

Getting Started with L^AT_EX on Microsoft[®] Windows[®]

Nathaniel Derby

<http://www.nderby.org>

July 30, 2008

ABSTRACT

L^AT_EX is a very powerful typographical software which, although used primarily for including mathematical formulae, has a much wider area of application. This is a guide intended to set someone up with L^AT_EX who has previously never had experience with it.

This is *not* intended to teach someone how to code with L^AT_EX. There are better references for that, to which the reader is referred. Rather, this document is strictly to show how to set up and run L^AT_EX on a Windows[®] computer.

Keywords: LaTeX.

TABLE OF CONTENTS

Intro: What is L^AT_EX, and why is it important?	2
Installing L^AT_EX on Windows	3
L ^A T _E X Itself: MiKTeX	3
Introduction	3
Downloading MiKTeX installation files with the Net Installer	5
Installing MiKTeX with the Net Installer	6
L ^A T _E X Shell: WinEdt	10
Introduction	10
Downloading WinEdt	10
Installing WinEdt	10
PDF Reader: Acrobat Lite	15
PS Reader: GSView	15
Using L^AT_EX	15
References	18
Acknowledgments	18

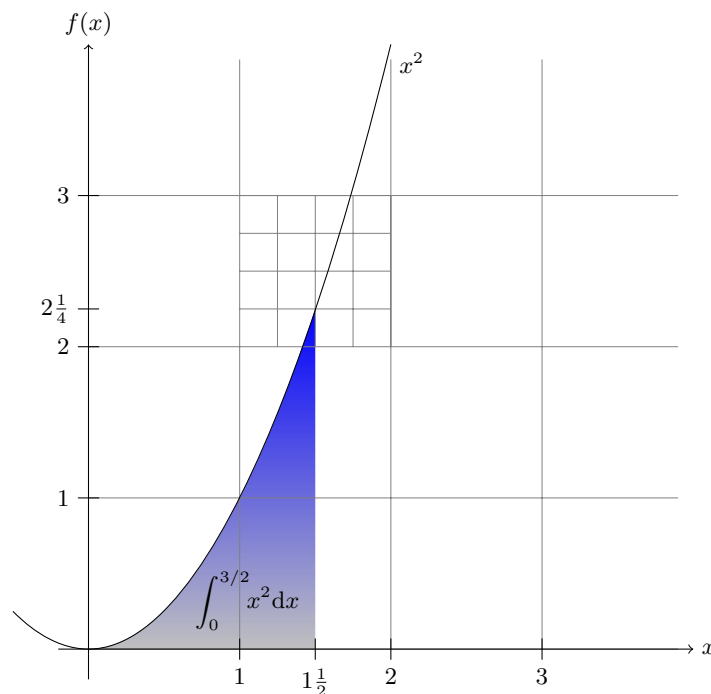


Figure 1: A graph produced by LATEX, from Tantau (2008, p. 375).

INTRO: WHAT IS LATEX, AND WHY IS IT IMPORTANT?

LATEX is a typographical software which can produce PDF (or other) files. It is arguably best known for producing high-quality mathematical output such as

$$\int_0^{3/2} x^2 dx = \left[\frac{x^3}{3} \right]_0^{3/2} = \frac{9}{8} \quad (1)$$

or

$$Y_t^{(1)} \equiv \sum_{u=1}^t \varepsilon_u = \begin{cases} \varepsilon_1 & t = 1 \\ Y_{t-1}^{(1)} + \varepsilon_t & t > 1 \end{cases} \quad (2)$$

or even the graphic shown in Figure 1. While these are compelling reasons for people who write about mathematics to switch to LATEX, this program has a much wider area of application. For example, notice that the equations and Figure are numbered, and that we can refer to them within the text as Equations (1) and (2), or Figure 1. LATEX internally keeps track of these numbers, so that if these numbers change later on (e.g., if another Figure is added later on before this one), these new numbers will be reflected in the document.

There are other reasons for using LATEX:

- *Typography Issues*: LATEX has superior typographical issues compared to MS Word, as shown in Figure 2 and explained in Taraborelli (2008). These include *Kerning* (when one letter can go insider “the space” of another, as in the first two letters of the word “Table”), *real small caps* (when the small caps are not just reduced-sized versions of the regularly-sized ones) and *ligatures* (when one letter flows into another.¹)
- *Table of Contents*: LATEX automatically creates a table of contents, such as the one shown at the beginning of this document.
- *PDF Hyperlinks*: LATEX can easily create PDF hyperlinks that link either to another part of the document (such as in the references to figures, footnotes, or bibliographic entries) or to a website (such as at the end of many bibliographic entries), as illustrated in this document.
- *Bibliographic Entries*: LATEX can automatically create custom-formatted bibliographic entries and references to them. In this document, for example, LATEX uses the format of the author’s last name, followed by the publication year in parentheses (e.g., Taraborelli (2008)), automatically adding a letter to the year for multiple publications by the same author within a given year (e.g., Derby (2008a) and Derby (2008b)).

¹However, ligatures are not present in the font used in this document.

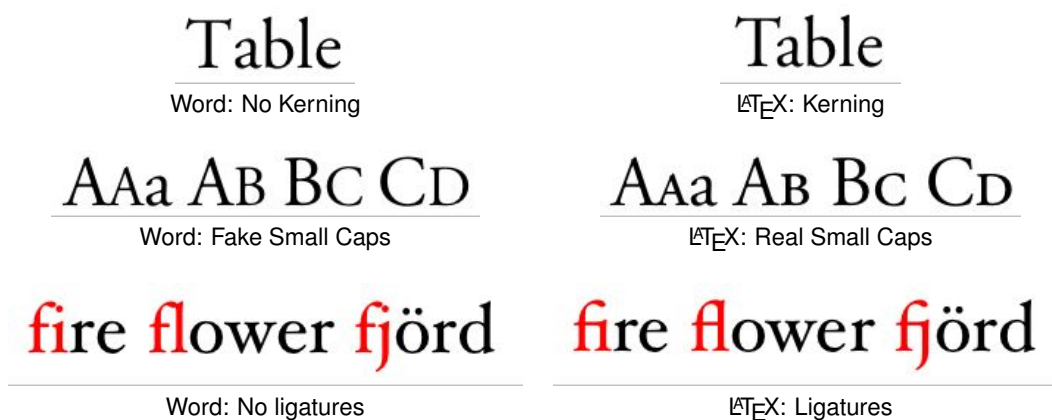


Figure 2: Comparisons between Word (left) and \LaTeX (right), as explained in Taraborelli (2008).

- *Numbering*: Equations, figures, footnotes, tables, and anything else that could be numbered are kept track of by \LaTeX . Thus, if you insert a new figure between Figures 1 and 2, the new figure is given the number 2 and the ones afterwards are renumbered appropriately. Furthermore, references to these figures are renumbered appropriately as well.
- *Spacing Issues*: \LaTeX automatically justifies and hyphenates words within a paragraph, puts extra spaces between paragraphs, and indents.²

For all these reasons (and more, as shown in Wierda (2008)), \LaTeX is a *typographical software* which is even used for printing books, which distinguishes it from word processing software like MS Word or OpenOffice.org. It is often used just in academic settings, but it can effectively be used in many other settings, as illustrated in Figure 3.

INSTALLING \LaTeX ON WINDOWS

One of the reasons \LaTeX is difficult to use at first is the fact that it is not one piece of software that you can just install. On Windows, we must actually install at least three different pieces of software:

1. \LaTeX itself. While it is very nice that \LaTeX is free, one hurdle for the beginning \LaTeX user is that there are several variants of it, such as MiKTeX, LyX, and PCTeX.
2. A \LaTeX shell. Some \LaTeX distributions, like PCTeX, come with their own shell – but others (like MiKTeX or LyX) do not.
3. A PDF viewer. For most of us, the free and popular Acrobat Reader is sufficient.
4. (Optional) A PS reader, like GhostView. Over the past few years, PDF has replaced PS for the format of most technical documents, but there are still some work done in PS. \LaTeX can produce both – but we will need a PS viewer to read the output.

The reader is not discouraged from trying out some of the different variants of \LaTeX , but this document will describe how to install one variant of \LaTeX and one shell, both of which the author has used for ten years and which are highly reliable, flexible, and updatable. This is not to say that any of the other variants are less desirable or effective. This is simply what the author has happily used for many years.

\LaTeX ITSELF: MIKTEX

Introduction

MiKTeX, available at miktex.org, is a free distribution of \LaTeX for Windows which is regularly updated. Note that \LaTeX isn't just a piece of software by itself; it is also many *packages*, which are macros which work within \LaTeX for certain purposes. For example, one package makes the slide shown as shown in the lower right of Figure 3, while another makes the fonts and spacing for this document. MiKTeX is a distribution of not just \LaTeX itself, but also of (almost) all widely-distributed packages which go with it.

²For this document, the indentation is set to zero.

RATE UPDATE TOOL

STAT/MATH 394 A
Derby

Handout 5
Last edited: 7/18/08, 7:04am

Functions of a Continuous Random Variable

Nathaniel B. Derby

Statistical Modeling
Consumer Deposit Products Group

October 23, 2007

Theory

Given a continuous random variable X , finding the pdf of a 1-1 function $Y = g(X)$ follows a straightforward process:

- Find

$$\begin{aligned} F_Y(y) &= P(Y \leq y) \\ &= P(g(X) \leq y) \\ &= \begin{cases} P(X \leq g^{-1}(y)) & g(\cdot) \text{ is increasing} \\ P(X \geq g^{-1}(y)) & g(\cdot) \text{ is decreasing} \end{cases} \\ &= \begin{cases} P(X \leq g^{-1}(y)) & g(\cdot) \text{ is increasing} \\ 1 - P(X < g^{-1}(y)) & g(\cdot) \text{ is decreasing} \end{cases} \\ &= \begin{cases} F_X(g^{-1}(y)) & g(\cdot) \text{ is increasing} \\ 1 - F_X(g^{-1}(y)) & g(\cdot) \text{ is decreasing} \end{cases} \end{aligned}$$

This document describes the program `Rate Update Tool.sas`, its component macros, and other related files. It is intended for the SAS programmer who either runs or modifies these programs. The reader who just wants to run the code can read just the **Directory Structure** and **Running the Code** sections.

