

# An Introduction to $\text{T}_{\text{E}}\text{X}$ and $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

Jim Diamond  
Jodrey School of Computer Science  
Acadia University

Last updated September 4, 2011

...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 T<sub>E</sub>X 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 T<sub>E</sub>Xbook

...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 T<sub>E</sub>X 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 T<sub>E</sub>Xbook

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 T<sub>E</sub>X 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 T<sub>E</sub>X “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 [red](#). 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 T<sub>E</sub>X. I didn’t use any fancy visual effects here, but if those sorts of things appeal to you, [here](#) is a presentation created with T<sub>E</sub>X-based tools which might perk your interest.

# Outline

- History
- Mark-up Languages vs. WYSIWYG
- Overview of T<sub>E</sub>X
- Sample of text input/output
- Sample of math input/output
- Extensions of T<sub>E</sub>X
- T<sub>E</sub>X and Graphics
- Getting T<sub>E</sub>X
- Sample L<sup>A</sup>T<sub>E</sub>X 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"  $\text{T}_{\text{E}}\text{X}$ ) had only "low-level" facilities
  - he did not define any high-level constructs to directly support structured documents
- Leslie Lamport created an extension (??) called  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  in the mid 1980's
  - his system had high-level constructs for structured documents
  - easier to use than "plain"  $\text{T}_{\text{E}}\text{X}$  ...sort of

## Mark-up Languages vs. WYSIWYG

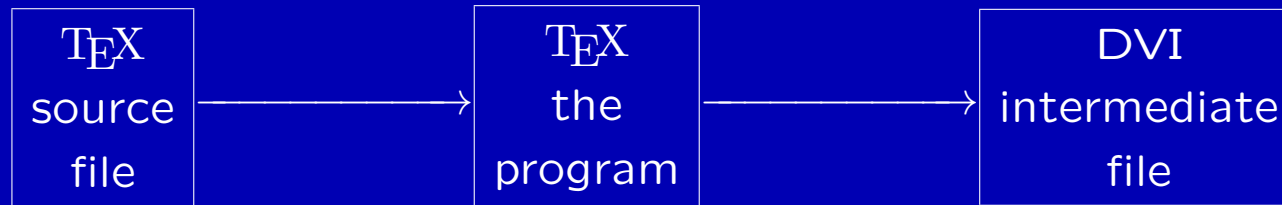
- Mark-up languages allow you to
  - use any plain text editor you want (namely, emacs :-)
  - use any other plain-text tools (grep, aspell, less, ...)
  - concentrate on content:
    - users of defined formats don't need to worry about details of font sizes, specific type faces, etc.
    - well-designed 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 $\text{T}_{\text{E}}\text{X}$ ( $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ is very similar)

- Create your  $\text{T}_{\text{E}}\text{X}$  source file — plain ASCII text
  - composed of the desired text + mark-up in  $\text{T}_{\text{E}}\text{X}$ 's macro language
- $\text{T}_{\text{E}}\text{X}$  (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

- $\text{\TeX}$  examples ( $\text{\LaTeX}$  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.

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

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

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

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

$2 + \left( \int_0^{\infty} \frac{\sin(x)}{x} dx \right)$

$< \lim_{n \rightarrow \infty} \sum_{i=1}^n \frac{1}{i}$

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

## Extensions of T<sub>E</sub>X

- L<sup>A</sup>T<sub>E</sub>X – an extension of T<sub>E</sub>X to facilitate document preparation
  - organized, well-thought out and cohesive (??)
- ConT<sub>E</sub>Xt – another set of macros for T<sub>E</sub>X
  - a more modern design than L<sup>A</sup>T<sub>E</sub>X, but not yet as well known
- eplain – some extensions to plain T<sub>E</sub>X to make writing your own macro package easier
- **pdf<sub>t</sub>ex** – a program which takes T<sub>E</sub>X files and directly creates Adobe Acrobat (**.pdf**) files
  - 100% ( $\pm \epsilon$ ) compatible with T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X, and ConT<sub>E</sub>Xt
  - 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) T<sub>E</sub>X files into HTML
  - not completely robust and all-encompassing, but a good way to convert math formulae into HTML
- All these are **free**

## Which T<sub>E</sub>X Should I Use?

- “I like to do it myself”
  - then you should use either
    - plain T<sub>E</sub>X (Knuth’s original version), or
    - eplain (plain T<sub>E</sub>X 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
    - L<sup>A</sup>T<sub>E</sub>X (the “popular” choice), or
    - ConT<sub>E</sub>Xt (a newer and maybe better system)
  - there are many, many L<sup>A</sup>T<sub>E</sub>X 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 L<sup>A</sup>T<sub>E</sub>X theses noted later in this presentation)

