

# L<sup>A</sup>T<sub>E</sub>X DOCUMENT PREPARATION SYSTEM (L<sup>A</sup>T<sub>E</sub>X) FOR AGRICULTURAL AND BIOLOGICAL ENGINEERS

C. Igathinathane

**ABSTRACT.** *LaTeX (L<sup>A</sup>T<sub>E</sub>X) is a document preparation typesetting system used by scientists and engineers worldwide for producing professional quality technical articles, books, and reports. It is the best system known for producing documents with mathematical equations and for very large books. Even though E<sub>T</sub><sub>X</sub> system is common in disciplines of mathematics, physics, and several engineering, it has not infiltrated into our Agricultural and Biological Engineering (ABE) specialization and possibly among professionals of American Society of Agricultural and Biological Engineers (ASABE), albeit a high-quality free software. E<sub>T</sub><sub>X</sub> system uses markups codes and text in ASCII, similar to HTML, when compiled produces consistent documents in several output formats. Being structural in coding, E<sub>T</sub><sub>X</sub> automates several common tasks of document preparation, such as table of contents, list of tables, list of figures, linked citations, building references, numbering equations, figures, and tables, index, nomenclature, glossary, multiple style formatting, updating elements in revision, etc. This educational paper introduces the E<sub>T</sub><sub>X</sub> system; its structure, obtaining the system and running the codes; utilizing article (Elsevier and ASABE) and thesis templates; and handling figure and table elements through examples. The merits and issues of the system were also discussed. Although there is a learning curve involved, the ABE students and ASABE professionals having the basic computer programming knowledge could easily benefit from E<sub>T</sub><sub>X</sub> system readily.*

**Keywords.** ASABE, Article, Document, Education, LaTeX, Thesis, Typesetting.

**L**aTeX is pronounced “Lay-tech” or “Lah-tech” and is an extensive document production typesetting system and a markup programming language. The logo of LaTeX is L<sup>A</sup>T<sub>E</sub>X, and a quick inspection of the logo demonstrates the capability of L<sup>A</sup>T<sub>E</sub>X in manipulating text elements. L<sup>A</sup>T<sub>E</sub>X is being used by scientists and engineers worldwide for the production of professional quality technical articles, books, and reports. At present, L<sup>A</sup>T<sub>E</sub>X is known as the system that produces the ultimate quality in technical articles that is also free. Hence, there is no wonder that several established publishers use this system as their production system (e.g., Elsevier, Springer, Wiley, CRC, SAS Institute, Oxford University Press, Cambridge University Press, BioMed Central, etc.). An online search will reveal the several advantages of using L<sup>A</sup>T<sub>E</sub>X over existing Text Processing Systems (TPS).

The history of the system can be traced back to 1978, when Donald E. Knuth (Knuth, 1984) created a computer typesetting program T<sub>E</sub>X aimed at overcoming the lack of typographic quality in the preparation of mathematical content rich technical manuscripts. T<sub>E</sub>X incorporates sound typesetting principles, known to only professional typesetters and not to general users, that improves the readability of documents. T<sub>E</sub>X now considered an error-free typesetting engine, a standard for scientific publications, and has been adopted by the

American Mathematical Society (AMS) and many other professional societies as their preferred format. L<sup>A</sup>T<sub>E</sub>X written by Leslie Lamport (Lamport, 1994) is a higher level macro package that is based on the plain T<sub>E</sub>X, a lower level programming language. L<sup>A</sup>T<sub>E</sub>X in the background actually runs the T<sub>E</sub>X typesetting engine. Introduction of L<sup>A</sup>T<sub>E</sub>X simplified the usage of T<sub>E</sub>X and made it popular and widely adopted.

L<sup>A</sup>T<sub>E</sub>X system uses ASCII text markups codes, similar to html producing webpages, when compiled produces the output based on the coded instructions in final format and such as pdf using pdfL<sup>A</sup>T<sub>E</sub>X compiler. Usage of L<sup>A</sup>T<sub>E</sub>X allows the author to concentrate on the content while it takes the role of typesetter and performs the formatting. The source file remains usually a very small sized text file, while keeping the other elements such as figures stored and accessed from a common location, which allows retaining several document versions intact. L<sup>A</sup>T<sub>E</sub>X automates several common tasks of document preparation such as table of contents, list of tables, list of figures, index (using MakeIndex), linked citations, building references (using BibL<sup>A</sup>T<sub>E</sub>X), numbering equations, figures, and tables — thus relieving the authors from the essential but less productive bookkeeping and possibly making the documents error free. Therefore L<sup>A</sup>T<sub>E</sub>X is the system of choice for typesetting books. L<sup>A</sup>T<sub>E</sub>X being structural in coding, it encourages and trains the users on good coding practices.

Since the document is made of coded commands, moving components such as sections, figures, tables, index items, etc., will be automatically updated when compiled every time, this is usually carried out manually in TPS. The engineering and science students will find it useful while preparing their thesis and subsequently using the codes for submission as journal article. Recognizing the application of L<sup>A</sup>T<sub>E</sub>X several universities (e.g., Cambridge, Harvard, Princeton, North Dakota

---

The author **Igathinathane Cannayen, ASABE Member Engineer**, Assistant Professor, Department of Agricultural and Biosystems Engineering, North Dakota State University, Fargo, North Dakota. **Corresponding author:** C. Igathinathane, Department of Agricultural and Biosystems Engineering, North Dakota State University, 1221 Albrecht Boulevard, Fargo, ND 58102; phone: 701-667-3011; fax: 701-667-3054; e-mail: Igathinathane.Cannayen@ndsu.edu

State, California Davis, Michigan, etc.) and journals have developed class (\*.cls) files and templates files (\*.tex). Document class files are combination of mostly T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X codes that has all the basic style formats, and templates are start-up sample L<sup>A</sup>T<sub>E</sub>X source files where users could easily fill in their contents and produce documents with consistent formats. The functionality of the L<sup>A</sup>T<sub>E</sub>X system is expanded constantly through user-developed packages (\*.sty), which are style files specifically made to address a particular aspect.

Several authentic print (Knuth, 1984; Kopka and Daly, 2004; Lamport, 1994; Mittelbach et al., 2004) and excellent free online sources are available (Chang, 2010; Downes, 2002; Krishnan, 2003; Oetiker et al., 2010; Roberts, 2009; Silmaril Consultants, 2009; Voß, 2010; Wikibook Contributors, 2008; Wilson, 2009) to learn T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X. Online sources are also available for BibT<sub>E</sub>X (Daly, 2010; Markey, 2009; Patashnik, 1988). A list of links of useful online L<sup>A</sup>T<sub>E</sub>X resources can be found at <http://www.eng.cam.ac.uk/help/tpl/textprocessing/> and <http://www.ee.iitb.ac.in/~trivedi/LatexHelp/latexmanual.htm>.

Although there is a learning curve associated, true with any computer program, L<sup>A</sup>T<sub>E</sub>X source codes resembling any structured programming language can be readily identified and learned by users having familiarity with computer programs. Although the L<sup>A</sup>T<sub>E</sub>X system is quite common in disciplines of mathematics, physics, and several engineering (electrical, electronics, computers, mechanical, etc.), it has not infiltrated into our Agricultural and Biological Engineering (ABE) specialization. This might be due to the lack of advertisement and awareness, albeit a free software that offers several features and produces excellent outputs both for screen and print. The software can also be a great tool for agricultural extension educators and specialists to produce professional quality documents and extension materials when the funds are limited for elaborate production. L<sup>A</sup>T<sub>E</sub>X provides all required tools and makes the author directly in-charge of production of professionally quality documents.

This paper introduces the L<sup>A</sup>T<sub>E</sub>X system, as an educational and academics tool, and illustrates from the point of view of ABE students and professionals the preparation of peer-reviewed journal article (Elsevier and ASABE) and thesis (North Dakota State University (NDSU)). ABE students and professional having the basic computer programming knowledge could able to easily utilize and benefit from the L<sup>A</sup>T<sub>E</sub>X system. The paper also discusses the merits and issues in using the L<sup>A</sup>T<sub>E</sub>X system.

## L<sup>A</sup>T<sub>E</sub>X SYSTEM

L<sup>A</sup>T<sub>E</sub>X is a versatile document processing system and is capable of several features than what are generally required in technical publications. L<sup>A</sup>T<sub>E</sub>X employs sound principles of typesetting while formatting the documents, a critical knowledge only known to typesetting professionals that most users are unaware of. Any typesetting challenge can be handled by L<sup>A</sup>T<sub>E</sub>X; the solution is either exists already or can be developed through user codes. L<sup>A</sup>T<sub>E</sub>X is also a growing system and several user-developers expand its capability through developed classes, templates, and packages. The Comprehensive TeX Archive Network (CTAN: <http://www.ctan.org>) archives the collections of all the T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X materials developed.

## STRUCTURE OF L<sup>A</sup>T<sub>E</sub>X DOCUMENT

L<sup>A</sup>T<sub>E</sub>X source code will have a simple serial structure of “preamble” followed by “document” itself (fig. 1). In the preamble, the various packages used, shortcuts definition, and settings modifications that affect the entire document were coded. In the document section, the front-matter (author, address, abstract, etc.), main text (sections, subsections, body text, figures, tables, equations, lists, etc.), and back-matter (bibliography/references, appendices, index, etc.) were coded.

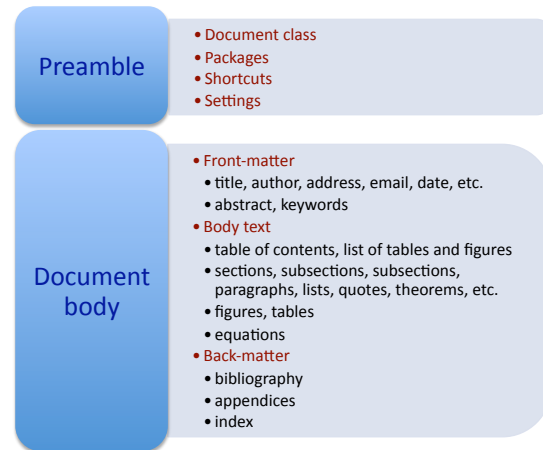


Figure 1. A simpler structure of a L<sup>A</sup>T<sub>E</sub>X document.

