Ontology of Virtual Entities

Example Independent Study Thesis

Presented in Partial Fulfillment of the Requirements for
the Degree Bachelor of Arts In Computer Science and
Philosophy in the
Department of Math and Computer Science at The College of
Wooster

by
Zachary A Phillips-Gary

The College of Wooster
2016

Advised by:

Denise , Ph.D

Awesome Philosophy Adviser

Abstract

Include a short summary of your thesis, including any pertinent results. This section is *not* optional for the Mathematics and Computer Science or Physics Department ISs, and the reader should be able to learn the meat of your thesis by reading this (short) section.

This work is dedicated to the future generations of Wooster students.

Acknowledgments

I would like to acknowledge Prof. Lowell Boone in the Physics Department for his suggestions and code.

Vita

Publications

Fields of Study Major field: Major

Minor field: Minor

Specialization: Area of IS research

Contents

List of Figures

List of Tables

List of Listings

Preface

The purpose of this document is to provide you with a template for typesetting your IS using LaTeX. LaTeX is very similar to HTML in the sense that it is a markup language. What does this mean? Well, basically it means you need only enter the commands for structuring your IS, i.e., identify chapters, sections, subsections, equations, quotes, etc. You do not need to worry about any of the formatting. The `woosterthesis` class takes care of all of the formatting.

Here is how I plan on introducing you to LaTeX. The Introduction gives some reasons for why one might find LaTeX superior to MS Word^{TM}. Chapter 2 will demonstrate how one starts typesetting a document and works with text in LaTeX. Chapter 3 discusses the creation of tables and how one puts figures into a thesis. Chapter 4 talks about creating a bibliography/references section and an index. There are three Appendices which discuss typesetting mathematics and computer program code. The Afterword will discuss some of the particulars of how a LaTeX document gets processed and what packages the `woosterthesis` class uses and are assumed to be available on your system.

Hopefully, this document will be enough to get you started. If you have questions please refer to **? ? ? ? ?**], or **?**].

CHAPTER 1

Introduction

So why would you want to use LaTeX instead of Microsoft Word™? I can think of several reasons. The main one for this author is that LaTeX takes care of all of the numbering automatically. This means that if you decide to rearrange material in your IS, you do not have to worry about renumbering or references. This makes it very easy to play around with the structure of your thesis. The second reason is that it is ultimately faster than Word™. How? Well, after a week or so of using LaTeX you will begin to remember the commands that you use frequently and won't have to use the LaTeX pallet in TeXShop or TeXnicCenter. So you can just type everything including the mathematics, where with Word™ you would have to use the Equation Editor.

I have also tried to make things more efficient by organizing the example folder as follows. There is a `username.tex` file which you will want to rename using your username and which is what you will enter all of the information about your IS into. `username.tex` also has explanations about other files that you might need to edit. In addition there are folders for chapters, appendices, styles, and figures. This structure is there to try and reduce file clutter and to help you stay organized. There should also be a .bib file which you can use as a model for your own .bib file. The .bib file has your bibliographic information.

LaTeX is really easy to learn. For an average IS, the author will only need to learn a handful of commands. For this small bit of effort, you get a tremendous amount of flexibility and a very beautiful document. The following chapters will introduce some of the common things a student might need to do in a thesis.

CHAPTER 2

In the beginning: Knuth said "Let there be TeX"

Now that I've tried to convince you that LaTeX is going to be better than Word™ for your IS, you're saying, "So how do I use it?" Well let's start with some basic things. First, how is a document structured in LaTeX?

A *document* for LaTeX is all the stuff that comes between the `\begin{document}` and `\end{document}` tags. The `username.tex` file has the `\begin{document}` and `\end{document}` tags. "OK, but how do I get my chapters to print?" You save the chapters in the `chapters` folder and put an `\include{chapters/chaptername}` command in `username.tex` after the `\begin{document}` and before the `\end{document}` tag. `username.tex` already has some examples of including chapters; you can just alter them to have your chapter names. I should also mention that the % symbol is used for comments. The `username.tex` file has a number of comments that are intended for you and try to explain what is happening. Oh, and if you need a % symbol enter `\%`.

Now to write your first chapter. I would recommend saving this chapter (chapter1.tex) under a different name and making changes to the new copy. The most basic structural elements that you need to know are the paragraph, `\chapter`, `\section`, and `\subsection`. A new paragraph is obtained by putting a blank line in the source file. The other commands are very easy to use. If I want to start a new section I enter `\section[My new section]{An example of making a new section and giving it a short name}` (the part in square brackets is optional) and get

2.1 An example of making a new section and giving it a short name

The `\chapter` and `\subsection` commands work in exactly the same manner. Each new chapter must have `\chapter[short name]{chapter name}` as its first line.

"Hey, wait a minute. What if I need to refer to that section? How can I do that?" It's actually as simple as adding`\label{labelname}` at the end of the `\chapter` command like`\section[My new section]{An example of making a new section and giving it a short name}\label{sec:newsec}`. Now I can refer to Section 2.1 by typing `\ref{sec:newsec}`. You can label just about anything and refer to the label to get an automatically generated number for the item. This means that you need to come up with a labeling scheme before you start writing and stick with it.

Some other things you'll need to be able to do include italicizing and bolding text and creating lists. These are also easy to accomplish. For example I can use `\emph` or `\textit` to italicize text. To italicize homework I would enter `\emph{homework}` or `\textit{homework}` to produce *homework*. To obtain **bold** text you would use the `\textbf` command. And what about lists?

There are several kinds of lists (enumerated, itemized, and descriptive) and each has its own place and environment. An enumerated list is good for outlining or ordered lists:

```
\begin{enumerate}
\item First main idea
\begin{enumerate}
\item First subpoint
\item\label{enum:1b} Second subpoint
\end{enumerate}
\item Second main idea
\end{enumerate}
```

1. First main idea
   (a) First subpoint
   (b) Second subpoint
2. Second main idea

The itemized list is good for unordered lists or bullet points:

```
\begin{itemize}
\item Idea
\item Idea
\item Idea
\item Idea
\end{itemize}
```

- Idea
- Idea
- Idea
- Idea

And the descriptive list is good for definitions; however, `amsthm` already has a definition environment, and you will most likely not need the description environment. In any event, here is an example:

```
\begin{description}
\item[First item:] Idea
\item[Second item:] Idea
\item[Third item:] Idea
\end{description}
```

First item: Idea

Second item: Idea

Third item: Idea

Notice the use of brackets in the last example. The brackets are optional and the text in the brackets is used as the label for the item. You should also note that you can label an item for later reference see 1b. There are several options for changing the format of the list environments and a package, `paralist`, for customizing lists which are described in section 3.3 of **?**].

2.2 Theorems, definitions, examples, oh my!

The next thing you'll probably need to do is enter definitions, theorems, and examples. Below you will find some examples. On the left you will see the text typed into the document and on the right what it looks like when formatted. These examples are intended to give you a sense of what type of mathematical expressions LaTeX handles. You should look at Appendix A for a more complete discussion of entering mathematics. In the beginning you will not know all of the commands that you need to enter. Don't worry. Each of the suggested editors has a palette that shows you a picture of what you want and puts the correct commands into the document when you click the picture. As you look at these examples, keep it in mind that some of them use some user defined commands which can be found in `styles/personal.tex`. Now lets look at Definition 2.1 **??**, Theorem 2.1, and equation 2.3.

```
\begin{defn}[One of Ramanujan's
 third order mock theta
 functions]\label{def1}
 \begin{equation}\label{introf(q)}
 f(q)=1+\sum_{y=1}^{\infty}
 \frac{q^{y^2}}{(1+q)^2(1+q^2)^2
 \cdots (1+q^y)^2}.
 \end{equation}\end{defn}
```

Definition 2.1 One of Ramanujan's third order mock theta functions:

$$f(q) = 1 + \sum_{y=1}^{\infty} \frac{q^{y^2}}{(1+q)^2(1+q^2)^2 \cdots (1+q^y)^2}. \tag{2.1}$$

```
\begin{thm}[Watson's
transformation of
$f(q)$]\label{introwatthm}
\begin{equation}\label{introf}
\qrfac{q}{\infty}
\sum_{y=0}^{\infty} q^{y^2}
 \qrfac[-2]{- q}{y}=1+
 \sum_{y=1}^{\infty}
 \frac{(-1)^{y}
 4q^{(3/2)y^2+
 (1/2)y}}{(1+q^{y})}.
 \end{equation}\end{thm}
```

Theorem 2.1 (Watson's transformation of $f(q)$).

$$(q)_\infty \sum_{y=0}^{\infty} q^{y^2} (-q)_y^{-2} = 1 + \sum_{y=1}^{\infty} \frac{(-1)^y 4q^{(3/2)y^2+(1/2)y}}{(1+q^y)}. \tag{2.2}$$

This is a more complicated example which uses the `\substack` command to have multiple summation criteria.

```
\begin{align}\label{m.1diasumtwo}
\left[NUM\right]_1^{(\fl)}(q;b;
\bvec{x})=&\ q\sum\limits_{
\substack{ 0\leq r,t
\leq\fl-1}}
q^{r+t}\sum\limits_
 {\substack{{\lambda
 \vdash (r+t)}\\
  \lambda/1^r\in V_t\\
  \ell(\lambda)\leq \fl-1}}
  \mathrm{s}_{(b,\lambda)}
  (\bvec{x}).\end{align}
```

$$[NUM]_1^{(\ell)}(q;b;\mathbf{x}) = q\sum_{0\leq r,t\leq \ell-1} q^{r+t}\sum_{\substack{\lambda\vdash(r+t)\\ \lambda/1^r\in V_t\\ \ell(\lambda)\leq\ell-1}} \mathrm{s}_{(b,\lambda)}(\mathbf{x}). \tag{2.3}$$

