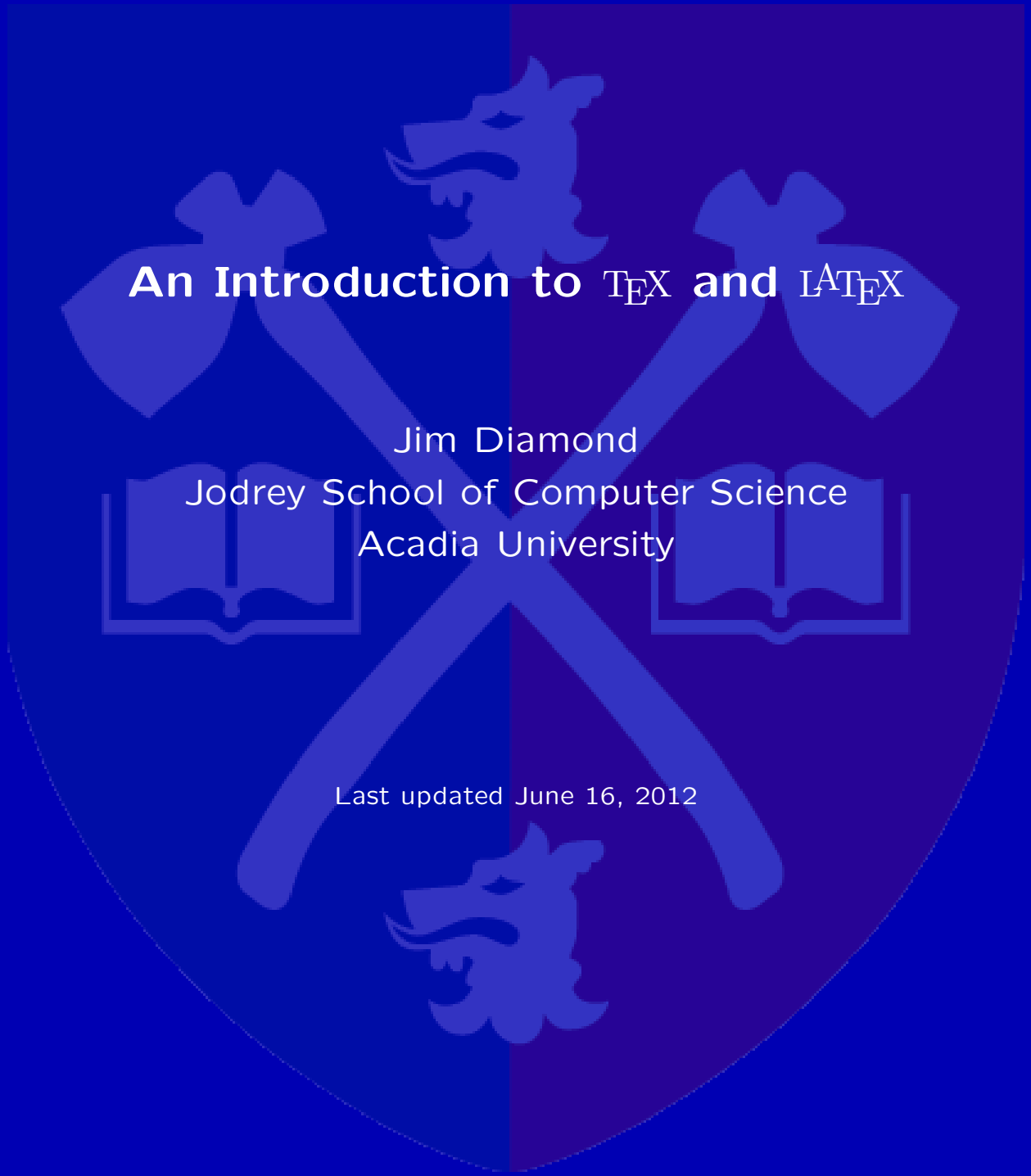
