# LaTeX for Physicists

Joshua B. Teves

# Contents

# Preface

This document is designed to help you create professional-looking documents with the LaTeX typesetting engine. This method of preparing documents is popular for a number of reasons. In the physics community, it is easy to typeset mathematical formulae and include figures. Many journals, such as those published by the American Physical Society (APS), use LaTeX templates in order to typeset their articles as a result. In other communities, such as linguistics, LaTeX is the typesetting method of choice because it can accomodate a range of international characters (e.g., Greek letters). LaTeX is also free, which makes it agreeable to people of almost any persuasion.

There are two components of LaTeX: the language, and the typesetting engine. The language part is what you actually type, and the engine is the part that interprets the language. As a user, you really only have control over the language part, so that's the part this document focuses on. Nonetheless, some context for the engine is important.

One of the reasons that LaTeX makes such professional documents is automatic handling: the engine automatically keeps track of the number of chapter, sections, and pages you have. It knows how many characters to put on each line in order to keep the document easy to read, that it should hyphenate words spread over two lines, and the layout of the page that makes the most efficient use of space when placing figures. Fonts and font sizes are generally automatic, though you can override this. With the addition of the BibTeX bibliography system, you can use LaTeX to keep track of all of your citations and generate a bibliography at the end. This makes it completely feasible to create a many-hundred page document on your own, and for free.

This document is fairly comprehensive. In "Document Structure," you'll learn how to make a first document. Then, you'll learn how to structure them in LaTeX's language. Then, "Mathematics" will show you how to typeset mathematical formulae and things such as matrices. This is particularly crucial for mathematicians and physicists. "Figures" discusses how to make figures, captions, and tables. The "Using RevTeX" chapter discusses how to use RevTeX, which is the LaTeX package for APS journals. The "BibTeX" chapter shows how to make a bibliography using BibTeX, which is extremely useful as you start citing papers, projects, etc.

Such a powerful tool unfortunately requires careful practice. When you learn physics, you must do exercises in order to learn how to do problems. Thus,

with LaTeX you must do exercises in order to learn how to typeset documents properly. As a result, I've included exercises in this document that appear at the ends of sections and chapters. When following along with examples or doing exercises, DO NOT just copy and paste. You won't pay close attention if you highlight and copy instead of typing it out. Even though it takes longer, you'll find that you're far more aware of what you're doing and make that time spent very effective. I hope that this guide and the included exercises will help teach you what you need to generate professional documents.

I regret that the document is not as comprehensive as I would like, due to the time constraints that accompany being a college senior. In spite of this, I think this manual should be an excellent foundation for those who have never used it, and contain some glimmers of insight for those who are veterans.

Best of luck,
Joshua B. Teves

# Chapter 1

# Organization

## 1.1 A First Document

### A Very First Document

To create your first LaTeX document, load an editor of your choice. For Windows users, this is likely MikTeX . For Mac users, this is likely TeX shop. With your editor loaded, go ahead and type in the following:

```
\documentclass{article}

\usepackage{lipsum}

\begin{document}

\lipsum

\end{document}
```

Example 1.1.1: Your first document!

Be sure to type it exactly. Then, hit the "Typeset" button (it looks like a play button on MikTeX). You should get something like in Figure 1.1.1. If not, double check that you typeset everything exactly as shown. If you still have bugs, read on and see if you can narrow it down to any of the possibilities I mention.

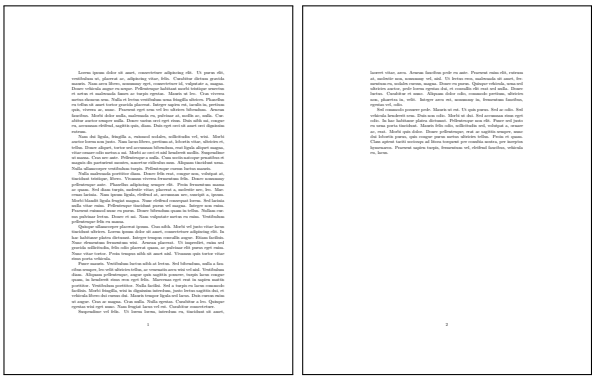

Figure 1.1.1: The result of typesetting Example 1.1.1.

Let's look at each part of the document we just made. First, we used a \ character, and typed the word begin. The \ character is how to issue commands in LaTeX. In this instance, the command was `documentclass`. In order for LaTeX to do it's job, it needs to know what kind of document it's about to make. That's what the curly braces { and } are for: they enclose what document class is going to be used. In this case, it's an article. Sometimes it might be a book, or a special kind of journal. Curly braces always enclose the *argument* of a command, which usually specificies a kind of information that a command needs. In any case, the `documentclass` command should be the first line of every LaTeX document you make. Note that since the \ character tells LaTeX it's issuing commands, if you use "\" in your document, LaTeX will think you're trying to issue it a command. Instead, to make a \ use `\textbackslash`.

That leads us the `usepackage` command. This simply tells LaTeX to use a "package," or a collection of LaTeX commands defined for you. The curly braces tell us we're using the `lipsum` package. When LaTeX sees this, it searches your computer for the `lipsum` package. If it finds it, then the package is included. If not, then LaTeX tells you that "lipsum.sty cannot be found." In this case, look up the package and install it. This applies to any other package: LaTeX will tell you that it cannot find the package, and you'll have to install it or, more likely, correct your spelling error.

Now that we've told LaTeX to use that package, we can tell LaTeX to begin the document. This is what the command `\begin{document}` does. The command `\begin` is actually an important one. This command tells LaTeX that you are starting a particular *environment*. An environment is a part of the document with certain rules. In this case, the environment is the `document`– in other words, the part that will actually get put on the screen. Any environment is ended with the `end` command, which is why the command `\end{document}` is at the end of our example. We're telling LaTeXthat the `document` part spans from the `\begin` to the `\end`. Everything before this environment is called the

*preamble*. Unlike the United States Constitution Preamble, nobody memorizes a LaTeX preamble (except perhaps yours truly). It is more of a "we, the things that the LaTeX engine needs to make a document" introduction than a "we, the people" kind of deal.

Now, what's the `\lipsum` command? That's where all of the text came from. The `\lipsum` command told LaTeXto fill in a very lengthy latin rant. it's often used as a filler text to see how something looks when you typeset it. Here, the point is that we can see how the text is printed out in our document. But that's an awfully annoying text. Plus, our example is missing several things. What's our document's name? Who wrote it? Can we say something that isn't in Latin? How did those paragraphs get there?

### My First Titled Document

First, get rid of the `lipsum` command. We want to give our document a title, so we're going to use the command `\title` where `\lipsum` used to be. Go ahead and title your document, "My First Document." We want people to know who you are, too, so use the `\author` command to tell LaTeX who you are on the next line down. Your resulting code should look like this:

```
\documentclass{article}

\begin{document}

\title{My First Document}
\author{Jimmy Dean: Sausage Master and Singer Extraordinaire}

\end{document}
```

Example 1.1.2: Putting a title and author in your document.

Go ahead and typeset that. When you finish, you should be greated by something surprising: precisely nothing.

Figure 1.1.2: Literally nothing

It turns out that LaTeX sometimes doesn't display things when you issue commands in the document. This is true of `\title` and `author`, as well as several others. In this case, we only told LaTeXwhat those values are, not to put them in the document. In order to get it to do that, we have to include the command `\maketitle`. Go ahead and add that below your `\author` command, like so:

```
\documentclass{article}
\begin{document}

\title{My First Document}
\author{Jimmy Dean: Sausage Master and Singer Extraordinaire}
\maketitle

\end{document}
```

Example 1.1.3: Making the title actually appear.

which should give you this:

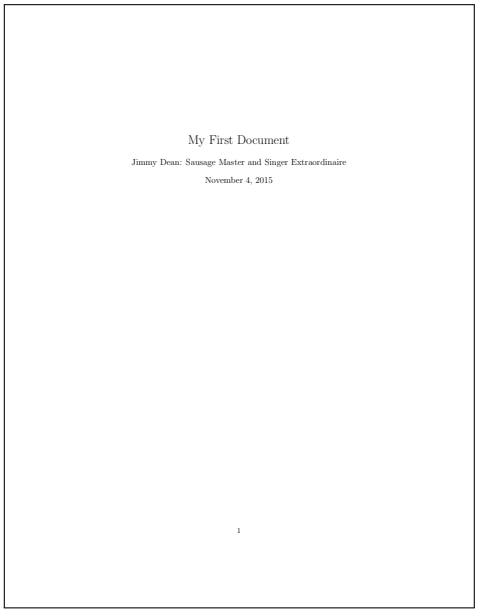


Figure 1.1.3: Example 1.1.3 result.

Hang on, that's weird. The date is also printed on our document. Why is that? It turns out that for the `article` class, LaTeX actually talks to your computer and figures out the date, then includes it as part of your title. Let's suppose for a moment that maybe you're typesetting your homework, and you'd really like it if the date on the paper wasn't the same as the date it's due. Then, before you issue the `\maketitle` command, you can issue the `\date` commmand with whatever you want inside. See here:

```
\documentclass{article}
\begin{document}

\title{My First Document}
\author{Jimmy Dean: Sausage Master and Singer Extraordinaire}
\date{Marchtember One-teenth}
\maketitle

\end{document}
```

Example 1.1.4: Changing the date to something more interesting.

