Introduction to LaTeX and Its Use in Thesis and Assignment Preparation

Alex Hagen[(1)*]

(1) Purdue University

October 5, 2016 at GRIS 103
Lafayette, IN

Introduction: Outline

- ⇒ TeX Examples
- ⇒ When to use TeX
- ⇒ Demos
  - → Hello World
  - → Assignments
  - → PUThesis
  - → Publishing
- ⇒ Features
  - → Floats
  - → Mathematics
  - → Table of Conents
  - → Bibliographies
- ⇒ Advanced TeX

Introduction: What is LaTeX?

What LaTeX IS

- ⇒ Powerful for scripting
- ⇒ Hassle free formatting
- ⇒ Important for publishers
- ⇒ Important for mathematicians
- ⇒ Creates high quality figures

What LaTeX ISN'T

- ⇒ Quick
- ⇒ User Friendly
- ⇒ What You See is What You Get Editor
- ⇒ Useful for Art

Source Code 1 C Function Pointer Wrapping and Passing using `dlsym`

```c
// wrapDL.c
#include <stdlib.h>// standard functions
#include <stdio.h> // io functions (printf)
#include <dlfcn.h> // dynamic linking functions

typedef struct Coin{
        void *dl;                  //the pointer to the dynamic library
        void *cointoss;} Coin; //the pointer to the passed function

Coin *Coin_new(int seed) {  //constructor
        Coin *coin=(Coin *)malloc(sizeof(Coin));  //allocate the structure's mem
        srand(seed);                              //seed the rand num generator
        coin->dl = dlopen("./_playDL.so", RTLD_LAZY);  //open the dynamic library
        if (coin->dl == NULL) {printf("%s\n", dlerror()); exit(1);}
          //exit and print error if one occurs
        typedef int (*coint_t)(int myfunds, int yourfunds, int *tosses);
          //create a type for the fcn
        coint_t cointoss = (coint_t)dlsym(coin->dl,"cointoss"); //link the fcn
        coin->cointoss = cointoss; //make the pointer of that fcn an object var
        return coin;} //return the structure

int Coin_toss(Coin *coin, int myfunds, int yourfunds, int *tosses){
        int total; //make an integer to store the total
        typedef int (*coint_t)(int myfunds, int yourfunds, int *tosses);
          //create a type for the fcn (reduntant from instantiator)
        coint_t cointoss; //create a function with that type
        cointoss=coin->cointoss; //pass the fcn pointer to the function
        total = cointoss(myfunds,yourfunds,tosses); //use the function
        return total;} //return the total loss/gain

void Coin_delete(Coin *coin) {
        dlclose(coin->dl); //close the dynamic library
        free(coin);} //free the structure variable memory
```

Problem 2 - Wrapping and Function Passing from C to Python Wrapping in python is made slightly harder by the requirement of converting things to C language types, but this can be done by the ctype library. Luckily, only integers and pointers to integers must be used in driving this C function, so the wrapper is very simple, and returns the output shown in Figure 2 when run with the source code shown in Source Code 2.

```
$ python p2_testscript.py
***   Test 2   ***
The desired output is:
Average gain = -0.02
Average # tosses = 132.36
Average gain = -0.02
Average # tosses = 264.22

Your output is:
Average gain = -0.02
Average # tosses = 132.36
Average gain = -0.02
Average # tosses = 264.22
```

Figure 2: Sample Output using Tester `p2_testscript.py`

Task 3 - Combining Scalar and Vector Visualization

Because the interesting parts of the current dataset are the vortexes, a scalar calculation of the vorticity of the dataset can provide supplementary data to the vector visualization. In order to determine the surfaces of interest, a slider bar was used to probe the vorticity magnitude data set. When the surfaces of interest were discovered (that which corresponded to the layer of air which is pushed away from the vortex, that which corresponded to the outer surface of the vortex, and that which corresponded to the inner core of the vortex), they were then plotted with streamlines supplemented. Because of the the focus of the visualization, a rake was created which followed both sides of the wing, to provide a symmetric visualization of the air being trapped in the vortexes. This coincided perfectly to be included in between the outer and inner surfaces provided by the vorticity dataset. The final rendering shown in Figure 6 provides an example of this.