Another thing that one might need to do is create piecewise definitions. This can be accomplished by using the `cases` environment. This example also uses the `\intertext` command to put text between displayed equations.

```
\begin{subequations}\label{2c1BP}
\begin{alignat}{2}\label{2c1BPa}
A_{y_1}:=&\begin{cases}
 1 &\text{for $y_1=0$},\\
\frac{-1)^{y_1}
4q^{y_1}q^{\binom{y_1}{2}}}
{\qrfac{q}{2y_1}(1+q^{y_1})}
&\text{for $y_1>0$}\end{cases}\\
\intertext{and} B_{y_1}:=&
\qrfac[-1]{-q}{y_1}\qrfac[-1]
{-q}{y_1}=\qrfac[-2]{-q}{y_1}
&.\label{2c1BPb}\end{alignat}
\end{subequations}
```

$$A_{y_1} := \begin{cases} 1 & \text{for } y_1 = 0, \\ \frac{-1)^{y_1} 4q^{y_1} q^{\binom{y_1}{2}}}{(q)_{2y_1}(1+q^{y_1})} & \text{for } y_1 > 0 \end{cases} \tag{2.4a}$$

and

$$B_{y_1} := (-q)_{y_1}^{-1}(-q)_{y_1}^{-1} = (-q)_{y_1}^{-2} \quad . \tag{2.4b}$$

Finally, if you need to incorporate examples into your thesis you can do it using the example environment, as seen in Example 2.1.

```
\begin{ex}[An example example]
\label{ex:ex}
This is an example of including an
 example. Kind of silly isn't it.
 \end{ex}
```

Example 2.1 (An example example). This is an example of including an example. Kind of silly isn't it.

2.3 Putting code in the main body of the thesis

There is one last textual item which Computer Science majors and probably some Mathematics majors will need to incorporate, pseudocode. To do this I would suggest using the `\lstlisting` environment.

Below is an example set up for the `listings` package. You could put your modifications to this set up into the `personal.tex` file in the `styles` folder. Documentation on the `listings` package can be found in the `doc` folder with the documentation for the other packages.

```
\lstset{
        language =Pascal, % pick a language style
        emph={return,natural, numbers, integers, increasing},
        emphstyle={\bfseries},% choose other keywords and a format
        linewidth=.95{\textwidth}, breaklines=true,commentstyle=\textit,
        stringstyle=\upshape,showspaces=false,numbers=left,
        numberstyle=\tiny,basicstyle=\small,xleftmargin=30pt,
        breakautoindent=true,captionpos=b
        }
```

The listing in Listing 2.1 gives an algorithm for finding the largest even integer in a given list of n integers. I have used the `mathescape` option to be able to incorporate mathematics in the listing. The actual code put in the thesis is given first and the formatted output follows.

```
\begin{lstlisting}[mathescape, caption= Find the location
of the largest even integer in a list,label=largesteven]
procedure $largestevenlocation$($a_1, a_2, \ldots, a_n$: integers)
$k$:=0
$largest$:=-$\infty$
for $i$:=1 to $n$
  if ($a_i$ is even and $a_i>largest$) then
  begin
    $k$:=$i$
    $largest$:=$a_i$
  end
end
return $k$
\end{lstlisting}
```

procedure *largestevenlocation* $(a_1, a_2, \ldots, a_n :$ **integers**$)$
$k := 0$
$largest := -\infty$
for $i := 1$ **to** n
  if $(a_i$ is even **and** $a_i > largest)$ **then**
  begin
    $k := i$
    $largest := a_i$
  end
end
return k

Listing 2.1: Find the location of the largest even integer in a list

The code in Listing 2.2 is an improvement on Binary search. The algorithm reduces the size of the search by a factor of four at each iteration. It provides another example of using the `\lstlisting` environment.

```
\begin{lstlisting}[mathescape,caption=Quartary search,
label=quartsearch]
procedure $quartarysearch$($x$: integer, $a_1, a_2,
 \ldots, a_n$: increasing integers)
$i$:=$1$
$j$:=$n$
while $i<j-2$
begin
  $l:=\lfloor(i+j)/4\rfloor$
  $m:=\lfloor(i+j)/2\rfloor$
  $u:=\lfloor3(i+j)/4\rfloor$
  if $x>a_m$ then
    if $x\leq a_u$ then
    begin
      $i:=m+1$
      $j:=u$
    end
    else
     $i:=u+1$
  else if $x>a_l$ then
    begin
      $i:=l+1$
      $j:=m$
    end
    else $j:=l$
end
if $x=a_i$ then $location:= i$
else if $x=a_j$ then $location:= j$
else if $x=a_{\lfloor(i+j)/2\rfloor}$ then
 $location:= \lfloor(i+j)/2\rfloor$
else $location:= 0$
return $location$
\end{lstlisting}
```

```
procedure quartarysearch(x: integer, a_1, a_2, ..., a_n: increasing integers)
i:=1
j:=n
while i < j − 2
begin
  l := ⌊(i + j)/4⌋
  m := ⌊(i + j)/2⌋
  u := ⌊3(i + j)/4⌋
  if x > a_m then
    if x ≤ a_u then
    begin
      i := m + 1
      j := u
    end
    else
     i := u + 1
  else if x > a_l then
    begin
      i := l + 1
      j := m
    end
    else j := l
end
if x = a_i then location := i
```

```
else if x = a_j then location := j
else if x = a_⌊(i+j)/2⌋ then location := ⌊(i + j)/2⌋
else location := 0
return location
```

Listing 2.2: Quartary search

2.4 What is in `username.tex`

Before we move on let's talk a little bit about what is at the beginning of `username.tex`. The file starts with `\documentclass{woosterthesis}`, which must be at the beginning of every IS. In the brackets are options for the woosterthesis class. The options are the same as for the `book` class with some additional options `abstractonly`, `alltt`, `blacklinks`, `code`, `dropcaps`, `euler`, `guass`, `index`, `kaukecopyright`, `palatino`, `picins`, `verbatim`, and `xetex`. The `kaukecopyright` option will put the arch symbol with the word mark on the copyright page. The `blacklinks` option will make the hyperlinks in the PDF version of the thesis black and suitable for printing; normally the links are colored to provide visual clues to the reader. The `code` option will use `listings` style to format program code examples. The `abstractonly` option will allow you to print just the Abstract. The `palatino` option will use the `pxfonts` package which uses the Palatino fonts. The `picins` option will use the `floatflt` package to allow text to wrap around images. `index` will allow the `makeidx` package to be loaded so that if you have index entries they will be added to an index (this reqires additional steps). `dropcaps` loads the `letterine` package for doing dropped capitals and `alltt` loads the `alltt` for using typewriter type in various ways. `verbatim` allows one to set verbatim what is entered. `euler` and `guass` load the `woofncychap` package with the named option which will change the look of chapter headings. Finally `xetex` will allow you to use the XeTeX extension of TEX for easy use of system fonts. Adding or deleting options from the comma separated list will change the appearance of the document and some options should only be used after consulting your advisor. Now let's move on to some other things that you'll need to deal with: figures, pictures, and tables.

CHAPTER 3

Working with figures and tables

3.1 Getting a simple figure in the document

In this chapter we want to talk about including figures and tables in the document. To insert a simple figure you can enter something like

```
\begin{figure}[!ht]
\begin{center}
\woopic{picture3}{.8}
\end{center}
\caption{Our first
 picture}\label{first}
\end{figure}
```

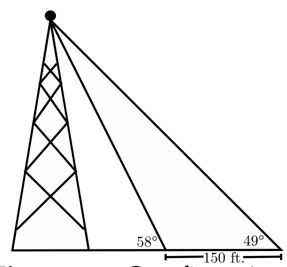


Figure 3.1: Our first picture

The `!ht` tell LaTeX to try and place the figure here no matter what or at the top of the next page. The `\woopic` command takes the name of the picture as the first argument and the scaling factor as the second argument. The scaling factor must be between zero and one and the figure name must have *no spaces*. Your figures can be in one of three formats: jpg, tif, or pdf. Captions are placed below the figure and your label should be placed after the caption.

In the next example we are using the woosterthesis option `picins` to typeset a picture inside a paragraph and have the text wrap around the figure. This option loads the `wrapfig` package. One thing to note is that the figures placed in this manner do not float with the other figures and as such numbering could get out of sequence. Keep an eye out for such behavior. This technique should be used sparingly in your thesis.

```
\newcommand{\sample}{Some text that is reused over and over
 again in the example. }
\begin{wrapfigure}{r}{2.2in}
\woopic{picture2}{.4}
\caption{Conchoid.}
\end{wrapfigure}
\sample\sample\sample\sample
```

Some text that is reused over and over again in the example. Some text that is reused over and over again in the example. Some text that is reused over and over again in the example. Some text that is reused over and over again in the example. Some text that is reused over and over again in the example. Some text that is reused over and over again in the example. Some text that is reused over and over again in the example. Some text that is reused over and over again in the example. Some text that is reused over and over again in the example.

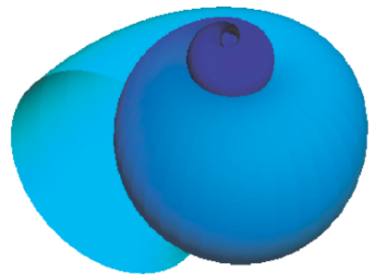


Figure 3.2: Conchoid.

3.1.1 Minipages

You can also create minipages in your documents to accomplish more complicated formatting. For example you could try the following which produces Figure 3.3.

```
\begin{minipage}[t][3 in][t]{1 in}
This is a minipage which is 3 in tall and 1 in wide.
 Top Text Text Text Text.\end{minipage}\hfill
\begin{minipage}[t][3 in][c]{1 in}
This is a minipage which is 3 in tall and 1 in wide.
 Center Text Text Text Text.\end{minipage}\hfill
\begin{minipage}[t][3 in][b]{1 in}
This is a minipage which is 3 in tall and 1 in wide.
 Bottom Text Text Text Text.\end{minipage}
```

In the example above, the syntax `\begin{minipage}[t][3 in][t]{1 in}` follows the convention

`\begin{minipage}[minipageposition][height][textposition]{width}`

3.1.1.1 How to get more than one picture in the same figure

You can use minipages to put more than one picture in a figure. Here is an example of how to do this.

```
\begin{minipage}[!ht]{6cm}
\woopic{picture1}{.4}
```

This is a mini-page which is 3 in tall and 1 in wide. Top Text Text Text Text.