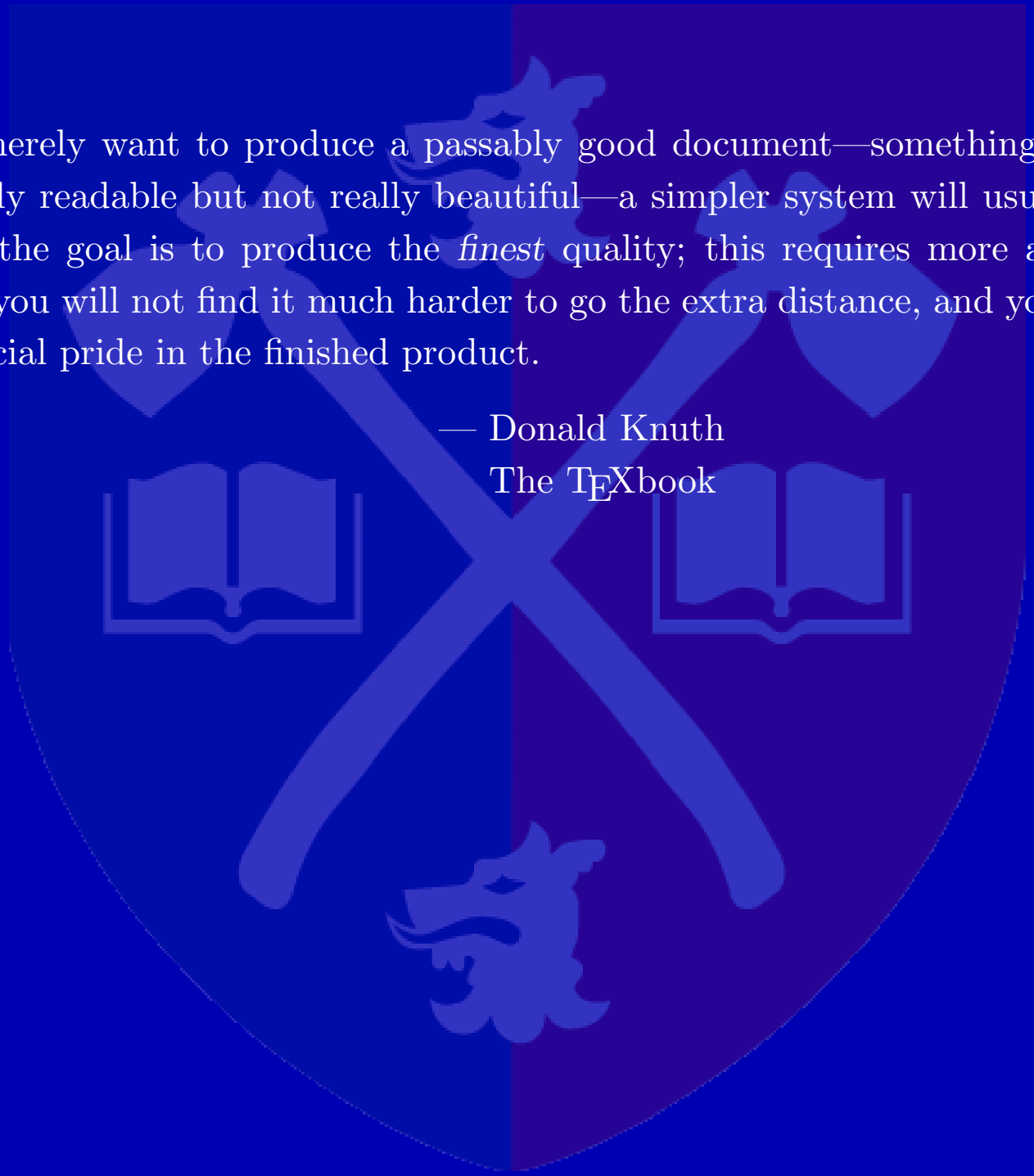