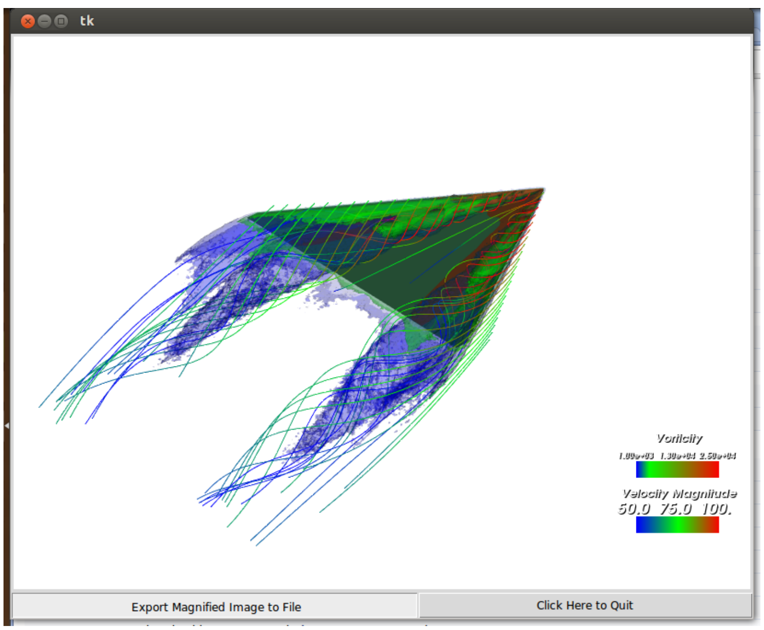


Figure 6: Combined Rendering of CFD Velocity Vector and Vorticity Scalar Data Set (**Link to Hi-Resolution**)

NUCL 575 HMWK 2

Alex Hagen

9/26/13

Problem 1 [1, prob. 8.1] The backpropagation training algorithm is developed in Section 8.3, using the logistic function as the activation function where the derivative has the convenient form given in Equation 8.1-4. Derive the backpropagation training algorithm for the case where the activation function is an arctan function where the derivative is given by Equation 8.1-6.

For the case where the activation function is given by as

$$\Phi(I) = \frac{2}{\pi} \arctan(\alpha I) \tag{8.1-5}$$

we must find the result, given by

$$\frac{\partial \Phi(I)}{\partial I} = \frac{2}{\pi}\left[\frac{\alpha}{1+\alpha^2 I^2}\right] \tag{8.1-6}$$

This can be done by developing the backpropagation training algorithm as in section 8.3.

We must first relate the total error ε to each parameter involved, of which we have the activation function $\Phi(I)$ and the intensity αI. The derivative of this total error is proportional to the change we will make to the weight at that point, as given by

$$\Delta w_{pq.k} = -\eta_{p.q} \frac{\partial \varepsilon_q^2}{\partial w_{pq.k}} \tag{8.3-3}$$

So, the critical part of this is to find the derivative $\frac{\partial \varepsilon_q^2}{\partial w_{pq.k}}$. This can be found by the chain rule, therefore

$$\frac{\partial \varepsilon_q^2}{\partial w_{pq.k}} = \frac{\partial \varepsilon_q^2}{\partial \Phi_{q.k}} \frac{\partial \Phi_{q.k}}{\partial I_{q.k}} \frac{\partial I_{q.k}}{\partial w_{pq.k}} \tag{8.3-4}$$

Because of the solutions given in section 8.3, and the result obtained in Equation , we can put all of these pieces together to get a full algorithm.

$$\begin{array}{ccccccc}
\frac{\partial \varepsilon_q^2}{\partial w_{pq.k}} & = & \underbrace{\frac{\partial \varepsilon_q^2}{\partial \Phi_{q.k}}} & \cdot & \underbrace{\frac{\partial \Phi_{q.k}}{\partial I_{q.k}}} & \cdot & \underbrace{\frac{\partial I_{q.k}}{\partial w_{pq.k}}} \\
\frac{\partial \varepsilon_q^2}{\partial w_{pq.k}} & = & -2\left[T_q - \Phi_{q.k}\right] & \cdot & \frac{2}{\pi}\left[\frac{\alpha}{1+\alpha^2 I^2}\right] & \cdot & \Phi_{p.j}
\end{array}$$

Therefore, similar to Equations 8.3-9 - 8.3-12, we have

$$\frac{\partial \varepsilon_q^2}{\partial w_{pq.k}} = -\frac{4}{\pi}\phi_{w.k}\left[T_q - \Phi_{q.k}\right]\left[\frac{\alpha}{1+\alpha^2 I^2}\right] = \delta_{pq.k}\Phi_{p.j}$$

and

$$w_{pq.k}(N-1) = w_{pq.k}(N) - \eta_{p.q}\delta_{pq.k}\Phi_{p.j}$$

NUMERICAL METHODS

As described in Brian Bradie's *A Friendly Introduction to Numerical Analysis* Bradie [1], and given the setup in Figure 7, where the boundary condition at $j = 0$, $i = 0$ is vacuum, reflected boundary at $j = 1$ and $i = 2$, and running at 45^o.