This tool and all related files are in the directory

Z:\Statistical Modeling\Projects\Rate Updates

as shown in Figure 1. This documentation is in the `Documentation` subdirectory of this directory.

Academic Manuscript

Business Manuscript

December 2006

Happy Holidays!

This year has been one of settling down for me, and of adventure for Charles. Last year at this time I was unemployed after quitting my job at T-Mobile. Since March of this year, I have been happily employed as a statistician at the headquarters of Washington Mutual in downtown Seattle – within walking distance of our apartment. At 33 years of age, this is the first job I've ever had with full benefits – a sign that I've definitely been in school way too long! This job is great not just because it involves many of my strengths (e.g., SAS programming, time series analysis, business statistics), but also because I have a boss who gives me almost complete control over my work. For one thing, he's a business manager and not a "math guy", and so almost always defers to my judgement about mathematical or computing matters. For another, he gives me lots of time flexibility, allowing me to go to the university for lectures, to work from home from time to time, and even (see below) to work from Germany for a week via my laptop. Most importantly, though, he gives me ample time to think



The view from the top of my office building.

Personal Letter



Guidelines for Organizing SAS Project Files

Nathaniel Derby

Statist Pro Data Analytics
Seattle, WA, USA

SAS Global Forum, 2008



Nathaniel Derby Organizing SAS Files 1 / 7

Slide Show

Figure 3: L^AT_EX used in different situations.

The MiKTeX installer files can be downloaded from that website, or (for students of mine who don't have that good of an internet connection) it can be provided by me on a CD. There are two kinds of installations available from MiKTeX:

- *The Basic Installer*: This is a relatively small batch of files which can easily be downloaded from a computer with a decent internet connection. It is small because most of the accompanying packages are not included. Rather, they are downloaded and installed as they are needed. This may be ideal for many situations, but it is clearly not appropriate for any computer (such as a laptop) which may not always be connected to the internet.
- *The Net Installer*: This includes the basic L^AT_EX installation files and all the packages. As such, it is an extremely large collection of files, and even for a fast internet connection, it can take hours to download all the files. The advantage, of course, is that it includes *all* of the packages.

My personal preference is to use the net installer for the complete MiKTeX files, for three reasons:

- If you use two or more computers (as I do), you would probably prefer to have the same L^AT_EX builds and packages set up at every computer. MiKTeX is updated constantly, which is generally a very good thing, but it can also mean that a package that works one way on one computer will do something a little different at another one, just because the L^AT_EX installations were done at different times (resulting in slightly different packages). I have never had an incident where this has happened – but the potential is there.
- Every now and then, the newest version of MiKTeX has had problems. This *is* an incident that has happened to me – it's rare, but it can happen and it has happened.

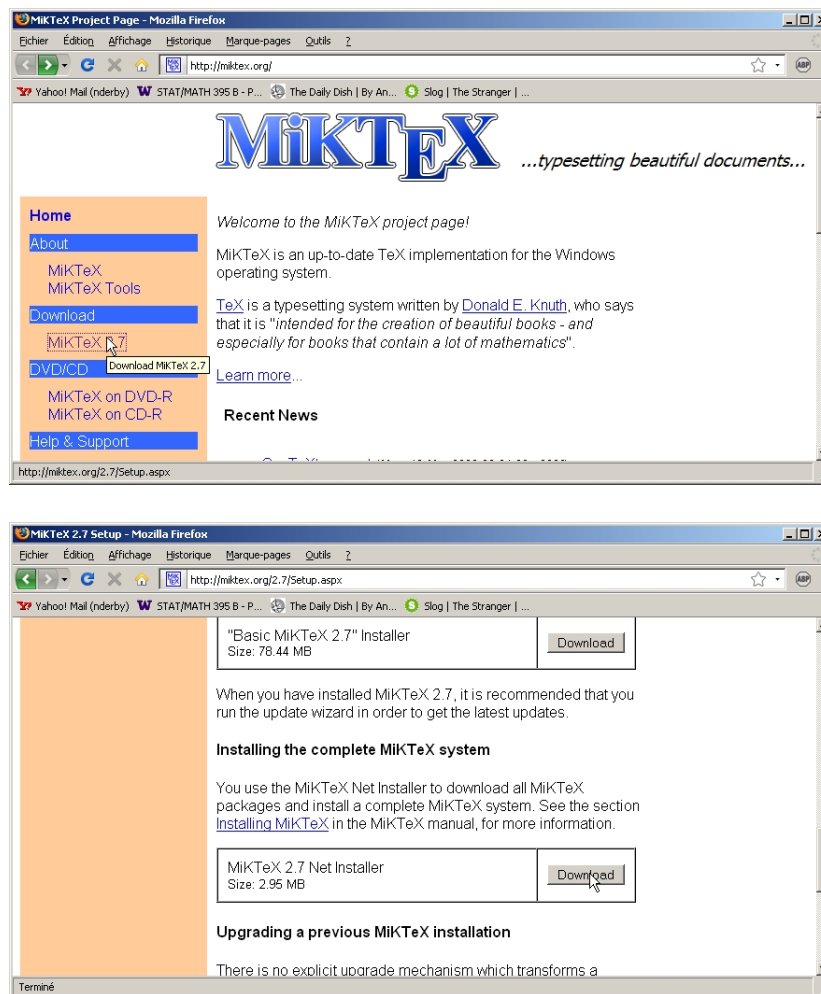


Figure 4: Downloading the net installer from `miktex.org`.

- MiKTeX will install packages as they are called from your \LaTeX code. With the net installer, the installation files are called from your computer, whereas on the basic installer, they are called from the internet. I would much rather have those files on my computer, to save time (downloading from the internet can take a little time) and to avoid a situation where, for whatever reason, I am not connected to the internet.

Clearly there has to be a balance between using a net installer from a specific point in time and using an updated one. For me, I choose to update my net installer every year or two. If you choose to be more cutting edge, you may choose to update yours every few months. However, it is possible to update certain packages on your own, outside of MiKTeX.

Once again, it should be noted that the net installer takes up much more room on a computer than the basic installer. If you have a computer with a small hard drive, it would probably be *not* recommended to use the net installer. However, most modern computers have considerable space, so this is not a problem.

Downloading MiKTeX installation files with the Net Installer

Before we can install MiKTeX with the net installer, we must first get the installation files – either from the website (`miktex.org`) or from Nate on a CD. If you have the installation files already, you can skip to the next section.

To get the installation files, first go to `miktex.org` and download the net installer, as shown in Figure 4. This will download the file `setup-x.x.xxxx.exe`, where `x.x.xxxx` is the version number (as of now (7/28/07), the current version is 2.7.3092). This file by itself is very small (2.95 MB), and so is easy to download. Once you download it, double click it. The installer will then ask various questions, as shown in Figure 5(a)-(f). Indicate your choices, and then the download will begin. The steps shown in Figure 5 are detailed below:

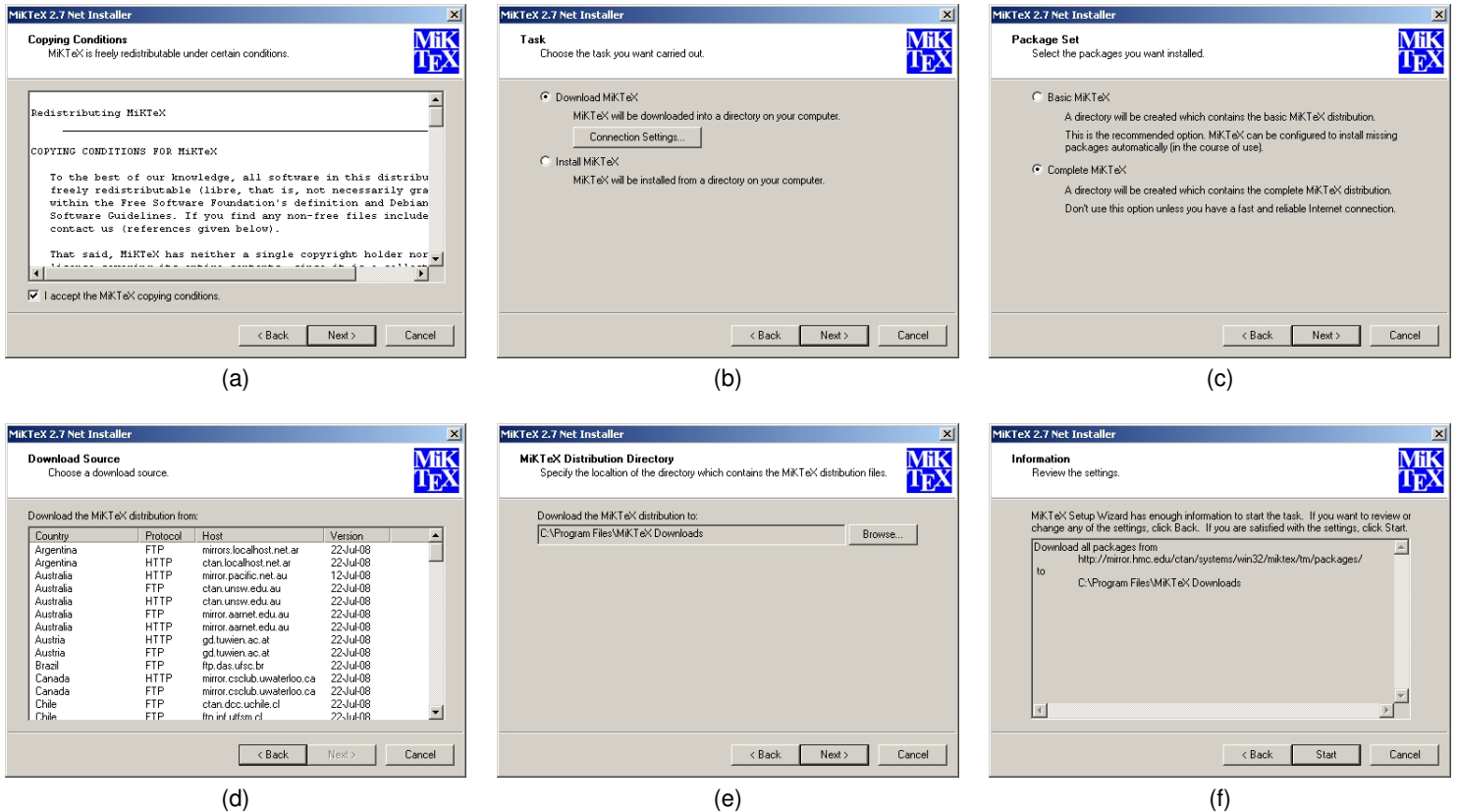
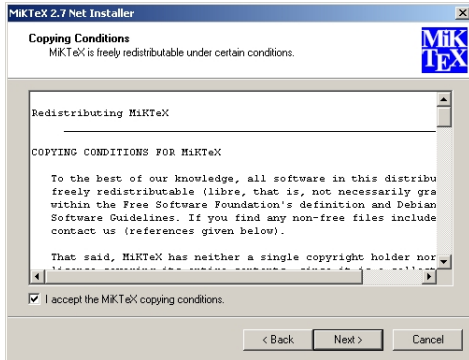


