Technical Documentation

Depth Estimation: Technical Documentation

Aaron Rhim[a,1], **Nathaniel Hawron**[b,2], **Zachrey Zhu**[c,3], **Marc Bondoger**[d,4], **Leo Kaiya**[e,5] **and Axel Bendl**[f,6]

[a, b, c, d, e, f] *University of British Columbia*

Dr. Warren Hare

Abstract—Welcome to tau (τ) LaTeX class designed especially for your lab reports or academic articles. In this example template, we will guide you through the process of using and customizing this document to your needs. For more information of this class check out the appendix section. There, you will find codes that define key aspects of the template, allowing you to explore and modify them.

Keywords—*LaTeX research report, academic article, tau class*

1. Introduction

We must research what has been done in terms of the AI side of the research. This involves researching in different architectures that have been created and used and how we can incorporate any state-of-the-art models etc. Don't forget to document any sources that you've found at the bottom of this document. We will eventually go through each source and note any that we used and get rid of any that we didn't use. Feel free to use ChatGPT's Deep Research option. Will replace this paragraph with a true introduction at the end.

2. Title

The `\maketitle` command generates the title and author information section, including the professor name and affiliations. The title can be modified in tau-class/tau.cls/title style section.

By default, *tau class* shows the title on the left. However, you can change `\raggedright` to `\centering` in `\titlepos` to move the title to the center or, modify it to your own preferences.

In addition to the `\title` command, a custom command named `\journalname` has been added to include more information.

If you do not need this command, you can undefined it and the content will be adjusted automatically.

3. Abstract

The abstract and keywords are defined using the `\keywords` and `\begin{abstract}␣\end{abstract}` commands respectively. For the abstract to appear, make sure the `\tauabstract` command is always included after the beginning of the document.

If the keywords are not declared in the preamble, the content will be adjusted automatically.

4. Document style options

4.1. Tau start

We included the `\taustart{}` command, which provides a personalized lettrine for the beginning of a paragraph.

4.2. Line numbering

By implementing the *lineno* package, the line numbering of the document can be placed with the command `\linenumbers`.

I recommend placing the command after the abstract and table of contents for a better appearance.

4.3. Table of contents

The *tau class* provides a customized design for the table of contents. Each level of the ToC provides a preview of the content and its location in the document.

5. Tables and figures

5.1. Tables

Table 1 shows an example table. The `\tabletext{}` is used to add notes to tables easily.

Table 1. Astronomical Object Data

Object	Distance (Light Years)
Alpha Centauri	4.37
Betelgeuse	642.5
Andromeda Galaxy	2.537 million

Note: The table contains data of some famous celestial objects.

5.2. Figures

Fig. 1 shows an example figure.

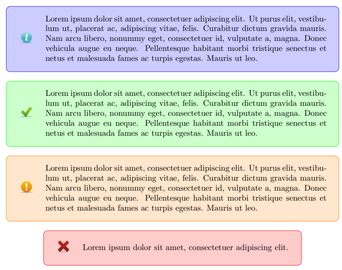


Figure 1. Example figure obtained from PGFPlots [1].

Fig. 2 shows an example of two figures that covers the width of the page. It can be placed at the top or bottom of the page. The space between the figures can also be changed using the `\hspace{Xpt}` command.