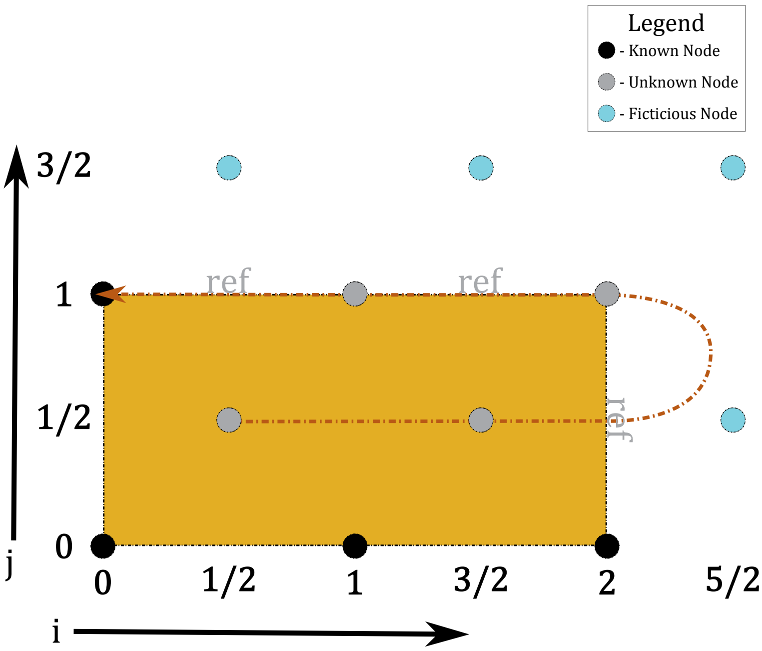


Figure 7: Illustration of Finite Difference Method

The following will solve the problem
For point $(\frac{1}{2}, \frac{1}{2})$

$$\psi_{\frac{1}{2},\frac{1}{2}} = \frac{S_{\frac{1}{2},\frac{1}{2}} + \frac{2|\mu_m|}{\Delta x}(\psi_{1,1} - \psi_{0,0}) + \frac{2|\eta_m|}{\Delta y}(\psi_{1,0} - \psi_{0,1})}{\sigma_{tr,\frac{1}{2},\frac{1}{2}} + \frac{2|\mu_m|}{\Delta x} + \frac{2|\eta_m|}{\Delta y}} \tag{9}$$

where $S_{\frac{1}{2},\frac{1}{2}} = \nu\sigma_{f,\frac{1}{2},\frac{1}{2}} \cdot \phi_{old,\frac{1}{2},\frac{1}{2}}$, and $\sigma_{tr,\frac{1}{2},\frac{1}{2}}$, μ_m, Δx, Δy, and η_m are known quantities. in this case, $\psi_{0,0}$, $\psi_{0,1}$, and $\psi_{1,0}$ are known (vacuum boundary). Thus, the above equation has two unknowns: $\psi_{\frac{1}{2},\frac{1}{2}}$, and $\psi_{1,1}$.

Now we move on to point $(\frac{3}{2}, \frac{1}{2})$, and we have the equation

$$\psi_{\frac{3}{2},\frac{1}{2}} = \frac{S_{\frac{3}{2},\frac{1}{2}} + \frac{2|\mu_m|}{\Delta x}(\psi_{2,1} - \psi_{1,0}) + \frac{2|\eta_m|}{\Delta y}(\psi_{1,1} - \psi_{0,2})}{\sigma_{tr,\frac{3}{2},\frac{1}{2}} + \frac{2|\mu_m|}{\Delta x} + \frac{2|\eta_m|}{\Delta y}} \tag{10}$$