## OBTAINING AND INSTALLING L<sup>A</sup>T<sub>E</sub>X

To run L<sup>A</sup>T<sub>E</sub>X on a source code file (\*.tex) a compiler and an editor program are required. These can be obtained for free or purchased from commercial sources. For Windows machines, the compiler is MiKTeX 2.9 (<http://miktex.org/2.9/setup>). The MiKTeX system installs all required T<sub>E</sub>X files and other necessary programs (GhostScript and GSView), and that includes TeXworks (<http://www.tug.org/texworks/>) a front-end editor. Some of the free integrated development environments front-end editors exclusive for L<sup>A</sup>T<sub>E</sub>X that have several user-friendly features are TeXnicCenter (<http://www.texniccenter.org/>), Texmaker (<http://www.xmlmath.net/texmaker/>), TexMakerX 2.0 (<http://texmakerx.sourceforge.net/>), WinShell (<http://www.winshell.org/>), etc. Alternatively, proTeXt (<http://www.tug.org/protext/>) based on MiKTeX can be used, which is an easy-to-install L<sup>A</sup>T<sub>E</sub>X system for Windows that also installs the front-end TeXnicCenter. It is also necessary to have Adobe Reader (<http://get.adobe.com/reader/>) or Adobe Acrobat (<http://www.adobe.com/>) to view and use the pdf file created. For a Mac machines, MacT<sub>E</sub>X-2010 (<http://www.tug.org/mactex/>) provides complete installation including the front-end editor TeXShop. The installation information are available in the respective websites.

## WORKING EXAMPLES OF L<sup>A</sup>T<sub>E</sub>X DOCUMENT

### A Minimal L<sup>A</sup>T<sub>E</sub>X Document

Basically, a L<sup>A</sup>T<sub>E</sub>X document is composed of commands and text. Commands are instructions to the compiler and they were

identified by a backslash as the starting character ‘\’ and followed by the optional arguments enclosed in square brackets ‘[...]’, and the necessary arguments enclosed in braces ‘{...}’ (e.g., `\includegraphics [width=5in]{fig1.pdf}` and `\noindent`). The numbers of these arguments vary with commands. The other regular text materials are simple typed ASCII text, and they fall into two categories: “math” and “text.” Correspondingly, these are “text mode,” which is the default mode, and “math mode,” which is employed for mathematical elements where the material is enclosed between dollar characters ‘\$...\$’. These modes determine how the typed codes were typeset and outputs produced. Figure 2 shows a small two-line  $\LaTeX$  document illustrating the document structure (fig. 1) elements, modes, and the resulting final pdf output.

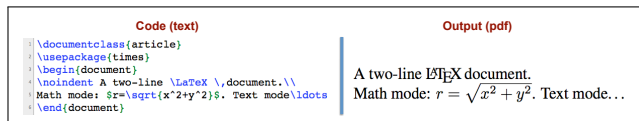


Figure 2. A two-line  $\LaTeX$  document source code and its pdf output.

### Description of the Document Source Code

Although minimal on code, figure 2 follows the document structure (fig. 1). The preamble section has the `\documentclass` and the class argument chosen was `article`. The other class arguments possible are `book`, `letter`, `report`, `slides`, and `proc`. The `\usepackage{times}` command replaces the default “Computer Modern Font” by “Times Roman Font.” The commands `\begin{document}` and `\end{document}` on lines 3 and 6 signify the main document environment, where the actual content of the document goes. The simple command `\noindent` on line 4, having no argument, suppresses the default indent provided at the start of any paragraph. The command `\LaTeX` is a special command that produces the logo  $\LaTeX$ , `\,` provides a thin space, and `\\` means carriage return or newline. The math mode on line 5 produced the mathematical online equation and it uses `\sqrt{...}` command to produced square root symbol over the entire argument, `^` operator produced the superscripts, and the `\ldots` produced the ellipses. In essence, every aspect of the document including the formatting instructions should be coded in the input (`*.tex`) file. The compactness of text and mathematical elements and overall quality of output were apparent. Figure 2 thus demonstrates how documents of simple nature can be typeset logically.

Inclusion of other elements of the  $\LaTeX$  document structure (fig. 1) will lead to documents complete in all respects, and will be discussed later. Selection of different document classes (`*.cls`) will produce outputs of unique style, and will be enhanced by a unique set of packages (`*.sty`) and commands. The document class files are written basically in plain  $\TeX$  as well as with  $\LaTeX$  codes and their development was beyond the scope of this paper.

By now the readers would have realize that  $\LaTeX$  is not like the usual menu-driven TPS where they directly type the final output (What You See Is What You Get (WYSIWYG) system); and with  $\LaTeX$  they have to type the code and compile it to see the result that is sometimes termed “What You Think Is

What You Get” (WYTIWYG) system. One issue with WYSIWYG TPS is no two users will produce the similar outputs, unless they use a common template, but with  $\LaTeX$  structural coding principle the outputs will be similar.

### RUNNING A $\LaTeX$ DOCUMENT SOURCE CODE

It is a good idea to try out installed  $\LaTeX$  system through a minimal document such as this (fig. 2). In most cases, the sequence of operations performed are (1) “compiling or building,” and (2) “viewing” to obtain the final output. Sometimes, these commands can be combined through shortcuts. The command “Typeset” also refers to compiling. Typesetting or compiling being the common operation, and they will be performed on the compiler ( $\LaTeX$ ,  $\BibTeX$ ,  $\MakeIndex$ , etc.) selected or active at the moment.  $\LaTeX$  and  $\BibTeX$  during compilation would generate several auxiliary files with extensions such as `aux`, `bbl`, `blg`, `fff`, `lof`, `log`, `lot`, `out`, `spl`, `synctex.gz`, `toc`, and `ttt`, and these after finalization can be deleted, while retaining the source (`tex`) and output (`pdf`) files.

Errors and warnings, if any, during compilation will be shown notified through console window or generated log files (`*.log`). The ‘log’ file can be opened through the text editor and inspected to debug or fix the errors. The error messages with indicated line numbers, and the console help inquiries could guide the debugging. Similar to any programming language, a simple missing matching curly braces or ‘\$’ will interrupt the compilation producing errors and warnings.

Many a times it helps to make the compiler to run through completely, when faced with simple error, which resolves the issues with the automatic compiler support and code substitution. Running the compiler through when stalled is performed by issuing commands ‘r’ (`\nonstopmode`: run without stopping), ‘q’ (`\batchmode`: run quietly), and ‘x’ (quit). Typing ‘h’ will display some help programmed in the compiler. Observing such outputs with bugs sometimes helps solving the coding issues. However, the user needs to fix errors in the code later after identifying them. Based on the contents,  $\LaTeX$  requires a few sweeps of compilation (one to three) to resolve various citation links (floats, equations, table of contents, lists of figures and tables, etc.) during compilation.

### A SAMPLE ELSEVIER JOURNAL ARTICLE

Elsevier is one of the leading international publishers that uses  $\LaTeX$  and encourages the authors to use the system for article preparation. Elsevier’s “elsarticle.cls”  $\LaTeX$  document class webpage (<http://www.elsevier.com/t/find/authorsview.authors/elsarticle>) has all the information (class, bibliography style files, and manuscript template files) required for preparing journal articles. Twelve of these templates and bibliographic styles ([http://www.elsevier.com/framework\\_authors/misc/journal\\_refstyles.pdf](http://www.elsevier.com/framework_authors/misc/journal_refstyles.pdf)), having a lot of similarities, would support several hundreds of the Elsevier journal article styles. The webpage: <http://support.river-valley.com/wiki/index.php?title=Elasarticle.cls> provides author support for “elsarticle.cls.”



## A Modified L<sup>A</sup>T<sub>E</sub>X Elsevier Template for Articles

From the Elsevier “elsarticle” document class web page mentioned earlier, a modified “elsarticle-template-2-harv.tex” template with additional commands and comments, while removing the supplied supplemental information, is shown in figure 3. It can also be noted that this template follows the document structure (fig. 1) and extends the two-line document (fig. 2) to a full-fledged article. With color-coded front-end editors, the comments (red), commands (blue), and normal text (black) will be seen in different colors (fig. 3).

Comments are those lines with the first character ‘%’, and will be ignored by the compiler, hence not appear in the final output. Comments serve the same purpose they perform with programming languages as providing helpful code explanation documentation. Commenting also serves in an indirect way to include or exclude the commands before processing. Thus, the line 4 (fig. 3) is active and calls for several options used therein, while line 5 is not. A simple switching of the ‘%’ between lines 4 and 5 will transform the two-column final formatted output to an usual single column review style double spaced output. Similarly, based on the needs, the commands on lines 21–24 can be preferentially commented out. These commands, as they read, produce the respective components of the document, and their placement in the code reflects their location on the output.

Lines 27–43 (fig. 3) outlines the main text block that is composed of sections and subsections. The text that goes into the required arguments of `\section`, `\subsection`, `\subsubsection`, etc., will be typeset as respective headings in designed formats coded in the document class. The regular body text that makes most of the content (text and math modes) will be coded below these headings as paragraphs of text. A blank line (or lines) between the paragraphs makes individual paragraphs in the output and the first line is indented by default.

Bibliography generation is performed by including the bibliography style file (\*.bst) and the user bibliography database file (\*.bib) as depicted on lines 44–45 (fig. 3). It is also possible to generate the references by coding manually the text as “\bitem” inside a “\thebibliography” environment.

## Completed Elsevier Template and Output of this Paper

Coding of the template (fig. 3) considering “Computers and Electronics in Agriculture” as an example target journal with specific information and additional commands that reproduces the present paper is shown in figure 4. It can be seen that the completed template follows the logical development of the manuscript by populating the template with relevant material. More details on using the “elsarticle.cls” and various options can be found elsewhere (Elsevier Ltd., 2009). The first page of the generated output from the completed template in a two-column final style is presented in figure 5.

