LaTeX Tutorial for ENGR 100-600 at the University of Michigan

Created January 2025
Updated January 2025

ENGR 100-600 Staff and IAs
College of Engineering
University of Michigan
Ann Arbor, MI 48109

Table of Contents

1 Introduction to LaTeX

LaTeX is a typesetting language that allows you to create beautiful documents without spending too much time formatting, especially with mathematical equations, figures, and tables.

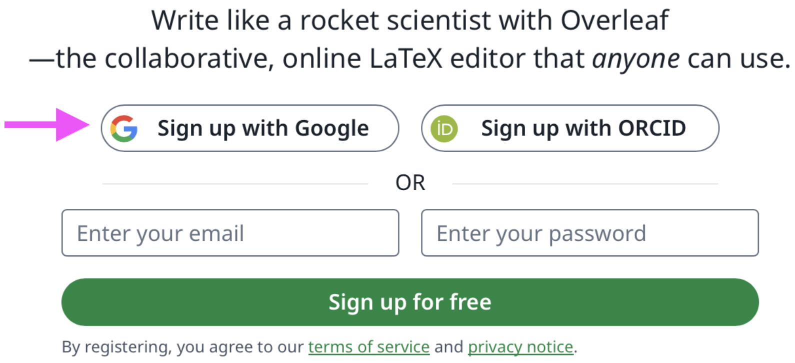
For this class, we provide you with templates for your technical communication assignments with the formatting already set. This means that all you need to do to use these documents is learn the basic commands so that you can add more headings, figures, tables, and text.

In this tutorial, we will go over the basic LaTeX information you need for creating written documents in ENGR 100-600. This tutorial serves as both an introduction to creating documents using LaTeX as well as an example of how to do things in LaTeX, such as making tables, referencing figures, creating equations, etc.

2 Introduction to Overleaf

LaTeX is sort of like a programming language, so in theory you can code it in anything as simple as the text editor on your computer. In practice, it is much easier to code in an integrated development environment (IDE) so that you can run your code, and hence visualize your document, as you go. For this, we suggest using **Overleaf**, an online collaborative IDE that is free for U-M students.

Go to overleaf.com and hit the Sign Up button like in Figure 1. Create your Overleaf account with Google by signing into your UMich Google account. The university offers a premium Overleaf account for free – you should make use of it!



Write like a rocket scientist with Overleaf
—the collaborative, online LaTeX editor that *anyone* can use.

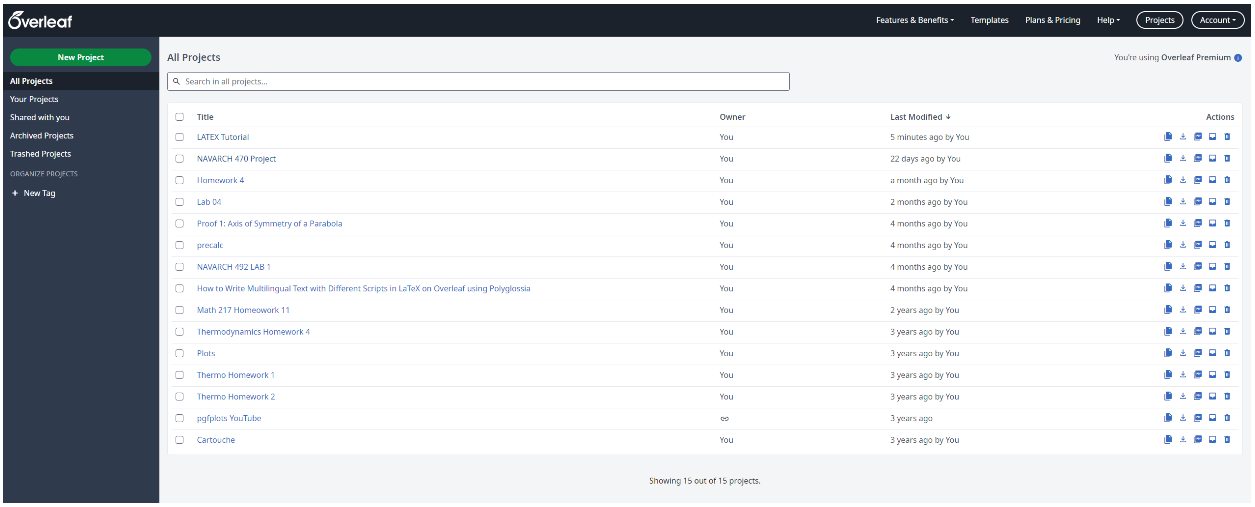
Enter your email

Enter your password

By registering, you agree to our terms of service and privacy notice.

Figure 1: Sign-up screen for Overleaf. Click "Sign up with Google" and then use your U-M email address.

Follow the steps to finish setting up your account. You'll be greeted with the screen shown in Figure 2 when you successfully have an account. Of course, you will not have any files.



All Projects

You're using **Overleaf Premium**

☐	Title	Owner	Last Modified ↓	Actions
☐	LATEX Tutorial	You	5 minutes ago by You	
☐	NAVARCH 470 Project	You	22 days ago by You	
☐	Homework 4	You	a month ago by You	
☐	Lab 04	You	2 months ago by You	
☐	Proof 1: Axis of Symmetry of a Parabola	You	4 months ago by You	
☐	precalc	You	4 months ago by You	
☐	NAVARCH 492 LAB 1	You	4 months ago by You	
☐	How to Write Multilingual Text with Different Scripts in LaTeX on Overleaf using Polyglossia	You	4 months ago by You	
☐	Math 217 Homeowork 11	You	2 years ago by You	
☐	Thermodynamics Homework 4	You	3 years ago by You	
☐	Plots	You	3 years ago by You	
☐	Thermo Homework 1	You	3 years ago by You	
☐	Thermo Homework 2	You	3 years ago by You	
☐	pgfplots YouTube	🔗	3 years ago	
☐	Cartouche	You	3 years ago by You	

Showing 15 out of 15 projects.

Figure 2: Overleaf Home Screen. If you just made an Overleaf account, there won't be anything listed here, and that's okay!

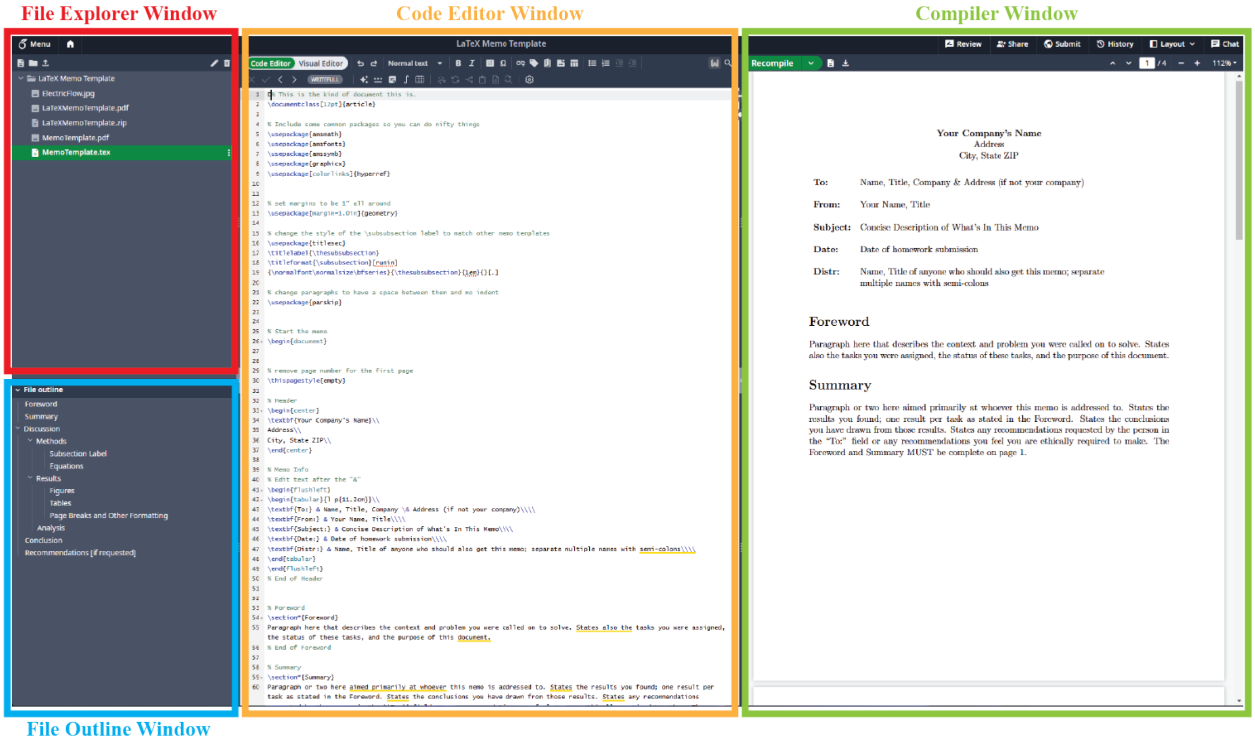
Here, you can sort and tag your projects, similar to your Google Drive or OneDrive. We highly recommend making use of the 'tags' feature for better organization!