This is a mini-page which is 3 in tall and 1 in wide. Center Text Text Text Text.

This is a mini-page which is 3 in tall and 1 in wide. Bottom Text Text Text Text.

Figure 3.3: Minipage example

```
\par
\caption[What goes in the List of Figures]{Left}
\end{minipage}
\hfill
\begin{minipage}[!ht]{6cm}
\woopic{picture2}{.4}
\end{picture}\par
\caption{Right}
\end{minipage}
```

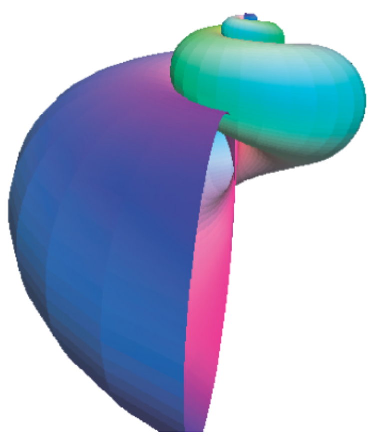


Figure 3.4: Left

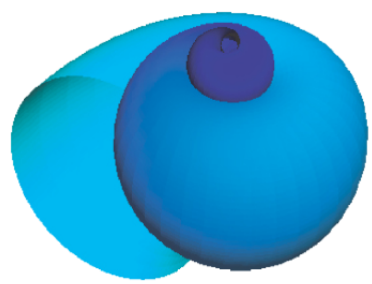
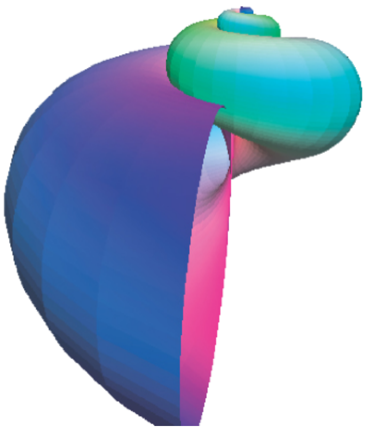


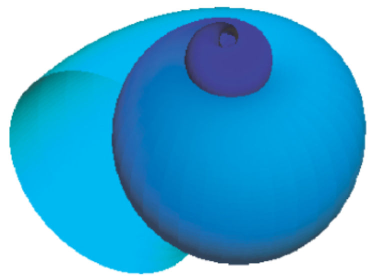
Figure 3.5: Right

You can also use the `subfigure` package to do this.

```
\begin{figure}[!ht]\centering
\subfigure[What goes in the List][Large conchoid]
{\woopic{picture1}{.4}\label{fig3:left}}
\qquad
\subfigure[What goes in the List][Small conchoid]
{\woopic{picture2}{.4}\label{fig3:right}}
\caption{Two pictures in one figure}\label{fig3}
\end{figure}
```



(a) Large conchoid



(b) Small conchoid

Figure 3.6: Two pictures in one figure

We should now be able to refer to either Figure 3.6 (a) or Figure 3.6 (b) using the labels we gave to the left and right images.

The reader is referred to Chapters 8, 9, and 16 of **?**] or to Chapters 6 and 10 of **?**] for a complete discussion of figures and graphics.

3.2 Tables

Tables are fairly easy to set up. Here is a simple table

```
\begin{table}[!ht]
\begin{center}
\begin{tabular}{r l}
  $\underline{\textnormal{District}}$ &
  $\underline{\textnormal{Population}}$\\
   Applewood & 8280 \\
   Boxwood & 4600  \\
   Central & 5220
   \end{tabular}\caption{Our first table}
   \end{center}
\end{table}
```

District	Population
Applewood	8280
Boxwood	4600
Central	5220

Table 3.1: Our first table

In `\begin{tabular}{r l}` the two "r" and "l" indicate that we have two columns with right and left aligned entries and no lines dividing cells or around the table. I can make the table look more like a spreadsheet by doing

```
\begin{table}[!ht]
\begin{center}
\begin{tabular}{|r|l|}
\hline
  {\textnormal{District}} &
  {\textnormal{Population}}\\ \hline
   Applewood & 8280 \\ \hline
   Boxwood & 4600  \\ \hline
   Central & 5220\\ \hline
   \end{tabular}\caption{Our first table again}
   \end{center}
\end{table}
```

District	Population
Applewood	8280
Boxwood	4600
Central	5220

Table 3.2: Our first table again

Here is a more complicated example of a table.

```
\begin{table}[!ht]
\centerline{
\begin{tabular}{|l||r|r|r|r|} \hline
\emph{Reprojection} & \multicolumn{3}{|c|}{\emph{Largest
 Reduction of Curvature}}
  & \emph{Average} \\ \cline{2-4}
\emph{Method} & \emph{Original} & \emph{Reprojected} &
 \emph{at} &
  \emph{Reduction} \\
 & \emph{Curvature} & \emph{Curvature} &
  \emph{Rotation} & \emph{of Curvature} \\
  \hline \hline
ZEEL & 0.0358 & 0.0245 &
 $\degree{45}$ & 0.0050 \\ \hline
```

```
ZEEL ext.\ & 0.0358 & 0.0245 &
 $\degree{45}$ & 0.0059 \\ \hline
Regridding & 0.0428 & 0.0166 &
 $\degree{75}$ & 0.0159 \\ \hline
Block & 0.0358 & 0.0103 &
 $\degree{45}$ & 0.0163 \\ \hline
\end{tabular}}
\caption{Reduction of curvature by each
 reprojection method\label{tbl:kreduce}}
\end{table}
```

Reprojection Method	*Largest Reduction of Curvature*			*Average Reduction of Curvature*
	Original Curvature	*Reprojected Curvature*	*at Rotation*	
ZEEL	0.0358	0.0245	45°	0.0050
ZEEL ext.	0.0358	0.0245	45°	0.0059
Regridding	0.0428	0.0166	75°	0.0159
Block	0.0358	0.0103	45°	0.0163

Table 3.3: Reduction of curvature by each reprojection method

Please refer to Chapter 6 of **?**] for a complete discussion of tables and tabular environments.

Chapter 4

Working with bibliographies and indicies

I would highly recommend that you use BibTEX to create your bibliography. BibTEX processes a special .bib file. The .bib file is where you enter your bibliographic information. A sample entry looks something like

```
@article{feu02,
author= {Thomas~Feuerstack},
title= {Introduction to pdf{\TeX{}}},
journal= {TUGboat},
volume= {23},
pages= {329--334},
number= {3/4},
url= {http://www.tug.org/TUGboat/Articles/tb23-3-4/tb75feu.pdf},
year= 2002}
```

or

```
@book{mgbcr04,
author= {Frank~Mittelbach and Michel~Goossens and
Johannes~Braams and David~Carlisle and Chris~Rowley},
title= {The \LaTeX\ Companion},
publisher= {Addison Wesley Professional},
edition= {2nd},
address= {New York},
year= 2004}
```

For a Web site I would recommend the following

```
@misc{brei04,
author =  {Jon~Breitenbucher},
title =  {{W}ooster related {L}a{T}e{X} files},
url =  {http://jbreitenbuch.wooster.edu/~jonb/latex/},
howpublished= {World Wide Web},
year= 2004,
note =  {Accessed on 03/11/2004}}
```

You can make a reference by typing `\citet{mgbcr04}` to produce **?**]. Other forms for citation include `\citep{mgbcr04}` or `\citeauthor {mgbcr04}` to produce [**?**] or **?** respectively. You can consult **?**] or **?**] to find out how to format entries in the .bib file and what options each reference type has.[1]

Indicies are also relatively easy to create. If I wanted to have Wooster show up in the index, I would enter `Wooster\index{Wooster}` in my source file. I could create a subentry for User Services by entering `User Services\index{Wooster!User Services}`. A subsubentry for Help Desk would be entered as `\index{Wooster!User Services!Help Desk}`.

To create the index one needs to make sure to uncomment the `\makeindex` command in the `username.tex` file. One also needs to uncomment the makeidx entry in the `styles/packages.tex` file and then run the Makeindex program. Consult **?**] or **?**] for further information.

[1] You could also use footnotes if your department called for that.

APPENDIX A

Typesetting Mathematical Formulae

This appendix is taken from **?**] under the GNU open source documentation license. This appendix addresses the main strength of TEX: mathematical typesetting. But be warned, this appendix only scratches the surface. While the things explained here are sufficient for many people, don't despair if you can't find a solution to your mathematical typesetting needs here. It is highly likely that your problem is addressed in AMS-LATEX[1] or some other package.

A.1 General

LATEX has a special mode for typesetting mathematics. Mathematical text within a paragraph is entered between `\(` and `\)`, between `$` and `$` or between `\begin{math}` and `\end{math}`.

```
Add $a$ squared and $b$ squared
to get $c$ squared. Or, using
a more mathematical approach:
$c^{2}=a^{2}+b^{2}$
```

Add a squared and b squared to get c squared. Or, using a more mathematical approach: $c^{2}=a^{2}+b^{2}$

```
\TeX{} is pronounced as
$\tau\epsilon$.\\[6pt]
100~m$^{3}$ of water\\[6pt]
This comes from my $\heartsuit$
```

TEX is pronounced as $\tau\epsilon$.

100 m^{3} of water

This comes from my $\heartsuit$

It is preferable to *display* larger mathematical equations or formulae, rather than to typeset them on separate lines. This means you enclose them in `\[` and `\]` or between `\begin{displaymath}` and `\end{displaymath}`. This produces formulae which are not numbered. If you want LATEX to number them, you can use the equation environment.