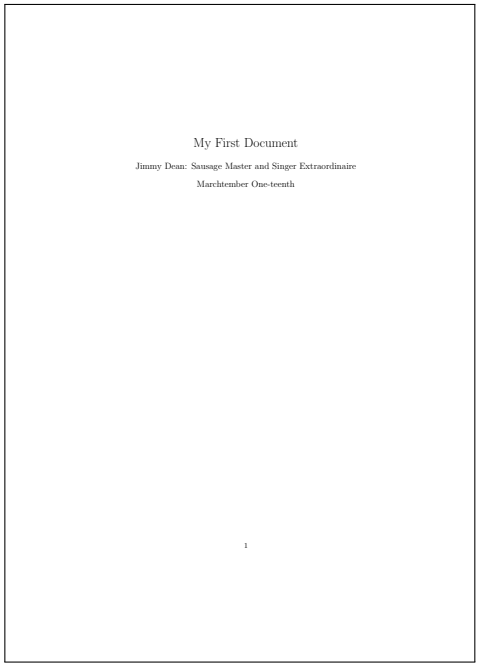


Figure 1.1.4: Example 1.1.4 result.

### My First Document with a Body

Alternatively, if you want to leave the date out of it, you can just put nothing between the curly braces. LaTeX will then interpret the date as being a blank, and not include it in the title.

This title looks very, very nice and all, but most people like their documents to have content. Let's add some. Below your `\maketitle` command, skip a line, and then just type the blurb,

"I like big blurbs and I cannot lie. You other brothers can't deny, that when a blurb walks in with an itty bitty meaning, and a dumb thing in in your face, you get annoyed. You want the blurb to stop talking now."

My recommendation is to split things into separate lines by punctuation: in other words, hit enter after each comma, period, semicolon, or whatever. LaTeX mostly ignores "white space," like when you hit tab or when you split things across lines. This keeps everything to a limited number of characters per line, which is way easier to edit when you do something wrong. If you follow those guidelines, you should end up with something like this:

```
\documentclass{article}
\begin{document}

\title{My First Document}
\author{Jimmy Dean: Sausage Master and Singer Extraordinaire}
\date{Marchtember One-teenth}
\maketitle

I like big blurbs and I cannot lie.
You other brothers can't deny,
that when a blurb walks in with an itty bitty meaning,
and a dumb thing in in your face,
you get annoyed.
You want the blurb to stop talking now.

\end{document}
```

Example 1.1.5: Changing the date to something more interesting.

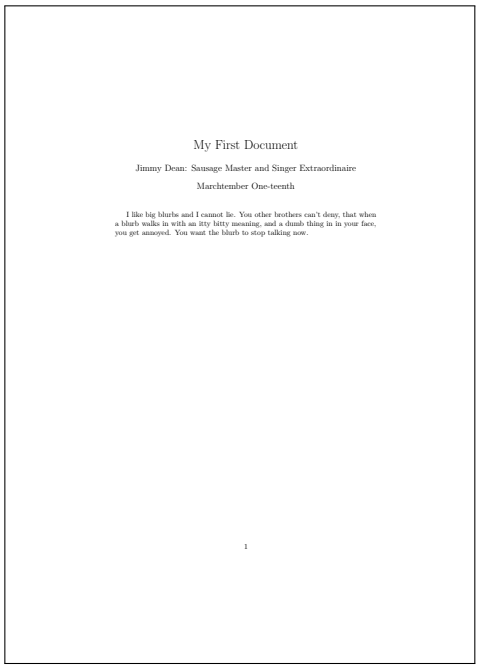


Figure 1.1.5: Example 1.1.5 result.

Now I image that you sometimes want more than one paragraph to show. When you want LaTeX to make a paragraph break, use the command `\par`. Usually when I use that command, I place a blank line after the previous paragraph, type `par`, and then start the new paragraph on the line below. This makes it really easy to see paragraph breaks in the source code, and again, does not affect the outcome of the final document. Go ahead and add a new paragraph, following Example 1.1.6. (Note: we don't need to tell LaTeX to make a new paragraph after the `\maketitle` command because LaTeX is smart enough to know that it needs to indent after making the title.)

```
\documentclass{article}
\begin{document}

\title{My First Document}
\author{Jimmy Dean: Sausage Master and Singer Extraordinaire}
\date{Marchtember One-teenth}
\maketitle

I like big blurbs and I cannot lie.
You other brothers can't deny,
that when a blurb walks in with an itty bitty meaning,
and a dumb thing in in your face,
you get annoyed.
You want the blurb to stop talking now.

\par
You won't believe this one simple trick with software!
Turn messes of Microsoft Word into beautiful code
with this one free program!
Typeset formulae so beautiful that calligraphers are jealous
with this non-plastic latex!
We're totally not BuzzFeed!

\end{document}
```

Example 1.1.6: Two paragraphs in one document.

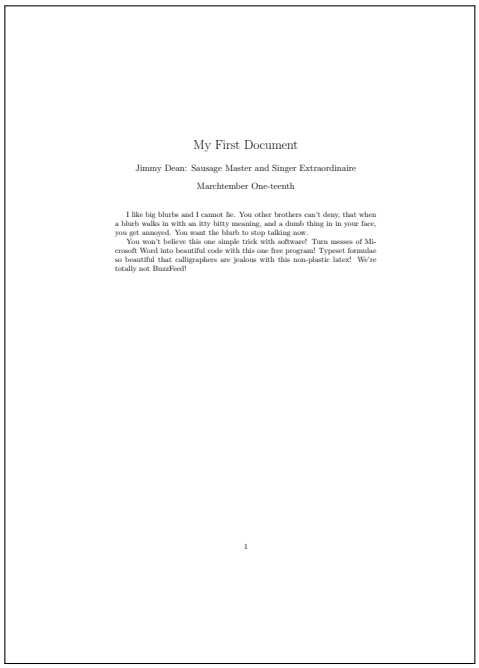


Figure 1.1.6: Example 1.1.6 result.

These are the basics, and we have our first document. Good job! Be sure to at least read Exercises 3 and 4 for an explanation of the `verbatim` environment. (It is preferable, of course, that you do the Exercises).

## Exercises

1. Create a new document with the name, "My Second Document," with your name as the author. Don't specify a date.

2. Create a new document with the name, "Paragraph Practice." Put "Para Graph" as the author, and make sure that the document's date does not appear. Write three paragraphs in your document.

3. Recreate the text of the first subsection. In order to get the computer-looking font, use the command `\verb`, and enclose what you want to appear with + signs. (e.g., `\verb+computer font+` gives `computer font`). This actually makes LaTeX interpret everything as plain text and then put it in the computery looking font. Ignore figures and examples.

4. In addition to the `\verb` command, there is a `verbatim` environment. Unlike `\verb`, it doesn't stay in-line, and it is an environment instead of a command. Recall that you start an environment with `\begin{environment_name}` and end it with `\end{environment_name)`. Using this, create a document

that has a normal paragraph, followed by that paragraph's source code in the `verbatim` environment.

## 1.2 Document Structure

### Enumerate and Itemize

In addition to splitting things up by pagraph, you can use the environments `itemize` and `enumerate` in order to make clean-looking lists. The `itemize` environment will make a list where each item has a clear separation from the others, and a bullet point (or marker of your choice). We'll make a practice grocery list. (It is said that true masters of LaTeX make their grocery lists in it. By like one person. Okay, it's me).

```
\documentclass{article}

\begin{document}

\begin{itemize}
\item Vegan Butter
\item Vegan Milk
\item Vegan Eggs
\item Cashews. Lots of Cashews.
\end{itemize}

\end{document}
```

Example 1.2.1: A grocery list using the itemize environment.

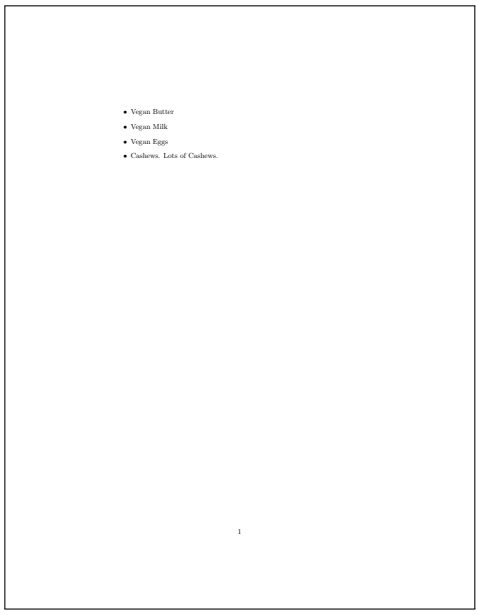


Figure 1.2.1: Example 1.2.1 result.

Note that each item is literally a command called `item`. I tabbed those out because all of the items are short and that makes it clear that those items belong to the `itemize` environment. Go ahead and try making your own grocery list.

In addition to plain items, we can actually specify the marker we want for each item. These are called *optional arguments*, since you don't need to tell LaTeX what marker to use if you're using the default. When you want to use optional arguments, use square brackets ([ and]) around the argument. In this case, I'll show you how to turn every item's marker into a dash instead of a bullet.

```
\documentclass{article}

\begin{document}

\begin{itemize}
\item[-] Vegan Butter
\item[-] Vegan Milk
\item[-] Vegan Eggs
\item[-] Cashews. Lots of Cashews.
\end{itemize}

\end{document}
```

Example 1.2.2: A grocery list using the itemize environment, now with dashes instead of bullets.