An Introduction to TEX and LATEX

Jim Diamond
Jodrey School of Computer Science
Acadia University

Last updated June 16, 2012



. . . If you merely want to produce a passably good document—something acceptable and basically readable but not really beautiful—a simpler system will usually suffice. With TEX the goal is to produce the *finest* quality; this requires more attention to detail, but you will not find it much harder to go the extra distance, and you'll be able to take special pride in the finished product.

— Donald Knuth
The TEXbook

. . . If you merely want to produce a passably good document—something acceptable and basically readable but not really beautiful—a simpler system will usually suffice. With TEX the goal is to produce the *finest* quality; this requires more attention to detail, but you will not find it much harder to go the extra distance, and you'll be able to take special pride in the finished product.

— Donald Knuth
The TEXbook

Good typesetting is good for everyone since it's there to make the job of reading easier, and there's no excuse for the low-grade output you get from MS Word.

. . .

Good typesetting is a courtesy to the reader and it makes the work easier to read. Good typesetting is no signal of anything since it is un-noticeable by the reader — if done properly.

Bad typesetting, on the other hand, does send out signals. It says you have contempt for the reader and you're willing to do low-grade work with no care for quality.

— Rowland McDonnell
comp.text.tex, January 27, 2007

Preface

This presentation is a very brief introduction to the TEX family of document preparation tools.

It is by no means complete; in fact, it is a work in progress. If you have any suggestions for additions or improvements to the content, please let Jim Diamond know.

In the examples in this presentation, yellow often signifies text that would appear in your TEX "source" document. In most cases this is followed by the corresponding typeset version, shown in white. Yellow is also used to denote literal uses of filenames and program names.

URLs are given in brightred. These might be links to web sites, "mailto"s, or even commands for Adobe Acrobat Reader to run.

This presentation was itself made up using TEX. I didn't use any fancy visual effects here, but if those sorts of things appeal to you, here is a presentation created with TEX-based tools which might perk your interest. A more modern approach uses LATEX's beamer package.

Outline

- History
- Mark-up Languages vs. WYSIWYG
- Overview of TEX
- Sample of text input/output
- Sample of math input/output
- Extensions of TEX
- TEX and Graphics
- Getting TEX
- Sample LATEX Thesis Documents

History

- Donald Knuth of Stanford U was writing his *Art of Computer Programming* series
  - it was intended to be a seven book series
- Knuth got bogged down in the typesetting process
  - there is a lot of math in his books
  - there was no suitable typesetting system to be found
- So he wrote his own — ca. 1982
- He made the system available to everyone (for free!)
  - available for Unix, Linux, VMS, M$, Apple, ...
- Knuth's original system ("plain" TEX) had only "low-level" facilities
  - he did not define any high-level constructs to directly support structured documents
- Leslie Lamport created an extension (??) called LATEX in the mid 1980's
  - his system had high-level constructs for structured documents
  - easier to use than "plain" TEX ...sort of

Mark-up Languages vs. WYSIWYG

- Mark-up languages allow you to
  - use any plain text editor you want (namely, emacs :-)
  - use any other plain-text tools (grep, aspell, less, ...)
  - concentrate on content:
    - users of defined formats don't need to worry about details of font sizes, specific type faces, etc.
    - well-designed mark-up languages allow instant change of format
- WYSIWYG systems
  - typically use a binary file format (normally proprietary)
    - proprietary file formats suffer from obsolescence
    - you are stuck with one "editor" — generally very rudimentary
  - encourage people to think they are typographers
  - confuse content preparation with typographic design
  - gives rise to the "macdinking" phenomenon

:macdink: /mak'dink/ vt. [from the Apple Macintosh, which is said to encourage such behavior] To make many incremental and unnecessary cosmetic changes to a program or file. Often the subject of the macdinking would be better off without them. "When I left at 11 P.M. last night, he was still macdinking the slides for his presentation." See also {fritterware}, {window shopping}.