Line 1 makes ‘twocolumn’ and ‘final’ options of `elsarticle` class active. Lines 3–6 include several useful packages. Based on the requirements of the article the packages are included (e.g., if Euro currency symbol needed, then ‘eurosym’ package is included), rather than including available packages. Some packages have optional arguments (lines 4, 6) shown enclosed in square brackets. Line 8 uses the ‘egu’ cite style that produces author names and year, while ‘plain’

```

1 %% This is file 'elsarticle-template-2-harv.tex'
2 %% Copyright 2009 Elsevier Ltd
3 -----
4 \documentclass[final,authoryear,5p,times,twocolumn]{elsarticle}
5 %\documentclass[authoryear,preprint,review,12pt]{elsarticle}
6 \usepackage{graphicx,amssymb,amsmath}
7 %% Add other packages and \newcommand shortcuts in this preamble
8 -----
9 \begin{document}
10   \begin{frontmatter}
11     \title{}
12     \author{} %% Repeat for additional authors
13     \address{} %% Repeat for additional authors
14     \begin{abstract}
15       ... Text of Abstract goes here ...
16     \end{abstract}
17     \begin{keyword}
18       ... Keywords here, in the form: keyword \sep keyword
19     \end{keyword}
20   \end{frontmatter}
21   \linenumbers %% Comment this line (add %) to remove line numbers
22   \tableofcontents %% Comment this line to remove table of contents
23   \listoftables %% Comment this line to remove list of tables
24   \listoffigures %% Comment this line to remove list of figures
25 -----
26   %% Main text
27   \section{} %% E.g.: Introduction (will be numbered 1.)
28     ... Text of this section goes here ...
29   \section{} %% E.g.: Materials and Methods (will be numbered 2.)
30     ... Text of this section goes here ...
31   \subsection{} %% E.g.: Test materials (will be numbered 2.1.)
32     ... Text of this subsection goes here ...
33   \subsubsection{} %% E.g.: Preparation of samples (will be numbered 2.1.1.)
34     ... Text of this subsubsection goes here ...
35   \section{} %% E.g.: Results and Discussion (will be numbered 3.)
36     ... Text of this section goes here ...
37   \subsection{} %% E.g.: Test results (will be numbered 3.1.)
38     ... Text of this section goes here ...
39   \section{} %% E.g.: Conclusion (will be numbered 4.)
40     ... Text of this section goes here ...
41   \section*{} %% E.g.: Acknowledgments (will be unnumbered)
42     ... Text of this section goes here ...
43   \section*{} %% E.g.: References (will be unnumbered)
44   \bibliographystyle{model2-names} %% References generated automatically
45   \bibliography{<your-bib-database>} %% based on .bgi and .bib files
46 \end{document}
47 -----

```

Figure 3. A sample Elsevier journal manuscript modified template.

```

1 \documentclass[final,5p,times,twocolumn,number]{elsarticle}
2
3 \usepackage{graphicx,amssymb,amsmath,lscape,rotating,booktabs,units}
4 \usepackage{colorlinks}{hyperref}
5 \usepackage{float,lineno,fancyhdr,ifthen,caption,threeparttable,lipsum,xspace}
6 \usepackage{usenames,dvipsnames,table}{xcolor}
7
8 \citestyle{egu}
9 \newcommand\lx{\LaTeX\space} \newcommand\tr{\TeX\space} \newcommand\bx{Bib\{\TeX\space}
10
11 \begin{document}
12 \begin{frontmatter}
13 \title{\vspace{-0.5in}\includegraphics[width=\textwidth]{title-else.pdf}}
14 \LARGE{LaTeX Document Preparation System (\lx) for Agricultural and Biological Engineers)}
15
16 \author[ndsu]{C.-Igathinathane\, \corres{cr1}}
17 \cotext{cr1}{Corresponding author. Tel.:+1 701 667 3011; fax:+1 701 667 3054.}
18 \ead{Igathinathane.Cannayen@ndsu.edu}
19 \author[sec]{Second-Author\,}
20 \address[ndsu]{Department of Agricultural and Biosystems Engineering, North Dakota State
21 University, 1221 Albrecht Boulevard, Fargo, ND 58102, USA}
22 \address[sec]{Second author's Department, Organization, Street address, City, Zip, and
23 Country information goes here \ldots}
24
25 \begin{abstract}
26 LaTeX (\lx) is a document preparation typesetting system used by scientists and engineers
27 worldwide for producing professional quality technical articles, books, and reports. It is
28 the best system known for producing documents with mathematical equations and for very
29 large books. Even though \lx system is common in disciplines of mathematics, physics, and
30 ..... Text skipped .....
31
32 of the system were also discussed. Although there is a learning curve involved, the ABEN
33 students and ASABE professionals having the basic computer programming knowledge could
34 easily benefit from \lx system readily.
35 \end{abstract}
36
37 \begin{keyword}
38 ASABE \sep Article \sep Document \sep LaTeX \sep Paper \sep Thesis \sep Typesetting
39 \end{keyword}
40 \end{frontmatter}
41
42 \section{Introduction}
43 LaTeX is pronounced ‘Lay-tech’ or ‘Lah-tech’ and is an extensive document production
44 typesetting system and a markup programming language. The logo of LaTeX is \lx, and a
45 quick inspection of the logo demonstrates the capability of \lx in manipulating text
46 elements. \lx is being used by scientists and engineers worldwide for the production of
47 professional quality technical articles, books, and reports. At present, \lx is known as
48 the system that produces the ultimate quality in technical articles that is also free.
49 Hence, there is no wonder that several established publishers use this system as their
50 production system (e.g., Elsevier, Springer, Wiley, CRC, SAS Institute, Oxford University
51 Press, Cambridge University Press, BioMed Central, etc.). An online search will reveal the
52 several advantages of using \lx over existing Text Processing Systems (TPS).
53
54 The history of the system can be traced back to 1978, when Donald E. Knuth
55 \citep{Knuth1984} created a computer typesetting program \lx aimed at overcoming the lack
56 of typographic quality in the preparation of mathematical content rich technical
57 manuscripts. \lx incorporates sound typesetting principles, known to only professional
58 typesetters and not to general users, that improves the readability of documents. \lx now

```

Figure 4. The completed Elsevier journal manuscript template with source code input.

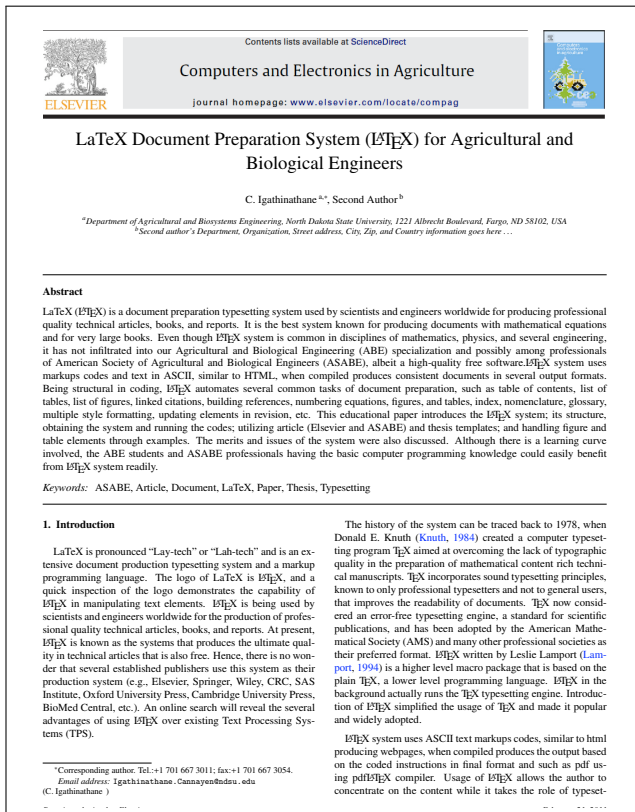


Figure 5. A sample first page of Elsevier journal manuscript generated output.

the bracketed numbers. Line 9 defines shortcuts for  $\LaTeX$ ,  $\TeX$ , and  $\BibTeX$  logos, this means simply typing the shortcut, for example, `\lx` when compiled will produce the  $\LaTeX$  logo in the output (fig. 5). In the `title` command (line 13) argument field a graphic file `title-else.pdf` was included to show a typical header figure of the selected journal and the font size was increased by `\LARGE` commands (fig. 5). The optional arguments `ndsu` of `\author` (line 16) is used as a key to link the `\address` entry (line 20), and so on, for other authors (two authors are shown as an example; lines 19, 21). One of the author should be identified as corresponding author through `\corref{...}` command (line 16), and the relevant details are supplied through `\cortext{...}` command (line 17). Additional authors are added as required.

The text of `abstract` was coded in its environment (`\begin{abstract}` and `\end{abstract}`) as shown (lines 23–25). Similarly, the `keywords` coded in its environment, while individual keywords are separated by `\sep` command, specific to `elsarticle` (lines 27–29). It can be noted that all these elements are coded inside the `frontmatter` environment (lines 12–30).

The main body text is coded from line 32 onwards, opening with a section title produced by the command `\section{Introduction}`, followed by regular text and shortcuts (multi-line 33). It can be seen other than the usage of shortcut `\lx` the body text of multi-line 33 is common running text as it is typed on any word processor. The blank on line 34 is created by pressing the “return” key, signifying the end of a paragraph, and this makes the compiler to end the paragraph and start the next one with an indent.

The second paragraph (multi-line 35) uses a reference citing command `\citep{Knuth1984}`, which cites the author and year in parenthesis and creates a full citation entry on the “Reference” section of the article, based on the bibliography style and database used (fig. 3). Handling of bibliography was performed through  $\BibTeX$  program (Daly, 2010; Markey, 2009; Patashnik, 1988). Citing references while typing the text and building the citations and references automatically in correct format are an attractive feature of  $\LaTeX$ . This avoids the “classical error” of uncited and unlisted references, commonly encountered in manuscript preparation. The other missing features that make a technical paper are figures and tables, known as `floats`, and their handling will be discussed later.

