Why TEX?

Jim Hefferon
St Michael's College
http://joshua.smcvt.edu/hefferon.html

From time to time someone may ask you for a list of TEX's strengths. They may want to explain to an administrator why to install it, they may have been advised to use it, or they may simply have found it on their system and want to know what it does. Starting on the next page is a list that you can give to them (it starts on a fresh page to make neater copies). It is understandable to anyone with experience in computing systems or programming.

This article grew out of a Usenet discussion[1] started by Fabrice Popineau. I'd like to thank the contributers to that discussion, who are too numerous to name singly. I would like also to thank Lynne Hefferon and Peter Flynn for their help.

[1] Thread: *10 best reasons to use TeX ?* on `comp.text.tex`, Nov 2001.

Why TEX?

TEX is a system for computer typesetting — for placing text on a page. It is well known in the area of typesetting mathematics and other technical material.

But 'well known' is a relative term. Most computer users who are not scientists do not know TEX. This document is for you if you have heard a little and want an overview of its strengths.

1 Introduction

You no doubt already use other tools so we can start by comparing TEX to the two most common ways of placing text on a computer.

1.1 Compared to word processors

Most people arrange material on a page with a word processor.

Word processors are easy to begin with. To get a blank line between two paragraphs you enter it in. To make a reference to the bibliography you type it into the text in the style that you need. Seems simple. You know what you want and you just do it.

But as the document gets to be a bigger and tougher job, laying it out yourself becomes a problem. Seeing that there is the same amount of vertical space between all of the paragraphs in a twenty-page article is error-prone work. So is making sure that all of the bibliographic entries follow the requirements. And, very few authors have the knowledge and aesthetic eye to correctly place and size symbols in an equation.

In contrast, TEX authors find it easy to have systematic formatting, even when they have complicating elements such as mathematics or tables.

So TEX is like a word processor in that both put text on a page. But it is different in many ways, one of which is that it automates much of the job.

1.2 Compared to HTML

In HTML you might emphasize a point with italics by typing '`it's <i>hot</i> here`' and only later, as the material is run through a browser, is it actually formatted. TEX works in the same way: you might type '`it's \emph{hot} here`' and later run that file through the program to do the typesetting.

Thus TEX is like HTML in that the step of typing the material is separate from the step of setting the material. Unlike HTML, however, TEX can be used as a programming language. You can branch with '`if`' constructs, use variables, etc. So, while becoming a TEX expert takes longer than becoming an HTML expert, the gain is that TEX gives you the power to do more.

This power has allowed people in the TEX community to accomplish great things with it and, even if you never become a wizard yourself, you can use their wizardry in your work. Browse around the Comprehensive TEX Archive Network[2] and you will see that there are solutions available, usually freely available, for a very wide range of needs.

1.3 TEXing

An installation of TEX has many components. But for a first impression, the figure on the next page gives the core flow of information. The material that you have entered in `filename.tex` is processed (technically, 'expanded by the macro package') into a form that is understandable to the main TEX engine, which performs the typesetting. Then you convert the result to the output format that you want. We'll discuss parts of this flow in the next section.

2 Ten Reasons

These are the reasons most often cited for using TEX, grouped into four areas: Output Quality, Superior Engineering, Freedom, and Popularity.

Output Quality

You write documents to be read and understood. Your first concern should be: how good is the output? Is it as readable and as useful as possible? Is it, even, beautiful?

(*i*) TEX has the best output What you end with, the symbols on the page, is as usable — and beautiful — as a non-professional can produce.

This especially holds for complex documents, such as ones with mathematics; see Appendix A. It also holds for documents that are complex in other ways: with many tables, or many cross references or hyper-links, or just with many pages.

The usual way of working with a word processor, clicking the material in, is not suited to complex documents so we cannot fairly compare such output with TEX. However, even on simple documents TEX has advantages. Compare the two samples in Appendix B. These are short and the typographic differences are subtle but even a non-expert may see that the TEX page looks "more right". The word processor's page has some lines with wide gaps between words and some lines with too many words stuffed in (contrast the second paragraph's second line with its third). TEX's output is more readable.