— The Jargon File, Version 4.3.0, 30 Apr 2001

Overview of TeX (LaTeX is very similar)

- Create your TeX source file — plain ASCII text
  - composed of the desired text + mark-up in TeX's macro language
- TeX (the program) "compiles" the source file into a "DVI" (DeVice Independent) file

- A post-processor programs convert the DVI file into a format suitable for your output device
  - *e.g.*, X window screen (`xdvi`), generic PostScript (`dvips`), PDF (`dvipdfm`), ...

- Within the limits of device resolution, the output of the DVI converter is the same on **all** devices (this is a Very Good Thing)

Some Mark-up for Plain Text — Or Maybe Not

```
If we use no mark-up at all, TeX just fills and justifies the lines.
A blank line in the input tells TeX to start a new paragraph.

In this paper we examine the cyclic structure of graphs with
edges labelled by elements of a partial order under the operation
of deleting any edge whose label is less than or equal
to all labels of edges of some cycle containing that edge.

We show that all graphs obtained after repeating the above
operation as many times as possible have similar structure
with respect to the number of edges remaining and thus, in
particular, the presence or absence of cycles.  So there!
```

If we use no markup at all, TeX just fills and justifies the lines. A blank line in the input tells TeX to start a new paragraph.

In this paper we examine the cyclic structure of graphs with edges labelled by elements of a partial order under the operation of deleting any edge whose label is less than or equal to all labels of edges of some cycle containing that edge.

We show that all graphs obtained after repeating the above operation as many times as possible have similar structure with respect to the number of edges remaining and thus, in particular, the presence or absence of cycles. So there!

Some Generic Mark-up

```
\Section Writing Sections of A Paper

Notice that when we use the Section ``command'' we skip some space,
we automagically compute the number of the next section, and then we
output the number and title of the section in a suitable font.

\Subsection Subsections of A Paper

This skips a smaller amount of space, computes the next
subsection number, and outputs it in a different font.
```

5.2. Writing Sections of A Paper

Notice that when we use the Section "command" we skip some space, we automagically compute the number of the next section, and then we output the number and title of the section in a suitable font.

5.2.1 Subsections of A Paper

This skips a smaller amount of space, computes the next subsection number, and outputs it in a different font.

TEX Mark-up

- Most "commands" start with "\"
  - not <blah> ... </blah> like HTML or XML
- Commands are either TEX primitives or "macros"
  - TEX allows you to extend the mark-up language by defining your own macros
- Macros look a bit like functions, but are a bit trickier to create
  - examine this example **only** if you enjoy programming and plan to use (plain) TEX!

```
\def\Section #1 \par
{
    \bigbreak\vskip\parskip
    \need 1.5 in
    \global\advance\secno by 1    % Not before the \need !!!
    \subno=0\subsubno=0           % Reset these counters!
    \noindent
    {\SectionTitleFont \the\secno.\hskip 1em #1}
    \par
}
```

Typesetting Math: Some Examples

- TEX examples (LATEX allows this syntax, but also has alternative forms)

```
Thus $\omega(x) < \log(x)\ \forall x > 0$, as required.
```

Thus $\omega(x) < \log(x)\ \forall x > 0$, as required.

```
$${49 \choose 6} \approx 1.398 \times 10^7$$
```

$$\binom{49}{6} \approx 1.398 \times 10^7$$

```
$$\int_0^\infty {\sin(x)\over x} \, dx \eqno(15)$$
```

$$\int_0^\infty \frac{\sin(x)}{x}\,dx \tag{15}$$

```
$$2 + \left( \int_0^\infty {\sin(x) \over x} \, dx \right)
    < \lim_{n \rightarrow \infty} \sum_{i=1}^\infty {1\over i}$$
```

$$2 + \left(\int_0^\infty \frac{\sin(x)}{x}\,dx \right) < \lim_{n \rightarrow \infty} \sum_{i=1}^{n} \frac{1}{i}$$

