TeX and LaTeX 2e

Michael Downes

Most mathematicians today use some variant of TeX to write their mathematical papers and books. Created around 1980 by computer scientist Donald E. Knuth, TeX dramatically changed the process of preparing and distributing mathematical literature. In the ensuing two decades various streams of TeX-related development have sprung up, diverged, converged, and sometimes evaporated in the face of newer software, with the associated terminology proliferating in equal measure—as anyone familiar with software evolution would expect. The aim of this article is to explain some of the terminology and clarify certain distinctions of interest for mathematicians.

Currently, for authors who intend to publish an article or book with the AMS, writing it with LaTeX is particularly recommended because the LaTeX document format is

1. oriented towards capturing the inherent logical structure of the document, which is critically important for long-term archiving;
2. capable of serving as a source from which many other formats can be automatically generated (e.g., HTML, PDF);
3. well established and stable (also good for archiving purposes);
4. readily exchangeable with colleagues;
5. both standardized and flexible in a way that seems well suited to mathematical material;
6. easy to feed directly into the AMS production system.

There is a significant distinction between the current version of LaTeX, known as LaTeX 2e, and the preceding version, known as LaTeX 2.09, superseded by LaTeX 2e in 1994. The AMS definitely recommends LaTeX 2e to its authors and advises anyone still using LaTeX 2.09 to phase it out of use at the earliest reasonable opportunity, because LaTeX 2e is much easier to work with, both for authors and for publishers.

Michael Downes is a publishing technology specialist at the American Mathematical Society. His email address is mjd@ams.org.

How Do I Use LaTeX to Write a Document?

Strictly speaking, you don't. You use some other program to write a LaTeX document, and then you invoke LaTeX to *typeset* or *compile* the document. Just about any mainstream text-editor program such as Alpha, Emacs, BBEdit, WinEdt, or vi can be used to write LaTeX documents. Yes, even NotePad in a pinch.

The result of the typesetting is a DVI or PDF file, which you can then print or view on screen at your leisure, using other programs such as `xdvi`, Acrobat Reader, `ghostview`, or `dvips`. In effect, by a slight change of perspective, the *typeset* operation could be understood to mean *save as DVI* or *save as PDF*.

Some programs (e.g., Scientific Word or Textures) integrate the multistep process into something closer to a typical word processor, or WYSIWYG ("What You See Is What You Get") interface, but the fact that LaTeX software is mostly non-WYSIWYG is normally regarded as a virtue rather than a drawback. Among other things, a non-WYSIWYG approach helps sensitize authors to the kind of discrimination between visual appearances and essential information that they need to make if they do not want what they write to be inadvertently encumbered by limitations of the medium (or software, or printer, or type of computer monitor) in which it is originally produced.

How Do I Get LaTeX If I Do Not Have It?

One of the best ways to get LaTeX up and running on a new computer is with the TeX Live CD [12] offered as a benefit of membership in the TeX Users Group. (The CD is not sold separately; the only way to get one is by becoming a member.)

Some other suggested sources for getting LaTeX may be found on the AMS *TeX Resources* webpage [11]. The main decision might be whether to go with one of the free TeX systems or pay money for a commercial one. A commercial TeX system will not be cheap (usually $300–$500), but apart from coming with a telephone number to call technical support, it will also tend to be easier to install and more tightly integrated into a given operating system. The Y&Y TeX system, for example, includes a capability for saving LaTeX equations to the Windows clipboard in Windows MetaFile format so that they can be pasted easily into other Windows applications.

From TeX to LaTeX

Rather than attempting to be all things in a single program, TeX is designed with modularity in mind. Thus TeX itself provides only fundamental typesetting capabilities and does not incorporate editing, printing, or previewing capabilities. Instead, the result of running TeX is a graphics file in a format called DVI (for device-independent) that is designed to make it as easy as possible for other programs to print or preview DVI files on an arbitrary printing device or computer screen. This may seem unremarkable nowadays, but it was far from commonplace back in 1980 when Knuth was developing TeX. At that time, the publisher's version of an article or book was usually held in a proprietary format that could be viewed or printed only with special-purpose commercial typesetting equipment.

The typesetting operations of TeX are applied on a very low level. They address the tasks of

- stringing characters together in words and paragraphs,
- positioning symbols properly in mathematical formulas,
- automatically finding good page breaks, and
- dealing with footnotes and other floating objects (such as figures and tables).