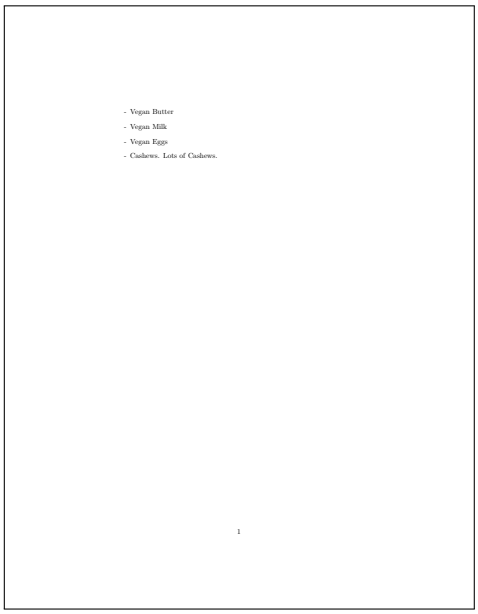


Figure 1.2.2: Example 1.2.2 result.

Now, maybe we want a numbered list. Take, for example, a count of students that showed up to a meeting. We can use the `enumerate` environment instead of the `itemize` environment, which actually counts the number for us. Since the counting happens under the hood, we can swap people around and LaTeX will do a recount for us– no extra work needed. This is particularly

handy. You use the same format, with each `item` the same as in itemize. In the following example, we make a list of five students: Average Joe, Slow Sally, Clever Clarissa, and Always-Late Amos.

```
\documentclass{article}

\begin{document}

\begin{enumerate}
\item Average Joe
\item Slow Sally
\item Clever Clarissa
\item Always-Late Amos
\end{enumerate}

\end{document}
```

Example 1.2.3: A grocery list using the itemize environment, now with dashes instead of bullets.

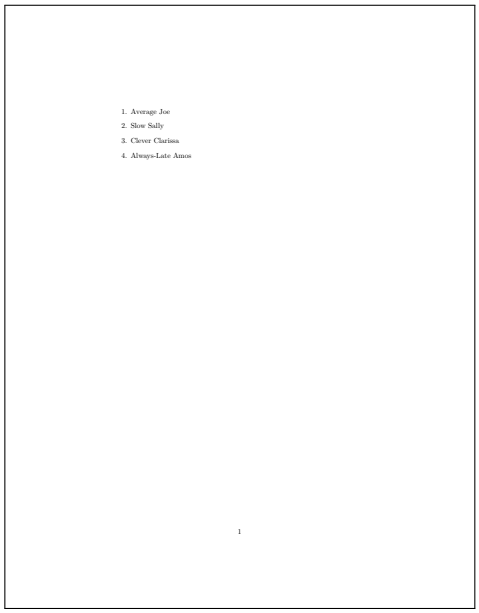


Figure 1.2.3: Example 1.2.3 result.

These environments should give you a reasonably good start for making lists

and numbering things. You can find more online. Truthfully,though, keeping it simple is best.

### Sectioning

You may have noticed that in this document, there have been chapter, section, and example/figure numbers. This would be appalling to keep track of without LaTeX's engine. Fortunately, however, LaTeXhas several commands to help you keep track of everything. The engine and language assume the following hierarchy, with the lower-numbered levels being broader:

1. Parts (book classes only)
2. Chapters (book classes only)
3. Sections
4. Subsections
5. Subsubsections

Note that Parts and Chapters are only used for book classes, since adding that level of scope to other document classes is simply wasteful. In order to tell LaTeX that you want to make a new segment of the document at one of these levels, you can simply use the command that contains its name. For example, making a new section just requires the `\section{}` command, with the name of the section between the curly braces.

The point of using these sectioning tools is to guide the reader through the material. In scientific papers, the sections are often the:

1. Abstract, an exceedingly brief summary of the article.
2. Introduction, which gives context for the article.
3. Methods, which explains how a study was conducted.
4. Analysis, which explains what the study did with its results.
5. Conclusions, which recapitulates what the author(s) think(s) most important.
6. Acknowledgments, which thanks people who helped with the study and funding agencies.

For an example, we'll write a mock scientific paper in the standard `article` class. That means we'll need to make each item above a section. A skeletal outline would look something like:

```
\documentclass{article}

\begin{document}

\title{Scientific Article}
\author{A Real Scientist}

\maketitle

\section{Abstract}

\section{Introduction}

\section{Analysis}

\section{Conclusions}

\section{Acknowledgments}

\end{document}
```

Example 1.2.4: A skeletal outline of a scientific document.

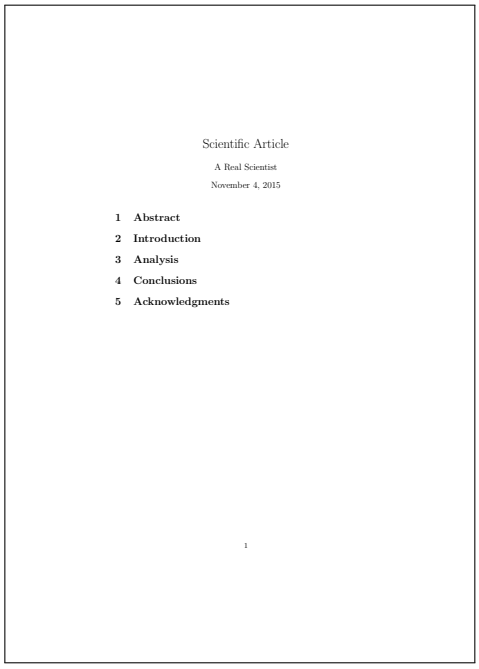


Figure 1.2.4: Example 1.2.4 result.

Note that each section is numbered and forces a line break and some space before the next section. Maybe we want to add subsections to be a little more descriptive. For the Introduction, let's say you'll have several parts:

1. This is Cool Stuff

2. We Promise it Matters

3. These Other Studies Say it Does, Too

4. So We Did a Thing

To add them to our document as subsections, we just have to add them as subsections via the `\subsection{}` command with their names in between the curly braces. Just so you're aware, some TEXnicians like to tab before they type in the `subsection` command in their source LATEX code. I don't, because then everything is compressed to the left.

```
\documentclass{article}

\begin{document}

\title{Scientific Article}
\author{A Real Scientist}

\maketitle

\section{Abstract}

\section{Introduction}

\subsection{This is Cool Stuff}

\subsection{We Promise it Matters}

\subsection{These Other Studies Say it Does, Too}

\subsection{So We Did a Thing}

\section{Analysis}

\section{Conclusions}

\section{Acknowledgments}

\end{document}
```

Example 1.2.5: A slightly less skeletal outline of a scientific document.

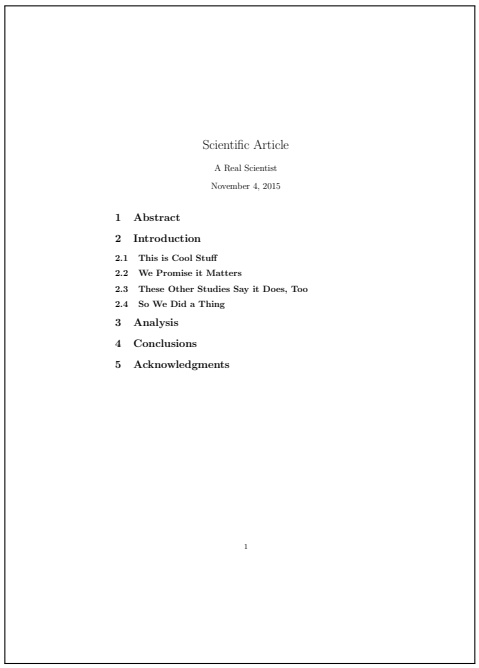


Figure 1.2.5: Example 1.2.5 result.

Note that the subsections are also numbered, with the section number being the first number, separated from the subsection number by a period. If no you want something to not be numbered, just put a star by the command, like `\section*{Don't Number Me}`.

To show you how far the scope of this extends, we'll have to make a book. This means that instead of the `article` class that we've been using, we're going to switch to the `book` class. Follow Example 1.2.6 to get yourself started.

```
\documentclass{book}

\begin{document}

\title{My First Book}
\author{A Published Author}

\maketitle

\chapter{The First One}

\section{The First Piece}

\section{A Different Piece}

\subsection{A Smaller Piece}

\subsection{An Even Smaller Piece}

\end{document}
```

Example 1.2.6: A slightly less skeletal outline of a scientific document.


Figure 1.2.6: Example 1.2.6 result. Note that there is blank page there. This is because it assumed odd pages will be on the left in a printing scenario, and the typesetting engine wants the chapter to start on the left side page.

Note that the chapters and subsections are all numbered. Now, if this book is going to be long, maybe (TOTAL PAGES HERE) pages, it should have a table of contents. This can be done simply, with the `\tableofcontents` command. Just place it after `\maketitle`. Note that if you want everything in the table of contents to be correct, you'll have to typeset twice. The first time figures out what all the chapters, sections, etc., are, and the second time will use that information to make a table of contents. We'll include one in this book really quickly. While following the example, be sure to typeset twice.

```
\documentclass{book}

\begin{document}

\title{My First Book}
\author{A Published Author}

\maketitle

\tableofcontents

\chapter{The First One}

\section{The First Piece}

\section{A Different Piece}

\subsection{A Smaller Piece}

\subsection{An Even Smaller Piece}

\end{document}
```

Example 1.2.7: A book with a table of contents.

