DRAFT VERSION AUGUST 10, 2020
Typeset using LATEX default style in AASTeX62

LaTeX Examples with Overleaf for ASTR 695

REBECCA C. LEVY[1] AND IMAN EXAMPLE[1, 2]

[1]*The University of Maryland*
[2]*NASA Goddard Space Flight Center*

ABSTRACT

This document is an example of how to use LaTeX - and specifically Overleaf - for scientific papers.

Contents

rlevy@astro.umd.edu

1. INTRODUCTION

LaTeX is a flexible word processor used for scientific publications. Like anything, there is a learning curve to getting started, but the payoff over another word processor (like Word or Pages) is huge. There are three main areas where I think LaTeX really shines:

1. Typing equations and mathematical symbols
2. Automatically referencing citations, figures, tables, sections, etc in text
3. Flexibility in float (i.e. figure or table) placement.

Moreover, most plotting tools (matplotlib, Matlab, etc) have LaTeX interpreters, so you can use LaTeX-style fonts and symbols in your plots using LaTeX's syntax.

There are many ways to use LaTeX, such as:

- In the cloud:
    1. Overleaf: www.overleaf.com
    2. ShareLaTeX: www.sharelatex.com (newly merged with Overleaf)
- On your computer:
    1. TeXShop: https://pages.uoregon.edu/koch/texshop (my personal favorite because I like IDEs)
    2. TeXworks: http://www.tug.org/texworks
    3. MacTeX: http://www.tug.org/mactex
    4. MiKTeX: https://miktex.org/ (what I used when I had a Windows machine)

In the following sections, we'll go over the different parts of a LaTeX document.

Disclaimers: I am not a LaTeX expert! This is what works for me. I do a *lot* of Googling. This guide is in no way complete. See the source code for examples of some things I won't explicitly go over. To see a complete example, see http://journals.aas.org/authors/aastex/aasguide.html or https://www.overleaf.com/read/prpfjsqmgnqm. The latter is a paper I wrote that has some neat features that I struggled with, like tables input from uploaded files, sideways tables, multi-page tables, and multi-panel figures masquerading as tables (noticing a theme yet?).

1.1. *Basics of LaTeX*

To get started, I highly recommend consulting/ bookmarking the LaTeX Wikibook: https://en.wikibooks.org/wiki/LaTeX.

There are different LaTeX compilers. Generally, you won't need to worry about them and can use pdfLaTex (which is usually the default)[1].

LaTeX has some special symbols that it uses internally. These include (but are not limited to): % & _ { } $ # \. To insert them as a symbol in a document, use `\%`. The exception is \ for which you use `\textbackslash`.

To get quotation marks that look nice, you need to use `` ``'' ``, which will produce "a quote with nice quotation marks". If instead you use the normal double quotation mark key, you'll get "a quote with not as nice quotation marks." Depending on your compiler, these will sometimes look even worse (e.g. ”This makes my skin crawl”).

To comment out a single line, use % (everything in that line after % will be commented out). To comment out a block of text, you can use %{...} (everything within the brackets will be commented out). If, however, the block of text has line breaks in it, you'll need to use `\begin{comment}...\end{comment}''` from the comment package.

Here is a list of common accents to add to letters. Replace the letter in brackets with whatever letter you want.

- á: `\'{a}`
- è: `` \`{e} ``
- ü: `\"{u}`
- ñ: `\~{n}`

[1] If you're using a non-standard font or one from a font file on your machine, you might have to use LuaLaTex. This is the only situation I've had to change compilers, and the error message told me how to fix the problem.

- ô: `\^{o}`

Your turn: Practice typing some symbols and words/names with accents.
Ex: François bought a shirt for $5. It was 80% off.

There are several ways to start a new line in LaTeX. One is to hit return twice (as I've been doing so far). Another is to use `\\` at the end of a line:
Here's a new line (notice there's no indent with this method). A third way is to use `\newline`:
but this is rare. To get a blank line between paragraphs, use two returns and `\\`

so you get a gap like this.
Another useful formatting command is `noindent`, which undoes the automatic indentation. `\clearpage` starts a new page, which is especially useful when writing proposals with strict page limits for each section.