To create a new project, click the green 'New Project' button. This will open a blank document where you can start from scratch! To open an old project, you may need to select the 'All Projects' tab if you do not own the file.

3 Beginning Your Document

Similar to other programming languages, you will need to configure the LaTeX document in order to achieve the desired result. Since we have provided the templates for your use, we have already set up the document and added comments to help you understand the document set-up. `Comments` are lines within your code (that start with a percent sign, %) that do not appear in your final document.

Once you are in the document, your screen will look like Figure 3.



```latex
% This is the kind of document this is.
\documentclass[12pt]{article}

% Include some common packages so you can do nifty things
\usepackage{amsmath}
\usepackage{amsfonts}
\usepackage{amssymb}
\usepackage{graphicx}
\usepackage[colorlinks]{hyperref}


% set margins to be 1" all around
\usepackage[margin=1.0in]{geometry}

% change the style of the \subsubsection label to match other memo templates
\usepackage{titlesec}
\titlelabel{\thesubsubsection}
\titleformat{\subsubsection}[runin]
{\normalfont\normalsize\bfseries}{\thesubsubsection}{1em}{}[.]

% change paragraphs to have a space between them and no indent
\usepackage{parskip}


% Start the memo
\begin{document}


% remove page number for the first page
\thispagestyle{empty}

% Header
\begin{center}
\textbf{Your Company's Name}\\
Address\\
City, State ZIP\\
\end{center}

% Memo Info
% Edit text after the "&"
\begin{flushleft}
\begin{tabular}{l p{11.2cm}}\\
\textbf{To:} & Name, Title, Company \& Address (if not your company)\\\\
\textbf{From:} & Your Name, Title\\\\
\textbf{Subject:} & Concise Description of What's In This Memo\\\\
\textbf{Date:} & Date of homework submission\\\\
\textbf{Distr:} & Name, Title of anyone who should also get this memo; separate multiple names with semi-colons\\\\
\end{tabular}
\end{flushleft}
% End of Header


% Foreword
\section*{Foreword}
Paragraph here that describes the context and problem you were called on to solve. States also the tasks you were assigned, the status of these tasks, and the purpose of this document.
% End of Foreword

% Summary
\section*{Summary}
Paragraph or two here aimed primarily at whoever this memo is addressed to. States the results you found; one result per task as stated in the Foreword. States the conclusions you have drawn from those results. States any recommendations
```

Your Company's Name
Address
City, State ZIP

To:	Name, Title, Company & Address (if not your company)
From:	Your Name, Title
Subject:	Concise Description of What's In This Memo
Date:	Date of homework submission
Distr:	Name, Title of anyone who should also get this memo; separate multiple names with semi-colons

Foreword

Paragraph here that describes the context and problem you were called on to solve. States also the tasks you were assigned, the status of these tasks, and the purpose of this document.

Summary

Paragraph or two here aimed primarily at whoever this memo is addressed to. States the results you found; one result per task as stated in the Foreword. States the conclusions you have drawn from those results. States any recommendations requested by the person in the "To:" field or any recommendations you feel you are ethically required to make. The Foreword and Summary MUST be complete on page 1.

Figure 3: Overleaf Editor Layout

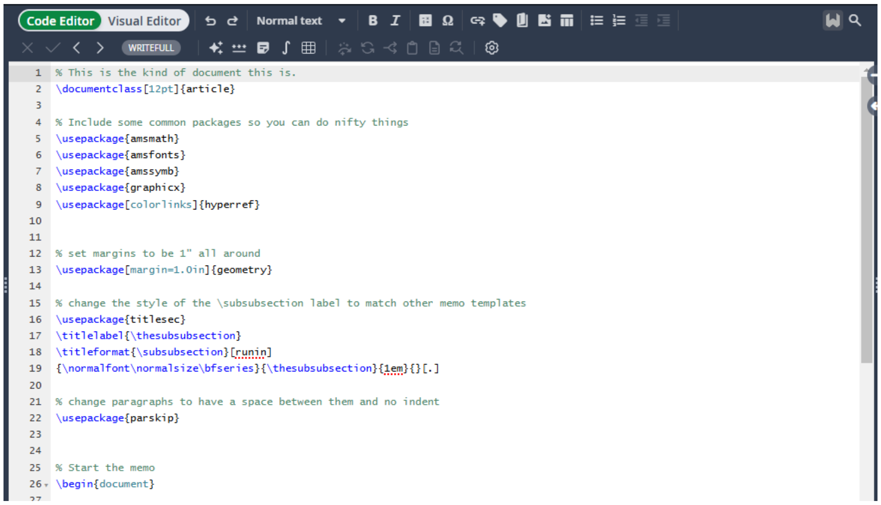
The important things to note are as follows:

- The "Code Editor" window, where you can actually type to edit your document.
- The "Compile" window, where you can see your document as it will appear in its final formatting.
- The file explorer, where your project documents are kept.
- The file outline, showing you the different sections and subsections in your document.

3.1 Code Editor

Don't let the word "code" freak you out! There's not *that* much coding involved. The code is mostly just the text of your document, and anything related to formatting (italics/bold, section headers, hyperlinks, tables/figures/equations, bulleted lists, figure references, paragraph spacing, etc.) is done as "code".

