Using LaTeX for COMPSCI 390A

College of Information and Computer Sciences
University of Massachusetts

Spring 2021

1 Introduction

LaTeX is a programming language for creating `.pdf` text documents. Source code for a LaTeX document is stored in a file with the extension `.tex`. LaTeX is often used for formatting books and articles, and is also used extensively in computer science graduate studies. For example, see this page, which provides instructions for submitting research papers to one of the top ML conferences in 2020—one of the first lines provides a link to an example LaTeX project for formatting a paper submission, and all (or nearly all) submissions are created using LaTeX. This isn't because conferences force unwilling authors to use LaTeX—authors almost always *want* to use LaTeX, and such a large portion do that many conferences have stopped putting the effort in to support other systems.

One of the main benefits of using LaTeX is that it is very easy to create nicely formatted equations like this:

$$\begin{aligned}
f(x) =& x^2 + 2 \\
\frac{df(x)}{dx} =& \frac{d}{dx}\left(x^2 + 2\right) \\
=& 2x. \\
\int_0^1 f(x)\,dx =& \left(\frac{1}{3}x^3 + x^2\right)\Big|_0^1 \\
=& \frac{4}{3}.
\end{aligned}$$

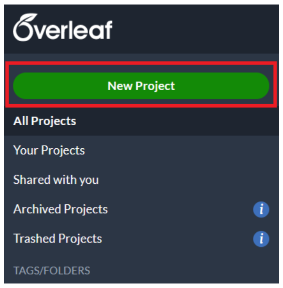
Even better, LaTeX makes adding references very easy—it can automatically create your bibliography for you (using various different styles)! If I type `cite{Sutton1998}` in the source, it creates the following citation and the bibliography at the end of this document: Sutton and Barto (1998). This does require a little bit of setup—you must define `Sutton1998` in a `.bib` (bibliography) file. However, the definitions of references can largely be copied directly from Google Scholar. See Appendix A for more details on citations. This may be a good reason to use LaTeX for other courses as well!

The reminder of this document will help you get set up to complete the math portion of Homework 2.

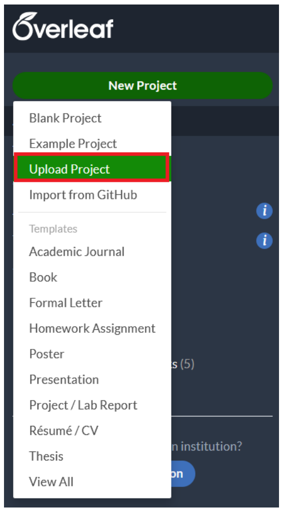
2 Setting up LaTeX

Like other programming languages, LaTeX source code must be compiled (in this case, into a `.pdf` file rather than an executable). You *can* download compilers and IDEs for writing LaTeX on your computer. However, LaTeXcompilers are often quite large and can be challenging to set up properly. Thankfully, there are online IDEs that work wonderfully. We recommend using a free account on Overleaf.

Once you have signed up for an account, click the "New Project" button in the top right:

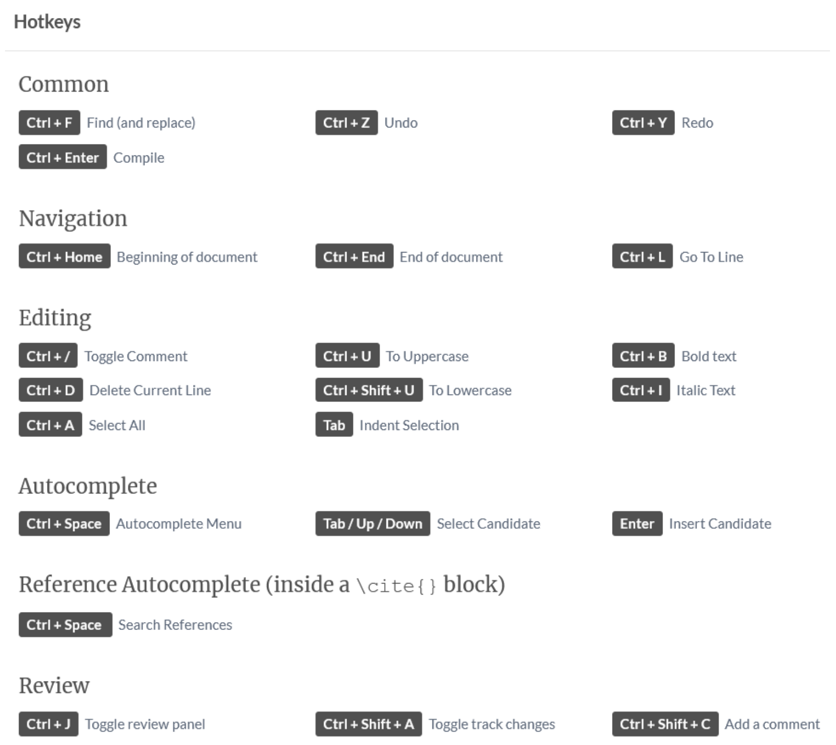


Select "Upload Project":



You will then be prompted to select a `.zip` file. Select the `.zip` file provided with the Homework 2 assignment titled `hw2_written.zip`. This will create a project for you. If `main.tex` is not already selected on the left side, click `main.tex` to open the main LaTeX file.

Finally, make note of the common keyboard shortcuts that you might use. These can be found by clicking the Menu button in the top left, and then selecting "Show Hotkeys" near the bottom of the menu that comes up. On Windows, the hotkeys are:


Hotkeys

Common

Ctrl + F	Find (and replace)	Ctrl + Z	Undo	Ctrl + Y	Redo
Ctrl + Enter	Compile				

Navigation

Ctrl + Home	Beginning of document	Ctrl + End	End of document	Ctrl + L	Go To Line

Editing

Ctrl + /	Toggle Comment	Ctrl + U	To Uppercase	Ctrl + B	Bold text
Ctrl + D	Delete Current Line	Ctrl + Shift + U	To Lowercase	Ctrl + I	Italic Text
Ctrl + A	Select All	Tab	Indent Selection		

Autocomplete

Ctrl + Space	Autocomplete Menu	Tab / Up / Down	Select Candidate	Enter	Insert Candidate

Reference Autocomplete (inside a `\cite{}` block)

Ctrl + Space	Search References

Review

Ctrl + J	Toggle review panel	Ctrl + Shift + A	Toggle track changes	Ctrl + Shift + C	Add a comment

For this assignment, you will be filling in your answers into the shell that we have provided in `main.tex`. In LaTeX any text after the % symbol is a comment. Search for the comment `% <your_answer_here>`, which is present in Homework 2 at each of the locations where you will enter your answers.

You should see the source code in `main.tex` on the left, with a window on the right showing the compiled `.pdf`. To recompile the `.pdf` (e.g., after you make changes), you can click the "Recompile" button or you can use the keyboard shortcut.

3 Equations

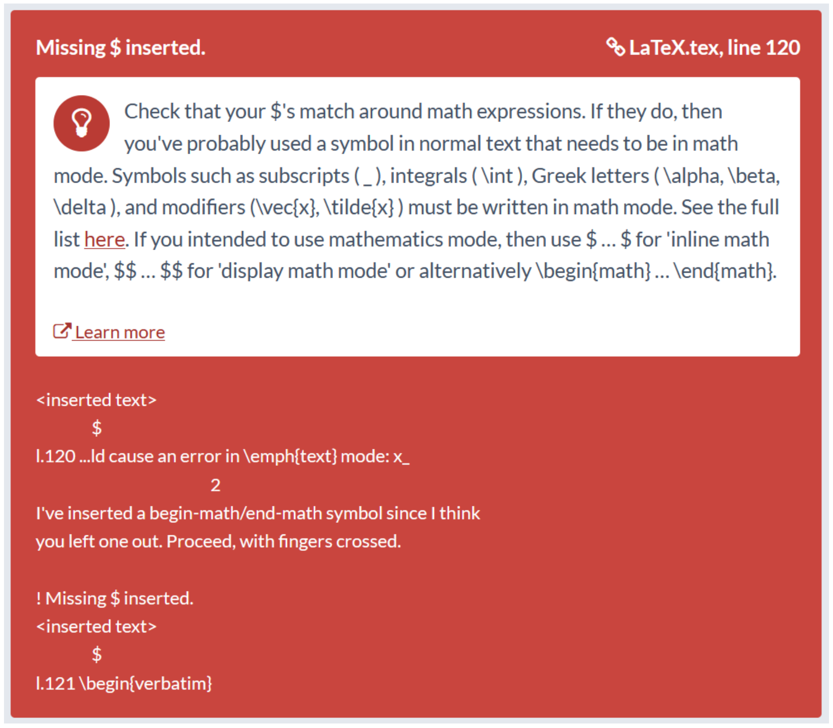
You can write math inline with text like this $f(x) = x^2$. In the source, that's

```
You can write math inline with text like this $f(x)=x^2$.
```