It can be observed that the various sophisticated final formats that appear on the output (fig. 5) in the form of title, abstract, footnote and main text were the result of actual codes built in the `elsarticle` class, while keeping the template simple, logical and user-friendly. Thus it is possible to create back-end document class (`*.cls`) and bibliography style (`*.bst`) files with corresponding front-end simple-to-use templates to produce custom-made outputs, as followed by different journals and thesis of several universities. However, development of a new document class is a very involved process — usually performed by professionals; however, the  $\LaTeX$  templates could be used easily by general computer-aware users.

## DEVELOPMENT OF ASABE JOURNAL ARTICLE CLASS

One way of creating a class file is to modify an existing one to suit the requirement of the proposed format. Most of the journal article class files inherit the basic `article.cls` and modifies it to suit their requirement. This requires an understanding of the plain  $\TeX$  and other  $\LaTeX$  codes used in the class files. A simplified new `asabe.cls` class and a `asabe-journal-template.tex` template were developed based on the codes of `elsarticle` class and the corresponding modified template (fig3). The  $\LaTeX$  output of the first page of the present article using the developed `asabe.cls` producing output similar to ASABE journal (e.g., Applied Engineering in Agriculture) is shown in figure 6. Using and populating the template (not shown) to generate the article is similar to that followed earlier (fig. 4).

Document class file (`asabe.cls`) development involves coding of all required format specifications to create the intended article style. The codes of the class are internal (back-end), so that the articles can be created with least modification through template (front-end), outside of the class file. Actually during compilation all the template codes will be substituted from the inherited class and package codes to generate the final output.

The major modifications of `asabe` class include: small capitals font for the title and sections, simplified author list, the abstract text and keywords in italics, drop-capital (`lettrine` package) for the first letter, and authors information given as a footnote (`\footnotetext` command), bold table and figure captions, specific odd and even page footer details with a different first page (`fancyhdr, lastpage` packages), and Chicago bibliography style (`chicagoa.bst`) formatting that were usually followed in the ASABE journals (fig. 6). With these classes and template files, the users could able to generate outputs that resemble professionally produced ASABE style offprints. It can be realized that the output qual-

LaTeX DOCUMENT PREPARATION SYSTEM (L<sup>A</sup>T<sub>E</sub>X) FOR AGRICULTURAL AND BIOLOGICAL ENGINEERS

C. Igathinathane

**ABSTRACT.** LaTeX (L<sup>A</sup>T<sub>E</sub>X) is a document preparation typesetting system used by scientists and engineers worldwide for producing professional quality technical articles, books, and reports. It is the best system known for producing documents with mathematical equations and for very large books. Even though L<sup>A</sup>T<sub>E</sub>X system is common in disciplines of mathematics, physics, and several engineering, it has not infiltrated into our Agricultural and Biological Engineering (ABE) specialization and possibly among professionals of American Society of Agricultural and Biological Engineers (ASABE), albeit a high-quality free software. L<sup>A</sup>T<sub>E</sub>X system uses markups codes and text in ASCII, similar to HTML, when compiled produces consistent documents in several output formats. Being structural in coding, L<sup>A</sup>T<sub>E</sub>X automates several common tasks of document preparation, such as table of contents, list of tables, list of figures, linked citations, building references, numbering equations, figures, and tables, index, nomenclature, glossary, multiple style formatting, updating elements in revision, etc. This educational paper introduces the L<sup>A</sup>T<sub>E</sub>X system, its structure, obtaining the system and running the code; utilizing article (Elzevier and ASABE) and thesis templates; and handling figure and table elements through examples. The merits and issues of the system were also discussed. Although there is a learning curve involved, the ABE students and ASABE professionals having the basic computer programming knowledge could easily benefit from L<sup>A</sup>T<sub>E</sub>X system readily.

**Keywords.** ASABE, Article, Document, Education, LaTeX, Thesis, Typesetting

LaTeX is pronounced "Lay-tech" or "Lah-tech" and is an extensive document production typesetting system and a markup programming language. The logo of LaTeX is L<sup>A</sup>T<sub>E</sub>X, and a quick inspection of the logo demonstrates the capability of L<sup>A</sup>T<sub>E</sub>X in manipulating text elements. L<sup>A</sup>T<sub>E</sub>X is being used by scientists and engineers worldwide for the production of professional quality technical articles, books, and reports. At present, L<sup>A</sup>T<sub>E</sub>X is known as the system that produces the ultimate quality in technical articles that is also free. Hence, there is no wonder that several established publishers use this system as their production system (e.g., Elsevier, Springer, Wiley, CRC, SAS Institute, Oxford University Press, Cambridge University Press, BioMed Central, etc.). An online search will reveal the several advantages of using L<sup>A</sup>T<sub>E</sub>X over existing Text Processing Systems (TPS).

The history of the system can be traced back to 1978, when Donald E. Knuth (Knuth, 1984) created a computer typesetting program T<sub>E</sub>X aimed at overcoming the lack of typographic quality in the preparation of mathematical content rich technical manuscripts. T<sub>E</sub>X incorporates sound typesetting principles, known to only professional typesetters and not to general users, that improves the readability of documents. T<sub>E</sub>X now considered an error-free typesetting engine, a standard for scientific publications, and has been adopted by the

American Mathematical Society (AMS) and many other professional societies as their preferred format. L<sup>A</sup>T<sub>E</sub>X written by Leslie Lamport (Lamport, 1994) is a higher level macro package that is based on the plain T<sub>E</sub>X, a lower level programming language. L<sup>A</sup>T<sub>E</sub>X in the background actually runs the T<sub>E</sub>X typesetting engine. Introduction of L<sup>A</sup>T<sub>E</sub>X simplified the usage of T<sub>E</sub>X and made it popular and widely adopted.

L<sup>A</sup>T<sub>E</sub>X system uses ASCII text markups codes, similar to html producing webpages, when compiled produces the output based on the coded instructions in final format and such as pdf using pdfl<sup>A</sup>T<sub>E</sub>X compiler. Usage of L<sup>A</sup>T<sub>E</sub>X allows the author to concentrate on the content while it takes the role of typesetter and performs the formatting. The source file remains usually a very small sized text file, while keeping the other elements such as figures stored and accessed from a common location, which allows retaining several document versions intact. L<sup>A</sup>T<sub>E</sub>X automates several common tasks of document preparation such as table of contents, list of tables, list of figures, index (using MakeIndex), linked citations, building references (using BibL<sup>A</sup>T<sub>E</sub>X), numbering equations, figures, and tables — thus relieving the authors from the essential but less productive bookkeeping and possibly making the documents error free. Therefore L<sup>A</sup>T<sub>E</sub>X is the system of choice for typesetting books. L<sup>A</sup>T<sub>E</sub>X being structural in coding, it encourages and trains the users on good coding practices.

Since the document is made of coded commands, moving components such as sections, figures, tables, index items, etc., will be automatically updated when compiled every time, this is usually carried out manually in TPS. The engineering and science students will find it useful while preparing their thesis and subsequently using the codes for submission as journal article. Recognizing the application of L<sup>A</sup>T<sub>E</sub>X several universities (e.g., Cambridge, Harvard, Princeton, North Dakota

The author Igathinathane Camayen, ASABE Member Engineer, Assistant Professor, Department of Agricultural and Biosystems Engineering, North Dakota State University, Fargo, North Dakota. Corresponding author: C. Igathinathane, Department of Agricultural and Biosystems Engineering, North Dakota State University, 221 Albrecht Boulevard, Fargo, ND 58102; phone: 701-667-3011; fax: 701-667-3054; e-mail: igathinathane.camayen@ndsu.edu

Applied Engineering in Agriculture  
© 2011 American Society of Agricultural and Biological Engineers

Vol. No: 1-12

1

Figure 6. L<sup>A</sup>T<sub>E</sub>X output of the first page of present paper in ASABE "Applied Engineering in Agriculture" format.

ity produced by free L<sup>A</sup>T<sub>E</sub>X (figs. 5–6), in usable pdf format, cannot be easily achieved by commercial TPS.

MATHEMATICAL ELEMENTS HANDLING CAPABILITY OF L<sup>A</sup>T<sub>E</sub>X - A COMPARISON

Environments such as 'equation' and 'align' for single and set of equations, respectively, create numbered and unnumbered equations. A sample of mathematical elements (equations) from the published ASABE journal articles and the equivalent created by L<sup>A</sup>T<sub>E</sub>X is shown figure 7 as a comparison.

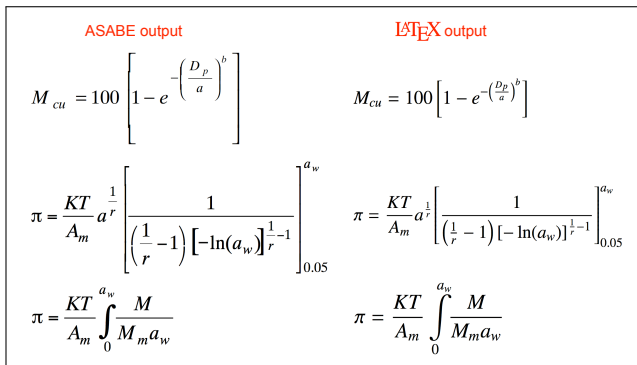


Figure 7. Comparison of mathematical elements from ASABE journal and L<sup>A</sup>T<sub>E</sub>X outputs.

It is evident from the outputs (fig. 7) that L<sup>A</sup>T<sub>E</sub>X produces the

best output that is compact without dead spaces and elements properly arranged in their appropriate location. It is also possible with L<sup>A</sup>T<sub>E</sub>X to have these complicated equations "in-line" without any loss of quality as:  $M_{cu} = 100 \left[ 1 - e^{-\left(\frac{D_p}{a}\right)^b} \right]$ ;

$$\pi = \frac{KT}{A_m} a^{\frac{1}{2}} \left[ \frac{1}{\left(\frac{1}{r}-1\right) [-\ln(a_w)]^{\frac{1}{2}-1}} \right]_{0.05}^{a_w}; \quad \pi = \frac{KT}{A_m} \int_0^{a_w} \frac{M}{M_m a_w}$$

always possible with TPS. The L<sup>A</sup>T<sub>E</sub>X codes that were used to create standalone and in-line equations were almost similar. Every imaginable feature of handling mathematical elements were built already in L<sup>A</sup>T<sub>E</sub>X system or obtained through user-developed packages.