[2] try `http://www.ctan.org/tex-archive/macros/latex/contrib/supported`

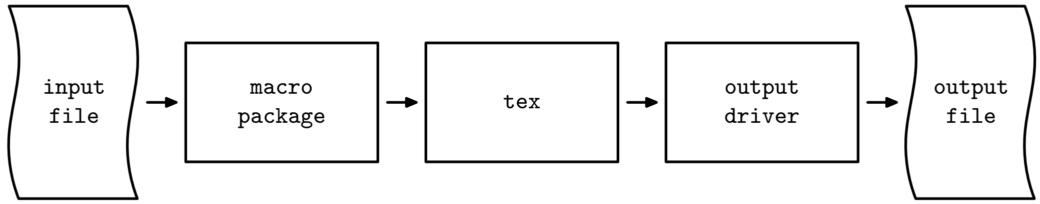

Figure 1: The basic information flow of TeX

(*ii*) **TeX knows typesetting** Expertise is coded into TeX.

Appendix B is an example. TeX's more even lines are a consequence of its more sophisticated algorithms for making paragraphs and for hyphenating.

Another way that this expertise gives better output comes in setting technical material. TeX moves the task, as much as possible, into the software. For instance, it automatically classifies each mathematical symbol as a variable, or a relation, etc., and sets them with appropriate amounts of surrounding space. It also sizes superscripts, and many other things. The result is that, because your document follows the conventions of professional typesetting, your readers will know exactly what you mean. You almost never have to fret with the formulas. They just come out right.

The quality of output that you get is the single best reason to use TeX.

Superior Engineering

Everyone has been frustrated with software that is slow, fat, buggy, or that undergoes frequent incompatible version changes. TeX will not give you those troubles; from a Computer Science standpoint, TeX is very impressive.

(*iii*) **TeX is fast** TeX was written by D Knuth, one of the world's leading experts in the design of algorithms. It ran quickly when it was developed in 1978 and so on today's machines it is very fast. It is easy on your computer's memory and disk space, too.

(*iv*) **TeX is stable** TeX is in wide use, with a long history. It has been tested by millions of users on demanding input. It will never eat your document. Never.

But there is more here than just that the program is reliable. TeX's designer has frozen the central engine, the actual `tex` program. Documents that run today will still run in ten years, or fifty. So 'stable' means more than that it actually works; it means that it will continue to work, forever.

(*v*) **Stable but not rigid** A system locked into 1978's technology would today have gaps. That's why TeX is extendable, so that innovations can be added on — layered over the underlying engine.

An example is the LaTeX macro package, which is the most popular way to use TeX today. It is a front end to the engine, affecting the way authors input their work. It adds conveniences such as automatic cross references, indexing, a table of contents, automatic numbering of chapters, sections, theorems, etc., in a variety of styles, and a straightforward but powerful way to make tables.

LaTeX also adds a philosophy of encouraging authors to structure their document by meaning rather than by appearance. For instance, the way that most word processor users make a section heading is by typing the title, highlighting it with the mouse, clicking in a menu to select boldface, clicking in another menu for the point size, and then adding the white space above and below. LaTeX authors type `\section{`*title*`}`. This has two advantages. First, since it is a computer language command, it makes the type style, size, and vertical spacing uniform throughout your document. (True, working with a computer language makes changing the default trickier. But on the other hand, if you have put in two dozen or more section headings by hand then chances are that you've erred in at least one.) The second advantage of LaTeX's approach is that it is self documenting. You can, for instance, automate producing a list of sections. (Some word processors can do logical structuring, although few authors use it.)

And, LaTeX itself is extendable. There are thousands of "style files," which do everything from adapting the basics to the needs of the American Math Society,[3] to making cross-references into hyper-references,[4] all the way to allowing you to

[3] `ftp://ftp.ams.org/pub/tex/doc/amsmath/short-math-guide.pdf`

[4] `http://www.ctan.org/tex-archive/macros/latex/contrib/supported/hyperref/doc/manual.pdf`

add epigraphs,[5] the short quotations that sometimes decorate the start or end of a chapter.