The $ symbol means to toggle between text-mode and math-mode. Different commands are allowed in math and text mode. For example, the following would cause an error in *text* mode:

```
x_2 is a variable.
```

When trying to recompile with this line in the source, I get the following error (this shows up in the window where the `.pdf` file would appear if it compiled properly):



```
Missing $ inserted.                                    LaTeX.tex, line 120

Check that your $'s match around math expressions. If they do, then
you've probably used a symbol in normal text that needs to be in math
mode. Symbols such as subscripts ( _ ), integrals ( \int ), Greek letters ( \alpha, \beta,
\delta ), and modifiers (\vec{x}, \tilde{x} ) must be written in math mode. See the full
list here. If you intended to use mathematics mode, then use $ … $ for 'inline math
mode', $$ … $$ for 'display math mode' or alternatively \begin{math} … \end{math}.

Learn more

<inserted text>
        $
l.120 ...ld cause an error in \emph{text} mode: x_
                                               2
I've inserted a begin-math/end-math symbol since I think
you left one out. Proceed, with fingers crossed.

! Missing $ inserted.
<inserted text>
        $
l.121 \begin{verbatim}
```

This is the LaTeX compiler saying that something went wrong—it tried inserting the missing $ symbol, because it ran into the underscore, _, which is not allowed in text mode. Notice that it also says the line number where this occurred in the top right. *Sometimes* you can also click on this red box, and you will be taken to the line with the error.

In this case, we can fix our source like this:

```
    $x_2$ is a variable.
```

The result is: x_2 is a variable.

Often you don't want equations to be embedded within your text. Instead, you may want them to stand out as an *equation block* of some sort, like this:

$$f(x) = ax^2 + bx + c.$$

This is achieved with the following code:

```
\begin{equation}
    f(x) = ax^2 + bx + c.
\end{equation}
```

Like most (good) programming languages (cough), spacing does not matter in LaTeX. For example, entering multiple spaces between words instead of one will make no difference in the rendered `.pdf`. In equation blocks, we often use tab-indenting to format equations so that they are easier to see.

Note: An alternative to a standard equation block is to use $$. Try to avoid this—it is better to use a proper equation block. However, here is an example using $$:

$$f(x) = ax^2 + bx + c.$$

Source:

```
$$
    f(x)=ax^2 + bx + c
$$
```

In LaTeX, newlines can *sometimes* be manually inserted with \\. However, equation blocks to not allow for newline commands. If you want equations that span multiple lines, there are a few options including: `array`, `eqnarray`, and `gather`. For now, we recommend beginning with `array`.

An `array` block is very much like the `equation` block that you already saw. However, it allows for multiple lines (denoted by \\):

$$\begin{gathered} f(x) = ax^2 + bx + c \\ g(x) = \int_{-\infty}^\infty \! \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}\,dx \end{gathered}$$

Source:

```
\begin{align}
    f(x) = ax^2 + bx + c\\
    %
    g(x) = \int_{-\infty}^\infty \! \frac{1}{\sqrt{2\pi\sigma^2}}
    e^{-\frac{(x-\mu)^2}{2\sigma^2}}\,dx
\end{align}
```

Notice that I placed a line with just a comment, %, between the two equation lines. This is optional, and just helps you to visually see where the source code is for each equation when you have large align blocks. Notice, however, that align blocks do *not* allow for empty lines. The following would cause an error due to the empty line:

```
\begin{align}
    f(x) = ax^2 + bx + c\\

    g(x) = \int_{-\infty}^\infty \! \frac{1}{\sqrt{2\pi\sigma^2}}
    e^{-\frac{(x-\mu)^2}{2\sigma^2}}\,dx
\end{align}
```

I guess this is an exception to the rule that whitespace doesn't matter in LaTeX ☺.