[1] `CTAN:/tex-archive/macros/latex/packages/amslatex`

```
Add $a$ squared and $b$ squared
to get $c$ squared. Or, using
a more mathematical approach:
\begin{displaymath}
c^{2}=a^{2}+b^{2}
\end{displaymath}
And just one more line.
```

Add a squared and b squared to get c squared. Or, using a more mathematical approach:

$$c^2 = a^2 + b^2$$

And just one more line.

You can reference an equation with `\label` and `\ref`

```
\begin{equation} \label{eq:eps}
\epsilon > 0
\end{equation}
From (\ref{eq:eps}), we gather
\ldots
```

$$\epsilon > 0 \tag{A.1}$$

From (A.1), we gather . . .

Note that expressions will be typeset in a different style if displayed:

```
$\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}$
```

$\lim_{n \to \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$

```
\begin{displaymath}
\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}
\end{displaymath}
```

$$\lim_{n \to \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

There are differences between *math mode* and *text mode*. For example in *math mode*:

1. Most spaces and linebreaks do not have any significance, as all spaces either are derived logically from the mathematical expressions or have to be specified using special commands such as `\,`, `\quad`, or `\qquad`.

2. Empty lines are not allowed. Only one paragraph per formula.

3. Each letter is considered to be the name of a variable and will be typeset as such. If you want to typeset normal text within a formula (normal upright font and normal spacing) then you have to enter the text using the `\textrm{...}` commands.

```
\begin{equation}
\forall x \in \mathbf{R}:
\qquad x^{2} \geq 0
\end{equation}
```

$$\forall x \in \mathbf{R}: \qquad x^{2} \geq 0 \tag{A.2}$$

```
\begin{equation}
x^{2} \geq 0\qquad
\textrm{for all }x\in\mathbf{R}
\end{equation}
```

$$x^2 \geq 0 \qquad \text{for all } x \in \mathbf{R} \tag{A.3}$$

Mathematicians can be very fussy about which symbols are used: it would be conventional here to use 'blackboard bold', bold symbols which is obtained using `\mathbb` from the package `amsfonts` or `amssymb`.

The last example becomes

```
\begin{displaymath}
x^{2} \geq 0\qquad
\textrm{for all }x\in\mathbb{R}
\end{displaymath}
```

$$x^2 \geq 0 \qquad \text{for all } x \in \mathbb{R}$$

A.2 Grouping in Math Mode

Most math mode commands act only on the next character. So if you want a command to affect several characters, you have to group them together using curly braces: `{...}`.

```
\begin{equation}
a^x+y \neq a^{x+y}
\end{equation}
```

$$a^x + y \neq a^{x+y} \tag{A.4}$$

A.3 Building Blocks of a Mathematical Formula

In this section, the most important commands used in mathematical typesetting will be described. Take a look at **?**] for a detailed list of commands for typesetting mathematical symbols.

Lowercase Greek letters are entered as `\alpha`, `\beta`, `\gamma`, ..., uppercase letters are entered as `\Gamma`, `\Delta`, ...[2]

```
$\lambda,\xi,\pi,\mu,\Phi,\Omega$
```

$\lambda, \xi, \pi, \mu, \Phi, \Omega$

Exponents and Subscripts can be specified using the `^` and the `_` character.

[2]There is no uppercase Alpha defined in $\LaTeX\,2_\varepsilon$ because it looks the same as a normal roman A. Once the new math coding is done, things will change.

```
$a_{1}$ \qquad $x^{2}$ \qquad
$e^{-\alpha t}$ \qquad
$a^{3}_{ij}$\\
$e^{x^2} \neq {e^x}^2$
```

a_{1} $\qquad$ x^{2} $\qquad$ $e^{-\alpha t}$ $\qquad$ a^{3}_{ij}
$e^{x^2} \neq {e^x}^2$

The **square root** is entered as `\sqrt`, the n^{th} root is generated with `\sqrt[`n`]`. The size of the root sign is determined automatically by LaTeX. If just the sign is needed, use `\surd`.

```
$\sqrt{x}$ \qquad
$\sqrt{ x^{2}+\sqrt{y} }$
\qquad $\sqrt[3]{2}$\\[3pt]
$\surd[x^2 + y^2]$
```

$\sqrt{x}$ $\qquad$ $\sqrt{ x^{2}+\sqrt{y} }$ $\qquad$ $\sqrt[3]{2}$
$\surd[x^2 + y^2]$

The commands `\overline` and `\underline` create **horizontal lines** directly over or under an expression.

```
$\overline{m+n}$
```

$\overline{m+n}$

The commands `\overbrace` and `\underbrace` create long **horizontal braces** over or under an expression.

```
$\underbrace{ a+b+\cdots+z }_{26}$
```

$\underbrace{ a+b+\cdots+z }_{26}$

To add mathematical accents such as small arrows or tilde signs to variables, you can use the commands given in **?**]. Wide hats and tildes covering several characters are generated with `\widetilde` and `\widehat`. The `'` symbol gives a prime.

```
\begin{displaymath}
y=x^{2}\qquad y'=2x\qquad y''=2
\end{displaymath}
```

$$y=x^{2}\qquad y'=2x\qquad y''=2$$

Vectors often are specified by adding a small arrow symbol on top of a variable. This is done with the `\vec` command. The two commands `\overrightarrow` and `\overleftarrow` are useful to denote the vector from A to B.

```
\begin{displaymath}
\vec a\quad\overrightarrow{AB}
\end{displaymath}
```

$$\vec a\quad\overrightarrow{AB}$$

Names of log-like functions are often typeset in an upright font and not in italic like variables. Therefore LaTeX supplies the following commands to typeset the most important function names:

```
\arccos   \cos    \csc   \exp   \ker     \limsup  \min   \sinh
\arcsin   \cosh   \deg   \gcd   \lg      \ln      \Pr    \sup
\arctan   \cot    \det   \hom   \lim     \log     \sec   \tan
\arg      \coth   \dim   \inf   \liminf  \max     \sin   \tanh
```

```
\[\lim_{x \rightarrow 0}
\frac{\sin x}{x}=1\]
```

$$\lim_{x \rightarrow 0} \frac{\sin x}{x}=1$$

For the modulo function, there are two commands: `\bmod` for the binary operator "$a \bmod b$" and `\pmod` for expressions such as "$x \equiv a \pmod{b}$."

A built-up **fraction** is typeset with the `\frac{...}{...}` command. Often the slashed form 1/2 is preferable, because it looks better for small amounts of 'fraction material.'

```
$1\frac{1}{2}$~hours
\begin{displaymath}
\frac{ x^{2} }{ k+1 }\qquad
x^{ \frac{2}{k+1} }\qquad
x^{ 1/2 }
\end{displaymath}
```

$1\frac{1}{2}$ hours

$$\frac{ x^{2} }{ k+1 }\qquad x^{ \frac{2}{k+1} }\qquad x^{ 1/2 }$$

To typeset binomial coefficients or similar structures, you can use either the command `\binom`{*num*}{*denom*} or `\genfrac`{*ldelim*}{*rdelim*}{*thickness*}{*style*}{*num*}{*denom*}. The second command can be used to produce customized fraction like output and more information can be found in **?**].

```
\begin{displaymath}
\binom{n}{k}\qquad
\genfrac{}{}{0pt}{}{x}{y+2}
\end{displaymath}
```

$$\binom{n}{k}\qquad \genfrac{}{}{0pt}{}{x}{y+2}$$

The **integral operator** is generated with `\int`, the **sum operator** with `\sum`. The upper and lower limits are specified with `^` and `_` like subscripts and superscripts.

```
\begin{displaymath}
\sum_{i=1}^{n} \qquad
\int_{0}^{\frac{\pi}{2}} \qquad
\end{displaymath}
```

$$\sum_{i=1}^{n} \qquad \int_{0}^{\frac{\pi}{2}} \qquad$$

For **braces** and other delimiters, there exist all types of symbols in TEX (e.g. $[\langle \| \updownarrow$). Round and square braces can be entered with the corresponding keys, curly braces with `\{`, all other delimiters are generated with special commands (e.g. `\updownarrow`). For a list of all delimiters available, check **?**].

```
\begin{displaymath}
{a,b,c}\neq\{a,b,c\}
\end{displaymath}
```

$$a,b,c \neq \{a,b,c\}$$

If you put the command `\left` in front of an opening delimiter or `\right` in front of a closing delimiter, TEX will automatically determine the correct size of the delimiter. Note that you must close every `\left` with a corresponding `\right`, and that the size is determined correctly only if both are typeset on the same line. If you don't want anything on the right, use the invisible '`\right .`'!

```
\begin{displaymath}
1 + \left( \frac{1}{ 1-x^{2} }
    \right) ^3
\end{displaymath}
```

$$1 + \left(\frac{1}{1-x^{2}} \right)^3$$

In some cases it is necessary to specify the correct size of a mathematical delimiter by hand, which can be done using the commands `\big`, `\Big`, `\bigg` and `\Bigg` as prefixes to most delimiter commands.[3]

```
$\Big( (x+1) (x-1) \Big) ^{2}$\\
$\big(\Big(\bigg(\Bigg($\quad
$\big\}\Big\}\bigg\}\Bigg\}$\quad
$\big\|\Big\|\bigg\|\Bigg\|$
```

$\Big((x+1) (x-1) \Big) ^{2}$
$\big(\Big(\bigg(\Bigg($ $\quad$ $\big\}\Big\}\bigg\}\Bigg\}$ $\quad$ $\big\|\Big\|\bigg\|\Bigg\|$

To enter **three dots** into a formula, you can use several commands. `\ldots` typesets the dots on the baseline, `\cdots` sets them centered. Besides that, there are the commands `\vdots` for vertical and `\ddots` for diagonal dots. You can find another example in section A.5.

```
\begin{displaymath}
x_{1},\ldots,x_{n} \qquad
x_{1}+\cdots+x_{n}
\end{displaymath}
```

$$x_{1},\ldots,x_{n} \qquad x_{1}+\cdots+x_{n}$$

[3] These commands do not work as expected if a size changing command has been used, or the `11pt` or `12pt` option has been specified. Use the exscale or amsmath packages to correct this behaviour.