## T<sub>E</sub>X and Graphics

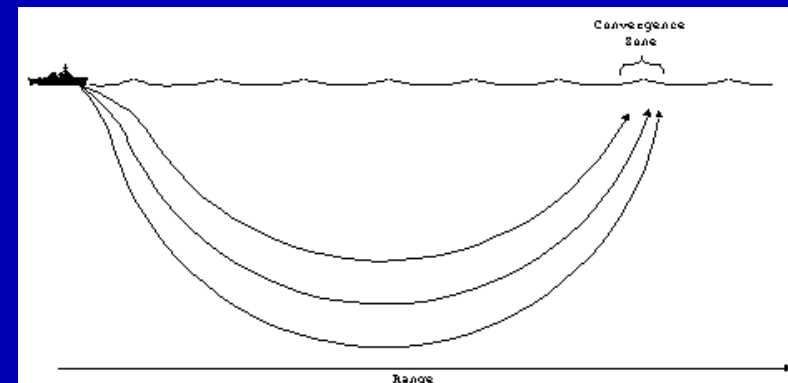
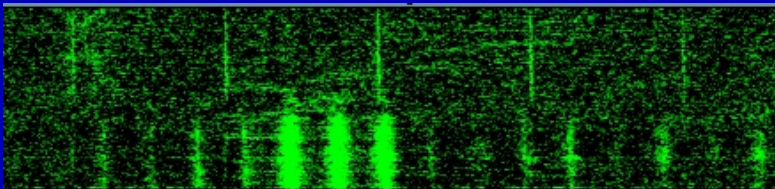
- Ability to do graphics in T<sub>E</sub>X was deliberately left out (!)
- Instead, “\special”s were put in to allow commands to be handled by DVI processors (or directly by pdftex)
- These \specials can instruct the DVI processor (or pdftex) to
  - insert graphics
  - control output devices (tray selection, two-sided printing, ...)
  - various and sundry other things...
- T<sub>E</sub>X 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 T<sub>E</sub>X; the boxes and lines used earlier are pure T<sub>E</sub>X