The real power of the `align` block comes from the alignment character, &. You can include this character on each line, and it will align the lines so that the alignment characters are vertically aligned. Good practice is to place the alignment character in an equation after the equals sign (or ≥ or ≤), but before the right hand side begins. Recall the first math block in this document:

$$\begin{aligned}
f(x) =& x^2 + 2\\
\frac{d f(x)}{d x}=& \frac{d}{d x}\left(x^2 + 2\right)\\
=& 2x.\\
\int_0^1 \! f(x) \,dx =& \left(\frac{1}{3}x^3 + x^2 \right)\Big\vert_0^1\\
=& \frac{4}{3}.
\end{aligned}$$

The source for this was:

```
\begin{align}
    f(x) =&x^2 + 2\\
    %
    \frac{d f(x)}{d x}=& \frac{d}{d x}\left(x^2 + 2\right )\\
    %
    =&2x.\\
    %
    \int_0^1 \! f(x) \,dx =&  \left (\frac{1}{3}x^3
    + x^2 \right )\Big \vert_0^1\\
    %
    =& \frac{4}{3}.
\end{align}
```

Notice the use of the alignment characters, as well as the comments to clearly separate equation lines. Here you'll also see some other useful commands. For example, `\left` and `\right` before parentheses scales the parentheses to be the height of the text inside the parentheses. Similarly, `\vert` produces a vertical line, and `\Big` makes this line larger. The commands `\!` and `\,` change the spacing, the former reduces the spacing between the surrounding symbols and the latter increases it slightly. Also notice that `\frac{a}{b}` produces $\frac{a}{b}$ when in math mode (align blocks, equation blocks, etc. all count as math mode).

If an equation is too long to fit on one line, you can split it onto two lines using an `align` block:

$$\begin{aligned}
f(x) =& ax^{25} + bx^{24} + cx^{23} + dx^{22} + ex^{21} + fx^{20} + gx^{19} + hx^{18} + ix^{17} + jx^{16} + kx^{15}\\
&+ lx^{14} + mx^{13} + nx^{12} + ox^{11} + px^{10} + qx^{9} + rx^{8} + sx^{7} + tx^{6} + ux^{5} + vx^{4}\\
&+ wx^{3} + xx^{2} + yx^{1} + z.
\end{aligned}$$

The source for this is:

```
\begin{align}
    f(x)=&ax^{25} + bx^{24} + cx^{23} + dx^{22} + ex^{21} + fx^{20}
    + gx^{19} + hx^{18} + ix^{17} + jx^{16} + kx^{15} \\
    %
    &+ lx^{14} + mx^{13} + nx^{12} + ox^{11} + px^{10} + qx^9 + rx^8
    + sx^7 + tx^6 + ux^5 + vx^4 \\
    %
    &+ wx^3 + xx^2 + yx^1 + z.
\end{align}
```

There are a few things to notice here. First, notice that the alignment character comes *before* the + symbol on lines that begin with +. This creates the desired alignment. Next, notice that we wrote `x^{25}` not `x^25`. This is because superscripts and subscripts can be thought of as taking in one object. In the case of `x^25`, that object is 2, and so this would render as x^25. Curly braces are used to lump text together into one object, and so `x^{25}` treats 25 as the superscript, resulting in x^{25}. Make note of this when writing sums so that you end up with $\sum_{i=1}^n x_i$ not $\sum_i = 1^n x_i$.

Finally, notice that you can write many useful symbols like greek letters: α, β, etc. You can find a list of common symbols here.

4 Conclusion

With what you've learned, you should be ready to tackle Homework 2! Some of the source code at the start and end of the file may still be a mystery, but you should be able to insert your answers in the specified locations. Good luck!