A.4 Math Spacing

If the spaces within formulae chosen by TeX are not satisfactory, they can be adjusted by inserting special spacing commands. There are some commands for small spaces: `\,` for $\frac{3}{18}$ quad (␣), `\:` for $\frac{4}{18}$ quad (␣) and `\;` for $\frac{5}{18}$ quad (␣). The escaped space character `\␣` generates a medium sized space and `\quad` (␣) and `\qquad` (␣) produce large spaces. The size of a quad corresponds to the width of the character 'M' of the current font. The `\!` command produces a negative space of $-\frac{3}{18}$ quad (␣).

```
\newcommand{\rd}{\mathrm{d}}
\begin{displaymath}
\int\!\!\!\int_{D} g(x,y)
  \, \rd x\, \rd y
\end{displaymath}
instead of
\begin{displaymath}
\int\int_{D} g(x,y)\rd x \rd y
\end{displaymath}
```

$$\int\!\!\!\int_{D} g(x,y)\, \mathrm{d} x\, \mathrm{d} y$$

instead of

$$\int\int_{D} g(x,y)\mathrm{d} x \mathrm{d} y$$

Note that 'd' in the differential is conventionally set in roman.

AMS-LaTeX provides another way for fine tuning the spacing between multiple integral signs, namely the `\iint`, `\iiint`, `\iiiint`, and `\idotsint` commands. With the `amsmath` package loaded, the above example can be typeset this way:

```
\newcommand{\rd}{\mathrm{d}}
\begin{displaymath}
\iint_{D} \, \rd x \, \rd y
\end{displaymath}
```

$$\iint_{D} \, \mathrm{d} x \, \mathrm{d} y$$

See the electronic document testmath.tex (distributed with AMS-LaTeX) or Chapter 8 of "The LaTeX Companion"[4] for further details.

A.5 Vertically Aligned Material

To typeset **arrays**, use the `array` environment. It works somewhat similar to the `tabular` environment. The `\\` command is used to break the lines.

[4] available at `CTAN:/tex-archive/info/ch8.*`.

```
\begin{displaymath}
\mathbf{X} =
\left( \begin{array}{ccc}
x_{11} & x_{12} & \ldots \\
x_{21} & x_{22} & \ldots \\
\vdots & \vdots & \ddots
\end{array} \right)
\end{displaymath}
```

$$\mathbf{X} = \left(\begin{array}{ccc} x_{11} & x_{12} & \ldots \\ x_{21} & x_{22} & \ldots \\ \vdots & \vdots & \ddots \end{array} \right)$$

The `array` environment can also be used to typeset expressions which have one big delimiter by using a "`.`" as an invisible right delimiter:

```
\begin{displaymath}
y = \left\{ \begin{array}{ll}
 a & \textrm{if $d>c$}\\
 b+x & \textrm{in the morning}\\
 l & \textrm{all day long}
  \end{array} \right.
\end{displaymath}
```

$$y = \left\{ \begin{array}{ll} a & \textrm{if $d>c$}\\ b+x & \textrm{in the morning}\\ l & \textrm{all day long} \end{array} \right.$$

For formulae running over several lines or for equation systems, you can use the environments `eqnarray`, and `eqnarray*` instead of `equation`. In `eqnarray` each line gets an equation number. The `eqnarray*` does not number anything.

The `eqnarray` and the `eqnarray*` environments work like a 3-column table of the form `{rcl}`, where the middle column can be used for the equal sign or the not-equal sign. Or any other sign you see fit. The \\ command breaks the lines.

```
\begin{eqnarray}
f(x) & = & \cos x     \\
f'(x) & = & -\sin x   \\
\int_{0}^{x} f(y)dy &
 = & \sin x
\end{eqnarray}
```

$$
\begin{array}{rcll}
f(x) & = & \cos x & \text{(A.5)} \\
f'(x) & = & -\sin x & \text{(A.6)} \\
\int_{0}^{x} f(y)dy & = & \sin x & \text{(A.7)}
\end{array}
$$

Notice that the space on either side of the the equal signs is rather large. It can be reduced by setting `\setlength\arraycolsep{2pt}`, as in the next example.

Long equations will not be automatically divided into neat bits. The author has to specify where to break them and how much to indent. The following two methods are the most common ones used to achieve this.

```
{\setlength\arraycolsep{2pt}
\begin{eqnarray}\notag
\sin x & = & x -\frac{x^{3}}{3!}
     +\frac{x^{5}}{5!}-{}
                   \\\notag
 & & {}-\frac{x^{7}}{7!}+{}\cdots
\end{eqnarray}}
```

$$\begin{aligned}\sin x \;=\; x - \frac{x^3}{3!} + \frac{x^5}{5!} - \\ -\frac{x^7}{7!} + \cdots\end{aligned}$$

```
\begin{eqnarray}\notag
\lefteqn{ \cos x = 1
     -\frac{x^{2}}{2!} +{} }
                   \\\notag
 & & {}+\frac{x^{4}}{4!}
     -\frac{x^{6}}{6!}+{}\cdots
\end{eqnarray}
```

$$\begin{aligned}&\cos x = 1 - \frac{x^2}{2!} + \\ &\qquad + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots\end{aligned}$$

The `\notag` command causes LaTeX to not generate a number for this equation.

It can be difficult to get vertically aligned equations to look right with these methods; the package amsmath provides a more powerful set of alternatives.

A.6 Math Font Size

In math mode, TeX selects the font size according to the context. Superscripts, for example, get typeset in a smaller font. If you want to typeset part of an equation in roman, don't use the `\textrm` command, because the font size switching mechanism will not work, as `\textrm` temporarily escapes to text mode. Use `\mathrm` instead to keep the size switching mechanism active. But pay attention, `\mathrm` will only work well on short items. Spaces are still not active and accented characters do not work.[5]

```
\begin{equation}
2^{\textrm{nd}} \quad
2^{\mathrm{nd}}
\end{equation}
```

$$2^{\textrm{nd}} \quad 2^{\mathrm{nd}} \tag{A.8}$$

Nevertheless, sometimes you need to tell LaTeX the correct font size. In math mode, the font size is set with the four commands:

displaystyle (123), textstyle (123), scriptstyle ($\scriptstyle 123$) and scriptscriptstyle ($\scriptscriptstyle 123$).

Changing styles also affects the way limits are displayed.

[5]The $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-LaTeX package makes the textrm command work with size changing.

```
\begin{displaymath}
\mathop{\mathrm{corr}}(X,Y)=
 \frac{\displaystyle
   \sum_{i=1}^n(x_i-\overline x)
   (y_i-\overline y)}
  {\displaystyle\biggl[
 \sum_{i=1}^n(x_i-\overline x)^2
\sum_{i=1}^n(y_i-\overline y)^2
\biggr]^{1/2}}
\end{displaymath}
```

$$\mathrm{corr}(X,Y)=\frac{\displaystyle\sum_{i=1}^n(x_i-\overline x)(y_i-\overline y)}{\displaystyle\biggl[\sum_{i=1}^n(x_i-\overline x)^2\sum_{i=1}^n(y_i-\overline y)^2\biggr]^{1/2}}$$

This is one of those examples in which we need larger brackets than the standard `\left[ \right]` provides.

A.7 Theorems, Laws, . . .

When writing mathematical documents, you probably need a way to typeset "Lemmas", "Definitions", "Axioms" and similar structures. LaTeX supports this with the command

newtheorem{*name*}[*counter*]{*text*}[*section*]

The *name* argument, is a short keyword used to identify the "theorem". With the *text* argument, you define the actual name of the "theorem" which will be printed in the final document.

The arguments in square brackets are optional. They are both used to specify the numbering used on the "theorem". With the *counter* argument you can specify the *name* of a previously declared "theorem". The new "theorem" will then be numbered in the same sequence. The *section* argument allows you to specify the sectional unit within which you want your "theorem" to be numbered.

After executing the newtheorem command in the preamble of your document, you can use the following command within the document.

`\begin{`*name*`}[`*text*`]` This is my interesting theorem `\end{`*name*`}`

This should be enough theory. The following examples will hopefully remove the final remains of doubt and make it clear that the `\newtheorem` environment is way too complex to understand.

```
% definitions for the document
% preamble
\newtheorem{law}{Law}
\newtheorem{jury}[law]{Jury}
%in the document
\begin{law} \label{law:box}
Don't hide in the witness box
\end{law}
\begin{jury}[The Twelve]
It could be you! So beware and
see law~\ref{law:box}\end{jury}
\begin{law}No, No, No\end{law}
```

Law 1. *Don't hide in the witness box*

Jury 2 (The Twelve)**.** *It could be you! So beware and see law 1*

Law 3. *No, No, No*

The "Jury" theorem uses the same counter as the "Law" theorem. Therefore it gets a number which is in sequence with the other "Laws". The argument in square brackets is used to specify a title or something similar for the theorem.

```
\flushleft
\newtheorem{mur}{Murphy}[section]
\begin{mur}
If there are two or more
ways to do something, and
one of those ways can result
in a catastrophe, then
someone will do it.\end{mur}
```

Murphy A.7.1. *If there are two or more ways to do something, and one of those ways can result in a catastrophe, then someone will do it.*

The "Murphy" theorem gets a number which is linked to the number of the current section. You could also use another unit, for example chapter or subsection.