Examples: When to Use LaTeX

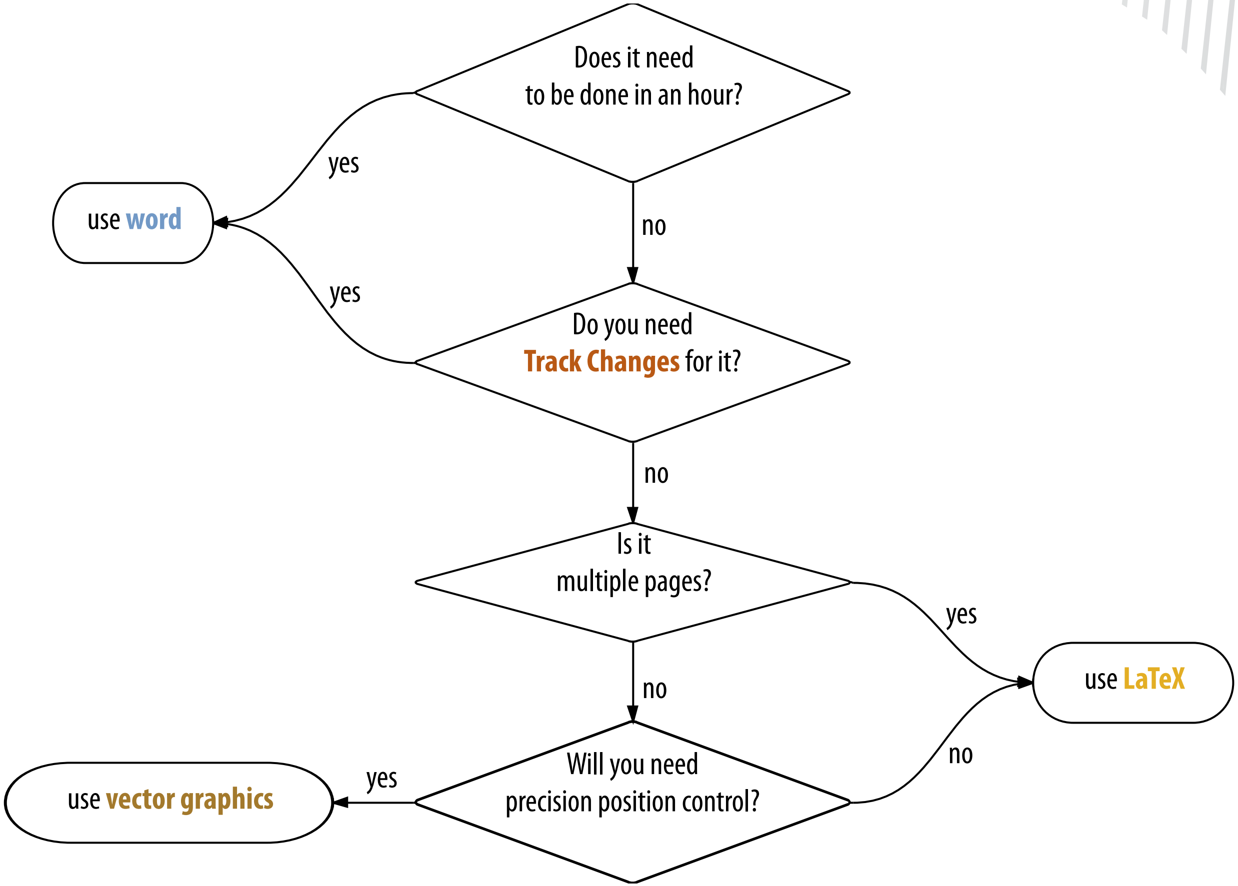


Figure 1: LaTeX Decision Flowchart

LaTeX Demo: Hello World

To Compile

$\Rightarrow$ Create a file `helloworld.tex`

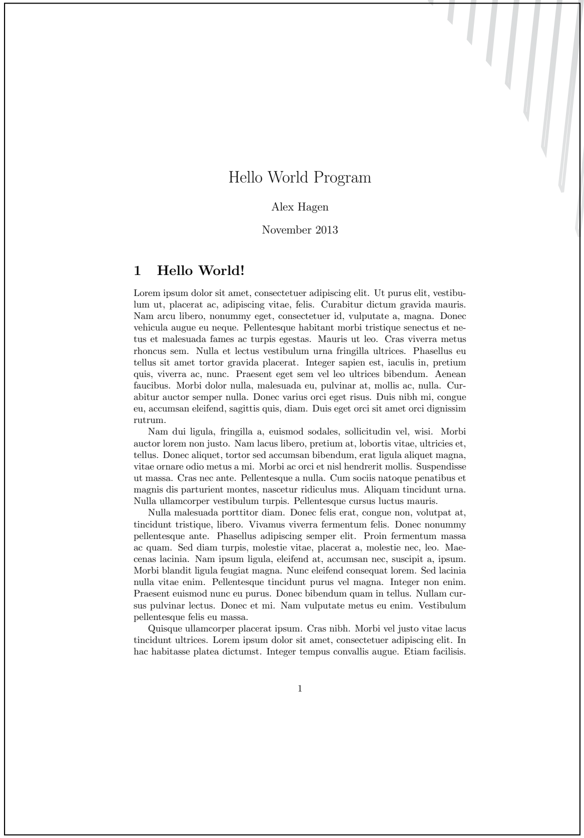
$\Rightarrow$ Open in tex editor (such as Texlive)

$\Rightarrow$ Compile this into a pdf using command or play button

```
$latex -output-format=pdf
  helloworld.tex
```

Table 1: Hello World LaTeX Code `helloworld.tex`

```
\documentclass{article}
\title{Hello World Program}
\author{Alex Hagen}
\date{November 2013}
\usepackage{lipsum}
\begin{document}
  \maketitle
  \section{Hello World!}
  \lipsum[1-5]
\end{document}
```


Hello World Program

Alex Hagen

November 2013

1 Hello World!

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

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

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

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis.