For authors, however, it is preferable to work on a higher level. For example, instead of writing

```
\begingroup
\rightskip=0pt plus.2\hsize
\leftskip=\rightskip
\parindent=0pt \parfillskip=0pt
\noindent
...
\par \endgroup
```

every time some text needs to be centered, the usual practice would be to define abbreviations (also known as TeX *macros*) such that one could get the same results by writing

```
\center ... \endcenter
```

By design, then, TeX is almost always used in conjunction with an auxiliary piece of software called a TeX *format* whose purpose is to bridge the gap between the low-level typesetting functions of TeX and a higher-level interface more suitable for authors. A TeX format is made by assembling all of the TeX macros that define the higher-level interface and precompiling them as a unit in order to reduce start-up time.

We are now in position to state a key point of the terminology: **LaTeX is a TeX format**. Some other well-known TeX formats are Plain TeX, AMS-TeX, `eplain`, `texinfo`, and `ConTeXt`. When we speak of running LaTeX, what we are really doing is running TeX+LaTeX; running `texinfo` means running TeX+`texinfo`, and so on.

The Plain TeX Format

Plain TeX is the generic example format that Knuth wrote to be distributed with TeX. It is not really designed for serious publishing use; for example, it provides only one font size: 10-point. Adding support for other sizes is not exactly difficult, but one has to do it oneself, and it can be rather tedious and error prone, especially when math fonts are involved. Plain TeX was, however, incorporated as a base element into just about all of the other TeX formats that came after it (vestiges in LaTeX include `\endgraf`, `\null`, `\empty`, `\slash`, `\*`, and `\pmatrix`, though none of these are documented in the LaTeX book [1]).

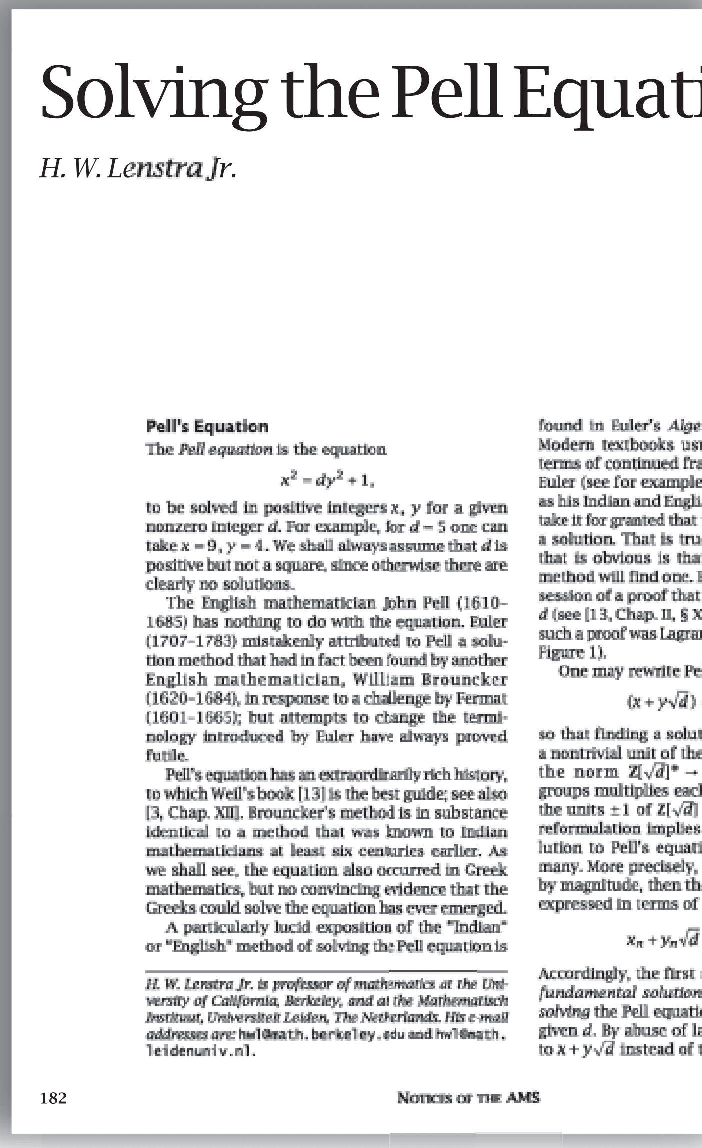
The AMS-TeX Format

Early on, when TeX first came to the attention of the AMS, the promise of putting high-quality mathematical typesetting into the hands of authors was extremely persuasive. The prospect of being able to directly use electronic files provided by the authors instead of retyping everything from manuscripts seemed absolutely compelling. It became apparent, however, after a little experimentation that something more than Plain TeX would be needed for AMS material. The AMS therefore underwrote the development of a TeX format that would be better able to handle the kind of material typically found in AMS publications. Although this format, "AMS-TeX", was used as early as 1981—for a short announcement in the *Notices* that a draft version of the AMS-TeX book could be purchased from the AMS (*The Joy of TeX* by Michael Spivak)—there were caveats in every AMS-TeX-related announcement over the next couple of years that it was still a work in progress. Some additional overhaul necessitated by the appearance of TeX82 was the occasion of further delay (cf. the acknowledgements in the 1986 printing of *The Joy of TeX*). So all in all it seems best to consider that AMS-TeX as we know it became available in 1984.