Just because LaTeX is the most popular macro package doesn't mean that it is the best one for you. Many others are available; see the the TeX Users Group's interest page.[6]

So TeX has been, and is being, developed and extended in many ways.

***(vi)* The input is plain text** TeX's source files are portable to any computing platform. They are also easy to produce automatically, for example when you want to write a report from material in a database. They are compact (all of the files for my 450-page textbook[7] and 125-page answer supplement fit easily on one floppy). And, they integrate with other tools such as search utilities.

Use of this type of input file stems from an overall mindset. TeX arose in the world of science and engineering where there is a tradition of cooperating closely with fellow workers. A binary input format, especially a proprietary one, is bad for cooperation: probably you have had to go through the trouble of upgrading because coworkers upgraded and you could no longer read their files. With TeX systems that experience is quite rare (the last time that there was a LaTeX release that lost some backward compatibility was in 1995).

There are even ways to run TeX directly from XML input, which many people think is the standard input format of the future. So, with the TeX formatting engine in the middle, the input front end may be adjusted to meet your needs, and changing times.

***(vii)* The output can be anything** As with inputting, TeX's outputting step is separate from its typesetting. The TeX engine's results can be converted to a printer language such as PostScript, or to a web language such as PDF or HTML, or, probably, to whatever will appear in the future. And, the typesetting (line breaks, etc.) will be the same no matter where your output appears.

See also the example section below.

Freedom

Most computer users have heard about Free and Open Sourced software and know that, as with the GNU programs, Linux, Apache, Perl, etc., this style of development can yield software that is first class. TeX, along with associated materials such as index makers or style files, falls into this category.

***(viii)* TeX is free** The source of the main `tex` engine is open. (In large part because of this it is thoroughly debugged. Knuth offers a reward for finding errors and no significant ones have been found in a very long time, despite many smart folks looking for them). All of the other main components are open, also.

***(ix)* TeX runs anywhere** Whatever meets your needs — Windows, Macintosh, a variety of Unix, or almost any other system — you can get TeX, either freely distributed or in a commercial implementation.[8]

So although the core of TeX was written some time ago, it fits well with today's trends.

Popularity

Using the same system as many other people has advantages. You can get answers to your questions. Your problems might well have already been solved. And, because of this large user base, your system is sure to be around for years.

***(x)* TeX is the standard** Most scientists, especially academic scientists, know TeX. As a result, many publishers of technical material are set up to work with it. In particular, TeX is the system preferred by the American Math Society.[9]

Because it is the standard, TeX's support by other technical software is the best. For example, there are editing modes to make input convenient, such as AUCTeX for Emacs. Another example is that most computer algebra systems, such as Maple and Mathematica, will give output in TeX. And no doubt technical software developed in the future will support TeX, also.

In addition, TeX is used by many people outside of the sciences, for all of the reasons given in this document. For instance, there is a way to produce beautiful critical edition texts.[10]

You wouldn't want to use a bad system simply because it is popular. TeX has earned its user base for sound reasons, some of them given above. Nonetheless, the existence of such a base is itself one reason to adopt a software package.

In summary, TeX was designed by one of the world's foremost computer scientists. That design makes it especially shine in areas where the system familiar to most computer users, word processors, falls short. Briefly, that is, it was designed well.