1.1.1. *Math Mode*

To type equations and access more symbols, you'll use math mode. This is where LaTeX *shines*. It takes a little while to get used to, but once you do, it's very efficient (and much better than Word). There are a few ways to do this. For inline text, bracket your mathematical expression with `$...$`: 4×10^6 M$_\odot$ or $y_{\rm final} = \int_0^\infty \log(x)\sin(x)dx$ (Be sure to look at the source code for some of the special characters.) If you want an equation to have its own line, use `$$...$$`:

$$V_{\rm Persic}(x) = V(R_{\rm opt})\left\{\left[0.72+0.44\log(L_B/L_{B*})\right]\frac{1.97x^{1.22}}{(x^2+0.78^2)^{1.43}}+1.6\exp\left[-0.6(L_B/L_{B*})\right]\frac{x^2}{x^2+1.5^2(L_B/L_{B*})^{0.4}}\right\}^{1/2}\ {\rm km\ s^{-1}}$$

This is best for long and/or complicated equations. Yet a third way to use math mode is `\begin{equation}...\end{equation}`. This has the added bonus of automatically numbering your equations and allowing you to reference them in the text:

$$V_{\rm Persic}(x) = V(R_{\rm opt})\left\{\left[0.72+0.44\log(L_B/L_{B*})\right]\frac{1.97x^{1.22}}{(x^2+0.78^2)^{1.43}}+1.6\exp\left[-0.6(L_B/L_{B*})\right]\frac{x^2}{x^2+1.5^2(L_B/L_{B*})^{0.4}}\right\}^{1/2}\ {\rm km\ s^{-1}} \tag{1}$$

Equation 1 gives the form of the Universal Rotation Curve (Persic et al. 1996) (see Section 4 for more about referencing).
Math mode also gives you Greek letters: α, Δ, δ. For a mostly complete list of symbols in LaTeX, see https://artofproblemsolving.com/wiki/index.php/LaTeX:Symbols. Two symbols that aren't on that site that I use somewhat frequently are $\gtrsim$ (`\gtrsim`) and $\lesssim$ (`\lesssim`).
To use the regular font style in math mode, use the command `{\rm}` or `\textrm{}`. You might want your text (in a variable name for example) to look like $V_{\rm rot}$ or $V_{\rm rot}$ instead of V_{rot} (note the different bracket placements in the source code). There is also the '`\mathcal{}` option, which gives you script capital letters such as $\mathcal{L}$ and $\mathcal{F}$. For lowercase script letters, you'll need to look through some other packages, but the commonly used ℓ is `$\ell$`.
To insert a space in math mode, you'll need to use \ between characters (see the example in the source code with the Greek letters above). Note this can also be used outside math mode to insert a space. Sometimes spacing with symbols in general can be weird, especially if you're using one of the shortcuts you made. Here I'm using the shortcut I made for Hα where I have to include a \ after the shortcut or Hαwould run into the next word like this. You can also insert a small space between characters using \   or ~. This looks especially nice for galaxy names (NGC253 v. NGC 253 v. NGC 253 v. NGC 253) and units (5pc v. 5 pc v. 5 pc v. 5 pc).

Your turn: Practice typing some expressions that use fractions, special characters, exponents, regular text, etc.
Ex : $\frac{-\hbar^2}{2m}\nabla^2\Psi(r)+V(r)\Psi(r)=E\Psi(r)$

1.2. *Basics of Overleaf*

Overleaf is a wonderful, flexible, cloud-based LaTeX editor. Overleaf and ShareLaTeX recently merged, and Overleaf v2 is new (this is my first time using it). Overleaf has free and paid versions. I've been fine using the free version so far. Documents are organized by projects, which contain all the supporting documents, images, etc that get complied into the final document. You can easily add, delete, or replace associated files from the file-tree to the left. Overleaf has wonderful find and replace capabilities (not sure where the button went in v2 but Cmd+f on a Mac works). Finally, if you *really* don't like the source code approach to LaTeX, there is a Rich Text editor, but other LaTeX editors may not have this feature, and you'll still have to use some of the source code commands.