Figure 5: Downloading the installation files from the MiKTeX net installer.

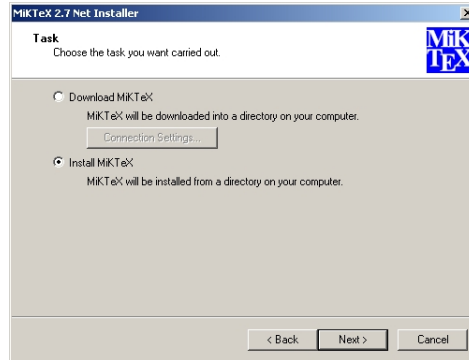
- Make sure you accept the conditions for redistributing MiKTeX (by copying it onto your computer, you are technically redistributing it).
- You would like to download the MiKTeX installation files, rather than install them (which we'll do in the next section).
- You may decide to download installation files for either the *basic MiKTeX* (the main, mostly-used files and packages) or the *complete MiKTeX* (all possible files and packages). My own personal preference is to download the complete MiKTeX files, for reasons discussed in the previous section. Note, however, that this option will require (much) more time to download the files, even with a fast internet connection.
- Decide on a *download source*. This is a choice of a server which should be close to you. MiKTeX is created and maintained by one person (Christian Schenk) in Germany. If everyone downloaded MiKTeX from his server, it would be overwhelmed. Therefore, there are many *mirror sites* around the world which provide the same files as his in Germany. Choose one that is not too far from home.
- Specify the directory where you wish to put your MiKTeX files. While technically this can be anywhere on the machine, you should ideally choose a location that makes logical sense, as *these files will remain on and be accessed by your machine for the duration of your MiKTeX installation*. As explained in detail in the next section, instead of installing the complete MiKTeX all at once, it is usual (and my own personal preference) to instead install files as you need them from this repository. Thus, I have my installation files stored in the directory `C:\Program Files\MiKTeX Downloads` (and the installation itself stored in `C:\Program Files\MiKTeX x.x`, where `x.x` is the main version number).
- The MiKTeX installer then reviews the information you entered, as a last check. If this is all correct, click on **START** to begin the download.

Installing MiKTeX with the Net Installer

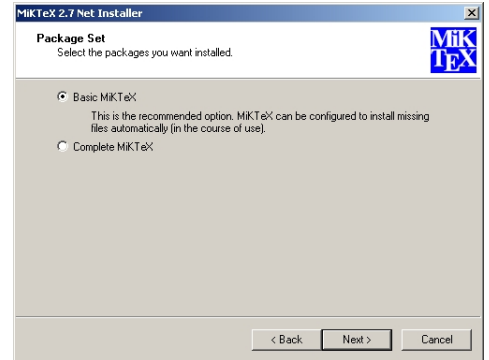
Now that you have downloaded all the installation files, we can begin actually installing MiKTeX – which, again, is \LaTeX , plus many related packages. To start with, we once again double click on the file `setup-x.x.xxxx.exe`. This begins the MiKTeX installer once again, as before, only this time, we wish to install files rather than download them. Figure 6(a)-(i) shows this progression, as detailed here:



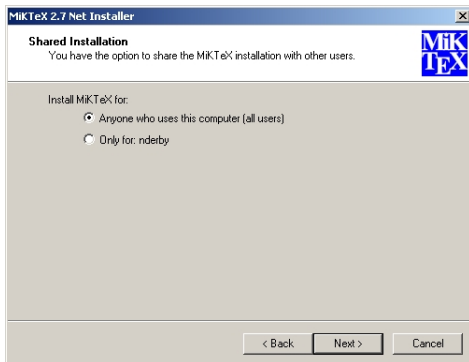
(a)



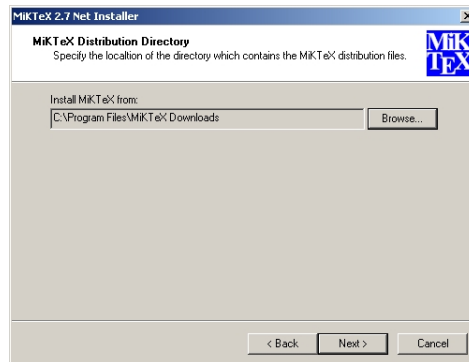
(b)



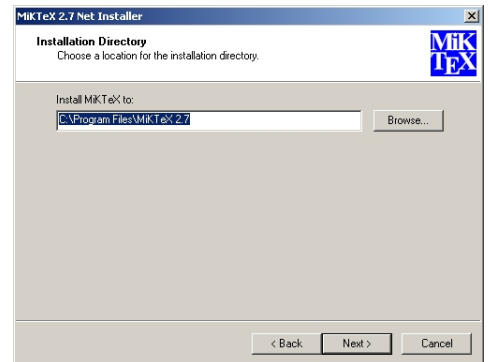
(c)



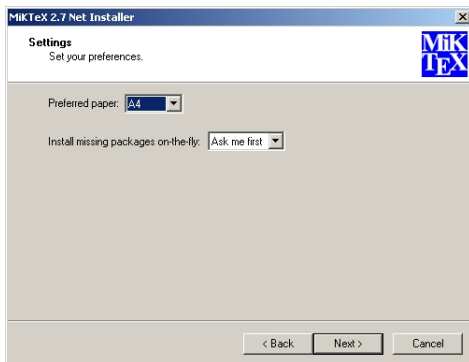
(d)



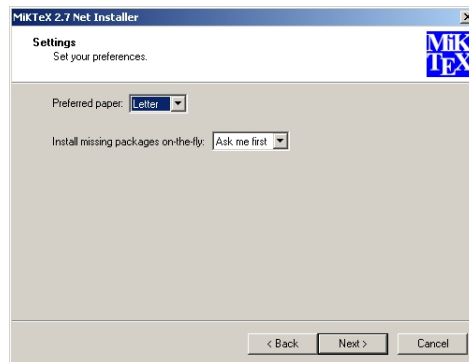
(e)



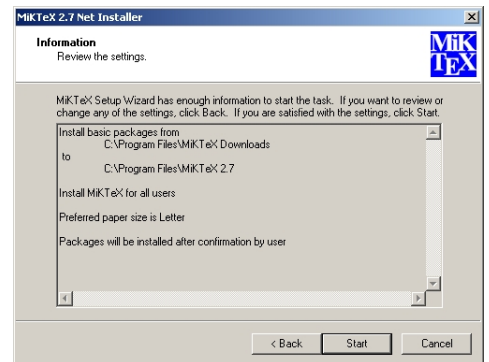
(f)



(g)



(h)



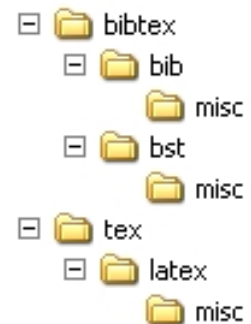
(i)

Figure 6: Installing MiKTeX from the net installer. For US users, the default paper size must be changed from A4 to Letter (g-h).

