Beautiful technical documents are typeset with TEX

Bob Rosebrugh

Abstract

The TEX typesetting system, introduced by Donald Knuth, has become the *de facto* standard for technical documents in (at least) Mathematics, Physics and Computer Science. Though it is a mark-up language (i.e. anti-wysiwyg) it is remarkably easy to use—with a little practice it is much easier than wysiwyg technical word-processors. More importantly, it gives each of us the chance to produce beautiful technical documents. We describe the steps in TEX'ing a document, with examples.

1 Introduction

Knuth's system [**?**] was begun in the late 1970's, surely ante-diluvian times from a computing perspective. He wanted to build a system which would combine the developing computer typesetting paradigm and Mathematics. Writing it was to occupy part of a sabbatical . . . and several years later the system matured, was adopted by the American Mathematical Society, and several 'macro packages' developed. The most influential was (and is even more so now) LATEX by Leslie Lamport[**?**]. A new version of LATEX called 2ϵ is less than a year old.

After the AMS adopted TEX it rapidly spread around the world. The advent of personal computers powerful enough to put TEX on the desktop (for PC's that means 386 class and above) has led to much wider use. Today, almost all published Mathematics and Computer Science is in TEX at some point during its production (some paper journals employ large numbers of monks.) Preprint servers that are prominent in Physics are TEX-based, and all of the new electronic journals in Mathematics use it.

How does TEX work? There are three essential steps:

- the ‘source code’ including mark-up is produced by any text editor and saved in a file `filename.tex`
- the TEX program ‘compiles’ the source (producing lots of errors at first, then less...) and produces the file `filename.dvi`
- the dvi file is fed to a graphics screen viewer or to a printer driver

Not by chance, a major attraction of the system is its portability. As a mark-up language, the TEX source uses only ASCII characters, like its young relative HTML, so it is ideal for transmitting across networks (we used to have some fun with BITNET, but that’s ancient history!) Knuth defined the output from TEX to be ‘device-independent’—the ‘dvi’ file which actually contains all the font and layout information (it is a relative of Postscript.) Thus, the easily shared source file is all that is needed for any implementation of TEX to produce a viewable and printable result anywhere at all. The result has been a remarkable proliferation of TEXies. It also means that all you need to know is how to do the mark-up, and that’s what the rest of this outline will cover.

The best way to learn TEX’s mark-up is by example, so we’ll provide the source code for this section later. The TEX advantage in technical typing is that the standards of good layout of mathematical symbols are built-in. For example to obtain

$$\int_1^x \frac{1}{t} dt = \log(x)$$

nicely set up requires no tedious cursoring about the page. All the standard math symbols, for example $\infty, \oint, \nabla$ among many others, and Greek letters, from α to Ω are available using mnemonic escapes rather than odd key combinations (and of course one can define one’s own ad lib.) Arrays are handled handily as in

$$\left(\begin{array}{ccc} x & x-y & z \\ z-w & y & x+y+z+w \end{array} \right)$$

again without layout fuss. A host of other capabilities are available, including document structuring commands like:

Theorem 1 *Any map can be four-coloured.*

Tables, tabbing and many other facilities are built-in and almost anything you can imagine has been built by someone and stored in one of the Web/ftp-accessible archives. Those archives are huge, continually updated and mirrored around the world.

To get started with TeX you need to remember is CTAN (for Common TeX Archive Network)—any Web search engine will point the way in a flash. When you get there you will find

- high-quality freeware implementations for your favourite platform, whatever that may be;
- packages of macros to produce easily everything from chemical bond diagrams to Category Theory.
- all sorts of documentation, even a book introducing TeX by Canadian mathematician Michael Doob.

Let's finish this appetizer with a picture:

$$\begin{tikzcd}
 & B \arrow[r] & C \\
A \arrow[ur] & &
\end{tikzcd}$$

Category Theory or Vectors?

Example source code is on the next page.

References

[1] Donald Knuth, *The TeXbook*, 1984, Addison-Wesley.

[2] Leslie Lamport, *LaTeX, User's guide and reference*, 1986, Addison-Wesley.

```
\documentstyle[12pt]{article}
\title{Beautiful technical documents\\ are typeset with \TeX} ...
\begin{document}
\maketitle ...
\section{Introduction}
Knuth's system \cite{ktb} was begun in the late 1970's, ...

For example to obtain
$$ \int_1^x \frac{1}{t} dt = \log(x) $$
...math symbols, for example $\infty, \oint, \nabla$ among many others, and
Greek letters, from $\alpha$ to $\Omega$ are available using
...
$$ \left( \begin{array}{ccc}
              x   & x-y & z\\
              z-w &  y  & x+y+z+w
          \end{array}\right)
$$
...including document structuring commands like:

\begin{theorem}\label{4-color} Any map can be four-coloured.
\end{theorem}
...
appetizer with a picture:
\begin{center}
\begin{picture}(100,200)
\put(10,10){A}
\put(20,20){\vector(1,1){50}} ...
\end{picture}

Category Theory or Vectors?
\end{center}

\begin{thebibliography}{99}
\bibitem{ktb} Donald Knuth, {\em The \TeX book}, 1984, Addison-Wesley.
...\end{thebibliography}
```