I use Overleaf almost exclusively because I can share documents with my collaborators who can see real-time updates. No more emailing old versions back and forth: the same link always works. You can also easily give different people read or write privileges using the "Share" button, so *that* collaborator doesn't mess up all the formatting or accidentally delete everything. And speaking of deleting everything, see Appendix B for how to easily backup your entire project to your computer.

You can quickly navigate to the same place in the source code as the PDF view (and vice-verse in v2). Simply double click on a location in the PDF view and it will take you to that location in the source code. In v2, you can also use the arrows in the center of the page to jump to the current location in either view.

To compile your source code, click the green "Recompile" button. My preference is to have Overleaf auto-compile (click the arrow on the Recompile button and turn Auto Compile on). However, I'll admit that this may be annoying for some people, especially if multiple people are editing the document. When there are errors with your document (or other attached files), the "logs and outputs" icon (piece of paper to the right of "Recompile") will have a red badge showing the number of errors. Click on it, and it will (try to) tell you what and/or where the problem is. If there's a yellow icon, there's a warning somewhere. Icons will also show up on the line that's throwing the error or warning in the source code or other file.

2. PREAMBLE

The preamble of a LaTeX document is where you define the kind of document you're working on, load any packages you'll need, define shortcuts and aliases, and give the title and author information.

2.1. *Document Class*

The general document class is "article". For this example I'm using aastex62 (AASTeX v6.2), which is the template for submissions to the Astrophysical Journal[2]. Most other journals will have their own templates as well. If you're using a template, you'll need the associated class and style files (see aastex62.cls and apjfonts.sty in the file-tree to the left). I find the preformatting of a template nice for most things. There are some options you can change, such as the number of columns and fontsize (e.g. `\documentclass[twocolumn,10pt]{aastex62}`).

Sometimes, however, the templates may not flexible enough if you're not writing a journal article. For more flexibility, I use the article class, but there's a lot more setup required. For an example of an article, see my draft thesis proposal. Here, I wanted some flexibility in how the title page was formatted. Note the increased number of packages loaded and section formatting.

There are many kinds of documents you can produce with LaTeX's pre-defined document classes such as résumés, presentations, reports, and books. Overleaf has a library of templates for various journals and document classes to get you started.

2.2. *Packages*

There are a number of packages you can load into LaTeX (similar to loading modules in Python). When using a template like aastex62, some packages are pre-loaded while others may not be compatible. For example, the commonly used package "graphicx" is pre-loaded in aastex62, while "caption" is incompatible. For a basic article, take a look at my draft thesis proposal for the packages I have loaded. This is why I start with a template like aastex62 if I can.

2.3. *Shortcuts and Aliases*

Shortcuts are very useful for commands you'll be typing a lot. For example, I can type `\ha` rather than `H$\alpha$` for Hα. This is also nice for commands with long names (example shown in Section 4.2.2).

2.4. *Title and Authors*

Before the title, there is the `\begin{document}` command (the `\end{document}` command goes at the very end of the document). The title of your document goes in the `\title{}` command. Overleaf automatically uses the title of the document as the title in your Projects list, but you can change this from the Projects list. For the authors in aastex, a single author can have multiple affiliations, and each co-author gets their own `\author{}` command, as shown. The `\email{}` command is specific to aastex and gives the email of the corresponding author.

[2] More info, downloads, and examples at http://journals.aas.org/authors/aastex/aasguide.html.

2.5. *Abstract*

Where the abstract for your paper goes. This is a specially formatted section, both in aastex and for articles in general, defined by the `\begin{abstract}..\end{abstract}` environment. It spans both columns of a twocolumn document.

3. MAIN BODY

Where everything else goes! Below are some commands for section formatting.

3.1. *This is a subsection*

Here is some text in a subsection. Note that the section numbering and text styles are determined by this template. If you want more customization, use `\documentclass{article}`.

3.1.1. *This is a sub-subsection*

Here is some text in a sub-subsection.