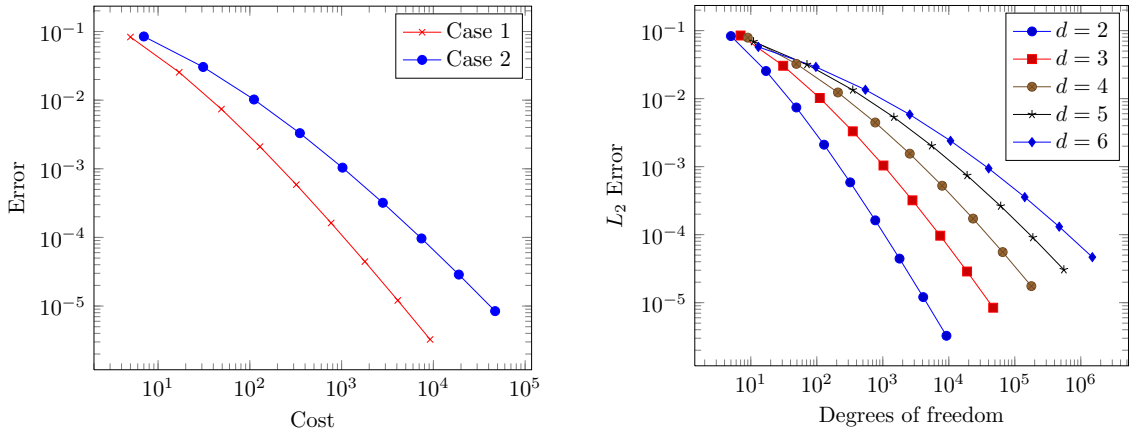
6. Tau packages

6.1. Tauenvs

This template has its own environment package *tauenvs.sty* designed to enhance the presentation of the document. Among these custom environments are *tauenv*, *info* and *note*.

There are two environments which have a predefined title. These can be included by the command `\begin{note}` and `\begin{info}`. All the environments have the same style.

An example using the tau environment is shown below.



(a) Example left figure. **(b)** Example right figure.

Figure 2. Example figure that covers the width of the page obtained from PGFPlots [1].

Environment with custom title

This is an example of the custom title environment. To add a title type `[frametitle=Your title]` next to the beginning of the environment (as shown in this example).

Tauenv is the only environment that you can customize its title. On the other hand, info and note adapt their title to Spanish automatically when this language package is defined.

6.2. Taubabel

In previous versions, we included a package called *taubabel*, which have all the commands that automatically translate from English to Spanish when this language package is defined.

By default, tau displays its content in English. However, at the beginning of the document you will find a recommendation when writing in Spanish.

Note: You may modify this package if you want to use other language than English or Spanish. This will make easier to translate the document without having to modify the class document.

7. Equation

Equation 1, shows the Schrödinger equation as an example.

$$\frac{\hbar^2}{2m}\nabla^2\Psi + V(\mathbf{r})\Psi = -i\hbar\frac{\partial\Psi}{\partial t} \tag{1}$$

The *amssymb* package was not necessary to include, because stix2 font incorporates mathematical symbols for writing quality equations. In case you choose another font, uncomment this package in tau-class/tau.cls/math packages.

If you want to change the values that adjust the spacing above and below the equations, play with `\setlength{\eqskip}{8pt}` value until the preferred spacing is set.

8. Adding codes

This class[1] includes the *listings* package, which offers customized features for adding codes in LaTeX documents specifically for C, C++, LaTeX and Matlab.

You can customize the format in tau-class/tau.cls/listings style.

```
function fibonacci_sequence(num_terms)
    % Initialize the first two terms of the sequence
    fib_sequence = [0, 1];

    if num_terms < 1
        disp('Number of terms should be greater than or equal to 1.');
        return;
    elseif num_terms == 1
        fprintf('Fibonacci Sequence:\n%d\n', fib_sequence(1));
        return;
    elseif num_terms == 2
        fprintf('Fibonacci Sequence:\n%d\n%d\n', fib_sequence(1), fib_sequence(2));
        return;
    end

    % Calculate and display the Fibonacci sequence
    for i = 3:num_terms
        fib_sequence(i) = fib_sequence(i-1) + fib_sequence(i-2);
    end

    fprintf('Fibonacci Sequence:\n');
    disp(fib_sequence);
end
```

Code 1. Example of Matlab code.

If line numbering is defined at the beginning of the document, I recommend placing the command `\nolinenumbers` at the start and `\linenumbers` at the end of the code.

This will temporarily remove line numbering and the code will look better.

9. References

The default formatting for references follows the IEEE style. You can modify the style of your references. See appendix for more information.

References

[1] PFGPlots, *A latex package to create plots.* [Online]. Available: https://pgfplots.sourceforge.net/.

[1] Hello there! I am a footnote :)