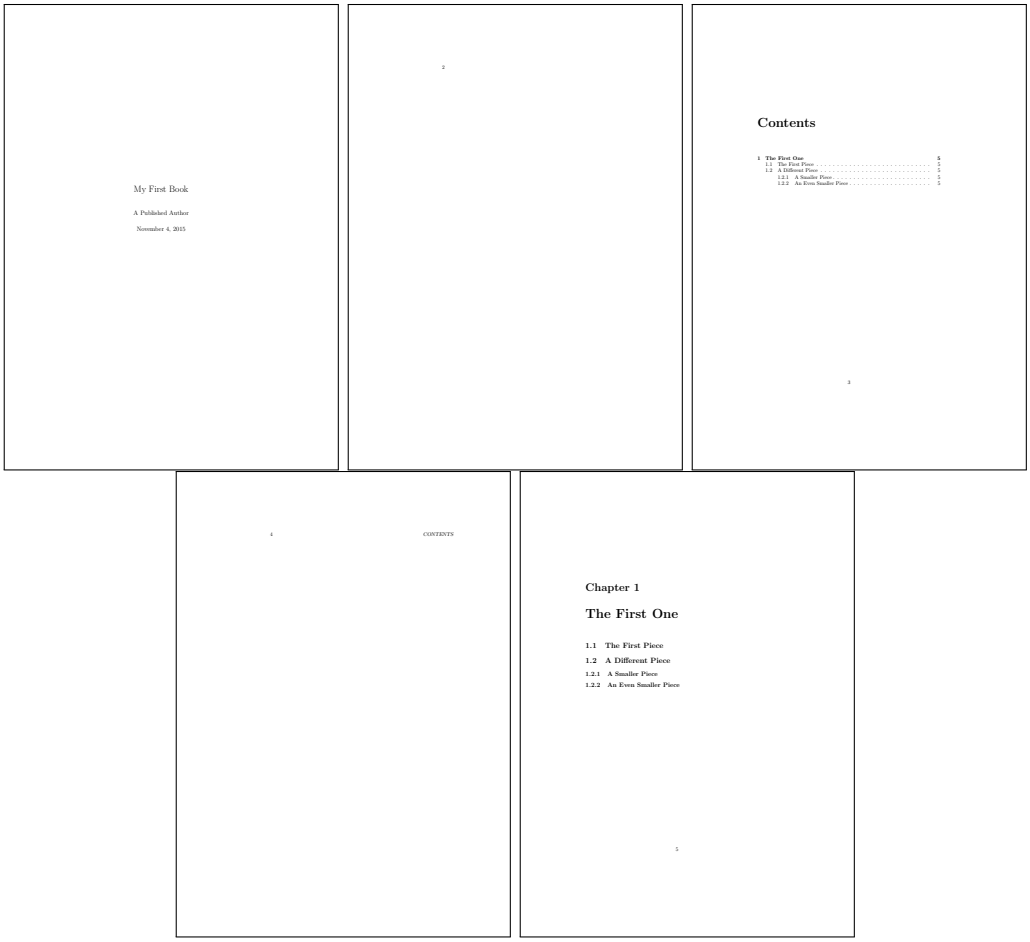


Figure 1.2.7: Example 1.2.7 results. Even with one chapter, there are already five pages.

This content is not necessarily helpful for physicists, since books are not something most physicists routinely write. It is, however, helpful to know about them and show you how the scope of these segments work.

## Exercises

For your benefit, you can just use this paragraph for the exercises:

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cil-

lum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

1. Make an un-numbered list of the sentences from the paragraph above, with bullets.
2. Make an un-numbered list of the sentences from the paragraph above, with dashes.
3. Make an un-numbered list of the sentences from the paragraph above, with no marker.
4. Make a numbered list of the sentences from the paragraph above.
5. Take the skeletal outline of the scientific article in Example 1.2.4 and then fill out each section with a paragraph.
6. Most journals don't number the Abstract or Acknowledgment sections of their papers. Take the slightly less skeletal outline from Example 1.2.5 and make both the Abstract and Acknowledgments un-numbered.
7. Make a mock article with 4 sections, each with 3 subsections. Each subsection should have a paragraph. Make the first section un-numbered. What happens to the subsections in its numbering?
8. Make a mock article with 4 sections, each with 3 subsections. Put a paragraph before the first subsection in each section, and a two paragraphs per sub-section. Make the first section and all of its subsections un-numbered. Generate a table of contents.
9. Take the article from the previous exercise, and add an un-numbered section at the end with no subsections.

## 1.3 Source Structure

When you have a larger project, you may need to separate out your LaTeX source files. This is especially true if you have a paper that would otherwise be many thousands of lines of code. One strategy to minimize such a mess is to spread your source code out over many files, perhaps one per section. (This is one part of the strategy I used to keep the source for this book manageable).

In order to take LaTeX from a different file, you need to make sure that LaTeX can find the other file first. For now, we'll have two files in the same directory. Make a new folder, and open LaTeX. Make one article called `main.tex`. Give it a title, author, and a section with the heading `Input Text`. Then, make a file called `input_text.tex`. Don't give it a document class or anything. Just put in a single sentence.

The way we're going to connect these two things is to use the `input{}` command. This command take a file name as an argument. If we put the

`\input{}` command on document A, then document B's filename inside of the input command, typesetting A will cause B to be included. So let's add `\input{input_text}` to our `main.tex` file, like so:

```
\documentclass{article}

\begin{document}

\title{A Divided Document}
\author{A Budding \TeX nician}

\maketitle

\input{input_text.tex}

\end{document}
```

Example 1.3.1: Including another file.

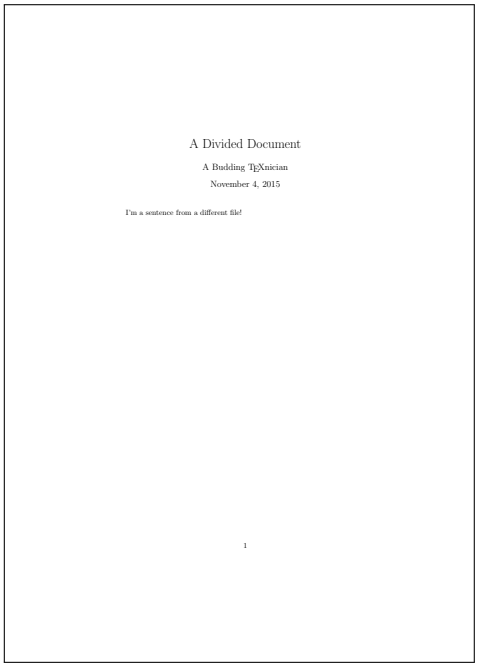


Figure 1.3.1: Example 1.3.1 results. It's all just one page.

You can also put things in multiple folders, which is useful for even larger documents. Go ahead and make a folder called "Folder." Go to that folder, and make a new document, creatively named `document.tex`. Put a sentence in it.

Now, go back to `main.tex` and add the following line:

```
\input{Folder/document.tex}
```

Example 1.3.2: Including another file from another folder.

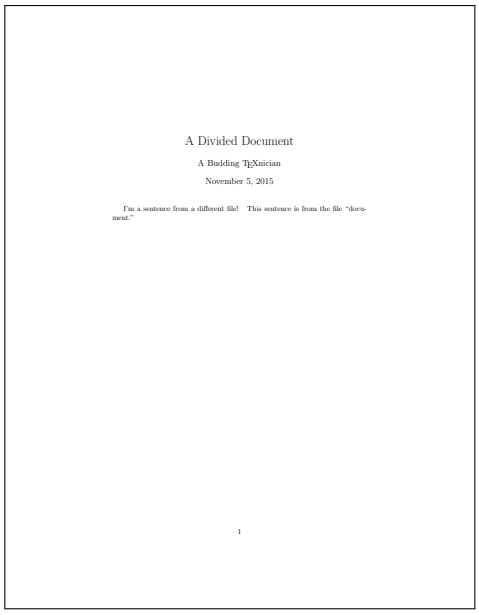


Figure 1.3.2: Example 1.3.3 results. Note that the new sentence isn't put on a separate line or anything.

Just so you know, if you have to move back a folder, use two dots (..) and follow it with a slash (/). Make sure that you're typing the folder path from the perspective of the file that you're typesetting. Most of the time, however, you'll find that more than a couple folders isn't necessary to keep everything organized.

### References and Labels

There are two commands, `\label{}` and `\ref{}`, that make referencing things substantially easier within your document. Place the `\label{}` command after a figure, equation, or section, with a name for the label. For example, you might call a section `equations`. Then, to reference the section number later, use the `ref` command to put the number of the thing you're referencing in the text. This is most easily explained through example:

```
\subsection{Whatever-thing}
\label{sec:whatever}
This is subsection \ref{sec:whatever}.
```

Example 1.3.3: An example of labels and referencing.

### Whatever-thing

This is subsection 1.3.

Figure 1.3.3: Example 1.3.3 results.

You have to be careful to typeset twice though, otherwise labels might not be accurate. This is for the same reason that you have to typeset twice to get an accurate table of contents.