Paragraph—Here is some text in a labeled paragraph.

For an example Appendix, see Appendix A.

4. REFERENCING

This is another place where I think LaTeX really shines. Referencing sections, appendices, figures, tables, and equations within your document is seamless and painless. Referencing bibliographic citations is slightly more work up front, but the payoff is enormous.

4.1. *Referencing Objects in Text*

Referencing sections, appendices, figure, tables, and equations in your document is easy. Use the `\label{key}` command to label the object to be referenced. You can see examples throughout where I've done this. I try to give them useful keys that also indicate what kind of object they are (i.e. fig for figure, tab for table, sec for section, etc). Get in the habit of labeling everything, even if you never reference it (although you should reference all your tables and figures). Then when you're ready to reference your object, simply use `\ref{key}`. Done. Easy. No need to update every instance of Figure 1 if you add a figure before it; LaTeX takes care of it for you.

4.2. *Bibliography*

Another reason I prefer to use a template like aastex is for the bibliography capabilities. The best bibliography package for LaTeX is "natbib", which is automatically included in aastex. natbib integrates fairly seamlessly with SAO/NASA ADS and bibliography preparation services like BibDesk. Using natbib on it's own can lead to some consternation, but there's a lot of help online. I'll refer you to this excellent article about using natbib and other reference styles: https://www.sharelatex.com/learn/Bibliography_management_with_natbib. Even if you're not using aastex, you can still use the aastex bibliography style file (yahapj.bst).

To make the bibliography, there are two lines that go where you want the bibliography to appear. The first is `\bibliographystyle{stylename}`. For ApJ citation style, use `\bibliographystyle{yahapj}` and be sure that yahapj.bst is in the file-tree. For other common citation styles, see https://www.sharelatex.com/learn/Natbib_bibliography_styles. The line is `\bibliography{bibfile}`, where bibfile is the name of the .bib file where the full citation information is stored (see Section 4.2.1). The References section will only show those references cited in the text. In other words, if you have ten references in your bibfile, but only cite two of them, only those two that have been cited will appear in the References section.

4.2.1. *Creating the Bibfile*

This is largely up to personal preference. Some people use software like BibDesk. Personally, I copy the bibliography info from ADS into the bibfile manually (and I cite as I write to save myself pain and suffering later). From ADS, simply copy the BibTeX formatted citation info into the .bib file. I usually rename the citation key with something I can remember easily (e.g. lastnameYY). Check out references.bib in the file-tree to see some examples.

4.2.2. *Citing References in Text*

Once your bibfile is set up, you're ready to cite things! There are two commands you'll need: `\citet{}` and `\citep{}`. `\citet` is used to cite things in the text: Levy et al. (2018) show that blah causes blah. `\citep` is used for parenthetical citations: Blah causes blah (Levy et al. 2018). If you're citing more than one reference, they get grouped into the same citation command: (Levy et al. 2018; Persic et al. 1996). You can also add additional text into the citation using `\citep[][]{}`: (e.g. Levy et al. 2018, and references therein).

If two (or more) citations have the same shortened author list and year, LaTeX will automatically label them a, b, etc chronologically based on the publication info given in the bibfile: Kennicutt (1998a) and Kennicutt (1998b), Kennicutt (1998a,b), or (Kennicutt 1998a,b). (Also check out how spiffy this looks in the list of references.)

Here is an example of using a citation alias. For example, rather than referring to Levy et al. 2018 in my next paper, I'd like it to show up as "Paper I". I can create an alias for that reference using `\defcitealias{paperkey}{Paper I}`. Then, to reference that citation alias in text, I would use `\citetalias{paperkey}` or `\citepalias{paperkey}`. But those are long, and I'm lazy, so I'll alias them as `\newcommand{\cta}{\citetalias}` and `\newcommand{\cpa}{\citepalias}`. Then I can write: "As shown in Paper I, blah blah blah" and "Blah blah blah (Paper I)". In the References section, the correct citation for Levy et al. 2018 is shown and all the linking is correct. It's a good idea to use citation aliases sparingly so your readers can easily follow your paper and to define the alias the first time you use it: e.g. Levy et al. (2018, hereafter Paper I).