1

Figure 2: Output of LaTeX on `helloworld.tex`

LaTeX Demo: Purdue Thesis

To Compile

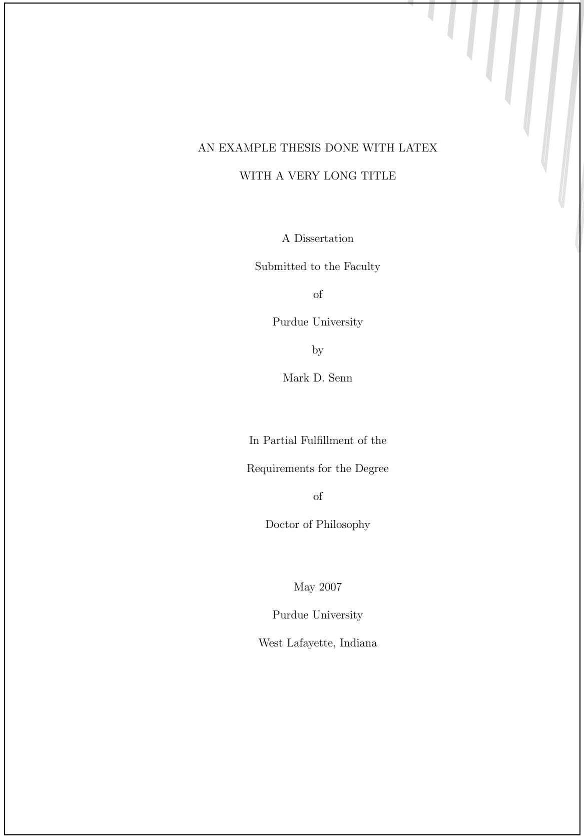
⇒ Download `puthesis.cls`

⇒ Download template files and `pulongtable.sty`

⇒ Put into one directory

⇒ Compile this into a pdf using command or play button

```
$latex -output-format=pdf
  thesis.tex
```


AN EXAMPLE THESIS DONE WITH LATEX

WITH A VERY LONG TITLE

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Mark D. Senn

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

May 2007

Purdue University

West Lafayette, Indiana

Figure 3: Output of Latex on Purdue Thesis Formatted Paper with `puthesis.cls`

LaTeX Demo: Elsevier and Other Journal Templates

To Compile

⇒ Download `elsarticle.cls`

⇒ Follow directions for adding authors, etc.

⇒ Put into one directory

⇒ Compile this into a pdf using command or play button

```
$latex -output-format=pdf
  elspaper.tex
```

Design and Optimization of a Neutron Sensor Utilizing Acoustic Fields - Experimentation and Simulation Using COMSOL and MCNP Physics Platforms

A. Hagen[a], K. Fischer[a], B. Archambault[b], S. Rogers[b], R. Taleyarkhan[a,b]

[a] *Purdue Metastable Fluids and Advanced Research Laboratory, 3601 Sagamore Parkway, Suite H., Lafayette, IN, 47905*
[b] *Sagamore Adams Laboratory, LLC., 3601 Sagamore Parkway, , Lafayette, IN, 47905*

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.

1. Introduction

The improvement of neutron detection is central to health physics, homeland security, and even fission power. The following will discuss Tension Metastable Fluid Detectors (TMFDs) as an alternative to state of the art detectors. It will then discuss a current optimization and design effort using multiphysics simulation as the core. Finally, it will discuss the accuracy the simulation is able to achieve and how that has been used to further the current system.

1.1. Motivation

The National Academy of Engineering has long posed prevention of nuclear terror as a top priority National Academy of Engineering [1]. To prevent nuclear terror, one must monitor the transport and use of special nuclear material (SNM), which is defined by the Atomic Energy Act of 1954 United States Congress [2]. This material gives off telltale neutron signatures, which makes neutron sensors an ideal way to track SNM. With this in mind, the United States has installed gamma and neutron detectors in many border portals, achieving a scanning rate of above 99% Gowadia [3]. It is clear that there is a great demand for neutron detectors in this capacity.

Another field which heavily relies on the use of neutron detectors is that of health physics.

Thus far, currently technology has been able to provide sensors to fulfil the requirements of both the Department of Homeland Security in terror prevention, and health physics in dose monitoring. The state of the art detectors that are used in the aforementioned applications are those of BF3 detectors and He3 detectors. He-3 or BF3 gas in placed under a high electrical bias in an evacuated chamber, and upon interaction in the chamber, ions are created, which are then measured as electrical pulses, indicating detection events Knoll [4]. Unfortunately, these detectors have many practical detriments for the applications proposed. First, these chambers also detect gamma events, and may even be saturated by these events. Second, they are extremely expensive, especially with no He3 being currently produced. Third, the detection systems are extremely complicated and require several expensive electrical components. Finally, in the case of He3 detectors, the NRC has stated that they will not consider any new neutron detection technology using He3 .

The TMFD is a system that can alleviate most, if not all, of the shortcomings of state of the art neutron detectors. In short, the TMFD uses tensile pressures to convert neutron events within a working fluid into macroscopic, visible bubbles. The tensile pressure brings the fluid so close to the spinodal limit that a neutron event will cause violent cavitation. The visibility and audibility of these detection events removes the need for all but a modest electrical drive train. The small probability for interactions of gammas in the working fluid makes the TMFD gamma insensitive up to $10^{11}\,\frac{\gamma}{cm^2\cdot s}$ experimentally (equivalent to $1\,m$ from a spent fuel rack), and $10^{22}\,\frac{\gamma}{cm^2\cdot s}$ theoreti-

Email address: `ahagen@purdue.edu` (A. Hagen)

Preprint submitted to Elsevier *November 22, 2013*

Figure 4: Output of Latex on Elsevier Formatted Paper with `elsarticle.cls`

Typesetting: Floats

- ⇒ Inserting of Figures, Tables, and Algorithms done through **floats**
- ⇒ Creates a caption environment and a separate "box" for the content
- ⇒ In papers/theses, TEX decides where to place floats - options are:
  - → Top of Page
  - → Bottom of Page
  - → Page of Floats
  - → Here Definitely
- ⇒ Are enumerated in the List of Tables, etc.

