texmf.mathlin/`.

Fonts

Choosing fonts

The default TEX font (i.e., typeface) is Computer Modern. But we also have access to a number of other fonts. Here are brief samples of what's available; there are longer samples in the Appendix. The use of the short name is explained below.

short name	*package name*	*sample*
cmr		This is Computer Modern
ccr	ccfonts	This is Concrete
pad	garamond	This is Garamond
put	utopia	This is Utopia
ppl	palatino	This is Palatino
pbk	bookman	This is Bookman
pnc	newcent	This is New Century Schoolbook
ptm	times	This is Times-Roman
bch	charter	This is Charter
panr	pandora	This is Pandora
phv	helvet	This is Helvetica
pag	avant	This is AvantGarde
fmv	malvern	This is Malvern
fmvx	malvernb	This is Malvern Bold
fmvx2	malvernbx	This is Malvern Extra Bold
pzc	chancery	*This is Zapf Chancery*
pcr	courier	This is Courier

Global changes

To set an entire document in a particular font, put a `\usepackage` command in the preamble. For example, to make New Century Schoolbook the

default font, your preamble should include this statement:

```
\usepackage{newcent}
```

Local changes

If you want to change only a portion of text to New Century Schoolbook, type the following in the body of your text:

```
{\fontfamily{pnc}\selectfont your text here}
```

In other words, use the *short name* of the font as the argument to the `\fontfamily` command, but use the *package name* as the argument to the `\usepackage` command. For more information on fonts, read Lamport, §3.1 and §3.3.8 and the *Companion*, §7.6.

concmath

Even when you set an entire document in a new font, the mathematics is typically still set in the (default) Computer Modern fonts. However, Concrete has been extended to a larger package, called concmath, that includes its own math fonts to match the Concrete text fonts. It even has its own versions of the AMS math symbols (see the following section) that can be accessed as options. For example, to use concmath with the full AMS symbols package, put

```
\usepackage[amssymb]{concmath}
```

in the preamble. For further details see the online or paper copy of *The concmath package*.

Mathematical packages

amsfonts, latexsym, and amssymb

The current LaTeX does not automatically load all the mathematical symbols that the earlier version did. You may need to load one of the packages latexsym or amsfonts (with a `\usepackage` declaration). The complete collection of symbols is in the amssymb package.

Symbols in the amssymb package

name	*code*	*example*	*output*
blackboard bold alphabet	`\mathbb`	`$\mathbb{R C Z}$`	$\mathbb{RCZ}$
Euler fractur alphabet	`\mathfrak`	`$\mathfrak{gl}_n(R)$`	$\mathfrak{gl}_n(R)$
bold symbols	`\boldsymbol`	`$\boldsymbol{\alpha_{\infty}}$`	$\boldsymbol{\alpha_{\infty}}$
"poor man's bold"	`\pmb`	`$\pmb{\oint}$`	$\pmb{\oint}$

Poor man's bold simply prints a second copy of each character, slightly offset from the first. Use it when there is no other bold version of the symbol you want.

amsmath

Probably one of the most useful packages is amsmath. This supercedes the package amstex which is discussed at length in the *Companion*. The package amsmath is itself a collection of "subpackages" which are loaded automatically with it. $\mathcal{AMS}$-LaTeX is equivalent to LaTeX with the amsmath package loaded. The following paragraphs show how the amsmath package allow you to handle displayed equations and matrices. (A "frozen" version of amstex still exists, so documents prepared with it can still be compiled.)

`align` versus `eqnarray`

The original LaTeX uses the `eqnarray` environment to construct a stack of equations that are vertically aligned on a symbol—usually an equals sign. But `eqnarray` puts too much space around that symbol, as the following illustrates:

Here is a single equation:

$$S' = -aSI$$

and here is a stack of equations aligned by `eqnarray*`:

$$\begin{aligned} S' &= -aSI \\ I' &= aSI - bI \\ R' &= bI \end{aligned}$$

```
Here is a single equation:
$$
S' = -aSI
$$
and here is a stack of equations
aligned by |eqnarray*|:
\begin{eqnarray*}
S' &=& -aSI \\
I' &=&  aSI - bI \\
R' &=&  bI
\end{eqnarray*}
```

The `align` environment fixes this. (We have used the `*` versions of both `eqnarray` and `align` here in order to suppress the automatic generation of equation numbers.)

Here is a single equation:

$$S' = -aSI$$

and here is a stack of equations aligned by `align*`:

$$\begin{aligned} S' &= -aSI \\ I' &= aSI - bI \\ R' &= bI \end{aligned}$$

```
Here is a single equation:
$$
S' = -aSI
$$
and here is a stack of equations
aligned by |align*|:
\begin{align*}
S' &= -aSI \\
I' &=  aSI - bI \\
R' &=  bI
\end{align*}
```

The gather environment

Sometimes you need to display a sequence of *unrelated* equations. However, displaying them separately (with separate `$$` commands) puts too much space between one and the next. The `gather` environment fixes this.

Some formulas from high school:

$$A = 4\pi R^2, \quad V = \tfrac{4}{3}\pi R^3$$
$$\sin^2 x + \cos^2 x = 1$$

Forget 'em!

```
Some formulas from high school:
\begin{gather*}
A = 4 \pi R^2, \quad
     V = \tfrac{4}{3}\pi R^3 \\
\sin^2{x} + \cos^2{x} = 1
\end{gather*}
Forget 'em!
```

The matrix environments

The amsmath package also provides simple ways to construct matrices. Besides the `pmatrix` environment shown below (which puts a matrix in parentheses), there are `bmatrix` (brackets), `vmatrix` (single vertical lines—e.g., for determinants), `Vmatrix` (double vertical lines), and `matrix` (no delimiters at all).

