An Incredibly Cool Thesis

A Thesis
Presented in Partial Fulfillment of the Requirements for the
Degree of Master of Science
with a
Major in Nuclear Engineering
in the
College of Graduate Studies
University of Idaho
by
Joe I. Vandal

Major Professor: Major Professor, Ph.D.
Committee Members: Committee Member, Ph.D.; Committee Member, Ph.D.
Department Administrator: Indrajit Charit, Ph.D.

May, 2023

Abstract

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.

Acknowledgements

This work and my coursework was completed under funding from...I would also like to thank //professor etc//. for their assistance

Dedication

To //person// who supported/inspired/etc. me.

Table of Contents

List of Tables

List of Figures

List of Codes

Acronyms

LWR Light Water Reactor.

Chapter 1

Introduction

LaTeX offers a lot of very nice features, including acronyms and citations. Dr. Trefor Bazett on Youtube has a great playlist to learn a lot of features.

1.1 Verbatim

Some symbols and words in LaTeX are interpreted as commands. In some cases you can escape the commands, but for others, the verbatim package can be handy. `\command` would normally be interpreted as a command by the compiler, but since I wrapped it with the inline verbatim environment it is interpreted literally. You can also do multiline verbatim sections, which may be useful for including code:

```
\command
\othercommand
```

The package also supports multiline comments, which allows block of LaTeXcode to remain in your .tex file without being generated into your document:

1.2 Citations

You can easily manage your citations using BibTex. Lamarsh is a good textbook for nuclear engineers [1]. The information is entered in the file References.bib in the working directory, and is called using the command `\cite{Lamarsh}` It automatically populates in the References section in the format specified either by a built in commmand, or a custom format defined by a .bst file in the working directory, in this case nsf.bst. This is handy because it automatically reorders the bibliography either alphabetically or in order of in text reference.

1.3 Custom Commands

The file uidahomasterthesis.cls contains some custom commands that will be useful. You can add your own, but I've included some common ones. `\flinak` prints $FLiNaK$. Custom commands can

include arguments, which may be optional. `\UF` prints UF_4, while `\UF[6]` prints UF_6. `\U` prints U , while `\U[235]` prints ^{235}U , and `\U[238]` prints ^{238}U . `\Xe` prints ^{135}Xe , while `\Xe[136]` prints ^{136}Xe and `\Xe[]` prints Xe

1.4 Acronyms

In general, acronyms should be fully spelled out for at least the first usage. The commands `\acs`, `\acl`, `\acf` are defined in the .cls file, and print the short version, long version, and both respectively. Doing this instead of typing out the acronym is good, because it automatically populates the List of Acronyms, and links the acronym to the List of Acronyms. It takes a little setup, defining acronyms in the .cls file, but it is worth it in the long run.

You call the full acronym Light Water Reactor (LWR) with `\acf{lwr}`. Later on, you can call the short version (LWR) with `\acs{lwr}` or the long version (Light Water Reactor). with `\acl{lwr}`.

Chapter 2

Background

You're probably gonna need to do some math. You can do inline equations like $a^2 + b^2 = c^2$ by wrapping with dollar signs. Especially useful for inline equations is `\nicefrac`, which is a package loaded, and does diagonal fractions instead of vertical. This can keep inline equations looking nice and uncramped. $K.E. = {}^{mv^2}/_{2}$. instead of $K.E. = \frac{1}{2}mv^2$. You will also want separate equations like Eqn. 2.1.

$$e = mc^2 \tag{Eqn. 2.1}$$

Curly braces are used in Eqn. 2.2 for multi character sub/superscripts. It also shows how to use greek characters. There are a ton of other features for LaTeX equations that you can find with Google.

$$-1 = e^{i\pi} \tag{Eqn. 2.2}$$

Referencing equation numbers using `\ref{label}` is handy because it updates the number if the numbering changes *e.g.* you add another equations.