A.8 Bold symbols

It is quite difficult to get bold symbols in LaTeX; this is probably intentional as amateur typesetters tend to overuse them. The font change command `\mathbf` gives bold letters, but these are roman (upright) whereas mathematical symbols are normally italic. There is a `\boldmath` command, but *this can only be used outside mathematics mode*. It works for symbols too.

```
\begin{displaymath}
\mu, M \qquad \mathbf{M} \qquad
\mbox{\boldmath $\mu, M$}
\end{displaymath}
```

$$\mu, M \qquad \mathbf{M} \qquad \boldsymbol{\mu, M}$$

Notice that the comma is bold too, which may not be what is required.

The package `amsbsy` (included by `amsmath`) makes this much easier as it includes a `\boldsymbol` command.

```
\begin{displaymath}
\mu, M \qquad
\boldsymbol{\mu}, \boldsymbol{M}
\end{displaymath}
```

$$\mu, M \qquad \boldsymbol{\mu}, \boldsymbol{M}$$

A.9 List of Mathematical Symbols

In the following tables, you find all the symbols normally accessible from *math mode*.

To use the symbols listed in Tables A.12–A.17,[6] the package `amssymb` must be loaded in the preamble of the document and the AMS math fonts must be installed, on the system. If the AMS package and fonts are not installed, on your system, have a look at `CTAN:/tex-archive/macros/latex/required/amslatex`

Table A.1: Math Mode Accents.

$\hat{a}$	`\hat{a}`	$\check{a}$	`\check{a}`	$\tilde{a}$	`\tilde{a}`	$\acute{a}$	`\acute{a}`
$\grave{a}$	`\grave{a}`	$\dot{a}$	`\dot{a}`	$\ddot{a}$	`\ddot{a}`	$\breve{a}$	`\breve{a}`
$\bar{a}$	`\bar{a}`	$\vec{a}$	`\vec{a}`	$\widehat{A}$	`\widehat{A}`	$\widetilde{A}$	`\widetilde{A}`

Table A.2: Lowercase Greek Letters.

α	`\alpha`	θ	`\theta`	o	`o`	υ	`\upsilon`
β	`\beta`	ϑ	`\vartheta`	π	`\pi`	ϕ	`\phi`
γ	`\gamma`	ι	`\iota`	ϖ	`\varpi`	φ	`\varphi`
δ	`\delta`	κ	`\kappa`	ρ	`\rho`	χ	`\chi`
ϵ	`\epsilon`	λ	`\lambda`	ϱ	`\varrho`	ψ	`\psi`
ε	`\varepsilon`	μ	`\mu`	σ	`\sigma`	ω	`\omega`
ζ	`\zeta`	ν	`\nu`	ς	`\varsigma`		
η	`\eta`	ξ	`\xi`	τ	`\tau`		

Table A.3: Uppercase Greek Letters.

Γ	`\Gamma`	Λ	`\Lambda`	Σ	`\Sigma`	Ψ	`\Psi`
Δ	`\Delta`	Ξ	`\Xi`	Υ	`\Upsilon`	Ω	`\Omega`
Θ	`\Theta`	Π	`\Pi`	Φ	`\Phi`		

[6] These tables were derived from `symbols.tex` by David Carlisle and subsequently changed extensively as suggested by Josef Tkadlec.

Table A.4: Binary Relations.

You can produce corresponding negations by adding a `\not` command as prefix to the following symbols.

$<$	`<`	$>$	`>`	$=$	`=`
$\leq$	`\leq` or `\le`	$\geq$	`\geq` or `\ge`	$\equiv$	`\equiv`
$\ll$	`\ll`	$\gg$	`\gg`	$\doteq$	`\doteq`
$\prec$	`\prec`	$\succ$	`\succ`	$\sim$	`\sim`
$\preceq$	`\preceq`	$\succeq$	`\succeq`	$\simeq$	`\simeq`
$\subset$	`\subset`	$\supset$	`\supset`	$\approx$	`\approx`
$\subseteq$	`\subseteq`	$\supseteq$	`\supseteq`	$\cong$	`\cong`
$\sqsubset$	`\sqsubset` [a]	$\sqsupset$	`\sqsupset` [a]	$\Join$	`\Join` [a]
$\sqsubseteq$	`\sqsubseteq`	$\sqsupseteq$	`\sqsupseteq`	$\bowtie$	`\bowtie`
$\in$	`\in`	$\ni$	`\ni ,\owns`	$\propto$	`\propto`
$\vdash$	`\vdash`	$\dashv$	`\dashv`	$\models$	`\models`
$\mid$	`\mid`	$\parallel$	`\parallel`	$\perp$	`\perp`
$\smile$	`\smile`	$\frown$	`\frown`	$\asymp$	`\asymp`
$:$	`:`	$\notin$	`\notin`	$\neq$	`\neq` or `\ne`

[a] Use the `latexsym` package to access this symbol

Table A.5: Binary Operators.

$+$	`+`	$-$	`-`	$\triangleleft$	`\triangleleft`
$\pm$	`\pm`	$\mp$	`\mp`	$\triangleright$	`\triangleright`
$\cdot$	`\cdot`	$\div$	`\div`	$\star$	`\star`
$\times$	`\times`	$\setminus$	`\setminus`	$\ast$	`\ast`
$\cup$	`\cup`	$\cap$	`\cap`	$\circ$	`\circ`
$\sqcup$	`\sqcup`	$\sqcap$	`\sqcap`	$\bullet$	`\bullet`
$\vee$	`\vee ,\lor`	$\wedge$	`\wedge ,\land`	$\diamond$	`\diamond`
$\oplus$	`\oplus`	$\ominus$	`\ominus`	$\uplus$	`\uplus`
$\odot$	`\odot`	$\oslash$	`\oslash`	$\amalg$	`\amalg`
$\otimes$	`\otimes`	$\bigcirc$	`\bigcirc`	$\dagger$	`\dagger`
$\bigtriangleup$	`\bigtriangleup`	$\bigtriangledown$	`\bigtriangledown`	$\ddagger$	`\ddagger`
$\lhd$	`\lhd` [a]	$\rhd$	`\rhd` [a]	$\wr$	`\wr`
$\unlhd$	`\unlhd` [a]	$\unrhd$	`\unrhd` [a]		

Table A.6: BIG Operators.

$\sum$	`\sum`	$\bigcup$	`\bigcup`	$\bigvee$	`\bigvee`	$\bigoplus$	`\bigoplus`
$\prod$	`\prod`	$\bigcap$	`\bigcap`	$\bigwedge$	`\bigwedge`	$\bigotimes$	`\bigotimes`
$\coprod$	`\coprod`	$\bigsqcup$	`\bigsqcup`			$\bigodot$	`\bigodot`
$\int$	`\int`	$\oint$	`\oint`			$\biguplus$	`\biguplus`

Table A.7: Arrows.

$\leftarrow$	`\leftarrow` or `\gets`	$\longleftarrow$	`\longleftarrow`	$\uparrow$	`\uparrow`
$\rightarrow$	`\rightarrow` or `\to`	$\longrightarrow$	`\longrightarrow`	$\downarrow$	`\downarrow`
$\leftrightarrow$	`\leftrightarrow`	$\longleftrightarrow$	`\longleftrightarrow`	$\updownarrow$	`\updownarrow`
$\Leftarrow$	`\Leftarrow`	$\Longleftarrow$	`\Longleftarrow`	$\Uparrow$	`\Uparrow`
$\Rightarrow$	`\Rightarrow`	$\Longrightarrow$	`\Longrightarrow`	$\Downarrow$	`\Downarrow`
$\Leftrightarrow$	`\Leftrightarrow`	$\Longleftrightarrow$	`\Longleftrightarrow`	$\Updownarrow$	`\Updownarrow`
$\mapsto$	`\mapsto`	$\longmapsto$	`\longmapsto`	$\nearrow$	`\nearrow`
$\hookleftarrow$	`\hookleftarrow`	$\hookrightarrow$	`\hookrightarrow`	$\searrow$	`\searrow`
$\leftharpoonup$	`\leftharpoonup`	$\rightharpoonup$	`\rightharpoonup`	$\swarrow$	`\swarrow`
$\leftharpoondown$	`\leftharpoondown`	$\rightharpoondown$	`\rightharpoondown`	$\nwarrow$	`\nwarrow`
$\rightleftharpoons$	`\rightleftharpoons`	$\iff$	`\iff` (bigger spaces)	$\leadsto$	`\leadsto` [a]

[a]Use the `latexsym` package to access this symbol

Table A.8: Delimiters.

$($	`(`	$)$	`)`	$\uparrow$	`\uparrow`	$\Uparrow$	`\Uparrow`
$[$	`[` or `\lbrack`	$]$	`]` or `\rbrack`	$\downarrow$	`\downarrow`	$\Downarrow$	`\Downarrow`
$\{$	`\{` or `\lbrace`	$\}$	`\}` or `\rbrace`	$\updownarrow$	`\updownarrow`	$\Updownarrow$	`\Updownarrow`
$\langle$	`\langle`	$\rangle$	`\rangle`	$\vert$	`|` or `\vert`	$\Vert$	`\\|` or `\Vert`
$\lfloor$	`\lfloor`	$\rfloor$	`\rfloor`	$\lceil$	`\lceil`	$\rceil$	`\rceil`
$/$	`/`	$\backslash$	`\backslash`		. (dual. empty)		

Table A.9: Large Delimiters.

$\lgroup$	`\lgroup`	$\rgroup$	`\rgroup`	$\lmoustache$	`\lmoustache`	$\rmoustache$	`\rmoustache`
$\arrowvert$	`\arrowvert`	$\Arrowvert$	`\Arrowvert`	$\bracevert$	`\bracevert`		

Table A.10: Miscellaneous Symbols.