The LaTeX Format

LaTeX is a TeX format written by computer scientist Leslie Lamport in 1983–5. It was modeled in many respects on a non-TeX precursor called Scribe. Like Scribe, LaTeX takes as one of its central principles that authors are better off concentrating on logical design rather than visual design when writing their documents. This is a step beyond the mere aggregation of lower-level typesetting details into

```
\documentclass{notices}
\usepackage{amsmath}
\usepackage{amsthm}
\usepackage{url}

\newtheorem*{thm}{Theorem}

\newcommand{\term}[1]{\emph{#1}}

\begin{document}

\title{Solving the Pell Equation}
\author{H. W. Lenstra Jr.}

\note{H. W. Lenstra Jr.\@ is professor of
  mathematics at the University of
  California, Berkeley, and at the
  Mathematisch Instituut, Universiteit
  Leiden, The Netherlands. His e-mail
  addresses are:
  \url{hwl@math.berkeley.edu} and
  \url{hwl@math.leidenuniv.nl}.}

\maketitle

\section{Pell’s Equation}

The \term{Pell equation} is the equation
\begin{displaymath}
x^{2} = dy^{2} + 1,
\end{displaymath}
to be solved in positive integers $x$, $y$
for a given nonzero integer $d$. For
example, for $d = 5$ one can take $x = 9$,
$y = 4$. We shall always assume that $d$ is
positive but not a square, since otherwise
there are clearly no solutions.

   ...
```


Solving the Pell Equati

H. W. Lenstra Jr.

Pell's Equation

The *Pell equation* is the equation

$$x^2 = dy^2 + 1,$$

to be solved in positive integers x, y for a given nonzero integer d. For example, for $d = 5$ one can take $x = 9$, $y = 4$. We shall always assume that d is positive but not a square, since otherwise there are clearly no solutions.

The English mathematician John Pell (1610–1685) has nothing to do with the equation. Euler (1707–1783) mistakenly attributed to Pell a solution method that had in fact been found by another English mathematician, William Brouncker (1620–1684), in response to a challenge by Fermat (1601–1665); but attempts to change the terminology introduced by Euler have always proved futile.

Pell's equation has an extraordinarily rich history, to which Weil's book [13] is the best guide; see also [3, Chap. XII]. Brouncker's method is in substance identical to a method that was known to Indian mathematicians at least six centuries earlier. As we shall see, the equation also occurred in Greek mathematics, but no convincing evidence that the Greeks could solve the equation has ever emerged.

A particularly lucid exposition of the "Indian" or "English" method of solving the Pell equation is

H. W. Lenstra Jr. is professor of mathematics at the University of California, Berkeley, and at the Mathematisch Instituut, Universiteit Leiden, The Netherlands. His e-mail addresses are: hwl@math.berkeley.edu and hwl@math.leidenuniv.nl.

found in Euler's *Algeb*
Modern textbooks usu
terms of continued frac
Euler (see for example
as his Indian and Englis
take it for granted that t
a solution. That is true,
that is obvious is that
method will find one. F
session of a proof that t
d (see [13, Chap. II, § XII
such a proof was Lagran
Figure 1).

One may rewrite Pell

$(x + y\sqrt{d}) \cdot$

so that finding a soluti
a nontrivial unit of the
the norm $\mathbf{Z}[\sqrt{d}]^* \to \mathbf{Z}$
groups multiplies each
the units ± 1 of $\mathbf{Z}[\sqrt{d}]$ a
reformulation implies
lution to Pell's equatio
many. More precisely, i
by magnitude, then the
expressed in terms of t

$x_n + y_n\sqrt{d} =$

Accordingly, the first s
fundamental solution
solving the Pell equatio
given d. By abuse of la
to $x + y\sqrt{d}$ instead of th

182 NOTICES OF THE AMS

Figure 1. A typical LaTeX file (cf. "Solving the Pell Equation", by H. W. Lenstra Jr., *Notices of the AMS* 49 (2002), 182–92).

a convenient higher-level set: defining macros `\vspace`, `\centering`, `\Large` makes it possible to write a section title as

```
\vspace{1.5cm}
\begin{centering}\Large
2. Section Title
\end{centering}
```