[5] `http://www.ctan.org/tex-archive/macros/latex/contrib/supported/epigraph`

[6] `http://www.tug.org/interest.html`

[7] `http://joshua.smcvt.edu/linalg.html`

[8] `http://www.tug.org/interest.html`

[9] `http://www.ams.org/tex`

[10] `http://www.ctan.org/tex-archive/macros/plain/contrib/edmac`

3 An Example

Anyone can see from the two appendices (and this document) that TEX's output quality is first-rate. However, some of the other points above might be less familiar. This section may help to make them more concrete. Someone asked on the discussion group `comp.text.tex`[11] whether TEX would be suitable for a large job where the text is generated from a database. Here is my reply.

```
> I'm contemplating using tex (or latex)
> to perform a mail merge. .. anywhere
> from 1,800 to 25,000 documents at a
> time .. What performance issues .. ?
I doubt TeX would slow you down. I just
wrote this Perl script
.......................................
#!/usr/bin/perl
$file_text="\\documentclass{letter}
\\begin{document}
\\begin{letter}{Addressee \\\\ Address}
\\opening{Dear Sir} Hi. \\closing{Thanks}
\\end{letter}
\\end{document}";
for ($i=0; $i<100; $i++) {
    $fn="test".$i;
    open(OUTFILE,">$fn.tex");
    print OUTFILE $file_text;
    close OUTFILE;
    system("latex $fn");
    system("dvips -Pwww -o$fn.ps $fn");
    system("rm $fn.aux");
    system("rm $fn.log");
}
.......................................
which writes 100 LaTeX letters, then
LaTeX's them, then converts to PostScript
for printing (and deletes a couple of
log files).  On my laptop (P3; I don't
know the MHz) execution took 22 secs.
So I think your bottleneck is more likely
to be your printing device.
```

This illustrates many of the points above. First, TEX's input is plain text, so I was able to generate it easily in a program. Second, TEX is fast, so I got the one hundred letters in no time. Third, I didn't have to ask the person what platform they were using because TEX runs anywhere. For that matter — fourth — I didn't have to ask what software vendor they had licenses with, because TEX is free.

That is, it illustrates that TEX is a practical professional tool. TEX helps solve problems.

[11] `http://groups.google.com/groups?q=comp.text.tex`

4 When Not to Use TEX

Despite my enthusiasm for TEX, my children write their school reports with a word processor. That's because TEX has a steeper learning curve and for their material the word processor is just fine.

A word processor suits your needs if your documents are brief, structurally simple, and entered by hand. If you will only ever write straightforward text, in short to medium-sized documents, and where good-enough typography is good enough, then stick with a word processor.

The opposite extreme, a document such as a brochure or an advertisement that is dominated by graphics, font changes, and color, is best tackled in a layout tool like Quark or Framemaker.

5 For More Information

The TEX Users Group[12] has much more information and many links, including more of an introduction[13] and a list of available distributions.[14] A good way to get started if you already have TEX installed is *The (Not So) Short Introduction to LATEX 2ε*.[15]

A A Sample of Mathematics

The first of the two following pages is an excerpt from *Theory of Recursive Functions and Effective Computability* by Rogers.

B A Sample of Plain Text

The other page has the first two paragraphs of *Zen in the Art of Archery* by Herrigel, on the left done in Microsoft's *Word* and on the right done in LATEX.

For each, I just selected the 12 point Times Roman font that came with my system and in other areas used the defaults, except that I made the line width be 3.5 inches. This is the layout of the edition of the book that sits on my shelf, and also lets you compare outputs here side by side.

Jim Hefferon
St Michael's College
2002-Mar-04

[12] `http://www.tug.org`
[13] `http://www.tug.org/whatis.html`
[14] `http://www.tug.org/interest.html`
[15] `http://www.ctan.org/tex-archive/info/lshort`

Recursive definitions are familiar in mathematics. For instance, the function f defined by

$$\begin{aligned} f(0) &= 1, \\ f(1) &= 1, \\ f(x+2) &= f(x+1) + f(x), \end{aligned}$$

gives the Fibonacci sequence: 1, 1, 2, 3, 5, 8, 13, ... (The study of *difference equations* concerns the problem of going from recursive definitions to algebraic definitions. The Fibonacci sequence is give by the algebraic definition

$$f(x) = \frac{\sqrt{5}}{5}(\frac{1+\sqrt{5}}{2})^{x+1} - \frac{\sqrt{5}}{5}(\frac{1-\sqrt{5}}{2})^{x+1}.)$$

The *primitive recursive functions* are an example of a broad and interesting class of functions that cam be obtained by such a formal characterization.