### Exercises

Use the long paragraph from the Document Structure section for these exercises.

1. Make a document that has three sections. Each section should be in a separate file, with a main file tying them all together. Make sure that the `\section{}` command is in the main file.

2. Make a document that has three sections, and with three subsections each. Put each section in a different folder, with each subsection in a different folder.

# Chapter 2

# Mathematics

## 2.1 The Equation Environment

Much like the other environments, the equation environment spans a portion of the document and makes special rules. In this case, it changes from a standard text interpretation of LaTeX to a mathematical one.

### Basic Rules and Commands

Before you dive in, there are some rules and commands that you should know about math mode:

1. There are no spaces.
2. All letters are italicized.
3. The equation is placed on its on own line.
4. Equations are numbered, unless otherwise specified.
5. Underscore (`_`) is treated as a command. The brackets enclose a subscript.
6. Carrot (`^`) is treated as a command. The brackets enclose a superscript/exponent.
7. For `^` and `_`, if your sub or superscript is only one character, you don't need brackets.
8. Trig functions are identified by there own commands: `\sin`,`\cos`,`\tan`
9. Integrals are made with the `\int` command. Limits of integration can be done using sub and superscripts (see above).
10. Insert text can be done with the `\text{}` command.

We're going to show all of these rules simultaneously in the following example. Note that I put spaces between things. They don't show up in the document, so it helps keeps things de-cluttered. If you keep things too close together, you or LaTeX might get confused and think that two commands are working together, resulting in nasty output from either your computer or from your confusion.

```
\begin{equation}
x = \int e^{- x} \sin a_1 t \text{ d}t
\end{equation}
```

Example 2.1.1: A mathematical equation. Note the text "d." This is to differentiate the d as the differential.

$$x = e^{-x} \sin a_1 t \tag{2.1}$$

Figure 2.1.1: Output of Example 2.1.1

### More Commands

There are a few other important things to know. If you want to make a square root sign ($\sqrt{}$), you have to use the command `\sqrt{}`. Whatever you put inside of the brackets will appear under the square root sign. If you want to make a fraction, you have to use the `\frac{}{}` command. Note that there are two sets of brackets. This is because the first set is the numerator argument, and the second is the denominator argument. Here's an example:

```
\begin{equation}
x = \frac{\sqrt{y}}{2}
\end{equation}
```

Example 2.1.2: A fraction with a square root.

$$x = \frac{\sqrt{y}}{2} \tag{2.2}$$

Figure 2.1.2: Output of Example 2.1.2

If you want to put an equation on its own line, but you don't want it numbered, you can use the `equation*` environment instead. It behaves the exact same way that the normal equation environment does, but without numbering.

### Parantheses and Brackets

In addition to normal parantheses and brackets, you may find that you need larger parentheses in order to encapsulate large numbers, fractions, or other expressions. In this case, you can use the `\left` and `\right` commands. This works for all brackets and parentheses. Simply type `\left(` (or whatever bracket you want), then the expression that you want, and then `\right)` to close it off.

```
\begin{equation*}
\left( \frac{a^2 + b^2}{c^2} \right) x = e^{x}
\end{equation*}
```

Example 2.1.3: A large paranthesis around a large fraction.

$$\left( \frac{a^2 + b^2}{c^2} \right) x = e^{x}$$

Figure 2.1.3: Example 2.1.3 output.

I'll show you another quick demonstration of paranthesis/bracket matching.

```
\begin{equation*}
\left( \left( \left( x \right) \right) \right)
\left[ \left[ \left[ x \right] \right] \right]
\end{equation*}
```

Example 2.1.4: Parantheses and brackets and such.

$$(((x)))\,[[[x]]]$$

Figure 2.1.4: Example 2.1.4 output.

You can also manually adjust their size with `\big` and `\bigg`.

```
\begin{equation*}
\bigg( \big( \left( x \right) \big) \bigg)
\bigg[ \big[ \left[ x \right] \big] \bigg]
\end{equation*}
```

Example 2.1.5: Parantheses and brackets and such.

$$\Big(\big(\left(x\right)\big)\Big)\Big[\big[\left[x\right]\big]\Big]$$

Figure 2.1.5: Example 2.1.5 output.

### In-line Mathematics

Sometimes you want a mathematical expression to show up in the middle of a sentence. In this case, you can surround the expression between dollar signs ($). Keep in mind that the math will be reduced to be the same height of the text. This means that if you make a fraction, the math font size will be very small. Take this as an example:

```
The exponential function,
$e^{x}$,
can be broke down into the Taylor series,
$1 + x + \frac{x^2}{2!}$.
```

Example 2.1.6: The taylor expansion of the exponential function.

The exponential function, $e^x$, can be broke down into the Taylor series, $1 + x + \frac{x^2}{2!}$.

Figure 2.1.6: Output of Example 2.1.6. Note that thh 2 in the third term is very difficult to see.

### Exercises

Keep all of these exercises on a document titled, "Equations Practice," with you as the author. Each question should be numbered via the enumerate environment. Typeset all of the following:

1. The Pythagorean theorem as a numbered equation.
2. The Sentence, "Is $e^{ie^{ie^{ie^{iz}}}}$ straightforward to break down?" (The answer is no, if you're wondering).
3. The numbered equation, $y = x(x + 1)$.
4. The sentence, "$x$ must be smaller than 1 if $x^{\frac{1}{2}} > x$."
5. The numbered equation, $c_1 = \frac{e^5}{5}$.

## 2.2 The amsmath Package

Though LaTeX is powerful by itself, it's important to use the best tool for the job. The American Mathematical Society has created a package that adds several very useful commands to LaTeX. It adds greek letters, some additional symbols, and all sorts of other goodies and capabilities.

### Greek Letters

In physics, we often run out of Roman characters and need to borrow from Greek. Sometimes, we even borrow from Hebrew if we run out of Greek letters. In any case, amsmath makes this very easy. Take any greek letter and simply

put a slash before it to make it appear. If you want the capital version, start the name with a capital letter. If you want the "lowercase" version, start the name with a lowercase letter.

```
\begin{equation*}
\gamma = \frac{1}{\sqrt{1 - \beta^2}}
\text{ and }
\psi = R(r) (\Phi (\phi) \Theta (\theta)
\end{equation*}
```

Example 2.2.1: A Greek Letters Example.

$$\gamma = \frac{1}{\sqrt{1 - \beta^2}} \text{ and } \psi = R(r)(\Phi(\phi)\Theta(\theta)$$

Figure 2.2.1: Output of Example 2.2.1.

If you can't find a greek letter, then you can try googling it or looking up the list of greek letters on a source such as wikipedia. In any case, amsmath comes with all of them, and you can simply write `\lettername` to get it.

### The align environment

Above, we have an equation that has an "and" right in the middle. It's terribly annoying. Let's break it into two separate lines. The align environment allows us to do precisely that, and to line up a character from one line to the next. In order to use this environment:

1. Start the environment using `\begin{environment}`.
2. Put the ampersand (`&`) character left of the character you want to line up.
3. Use `\\` to end a line and start a new one.
4. End the environment using `\end{environment}`.

```
\begin{align}
\gamma &= \frac{1}{\sqrt{1 - \beta^2}}\\
\psi &= R(r) (\Phi (\phi) \Theta (\theta)
\end{align}
```

Example 2.2.2: The align environment in action.

$$\gamma = \frac{1}{\sqrt{1-\beta^2}} \tag{2.3}$$
$$\psi = R(r)(\Phi(\phi)\Theta(\theta) \tag{2.4}$$

Figure 2.2.2: Output of Example 2.2.2. Note that each equation is numbered.

Sometimes you may decide to typeset a series of steps. This also makes the align environment incredibly useful.

```
\begin{align}
y  &= (x+1)^3\\
&= (x^2 + 2x + 1)(x + 1)\\
&= (x^3 + 2x^2 + x + x^2 + 2x + 1)\\
&= x^3 + 3x^2 + 3x + 1
\end{align}
```

Example 2.2.3: The align environment being used to show extended work.

$$\begin{aligned} y &= (x+1)^3 && (2.5)\\ &= (x^2+2x+1)(x+1) && (2.6)\\ &= (x^3+2x^2+x+x^2+2x+1) && (2.7)\\ &= x^3+3x^2+3x+1 && (2.8) \end{aligned}$$

Figure 2.2.3: Output of Example 2.2.3.

In our examples, everything is numbered on each line. It's possible to simply leave everything unnumbered with the align* environment, just like the equation* environment. In our example above, in fact, it's far preferable that way. Unfortunately there is no way to pick and choose lines to be numbered or not.

```
\begin{align*}
y  &= (x+1)^3\\
&= (x^2 + 2x + 1)(x + 1)\\
&= (x^3 + 2x^2 + x + x^2 + 2x + 1)\\
&= x^3 + 3x^2 + 3x + 1
\end{align*}
```

Example 2.2.4: The align environment being used to show extended work, without numbering.

$$\begin{aligned} y &= (x+1)^3 \\ &= (x^2 + 2x + 1)(x + 1) \\ &= (x^3 + 2x^2 + x + x^2 + 2x + 1) \\ &= x^3 + 3x^2 + 3x + 1 \end{aligned}$$

Figure 2.2.4: Output of Example 2.2.4. There's not a number for every single equation, which is far cleaner.