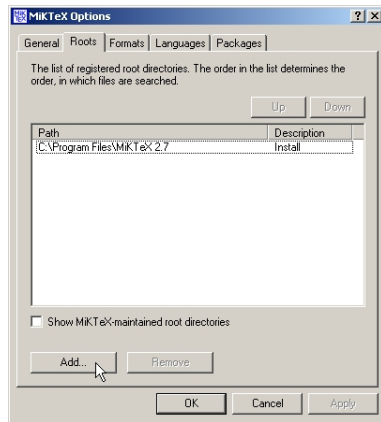
- (a) Make sure you accept the conditions for redistributing MiKTeX (by copying it onto your computer, you are technically redistributing it).
- (b) You would like to install MiKTeX, rather than download the installation files (which we did in the previous section).
- (c) You can either install the *basic MiKTeX* (the main, mostly-used files and packages) or the *complete MiKTeX* (all possible files and packages). My own personal preference is to install the basic MiKTeX files and to have additional packages and files be installed as needed.
- (d) Would you like to install it just for yourself or for anyone on the computer in question? Normally you would probably like to install it for anyone. However, if you happen to be on a computer for which you do not have the admin rights (e.g., a work computer), you can do it by installing it only for yourself.
- (e) Specify the directory where your installation files are found. If you downloaded the files from `miktex.org`, this should be the same location as shown in Figure 5(e). If you are installing files from a CD, you should probably transfer those installation files onto a local directory on your computer, since *these files will likely be accessed by your machine for the duration of your MiKTeX installation*. This is especially important if in step (c) you chose to install the basic MiKTeX and will install other files as needed. While technically this directory can be anywhere on the machine, you should ideally choose a location that makes logical sense. I have my installation files stored in the directory `C:\Program Files\MiKTeX Downloads`, which is close to my MiKTeX installation directory as shown in the next step.
- (f) Specify the directory where you wish to have MiKTeX installed in. I choose the default of `C:\Program Files\MiKTeX x.x`, where `x.x` is the main version number.
- (g-h) Two options are what your preferred paper size is and whether you want to be asked first before additional packages are installed. Since MiKTeX comes from Germany, the default paper size is the European one (A4), which must be changed to LETTER for US users. I keep the default of being notified when an additional package is installed, just to keep informed about what my computer is doing. An example of this will be shown later in this document, after we install all relevant software and start using \LaTeX .
- (i) The MiKTeX installer then reviews the information you entered, as a last check. If this is all correct, click on START to begin the download.

After MiKTeX is installed, we must now modify some settings. Click on `START → PROGRAMS → MIKTEX x.x3 → SETTINGS` to do this. We shall do the following, as illustrated in Figure 7(a)-(i):

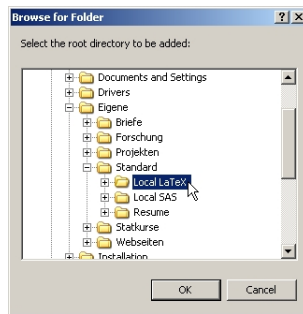
- (a) Under the ROOTS tab, we have a directory where \LaTeX files and packages are to be stored. We would like to add a *local directory* here, where we can put our own \LaTeX files and packages later on. This can be useful, for example, when we wish to access our own macros and/or bibliography files from many different \LaTeX source documents. We shall do this now and use it later. Click on ADD.
- (b-c) When adding a directory, we will need to create a directory where we will put our \LaTeX files. On my machine it is `C:\Eigene\Standard\Local LaTeX`, but yours will most likely be different. Such a directory must have a certain structure, as detailed at docs.miktex.org/manual/localadditions.html#id563052. For simplicity, simply copy the structure shown in Figure 7(c) and enlarged to the right. That is, `bibtex\bib\misc` will contain bibliography files, `bibtex\bst\misc` will contain files about how to format your bibliography, and `tex\latex\misc` will contain any other files. All of these will be described in later versions of this document. For now, we can just set up these directories.
- (d-e) Once we have added our local directory, we should move it up in our *path order*. That is, we want \LaTeX to first look in our local directory for files before looking through the general MiKTeX files. This way, we can redefine or modify existing \LaTeX files with the same name and be sure that MiKTeX follows our files and not its own ones.
- (f) Once we have defined our local directory and moved it up in our path order, we must refresh the file name database to make sure that MiKTeX knows which files are in our local directory. *This must be done every time we add a file to this directory*. Otherwise MiKTeX won't know that it is there. Note that this step is *not* necessary when we merely edit a file that MiKTeX already knows is there. This is done to inform MiKTeX of a new file – not to inform it that an existing file has changed.
- (g) The last thing that must be done is to inform MiKTeX that when a new package or file is to be installed, we want it to look through our local directory of installation files (i.e., that defined in Figure 6(e)) and not download it from the internet (which could take more time and/or result in a file that may not work optimally with our current installation of MiKTeX). We do this by first going to the PACKAGES tab and clicking on CHANGE.



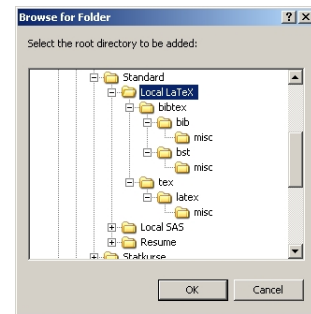
³where `x.x` is the main version number.



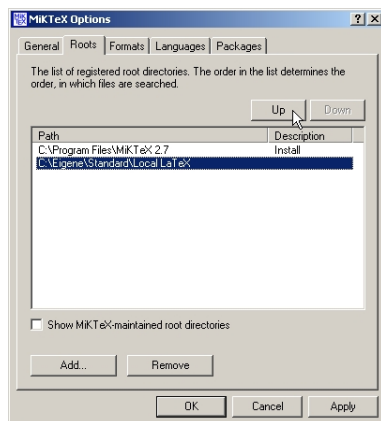
(a)



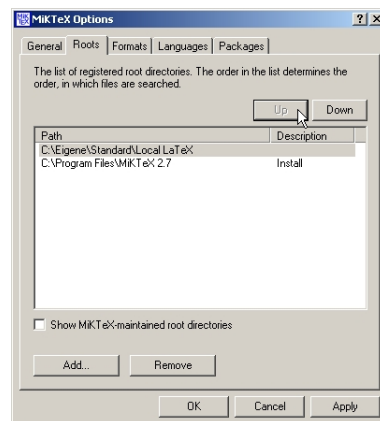
(b)



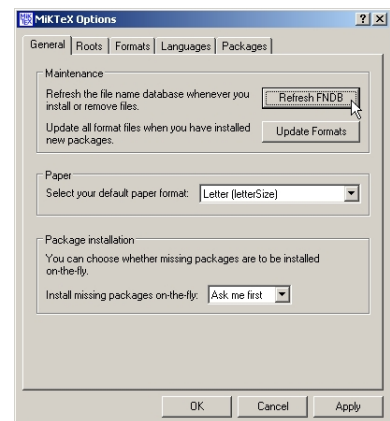
(c)



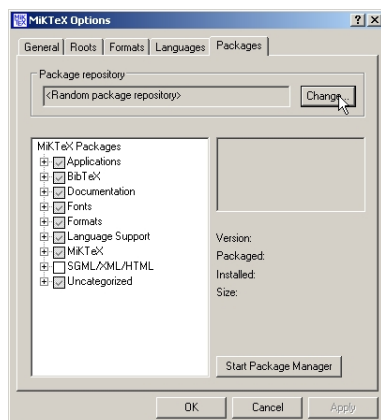
(d)



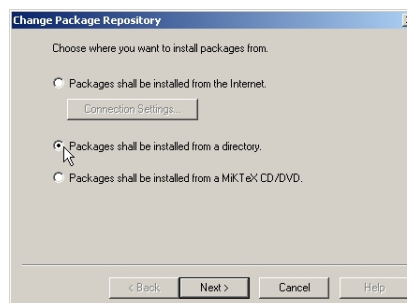
(e)



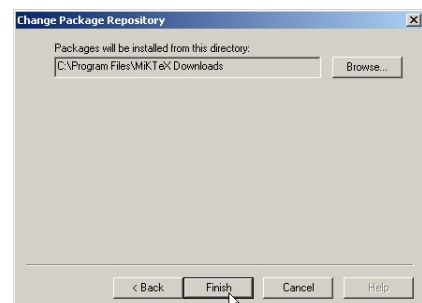
(f)



(g)



(h)



(i)

Figure 7: Modifying the default MiKTeX settings.

- (h) We then indicate that packages should be installed from a (local) directory.
- (i) As a last step, we clarify which directory the local directory is (which should be the same as in Figure 6(e)).

MiKTeX – and thus, \LaTeX – is now fully installed. However, it isn't yet very useful, since we don't have a \LaTeX *shell* (i.e., interface). For example, MiKTeX does not have an editor – you can compose your \LaTeX files from any editor (e.g., Notepad) and then run it from a DOS prompt. This is intentional; MiKTeX, and \LaTeX in general, is designed to be very flexible, so that a given user can use it in a variety of settings. However, the flip side of that is that it can be frustrating for the beginning user, who just wants a standard interface. One possible interface which is highly useful is WinEdt, which is described in the next section.

\LaTeX SHELL: WINEDT

Introduction

As mentioned at the end of the last section, MiKTeX itself does not have an interface. What it does is take a \LaTeX file (TEX, or .tex) and compile it into either a PDF or a PS file. The TEX file can be edited with any available software. Then MiKTeX is called from a DOS prompt, after which the output goes into a specified directory. To drive this process, the user would have to follow many steps which would be very complex:

- Edit the TEX file from within an editor.
- Call MiKTeX from a DOS prompt.
- Open the output (PDF or PS) from the appropriate directory, find the portion pertaining to the section you make the edit, and decide where to make further edits.
- Close the PDF/PS file (since you can't leave it open while running \LaTeX) and repeat.

WinEdt is a piece of software which drives all of this in a very effective way, which is highly flexible, and which can be used for editing many other documents (such as HTML or R files). It is basically an editor which calls MiKTeX, puts the output into the appropriate directory, closes and re-opens the output PDF/PS file, and even brings the user back to the original place that was previously viewed.