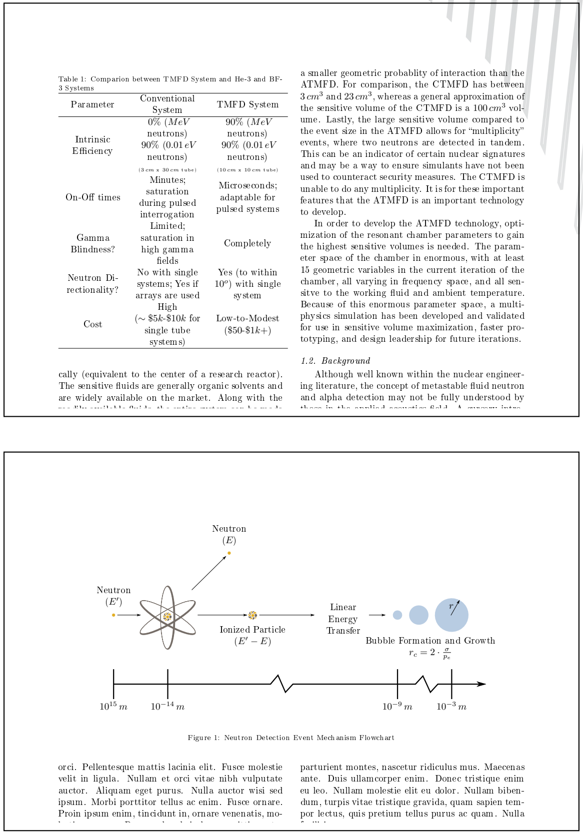


Table 1: Comparion between TMFD System and He-3 and BF-3 Systems

Parameter	Conventional System	TMFD System
Intrinsic Efficiency	0% (MeV neutrons) 90% (0.01 eV neutrons) (3 cm x 30 cm tube)	90% (MeV neutrons) 90% (0.01 eV neutrons) (10 cm x 10 cm tube)
On-Off times	Minutes; saturation during pulsed interrogation	Microseconds; adaptable for pulsed systems
Gamma Blindness?	Limited; saturation in high gamma fields	Completely
Neutron Directionality?	No with single systems; Yes if arrays are used	Yes (to within 10^o) with single system
Cost	High ($\sim$ \$5k-\$10k for single tube systems)	Low-to-Modest (\$50-\$1k+)

cally (equivalent to the center of a research reactor). The sensitive fluids are generally organic solvents and are widely available on the market. Along with the

a smaller geometric probablity of interaction than the ATMFD. For comparison, the CTMFD has between $3\,cm^3$ and $23\,cm^3$, whereas a general approximation of the sensitive volume of the CTMFD is a $100\,cm^3$ volume. Lastly, the large sensitive volume compared to the event size in the ATMFD allows for "multiplicity" events, where two neutrons are detected in tandem. This can be an indicator of certain nuclear signatures and may be a way to ensure simulants have not been used to counteract security measures. The CTMFD is unable to do any multiplicity. It is for these important features that the ATMFD is an important technology to develop.

In order to develop the ATMFD technology, optimization of the resonant chamber parameters to gain the highest sensitive volumes is needed. The parameter space of the chamber in enormous, with at least 15 geometric variables in the current iteration of the chamber, all varying in frequency space, and all sensitve to the working fluid and ambient temperature. Because of this enormous parameter space, a multiphysics simulation has been developed and validated for use in sensitive volume maximization, faster prototyping, and design leadership for future iterations.

1.2. Background

Although well known within the nuclear engineering literature, the concept of metastable fluid neutron and alpha detection may not be fully understood by

Figure 1: Neutron Detection Event Mechanism Flowchart

orci. Pellentesque mattis lacinia elit. Fusce molestie velit in ligula. Nullam et orci vitae nibh vulputate auctor. Aliquam eget purus. Nulla auctor wisi sed ipsum. Morbi porttitor tellus ac enim. Fusce ornare. Proin ipsum enim, tincidunt in, ornare venenatis, mo-

parturient montes, nascetur ridiculus mus. Maecenas ante. Duis ullamcorper enim. Donec tristique enim eu leo. Nullam molestie elit eu dolor. Nullam bibendum, turpis vitae tristique gravida, quam sapien tempor lectus, quis pretium tellus purus ac quam. Nulla

Figure 5: Placement of Several Different Floats

Typesetting: Mathematics

Math Features

- ⇒ Fractions
- ⇒ Symbols
- ⇒ Arrays
- ⇒ Cases
- ⇒ Numbering
- ⇒ Integration and sums
- ⇒ Limits
- ⇒ Super- and sub- scripts