But LaTeX goes further: authors simply write

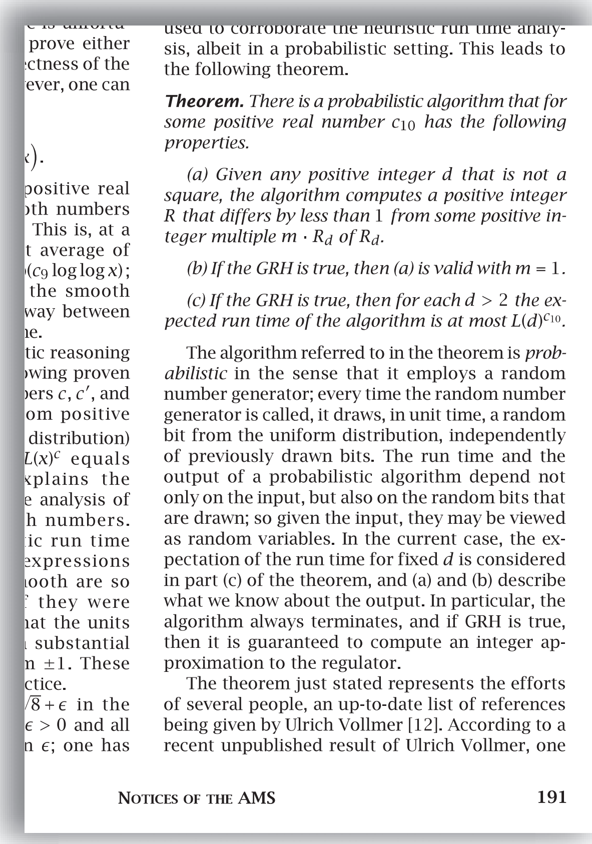
```
\section{Section Title}
```

and not only the visual appearance but the numbering is taken care of automatically. In other words, authors are encouraged to write in a way that describes the material conceptually, rather than visually—a practice sometimes referred to as *logical markup* or *conceptual markup*.[1] This will be recognizable to many as equivalent to the *stylesheet* feature of modern word processors; Lamport was well ahead of the curve in 1983 in recognizing the importance of this approach and building a TeX format around it that made it available to authors.

Another noteworthy early feature of LaTeX is that it was designed to interoperate nicely with auxiliary programs `makeindex` and BibTeX, which help automate the tasks of making indexes and bibliographies. The original versions of these programs are starting to show their age, but they continue in active use today—and a usable lifetime of over

[1] *Cf.* Standard Generalized Markup Language, `http://xml.coverpages.org/sgml.html`; *SGML became an ISO standard in 1986 and was the ancestor of HTML (HyperText Markup Language) and XML (Extensible Markup Language).*

```
\begin{thm}
There is a probabilistic algorithm that for some
positive real number $c_{10}$ has the following
properties.
\begin{enumerate}
\item \label{first} Given any positive integer $d$
that is not a square, the algorithm computes a
positive integer $R$ that differs by less than $1$
from some positive integer multiple $m\cdot R_d$ of
$R_d$.

\item If the GRH is true, then \ref{first} is valid
with $m = 1$.

\item If the GRH is true, then for each $d > 2$ the
expected run time of the algorithm is at most
$L(d)^{c_{10}}$.
\end{enumerate}
\end{thm}

The algorithm referred to in the theorem is
\term{probabilistic} in the sense that it employs a
random number generator; every time the random
number generator is called it draws, in unit time, a
random bit from the uniform distribution,
independently of previously drawn bits.
```



used to corroborate the heuristic run time analysis, albeit in a probabilistic setting. This leads to the following theorem.

Theorem. *There is a probabilistic algorithm that for some positive real number c_{10} has the following properties.*

(a) Given any positive integer d that is not a square, the algorithm computes a positive integer R that differs by less than 1 from some positive integer multiple $m \cdot R_d$ of R_d.

(b) If the GRH is true, then (a) is valid with $m = 1$.

(c) If the GRH is true, then for each $d > 2$ the expected run time of the algorithm is at most $L(d)^{c_{10}}$.

The algorithm referred to in the theorem is *probabilistic* in the sense that it employs a random number generator; every time the random number generator is called, it draws, in unit time, a random bit from the uniform distribution, independently of previously drawn bits. The run time and the output of a probabilistic algorithm depend not only on the input, but also on the random bits that are drawn; so given the input, they may be viewed as random variables. In the current case, the expectation of the run time for fixed d is considered in part (c) of the theorem, and (a) and (b) describe what we know about the output. In particular, the algorithm always terminates, and if GRH is true, then it is guaranteed to compute an integer approximation to the regulator.