Several complicated equations or mathematical elements can be built by series of simple commands in a logical manner producing the best quality output possible. For instance, the code used for the first and third in-line equations is:  $\$M_{cu}=100\left[1-e^{-\left(\frac{D_p}{a}\right)^b}\right]$  and  $\pi=\frac{KT}{A_m}\int_0^{a_w}\frac{M}{M_m a_w}$ . Notice the use of math mode ( $\$ \dots \$$ ); otherwise specific equation environments should be employed. Excellent online resources that deal with handling of mathematical elements are available to readers (AMS, 2002; Downes, 2002; Voß, 2010).

FLOATS HANDLING IN L<sup>A</sup>T<sub>E</sub>X

Tables and figures are considered as floats, and are placed by the L<sup>A</sup>T<sub>E</sub>X compiler depending on the available space. The common placement specifier ' $[htbp]$ ' that means **h** - here, **t** - top, **b** - bottom, **p** - page, helps to control the placement. The other option is '**H**' that forces the floats in the place where the user wants it and it uses the package '**float**.' This option may leave a lot of space before the float if enough room is not there, but gives absolute control to the users the way they were used to with general TPS. If enough room is not there the floats will be pushed to the next column or page. This is common with typesetting of professional documents.

With L<sup>A</sup>T<sub>E</sub>X the dimensions of the floats can be scaled to the requirement through several dimension options, which is also used to control their placement. The floats can be captioned, and will be sequentially numbered automatically. They can be referred through a declared label and cited in the text; and when the floats are moved around during revisions, these numbers and cited contents will be automatically updated — a definite advantage of using L<sup>A</sup>T<sub>E</sub>X.

A Figure Handling Example

A basic float figure code and its output (fig. 8) is shown side-by-side below:



Figure 8. A float figure example with source code.

Codes such as these were used in creating the other figures (figs. 1–14) used in this paper. The 'figure' environment, which requires 'graphicx' package, was used to include the figures of type jpg, png, and pdf to be compiled



with ‘`pdflatex`.’ This environment declares the figure caption and label (`asabefig`) that can be used to create a link and refer to and cite the figure in the text. For example, the command `\hflink{asabefig}` defined in ‘`asabe.cls`’ creates a citation hyperlink in the text such as ‘fig. 8’ and when clicked will automatically navigate to the item. Such hyperlinks work with all cited items such as equations, floats, references, pages, section heads, etc. The figure used here is ‘`afig.pdf`’ that is usually stored in the same folder as the  $\LaTeX$  source file; otherwise the path of the figures folder should be specified. Several other features of manipulating figures (dimensions, scale, rotation, clip, text wrap, frame, sub-figures, etc.) are possible through packages and specific commands.

### A Table Handling Example

A basic float table code and its output (fig. 9) is shown side-by-side below:

Code (text)	Output (pdf)																							
<pre> \begin{table}[htbp] \centering \caption{A sample float table} \begin{tabular}{llrc} \toprule \crops{\multicolumn{2}{c}{Moisture (\%)}\&amp;order\&amp;} \midrule(r){2-3} &amp; Wet &amp; Dry &amp; (t)\&amp; \midrule Corn &amp; 14.0 &amp; 16.28 &amp; 20\&amp; Wheat &amp; 13.5 &amp; 15.61 &amp; 15\&amp; \midrule Rice &amp; 12.8 &amp; 14.68 &amp; 4 \&amp; Canola &amp; 14.2 &amp; 16.55 &amp; 10\&amp; \bottomrule \end{tabular} \label{egtable} \end{table} </pre>	<p>Table 1: A sample float table</p> <table border="1"> <thead> <tr> <th rowspan="2">Crop</th> <th colspan="2">Moisture (%)</th> <th rowspan="2">Order</th> </tr> <tr> <th>Wet</th> <th>Dry</th> <th>(t)</th> </tr> </thead> <tbody> <tr> <td>Corn</td> <td>14.0</td> <td>16.28</td> <td>20</td> </tr> <tr> <td>Wheat</td> <td>13.5</td> <td>15.61</td> <td>15</td> </tr> <tr> <td>Rice</td> <td>12.8</td> <td>14.68</td> <td>4</td> </tr> <tr> <td>Canola</td> <td>14.2</td> <td>16.55</td> <td>10</td> </tr> </tbody> </table>	Crop	Moisture (%)		Order	Wet	Dry	(t)	Corn	14.0	16.28	20	Wheat	13.5	15.61	15	Rice	12.8	14.68	4	Canola	14.2	16.55	10
Crop	Moisture (%)		Order																					
	Wet	Dry		(t)																				
Corn	14.0	16.28	20																					
Wheat	13.5	15.61	15																					
Rice	12.8	14.68	4																					
Canola	14.2	16.55	10																					

Figure 9. A float table example with source code.

The ‘`tabular`’ environment creates the table, and the table specification (e.g., `{llrc}`) specifies the number of columns considered and their justification (`l` - left, `r` - right, and `c` - center, etc.). The commands ‘`\toprule`’, ‘`\midrule`’ and ‘`\bottomrule`’ draw horizontal lines and they require ‘`booktabs`’ package. The ‘`\cmidrule`’ command draws a thin line that spans only to the column numbers specified. Different column entries were entered as regular text and were separated by the column separator ‘`&`’ character, and new rows started by ‘`\&`’ as indicated in figure 9. The float ‘`table`’ environment, similar to ‘`figure`’ supports caption and label features. Mathematical characters and expressions can also be inserted in the tables through math mode. It should be noted that tables can be created as a part of the article source; however, it is also possible to have the codes on a separate ‘`*.tex`’ file and include (`\input{table1.tex}` command) in the article. Several other features of handling tables (coloring, decimal alignment, landscape table, three-part table, long table, etc.) are possible through packages and specific commands.

## HANDLING BIBLIOGRAPHY USING $\LaTeX$ AND $\BibTeX$

Handling bibliography can be performed directly with  $\LaTeX$  using built-in features and simpler commands, or using the auxiliary program  $\BibTeX$  — a bundled standard tool with  $\LaTeX$ . The package ‘`natbib`’, loaded by default with ‘`elsarticle.cls`’, is an useful package to manage citations both in numerical and author-year formats using various `\cite` commands explained in its documentation (Daly, 2010).  $\BibTeX$  program in combination with  $\LaTeX$  is one of the strong points of the system. Figure 10 shows a section of the generated references using the style ‘`chicago.bst`’:

```

Igathinathane, C., L. O. Pordesimo, E. P. Columbus, W. D. Batchelor,
and S. Sokhansanj (2009). Sieveless particle size distribution anal-
ysis of particulate materials through computer vision. Computers
and Electronics in Agriculture 66(2), 147–158.
Knuth, D. E. (1984). The TeXbook. Addison-Wesley Professional,
Reading, Massachusetts, US. ISBN: 0-201-13448-9.
Lamport, L. (1994). TeX: A Document Preparation System (2
ed.), Volume 1985. Addison-Wesley, Reading, Massachusetts, US.
ISBN: 0-201-52983-1.

```

Figure 10. An example of generated list of reference using with code.

### $\LaTeX$ built-in Bibliography

$\LaTeX$  directly allows the handling of bibliography through the ‘`thebibliography`’ environment. Such in-line bibliography items (`bibitem`) are manually coded and placed where they have to appear in the article. The `bibitem`’s were coded in a way to reflect all the required formatting features that is specific to the journal. The code of built-in bibliography of the output (fig. 10) is shown in figure 11:

```

\begin{thebibliography}{99}
\bibitem[Igathinathane et al.(2009)]{Igathi09147}
  Igathinathane, C., L. O. Pordesimo, E. P. Columbus, W. D. Batchelor,
  and S. Sokhansanj (2009). Sieveless particle size distribution
  analysis of particulate materials through computer vision.
  \emph{Computers and Electronics in Agriculture} 66(2), 147–158.
\bibitem[Knuth(2006)]{Knuth06}
  Knuth, D. E. (1984). The TeX book. Addison-Wesley Professional,
  Reading, Massachusetts, US. ISBN: 0-201-13448-9.
\end{thebibliography}

```

Figure 11. An example code of  $\LaTeX$  built-in bibliography items.

The ‘`bibitem`’ entries have unique keys, descriptions and reference text (fig. 11). The keys (e.g., `Igathi09147` and `Knuth06`) are user-defined unique identifiers and the descriptions before them (e.g., `Igathinathane et al.(2009)` and `Knuth (2006)`) are the information that will appear when the key is cited in the body text. The rest of the information (lines 312–315 and 317–318) is the actual reference text that appear as listing (fig. 10), with all formatting features specific to the journal.

Typing out the references in this manner is similar to typing in the references using TPS. However, the unique keys of the `bibitem`s can be cited in the text and an automatic citation hyperlink will be created, when clicked will navigate to that particular reference listing directly. When compiled, the references listing will appear in the coded style as final format (fig. 10). One shortcoming of this approach is when the article has to be formatted into a different journal, then all the format styles of the reference information of ‘`bibitem`s’ need to be recoded.

### $\BibTeX$ Automatic Bibliography

Automatic bibliography generation appears rather simpler. It involves simply including the style file (e.g., `modell-num-names.bst`) and the actual user bibliography database (e.g., `user-references.bib`) using commands `\bibliographystyle` and `\bibliography` commands (Daly, 2010; Markey, 2009; Patashnik, 1988), respectively (fig. 3). The publisher supplies the bibliographic style files specific to the journal (<http://www.elsevier.com/wps/find/authorsview.authors/elsarticle>), and the bibliography database should be developed by the user. With

these two commands when the source code compiled in the sequence of  $\LaTeX$ , Bib $\TeX$ ,  $\LaTeX$  and  $\LaTeX$  will generate the citation text links as well as the list of references automatically. This multiple compilation is required to resolve the links and to generate the required files, mostly during the first time. The Bib $\TeX$  compilation actually generates a `*.bbl` file having  $\LaTeX$  `'thebibliography'` environment codes. The ASABE journals follow a reference style similar to `'chicagoa.bst'`, which was used in this paper.