Your turn: Find an article to cite, add its info to the bibfile, and cite it.

5. FIGURES AND TABLES

The amount of time you can spend futzing with figures and tables in LaTeX is infinite. Figures and tables are floats, which means they can move anywhere in the document. Usually, this is wonderful because LaTeX will find the best placement for your figure or table based on when it's referenced (see Section 4.1). Other times, it means all your figures/tables end up at the end of the document for no apparent reason. There are also ways to force floats to appear exactly where you tell them to. However, this generally results in a lot of the aforementioned futzing, and I've found that letting some figures/tables float and fixing others generally doesn't end well.

5.1. *Figures*

The most basic way to insert a figure is:

```
\begin{figure}[!h]
\label{fig:umdastro}
\centering
\includegraphics[width=0.2\textwidth]{umdastro.png}
\caption{This is my figure.}
\end{figure}
```

which produces:

Figure 1. This is my figure.

To break this down a bit further, `\begin{figure}[!h]` begins the figure environment with a location tag in []. `h` means "here", and `!` tells LaTeX to override its automatic figure placement. So `[!h]` means that you want the figure to appear *exactly*

Figure 2. This is my figure that spans the whole page width in a two column document.

(a)

(b)

Figure 3. (a) UMD astro logo. (b) UMD logo.

where you inserted it (Normally, this isn't a great idea.). Other location tags are `t` (top), `b` (bottom), and `p` (on a page by itself). Note, you can stack location tags and LaTeX will use them in the order they appear: `[htb]`. You can also use the figure environment without a location tag and leave LaTeX to is own devices.

The main command to insert the figure is `\includegraphics[width=0.2\textwidth]{umdastro.png}` which defines what file to insert and how big to make it. If this were a two column document, the figure would be confined to a single column and would still only occupy 20% of the column's width. If you want a figure to span both columns of a two column document, use the `\begin{figure*}[]...\end{figure*}` environment, as shown in Figure 2.

Your turn: Insert an image!

5.1.1. *Subfigures*

There are three ways to create subfigures:

1. Make the subfigure outside of LaTeX. You can do this either in the programming language you used to make the figure in the first place or in PowerPoint, Photoshop, etc.
2. Use the subcaption or subfig packages to access the subfigure environment. These aren't compatible with aastex, so I can't show an example, but see the following link: https://en.wikibooks.org/wiki/LaTeX/Floats,_Figures_and_Captions#Subfloats.
3. Use `\gridline` is aastex, which is very flexible. See the following link for more info: http://journals.aas.org/authors/aastex/aasguide.html#new_figure_features. You can do things like Figures 3, 4, and 5.

Your turn: Change the document to have two columns. Notice how the figures differ.

5.1.2. *Line Wrapping Around Figures*

For some applications, you may want to wrap the text around a figure. This is especially useful when writing proposals with strict page limits. You'll need to use the wrapfig package. You can see examples in this recent proposal of mine and at this

(a)

(b)

Figure 4. (a) UMD astro logo. (b) UMD logo.

UMD Astro

UMD

Testudo

The Maryland M

Figure 5. A bunch of logos.

link: https://www.sharelatex.com/learn/Wrapping_text_around_figures. The positioning requires a ton of futzing, so I wouldn't recommend this in general.

5.2. *Tables*

We've reached it, the skeleton in the LaTeX closet: tables. Here are two links about tables: https://www.sharelatex.com/learn/Tables and https://en.wikibooks.org/wiki/LaTeX/Tables. The main problem with tables in LaTeX is that they're tedious.