Do the matrices

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

generate the group $SL_2(\mathbb{Z})$?

```
Do the matrices
$$
\begin{pmatrix}
1 & 1 \\ 0 & 1
\end{pmatrix}
\quad \text{and} \quad
\begin{pmatrix}
0 & -1 \\ 1 & 0
\end{pmatrix}
$$
generate the group $SL_2(\mathbb Z)$?
```

Packages, like classes, can be declared with optional arguments. For example, the declaration

```
\usepackage[righttag]{amsmath}
```

will put any automatically-generated equation numbers on the right instead of on the left (which is the default). There is a list of $\mathcal{AMS}$ packages and options in §8.6.5 of the *Companion*. Read chapter 8 of the *Companion* for full descriptions of amsmath, or consult the *$\mathcal{AMS}$-LaTeX Version 1.2 User's Guide* (paper or online).

Other packages

shortvrb

This *Sampler* frequently uses typewriter type to denote LaTeX commands. However, the `\texttt` command (which is the successor of the `\tt` command in the old version) will not always work. If you type `\texttt{$x^2$}` with the aim of producing the output `$x^2$`, you will be unsuccessful. TeX will "see" the control characters `$` and `^` and will print x^2. To get the output you want, you must inhibit the action of the control characters; that can be done with the `\verb` command, described in §3.7 of Lamport. The shortvrb package provides a short-cut. Put the following into the preamble:

```
\usepackage{shortvrb}
\MakeShortVerb{|}
```

Then, to produce the output `$x^2$`, put `|$x^2$|` in your text. The line `\MakeShortVerb{|}` declares the character "`|`" to be the delimiter for text you want to appear exactly as typed. You can replace it by any other ordinary character you wish. [How did I type this paragraph?]

timestamp, draft, and timedraft

We have some local input files that help you keep track of drafts. One of them (timestamp) puts the date and time a document is created on the bottom of each page when it is printed; another (draft) puts a large "DRAFT" diagonally across each page; and a third (timedraft) does both. Load (at most one of) these with an `\input` declaration in the preamble:

`\input{timestamp}`,
or `\input{draft}`,
or `\input{timedraft}`.

The printed (but not the screen) version of this page shows the effect of using timedraft.

array

The array package increases the power and flexibility of LaTeX's `tabular` and `array` environments. The first two columns of the table of symbols on page 4 provide an example. I first set it like this:

name	*code*
blackboard bold alphabet	`\mathbb`
"poor man's bold"	`\pmb`

```
\begin{tabular}{p{75pt}l}
\multicolumn{1}{c}{\textit{name}} &
     \multicolumn{1}{c}{\textit{code}}\\ [2pt]
\hline\\[-12pt]
blackboard bold alphabet & |\mathbb| \\
``poor man's bold'' & |\pmb|
\end{tabular}
```

(The shaded text "`p{75pt}`" says that the first column should be set as a `parbox` 75pt wide. Its purpose is to force a long entry (e.g., "blackboard bold alphabet") to spill over to a second line instead of sprawling horizontally. (See

Lamport, §C.10.2.) But text in a `parbox` is justified, so "poor man's bold", looks much too spread out. What's needed is a `\raggedright` declaration to apply to entries in the first column. The array package provides an easy way to this. To make the declaration "*decl.*" apply to a given column in a tabular or array environment, add `>{`*decl.*`}` to the argument for that column.

name	*code*
blackboard bold alphabet	`\mathbb`
"poor man's bold"	`\pmb`

```
\begin{tabular}{>{\raggedright}p{75pt}l}
\multicolumn{1}{c}{\textit{name}} &
     \multicolumn{1}{c}{\textit{code}}\\ [2pt]
\hline\\[-12pt]
blackboard bold alphabet & |\mathbb| \\
``poor man's bold'' & |\pmb|
\end{tabular}
```

The pifont package: symbol and dingbat

Our PostScript printers have two more useful fonts built in: symbol and dingbat. These can be most readily accessed through the pifont package. There are a couple hundred dingbats, and you can refer to them by number, using the `\ding` command. For instance, `\ding{42}` produces "✎", a useful symbol. There is a table of dingbats on page 336 of the *Companion*; it tells you the decimal number keyed to each dingbat.

A similar arrangement works with PostScript symbol font, which has the *short name* `psy`. To obtain it, use the command `\Pisymbol{psy}`. For example, `\Pisymbol{psy}{195}` produces "℘". A keyed table of the symbol font appears on page 338 of the *Companion*.

Output

PostScript and PDF versions

Use `dvips` to generate a Postscript version of your file for printing. (The command `dvips` *filename* will produce *filename*`.ps` from *filename*`.dvi`.) You can also create a a PDF version of your file, for example to make the document available on your Web page. One way is to apply `ps2pdf` to PostScript output: `ps2pdf` *filename*`.ps`. A second way is to use `dvipdf` directly on the DVI file (in the same way you use `dvips`): `dvipdf` *filename*. Both of these include embedded PostScript graphics in the output PDF file. There is a third way to create a PDF file, using `pdflatex`, but this has the disadvantage of eliminating the graphics.

Screen viewing

Use `xdvi` to preview your document on a monitor before printing it. The current version of `xdvi` will display all the special fonts, but it will not display the `timestamp` or `draft` overprinting that will appear on the printed version. You can view the Postscript version itself using `ghostview`; the command is `gv` *filename*`.ps`. The image is not as crisp as the xdvi image, though. You can view the PDF version with Acrobat Reader: `acroread` *filename*`.pdf`. The free viewer `xpdf` also works, but its image is less sharp.