In addition to mathematical equations, you may also want to include chemical (Rxn. 2.1 or nuclear (Rxn. 2.2) reactions. This template has a custom environment to have a separate counter and present with a different label than mathematical expressions.

$$CH_4 + 2O_2 \rightarrow CO_2 + 2H_2O \tag{Rxn. 2.1}$$

$$^6Li + n \rightarrow {}^3H + \alpha \tag{Rxn. 2.2}$$

Chapter 3

Results and Analysis

Tables and figures are handled well by LaTeX.

3.1 Tables

Tables can be generated using the tabular environment. To make it possible to reference them, include a caption, and automatically populate the list of tables, wrap this in the table environment.

Table 3.1: Relevant nuclear constants [1].

	γ (%)	λ (hr^{-1})	σ (Mb)
^{135}I	6.39	0.1035	-
^{135}Xe	0.237	0.0753	2.65

The caption contains two arguments. The first is wrapped in square brackets and is the "short" version that will populate in the List of Tables. The second is the "long" version, wrapped in curly braces, that will populate above the table. Table 3.1 has 4 columns, each of which has centered data, with a vertical line after the first column. This is specified by the argument `{c|ccc}` after `\begin{tabular}`. Columns are delimited by & and rows are delimited by \\. The label should come after `\end{tabular}` so the compiler doesn't get confused.

3.2 Figures

The figure environment can be used to load in various images and other figures. Like with the table environment, you can develop a label to reference the figure in text, and both a short version and long version of the caption. This is handy to keep your List of Figures tidy while allowing for very descriptive captions under the figure.

Static Images

These are included using the `\includegraphics[options]{name}` command. There is a lot more information on the internet, including Dr. Trefor's videos. Figure 3.1 is an example. The argument after the `\begin{figure}` is used to position the figure.

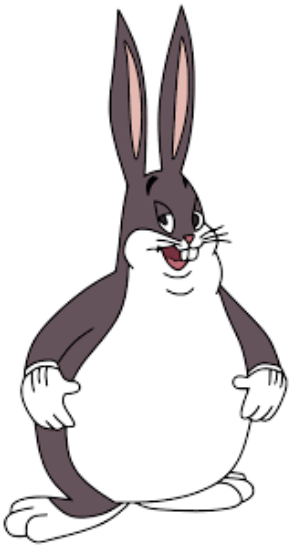

Figure 3.1: A humorous image of an overweight Buggs Bunny, often referred to as 'Big Chungus'.

User Generated Drawings

Tikz is a package that allows you to make very nice drawings right in LaTeX . Dr. Trefor has a tutorial. The learning curve is steep, and honestly is more difficult to use than GUI sandbox based drawing application, but it has the benefit of being able to keep typeface formatting very consistent. See how nice the equations look in Figure 3.2? Try doing that in MS Paint.

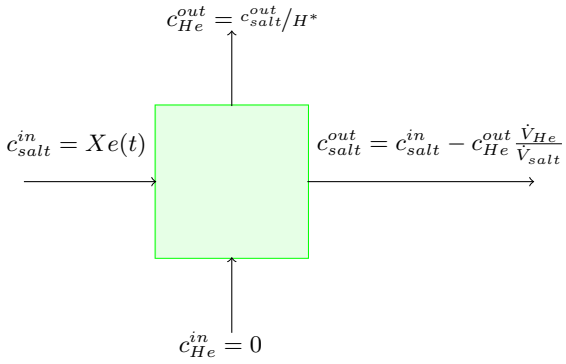

Figure 3.2: Schematic Drawing of Xenon Stripping Module

Animations

The graphicx package cannot accept .gif files, so the animate package can be used as a proxy. You will need each frame to be an individual .png, but this allows you to stack images over one another, which can give a time component to 2-D plots, or allow you to look at multiple planes of a drawing in a single figure. You will need to open the file in a proper PDF viewer like Adobe - web browsers and PDF previewers found in Overleaf, Atom, VSCode, etc. cannot handle this powerful functionality. I'm not sure if this functionality is allowed in the actual thesis submission, but it can be useful in presentations or meetings. Figure 3.3 is a visualization of the Bateman equations causing the xenon spike following the shutdown of a nuclear reactor. It loops through bar-200 to bar-248 at 2 frames per second. You can also step through frame by frame using the arrow controls. Note that feature can cause compile times to be very long so you may wish to comment it out.