Figure 4 shows what your code editor window should look like. At the top of the window is a ribbon full of tools and shortcuts that help you write the document. We won't go over what each one does, but if you hover your cursor over the icons, a short description should show up. You can also go to the Overleaf User Guide to learn more about how to use Overleaf, beyond what we're covering in this tutorial.



```
% This is the kind of document this is.
\documentclass[12pt]{article}

% Include some common packages so you can do nifty things
\usepackage{amsmath}
\usepackage{amsfonts}
\usepackage{amssymb}
\usepackage{graphicx}
\usepackage[colorlinks]{hyperref}


% set margins to be 1" all around
\usepackage[margin=1.0in]{geometry}

% change the style of the \subsubsection label to match other memo templates
\usepackage{titlesec}
\titlelabel{\thesubsubsection}
\titleformat{\subsubsection}[runin]
{\normalfont\normalsize\bfseries}{\thesubsubsection}{1em}{}[.]

% change paragraphs to have a space between them and no indent
\usepackage{parskip}


% Start the memo
\begin{document}
```

Figure 4: Example Code Editor Window

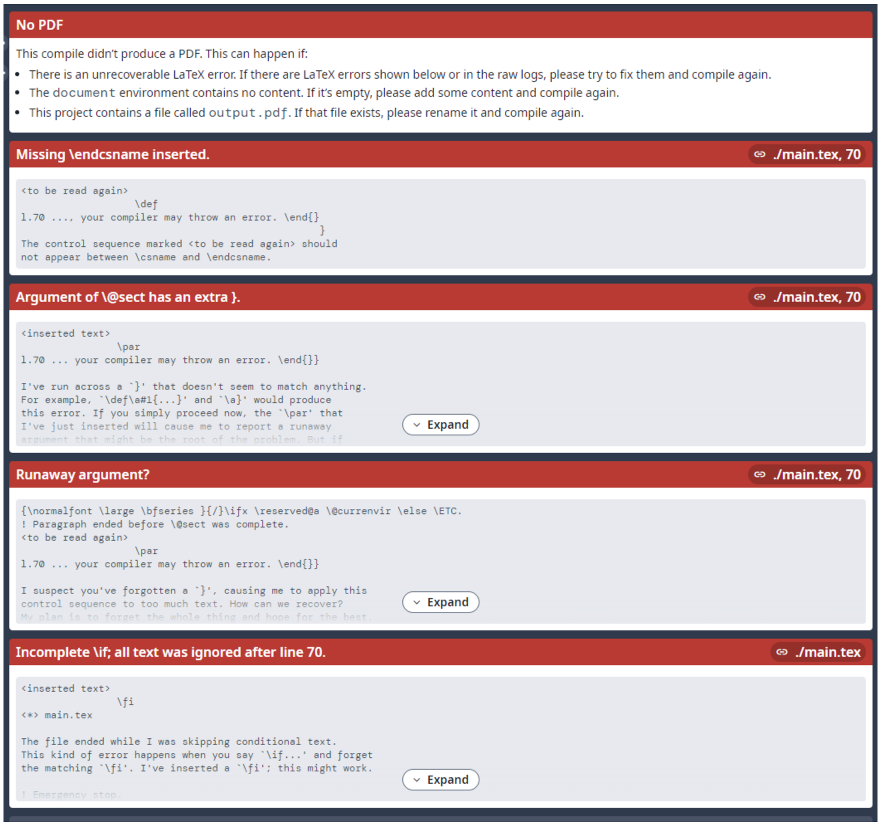
Each document needs a `\begin{document}` and an `\end{document}`, which tells LaTeX where your document starts and ends. Ahead of the `\begin{document}` is all of the packages and initial format setup. For the most part, the packages the ENGR 100-600 team included will be all you need. There should be no need to touch anything above `\begin{document}`. There should be nothing after `\end{document}`.

Since Overleaf is an online collaborative IDE, you and all of your teammates can work together at the same time on one document (kind of like working on a Google Document). You are able to follow each others' changes in real time and add "comments" like Google Documents on lines of code so that others can see.

3.2 Compiling

The compile window on the far right shows the document formatted. Basically, it interprets all the commands that are included in the code window and spits them out in a visual format. You cannot edit the file in the compile window, only the editor, but you can "jump" to a section in your editor by double-clicking the line in your compile window. When you make changes in the code editor, hit the "recompile" button, or simply ctrl + s, to view your changes.

If you make an mistake in the code, your compiler may throw an error, which will look like Figure 5. Yeah, this can be scary... especially if you're close to finishing your document. Just take a breath and look where your code has turned to red in the Code Editor window. Chances are, you've misplaced a curly brace ({}) in a command. Follow the instructions that the compiler gives you, and you will have it fixed :)



No PDF

This compile didn't produce a PDF. This can happen if:

- There is an unrecoverable LaTeX error. If there are LaTeX errors shown below or in the raw logs, please try to fix them and compile again.
- The `document` environment contains no content. If it's empty, please add some content and compile again.
- This project contains a file called `output.pdf`. If that file exists, please rename it and compile again.

Missing \endcsname inserted. ./main.tex, 70

```
<to be read again>
                   \def
l.70 ..., your compiler may throw an error. \end{}
                                                  }
The control sequence marked <to be read again> should
not appear between \csname and \endcsname.
```

Argument of \@sect has an extra }. ./main.tex, 70

```
<inserted text>
                \par
l.70 ... your compiler may throw an error. \end{}}

I've run across a `}' that doesn't seem to match anything.
For example, `\def\a#1{...}' and `\a}' would produce
this error. If you simply proceed now, the `\par' that
I've just inserted will cause me to report a runaway
argument that might be the root of the problem. But if
```

Runaway argument? ./main.tex, 70

```
{\normalfont \large \bfseries }{/}\ifx \reserved@a \@currenvir \else \ETC.
! Paragraph ended before \@sect was complete.
<to be read again>
                   \par
l.70 ... your compiler may throw an error. \end{}}

I suspect you've forgotten a `}', causing me to apply this
control sequence to too much text. How can we recover?
My plan is to forget the whole thing and hope for the best.
```

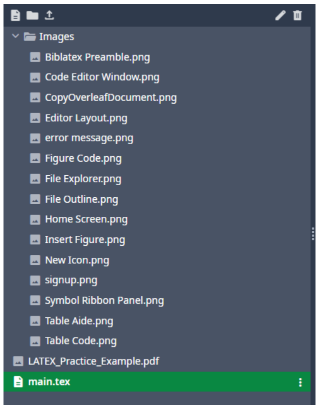
Incomplete \if; all text was ignored after line 70. ./main.tex

```
<inserted text>
                \fi
<*> main.tex

The file ended while I was skipping conditional text.
This kind of error happens when you say `\if...' and forget
the matching `\fi'. I've inserted a `\fi'; this might work.

! Emergency stop.
```

Figure 5: Compiler Error Message

3.3 File Explorer

The file explorer (on the top left) is where you keep the files you will insert into the document, as you can see in Figure 6. You can insert a figure either by using the upload button in the top left or by copy-pasting or by drag-and-dropping a file into the code window. After uploading a file to your file explorer, you might get a pop-up window like Figure 11 that gives you options on formatting your figures. Go to Section §5.1 on page 12 to read more about inserting figures into your document.



- Images
  - Biblatex Preamble.png
  - Code Editor Window.png
  - CopyOverleafDocument.png
  - Editor Layout.png
  - error message.png
  - Figure Code.png
  - File Explorer.png
  - File Outline.png
  - Home Screen.png
  - Insert Figure.png
  - New Icon.png
  - signup.png
  - Symbol Ribbon Panel.png
  - Table Aide.png
  - Table Code.png
- LATEX_Practice_Example.pdf
- main.tex

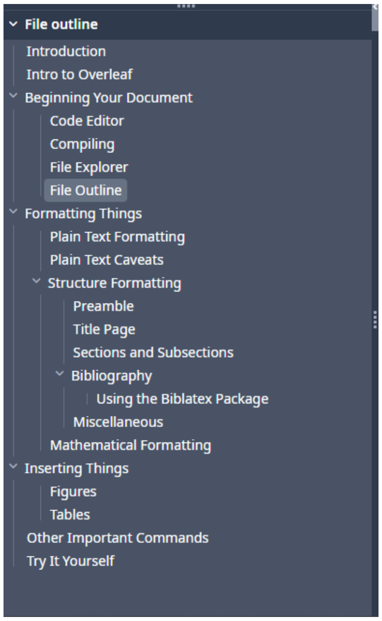
Figure 6: File Explorer

You can view your images by clicking on the file in the file explorer. Your code editor window will be replaced by your image – don't freak out! You can get back to your code by clicking on your "main.tex" file.