### Matrices in amsmath

One of the most powerful things that amsmath can do is provide the ability to make a matrix in LaTeX. This is something that's very helpful in some instances. In any case, it's very easy to make one. There are several kinds of matrices, with each enclosed by a different kind of delimeter:

1. Paranthetical (pmatrix)
2. Bracket (bmatrix)
3. Braces (Bmatrix)
4. Vertical Bars (vmatrix)
5. Double Vertical Bars (Vmatrix)

Each of these can be made using their name as an environment. Once in the environment, the ampersand separates entries in the same row, and a \\ tells LaTeX to start a new row.

```
\begin{align*}
A
&=  \begin{pmatrix}
1 & 2\\
3 & 4
\end{pmatrix}\\
&= \begin{bmatrix}
1 & 2\\
3 & 4
\end{bmatrix}\\
&= \begin{Bmatrix}
1 & 2\\
3 & 4
\end{Bmatrix}\\
&= \begin{vmatrix}
1 & 2\\
3 & 4
\end{vmatrix}\\
&= \begin{Vmatrix}
1 & 2\\
3 & 4
\end{Vmatrix}
\end{align*}
```

Example 2.2.5: All of the types of matrix.

$$
\begin{aligned}
A &= \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \\
&= \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \\
&= \begin{Bmatrix} 1 & 2 \\ 3 & 4 \end{Bmatrix} \\
&= \begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix} \\
&= \begin{Vmatrix} 1 & 2 \\ 3 & 4 \end{Vmatrix}
\end{aligned}
$$

Figure 2.2.5: Output of Example 2.2.5.

### Additional Commands

The amsmath package also contains several useful commands that aren't found in normal LaTeX. For example, amsmath lets you make vectors using the `\vec{}` command, which places an arrow over its argument. If you're mega-old-school like me, you can also make vectors using the `\underline{}` command, which does precisely what you think it does. You can put a hat ($\hat{}$) over things using the `\hat{}` command. Hats are used for both unit vectors and, later on, operators. The amsmath package also provides a dot product symbol via `\cdot`. To make a cross product, use the `\times` command. These are some of the more common symbols, but if you google, "amsmath symbols" you'll get a very complete list. I'll give you one example with these.

```
\begin{align*}
\vec{a} &= \underline{a}\\
&= (\vec{b} \cdot \vec{c}) \hat{\vec{d}}\\
&= \underline{e} \times \underline{\hat{f}}
\end{align*}
```

Example 2.2.6: Symbols in amsmath.

$$\begin{aligned}\vec{a} &= \underline{a} \\ &= (\vec{b}\cdot\vec{c})\hat{\vec{d}} \\ &= \underline{e}\times\underline{\hat{f}}\end{aligned}$$

Figure 2.2.6: Output of Example 2.2.6.

### Exercises

Put these exercises in a document called, "Exercises for amsmath," with your name as author.

1. Show all four steps of the expansion of $(x+1)^4$ using align. Make them un-numbered.
2. Make the numbered equation $\alpha^2 = \sqrt{\beta^2+\gamma^2}$.
3. Make the un-numbered equation $e^{i\pi}+1=0$
4. Make a 3x3 bar matrix equal to the vector a, as a numbered equation. The matrix should just contain numbers 1-9, increasing from left to right.
5. Make a 4x4 parenthetical matrix alone as an un-numbered equation, containing 16 letters of the greek alphabet.
6. The sentence, "An example of the triple product rule is $\vec{a}\cdot(\vec{b}\times\vec{c}) = \vec{b}\cdot(\vec{c}\times\vec{a})$."
7. The numbered equation $\hat{\vec{x}}\times\hat{\vec{y}}=\hat{\vec{z}}$.

## 2.3 Additional Exercises

These exercises are extremely highly recommended. Typeset all of these on a document titled, "Mathematical Typesetting Exercises," with your name as the author. Be sure to use the amsmath package.

1. The quadratic formula as a numbered equation.
2. The law of sines as an un-numbered equation.
3. The definition of a Laplacian in spherical coordinates as a numbered equation. (You can google this if you don't know it). To make a partial differential, use `\partial`.

4. The integral
$$\int_0^\infty e^{-t^2}\mathrm{d}t = \frac{2}{\sqrt{\pi}}$$
as an un-numbered equation. Make sure that there is a space between the integrand and the differential, and that the differential is not italicized. The symbol for infinity is $\infty$.

5. The steps to solving the integral $\int_a^b te^t\mathrm{d}t$. Use the align environment to make sure that your "=" line up, and don't number the equations. (Hint: integration by parts).

6. The steps to expanding the polynomial $(x-2)^5$. Use the align environment and make each equation numbered.

7. A parenthetical matrix, un-numbered. It should have four rows and four columns. Row One should have four different greek letters. Row Two should have four different fractions. Row Three should have four different exponents. Row Four should have a greek letter, an exponent, a fraction, and a polynomial.

8. The numbered expression
$$x = \frac{1+\frac{1}{y}}{1-\frac{1}{e^{\frac{1}{e^y}}}}$$

9. The numbered expression
$$\begin{aligned}\sin x &= \sum_{n=0}^{\infty}\frac{(-1)^n x^n}{n!}\\ &\approx 1 - x + \frac{x^3}{3!} - \frac{x^5}{5!} + \ldots\end{aligned}$$
You'll have to look up how to make the sum symbol and how to get the `\approx` symbol. Looking things up is a valuable LaTeX skill, so no complaining.

10. Do what you did for the previous expression, but with the expansions for cosine. You can find them on the Wikipedia article "Taylor Expansions," in the section "Expansions of common MacLaurin Series."

# Chapter 3

# Figures

## 3.1 Importing Pictures

Just be warned that for figures, LaTeX has its own ideas about where they should go, that often disagree with yours. This is one of those times. I've numbered the figures and examples, so you'll want to match figure and example numbers, not look immediately after the examples for the figure. Sometimes it's two or three pages forward of the example.

### My First Picture

Go onto the internet and download a picture of a rubber duck. Rename it "duckie.jpg" and put it in the same folder as a new document. Now, you need to know about the `graphicx` package. Include that package in your preamble, via `\usepackage{graphicx}`. Now, you can import a picture using the `includegraphics` picture.

```
\includegraphics{duckie.jpg}
```

Example 3.1.1: Putting a rubber duckie in our document.

### Rotating a Picture

Quackers is awful cute, isn't he? Let's say that we're working on a research project, and a careless colleague didn't rotate their picture correctly. Fortunately, LaTeX comes with the tools we need to correct this. the `\includegraphics` command can take an optional argument. In this case, we can include the option `angle=90` to rotate Quackers 90 degrees.

Figure 3.1.1: A rubber duckie! His name is Quackers. We brought him here in Example 3.1.1.

```
\includegraphics[angle=90]{duckie.jpg}
```

Example 3.1.2: Rotating Quackers.

### Scaling a Picture

We can also make Quackers bigger and smaller using the `scale` optional argument. We set `scale` equal to a fraction of the original size. If we want to make Quackers smaller, we pick a scale less than one. If we want to make Quackers bigger, we pick a scale more than one. We'll do two examples in a row for this.

Figure 3.1.2: Quackers feels dizzy from Example 3.1.2.

```
\includegraphics[scale=.5]{duckie.jpg}
```

Example 3.1.3: Small Quackers.

```
\includegraphics[scale=2]{duckie.jpg}
```

Example 3.1.4: Big Quackers.

This is pretty good start to importing images. There are some other things

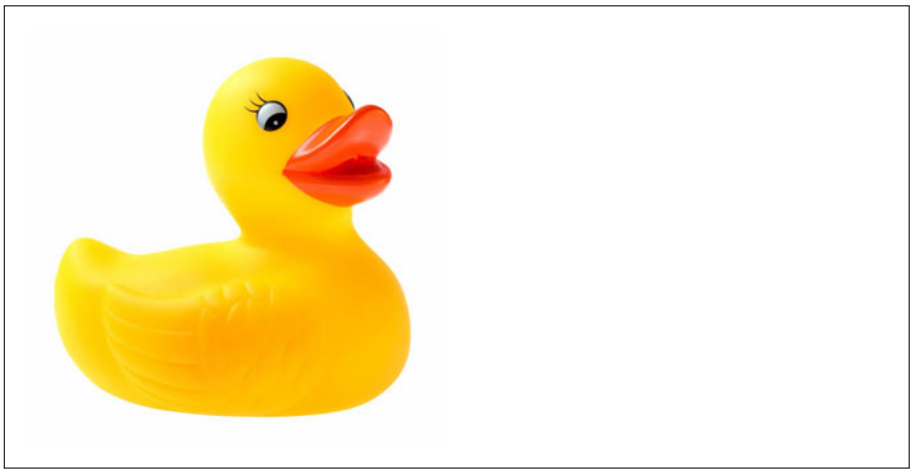

Figure 3.1.3: Quackers feels shorter as a result of Example 3.1.3.

you should know, though. You should try to use .svg or .eps files instead of .jpg files as much as possible. These are vector graphics, which means that you can stretch them and that they will scale instead of being pixellated messes, unlike .jpg files. This makes your document far more neat and tidy.

### Exercises

Do these all on a document titled, "Importing Pictures Exercises."