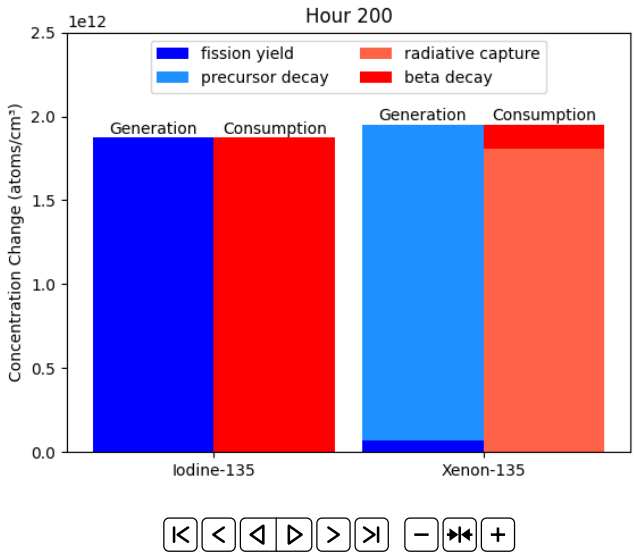

Figure 3.3: Nuclide Concentration Rates of Change - Reactor Scram

Chapter 4

Conclusions

4.1 Limitations

Bulleted lists are a good way to present statements when the order doesn't really matter. You also can nest lists if applicable.

- first bullet;
    - first subbullet;
    - second subbullet;
- second bullet; and
- third bullet;

4.2 Future Work

Numbered lists are another choice, and are good for sequential discussion.

1) First, put on your socks;
2) Then, put on your shoes;

4.3 Summary Remarks

Inline numbered lists are useful for a couple reasons: 1) To save space; and 2) to improve the flow of reading the list;

References

[1] Lamarsh, John R., Baratta, Anthony J., 2001. Introduction to Nuclear Engineering. Pretice Hall, Upper Sadle River, New Jersey, 3rd edition.

Appendix A

Codes

You may have done some coding in your thesis. You can share it with the verbatim package, but it looks a lot nicer to have a specific environment for code blocks. You can even include language specific syntax highlighting. The style file mypythonhighlight.sty in the working directory is set up for python, but you can find packages for other languages too!. I made a custom code environment which gives you a caption over the code block and lists it automatically in the List of Codes.

The simplest way to include code is to type it directly in the .tex file in the `\begin{python}` environment defined by the .sty file. Just like with figures, tables, and equations, you can label and reference them. Just like with the verbatim environment, indentation is preserved, as displayed by the (useless) example, Code 1

Code 1: Hello!

```
1     print("Hello World") #comment
2     try:
3         a=2/x
4     except ZeroDivisionError:
5         print('undefined')
```

You might be discussing a single aspect of your code in the body of your thesis. Inline codes like `import numpy` are very useful for doing this, distinguishing the commands from regular text by using the code font and syntax coloring.

My preferred way of including codes in the document is by using the custom `\inputpython` command. Code 2 displays lines 1 to 3 of test.py, located in the py subdirectory. It's an example of a very nice python feature, fstrings.

Code 2: F strings

```
1 x = 4
2 print(f"The numeral four: {x}")
3 #comment
```

You can customize the syntax highlighting by digging into mypythonhighlight.sty. This may be nice if you are using a specific library and want the keywords in that library to be highlighted as keywords, but the .sty file doesn't identify them as keywords by default.