If you have lots of images and files you'd like to insert into your compiled document, you may want to organize them with folders. Click the folder icon in Figure 8 to add a new folder to your filer explorer. You can see how we reference figures inside folders if you look at the LaTeX code for this document in the Code Editor window in Overleaf.

3.4 File Outline

Much like the title suggests, the file outline (on the bottom left) is an organizational tree that helps you visualize the structure of your document, as seen in Figure 7.



File outline

- Introduction
- Intro to Overleaf
- Beginning Your Document
  - Code Editor
  - Compiling
  - File Explorer
  - File Outline
- Formatting Things
  - Plain Text Formatting
  - Plain Text Caveats
  - Structure Formatting
    - Preamble
    - Title Page
    - Sections and Subsections
    - Bibliography
      - Using the Biblatex Package
    - Miscellaneous
  - Mathematical Formatting
- Inserting Things
  - Figures
  - Tables
- Other Important Commands
- Try It Yourself

Figure 7: File Outline

You can easily navigate to different parts of your code by simply clicking on the appropriate section title. You can collapse larger sections using the drop-down arrows if needed. You can "jump" to specific sections in your document by double-clicking the section titles in the file outline.

4 Formatting Things

Typesetting can be tricky in LaTeX. Here's a quick guide to some things you might encounter writing your technical communication assignments for ENGR 100-600.

4.1 Plain Text Formatting

To alter the formatting of your plain text, LaTeX uses commands.

- To **bold** a word, use `\textbf{}` or just ctrl + b, where your word or phrase lies within the parentheses.
- To *italicize* a word, use `\textit{}` or just ctrl + i.
- To underline a word, use `\underline{}` or just ctrl + u.

4.2 Plain Text Caveats

There are some caveats due to the nature of programming in a typesetting language.

- To use quotation marks (“ ”) in plain text, use the back ticks (`) in place of the left quotation mark and the apostrophes (´) in place of the right quotation mark in your code. They can be single (‘ ’) or double (“ ”).
- Some symbols have special meanings in LaTeX. For example, to use the percent sign (%), the pound sign (#), the ampersand (&), or the dollar sign ($) in plain text, you will need to precede each symbol with a backslash (\) in your code.

4.3 Inserting Hyperlinks

In addition to all the other plain text formatting you can do in LaTeX, you can also insert hyperlinks. `Hyperlinks` are links in your current document that redirect you to another URL, document, or file. We use hyperlinks in this LaTeX tutorial to direct you to other places that can better explain certain commands and concepts. You can open our code editor to see how hyperlinks work, or you can read about them in the Overleaf guide (see what I did there! ;)).

Overleaf denotes hyperlinks differently from plain text. The default formatting is a colored box outlining the links, but we've changed the formatting in our preamble using the `hyperref` and `hypersetup` packages so that a hyperlink appears as different colored text depending on what it redirects you to.

Hyperlinks within technical documents can greatly enhance your reader's ability to navigate your document. However, you should not use hyperlinks *by themselves or alone* to reference the resources and research that you have done. You should always properly cite any external links that you want to reference. Including a hyperlink *in addition* to your formal citation is acceptable, and possibly a really good idea depending on the reference. Go to Section §4.4.4 on page 9 to read about references and citations and how they apply to ENGR 100-600.

4.4 Structure Formatting

In addition to formatting your plain text, you will also have to format the structure of your technical communication assignments. This includes adding a title/title page, section headers, subsections, page breaks and page numbers. In the context of ENGR 100-600, the templates we provide already structure the title page and give you some basic structure such as section headers and page numbers. This section will help guide you through the rest of your formatting.

4.4.1 Preamble

This refers to the first few lines of code in your Overleaf file that are used to set up the document. Here, you can change things like the paper size, margins, the look of the sections, and more. You do this by including packages into your file. There are also additional packages you can include that allow you to do special functions in LaTeX, like the `bibliography` package, the `hyperref` package, and the `fancyhdr` package.

In the template technical communication documents we've provided you, we commented the purpose of each of the packages we've included. Usually you won't have to include any extra ones, but there may be a case where you want to do something fancy that we haven't included instructions to.

After the preamble, you begin your document with `\begin{document}`, and end your document with `\end{document}`. Be sure all of your code lies within these two lines.

4.4.2 Title Page

The title page of both the memo template and report template have been already formatted for you. We used commands to center text, left align text, bold text, and change font sizes as needed and you can learn more by reading the comments. You can use the `\tableofcontents` command to add a table of contents to the beginning of your report. Just remember to start it on a new page using the `\newpage` command!

4.4.3 Sections and Subsections

Sections and subsections are crucial for the reader to understand your report. You can create a new section header (the big ones) by using the command `\section{}`, where your section header is within the curly braces. The nice thing about LaTeX is that it will keep track of your sections without you having to manually update the section numbers or table of contents. `\section*{}` will create sections that are unnumbered.

To create a new subsection (the medium sized ones), you can use the `\subsection{}` command, where the subsection heading is within the curly braces. Similarly, to create a sub-subsection (the little ones), you can use the `\subsubsection{}` command, where the sub-subsection heading is within the curly braces.

4.4.4 Bibliography

This section is always a doozy. There are a couple shortcuts that LaTeX provides when creating a bibliography (also known as a references page or a "works cited" page). For the purposes of ENGR 100-600, it might be easier to format your own citations and manually input them into your bibliography.

Using the Biblatex Package To streamline citing your sources, LaTeX has a package you can use called `biblatex`. To learn more about it, you can visit the Overleaf Biblatex Guide.

To use the biblatex package, you must first build a `bibtex` data file. You can create a bibtex data file by going to the file editor and clicking the "new file" icon in the top left, like in Figure 8. Be sure to rename the file with a `.bib` extension, like `datafile.bib`. This creates a blank file with the correct extension, which we can use with the bibtex package. The bibtex data file includes all your citation information (such as author, title, year, publisher, etc) in a specific syntax so that the biblatex package can read all the information. In your bibtex data file, you must list each source separately and specify the type of source (article, book, webpage, etc) you are citing. Please go to the Overleaf Biblatex Guide to view templates and the structure of your bibtex file – we can't include all the info here!

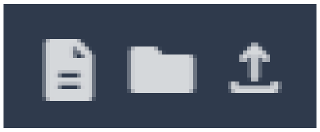
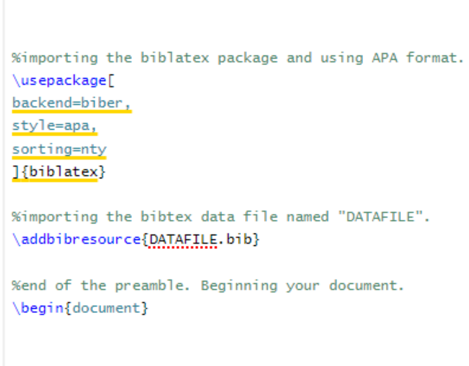


Figure 8: New File Icon

You will use this bibtex file to reference the data file in your main code using commands and labels. These commands can create a "Works Cited" page and in-line citations. You can create labels for each entry to reference in your main code using the biblatex package.