Extensions of TEX

- LATEX – an extension of TEX to facilitate document preparation
  - organized, well-thought out and cohesive (??)
- ConTEXt – another set of macros for TEX
  - a more modern design than LATEX, but not yet as well known
- eplain – some extensions to plain TEX to make writing your own macro package easier
- pdftex – a program which takes TEX files and directly creates Adobe Acrobat (.pdf) files
  - 100% ($\pm\epsilon$) compatible with TEX, LATEX, and ConTEXt
  - allows you to use bookmark (and other) features of Acrobat documents
    - e.g., URLs: http://cs.acadiau.ca
    - e.g., actions: show a movie
- tth – a program which converts (some) TEX files into HTML
  - not completely robust and all-encompassing, but a good way to convert math formulae into HTML
- All these are **free**

Which TeX Should I Use?

- "I like to do it myself"
  - then you should use either
    - plain TeX (Knuth's original version), or
    - eplain (plain TeX augmented with some facilities to perform common typesetting tasks)
  - warning: the initial learning curve will be steep!
  - this choice will probably **not** appeal to people who don't enjoy programming
- "I need instant gratification"
  - then you should use either
    - LaTeX (the "popular" choice), or
    - ConTeXt (a newer and maybe better system)
  - there are many, many LaTeX packages out there to assist you
    - finding the one you want might be time consuming
  - you will be able to get moving quickly (for thesis students: especially with the sample LaTeX theses noted later in this presentation)

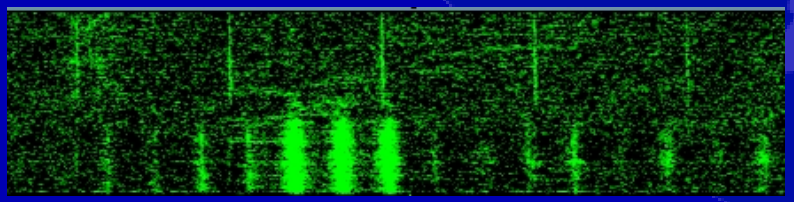
TEX and Graphics

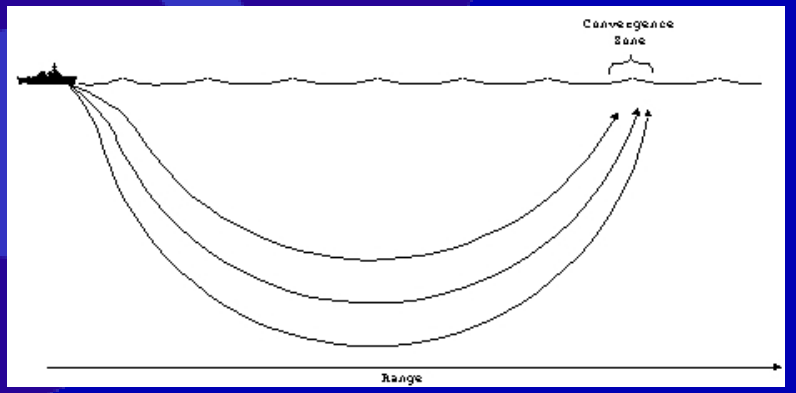
- Ability to do graphics in TEX was deliberately left out (!)
- Instead, "\special"s were put in to allow commands to be handled by DVI processors (or directly by pdftex)
- These \specials can instruct the DVI processor (or pdftex) to
  - insert graphics
  - control output devices (tray selection, two-sided printing, ...)
  - various and sundry other things...
- TEX just has to know how much space to reserve for graphics
  - macro packages are available to read encapsulated PostScript (.eps) and other graphics files to see how big the graphic's "bounding box" is
- pdftex can directly include JPEG, PNG and PDF files
- Rudimentary "picture" facilities exist entirely within TEX; the boxes and lines used earlier are pure TEX