References

R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction.* MIT Press, Cambridge, MA, 1998.

A Citations

To include a bibliography, this document uses the package `natbib`:

```
\usepackage[round]{natbib}
```

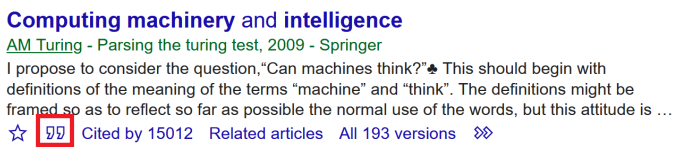
Additionally, near the end of the document you will find the following two lines. The first specifies the name of the bibliography (`.bib`) file—`mybib.bib`, and the second specifies the styling to use for the bibliography and references (`abbrvnat`):

```
\bibliography{mybib}
\bibliographystyle{abbrvnat}
```

Inside of `mybib.bib` is the following definition of the reference (Sutton and Barto, 1998):

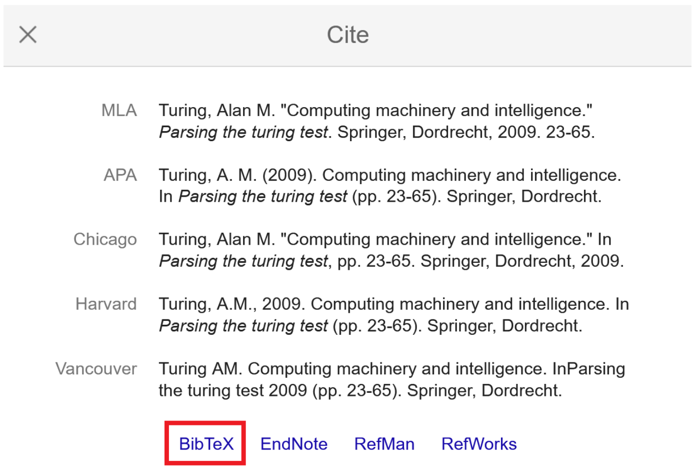
```
@book{Sutton1998,
  Author="R. S. Sutton and A. G. Barto",
  Title="Reinforcement Learning: {A}n Introduction",
  Publisher="MIT Press",
  Year="1998",
  Address="Cambridge, MA",
}
```

You can manually create the entries, with useful information for this approach here. Alternatively, you can find these definitions on Google Scholar. First, go to https://scholar.google.com/. Search for a paper, and once you have found it, click the quotation marks:



Computing machinery and **intelligence**
AM Turing - Parsing the turing test, 2009 - Springer
I propose to consider the question,"Can machines think?"♣ This should begin with definitions of the meaning of the terms "machine" and "think". The definitions might be framed so as to reflect so far as possible the normal use of the words, but this attitude is …

Next, click "BibTeX" to obtain the `.bib` source for the selected paper reference:



MLA	Turing, Alan M. "Computing machinery and intelligence." *Parsing the turing test*. Springer, Dordrecht, 2009. 23-65.
APA	Turing, A. M. (2009). Computing machinery and intelligence. In *Parsing the turing test* (pp. 23-65). Springer, Dordrecht.
Chicago	Turing, Alan M. "Computing machinery and intelligence." In *Parsing the turing test*, pp. 23-65. Springer, Dordrecht, 2009.
Harvard	Turing, A.M., 2009. Computing machinery and intelligence. In *Parsing the turing test* (pp. 23-65). Springer, Dordrecht.
Vancouver	Turing AM. Computing machinery and intelligence. InParsing the turing test 2009 (pp. 23-65). Springer, Dordrecht.

This should then provide you with the definition of the selected reference, which you can copy into your `.bib` file:

```
@incollection{turing2009computing,
  title={Computing machinery and intelligence},
  author={Turing, Alan M},
  booktitle={Parsing the turing test},
  pages={23--65},
  year={2009},
  publisher={Springer}
}
```

Often people keep one massive `.bib` file with all of the references they have used in the past. This way, you only need to add a reference to this file once—after that referencing it is as simple as calling:

```
\cite{turing2009computing}
```