After you have set up your bibtex data file, you can begin working in your main code to format your bibliography. Regarding your "Works Cited" page, you have the option to change the *style* of the citations. There are many styles of citations depending on the purpose of your document and your audience. For the purposes of ENGR 100-600, use the `APA` style of citations by using the `style=apa` parameter and the `sorting=nty` parameter when including the biblatex package. Figure 9 shows how importing the biblatex package will look in your preamble.



```
%importing the biblatex package and using APA format.
\usepackage[
backend=biber,
style=apa,
sorting=nty
]{biblatex}

%importing the bibtex data file named "DATAFILE".
\addbibresource{DATAFILE.bib}

%end of the preamble. Beginning your document.
\begin{document}
```

Figure 9: Biblatex Package

Some other helpful items:

- To change the title of the "Works Cited" page, use the `\printbibliography[title={TITLE HERE}]` command instead of just the `\printbibliography` command.
- To create in-text citations, use the command `\cite{}`, where the label of your entry lies within the curly braces ({}).

4.4.5 Miscellaneous Structure Formatting

Here are other miscellaneous formatting things you might find interesting:

- To comment out code so it isn't seen in your document, start each line with a percent (%).
- To begin a new page (also known as a page break) in your document, you can use `\newpage`.
- To customize the page numbers on your document, you can use the `fancyhdf` package. To learn more, visit the Overleaf Learning Guide. For now, we've set up the page numbers in the templates provided.

4.5 Mathematical Formatting

There is some notation LaTeX uses to help differentiate between plain text and mathematical text, equations, symbols, or variables.

- You can refer to the "Symbol" ribbon panel at the top of your code editor as seen in Figure 10. Here, you can insert math equations, various symbols, Greek letters, and common notation without remembering how to type out the command.

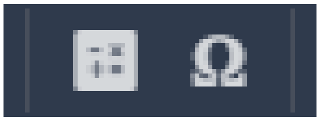


Figure 10: Symbol Ribbon Panel

- To denote a mathematical equation or symbol, use the dollar sign ($) on the left side to "begin", and use another dollar sign on the right side to "end" your mathematical text (`$ MATH TEXT HERE $`). This will insert the equation inline, like this: $e^{i\pi} = -1$. Be sure to always use dollar signs in pairs.
- To separate the math onto a new line, use two dollar signs (`$$ MATH TEXT HERE $$`), like so:

$$e^{i\pi} = -1$$

- To have numbered equations on their own lines, you can use `\begin{equation}` and `\end{equation}` to bookend your mathematical text. You can use `\equation*` or just (`$$ MATH TEXT HERE $$`) for unnumbered equations, like the two examples above. Go to Section §5.3 on page 15 to read about labeling equations.

$$E = mc^2 \tag{1}$$

- To denote superscripts (such as x^2), use the caret sign (ˆ). To denote subscripts (such as x_2), use the underscore sign (_).
- To denote a several characters or a phrase as a superscript or subscript (such as F_{n-1}), use the curly braces ({}) to group the phrase.
- To use Greek letters and common notation, there are specific commands such as `\sum` for $\sum$, or `\lambda` for λ, or `\nabla` for ∇. Remember to denote a mathematical expression prior to using these commands. Greek letters are just their names, preceded by a backslash, with the lowercase versions beginning with a lowercase letter and the uppercase versions beginning with an uppercase letter. For example, `\$Delta$` produces Δ, and `$\delta$` produces δ.

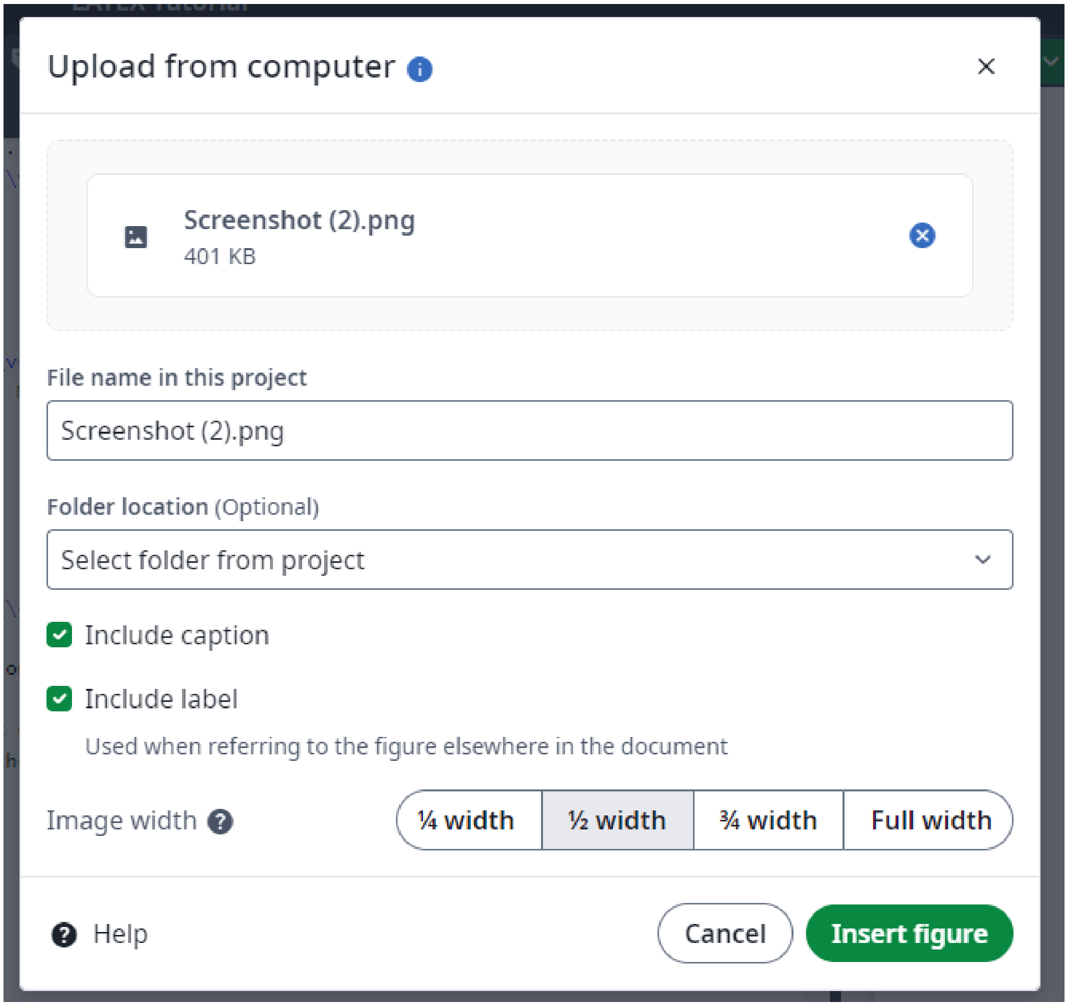
5 Inserting Things

For your documents, you will be expected to insert and reference figures and tables. This section will help you insert and reference both figures and tables. We also highly recommend the Overleaf Documentation for Figures and Tables.

5.1 Figures

The fourth symbol in the "Insert" ribbon panel is the button to insert figures. You can insert files that are not already in your current file explorer, figures that are in other projects, and figures from a URL. You can also drag-and-drop images into Overleaf.

LaTeX needs a package called `graphics` to manage images, since it cannot do so by itself. To use this package, be sure to include `\usepackage{graphics}` in your preamble. Then, you can begin inserting your figures with `\begin{figure}` and ending your figures with `\end{figure}`. If you use the button in the ribbon, it will automatically add these bookends. Figure 11 shows what using the ribbon to insert a figure looks like.



Upload from computer

Screenshot (2).png
401 KB

File name in this project
Screenshot (2).png

Folder location (Optional)
Select folder from project

☑ Include caption

☑ Include label
Used when referring to the figure elsewhere in the document

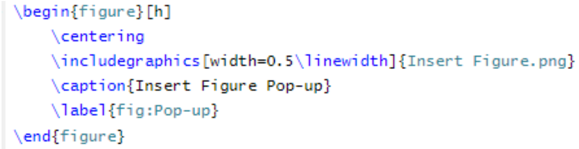
Image width: ¼ width | [✓ ½ width] | ¾ width | Full width

Figure 11: Insert Window

You can rename the file in Overleaf – it won't change the name of the file in your computer's directory. Overleaf will also throw a big yellow error box if you accidentally rename the file to something already in your file explorer. In this pop-up window, you have the ability to change the image size. You can change this later in your code using the `\width` parameter in the `\includegraphics` command. Overleaf will also ask you to include a caption and a label. Keep these boxes checked.