The theorem just stated represents the efforts of several people, an up-to-date list of references being given by Ulrich Vollmer [12]. According to a recent unpublished result of Ulrich Vollmer, one

NOTICES OF THE AMS 191

Figure 2. An example theorem as it would usually be done in LaTeX (from "Solving the Pell Equation", by H. W. Lenstra Jr., *Notices of the AMS* 49 (2002), 182–92).

fifteen years is no small accomplishment in the software industry.

"Old" LaTeX 2.09 versus "New" LaTeX 2e

LaTeX 2e is the name used when distinguishing the current version of LaTeX from its predecessor, LaTeX 2.09. At the time when the work for LaTeX 2e was being carried out (1994), there were also ambitious plans laid for a substantially new and improved version of LaTeX, to be known as LaTeX 3. Thus LaTeX 2e was intended chiefly to consolidate the existing state of LaTeX and unify some branches of development that had begun branching off in their own directions (including NFSS and AMS-LaTeX; see below). Even so, LaTeX 2e does include some significant improvements over LaTeX 2.09:

1. LaTeX 2e shields the author from various kinds of troublesome complications concerning fonts, especially mathematics fonts.
2. LaTeX 2e provides a powerful unified interface for putting various kinds of figures and diagrams into a document, a notorious source of difficulties for authors.
3. LaTeX 2e has a coherent "package" system that makes it fairly easy to write special-purpose packages that add new capabilities. Consequently, a great many extension packages for LaTeX 2e are available, and more are being added all the time.
4. LaTeX 2e gives authors access to certain modern packages that do not work at all with old LaTeX. One example is the `hyperref` package, which facilitates producing PDF files from LaTeX documents with active links for bibliography citations, equations, figures, theorems, and sections.

As for LaTeX 3, it has not yet been released, perhaps chiefly (in my opinion) because the ambitious scope of the project has not been matched by sufficient funding. One would have hoped to see it reach a level that could support at least one full-time programmer. Although a fair amount of progress has been made anyway, it is the product of dedicated volunteers doing LaTeX development in their spare time.

Is It Old LaTeX or LaTeX 2e?

Older LaTeX documents begin with `\documentstyle`, whereas LaTeX 2e documents begin with `\documentclass`. There are two other major differences: In old LaTeX, packages are invoked via the option list of the `\documentstyle` command, whereas in LaTeX 2e they are invoked through a separate command, `\usepackage`; and in old LaTeX, font changes have the form `{\bf...}`, whereas in LaTeX 2e they have the form `\textbf{...}` or `\mathbf{...}`. (The old forms of the font commands still work in LaTeX 2e, for the sake of compatibility, but in new documents it is advisable to avoid them.)

Here are some other commands (to name a few) that would indicate a document written specifically for LaTeX 2e, since they were not present in LaTeX 2.09:

```
\providecommand        \begin{lrbox}
\emph                  \frontmatter
\includegraphics       \backmatter
```

What Is AMS-LaTeX?

Is AMS-LaTeX a TeX format too? Well, yes and no. It used to be, but only for pragmatic reasons, not because it differed very much from LaTeX. And nowadays it makes little sense to ask "Should I use LaTeX or AMS-LaTeX?" because using AMS-LaTeX implies using LaTeX, just as using LaTeX implies using TeX.

Much of the development of AMS-TeX and LaTeX occurred in parallel in the early 1980s when TeX was in its infancy (see "Some Historical Notes" below). In many respects the features they offered were complementary rather than redundant. AMS-TeX took more pains than LaTeX to address certain finer points of mathematical typography: for example, getting good interline spacing in matrices or proper positioning of doubled math accents. On the other hand, the lack of automatic numbering and cross-referencing in AMS-TeX was a notable drawback. Nor did it have facilities comparable to LaTeX's for handling indexes, working with a separate database file for bibliographic data, splitting books up into chapters, or producing simple pictures. Growing recognition of this complementarity led to a rising demand in the late 1980s among mathematician-authors for a way to get the best features of both formats.

Obviously, combining AMS-TeX and LaTeX could have been done by adding LaTeX features to AMS-TeX or adding AMS-TeX features to LaTeX. Having settled after some debate on the latter option, the AMS sponsored the necessary development in 1989–90 that led to the `amsmath` and `amssymb` packages described in the *Short Math Guide for LaTeX* [3], which were distributed together with some AMS document styles, `amsbook` and `amsart`. This distribution was called collectively AMS-LaTeX.

There was one key problem in LaTeX 2.09 that had to be solved in the development of AMS-LaTeX. The handling of math fonts just was not good enough. Misfeatures such as printing bold subscripts at full size instead of subscript size were typical. And in LaTeX 2.09 most of the font setup was built into the format, making it difficult to correct the problems without changing the format itself.