The list of references will be built only based on the citations commands (e.g., `\citep{key}` or `\citet{key}`) with appropriate keys typed out in the body text (Daly, 2010; Markey, 2009; Patashnik, 1988) while developing the article. This process is also known "Cite While You Write." Thus the list of references generated will be based only on cited references, hence avoiding the classical error of erroneous (uncited and unlisted) references and citations. The bibliography style files (`*.bst`) supply the formats required to specific journal. If the article has to be formatted into an another journal style, only changing the appropriate `*.bst` file handles the issue. The argument of the `\citestyle{...}` command of `'natbib'` such as `'egu'` will produce the "author-year, semicolon with round brackets" (fig. 4), `'plain'` the "numbered, comma with square brackets," and `'nature'` the "superscripts" in the text body and reference listing (Daly, 2010). The behavior of citations in the text and the reference generation can be altered by various style files and citation commands of Bib $\TeX$  (Markey, 2009; Patashnik, 1988).

### Building Bib $\TeX$ User Bibliography Database

A sample bibliography entry of type article (Igathinathane et al., 2009) that is used in this article as an example (fig. 12) is shown below:

```

330 @article{Igathi09147,
331   author = "C. Igathinathane and L. O. Pordesimo and E. P. Columbus and
332   W. D. Batchelor and S. Sokhansanj",
333   title = "Sieveless particle size distribution analysis of particulate
334   materials through computer vision",
335   journal = "Computers and Electronics in Agriculture",
336   volume = "66",
337   number = "2",
338   pages = "147-158",
339   year = "2009"
340 }

```

Figure 12. Bib $\TeX$  entry example for automatic citing and reference generation.

Among the several types of Bib $\TeX$  entries, the commonly used ones are: `article`, `book`, `conference`, `proceedings`, `masterthesis`, `phdthesis`, `techreport`, `misc`, etc. (Daly, 2010). The entries start with '@' symbol followed by the entry type, the unique key, fields inputs enclosed in quotes and separated by comma, and all enclosed in curly braces (fig. 12). These entries have their required (e.g., `author`, `title`, `journal`, `year`, etc.), and optional fields (`volume`, `number`, `pages`, etc.). Each field is assigned the appropriate values following '=' sign within double quotes ("...") or curly braces. Multiple authors are separated by the word `'and'`, and their initials are separated by a space. Order of the fields is immaterial. It can be noticed that no formatting information is included in these entries, and formatting is performed during compilation based on the bibliography style file included in the source code. The collection of such entries will make a small sized flat-file (`*.bib`) that allows for "Write

Once and Read Many" operation; thus the authors can have a master `'bib'` file of all useful references and automatically draw only the required item from that to create the citations and references in the article.

The bibliography entries can either be typed in manually, following the necessary fields of entry types after opening a new or existing `bib` file, or the filled entries can be obtained from the web sources using Bib $\TeX$  export citation commands. For example: (1) With `http://www.sciencedirect.com` the selected articles, after searching, using the links "Export citations" and choosing "Export format" as "BibTeX format" and "Export"; (2) With `http://scholar.google.com` after setting the "Scholar Preferences" to "Show links to import citations into BibTeX" of "Bibliography Manager" and "Import into BibTeX" – will populate the entry codes that can be simply copied to the `bib` file. Other services such as Web of Science<sup>®</sup> and Scopus<sup>™</sup> usually available to students and staff also export citations in Bib $\TeX$  format. Although these methods are quicker and labor saving, some of the fields need to be added or corrected or some unsearchable items need to manually typed in. This Bib $\TeX$  file codes as well as the  $\LaTeX$  codes allow the document to be formatted automatically in variety of ways by selecting different bibliography styles and document class options, which is one of the greatest advantage of using  $\LaTeX$ .

### THESIS PREPARATION USING $\LaTeX$

Several established universities have developed their  $\LaTeX$  graduate thesis document class files with templates and made them available to students through "Graduate School Office" or through other departments. A sample of the thesis template of NDSU is shown in figure 13.

```

1 \documentclass[phd]{ndsuthesis}% other options: thesis, ma-thesis
2
3 \begin{document}
4 \title{<alternate title>}<title>
5 \name{<first.name>}<last.name>
6 \college{<college.name>}
7 \department{<dept.name>}
8 \professor{<prof.name>}
9 \date{<date>}
10 \coverpage[20mm]% white space between elements
11
12 \begin{abstract}
13 <abstract.text>
14 \end{abstract}
15 \section*{Acknowledgments}% unnumbered section
16 <acknowledgments.text>
17
18 \tableofcontents
19 \listoffigures
20 \listoftables
21
22 \mainbody% main matter formats loaded - Arabic numeral pages
23 \section[< toc.entry >]{< title >}% 1st level - e.g. Introduction
24 <section.text>
25 \subsection[< toc.entry >]{< title >}% 2nd level
26 <subsection.text>
27 \subsubsection[< toc.entry >]{< title >}% 3rd level
28 <subsubsection.text>
29
30 \section[< toc.entry >]{< title >}% 1st level - e.g. Materials
31 \subsection[< toc.entry >]{< title >}% 2nd level
32 \subsubsection[< toc.entry >]{< title >}% 3rd level
33 % . . .
34 % other sections, subsections, and subsubsections and their text
35 % . . .
36 \bibliographystyle{ndsuthesis}
37 \bibliography{<references.bib>}% user reference file
38
39 \appendix[< toc.entry >]{< title >}
40 <appendix.text>
41 \end{document}

```

Figure 13. NDSU thesis  $\LaTeX$  template.

The thesis template (fig. 13) is similar to that of Elsevier journal manuscript (fig. 3), but much simpler. The



three files used in the thesis template are document class (line 1: `ndsuthesis.cls`), bibliography style (line 36: `ndsuthesis.bst`), and the bibliography data file (line 37: `*.bib`) (NDSU, 2008). The template adopted from Calvo (2004) can be filled with details based on the supplied pointers as followed earlier (fig. 4). The optional arguments of “`< toc.entry >`” (e.g., line 23) are the short titles to be used in the “Table of Contents,” while the longer version used in the body text. Populating the various sections and details is based on the content of the thesis.

It is not uncommon to include the section text as files using commands such as `\input{introduction.tex}`. Including different sections as individual `*.tex` files of basic codes without any preamble is the common method utilized in the typesetting of large theses and books. This method of file inclusion and selective commenting allows for compiling only specific sections, useful while developing the document.

Figure 14 shows the output of a completed template with fictitious entries (not shown). As seen earlier, the codes in the template will create all the required formats (e.g., All uppercase letters for title and degree), spacing, and style based on the input fields that may be written in lower or title case. An example of a complete unpublished masters thesis using the NDSU L<sup>A</sup>T<sub>E</sub>X template can be found elsewhere (Braaten, 2005; Calvo, 2004). These thesis examples demonstrate the various capabilities and automatically generated items such as, table of contents, list of tables, list of figures, etc.

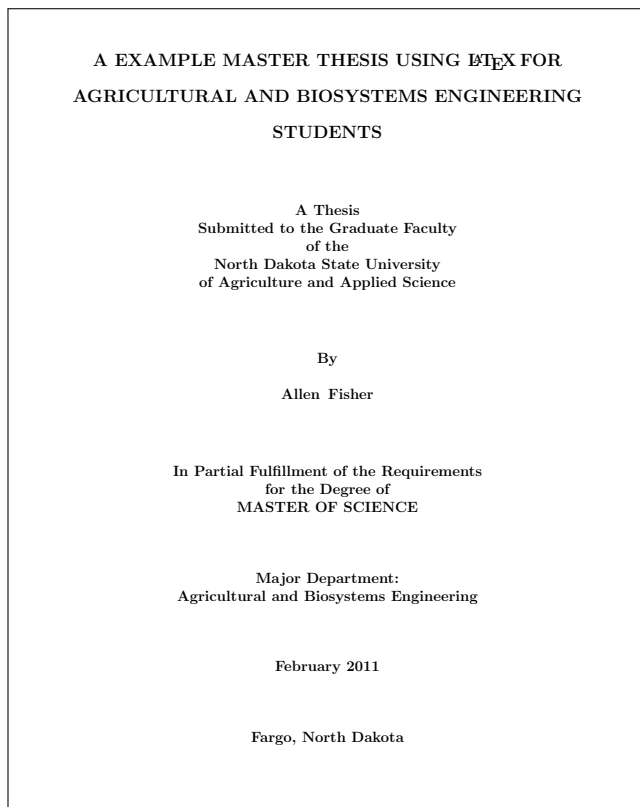


Figure 14. A sample cover page of NDSU Masters thesis using L<sup>A</sup>T<sub>E</sub>X template.

## OTHER COMMON FEATURES OF DOCUMENTS

The other features that are frequently encountered in technical documents are:

**Lists:** Lists can be numbered, unnumbered (bullets), and description (such as this) produced using ‘`enumerate`’, ‘`itemize`’, and ‘`description`’ environments. The individual items are coded using the `\item` command. Nesting of the environments will produce sub-levels.

**Tabs:** The ‘`tabbing`’ environment allows to set tabs to align text in columns. The commonly used commands are `\=` - set tab stops, `\>` - advance to next tab stop, `\kill` - setting tab without producing text, and `\|` - next line.

**Fonts:** Application of fonts can be divided into the following groups, with commands and examples shown in parenthesis, as family (e.g., `\rmfamily` ⇒ Roman, `\sffamily` ⇒ Sans serif, `\ttfamily` ⇒ Typewriter); series (e.g., `\mdseries` ⇒ Medium series, `\bfseries` ⇒ **Bold series**); shape (e.g., `\upshape` ⇒ Upright shape, `\itshape` ⇒ *Italic shape*, `\slshape` ⇒ *Slanted shape*, `\scshape` ⇒ SMALL CAPITALS SHAPE); and size (e.g., `\tiny` ⇒ tiny, `\scriptsize` ⇒ scriptsize, `\footnotesize` ⇒ footnotesize, `\small` ⇒ small, `\normalsize` ⇒ normalsize, `\large` ⇒ large, `\Large` ⇒ **Large**, and the other bigger sizes (examples not shown) are `\LARGE`, `\huge`, and `\Huge`). These commands can also be compounded and will be applied to the text that appears after these commands and enclosed in braces (e.g., `{\sffamily\tiny\itshape\bfseries ‘Example text’}` ⇒ *Example text*). It is also possible to load several other fonts as well as work with fonts of other languages with suitable packages and commands.