To create a basic table,

```
\begin{table}[h!]
\centering
\begin{tabular}{r | l c}
\hline\hline
Col1 & Col2 & Col3 \\
\hline
1 & 6$\pm$1 & 8$^{+0.3}_{-0.2}$\\
2 & 1 & 7 \\
3 & 9 & 9 \\
4 & 3 & 12 \\
5 & 4 & 2 \\
\hline
\end{tabular}
\caption{A basic table}
\label{tab:table1}
\end{table}
```

which will produce:

Col1	Col2	Col3
1	6±1	$8^{+0.3}_{-0.2}$
2	1	7
3	9	4
4	3	12
5		

Table 1. A basic table

Let's break down the syntax. Since tables are floats, you can again control their placement with `[!htb]`. The `{r l c}` controls the number of columns and their alignment (right, left, center). The | adds vertical lines. `\hline` adds horizontal lines. Table entries are separated by `&`, and new rows created with `\\`. If the number of `&` per line doesn't match the number of columns −1, LaTeX will throw an error, even if the cells are blank.

Aastex has some unique (and very nice) table environments, which are described in detail at http://journals.aas.org/authors/aastex/aasguide.html#tables. These are automatically formatted in the correct style for publication in ApJ. They do have some unique syntax. They also have nice ways to rotate tables and make tables that span more than one page.

With few exceptions, you should never manually type a LaTeX table. You will regret it when you have to update the entries. The way around this is that you can insert other .tex files into the main LaTeX document using the `\input{}` command. So my suggestion is that you script your LaTeX table generation to pull in whatever data will go in the table and write out the headers, footers, and correctly formatted data. I have scripts to do this in Matlab and Python if you'd like to see examples. There are also online code generators that will do this for you. An example input table is Table 2 (adapted from Levy et al. 2018), which is also an example of a aastex "deluxetable".

Acknowledgments of people who helped you and funding sources go here. Also any acknowledgements that telescopes/instruments/etc require should go here. R.C.L. would like to thank Peter Teuben for his help with `git`.

Table 2. Parameters for the Persic Profile Fits

Name	CO V(R_{opt})	CO L_B	CO R_{opt}	Hα V(R_{opt})	Hα L_B	Hα R_{opt}
	(km/s)	($\times 10^8$ $L_{B\odot}$)	(arcsec)	(km/s)	($\times 10^8$ $L_{B\odot}$)	(arcsec)
IC1199	168.7	2.8	7.2	181.7	8.8	14.4
NGC2253	145.9	1.7	6.3	146.6	2.0	10.5

NOTE—V(R_{opt}), L_B, and R_{opt} for CO and Hα are the parameters from the Persic Profile fitting (Equation 1).

REFERENCES

Kennicutt, Jr., R. C. 1998a, ARA&A, 36, 189

—. 1998b, ApJ, 498, 541

Levy, R. C., Bolatto, A. D., Teuben, P., et al. 2018, ApJ, 860, 92

Persic, M., Salucci, P., & Stel, F. 1996, MNRAS, 281, 27

APPENDIX

A. EXAMPLE APPENDIX

Appendices work just like regular sections. You can add figures, tables, etc.

B. BACKING UP WITH GIT

The cloud is wonderful until it goes "poof". Overleaf has some internal version control (see the "History" button), but I always make sure I have a backed up version of my projects on my laptop. This is great for a number of reasons including if I accidentally delete something or really screw something up (definitely been there...) or if I want to work on my document while I'm not online. An easy way to backup is to use `git` (all credit to Peter Teuben for showing me this).

First, to create the link between your Overleaf project and computer, open a terminal and navigate to a directory where you want to store your backup. In Overleaf, click on "Share" and copy the read and edit link. In the terminal, type:

```
git clone https://git.overleaf.com/19205845gqmwfqgjwnhb backup
```

where the copied link to the project is `https://www.overleaf.com/19205845gqmwfqgjwnhb` and the "`www`" is replaced with "`git`". `backup` is the name of a new directory that will be created; if `backup` already exists, this command will fail. You only need to do this once per project. Then anytime you want to backup your project, simply

```
cd backup
git pull
```

Boom. Done. All backed up. All changes are updated, new files are added, and old files are deleted.

This is currently not supported with Overleaf v2 :(Hopefully this comes back up soon because it is very, very nice. This will work with documents created under v1 and not yet exported to v2. In the mean time, for v2 files you can download the source (all files associated with this project) as a .zip file under Menu.