$\dots$	`\dots`	$\cdots$	`\cdots`	$\vdots$	`\vdots`	$\ddots$	`\ddots`
$\hbar$	`\hbar`	$\imath$	`\imath`	$\jmath$	`\jmath`	ℓ	`\ell`
$\Re$	`\Re`	$\Im$	`\Im`	$\aleph$	`\aleph`	$\wp$	`\wp`
$\forall$	`\forall`	$\exists$	`\exists`	$\mho$	`\mho` [a]	∂	`\partial`
$'$	`'`	$\prime$	`\prime`	$\emptyset$	`\emptyset`	∞	`\infty`
∇	`\nabla`	$\triangle$	`\triangle`	$\Box$	`\Box` [a]	$\Diamond$	`\Diamond` [a]
$\bot$	`\bot`	$\top$	`\top`	$\angle$	`\angle`	$\surd$	`\surd`
$\diamondsuit$	`\diamondsuit`	$\heartsuit$	`\heartsuit`	$\clubsuit$	`\clubsuit`	$\spadesuit$	`\spadesuit`
$\neg$	`\neg` or `\lnot`	$\flat$	`\flat`	$\natural$	`\natural`	$\sharp$	`\sharp`

[a]Use the `latexsym` package to access this symbol

Table A.11: Non-Mathematical Symbols.

These symbols can also be used in text mode.

†	\dag	§	\S	©	\copyright
‡	\ddag	¶	\P	£	\pounds

Table A.12: AMS Delimiters.

$\ulcorner$	\ulcorner	$\urcorner$	\urcorner	$\llcorner$	\llcorner	$\lrcorner$	\lrcorner

Table A.13: AMS Greek and Hebrew.

$\digamma$	\digamma	$\varkappa$	\varkappa	$\beth$	\beth	$\daleth$	\daleth	$\gimel$	\gimel

Table A.14: AMS Binary Relations.

$\lessdot$	\lessdot	$\gtrdot$	\gtrdot	$\doteqdot$	\doteqdot or \Doteq
$\leqslant$	\leqslant	$\geqslant$	\geqslant	$\risingdotseq$	\risingdotseq
$\eqslantless$	\eqslantless	$\eqslantgtr$	\eqslantgtr	$\fallingdotseq$	\fallingdotseq
$\leqq$	\leqq	$\geqq$	\geqq	$\eqcirc$	\eqcirc
$\lll$	\lll or \llless	$\ggg$	\ggg or \gggtr	$\circeq$	\circeq
$\lesssim$	\lesssim	$\gtrsim$	\gtrsim	$\triangleq$	\triangleq
$\lessapprox$	\lessapprox	$\gtrapprox$	\gtrapprox	$\bumpeq$	\bumpeq
$\lessgtr$	\lessgtr	$\gtrless$	\gtrless	$\Bumpeq$	\Bumpeq
$\lesseqgtr$	\lesseqgtr	$\gtreqless$	\gtreqless	$\thicksim$	\thicksim
$\lesseqqgtr$	\lesseqqgtr	$\gtreqqless$	\gtreqqless	$\thickapprox$	\thickapprox
$\preccurlyeq$	\preccurlyeq	$\succcurlyeq$	\succcurlyeq	$\approxeq$	\approxeq

Table A.15: AMS Binary Relations Continued.

$\curlyeqprec$	\curlyeqprec	$\curlyeqsucc$	\curlyeqsucc	$\backsim$	\backsim
$\precsim$	\precsim	$\succsim$	\succsim	$\backsimeq$	\backsimeq
$\precapprox$	\precapprox	$\succapprox$	\succapprox	$\vDash$	\vDash
$\subseteqq$	\subseteqq	$\supseteqq$	\supseteqq	$\Vdash$	\Vdash
$\Subset$	\Subset	$\Supset$	\Supset	$\Vvdash$	\Vvdash
$\sqsubset$	\sqsubset	$\sqsupset$	\sqsupset	$\backepsilon$	\backepsilon
$\therefore$	\therefore	$\because$	\because	$\varpropto$	\varpropto
$\shortmid$	\shortmid	$\shortparallel$	\shortparallel	$\between$	\between
$\smallsmile$	\smallsmile	$\smallfrown$	\smallfrown	$\pitchfork$	\pitchfork
$\vartriangleleft$	\vartriangleleft	$\vartriangleright$	\vartriangleright	$\blacktriangleleft$	\blacktriangleleft
$\trianglelefteq$	\trianglelefteq	$\trianglerighteq$	\trianglerighteq	$\blacktriangleright$	\blacktriangleright

Table A.16: AMS Arrows.

$\dashleftarrow$	\dashleftarrow	$\dashrightarrow$	\dashrightarrow	$\multimap$	\multimap
$\leftleftarrows$	\leftleftarrows	$\rightrightarrows$	\rightrightarrows	$\upuparrows$	\upuparrows
$\leftrightarrows$	\leftrightarrows	$\rightleftarrows$	\rightleftarrows	$\downdownarrows$	\downdownarrows
$\Lleftarrow$	\Lleftarrow	$\Rrightarrow$	\Rrightarrow	$\upharpoonleft$	\upharpoonleft
$\twoheadleftarrow$	\twoheadleftarrow	$\twoheadrightarrow$	\twoheadrightarrow	$\upharpoonright$	\upharpoonright
$\leftarrowtail$	\leftarrowtail	$\rightarrowtail$	\rightarrowtail	$\downharpoonleft$	\downharpoonleft
$\leftrightharpoons$	\leftrightharpoons	$\rightleftharpoons$	\rightleftharpoons	$\downharpoonright$	\downharpoonright
$\Lsh$	\Lsh	$\Rsh$	\Rsh	$\rightsquigarrow$	\rightsquigarrow
$\looparrowleft$	\looparrowleft	$\looparrowright$	\looparrowright	$\leftrightsquigarrow$	\leftrightsquigarrow
$\curvearrowleft$	\curvearrowleft	$\curvearrowright$	\curvearrowright		
$\circlearrowleft$	\circlearrowleft	$\circlearrowright$	\circlearrowright		

Table A.17: AMS Negated Binary Relations and Arrows.

$\nless$	\nless	$\ngtr$	\ngtr	$\varsubsetneqq$	\varsubsetneqq
$\lneq$	\lneq	$\gneq$	\gneq	$\varsupsetneqq$	\varsupsetneqq
$\nleq$	\nleq	$\ngeq$	\ngeq	$\nsubseteqq$	\nsubseteqq
$\nleqslant$	\nleqslant	$\ngeqslant$	\ngeqslant	$\nsupseteqq$	\nsupseteqq
$\lneqq$	\lneqq	$\gneqq$	\gneqq	$\nmid$	\nmid
$\lvertneqq$	\lvertneqq	$\gvertneqq$	\gvertneqq	$\nparallel$	\nparallel
$\nleqq$	\nleqq	$\ngeqq$	\ngeqq	$\nshortmid$	\nshortmid
$\lnsim$	\lnsim	$\gnsim$	\gnsim	$\nshortparallel$	\nshortparallel
$\lnapprox$	\lnapprox	$\gnapprox$	\gnapprox	$\nsim$	\nsim
$\nprec$	\nprec	$\nsucc$	\nsucc	$\ncong$	\ncong
$\npreceq$	\npreceq	$\nsucceq$	\nsucceq	$\nvdash$	\nvdash
$\precneqq$	\precneqq	$\succneqq$	\succneqq	$\nvDash$	\nvDash
$\precnsim$	\precnsim	$\succnsim$	\succnsim	$\nVdash$	\nVdash
$\precnapprox$	\precnapprox	$\succnapprox$	\succnapprox	$\nVDash$	\nVDash
$\subsetneq$	\subsetneq	$\supsetneq$	\supsetneq	$\ntriangleleft$	\ntriangleleft
$\varsubsetneq$	\varsubsetneq	$\varsupsetneq$	\varsupsetneq	$\ntriangleright$	\ntriangleright
$\nsubseteq$	\nsubseteq	$\nsupseteq$	\nsupseteq	$\ntrianglelefteq$	\ntrianglelefteq
$\subsetneqq$	\subsetneqq	$\supsetneqq$	\supsetneqq	$\ntrianglerighteq$	\ntrianglerighteq
$\nleftarrow$	\nleftarrow	$\nrightarrow$	\nrightarrow	$\nleftrightarrow$	\nleftrightarrow
$\nLeftarrow$	\nLeftarrow	$\nRightarrow$	\nRightarrow	$\nLeftrightarrow$	\nLeftrightarrow

Table A.18: AMS Binary Operators.

$\dotplus$	\dotplus	$\centerdot$	\centerdot	$\intercal$	\intercal
$\ltimes$	\ltimes	$\rtimes$	\rtimes	$\divideontimes$	\divideontimes
$\Cup$	\Cup or \doublecup	$\Cap$	\Cap or \doublecap	$\smallsetminus$	\smallsetminus
$\veebar$	\veebar	$\barwedge$	\barwedge	$\doublebarwedge$	\doublebarwedge
$\boxplus$	\boxplus	$\boxminus$	\boxminus	$\circleddash$	\circleddash
$\boxtimes$	\boxtimes	$\boxdot$	\boxdot	$\circledcirc$	\circledcirc
$\leftthreetimes$	\leftthreetimes	$\rightthreetimes$	\rightthreetimes	$\circledast$	\circledast
$\curlyvee$	\curlyvee	$\curlywedge$	\curlywedge		

Table A.19: AMS Miscellaneous.

$\hbar$	\hbar	$\hslash$	\hslash	$\Bbbk$	\Bbbk
$\square$	\square	$\blacksquare$	\blacksquare	$\circledS$	\circledS
$\vartriangle$	\vartriangle	$\blacktriangle$	\blacktriangle	$\complement$	\complement
$\triangledown$	\triangledown	$\blacktriangledown$	\blacktriangledown	$\Game$	\Game
$\lozenge$	\lozenge	$\blacklozenge$	\blacklozenge	$\bigstar$	\bigstar
$\angle$	\angle	$\measuredangle$	\measuredangle	$\sphericalangle$	\sphericalangle
$\diagup$	\diagup	$\diagdown$	\diagdown	$\backprime$	\backprime
$\nexists$	\nexists	$\Finv$	\Finv	$\varnothing$	\varnothing
$\eth$	\eth	$\mho$	\mho		

Table A.20: Math Alphabets.