**Margins:** Using ‘`geometry`’ package, the various margins of the document can be easily set. For example, a direct command of `\usepackage[top=1.1in, bottom=1.25in, left=1.5in, right=1.2in]{geometry}` with options sets all the selected margins in the document.

**Line spacing:** The value (decimal) of the argument of `\baselinestretch` command will be used to alter the line spacing. For example, `\renewcommand{\baselinestretch}{2}` produces a double spacing. Alternatively, the ‘`setspace`’ package and `\singlespacing`, `\doublespacing`, or `\onehalfspacing` commands directly produce the corresponding line spacing.

**Alignment:** Within the scope the commands such as `\centering`, `\raggedright`, and `\raggedleft` are equivalent to ‘`center`’, ‘`flushleft`’, and ‘`flushright`’ environments, respectively.

**Package help:** Every package comes with its documentation usually in the form of `*.pdf` file, which explains all the possible usage and instructions along with the source codes. A web search will find them or could be directly obtained from CTAN (<http://www.ctan.org>).

# MERITS AND ISSUES OF USING L<sup>A</sup>T<sub>E</sub>X

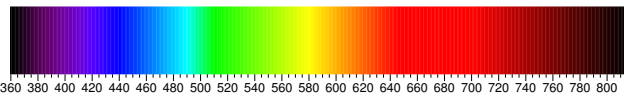
The advantages and the possible issues of using L<sup>A</sup>T<sub>E</sub>X as the system especially for preparing articles, thesis, and books from the view point of ABE students and professionals, both from beginners and advanced users, are listed and discussed.

## ADVANTAGES

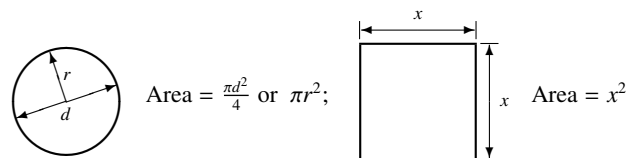
- L<sup>A</sup>T<sub>E</sub>X is an open source, free yet advanced, software that can be readily downloaded and installed easily in every type of computing system (Windows, Linux, Unix, DOS, and Mac OS). L<sup>A</sup>T<sub>E</sub>X is also a system that grows benefiting from the user-developed codes (classes, packages, and templates), and all these updates are again open source and free to use.
- L<sup>A</sup>T<sub>E</sub>X allows the user to concentrate on the content while L<sup>A</sup>T<sub>E</sub>X performs the consistent formatting. In other words, the author does the “writing” and the L<sup>A</sup>T<sub>E</sub>X compiler performs the consistent “formatting.” — In TPS, without a template different users will produce different outputs that lack consistency, but using a L<sup>A</sup>T<sub>E</sub>X class file, an essential argument of ‘`documentclass`,’ ensures consistency across users.
- L<sup>A</sup>T<sub>E</sub>X packs in the sound principles of professional typesetting while formatting the documents. This introduces the concept of “readability” of documents that takes care of features like number of words per line, their spacing, hyphenation, spacing of elements in reference to font size, ligature, etc. — Authors in general may not be aware of these principles of typesetting, and they go by “visual formatting” to their personal liking, sometimes violating the principles of typesetting, resulting in documents that lack consistency across authors.
- L<sup>A</sup>T<sub>E</sub>X automates and updates several aspects of document such as, table of contents (short and extended), list of tables, list of figures, index (multi-level), bibliography, nomenclature, glossary, among several other features. As L<sup>A</sup>T<sub>E</sub>X forces the users to follow the “structural” principles, automation of these features was possible and fully realized. — Although such automation was possible with some TPS, the users are mostly unaware or rarely use them. Hence, this opportunity is usually missed with TPS, but the benefits come natural with L<sup>A</sup>T<sub>E</sub>X as it is a “structural” language.
- L<sup>A</sup>T<sub>E</sub>X is an excellent choice of a document preparation system for technical theses, reports, and books. For thesis, some of the universities have developed their L<sup>A</sup>T<sub>E</sub>X document class and template files, and when utilized will create uniform feel for all the thesis prepared. — This uniform style among thesis is possible with other TPS as well through templates, yet the other automation benefits are not quite common with TPS.
- Moving document elements while revising the document that calls for updating the numbers of the cited elements (headings, equations, floats, table of contents, index, etc.) will be handled automatically. — This in the traditional manual way will be tedious and highly error prone.
- The user can have all the references in one place as a Bib<sub>L</sub>T<sub>E</sub>X (\*`.bib`) flat-file that can have several hundred entries, yet being ASCII the size will be quite small. For example, a 100 article entry, with 1757 lines of data, the size of the bib file is 68 KB, as opposed to the same content in TPS doc file is 192 KB. Such master bib file will serve as “Once Write and Read Many” mode of operation and can be subjected to several style formats directly. — Usually such automation with TPS may require additional commercial software (e.g., EndNote).
- The references will be automatically generated with proper format when appropriate style files were used. This avoids the classical error of uncited references and unlisted references, which eliminates the need for the reviewer to check for this unproductive and easily avoidable mistake from the authors. With some styles (e.g., `chicagoa.bst`) the reference items get sorted alphabetically. — This is a clear advantage over other manual document preparation systems.
- L<sup>A</sup>T<sub>E</sub>X measurements are very accurate and the smallest dimension it can recognize is ‘`sp`’ (scaled point) and 65536 sp make 1 pt that in turn  $\approx 0.351366666666667$  mm (Wikibook Contributors, 2008). Therefore the smallest dimension that is available in L<sup>A</sup>T<sub>E</sub>X for manipulating elements (e.g., moving and sizing) is approximately  $\frac{1}{186517}$  mm = 0.00536  $\mu$ m. — Such fine tuning of elements is unheard of with TPS in general.
- Users can generate the submission ready double-spaced “review” as well as two-column, double sided, single-spaced “final” formats of the paper from the same source by utilizing appropriate options in the document class (e.g., `elsarticle.cls`). — Usually with TPS the user have to create two different versions manually.
- It is possible to submit the source codes files (\*`.tex`) directly to the publishers (e.g., Elsevier Editorial System - EES) and generate the official submitted version (\*`.pdf`). This method subsequently allows for direct usage of the codes by the publishers during proofs production, without having to re-key or convert articles submitted using other TPS. Hence usage of L<sup>A</sup>T<sub>E</sub>X results in quicker production and fewer errors in typesetting.
- Journal articles that are camera ready, professionally typeset, journal-feel, compact (usually  $\leq 12$  pages), offprint like, easy to maintain, having better readability can be prepared using L<sup>A</sup>T<sub>E</sub>X document classes (\*`.cls`) and templates furnished by several journals. — However, with TPS the users usually end up with only the double-spaced version of the pre-submitted article (editable, very long,  $\approx 20$ –40 pages), and the official generated version of submitted article (pdf non-editable). The TPS tools either not capable or not encourage the authors to make outputs that resembles the final offprint, and they usually wait (sometimes for years) for the article to be finally published to see the paper in journal format.
- Several journals due to copyright restrictions will not allow to post the published versions of the articles on the websites of the authors; however, the journals allow posting the preprint version prepared by the author. With L<sup>A</sup>T<sub>E</sub>X as the system of document preparation, the user can produce an output that has the journal-feel and almost matches published article, which enhances the authors’ visibility and possible future citations from other readers.
- Advanced conditional formatting and handling of other features can be performed in L<sup>A</sup>T<sub>E</sub>X using the ‘`ifthen`’ package. For instance, the command `\ifthenelse{\boolean{@twocolumn}}{\linenumbers}` produces line numbers only when the document mode is single

column format (e.g., review format). This is similar to using the “If-Then-Else” statement frequently used in programming languages for conditional controls.

- Document annotation features such as **strikeout** (“~~deleted text~~”), **inserted** (“**newly added**”), and **highlighted** (“**deserves attention**”) text materials are incorporated using `\sout{text}`, `\textcolor{color}{text}`, and `\hl{text}` commands. To use these features, ‘ulem’, ‘color’, and ‘soul’ packages should be included. Footnotes, margin notes (shown in orange-red color), and end notes were also used to annotate the manuscript. It is equally possible to have these features in black & white and with shades of grey. These commands can be simplified by defining shortcuts.
- Using advanced conditional formatting, a single source code could produce the “Annotated” (color-coded revised version showing inserted and deleted text) and “Revised” (updated final) versions of the journals articles. It is a common practice in the peer-reviewed article publication process during revision to create such versions. This usually takes preparation of two different versions with usual TPS.
- With L<sup>A</sup>T<sub>E</sub>X book document class such as ‘memoir.cls,’ high quality books with several professional layouts can be prepared. For very large books L<sup>A</sup>T<sub>E</sub>X is the non-crashing reliable system available and offers several tasks automation.
- Advanced use of L<sup>A</sup>T<sub>E</sub>X allows for drawing figures through ‘pstricks’ or ‘eepic’ packages that offer extended capabilities and produces good quality vectorized (\*.eps files) mathematical, graphical, flowchart, and geometrical figures (Goossens et al., 2008). The method involves only codes prepared in ASCII text. — This drawing capability is available through specific drawing tools in other TPS. Shown here is a generated picture using simple codes in ‘picture’ environment using `\line`, and `\multiput` commands among others (Kern, 2007a,b; Mittelbach et al., 2004):



However, it is also possible to draw some simple pictures using ‘picture’ environment directly in L<sup>A</sup>T<sub>E</sub>X, but they were restricted in their range. Shown below are simple drawings that used `\circle`, `\vector`, and `\framebox` commands among others.



The above drawings are vector based and will retain their quality at any level of magnification.