Fortunately, a solution for this already existed, thanks to a couple of German programmers (Frank Mittelbach and Rainer Schöpf), in the form of a thoroughly overhauled and improved font handling scheme for LaTeX known as NFSS, for *New Font Selection Scheme*. In order to simplify installation, a copy of NFSS was included in the AMS-LaTeX distribution along with instructions on how to build a new LaTeX format with it. Depending on how it was installed, this format was called LaTeX+NFSS or AMS-LaTeX.

When LaTeX 2e came out, it incorporated the New Font Selection Scheme as standard into the LaTeX format. This meant that since 1994 it has no longer been necessary to build a separate format called AMS-LaTeX: one simply uses AMS packages with LaTeX in the same way as one uses packages from any other source. For reasons of history and convenience, the ones that originated in the old AMS-LaTeX distribution are sometimes still referred to collectively as AMS-LaTeX. Any decent TeX system that one gets nowadays will include copies of all the AMS-LaTeX files as a matter of course, along with many other extra LaTeX packages.

Further Progress in Conceptual Markup

In the years since LaTeX first appeared there has been a steady increase in awareness among publishers and authors of the importance of conceptual markup for capturing the essence of written material, especially when the material has great intrinsic complexity as, for example, in the scientific literature. Although LaTeX was a major step forward, in some areas the principles of conceptual markup were only sketchily applied, e.g., math formulas and bibliographies.

The claim that LaTeX does not sufficiently apply the principle of conceptual markup for bibliographies is in a sense unfair, because the usual method is to use BibTeX, which breaks down the components of a typical reference in a natural and well-designed way. The only problem is that, in practice, the conceptual markup present in the BibTeX database file is lost when entries are imported into a LaTeX file. As a consequence, the bibliography part of the LaTeX document ends up being less well structured than one would wish if it is to be used as the primary archival form of the document (and it probably should be if it is the form used by the author in the writing process).

This loss of internal structure was the primary impetus behind the development of a LaTeX package called `amsrefs`, released in January 2002 [15]. Using the `amsrefs` package makes it easy to retain the internal structure when importing bibliography entries into a LaTeX file.

Concerning math formulas, noteworthy recent developments outside of LaTeX include the MathML [13] and OpenMath [14] initiatives. An adequate discussion of these is beyond the scope of this article, but any readers who are somewhat acquainted with them already will recognize that converting LaTeX math formulas to *Presentation MathML* is far more feasible than converting them to *Content MathML* and that this is precisely because many elements in a LaTeX math formula are described in terms of visual appearance rather than meaning. If something is written in LaTeX as a superscript, human readers have little difficulty sorting out the actual meaning, but conversion software cannot readily tell whether the superscript means "exponent" or something else, and the distinction is essential when converting to Content MathML or OpenMath.

Some Historical Notes about TeX

On January 4, 1978, when Donald Knuth gave his Gibbs Lecture in Atlanta on "Mathematical Typography" (*Bull. Amer. Math. Soc. (N.S.)* **1** (1979), 337–72), the development of TeX was well under way, and a usable version of TeX was released into the wild—to people outside Stanford, that is—in September 1978. Over the next year or two of real-world use by himself and others, however, Knuth began to view that first version with increasing dissatisfaction and by 1980 felt compelled to rewrite it almost from scratch. Among other things, he wanted to change the programming language from SAIL to PASCAL, since the limited availability of SAIL on other computer systems hindered many people from using TeX. In the rewriting process he added many features needed for professional-quality typesetting and made some significant syntactic changes in the TeX macro language, particularly with respect to conditionals and fonts.

Although Knuth denominated the new version TeX82 to distinguish it from its precursor (TeX78), the change log shows that a number of significant primitives were still being added in late 1982 and early 1983—some of them in response to feedback from Leslie Lamport, who was hard at work on LaTeX at that time:

12/02/82:`\everymath`,`\everydisplay` added
12/02/82:`\futurelet` added
12/07/82:`\endinput` added
12/25/82:`\jobname primitive`
12/27/82:`\pagetotal`,`\pagegoal` added
01/06/83:`\pageshrink` etc. added
01/06/83:`\floatingpenalty` etc. added

Several almost-final versions, 0.99, 0.999, ..., of TeX were released in 1983 to permit others to try them out and report bugs, but it was not until December 3, 1983, that version 1.0 was released. This date probably has a better claim than any other to be considered the true birthday of TeX as we know it; it is no coincidence that usage of TeX really took off after that, with new formats blooming into life everywhere like desert flowers after a rainstorm.