Example	Command	Required package
ABCdef	`\mathrm{ABCdef}`	
ABCdef	`\mathit{ABCdef}`	
$ABCdef$	`\mathnormal{ABCdef}`	
$\mathcal{ABC}$	`\mathcal{ABC}`	
$\mathcal{ABC}$	`\mathcal{ABC}`	`eucal` with option: or
	`\mathscr{ABC}`	`eucal` with option: mathscr
$\mathfrak{ABCdef}$	`\mathfrak{ABCdef}`	`eufrak`
$\mathbb{ABC}$	`\mathbb{ABC}`	`amsfonts` or `amssymb`

APPENDIX B

Examples of Java Code

Here are some examples of Java source using the `listings` package. I have entered the following before any code examples to format the code as shown.

```
\lstset{language=java}
\lstset{backgroundcolor=\color{white},rulecolor=\color{black}}
\lstset{linewidth=.95\textwidth,breaklines=true}
\lstset{commentstyle=\textit,stringstyle=\upshape,showspaces=false}
\lstset{frame = trbl, frameround=tttt}
\lstset{numbers=left,numberstyle=\tiny,basicstyle=\small}
\lstset{commentstyle=\normalfont\itshape,breakautoindent=true}
\lstset{abovecaptionskip=1.2\baselineskip,xleftmargin=30pt}
\lstset{framesep=6pt}
```

I have included the code by entering

```
\begin{singlespace}
\lstinputlisting[caption=Clock Code,label=clock]{source/Clock.java}
\end{singlespace}
```

```java
// file: Clock.java
public class Clock extends UpdateApplet {
    public void paint( java.awt.Graphics g ) {
        g.drawString( new java.util.Date().toString( ), 10, 25 );
    }
}
```

Listing B.1: Clock Code

```java
//file: Consumer.java
import java.util.Vector;

public class Consumer implements Runnable
{
    Producer producer;

    Consumer( Producer producer ) {
        this.producer = producer;
    }

    public void run() {
        while ( true ) {
            String message = producer.getMessage();
            System.out.println("Got message: "+ message);
                        try {
                                Thread.sleep( 2000 );
                        } catch ( InterruptedException e ) { }
        }
    }

    public static void main(String args[]) {
        Producer producer = new Producer();
        new Thread( producer ).start();
        Consumer consumer = new Consumer( producer );
        new Thread( consumer ).start();
    }
}
```

Listing B.2: Consumer

```java
//file: EvilEmpire.java
import java.net.*;

public class EvilEmpire {
  public static void main(String[] args) throws Exception{
    try {
      Socket s = new Socket("???.???.???.???", 80);
      System.out.println("Connected!");
    }
    catch (SecurityException e) {
      System.out.println("SecurityException: could not connect.");
    }
  }
}
```

Listing B.3: EvilEmpire Code

APPENDIX C

C++ Examples

This appendix demonstrates the `listings` packages ability to format C++ code.

```
#include "Motion.h"


Motion::Motion(int _steps) : TimeSeries(_steps) {}

Motion::Motion(Noise2 *_noise) : TimeSeries(_noise->GetSteps()) {
  noise = _noise;
}

Motion::~Motion() {
  delete noise;
}

void Motion::SyncWithNoise() {
  if (noise != NULL) {
    this->Initialize();
    double sum = 0;
    int getsteps = this->GetSteps();
    for (int i = 0; i < getsteps; i++) {
      sum += noise->GetData(i);
      this->SetData(i, sum);
    }
  } else {
    fprintf(stderr, "%s\n", MOTION_NOISE_ERR);
  }
}
```

Listing C.1: Motion Class

```
#include <unistd.h>
#include "Plotter.h"


void Plotter::MakePlot(char *filename) {
  ofstream fout(FILE_PLOT);
  fout << "set␣data␣style␣linespoints" << endl
       << "plot␣\"" << filename << "\"" << endl;
```

```
  fout.close();

  int pid, status;
  pid = fork();
  if (pid >= 0) {
    if (pid == 0) {
      execl(FILE_GNUPLOT, "gnuplot", "-persist", FILE_PLOT, NULL);
      fprintf(stderr, "%s \"gnuplot\"", EXEC_ERR);
      exit(0);
    } else {
      wait(status);
    }
  } else {
    fprintf(stderr, "%s \"gnuplot\"", FORK_ERR);
  }

  /*  pid = fork();
  if (pid >= 0) {
    if (pid == 0) {
      execlp("rm", FILE_PLOT, NULL);
      fprintf(stderr, "%s \"rm\"", EXEC_ERR);
      exit(0);
    } else {
      wait(status);
    }
  } else {
    fprintf(stderr, "%s \"rm\"", FORK_ERR);
    }*/

}
```

Listing C.2: Plotter Class

```
#include "Simulation.h"


Simulation::Simulation(int _steps, double H) {
  noise = new Noise2(_steps);
  motion = new Motion(noise);
}

Simulation::~Simulation() {
  delete noise;
  delete motion;
}

void Simulation::Analyze() {
  noiseplotter.MakePlot("noise");
  motionplotter.MakePlot("motion");
}
```

Listing C.3: Simulation Class

```
#include "Simulation.h"
```

```


Simulation::Simulation(int _steps, double H) {
  noise = new Noise2(_steps);
  motion = new Motion(noise);
}

Simulation::~Simulation() {
  delete noise;
  delete motion;
}

void Simulation::Analyze() {
  noiseplotter.MakePlot("noise");
  motionplotter.MakePlot("motion");
}
```

Listing C.4: Simulation Class

```
#include "Simulation.h"


Simulation::Simulation(int _steps, double H) {
  noise = new Noise2(_steps);
  motion = new Motion(noise);
}

Simulation::~Simulation() {
  delete noise;
  delete motion;
}

void Simulation::Analyze() {
  noiseplotter.MakePlot("noise");
  motionplotter.MakePlot("motion");
}
```

Listing C.5: Simulation Class

```
#include "Simulation.h"


Simulation::Simulation(int _steps, double H) {
  noise = new Noise2(_steps);
  motion = new Motion(noise);
}

Simulation::~Simulation() {
  delete noise;
  delete motion;
}

void Simulation::Analyze() {
  noiseplotter.MakePlot("noise");
```

```
    motionplotter.MakePlot("motion");
}
```

Listing C.6: Simulation Class

```
#include "Simulation.h"


Simulation::Simulation(int _steps, double H) {
    noise = new Noise2(_steps);
    motion = new Motion(noise);
}

Simulation::~Simulation() {
    delete noise;
    delete motion;
}

void Simulation::Analyze() {
    noiseplotter.MakePlot("noise");
    motionplotter.MakePlot("motion");
}
```

Listing C.7: Simulation Class

```
#include "Simulation.h"


Simulation::Simulation(int _steps, double H) {
    noise = new Noise2(_steps);
    motion = new Motion(noise);
}

Simulation::~Simulation() {
    delete noise;
    delete motion;
}

void Simulation::Analyze() {
    noiseplotter.MakePlot("noise");
    motionplotter.MakePlot("motion");
}
```

Listing C.8: Simulation Class

```
#include "Simulation.h"


Simulation::Simulation(int _steps, double H) {
    noise = new Noise2(_steps);
    motion = new Motion(noise);
}
```

```

Simulation::~Simulation() {
  delete noise;
  delete motion;
}

void Simulation::Analyze() {
  noiseplotter.MakePlot("noise");
  motionplotter.MakePlot("motion");
}
```

Listing C.9: Simulation Class

Afterword

So how does a LaTeX session work? LaTeX loads the document class with any specified options and uses the information in the document class to decide on how the document will be formatted. At this point LaTeX loads any packages that the user has specified. Packages extend the basic LaTeX commands and formatting for special situations. `woosterthesis` loads a number of packages by default and it is assumed you have these installed on your system. They are: `ifpdf`, `textpos`, `geometry`, `amsthm`, `amssymb`, `amsmath`, `setspace`, `fancyhdr`, `graphicx`, `eso-pic`, `listings`, `natbib`, `makeidx`, `verbatim`, `lettrine`, `alltt`, `fontenc`, `pxfonts`, `floatflt`, `float`, `caption`, `subfigure`, and `ifthen`. The `woosterthesis` class assumes you are using pdfTeX (support for postscript based TeX has been dropped as of 2006/17/11).

The `hyperref` package will make your thesis a linked document. `amsthm` is for altering the Theorem environments. `amsmath` implements almost all of the mathematical symbols. `amssymb` adds the mathematical symbols not present in `amsmath`. `graphicx` and `eso-pic` are used to place graphics files in the thesis. `geometry` is used to set up the margins for the thesis. `setspace` is used to alter spacing by allowing a `singlespace`, `doublespace`, and `onehalfspace` environments. `natbib` formats references in parentheses with author and year. Documentation is included for some of the packages in the `doc` folder.

These packages should all be installed with a full installation of TeXLive on OS X or XP. On OS X one can use the the MacTeX installer as i-Installer is no longer supported as of 2007/1/1. On XP/Vista one can use MikTeX to install all available packages which will install all of the above. By default the MikTeX install does a minimal installation. You will need to run the updater to make your MikTeX installation aware of all the new packages.

There is also a new TeX engine called XeTeX which allows one to use the native fonts on your system as text fonts in the document. More information can be found at the XeTeX homepage. If using XeTeX you will also need `fontspec` and `xltxtra` which should be installed with XeTeX .

Once the packages are loaded, LaTeX begins to process the commands contained between the `document` tags. As it processes the commands, a number of auxiliary files are created. These files contain information needed for things like the Bibliography, Table of Contents, List of Figures, etc. We then process the file a second time to allow LaTeX to use its auxiliary files to fill in information. Some information may require three passes before it is displayed. Once LaTeX is done you are presented with a PDF of the output.