Getting Started with TeX, LaTeX, and TeXnicCenter

Goals

The goals for this project are quite modest. In short—you will learn just enough to get started learning about TeX and LaTex. By completing this assignment,

- you will learn what TeX, LaTeX and TeXnicCenter are,
- you will learn the basics steps for preparing a document using TeXnicCenter,
- you will learn a few formatting commands, and,
- you will receive a couple of pointers on what to do if you want to learn more.

Both TeX and LaTeX are extremely complicated. We could easily devote a course to each and only scratch their surfaces. However, you don't need to master these programs to use them productively. Most users learn just enough to meet their initial needs learning more as their needs change. This is probably the way you use most word processors, e.g., Microsoft Word.

Background

TeX is the brainchild of Donald Knuth. Knuth, a mathematician and computer scientist at Stanford University, was distressed at the appearance of a new edition of one of his books. He began what he thought would be a year-long project to create a computer typesetting system to address these shortcomings. This led to a series of projects over the next decade, including TeX.

TeX is a typesetting system specifically designed to produce extremely high-quality documents. It is especially useful for typesetting mathematics since formatting mathematics is particularly challenging. TeX's primary goal is uncompromising quality, and, as a consequence, TeX is extremely difficult to master. There are other programs that are much easier to use but none even approach the quality that can be achieved with TeX.

Because of the difficulty in using TeX, LaTeX was created. It provides a number of simplifications (basically macros) that make it easier to use. While LaTeX does not provide the versatility of TeX, the output is of comparable quality. And since LaTeX is built on top on TeX, it is possible (but not necessarily easy) to use TeX from within LaTeX for greater control. This isn't to say that LaTeX is particularly easy to use. It is just simpler than TeX.

The simplest way to get started is by looking at sample files. This is the approach we will take. You will initially want to look at two files: `small2e.tex` and `sample2e.tex`.

About TeXnicCenter

TeX and LaTeX are markup languages (like HTML). They are definitely *NOT* what you see is what you get (WYSIWYG) programs. To use TeX or LaTeX, you first need to create a source file for your document. This is a simple text file that contains the text of your document along with commands that describe how the document should be formatted. This file is then compiled to generate the image file for the document. This could be in any of a number of formats including PostScript (PS), Portable Document Format (PDF), or Device Independent (DVI) format. This output file can then be viewed or printed with the appropriate program, e.g., Adobe Acrobat Reader for PDF files.

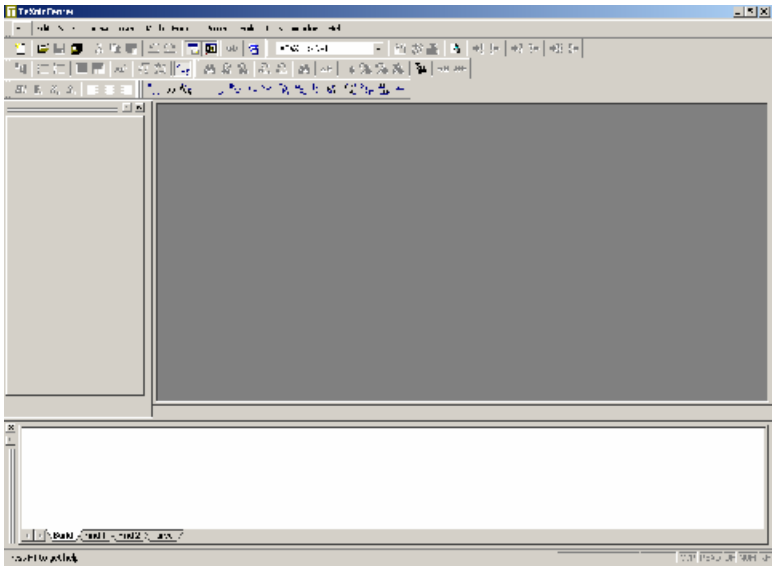
In completing this assignment, you will actually be working with several different programs. MiKTeX is the TeX compiler you will actually be using. However, you will not use it directly. TeXnicCenter provides a graphical user interface (GUI) to manage this process of creating, compiling, viewing, and printing TeX and LaTeX documents. It provides an interface to the other programs you will need, calling or forking to them as needed.

In particular, TeXnicCenter provides a GUI-based text editor that you can use to create the original source file for your document. The nice thing about this editor is that it provides a number of menus that can be used to embed TeX's markup commands. This means that you don't need to memorize TeX commands in order to use TeX. For example, if you need a particular mathematical symbol, you can find it on a menu, click on it, and the appropriate markup command will be automatically added to your document at the current insertion point (i.e., at the cursor).