The version numbers of TeX increased to 2.x and eventually 3.x as time went by. The release of version 3.0 in March 1990 established the set of primitives (built-in TeX commands) and behavior that constitute what I would call "standard TeX" and that is described in current editions of *The TeXbook* [16]. Although there has been some further development since then by people other than Knuth, it has been done under different names (e-TeX, Omega, pdfTeX, and so forth) and is not covered by *The TeXbook*.

In retrospect it seems fitting to consider TeX78 as an alpha version of the software, giving us this chronology:

- TeX78 = alpha version
- TeX82 preliminary releases (1982–3) = beta version
- TeX 1.0 = first version of TeX as we know it: December 3, 1983
- TeX 2.0 = November 27, 1985[2]
- TeX 3.0 = the last major version of TeX: March 25, 1990
- TeX 3.14159... = **the** final version of TeX

Knuth has chosen to increment the minor version number after 3.0 by using the digits of π, adding a digit whenever a new bug fix is done (a rare event nowadays), with the idea that after his death TeX itself will be permanently frozen and the version number will change from an approximation of π to π itself. The approximation currently stands at 3.14159.

e-TeX, pdfTeX, and Omega

Three descendants of TeX seem worth singling out for special mention. Among other things, many readers may find that they already have working copies of all three included in their current TeX system.

e-TeX. Practically speaking, this version of TeX is quite close to the original, since many of its enhancements address technical limitations and are noticeable mainly to programmers. But two of the more visible additions are (1) support for right-to-left typesetting, as needed for languages like Arabic, and (2) a `\middle` command to go along with `\left` and `\right`.

Omega. The chief distinction of Omega is its ambitious goal of doing professional-quality typesetting in all the world's languages: Arabic, Chinese, Greek, Japanese, Sanskrit, Tibetan, etc. The set of languages that one can practically handle with Omega falls short of being all-inclusive, of course, but the remarkable thing is how many languages are supported already.

[2]*Conjectural; the historical record for this release seems to be unclear. The 11/27/85 date is the date of the last change recorded in* `tex82.bug` *after the release of version 1.5 and prior to other changes designated as belonging to version 2.1. An announcement by David Fuchs in the March 1986 issue of* TUGboat *stated that "TeX1.5, when used with the new CM fonts, is officially called TeX2.0." Should this be interpreted, perhaps, to mean that the release date of 2.0 is the same as for 1.5?*

pdfTeX. As its name suggests, pdfTeX is much like TeX but directly produces its output in PDF form (Portable Document Format) rather than DVI. Many people are using it nowadays because it enables authors to readily achieve a number of useful PDF effects while continuing to write LaTeX files in their accustomed way.

Since e-TeX, pdfTeX, and Omega are variants of TeX, they too work with formats in the same way as TeX. This increases the potential format names in combinatorial fashion: in theory for e-TeX we could have Plain e-TeX, e-AMS-TeX, e-LaTeX, e-ConTeXt, and similarly for the others. In practice we have `etex` (= Plain e-TeX) and `elatex`, `pdftex` and `pdflatex`, `omega` and `lambda` (= `omegalatex`), and `gamma` (= `omegacontext`). Because the enhancements in e-TeX and pdfTeX do not interfere with each other, they can be, and have been, combined, giving us also `pdfetex` and `pdfelatex`.

Some other alternatives on the horizon—e.g., ConTeXt, Publicon™, or XML/Docbook+MathML—are already sufficiently usable to make them interesting avenues of experimentation for more intrepid authors, but it seems premature to recommend them for everyone. Some links for these and other software are given on the webpage *Authoring Software for Mathematicians* (`http://www.ams.org/tools/authoring-software.html`).

Further Information

For someone starting out with LaTeX, if mathematics is an important part of the material to be written, here is the minimal set of documentation that I would recommend:

[1] *LaTeX: A Document Preparation System,* 2nd edition, Leslie Lamport, Addison-Wesley, 1994. This is the authoritative primary reference for LaTeX. *Note: It is important to follow the 2nd edition (1994) rather than the 1st edition (1985), because the 1st edition is for LaTeX2.09, while the 2nd edition describes LaTeX 2e.*

[2] *LaTeX Guides.* These are partly intended to describe the differences between LaTeX 2.09 and LaTeX 2e, but they also serve as a good overview for some areas that had only sketchy coverage in the first edition of the LaTeX book: industrial-strength font setup, incorporating graphics, writing a new package.