TEX and Graphics: Examples

```
\centerline{\includegraphics[width=2.4truein]{Ceramics-300-30}}
```

```
\centerline{\includegraphics[width=0.45\hsize]{lofar}
    \qquad
    \includegraphics[width=0.45\hsize]{conv-zone}}
```



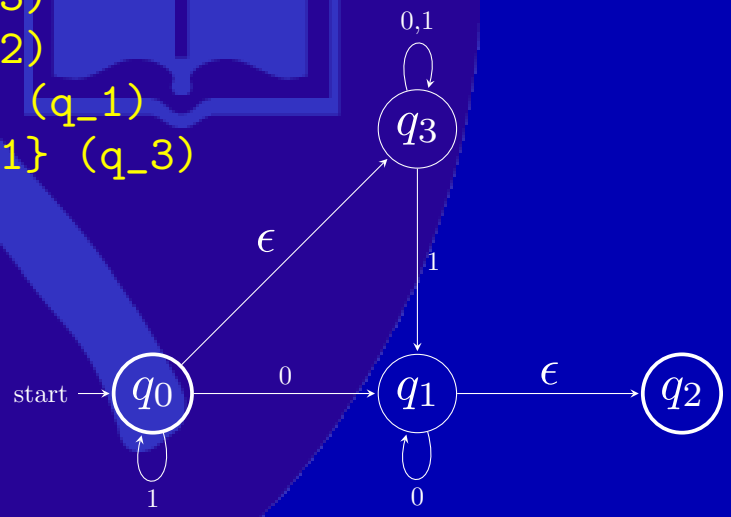


TEX and Graphics: PSTricks and TikZ/PGF

- The previous examples show how graphics files created by other programs could be inserted into a TEX document
  - these can be bitmap images (png, jpeg, ...) or vector graphics (eps)
- There are a number of packages that let you create diagrams by (textually) describing what you want to do **within** your TEX (or LATEX) document
- PSTricks ("PostScript tricks")
  - this system works with `tex` (and `latex`), but not `pdftex` or `pdflatex`
  - you can do amazingly tricky things, but there is a learning curve involved
  - see the manual `here`; there are many examples to get you going
- More recently, a package called TikZ/PGF became available which also allows you to embed the graphics commands inside your TEX or LATEX document

TEX and Graphics: TikZ/PGF Example

```
\tikzpicture [shorten >=1pt,>=stealth,node distance=3.5cm,auto]
    \node[state,initial,accepting]     (q_0) {$q_0$};
    \node[state]                       (q_1) [right of=q_0] {$q_1$};
    \node[state,accepting]             (q_2) [right of=q_1] {$q_2$};
    \node[state]                       (q_3) [above of=q_1] {$q_3$};
    \path[->]
        (q_0) edge node {0} (q_1)
              edge [loop below] node {1} (q_0)
              edge node {$\epsilon$} (q_3)
        (q_1) edge node {$\epsilon$} (q_2)
              edge [loop below] node {0} (q_1)
        (q_3) edge [loop above] node {0,1} (q_3)
              edge node {1} (q_1);
\endtikzpicture
```



TEX and Graphics: Another TikZ/PGF Example

```
\tikzpicture
\tikzstyle{level 1}=[sibling angle=120]
\tikzstyle{level 2}=[sibling angle=60]
\tikzstyle{level 3}=[sibling angle=30]
\tikzstyle{every node}=[fill]
\tikzstyle{edge from parent}=[snake=expanding waves,
          segment length=1.8mm,segment angle=10,draw]

\tikz [grow cyclic,shape=circle,very thick,
       level distance=20mm,cap=round]
  \node {} child [color=\A] foreach \A in {red,green,blue}
    { node {} child [color=\A!50!\B] foreach \B in {red,green,blue}
      { node {} child [color=\A!50!\B!50!\C] foreach \C in
                                    {black,gray,white}
        { node {} }
      }
    };
\endtikzpicture
```