Running TeXnicCenter

You can install TeX on your personal computer or you can use it in Olin 207, the computer lab. To start TeXnicCenter, from the ***Start*** menu select ***Programs→TeXnicCenter***

Normally, the initial screen should look something like this:



However, you may encounter three problems the first time you use it in the computer lab. First, you may need to close a "Tip of the day" screen. Next, the first time you use it, you may also be asked to configure TeXnicCenter. If you need to configure TeXnicCenter, you will need to tell TeXnicCenter where to find the TeX software. The location is **`C:\Program Files\MikTeX 2.5\miktex\bin`**. For everything else, just accept the defaults clicking NEXT and FINISH as needed.) Finally, in the lab, you'll need to work from the desktop, your H: drive, or from a removable drive. TeXnicCenter creates a number of working files and the lab computers are particularly about where new files can be created.

Creating new source files, opening existing files, and savings files is done in the normal manner. For example, to create a new file, select ***File→New***. An edit pane will open in the window.

The following text is from the file `small2e.tex`. (This file can be found in the directory **`c:\Program Files\MikTeX 2.5\latex\base\`** on the lab computers.) It provides a model for a simple LaTeX document. Locate and open this file on your system or open a new file and enter the following:

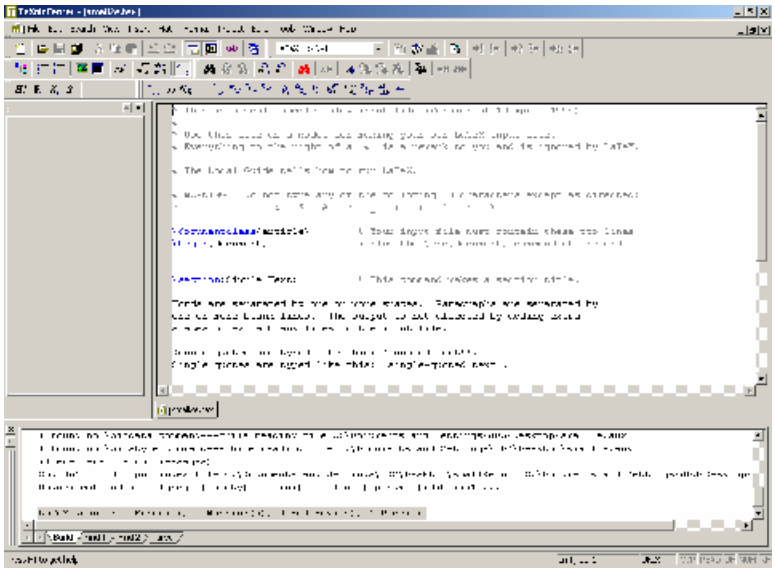
```
% This is a small sample LaTeX input file (Version of 10 April 1994)
%
% Use this file as a model for making your own LaTeX input file.
% Everything to the right of a  %  is a remark to you and is ignored by LaTeX.

% The Local Guide tells how to run LaTeX.

% WARNING!  Do not type any of the following 10 characters except as directed:
%               &   $   #   %   _   {   }   ^   ~   \
```

```
\documentclass{article}        % Your input file must contain these two lines
\begin{document}               % plus the \end{document} command at the end.


\section{Simple Text}          % This command makes a section title.

Words are separated by one or more spaces.  Paragraphs are separated by
one or more blank lines.  The output is not affected by adding extra
spaces or extra blank lines to the input file.

Double quotes are typed like this: ``quoted text''.
Single quotes are typed like this: `single-quoted text'.

Long dashes are typed as three dash characters---like this.

Emphasized text is typed like this: \emph{this is emphasized}.
Bold       text is typed like this: \textbf{this is bold}.

\subsection{A Warning or Two}  % This command makes a subsection title.

If you get too much space after a mid-sentence period---abbreviations
like etc.\ are the common culprits)---then type a backslash followed by
a space after the period, as in this sentence.

Remember, don't type the 10 special characters (such as dollar sign and
backslash) except as directed!  The following seven are printed by
typing a backslash in front of them:  \$  \&  \#  \%  \_  \{  and  \}.
The manual tells how to make other symbols.

\end{document}                 % The input file ends with this command.
```

This file illustrates the basic layout of a TeX or LaTeX file. Lines beginning with a percent symbol are comments. Words beginning with a backslash are markup commands. Curly brackets are used to group information. Percent symbols, backslashes, and curly brackets are part of a small set of reserved characters that have special meanings in TeX and LaTeX. If you need to use these in a file, you need to escape them.

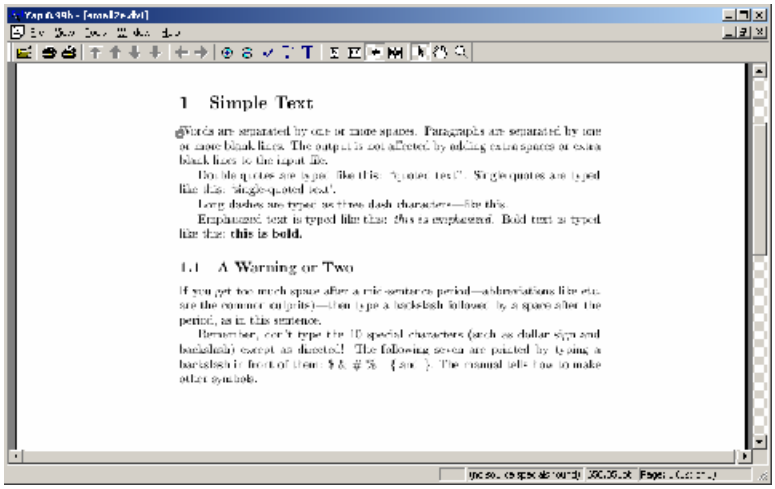
Copy this file to your desktop or disc before you proceed.