1. Take your picture of a rubber duck and rotate it 180 degrees.
2. Take your picture of a rubber duck, rotate it 270 degrees, and then make it twice as large.
3. Take your picture of a rubber duck, rotate it 45 degrees, and then make it half as large.
4. You can also define how wide you want the image to be with the "width" optional argument. You can't use that and scale, but sometimes you'll want to use width instead. You can set with equal to any number of inches or centimeters. Make your picture of a rubber duck 5 centimeters wide, using `width=5cm`.

## 3.2 The Figure Environment

### How to Figure

Just like any other environment, you start the environment with the `\begin{}` command, and end it with the `\end{}` command. In between that, you'll

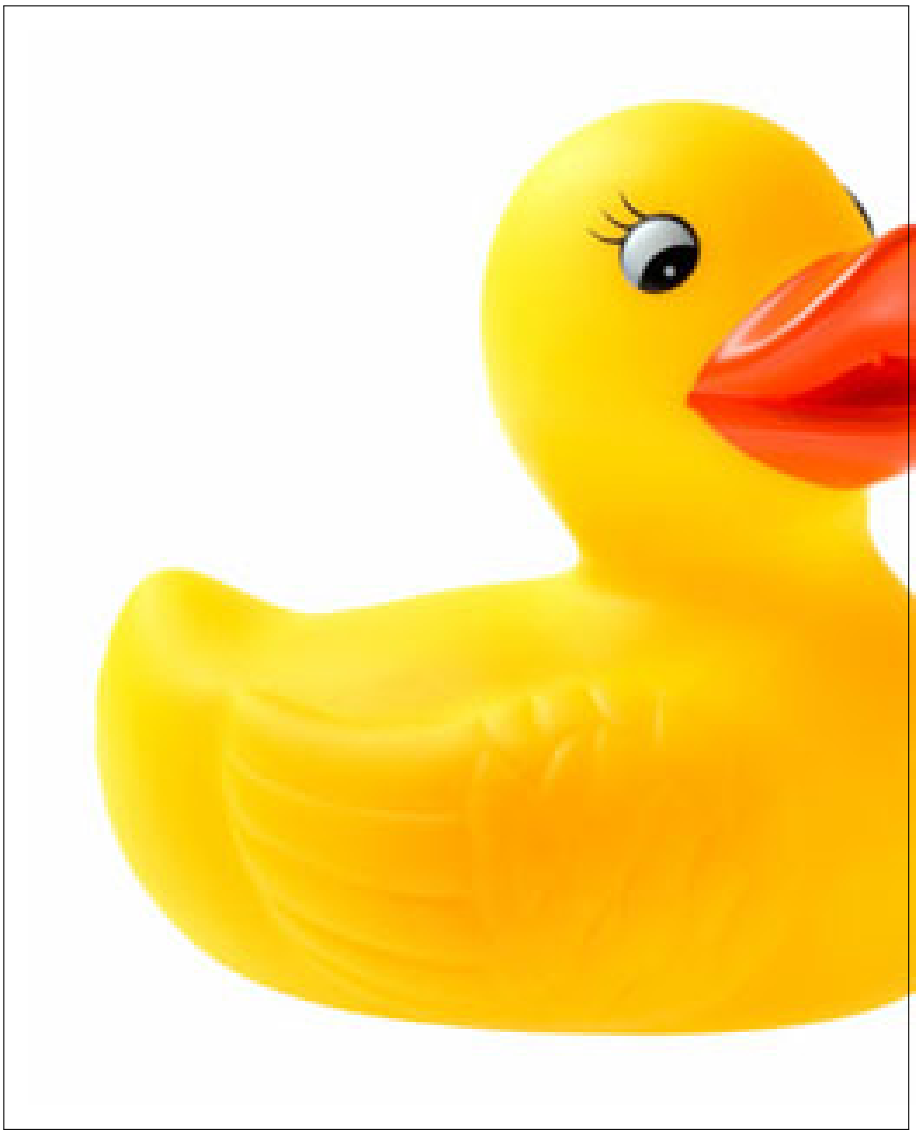

Figure 3.1.4: Quackers feels taller as a result of Example 3.1.4.

include a graphic or whatever else you plan to call a figure, and perhaps include a caption. First, I'll show you how to make Quackers from the previous section as a Figure.

```
\begin{figure}
\includegraphics{duckie.jpg}
\end{figure}
```

Example 3.2.1: Turning Quackers into a figure.

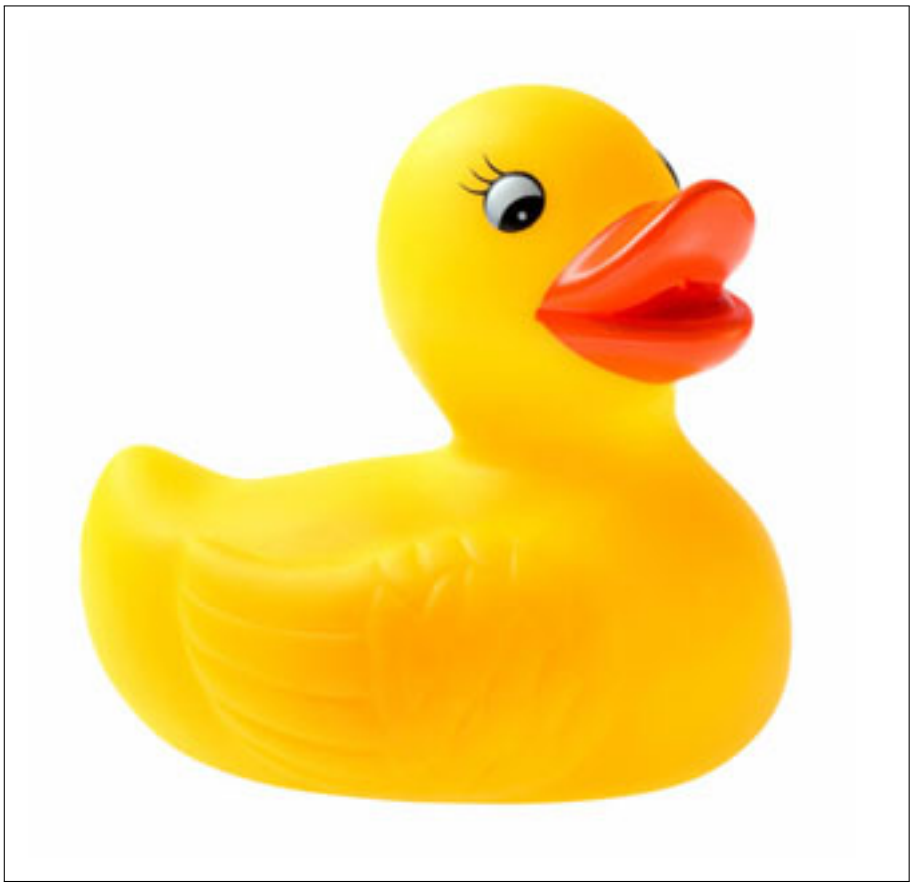

Figure 3.2.1: He's such a happy figure from Example 3.2.1.

Quackers doesn't have a figure number yet. That only works if we include a caption with him.

### How to Caption

Now we're going to give Quackers a caption. All you have to do is put the `\caption{}` command at the end of the figure environment. Fill the area between the brackets with the caption of your choice. In this case, we'll just label the figure "Quackers!!!!" Since Quackers is his name.

```
\begin{figure}
\includegraphics{duckie.jpg}
\caption{Quackers!!!!!!}
\end{figure}
```

Example 3.2.2: Giving Quackers a caption.

Figure 1: Quackers!!!!!!

Figure 3.2.2: Quackers has a caption and figure number because of Example 3.2.3.

### Figure Placement

Figures are extremely annoying about where they go. You'll find yourself trying to make a document, and suddenly realize that nothing is going where you want it to. At all. LaTeX includes some toools to try and rectify this, but mostly it does whatever it feels like doing that day. Sometimes the figure will go on the pagte that you like, but other times it will go on something two pages later for virtually no reason. There are a couple of things that you can do to try and get it to go where you want. The first is to try putting the optional argument "h" in when you start the figure environment. That basically says, "LaTeX,

buddy, could you please put this figure in this spot where I typed it?" Then if that doesn't work, "H" is another optional argument. This means, "LaTeX, you frustrating piece of software, I DEMAND that you put this figure here." That occassionally works, but mostly not. Ultimately, try not to obsess over figure placement in LaTeX, because it's not worth it. If you're going to try and fix it, wait until you're absolute and completely sure that you're done with all other aspects of the document. Trying to move figures is a huge time sink, and you don't want to waste time with it until you're done. I'm not going to bother showing a result since it's not relevant, but here's an example to show you how to try and tell LaTeX what to do:

```
\begin{figure}[H]
\includegraphics{duckie.jpg}
\caption{Quackers!!!!!!}
\end{figure}
```

Example 3.2.3: Attempting to get a figure to go to a particular place. Good luck.

### How to Label and Reference

References and labels are really easy to do in LaTeX, which allows you to aavoid numbering things. This is especially convenient if you have to rearrange things later. When making this book, for example, I had to rearrange several sections after I was done with a chapter. This would have been a total nightmare to try and re-number, similar as trying to re-number a list after you changed the order.