- A *caption* is a short phrase that describes your figure or table. It should be concise but accurately describe the contents.
- A *label* is like a tag for a specific section of code, or entry. Similar to the bibliography, labeling your figures and tables will allow LaTeX to automatically keep track of the numbering for you. You will also be able to reference any figure you've labeled with the command `\ref{fig:LABEL}` where "LABEL" is the name of your label.
- The `\ref{}` command *only* spits out a number. Therefore, in the text, when referencing your labeled figure, table, or equation, you have the choice to use Figure, Fig., etc. You are free to, within reason, create your own style here, but be sure to stay consistent throughout your document.

Figure 12 has example code. "Insert Figure" is the name of the file we are inserting. To reference the figure in our plain text, we've created a label called `fig:Pop-up`. If you use the `fig:` qualifier before Figures and the `tab:` qualifier before tables, they will be counted in parallel, and your figure and table numbers will run independently from each other and of the section number.



```
\begin{figure}[h]
    \centering
    \includegraphics[width=0.5\linewidth]{Insert Figure.png}
    \caption{Insert Figure Pop-up}
    \label{fig:Pop-up}
\end{figure}
```

Figure 12: Example Code for Inserting Figures

You can change the width of the figure by changing the `width=` argument in the `\includegraphics` command. The width in this command refers to the fraction of the linewidth that the page takes up. Therefore, a width of `0.5\linedwith` on an 8.5x11 page with 1 inch margins will be 3.25 inches wide ($0.5 \cdot (8.5'' - 2 \times 1'')$), and as tall as the aspect ratio dictates.

5.2 Tables

The rightmost symbol in the "Insert" ribbon panel is the button to insert a table. This will allow you to easily insert a table up to 10x10 without having to create the formatting yourself. Should you need a table larger than this, insert a table with 10 rows or columns (whichever you need more than 10 of) and then edit the tabular environment yourself.

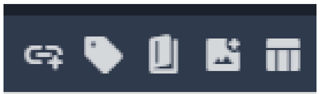


Figure 13: Insert Ribbon Panel

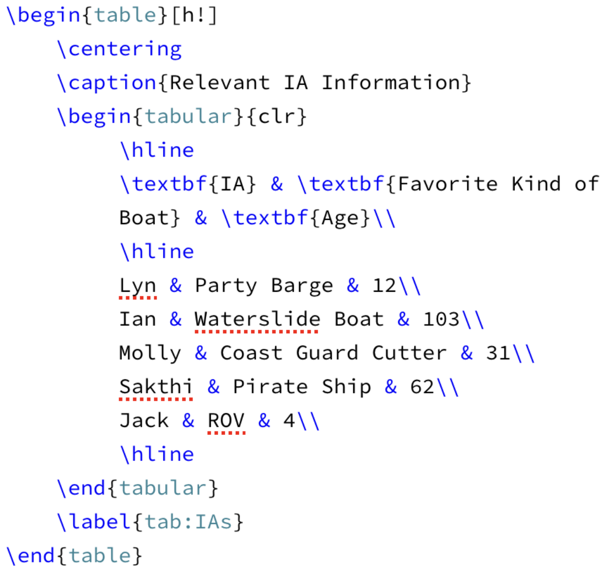
Tables are in a `tabular` environment, book-ended by `\begin{tabular}` and `\end{tabular}`. If you use the button in the ribbon, it will automatically add these, just like it does for figures. The argument for the tabular environment is the number of rows. The number of characters you have inside the braces is the number of columns. Left aligned columns are indicated by a lowercase letter L (l), center aligned columns are indicated by a lowercase C (c), and right aligned columns are indicated by a lowercase R (r). The first line of a table will be `\begin{tabular}{rlc...rlc}` where the information in the second pair of curly braces is the centering for each column you have.

In the table itself, the columns are separated by ampersands (&) and the lines are separated by a double back slash (\\). You can use the command `\hline` to create a horizontal line. If you wish to include vertical lines in your table (not recommended except in very specific cases) you can insert a vertical bar (|) (shift + backslash) between the corresponding (rs), (ls), and (cs). Therefore, using the command `\begin{tabular}{r|ccc}` will create a four-column table with the left column right aligned, the other three center aligned, and with a vertical line between the first and second columns.

See the code below Table 1 that was used to generate it.

Table 1: Relevant IA Information

IA	**Favorite Kind of Boat**	**Age**
Lyn	Party Barge	12
Ian	Waterslide Boat	103
Molly	Coast Guard Cutter	31
Sakthi	Pirate Ship	62
Jack	ROV	4



```
\begin{table}[h!]
    \centering
    \caption{Relevant IA Information}
    \begin{tabular}{clr}
        \hline
        \textbf{IA} & \textbf{Favorite Kind of
        Boat} & \textbf{Age}\\
        \hline
        Lyn & Party Barge & 12\\
        Ian & Waterslide Boat & 103\\
        Molly & Coast Guard Cutter & 31\\
        Sakthi & Pirate Ship & 62\\
        Jack & ROV & 4\\
        \hline
    \end{tabular}
    \label{tab:IAs}
\end{table}
```

Figure 14: Example Table Code

Creating the actual tables in LaTeX is probably the one thing that really is harder to do than in other programs. One thing that Prof. Alford has found helpful is the Tables Generator website. You can copy and paste in values from some other program (such as Excel), add in your basic formatting/style, and then generate the LaTeX "code" for the table. You can copy the LaTeX version of the table into your Overleaf document and then tweak it from there.

5.3 Equations

To label your equations (to reference them in the text), you will need to use the `\begin{equation}` and `\end{equation}` environment, not the $$ – $$ environment. Simply add the line `\label{eq:LABEL}` just after the `\begin{equation}`. Just like figures and tables, you can reference the equation numbers using `\ref{eq:LABEL}` in the text. It is common to reference equations either as "Eq. 1" or as "(1)." This is how we did Eq. 1 back on page 11.

5.4 Tip: Keeping Your Labels and Numbers Together

It's good style to have your label and number always be together, such as "Figure 12". If the "Figure" part is on one line but the "12" part is on the next part, that can be confusing. You can use the *tilde* character to represent a "non-breaking space" so that your label and number don't get separated. It looks like this:

> As shown in `Fig.~\ref{fig:LABEL}`, the ROV...

You can see more examples if you look at the LaTeX code for this document in the Code Editor window in Overleaf.

5.5 Why is That There?

Tables and figures are *floats*: that means that LaTeX will automatically organize your figures and tables in a reasonable way. This is especially nice because LaTeXwill always make sure your figures and tables never get separated from their captions and figures and tables will never span over a page break. This can save you a *lot* of time in formatting your document!

You can tell LaTeX where you want your figures and tables placed, such as *right **h**ere*, *at the **t**op of the page*, *at the **b**ottom of the page*, or *on a **p**age of all figures and/or tables.* You should always give LaTeX at least two options for placing figures so it has enough "wiggle room" to place everything for you.