Downloading WinEdt

Unlike MiKTeX, WinEdt is actually a very small piece of software. Indeed, since it basically does is drive *other* pieces of software, it doesn't need to be very big. WinEdt installation files will be provided by Nate on a CD, but you can also download it from their website, winedt.com (and notable *not* winedit.com, which is an entirely different product). This is software written by one person, Aleksander Simonic, who started writing it as a grad student. Unlike MiKTeX and the PDF/PS reading software described in this document, *WinEdt is not free*. Rather, it can be used without a license for 30 days, after which it will give a pop-up notification every time it is loaded. A student (lifetime) license costs \$30. While it can technically be used without a license forever, the reader is encouraged to try it out for 30 days and pay the fee if he/she finds it useful after that time.

WinEdt has a major new release every two years or so, and the current build is version 5.5, from 10/3/07. To download this version, go to the website mentioned above, click on DOWNLOADS from the menu on the left, and choose the download from the mirror nearest you (which is probably the one in the US). Note that beta version 5.6 is also available at this time (7/29/08), and the reader is free to try that version – but for stability and thorough testing, the reader is suggested to use version 5.5.

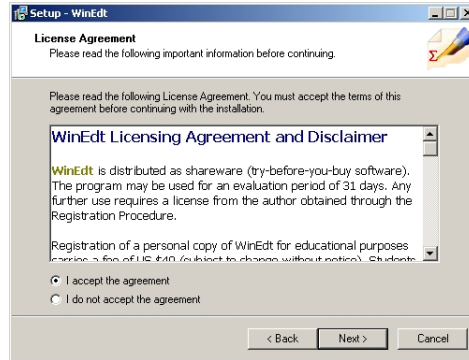
Installing WinEdt

Once downloaded, you should have the file `winedt5.5.exe`. Double click on that, and you should see a series of screens from Figure 8(a)-(i), which we shall go into detail with here:

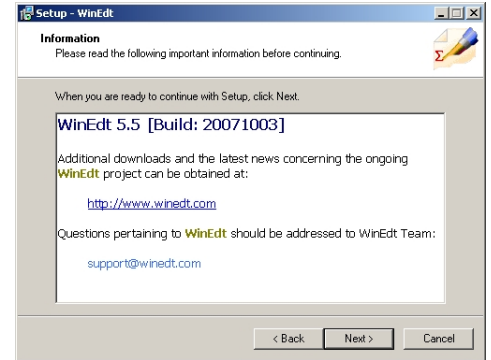
- (a) Welcome to the WinEdt setup wizard. Click on NEXT.
- (b) Accept the user agreement.
- (c) Some extra information. Click on NEXT.
- (d-e) By default, WinEdt is installed in the directory `C:\Program Files\WinEdt Team\WinEdt`. I find it simpler to install it into `C:\Program Files\WinEdt`.
- (f) What would you like your start menu folder to be called?
- (g) Select additional tasks you would like to the installer to install.



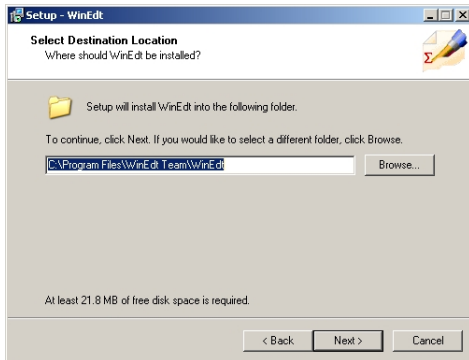
(a)



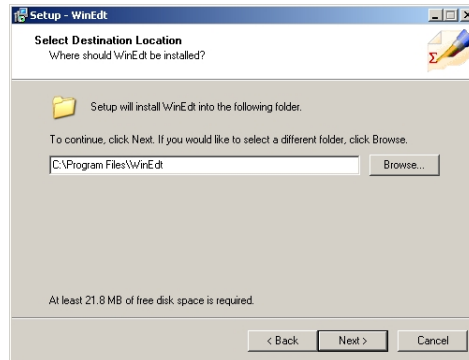
(b)



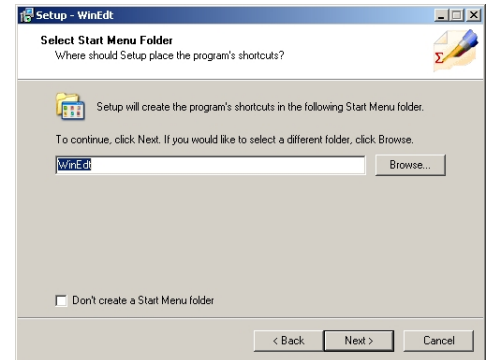
(c)



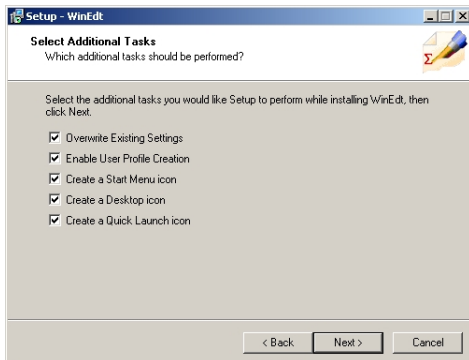
(d)



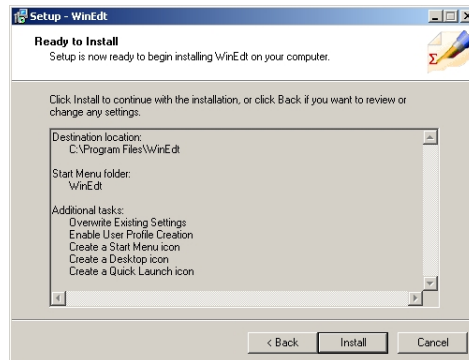
(e)



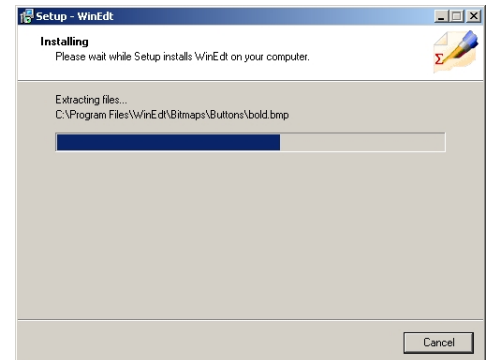
(f)



(g)



(h)



(i)

Figure 8: Installing WinEdt.

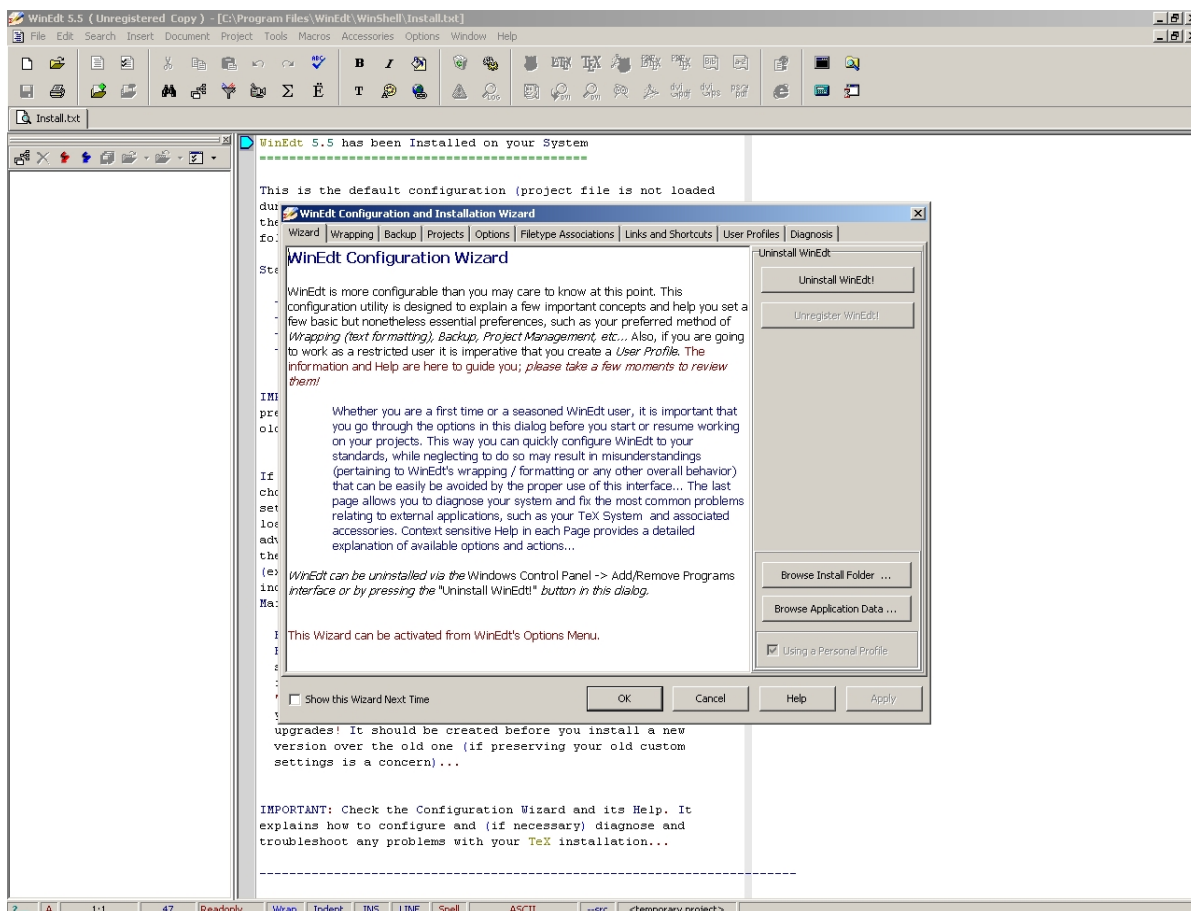


Figure 9: The opening screen of WinEdt after installation.

(h-i) The installer presents a summary of what is to be installed, and then installs it.