Table 2: Latex Code to Generate Helmholtz and Slowing Down Equations

```
The \textbf{Scalar-Helmholtz Equation} is given in
  Equation \ref{eq:Scalar-Helmholtz-Equation}:
$D\nabla^{2}\phi-\Sigma_{a}\phi+s=0$
  \label{eq:Scalar-Helmholtz-Equation}
The \textbf{Neutron Slowing Down} equation is given
  in Equation \ref{eq:Neutron-Slowing-Down-Equation}:
$\left[D(E)B^{2}+\Sigma_{t}(E)\right]
  \varphi(E)-\int\Sigma_{s}(E'\rightarrow E)
  \varphi(E')dE' =\lambda\chi(E)\int\nu\Sigma_{f}(E')
  \varphi(E')dE'$
  \label{eq:Neutron-Slowing-Down-Equation}
```

The **Scalar-Helmholtz Equation** is given in Equation 1:

$$D\nabla^{2}\phi-\Sigma_{a}\phi+s=0 \tag{1}$$

The **Neutron Slowing Down** equation is given in Equation 2:

$$\left[D(E)B^{2}+\Sigma_{t}(E)\right]\varphi(E)-\int\Sigma_{s}(E'\rightarrow E)\varphi(E')dE' \\ =\lambda\chi(E)\int\nu\Sigma_{f}(E')\varphi(E')dE' \tag{2}$$

Figure 6: Mathematics Output

Document Organization: Sectioning

Table 3: Latex Code to Generate Sectioning

```
\tableofcontents{}
\section{Section}
First Line of the Section
\subsection{Subsection}
First Line of the Subsection
\paragraph{Paragraph}
First Line of the Paragraph
\subparagraph{Subparagraph}
First Line of the Subparagraph
```

Contents

1 Section

First Line of the Section

1.1 Subsection

First Line of the Subsection

Paragraph First Line of the Paragraph

Subparagraph First Line of the Subparagraph

Figure 7: Sectioning Examples

Sectioning Commands

⇒ Can section each document to help with organization

⇒ Outline Type

→ Part

→ Chapter

→ Section

→ Subsection

→ Subsubsection

→ ...

→ Paragraph

→ Subparagraph

⇒ Automatically included in Table of Contents

Document Organization: Bibliographies

Bibliography Features

- ⇒ Use "bibtex" files
- ⇒ Can be updated through text files
- ⇒ Mendeley supports bibtex output
- ⇒ Only lists cited articles
- ⇒ Can add page numbers to citations

Table 4: Latex Code to Cite and Generate a Bibliography

```
Some text here with a citation \cite{Knoll2000}, a
  multiple citation \cite{Engineering2012,Fried...
  , and a page after citation \cite[p. 1]{UnitedStates...
\bibliographystyle{plain}
\bibliography{Masters_Thesis}
```

Table 5: Contents of BiBTeX Bibliography

```
@book{Knoll2000, author = {Knoll, G. F.},...
...
```

Some text here with a citation [3], a multiple citation [4, 1, 2], and a page after citation [5, p. 1].

References

[1] Friedrich & Dimmock Inc. Simax Glass Properties. Technical report, Friedrich & Dimmock, Inc., Millville, NJ, 2013.

[2] H. Gowadia. Preventing Nuclear Terrorism: Does DHS have an Effective and Efficient Nuclear Detection Strategy, 2012.

[3] G. F. Knoll. *Radiation Detection and Measurement.* John Wiley & Sons, Inc., Ann Arbor, MI, third edition, 2000.

[4] National Academy of Engineering. Grand Challenges - Engineering Challenges, 2012.

[5] United States Congress. Atomic Energy Act of 1954, 1954.

Figure 8: Bibliography Example

Graphical Interface: LyX High Level LaTeX Interface

LyX Editing and Features

- ⇒ Easy visible math editing
- ⇒ Table creation
- ⇒ Float insertion
- ⇒ Font support
- ⇒ Page Size and Layout Support
- ⇒ Some figure support
- ⇒ Labeling and Cross Referencing
- ⇒ Bibliographies
- ⇒ Support for any and all TeX Code