In order to reference figures, use the `\label{}` command after you put the caption in the figure environment. (Recall my description of how it works from the "Source Structure" subsection. Then, to reference the figure, simply use the `\ref{}` command. Just like with all references and labels, you'll have to typeset them twice in order to gaurantee accuracy. We'll do an example here.

```
\begin{figure}
\includegraphics{duckie.jpg}
\caption{Quackers!!!!!!}
\label{fig:quackers}
\end{figure}
You can see Quackers in Figure \ref{fig:quackers}.
```

Example 3.2.4: Giving Quackers a label. Note that the first three letters are "fig." This is to make it clear that this is a figure label, for me or anyone else reading source code. LaTeX could care less.

Figure 1: Quackers!!!!!!

You can see Quackers in Figure 1.

Figure 3.2.3: Quackers is easy to reference because of Example 3.2.4.

1. Name your rubber duck in the caption of the figure.
2. Make two rubber ducks, each rotated by 90 degrees, and name them different things. Name each duck in a sentence, indicating the figure using `\label` and `\ref`.

## 3.3 The Tabular Environment

### My First Table

The tabular environment is used for making tables. This is generally not used in the modern day, since people prefer to use figures to illustrate their points. This is more common in astronomy, though. You need two arguments to start

beginning of the tabular environment. The first is the word tabular. The second is a collection of characters describing the number and format of the columns. An l character makes a left-aligned column, an r a right-aligned, and a c a center-aligned. Putting a — character between two letters places a line between the columns. Once you start the environment, use the `&` character to split the columns, and the `\\` character to start a new row. End the environment using the `\end{tabular}` command.

```
\begin{tabular}{l |c r}
\hline
left & center & right\\
word & other & word\\
\hline
\end{tabular}
```

Example 3.3.1: Building a table.

| left | center | right |
|---|---|---|
| word | other | word word |

Figure 3.3.1: The table from Example 3.3.1.

### The Table Environment

You can also use the table environment. It works literally the same way as the figure environment, except labelling things tables instead of figures. I'm pretty sure you can figure this out.

### Exercises

1. Make a table where everything is centered, with three columns and five rows, along with a title row. The title should read F, C, and K. The other five rows should read the temperature in F, C, and K, with one row all being the same temperature.

2. Make a table with two columns, one in mi/h and one in km/h. Use the table environment to give it a caption. There should be five different speeds listed.

## 3.4 Additional Exercises

1. Make three figures of a rubber duck and then reference each of them in a table that shows the name of the duck in each figure. Make sure the name of the duck is in the caption.

It's difficult to make people do exercises like this, and you'll get better practice out of doing real documents. So I guess make real documents. Good luck.

# Chapter 4

# Using RevTEX

## 4.1 A First Document

First, you need to see how to use the template. Note that you may need to install it from the APS website. The APS has relatively strict author guidelines, which you should read carefully if you choose to submit. In any case, once you have it installed, you should use the line

```
\documentclass[10pt,twocolumn]{revtex4-1}
```

The optional argument 10pt tells LaTeX that you want your font size to be 10 pt. The argument twocolumn tells LaTeX to make your document two columns. This is a very compact way of filling a page and keeping it legible, and is thus very common in journals and other publications. It is less, suitable, however, for protracted math work, making it a poor choice for submitting homework or writing tests.

The actual document class is revtex4-1. If your RevTeX distribution isn't installed properly, you may get an error saying, "Class file revtex4-1.cls was not found." This is LaTeX telling you that it can't find the document class that it needs to build a document. Try troubleshooting using the APS suggestions on their website to get your class file in place.

In RevTeX, the abstract is actually its own environment instead of a section. In order to make your abstract, you'll have to use the `\begin{abstract}` and `\end{abstract}` commands, with the actual text of the abstract in the middle. You'll want to place the abstract environment before you make the title. Otherwise, LaTeX throws an error. In addition to the abstract environment, RevTeX also has an `\affiliation{}` option for the title. This is so that each author's affiliation is displayed on the document. You can put one affiliation per author, right after their names.

We're going to make a sample document without references first.

```
\documentclass[10pt,twocolumn]{revtex4-1}

\begin{document}

\title{Conducting \LaTeX\ Seminars on Saturday Mornings}
\author{Joshua B. Teves}
\affiliation{College of Charleston}

\begin{abstract}
Students were gathered together on a Saturday morning,
at approximately noon,
to talk about \LaTeX\ and make some \TeX\ nicians out of them.
\end{abstract}


\maketitle

\section{Introduction}

\par
This document is a test document.

\end{document}
```

Example 4.1.1: A first RevTEX document.

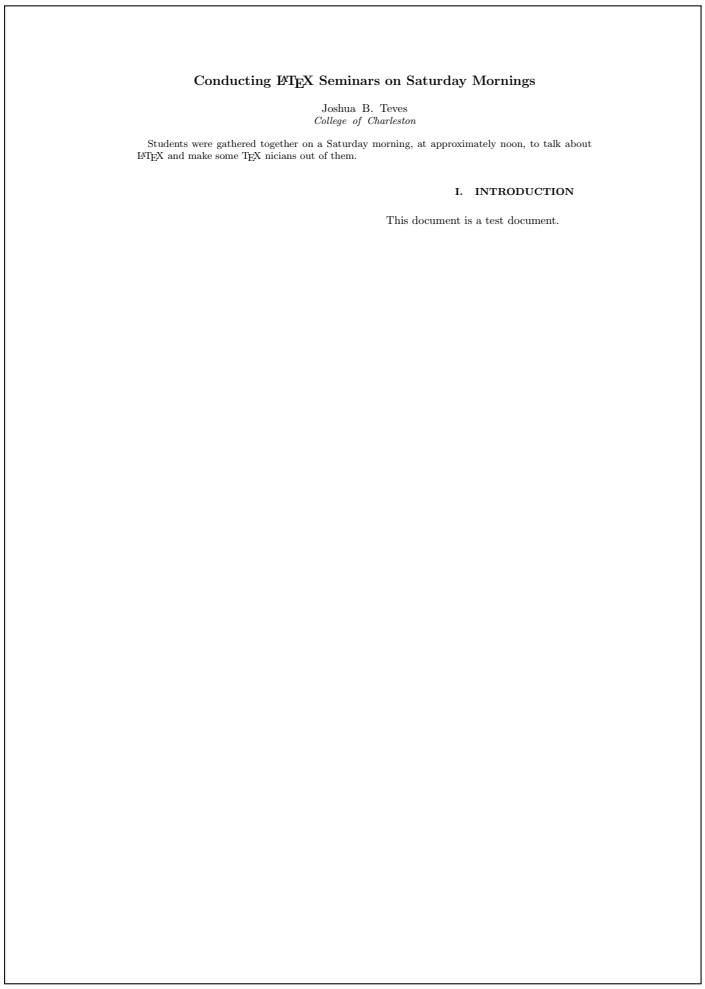


Figure 4.1.1: Example 4.1.1 result.

Note that since the document is two columns, the section starts on the right hand column. Generally, LaTeX would try to split things so that the horizontal level of the columns is the same, but right now it can't since there's only one sentence for it to put on there. Just so you know, this is how LaTeX handles all two column scenarios, unless the document class specificies otherwise.

To further demonstrate how the twocolumn looks, we're going to bring back our friend lipsum and fill out the document.

```
\documentclass[10pt,twocolumn]{revtex4-1}

\usepackage{lipsum}

\begin{document}

\title{Conducting \LaTeX\ Seminars on Saturday Mornings}
\author{Joshua B. Teves}
\affiliation{College of Charleston}

\begin{abstract}
Students were gathered together on a Saturday morning,
at approximately noon,
to talk about \LaTeX\ and make some \TeX\ nicians out of them.
\end{abstract}


\maketitle

\section{Introduction}

\lipsum

\end{document}
```

Example 4.1.2: RevTEX document with actual text.

**Conducting LaTeX Seminars on Saturday Mornings**

Joshua B. Teves
*College of Charleston*

Students were gathered together on a Saturday morning, at approximately noon, to talk about LaTeX and make some TeX nicians out of them.

**I. INTRODUCTION**

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Figure 4.1.2: Example 4.1.2 result.

RevTEXalso lets you make a bibliography using either BibTEX (see the appendix), which lets you split your references into a separate file, or using bibitem, which keeps your references on the same file. Either way, you start the bibliography using the command `\thebibliography`. In order to use bibitem, you have to manuaully alphabetize and provide a bunch of options, following the journal's style guide manually. BibTEX is the superior choice for anything more than a couple references.

The full author guidelines for RevTEX have been provided as a separate appendix.

# Chapter 5

# BibTEX

BibTEX is a bibliography-tracking system. It requires that you, in addition to typesetting your document, use the bibtex command. In both MikTEX and TEX Shop, this is available in the drop-down menu. There are three chief things:

1. A references.bib file.
2. Include the package `cite`.
3. Include the `\bibliography` command.

You need to make a file called, for example, `references.bib`. This file will contain your references. In your LATEX document, `\bibliography{filename.bib}` should be put where you want the bibliography to go in your document. You'll have to run bibtex on your document. The format for bibtex references is as follows:

```
@misc{ referencename,
author= "author",
title= "title",
year = "2006"}
```

When you want to cite something in your document, just use the command `\cite{}`.

```
As shown in \cite{referencename}
```

might yield, "As shown in [1]" or "As shown in Jaffraine and Berne 2012" depending on the journal type, or the citation type. The important thing is to make sure that your references file is in the same folder as your main document. You should typeset twice, then use bibTEX, then typeset twice again to make sure that everything is cited properly.