Definition The class of *primitive recursive functions* is the smallest class $\mathcal{C}$ (i.e., intersection of all classes $\mathcal{C}$) of functions such that

i. All *constant functions*, $\lambda x_1 x_2 \cdots x_k[m]$ are in $\mathcal{C}$, $1 \le k$, $0 \le m$;

ii. The *successor function*, $\lambda x[x+1]$, is in $\mathcal{C}$;

iii. All *identity functions*, $\lambda x_1 \cdots x_k[x_i]$ are in $\mathcal{C}$, $1 \le i \le k$;

iv. If f is a function of k variables in $\mathcal{C}$, and g_1, g_2, ... , g_k are (each) functions of m variables in $\mathcal{C}$, then the function $\lambda x_1 \cdots x_m[f(g_1(x_1, \ldots, x_m), \ldots, g_k(x_1, \ldots, x_m))]$ is in $\mathcal{C}$, $1 \le k, m$;

v. If h is a function of $k+1$ variables in $\mathcal{C}$, and g is a function of $k-1$ variables in $\mathcal{C}$, then the unique function f of k variables satisfying

$$\begin{aligned} &f(0, x_2, \ldots, x_k) = g(x_2, \ldots, x_k), \\ &f(y+1, x_2, \ldots, x_k) = h(y, f(y, x_2, \ldots, x_k), x_2, \ldots, x_k) \end{aligned}$$

is in $\mathcal{C}$, $1 \le k$. (For (v), "function of zero variables in $\mathcal{C}$" is taken to mean a fixed integer.)

At first sight it must seem intolerably degrading for Zen — however the reader may understand this word — to be associated with anything so mundane as archery. Even if he were willing to make a big concession, and to find archery distinguished as an "art," he would scarcely feel inclined to look behind this art for anything more than a decidedly sporting form of prowess. He therefore expects to be told something about the amazing feats of Japanese trick-artists who have the advantage of being able to rely on a time-honored and unbroken tradition in the use of bow and arrow. For in the Far East it is only a few generations since the old means of combat were replaced by modern weapons, and familiarity in the handling of them by no means fell into disuse, but went on propagating itself, and has since been cultivated in ever widening circles. Might one not expect, therefore, a description of the special ways in which archery is pursued today as a national sport in Japan?

Nothing could be more mistaken than this expectation. By archery in the traditional sense, which he esteems as an art and honors as a national heritage, the Japanese does not understand a sport but, strange as this may sound at first, a religious ritual. And consequently, by the "art" of archery he does not mean the ability of the sportsman, which can be controlled, more or less, by bodily exercises, but an ability whose origin is to be sought in spiritual exercises and whose aim consists in hitting a spiritual goal, so that fundamentally the marksman aims at himself and may even succeed in hitting himself.

At first sight it must seem intolerably degrading for Zen — however the reader may understand this word — to be associated with anything so mundane as archery. Even if he were willing to make the big concession, and to find archery distinguished as an "art," he would scarcely feel inclined to look behind this art for anything more than a decidedly sporting form of prowess. He therefore expects to be told something about the amazing feats of Japanese trick-artists who have the advantage of being able to rely on a time-honored and unbroken tradition in the use of bow and arrow. For in the Far East it is only a few generations since the old means of combat were replaced by modern methods, and familiarity in the handling of them by no means fell into disuse, but went on propagating itself, and has since been cultivated in ever widening circles. Might one not expect, therefore, a description of the special ways in which archery is pursued today as a national sport in Japan?

Nothing could be more mistaken than this expectation. By archery in the traditional sense, which he esteems as an art and honors as a national heritage, the Japanese does not understand a sport but, strange as this may sound at first, a religious ritual. And consequently, by the "art" of archery he does not mean the ability of the sportsman, which can be controlled, more or less, by bodily exercises, but an ability whose origin is to be sought in spiritual exercises and whose aim consists in hitting a spiritual goal, so that fundamentally the marksman aims at himself and may even succeed in hitting himself.