- LaTeX 2e *for authors,* `http://www.latex-project.org/guides/usrguide/usrguide.html`
- LaTeX 2e *graphics guide,* `http://www.ctan.org/tex-archive/macros/latex/required/graphics/grfguide.ps`
- LaTeX 2e *font selection,* `http://www.latex-project.org/guides/fntguide/fntguide.html`
- LaTeX 2e *for class and package writers,* `http://www.latex-project.org/guides/clsguide/clsguide.html`

These guides are free, and if you have a decent LaTeX system, you should find that they are already present on your computer in DVI, PostScript, or PDF form: look for, e.g., …`/texmf/doc/latex/base/usrguide.pdf`, `.dvi`, or `.ps`.

[3] *Short Math Guide for LaTeX*, `http://www.ams.org/tex/short-math-guide.html` (about 20 pages). Another free resource. It offers a concise overview of the features in LaTeX and the `amsmath` package that authors are likely to need when writing math formulas. It includes a list of the math symbols that are normally available in any standard installation of LaTeX. For questions about how to get other fonts and math symbols beyond those described therein, see the *Comprehensive LaTeX Symbol List* (below), which lives up to its name.

Other Useful Resources

[4] *Comprehensive LaTeX Symbol List,* S. Pakin. This is a rather large document to download, because it uses so many different math fonts! You will almost certainly want to get the PDF or PostScript version. `http://www.ctan.org/tex-archive/info/symbols/comprehensive/`

[5] *Math into LaTeX*, G. Grätzer, Birkhäuser, 2000. A book-length treatment of AMS-LaTeX. It contains, among other things, some useful information about practical aspects of using LaTeX that are not often addressed in other LaTeX documentation.

[6] *Using Imported Graphics in LaTeX 2e*, K. Reckdahl. `http://www.ctan.org/tex-archive/info/epslatex.pdf`

[7] *AMS-LaTeX* (webpage with various links), `http://www.ams.org/tex/amslatex.html`; see also `http://www.ams.org/tex/author-info.html`.

[8] *References for TeX and Friends,* P. Karp and M. Wiedmann. Among other things, includes individual documentation for each LaTeX command. `http://www.miwie.org/tex-refs/`

[9] *The Not So Short Introduction to LaTeX 2e*, T. Oetiker et al. This covers ground similar to the Lamport book in about 100 pages. It was originally written in German but has translations available in other languages. `http://www.ctan.org/tex-archive/info/lshort/english/`

[10] *A Guide to LaTeX*, H. Kopka and P. Daly, Addison Wesley, 1999. This is a lengthier treatment of LaTeX (about 600 pages), with many good examples and more depth of detail than the Lamport book. New or casual users might find it overkill.

[11] *TeX Resources,* `http://www.ams.org/tex/`; see also *What Is LaTeX?,* `http://www.ams.org/tools/what-is-latex.html`.

[12] *TeX Live CD,* `http://www.tug.org/texlive.html`; see also the *TeX Users Group* home page, `http://www.tug.org/welcome.html`

[13] *MathML,* `http://www.w3.org/Math`

[14] *OpenMath,* `http://www.nag.co.uk/projects/openmath/omsoc/society/description.html`

[15] *The amsrefs package,* `http://www.ams.org/tex/amsrefs.html`

[16] *The TeXbook,* D. E. Knuth, Addison Wesley, 1984. Describes TeX and the Plain TeX format.

There is a series of *Companion* books that are also worth looking at: *The LaTeX Companion*, *The LaTeX Graphics Companion*, *The LaTeX Web Companion*.

Beware of Obsolete Documentation!

In the case of *The LaTeX Companion*, it should be noted that Chapter 8, on mathematics, in the first edition (1994) became obsolete in some crucial details soon after publication, thanks to some unfortunate timing problems. Pending the appearance of the second edition, one could make do by using the first edition in conjunction with something more up-to-date for math-related questions, such as the *Short Math Guide* mentioned above.

But this is only one instance of a more general pitfall that LaTeX users should be careful to watch out for. There is a lot of helpful information freely available out there on the World Wide Web, but if one downloads something called, for example, "Essential Mathematical LaTeX" that bears a date of September 1989, it seems obvious that in the year 2002 whatever good and useful advice it may contain remains good and useful *only for those who are still using the version of LaTeX that was extant in 1989* and that advice that was originally good may now be downright bad for the unsuspecting reader if it concerns features which have changed in the meantime.

A variation of this pitfall is to go astray indirectly by following the example of a colleague. Your colleague's example might have been exemplary indeed when she first started writing that book of hers seven years ago using LaTeX2.09, but if you are just now beginning to write a book of your own, it behooves you to reflect for your own sake on the passing of time to see whether any labor-saving devices have been invented in the meantime that you can take advantage of. For example, using the LaTeX 2e font commands instead of their 2.09 predecessors would save you the bother of fussing with "italic corrections".