Getting Help

- The newsgroup comp.text.tex has many knowledgeable, civil people
- There are various books available:
  - "The TEXbook" (Donald Knuth)
  - "TEX by Topic" (Victor Eijkhout)
    - available for free(!) at http://www.eijkhout.net/tbt/
  - "A Gentle Introduction to TEX" (Michael Doob) also available for free; get yourself a copy at http://ctan.tug.org/tex-archive/info/gentle/gentle.pdf
  - "LATEX 2_ε" (Leslie Lamport et al.)
    - this should be packaged with your LATEX software; otherwise search the web for latex2e.pdf
  - the official ConTEXt documentation (or so it claims) is available at wiki.contextgarden.net/Official_ConTeXt_Documentation
- There are many packages and corresponding documentation available at http://ctan.tug.org/
- There are a number of tutorials listed in http://www.tex.ac.uk/cgi-bin/texfaq2html?label=tutorials*

More Help

- A web forum for LaTeX: http://www.latex-community.org/
- A four-page intro to LaTeX (sized for A4 paper): http://latex.silmaril.ie/veryshortguide/veryshortguide.pdf
- A two-page LaTeX cheat sheet: http://www.stdout.org/~winston/latex/latexsheet.pdf
  - see also the source
- The comprehensive list of symbols for LaTeX can be found here
- Yet more LaTeX information, including LaTeX *for Complete Novices* and *Using* LaTeX *to Write a PhD Thesis* http://theoval.cmp.uea.ac.uk/~nlct/latex/
- Three LaTeX tutorials and a tutorial for typesetting mathematics were here the last time I looked.

Getting TeX (and LaTeX and ConTeXt and eplain)

- Source is available for free, but...
- Much (much) easier to get pre-compiled and configured distribution
- Most Linux distributions have the TeX family available
  - you may need to configure your installation for 8.5×11 paper: use the `texconfig` program
- M$: go to http://www.miktex.org and download the latest and greatest version (on September 3, 2011 that was version 2.9)
- Or (for any OS) download and install the latest and greatest "texlive" (on September 3, 2011 that was `texlive2011-live-20110903`)
  - installation is very straightforward, but you can run it directly from a DVD if you don't want to install it on your computer

Sample LaTeX M.Sc. Thesis Document (created by Brian Demmings, M.Sc.)

- Here is a "skeleton" Acadia M.Sc. thesis document to give you a running start, should you wish to use LaTeX for your thesis
  - note that this skeleton is made up of a number of files; you might find it convenient to separate (for example) different chapters into different source files
- This skeleton automagically creates the first few pages, the table of contents, lists of figures and tables, and so on
  - it also gives you the format for creating chapters, sections, subsections, and so on
  - it shows how you can cite bibliographic references and create the bibliography for your thesis in a convenient way
  - you can compare the sample thesis PDF file to the source files to see how things are done
- The sample chapters are each mini-tutorials for typesetting and/or LaTeX, and are useful reading for anyone getting started with LaTeX

Sample LaTeX B.Sc. Honours Thesis Document

- Here is a "skeleton" Acadia B.Sc.H. thesis document to give you a running start, should you wish to use LaTeX for your thesis
  - thanks to Brian Demmings for doing the initial work on this, before I took over the maintenance
  - thanks also to Dr. Jeff Hooper and Alex Sanford for assistance in various ways
  - this template is very similar to the master's thesis example
  - it differs vis-a-vis the formatting requirements and the frontmatter, but is otherwise quite similar

Summary

- Initially, you will have to do a bit more work to create documents with TEX than with a word processor
- In the long run, the automated facilities which help you produce large documents will pay off, saving you considerable time

- If you have any suggestions to improve this document, or any of the sample documents referred to here, please e-mail Jim Diamond.