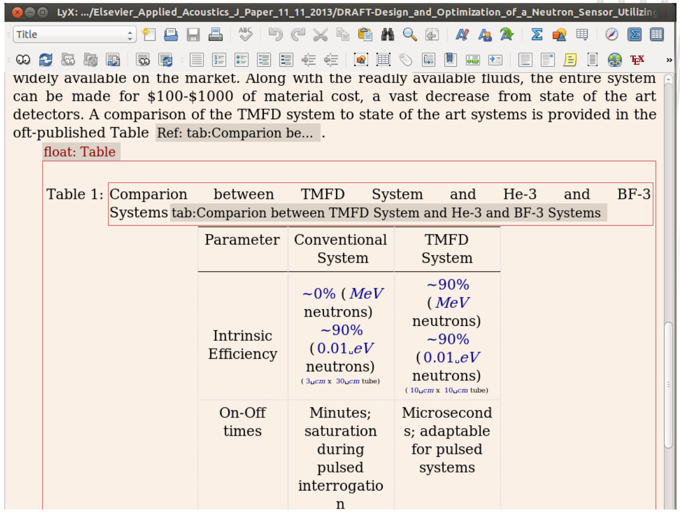


Figure 9: LyX Editing Environment

Advanced Figures: PDF_TEX Figures

Workflow

⇒ Vector Graphic Editing in Inkscape

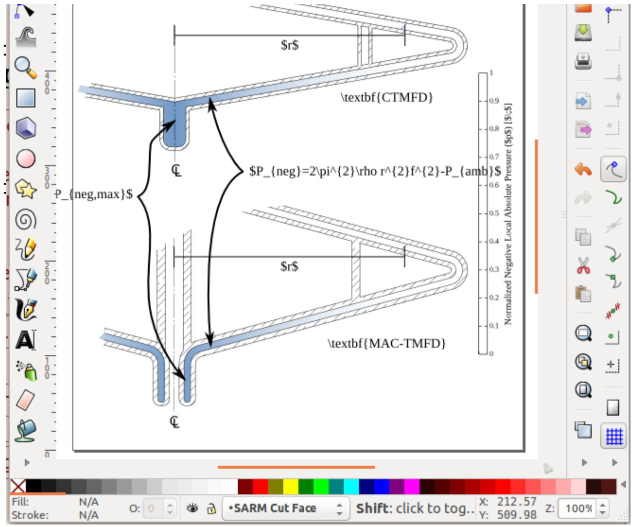


Figure 10: pdf_tex Editing in Inkscape

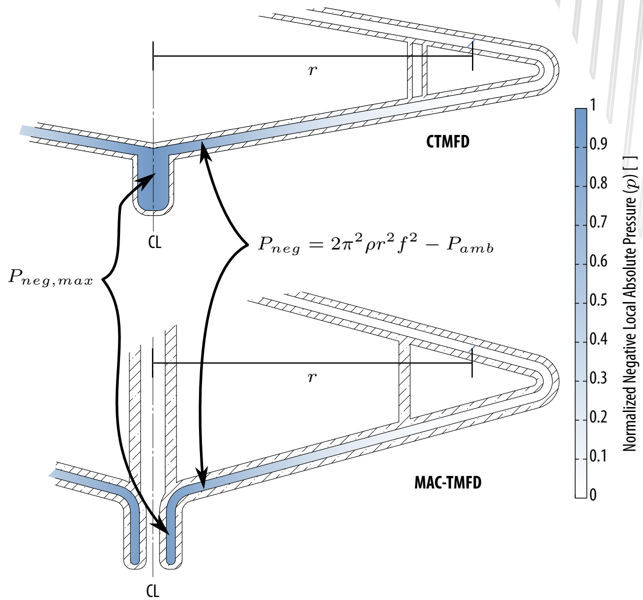


Figure 11: pdf_tex Figure Output

Advanced Figures: PGF Figures

Workflow

- ⇒ Data analysis and calculation using Python
- ⇒ Plotting using inclusion of TeX tags
- ⇒ savefig using `.pgf` extension and backend
- ⇒ inclusion in latex document with `\input` tag

Table 6: Source code for sin wave plotting with python to pgf

```
import matplotlib as mpl
mpl.use('pgf')
import matplotlib.pyplot as plt
import numpy as np
x = np.linspace(0, 6.5)
y1 = np.sin(x)
y2 = np.cos(x)
plt.plot(x, y1, 'r-', x, y2, 'b--')
plt.xlabel('x-coordinate ($x$) [$cm$]')
plt.ylabel('y-coordinate ($y$) [$cm$]')
plt.savefig('img/figure.pgf')
```

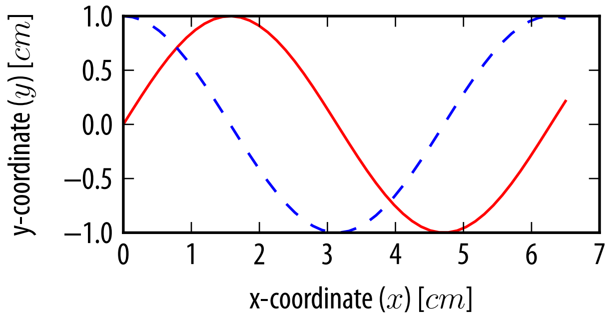


Figure 12: Example of pgf figure included in LaTeX

Advanced Document Tracking: Git Document Revision

Features

- ⇒ Can use content revision management software
- ⇒ ASCII files give meaning to diff
- ⇒ Decreases Space
- ⇒ .doc files will be copied every time they are committed
- ⇒ First add file, then commit with some comment
- ⇒ Put every line (even if not a new paragraph) on a new line
- ⇒ Will allow for rolling back of the content

```
$ git commit -am 'added a line \
  to the file'
$ git show
commit d...
Author: Alex Hagen <ahagen@pur...
Date:   Sun Nov 24 22:58:10 20...
    added a line to the file
diff --git a/helloworld.tex b/...
index ec6251c..7c91797 100644
--- a/helloworld.tex
+++ b/helloworld.tex
@@ -6,5 +6,6 @@
 \begin{document}
   \maketitle
   \section{Hello World!}
+   I wrote this about git!
   \lipsum[1-5]
 \end{document}
```