To tell LaTeX where to place your float, include the parameters that correspond to your locations in square brackets `[ ]` next to the `\begin{...}` statement for your figure or table. For example, if you want to include a figure here (`h`) if that works out, but at the top (`t`) if it doesn't, then you would have this line to start your figure:

```
\begin{figure}[ht]
```

Figures or tables that are too large to fit in the available space on the current page will be placed on the next page, with the next block of text continuing on the current page so that there isn't some large amount of blank space on the current page that would look weird to your reader. Generally, this is okay: the figure or table is just on the next page and you can refer the reader to that page using the `\pageref` citation.

If you really need the figure or table to be *right here*, even if that's not an ideal location, you can add an exclamation point to your parameters to tell LaTeX it's okay to override its usual decision making and put the figure there anyways:

```
\begin{figure}[h!]
```

You can see an example of this in Line 1 in Figure 14. When using `[h!]`, carefully check the “compiled document” to make sure that the figure or table is still readable and hasn’t been cut off. If something has been cutoff, you can you can reduce the size of the table or figure. But this is something you should do in your **final round of formatting**.

When you have a lot of floats, especially larger ones, sometimes LaTeX can get a little confused with placing your floats around your text and shoves *all* your floats to the end of your compiled `.pdf` document. You can force LaTeX to insert all the floats it’s trying to place so far by using the `\clearpage` command. This command will “clear the floats”; meaning, all the figures and tables in the document so far will all be placed before printing any of the following text. This command is *especially* helpful for any report that is very figure and table heavy, with small amounts of text scattered around large figures.

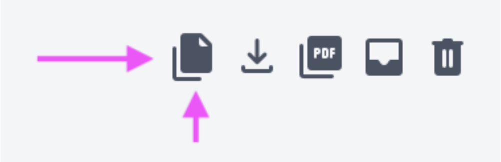
A combination of `[h!]` and `\clearpage` can usually solve any float placement issue you may run into. Ask any of your staff members if you’re having problems with float placement that you can’t solve with these two tricks... we’ll be able to help you! `\clearpage` is a much more extreme way of getting floats to place properly, so first, try `[h!]`.

6 Try It Yourself

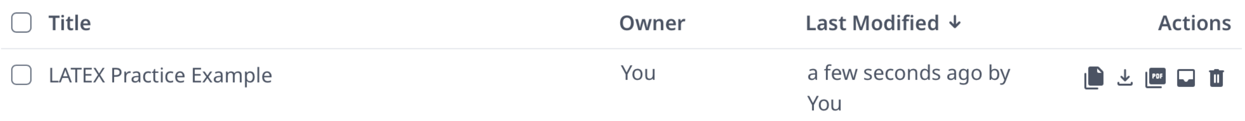
This section will take you through modifying an Overleaf template so that you can get some practice with LaTeX before creating your actual Individual Design Proposal, if you choose to use LaTeX for the assignment (we'd recommend it!). At the end of this section is an example of what your document should look like.

6.1 Make a Copy of the Template

Begin by accessing the ENGR 100-600 Individual Design Proposal Template from the course website or from this link. Join the project with your @umich email. Go to the homepage for Overleaf, and make a copy of the template, using the first of the four action buttons on the right of the screen:



Name the project something logical like *LATEX Practice Example*:



□ Title	Owner	Last Modified ↓	Actions
□ LATEX Practice Example	You	a few seconds ago by You	

Click the title of the document to open it.

6.2 Practice Updating the Header of the Frontmatter

Fill out the *to*, *from*, *subject*, *date*, and *distr.* fields in the frontmatter with some made-up data. Recompile the document to ensure that you have done this properly without changing anything. These pieces of information don't have to be correct now, as this is only a tutorial, so don't agonize over what to title your practice memo!

6.3 Practice Working with Figures

Now, remove Figure 1 and replace it with an image of your choosing that you have stored as a file on your computer. Import the image by dragging and dropping it into the code editor at the appropriate location. The upload window will appear. Before you hit OK, make the image have a width of 3/4, name the file something specific, and place it in the *My Images* folder. Once the code is generated, give the figure the label `fig:ROV`

Create a subsection after your Figure and give it a creative name. As text in the subsection, make one observation about your figure, and reference it in your text using the `\ref{}` command.

Now, take a screen shot (cmd + shift + 4 on a Mac, win + shift + s on a Windows). On Windows, it will automatically save it to your clipboard. On a Mac, if the screenshot appears in the lower right-hand corner, right click it and copy it. If it doesn't, it is already saved to your clipboard.

Remove Figure 2 and replace it with the image on your clipboard. Label this `fig:Controller`. Make this image to be on the same page as your section labeled **Controller**, and make sure that the image is *in* the section (not above it).

6.4 Practice Working with Tables

Under the buoyancy and ballast subsection, insert a table outlining three movie/TV characters, their names, what show/movie they appear in, their ages, and their jobs or passions. Each character will have their own row, with each column indicating aspects of who they are. Left-align the character names, right align the ages (so the ones places match up), and center the other two columns. The first row should be the column names, and the names should be in bold. You should have three horizontal lines in your table: one above the table, one below the table, and one between the first two rows. This is the layout identical to that seen in Table 1. You should reference the table in the subsection's text, and give the table a caption.

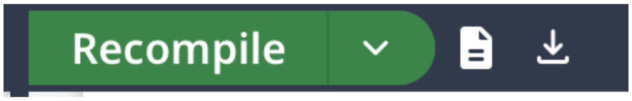
Make sure that the table appears in the section you intend it to by using `[h!]` and `\clearpage` if needed.

6.5 Practice Working with Equations

Finally, let's insert an equation with an equation number into the section labeled Overview of Vehicle Design. Add in Archimedes's equation, $F_B = \rho g \nabla$ as an out-of-line, numbered equation. Label it, and reference it in the text.

6.6 See Your Updated Document!

Click the *Recompile* button at the top of the window that shows the `.pdf` version of your document:



Fix any errors that you may have and then view your document! You can also download the document as a `.pdf` using the download button/icon near the *Recompile* button. You should recompile your document frequently as you write so that you catch any errors early.

An example practice memo, created by following the instructions above, can be found in the pages below. To view the code used to make this, access this link.

The LaTeX tutorial is complete. The following pages contain an example of what your "try it yourself" document should look like. Compare yours to the following pages to ensure you have correctly implemented what is discussed.

Global Friendship LLC
123 Squiggle St
Utqiagvik, AK 99723

To: His Majesty King Tutankhamun, Pharaoh, 18th Dynasty
King Julien, PhD, King of Lemurs, Madagascar

From: Perry the Platypus, Agent, OWCA

Subject: Promoting Cooperation Between Fictional Figures and Historical Figures from Over Three Millennia Ago

Date: July 4, 1776

Distr: Mahatma Gandhi, Peace Activist

Foreword

Sentence that describes the context and problem you were called on to solve. Sentences that state the tasks you were assigned and the status of these tasks (these tasks should reflect the layout of the body of the memo). Last sentence states the purpose of this document.

Summary

State the results you found; one result per task as stated in the Foreword. State the conclusions you have drawn from those results. State any recommendations requested by the person in the “To:” field or any recommendations you feel you are ethically required to make.

Proposed ROV Design

Introduction paragraph here that briefly describes the context for why you are designing an ROV. Include a sentence that describes the layout of the sections that follow. This paragraph does not have to be long.

Overview of Vehicle Design

One or two sentences that describe the goals for your design (e.g. maneuverability, transportability, sustainability, durability, aesthetic qualities, ease of building,...) and argue for the respective importance of whatever goals you chose. You should have no more than three goals. The vehicle was designed with regard to Archimedes's Principle, (1).

$$F_B = \rho g \nabla \qquad (1)$$

Sentence introducing your ROV, as shown in Fig. 1.

Figure 1: Important Information Regarding World Peace

Alpacas and World Peace

You can notice, as shown in Figure 1, that alpacas have ears, and therefore, can hear all the disagreements in the world.