- L<sup>A</sup>T<sub>E</sub>X can also be used to create conference posters (e.g., `a0poster.cls` and `sciposter.cls`) and presentation slides (eg. `beamer.cls` and `prosper.cls`) using appropriate classes and packages.
- Students could able to convert their thesis into a journal article with a few easy modifications, as the basic L<sup>A</sup>T<sub>E</sub>X code is the same irrespective of the documentclass or template.

- A knowledge of L<sup>A</sup>T<sub>E</sub>X forms an useful skill set to the students for pursuing an academic, research, or educational career. One can take advantage of the available various journal, book, curriculum vitae, report, and thesis styles provided by the publishers and other online sources.

## POSSIBLE ISSUES

- A steep learning curve is involved with L<sup>A</sup>T<sub>E</sub>X. However personnel with programming knowledge (e.g., ABE students) will find it easy to switch from any TPS. Such steep learning curve is always associated with any new programming language or software. Mastering and exploring several aspects of L<sup>A</sup>T<sub>E</sub>X definitely requires involvement from the users.
- Since L<sup>A</sup>T<sub>E</sub>X is an open source free software that is not a product of any commercial firm, there will not be an official support, but L<sup>A</sup>T<sub>E</sub>X purchased through commercial sources will. However, using the books, websites, mailing lists, and forums could solve most of the issues.
- Developing user-defined classes, templates, and packages will be quite complicated, as it requires knowledge of plain T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X codes. Although, it should be understood that with thorough knowledge and understanding of the existing codes, it is possible to develop them as they were added on regular basis by the developers throughout the world enriching the system. However, there is no need for a general user to venture in those areas, as most of the document preparation requirements could be addressed by employing the available sources.
- Debugging the codes needs some practice. As with any computer language, missing a symbol will stop the compilation with error messages or produce several errors. With experience, the users can able to decipher the error messages better and fix the codes easily.
- A reviewer or collaborating authors should know L<sup>A</sup>T<sub>E</sub>X in order to incorporate the suggestions and modify the document at source level. The “Track-changes” feature available with TPS is not directly available with L<sup>A</sup>T<sub>E</sub>X. However, this is not an issue for one with a working knowledge in L<sup>A</sup>T<sub>E</sub>X; as one can readily make annotations such as insert, delete, highlight, and comment the document with color codes and special notes (e.g., footnotes, margin notes, endnotes) as indicated earlier.
- Although “spelling check” facility is available with several L<sup>A</sup>T<sub>E</sub>X front-end editors, the “grammar checking” facility that is usually available with TPS is not available at present in L<sup>A</sup>T<sub>E</sub>X editors. This may not be an issue with native speakers of the language.
- Sometimes, especially when relied solely on online resources, there will be a lot of searching to find the right information to perform a particular task.

## CONCLUSIONS

L<sup>A</sup>T<sub>E</sub>X is an open source, free, high quality, document preparation typesetting system than can be easily adopted by Agricultural and Biological Engineering (ABE) students and professionals for thesis, journal articles, books, technical reports, and outreach materials. L<sup>A</sup>T<sub>E</sub>X handles mathematical material in an excellent manner and is the choice for large books, tech-

This is margin note shown in color.



nical reports, and thesis.  $\LaTeX$  automates several features of the document such as, table of contents; list of tables and figures; numbering chapters, sections, subsections, equations, tables, figures, etc.; cross-referencing items, generating index and glossary; and with  $\BibTeX$  automates references citing and references list generation from a common bibliography data file. Several  $\LaTeX$  document classes and templates, and bibliography styles supplied by various journals and other publishers can be taken advantage of to create professional quality consistent documents with a least effort.  $\LaTeX$  also opens up the possibilities of designing new document class and template files with unique styles and formats for future technical documents preparation. ABE students and professional having exposure in programming languages will find  $\LaTeX$  easy to learn and master. A knowledge of  $\LaTeX$  will serve continuously the students and professionals alike well in their academic and professional pursuits.

## ACKNOWLEDGMENTS

The author wishes to record his gratitude to his colleagues Dr. V. K. Ramesh and Dr. V. Ashok Kumar, Atmospheric Sciences, Indian Institute of Technology, Kharagpur, WB, India. The Northern Great Plains Research Laboratory (NG-PRL), USDA-ARS, Mandan, ND, US provided the facilities and this support was gratefully acknowledged.

## REFERENCES

AMS (2002). User's Guide for the amsmath Package. Version 2.0. Available at: <ftp://ftp.ams.org/ams/doc/amsmath/amslatex.pdf>. Accessed 6 February 2011.

Braaten, B. D. (2005). An Integral Equation Technique for Solving Electromagnetic Problems with Electrically Small and Electrically Large Regions. Unpublished PhD diss. Fargo, North Dakota: North Dakota State University, Department of Electrical and Computer Engineering. Available at: <http://www.ndsu.nodak.edu/ece/people/faculty/braaten/thesis.pdf>. Accessed 6 February 2011.

Calvo, J. A. (2004). Using The Ndsu Thesis Document Class for LaTeX. Unpublished PhD diss. Fargo, North Dakota: North Dakota State University, Department of Electrical and Computer Engineering. Available at: <http://www.math.ndsu.nodak.edu/resources/tex/ndsuthesis/mythesis.pdf>. Accessed 6 February 2011.

Chang, W. (2010).  $\LaTeX 2\epsilon$  Cheat Sheet. Available at: <http://www.stdout.org/winston/latex/latexsheet.pdf>. Accessed 6 February 2011.

Daly, P. W. (2010). Natural Sciences Citations and References (Author-Year and Numerical Schemes). Available at: <ftp://ftp.tex.ac.uk/tex-archive/macros/latex/contrib/natbib/natbib.pdf>. Accessed 6 February 2011.

Downes, M. (2002). Short Math Guide for  $\LaTeX$ . Version 1.09. American Mathematical Society. Available at: <ftp://ftp.ams.org/pub/tex/doc/amsmath/short-math-guide.pdf>. Accessed 6 February 2011.

Elsevier Ltd. (2009). elsarticle.cls A better way to format your submission. Available at: [http://www.elsevier.com/framework\\_authors/misc/elsdoc.pdf](http://www.elsevier.com/framework_authors/misc/elsdoc.pdf). Accessed 6 February 2011.

Goossens, M., S. Rahtz, and F. Mittelbach (2008). *The LATEX graphics companion*. Addison-Wesley Reading, MA.

Igathinathane, C., L. O. Pordesimo, E. P. Columbus, W. D. Batchelor, and S. Sokhansanj (2009). Sieveless particle size distribution analysis of particulate materials through computer

vision. *Computers and Electronics in Agriculture* 66(2), 147–158.

Kern, U. (2007a). Color extensions with the xcolor package - pstricks examples. v2.11. Available at: <http://ctan.math.utah.edu/ctan/tex-archive/macros/latex/contrib/xcolor/xcolor2.pdf>. Accessed 6 February 2011.

Kern, U. (2007b). Extending  $\LaTeX$ 's color facilities: the xcolor package. v2.11. Available at: <http://ftp.math.purdue.edu/mirrors/ctan.org/macros/latex/contrib/xcolor/xcolor.pdf>. Accessed 6 February 2011.

Knuth, D. E. (1984). *The  $\TeX$ book*. Addison-Wesley Professional, Reading, Massachusetts, US. ISBN: 0-201-13448-9.

Kopka, H. and P. W. Daly (2004). *A Guide to  $\LaTeX$ : and Electronic Publishing* (4th ed.). Addison-Wesley, Reading, MA. ISBN: 0-321-17385-6.

Krishnan, E. (2003).  $\LaTeX$  Tutorials - A Primer. Available at: <http://www.tug.org.in/tutorials.html>. Accessed 6 February 2011.

Lamport, L. (1994).  *$\LaTeX$ : A Document Preparation System* (2 ed.). Addison-Wesley, Reading, Massachusetts, US. ISBN: 0-201-52983-1.

Markey, N. (2009). Tame the BeaST - The B to X of  $\BibTeX$ . Available at: [http://www.ctan.org/tex-archive/info/bibtex/tamethebeast/ttb\\_en.pdf](http://www.ctan.org/tex-archive/info/bibtex/tamethebeast/ttb_en.pdf). Accessed 6 February 2011.

Mittelbach, F., M. Goossens, J. Braams, D. Carlisle, and C. Rowley (2004). *The  $\LaTeX$  companion*. Addison-Wesley, Reading, Massachusetts, US. ISBN: 0-201-36299-6.

NDSU (2008). LaTeX Document Class for NDSU Disquisitions. Department of Mathematics, North Dakota State University. Available at: <http://math.ndsu.nodak.edu/resources/tex/ndsuthesis/>. Accessed 6 February 2011.

Oetiker, T., H. Partl, I. Hyna, and E. Schlegl (2010). The Not So Short Introduction to  $\LaTeX 2\epsilon$ . Ver. 5.00; Available at: <http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf>. Accessed 6 February 2011.

Patashnik, O. (1988). *Bib $\TeX$ ing*. Online source. Available at: <http://bibtexml.sourceforge.net/btxdoc.pdf>. Accessed 6 February 2011.

Roberts, A. (2009). Getting to grips with LaTeX. Available at: <http://www.andy-roberts.net/misc/latex/index.html>. Accessed 6 February 2011.

Silmari Consultants (2009). The very short guide to typesetting with  $\LaTeX$ . Available at: <http://www.ctan.org/tex-archive/info/latex-veryshortguide/veryshortguide.pdf>. Accessed 6 February 2011.

Voß, H. (2010). Math mode – v. 2.47. Available at: <ftp://ctan.tug.org/tex-archive/info/math/voss/mathmode/Mathmode.pdf>. Accessed 6 February 2011.

Wikibook Contributors (2008).  $\LaTeX$ . Online source Wikibooks. Available at: <http://upload.wikimedia.org/wikipedia/commons/2/2d/LaTeX.pdf>. Accessed 6 February 2011.

Wilson, P. (2009). *The Memoir Class for Configurable Typesetting User Guide*. The Herries Press, Normandy Park, WA.