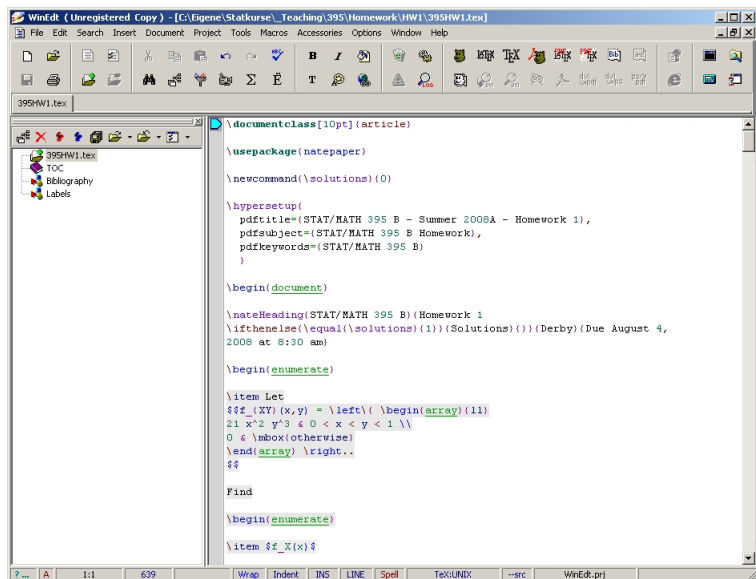
Once installed, you should see a screen that looks like Figure 9. This *configuration and installation wizard* explains lots of directions about how to configure WinEdt to your tastes. There is a lot of information here – my suggestion is to close it (and the rest of WinEdt, to close the installation) and read it later, as it can be found under OPTIONS → CONFIGURATION WIZARD....

Now, assuming we have Adobe Acrobat Reader installed,⁴ we are ready to start using \LaTeX . For starters, we can get used to the WinEdt interface. As an example, suppose we start by opening the \LaTeX source code for STAT/MATH 395 B Homework 1, which is 395HW1.tex. We can open it from within WinEdt, or, for better results, we can right click on it from within Windows Explorer and then click on OPEN WITH → CHOOSE PROGRAM. From this new window, click on BROWSE, and then WINEDT → WinEdt.exe. Lastly, check on the box entitled “ALWAYS USE THE SELECTED PROGRAM TO OPEN THIS KIND OF FILE”. This way, every TEX file will be opened by WinEdt.

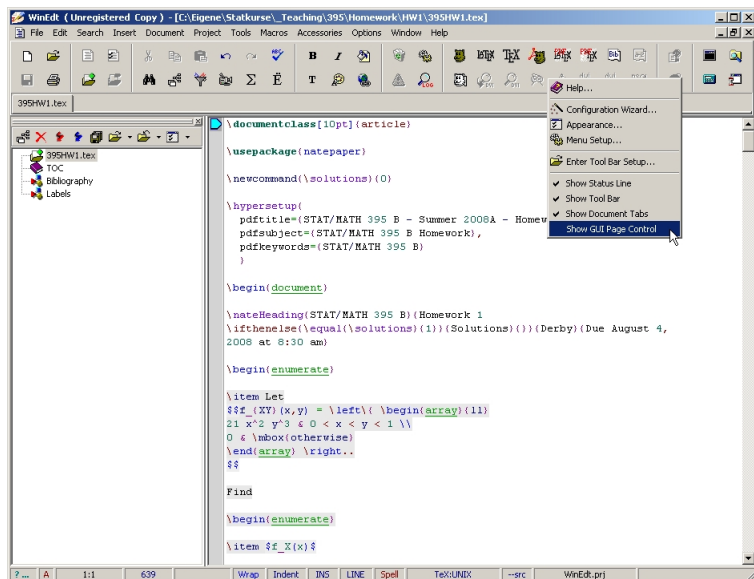
If we look at this file from within WinEdt, we get the result in Figure 10(a). This is the default view from within WinEdt. However, by right clicking on the upper right of the window as shown in Figure 10(b) and deciding whether to show the tool bar and/or the GUI page control, we can show more or less in the headings. In Figure 10(c), we have icons for various \LaTeX commands, while in Figure 10(d), we have minimal headings on top. WinEdt works the same, no matter what you have shown in the headings. Furthermore, WinEdt can be used to edit files other than TEX documents. As mentioned before, I personally use WinEdt for editing many kinds of documents, such as HTML, TXT, R, and others.

A \LaTeX source file is basically a program that tells \LaTeX how to make a PDF or PS document. We will get into the \LaTeX language later, but for now, we can *compile* the \LaTeX source code to produce our PDF file. To do this, the simplest way from within WinEdt is to click on the PDF TEXIFY icon as shown in Figure 11(a). A DOS command prompt will open up and, if this is the first time you are running a given package, a pop-up window will ask if you would like to install a given package, as shown in Figure 11(b). Click on INSTALL every time you are asked – all you are doing here is installing a new package from the repository you already installed yourself. When it is done, Adobe Reader will open up and show you your document.

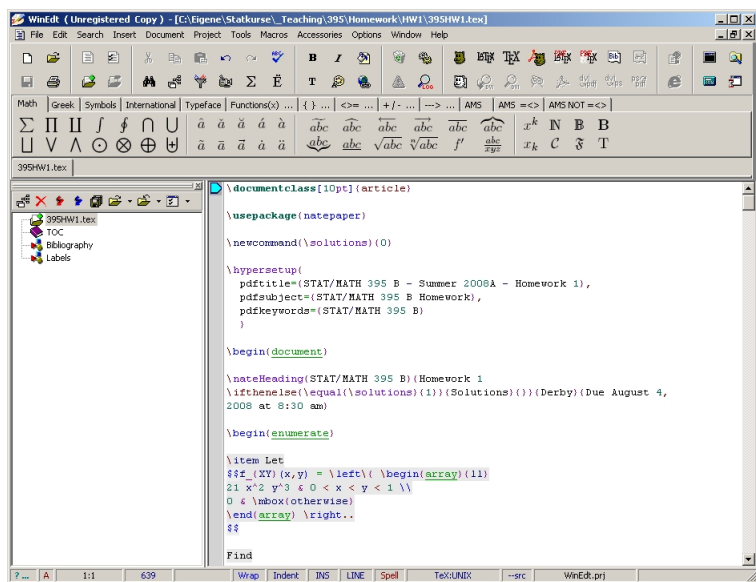
⁴Other PDF readers, such as Foxit, do not work with WinEdt.



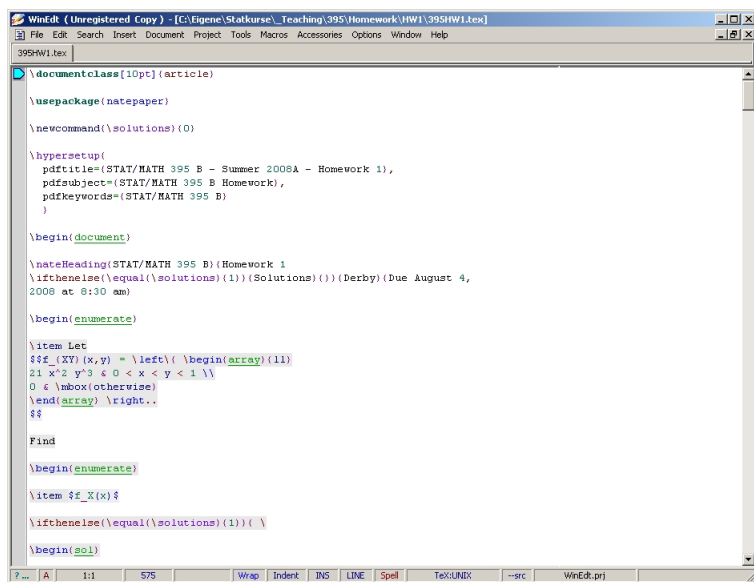
(a)



(b)

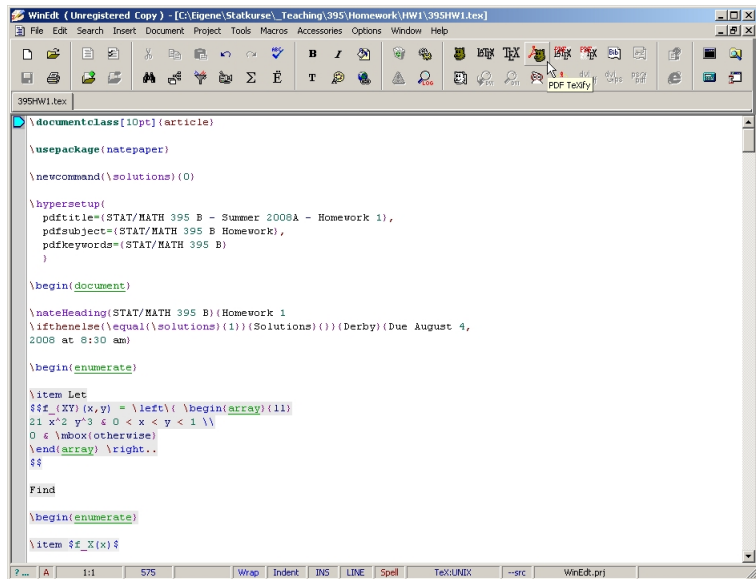


(c)