Once you have the document, you compile it. First, make sure ***LaTEX => PDF*** is displayed in the dropdown box on the second line. (You use this box to switch between TeX and LaTeX and to select the output format such as PDF or PS.) Next select ***Build → Current File → Build***. Any errors will be displayed in the pane at the bottom of the window. Your screen should look something like this:



(If you encounter any errors, double-click on the error in the error pane and the cursor in the edit pane will relocate to the corresponding source code correctly for most errors.)

To view the output, select ***Build → View Output***. A new window will open that looks something like this:



Notice that from this window, you can also print the document for this window. *Print both the source and output windows to turn in.* Compare the source with the output until you understand what each command in the source file does.

Repeat this with the file `sample2e.tex`. This is a large file showing more features. You do not need to turn in anything for `sample2e.tex`, but you should review the material in this file before moving on to the next section.

Typesetting Mathematics

Traditionally (i.e., before TeX), typesetting mathematics has been extremely difficult, time-consuming, and costly. Producing esthetically pleasing mathematics requires using a number of different font sizes and symbols as well as considerable attention to spacing, etc. TeX and LaTeX provide a rich symbol set and will automatically do much of the spacing, or will allow you to take control of the spacing when necessary.

Mathematical expressions are enclosed between dollar signs or between escaped parentheses. For example, if LaTeX encounters *a^b* or *\(a^b)*, it will format *a^b* using its rules for mathematical typesetting. Specifically, *a* and *b* will be in italics, *b* will be a superscript for *a*, and *b* will be displayed in a small font size.

Enter, compile, and print the output for the following file:

```
% This is a small sample LaTeX input file
%
\documentclass{article}
\begin{document}
\parskip = 1pc                %change spacing between paragraphs
\parindent = 0pc              %change paragraph indentation
\section{Math Formats}

This is a simple file to demonstrate typsetting mathematics with \LaTeX.
\TeX and \LaTeX support two modes for displaying mathematics, inline
expressions and display-style expressions.
These are described in the next two subsections.

\subsection{Inline Mathematics}
With inline mathematics, the mathematical expressions are displayed within
```

```
the body of the text.
For example, if I am talking about the discriminate for a quadratic equation,
i.e. \(b^2-4ac\),
I'll probably want to include the expression in the sentence.
Inline mathematics is enclosed between two \$'s or between a slash left
parenthesis and a slash right parenthesis.
\vskip 1pc

Here are some more examples.  I can use logical expressions such as $\neg x$ or
$A \wedge B$, algebraic expressions such $x^{n+1}$, or set notations such as
$A \subseteq B$ or $x \notin A$.

\subsection{Display-style Mathematics}
Display-style mathematics expects the equations or expressions to be
set off from the body of the text.
For example, the quadratic formula in display-style,
$$ x = {-b \pm \sqrt{b^2 - 4ac} \over 2a},$$
is too complicated to fit into the body of the text.
Display-style mathematics is enclosed between pairs of \$\$'s.

Display-style is useful for many things.
Here is an example using summation notation:
$$ \sum_{i=0}^{n-1} a^i = {{a^n-1} \over a-1} $$
Here is an example with a matrix:
$$
I_{3\times 3} = \left| \begin{array}{clcr}
1& 0& 0\\
0& 1& 0\\
0& 0& 1\\
\end{array}
\right|
$$

\end{document}
```

Using what you have learned, create a LaTeX source file that will generate the attached file. You'll probably want to use what you have just done as a template, but be careful not to overwrite the original file in case you want to come back to it.

What to Turn In

You should turn in:

a) the source listing and the output for `small2e.tex`,
b) the output for the file in the previous section, and,
c) both the source and output for your version for the attached document.

Due: Monday, October 9 at the start of class.

Where to Go From Here

If this assignment has whetted your appetite to learn more, you'll probably want to load the software on your computer and invest in a book. All the software we have been using is freely available and can be downloaded over the Internet at no cost. There are also a number of tutorials available over the Internet.

As for books, Leslie Lamport's *LaTeX: A Document Presentation System* is probably the best place to start with LaTeX. If you are ambitious enough to delve into TeX, the definitive treatment is Knuth's *TeXbook*. (Knuth also has published the source code for TeX so be sure you get the right book.) Other books are available but I'm not familiar with them.