Vehicle Subsystems

Sentence introducing the subsystems of the vehicle.

Frame. Describe your rationale for designing the frame the way you did. Don't forget to talk about the tether attachment point – the point where the tether attaches to the frame using the carabiner. You can add more figures here if it is helpful, but you don't have to. This should be one paragraph.

Thrusters. Describe your rationale for placing the thrusters where you did. You can add more figures here if it is helpful, but you don't have to. This should be one paragraph (and may only be 1-2 sentences).

Payload. Describe your rationale for placing the electronics canister and the batter canister where you did. You can add more figures here if it is helpful, but you don't have to. This should be one paragraph (and may only be 1-2 sentences).

Camera. Describe your rationale for placing the camera where you did. You can add more figures here if it is helpful, but you don't have to. This should be one paragraph (and will probably just be one sentence).

Buoyancy and Ballast. Describe your rationale for placing any buoyancy or ballast where you did; we would expect this rationale to refer to what you have learned in lab. You can add more figures here if it is helpful, but you don't have to. This should be one paragraph (and may only be 1-2 sentences). Table 1 outlines three historical figures and their contributions to global friendship.

Table 1: Inspirations for Global Cooperation

Character	Appears In	Age	Passion
Po Ping	Kung Fu Panda	15	The Furious Five
Candace Flynn	Phineas and Ferb	16	Busting her brothers
Philomena Cunk	Cunk on Earth	49	History

Controller

One or two sentences that describe the goals for your design (e.g. maneuverability, transportability, sustainability, durability, aesthetic qualities, ease of building,...) and argue for the respective importance of whatever goals you chose. You should have no more than three goals.

I am currently listening to the music from the playlist shown in Fig. 2.

Public Playlist

A Bossa Nova

Ian Rosenberg • 22 songs, 1 hr 29 min

#	Title	Album	🕒
1	Águas De Março Elis Regina, Antônio Carlos Jobim	Elis & Tom	3:34
2	O Pato Gilberto Gil	Gilbertos Samba	4:02
3	Só Danço Samba Stan Getz, Luiz Bonfá	Jazz Samba Encore!	3:36
4	O Leãozinho - Remixed Original ... Caetano Veloso	Bicho	3:05

Figure 2: Some Famous Bossa Nova Hits on a Playlist

Paragraph describing the rationale for your controller design. We expect that you will focus on the choice and placement of buttons, switches, etc. but you should describe anything that you think your reader should know.

Custom Part

One or two sentences that describe the purpose of the custom part.

Sentence introducing your custom part, as shown in Fig. 3.

Figure 3: Nice description of your custom part. Emphasize what problem this custom part solves.

Paragraph describing the rationale for your controller design.

Potential Drawbacks

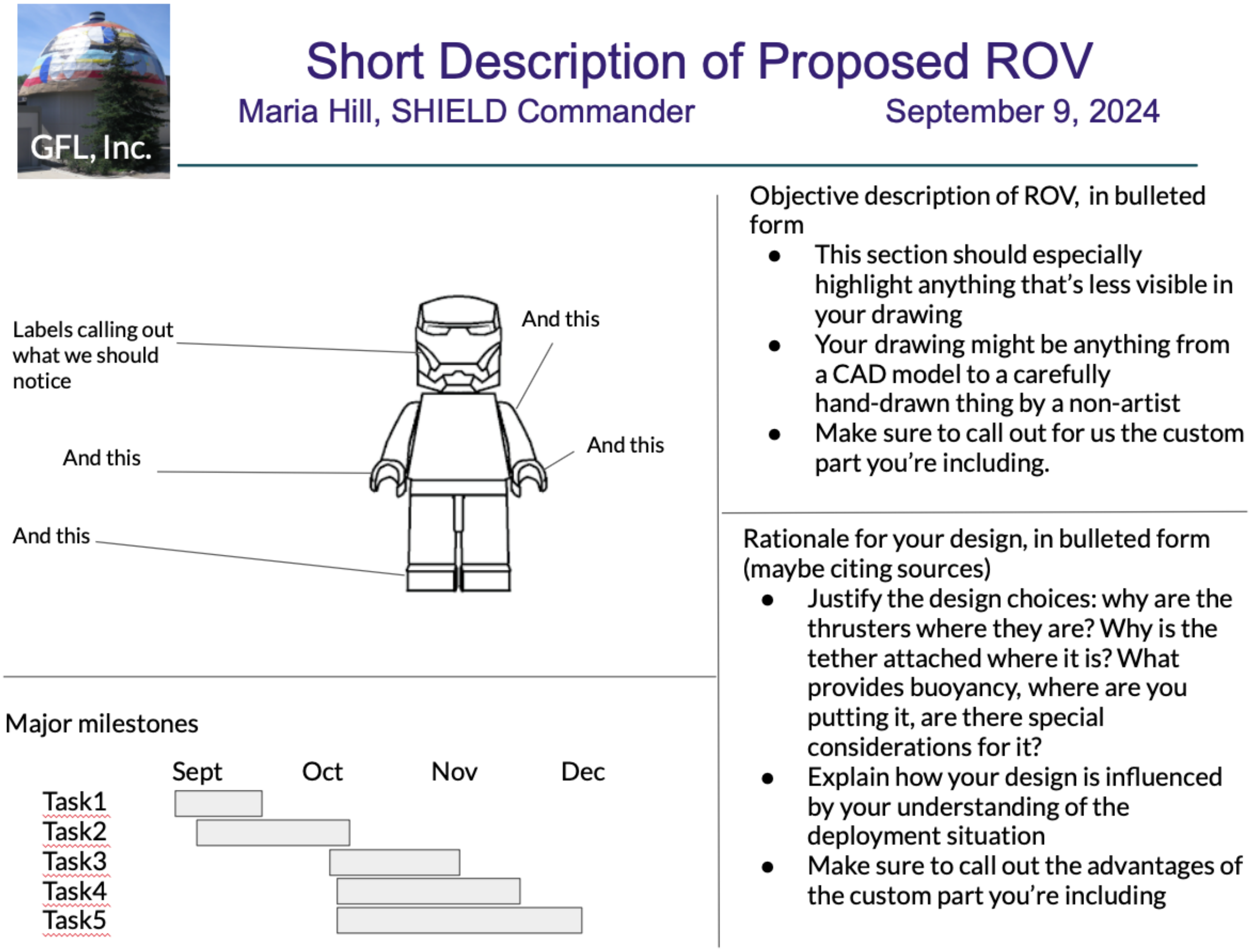
Paragraph describing the limitations inherent in this design and any foreseen challenges in building the vehicle and controller.

Conclusions

Describe your conclusions based on your tasks and results. This is where you can make commentary on everything you've done. For this short proposal, this will likely be only 3-4 sentences. Keep it concise.

Conclude this section by inviting contact (the client can contact you at uniqname@umich.edu, or whatever).

Appendix



GFL, Inc.

Short Description of Proposed ROV

Maria Hill, SHIELD Commander September 9, 2024

Labels calling out what we should notice
And this
And this
And this
And this

Major milestones

Sept
Oct
Nov
Dec
Task1
Task2
Task3
Task4
Task5

Objective description of ROV, in bulleted form

- This section should especially highlight anything that's less visible in your drawing
- Your drawing might be anything from a CAD model to a carefully hand-drawn thing by a non-artist
- Make sure to call out for us the custom part you're including.

Rationale for your design, in bulleted form (maybe citing sources)

- Justify the design choices: why are the thrusters where they are? Why is the tether attached where it is? What provides buoyancy, where are you putting it, are there special considerations for it?
- Explain how your design is influenced by your understanding of the deployment situation
- Make sure to call out the advantages of the custom part you're including