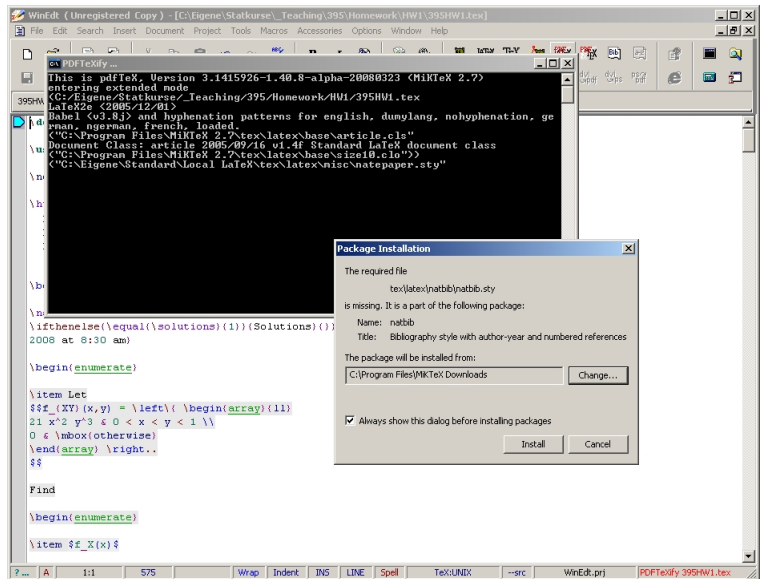


(d)

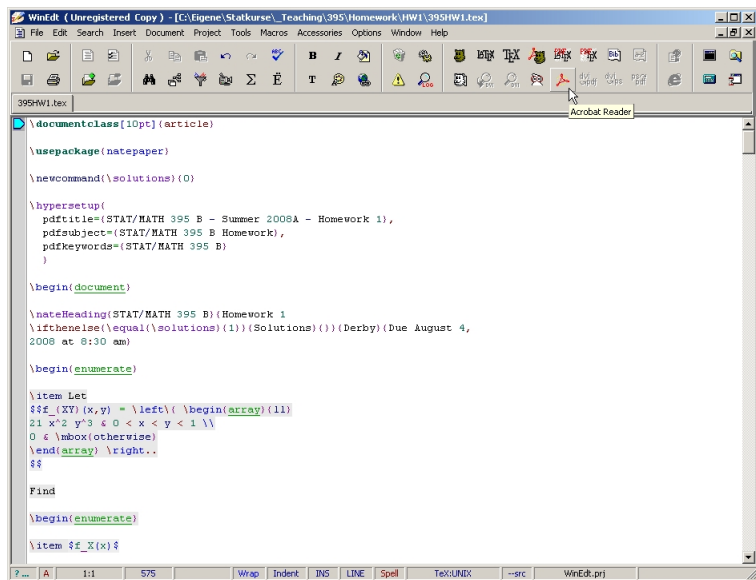
Figure 10: The opening screen of WinEdt after installation.



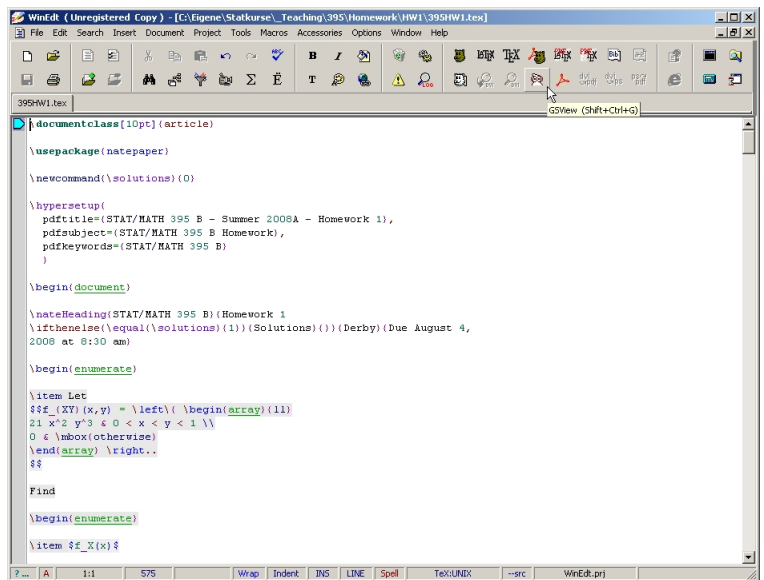
(a)



(b)



(c)



(d)

Figure 11: The opening screen of WinEdt after installation.

In WinEdt, whenever you click on an icon in the upper right, it is applied to the document that is shown, in whatever directory it happens to be in. For instance, if you click on the Adobe icon (as shown in Figure 11(c)), it will open up your document in the state it was when you last compiled it. If you click on the GSView icon (as shown in Figure 11(d)), it will open your document (the PS version if it exists, or the PDF version if not) from within that application. Irrelevant icons are greyed out. For example, the Internet Explorer icon is greyed out for a TEX document. If you have an HTML document open, that icon will be colored, but the Acrobat Reader and GSView icons will be greyed out.

As briefly mentioned before, WinEdt is highly modifiable. For example, my WinEdt settings are configured so that I call PDF TEXIFY (thus compiling the code and showing me the output) with F8, PDF LATEX (compiling the code) with F9, Acrobat Reader (showing the output) with F10, and Firefox (showing me updated HTML code) with F7. You can do this as shown in Figure 12(a)-(d). As shown here, we first go to OPTIONS → MENU SETUP. From there, within the MAIN MENU, we click on &Accessories, then &PDF, then PDF Te&Xify, then set the shortcut to your desired key.

PDF READER: ADOBE LITE

WinEdt only works with Acrobat Reader. Note that more than one PDF reader can be installed on a machine at once – so if you typically use another PDF reader, you don't have to give that up. However, if you (like me) find that Acrobat Reader is too resource-intensive, there is the option to use something called *Acrobat Lite*, which is a slimmed-down version of Acrobat Reader which uses fewer resources and which is (much) quicker. It is available from multiple vendors – just use Google to find it. The latest version (as of 7/29/08) is version 9, but as it is less than one month old, my personal preference is to wait a little more time before using it. You are free to do so.

PS READER: GSVIEW

PS documents are becoming much less common than they were 10 or even 5 years ago. However, for academics in math or statistics, you may still encounter some PS documents and/or be asked to submit something as a PS rather than PDF document. If that is the case, you will need to install both GSView and Ghostscript, both of which are available from pages.cs.wisc.edu/~ghost. Both are free.

USING LATEX

There is definitely a learning curve for learning how to code in LATEX, but fortunately there are also many resources for this purpose. The best of them are Goossens et al. (1994) for a comprehensive reference, and Griffiths and Higham (1997) for many useful examples. The latter book, with a little over 50 pages without the appendices, is mercifully slim, and thus very useful for the beginner who wants to start with simple examples. It is also useful for the price – less than \$30 new.

However, my own opinion is that it is simplest to start off with someone else's LATEX file and to modify it, step by step. For students of my STAT/MATH 395 B class, an opportunity for this is in using the source code for the homework and doing the homework using that source code. Thus, every time a homework is assigned on the class web site (nederby.org/stat395.html), the source code will also be included. However, do note that this source code will also need the package `natepaper.sty`, as described below. First, we should learn just a couple preliminaries.

The first thing to know about LATEX code is that it is designed to separate form from content. Thus, most of the formatting (margins, etc) is done in the heading of the document. Practically, this means that most (or all) of the formatting is done before the line

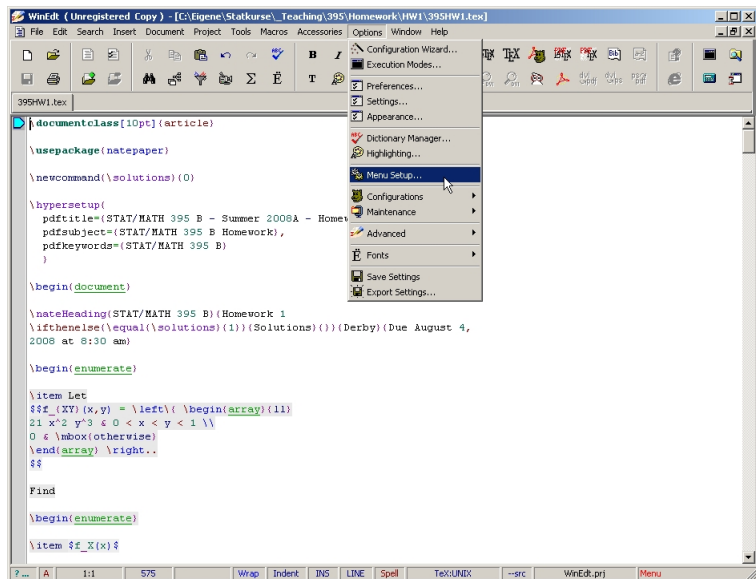
```
\begin{document}
```

and the content is included between this line and the last line of the file, which is, as expected,

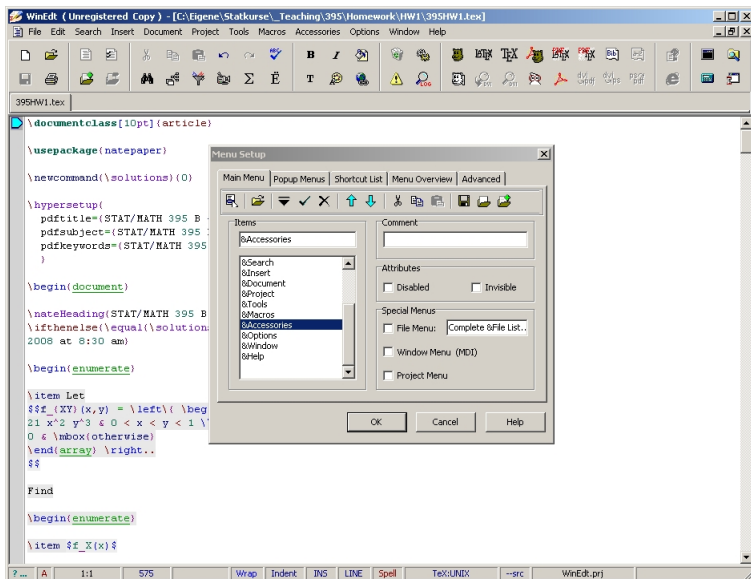
```
\end{document}
```

There are a few other things to know before looking at LATEX code:

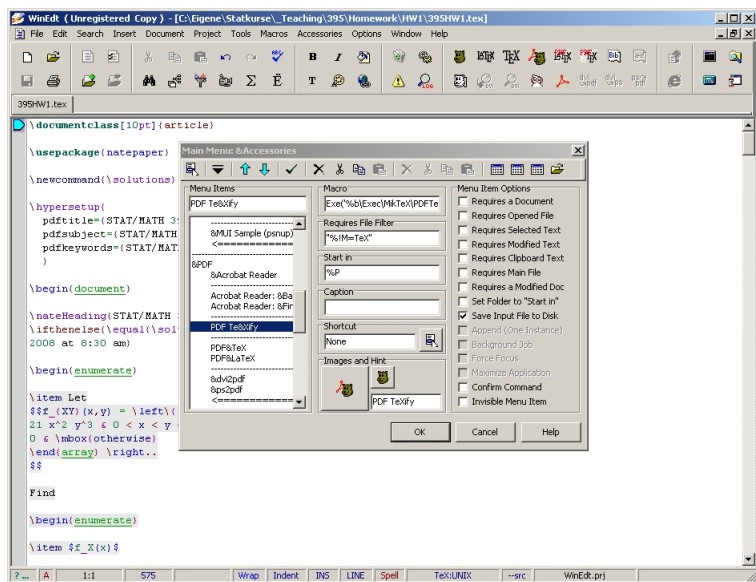
- Regular text is written as just regular text. *Math text*, which refers to mostly mathematical symbols such as x or $\int_3^4 f_X(x) dx$, are in between single dollar signs for inline text, or double dollar signs for centered text. For example, x is written as `x`.
- Special functions or characters in LATEX, such as the function that italicizes text, are preceded by a backslash (`\`). For example, *putting this into italics* is coded as `\emph{putting this into italics}`. A dollar sign is written



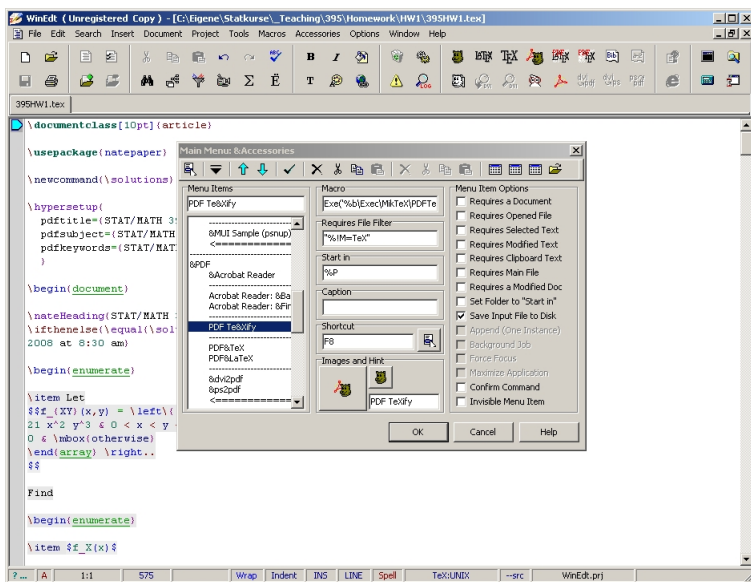
(a)



(b)



(c)



(d)

Figure 12: Setting up WinEdt to compile a \LaTeX document with a special key (in this case, F8).

as `\$` (since `$` indicates the beginning or end of math text, as noted above), and a backslash itself is coded as `\textbackslash`.

- \LaTeX uses a number of different *environments* which are designated by `\begin{xxx}` and `\end{xxx}`, and where anything between those statements are within that environment. The `document` environment has been described above already. Another example is the `itemize` environment, which is used to write this text right here. For example, this text is coded as

```
\begin{itemize}

...

\item \LaTeX\ uses a number of different \emph{environments} which are designated by
\verb"\begin{xxx}" and \verb"\end{xxx}", and where anything between those statements
are within that environment. The \texttt{document} environment has been described
above already. Another example is the \texttt{itemize} environment, which is used to
write this text right here. For example, this text is coded as

\end{itemize}
```

Furthermore, when a \LaTeX file (TEX) is compiled, a few more files are created. Let's suppose our file is named `foo.tex`. What is created when this file is compiled into a PDF document?

- `foo.tex.bak`: This is a backup of the last copy of `foo.tex` before it was saved. When \LaTeX compiled `foo.tex`, it first saves the file – which is generally good to know. The previous version of `foo.tex` is renamed as this backup file. This is done specifically in the common case where the user does not intentionally save anything himself. That is, all that is saved is done so when compiled by \LaTeX . In this case, the backup file represents the last compilable version of the file. Thus, if your new version of `foo.tex` crashes, you always have a “clean” version to go back to.
- `foo.aux`: This is called an *auxiliary file*, and it has all the numberings that \LaTeX does automatically – figure numbers, equation numbers, and the like. \LaTeX accesses it internally; unless you are purely interested in seeing it, there is no reason to look at it, ever. However, in some cases where \LaTeX mysteriously keeps crashing, the cause might be a corrupted AUX file. In this case, simply delete this file. This is a common fix for unexpected crashes.
- `foo.log`: This is a log file of notes of what \LaTeX did the last time it compiled `foo.tex`. This is identical to the information that is quickly shown on the DOS window when compiled. This can be useful for debugging purposes for the technically inclined.
- `foo.toc`: This keeps track of pages for the table of contents, if there is one included in `foo.tex`.
- `foo.bbl`: This keeps track of all bibliographic references called from within `foo.tex`.
- `foo.blg`: This is the log of what happened when \LaTeX 's internal bibliographic scanner (called BibTeX) collected the bibliographic information contained in `foo.bbl`.
- `foo.pdf` or `foo.ps`: The output file.

There may be other files as well, depending on what you are doing with your TEX document, but the main point to remember is that *all* of them are produced from `foo.tex`. If you are having a series of unexplained errors, just delete everything except for the main TEX file and start anew.

With this extremely short introduction, the reader is encouraged to look through and experiment with some sample \LaTeX code. For more information, Griffiths and Higham (1997) gives much more, better, and concise information than I could.

Lastly, for the reader who would like to experiment with my STAT/MATH 394 B homework files, I mentioned that the \LaTeX package `natepaper.sty` is needed. Let me explain a little more about this:

- \LaTeX packages are files which have the extension of CLS (for *class*) or STY (for *style file*). Every \LaTeX file starts with the following commands:

```
\documentclass{xxx}

\usepackage{yyy,zzz,...}
```

which says that your file is going to be of the class `xxx`, and which will use packages `yyy`, `zzz`, and others. Clearly you can have just one class, but many packages. Specialized classes are relatively rare, whereas style files are much more common. This may be by design – if you wrote a package as a class, you are forcing a user to choose between your class and another; whereas if you make it a style file, a user can combine it with anything else.

- \LaTeX packages, whether classes or style files, essentially define macros which can be used by the user instead of something much more complicated. In fact, \LaTeX itself is a set of macros which dramatically simplify the more complicated \TeX , developed by Donald Knuth in the early 1980s.
- `natepaper.sty` is a set of small macros that I typically use in many of my documents. For example, I prefer to denote a probability as $P(A)$ rather than $P(A)$, where the letter P , unlike all the other characters which surround it, is in plain text and not math text.⁵ Within the math text mode, this would normally be coded as `\mbox{P}(A)`, but with my (very small) macro, it can instead be coded as `\P(A)`. This may seem trivial, until you realize that I teach a probability class, and thus use this notation many, many times in my documents.

To set up MiKTeX so that it accesses the macros in `natepaper.sty`, place that file into the `tex\latex\misc` subdirectory of the local directory as shown in Figure 7(c), and then refresh the file name database as shown in Figure 7(f).

REFERENCES

Derby, N. (2008a), *User's Guide to %exportFromXL, Version 1.0*.

<http://exportFromXL.sf.net/docs/exportfromxlv1.0-ug-cur.pdf>

Derby, N. (2008b), *User's Guide to %exportToXL, Version 1.01*.

<http://exportToXL.sf.net/docs/exporttoxlv1.01-ug-cur.pdf>

Goossens, M., Mittelbach, F. and Samarin, A. (1994), *The \LaTeX Companion*, Addison-Wesley, Boston.

Griffiths, D. F. and Higham, D. J. (1997), *Learning \LaTeX* , Society for Industrial and Applied Mathematics, Philadelphia.

Tantau, T. (2008), *TikZ and PGF: Manual for Version 2.00*.

<http://sourceforge.net/projects/pgf/>

Taraborelli, D. (2008), The beauty of \LaTeX .

<http://nitens.org/taraborelli/latex>

Wierda, G. (ed.) (2008), *The TeX Showcase*.

<http://www.tug.org/texshowcase/>

ACKNOWLEDGMENTS

I thank my present and past employers for giving me such a high degree of flexibility in organizing my projects. I very much thank SAS technical support for helping me learn the basics of macro programming. I also thank Ron Fehd for providing me with a \LaTeX template for SAS conference papers (used here).

Lastly, and most importantly, I thank Charles for his patience and support.

® indicates USA trademark registration.

Copyright 2008 by Nathaniel Derby

⁵The logic behind this decision is that upper case math text letters should all denote random variables. Thus, " P " is a random variable, whereas " P " is a probability (or a Poisson distribution, depending on the context).