# TEX and Graphics: Examples

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



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



## $\text{T}_{\text{E}}\text{X}$ and Graphics: PSTricks and TikZ/PGF

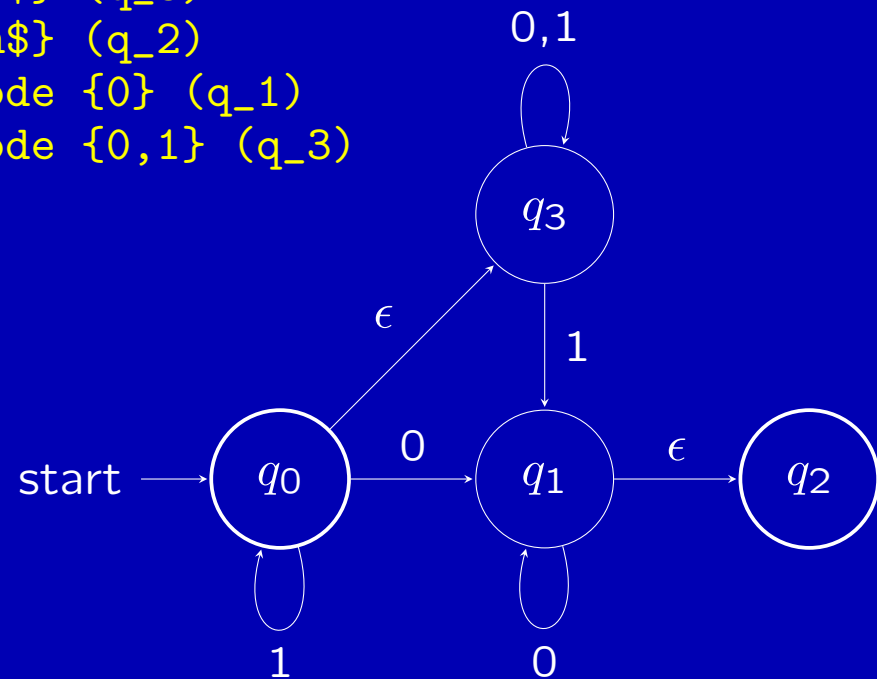
- The previous examples show how graphics files created by other programs could be inserted into a  $\text{T}_{\text{E}}\text{X}$  document
  - these can be bitmap images (png, jpeg, ...) or vector graphics (eps)
- There are a number of packages that let you create diagrams by (textually) describing what you want to do **within** your  $\text{T}_{\text{E}}\text{X}$  (or  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ) 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  $\text{T}_{\text{E}}\text{X}$  or  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  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)
           edge node {$\epsilon$} (q_2);
\endtikzpicture

```



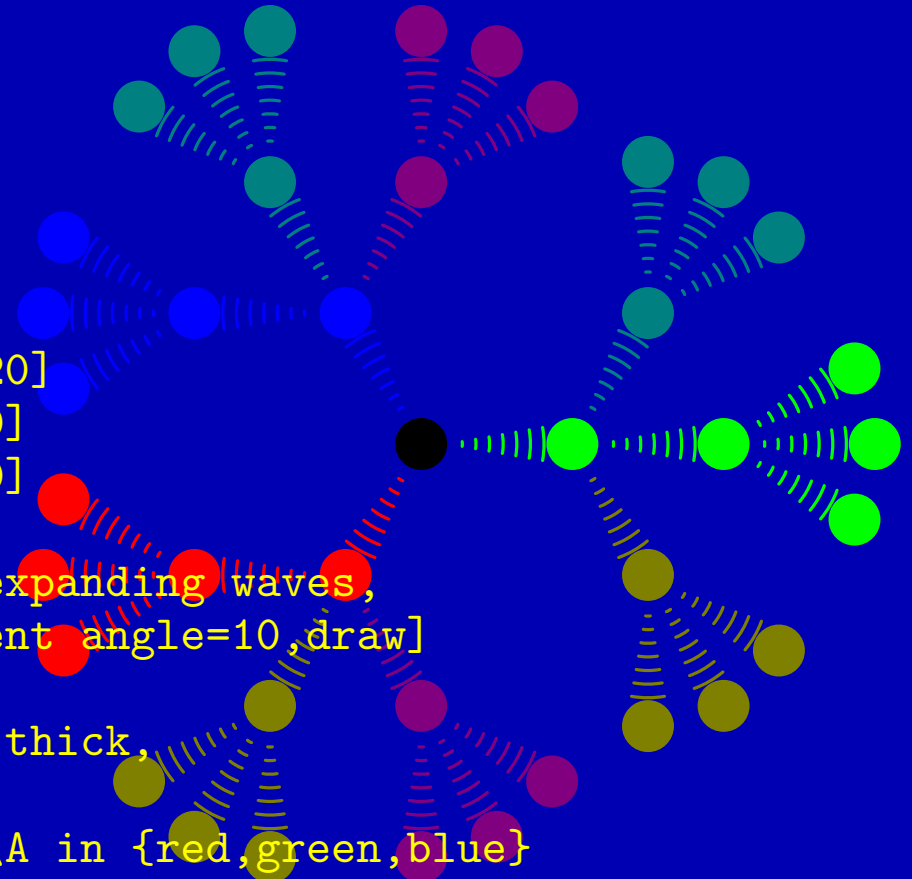
## TeX and Graphics: Another TikZ/PGF Example

```

\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 T<sub>E</sub>Xbook” (Donald Knuth)
  - “T<sub>E</sub>X by Topic” (Victor Eijkhout)
    - available for free(!) at <http://www.eijkhout.net/tbt/>
  - “A Gentle Introduction to T<sub>E</sub>X” (Michael Doob) also available for free; get yourself a copy at <http://ctan.tug.org/tex-archive/info/gentle/gentle.pdf>
  - “L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>” (Leslie Lamport et al.)
    - this should be packaged with your L<sup>A</sup>T<sub>E</sub>X software; otherwise search the web for [latex2e.pdf](#)
  - the official ConT<sub>E</sub>Xt documentation (or so it claims) is available at [wiki.contextgarden.net/Official\\_ConTeXt\\_Documentation](http://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\\*](http://www.tex.ac.uk/cgi-bin/texfaq2html?label=tutorials*)

## More Help

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

## Getting T<sub>E</sub>X (and L<sup>A</sup>T<sub>E</sub>X and ConT<sub>E</sub>Xt and eplain)

- Source is available for free, but...
- Much (much) easier to get pre-compiled and configured distribution
- Most Linux distributions have the T<sub>E</sub>X 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 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 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  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  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 automatically 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  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ , and are useful reading for anyone getting started with  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

## Sample L<sup>A</sup>T<sub>E</sub>X 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 L<sup>A</sup>T<sub>E</sub>X for your thesis
  - thanks to Dr. Jeff Hooper for putting this together
- This document is considerably less detailed (as a L<sup>A</sup>T<sub>E</sub>X tutorial) than the M.Sc. thesis referred to on the previous page
  - you may find looking at the M.Sc. sample useful even if you are an honours student
  - **but you should** use the honours thesis skeleton for an honours thesis, since there are very specific requirements for honours theses (and virtually no requirements for master’s theses!)

## Summary

- Initially, you will have to do a bit more work to create documents with  $\text{\TeX}$  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](#).