



School of Electrical and Information Engineering  
The University of Witwatersrand, Johannesburg  
ELEN0000 - Course Name

---

# Learning L<sup>A</sup>T<sub>E</sub>X

An Informal Introduction

*Kerren Ortlepp*  
123456

## *Abstract*

L<sup>A</sup>T<sub>E</sub>X is a typesetting software that can be used to write reports, create presentations, and much more! The creation of a L<sup>A</sup>T<sub>E</sub>X report is demonstrated throughout this document. Furthermore, there is a tutorial on figure and circuit creation using *Tikz*. By the end of this document you should know the basics of creating a L<sup>A</sup>T<sub>E</sub>X document and with a little bit of practice you'll be creating professional looking reports in no time!

May 8, 2016

# Learning L<sup>A</sup>T<sub>E</sub>X

Kerren Ortlepp

## Abstract

L<sup>A</sup>T<sub>E</sub>X is a typesetting software that can be used to write reports, create presentations, and much more! The creation of a L<sup>A</sup>T<sub>E</sub>X report is demonstrated throughout this document. Furthermore, there is a tutorial on figure and circuit creation using *Tikz*. By the end of this document you should know the basics of creating a L<sup>A</sup>T<sub>E</sub>X document and with a little bit of practice you'll be creating professional looking reports in no time!

You may have noticed that the title has been printed twice. This is just to show that L<sup>A</sup>T<sub>E</sub>X supports title pages and putting the title on the main document.

## 1 Software Installation

Let's start off by installing the latest version of TeXLive which is a L<sup>A</sup>T<sub>E</sub>X distribution for Windows, Linux, and Mac. Visit <http://mirror.ufs.ac.za/ctan/systems/texlive/Images/texlive2015-20150523.iso> and download the ISO file (2.7GB). You can check out this YouTube video I made on how to install it on a Windows PC, [https://youtu.be/Gmw-eKh\\_hYs](https://youtu.be/Gmw-eKh_hYs) **kerren'youtube**

Once you've installed TeXLive the next thing to do is to get an editing environment, there are many to choose from, for example:

1. [TeXstudio](#)
2. [Texmaker](#)
3. [Texworks](#)
4. [TeXnicCenter](#)
5. [Sublime Text with LatexTools](#)
6. [etc](#)

We also need a reference handler, you can make your reference lists yourself but it's much easier to use pre-existing software! The handlers I'd recommend would be:

1. [JabRef](#)
2. [Mendeley](#)
3. [Zotero](#)

For the sake of simplicity it would be good to start off with [TeXLive](#), [TeXstudio](#) and [JabRef](#)!

For this course we're going to avoid the installation process and use an online environment. If

you haven't already done so, visit <https://www.overleaf.com/> and register for an account. We'll be using this throughout the course for building L<sup>A</sup>T<sub>E</sub>X documents. We'll also use an online reference editor, visit <http://truben.no/latex/bibtex/>.

## 2 Introduction

We'll start off with creating a document. The first line of your document determines the font size, document type and whether or not the document is double column.

```
1 \documentclass[11pt, twocolumn]{article}
```

From this you should see that the format of a L<sup>A</sup>T<sub>E</sub>X environment is:

```
1 \environment[options]{code}
```

After you've created your document you need to fill in your preamble. This tells L<sup>A</sup>T<sub>E</sub>X what packages you want to use and allows you to define environments, commands, titles, authors, etc. The way to instruct L<sup>A</sup>T<sub>E</sub>X to use a package is to use the command:

```
1 \usepackage[options]{package_name}
```

Some examples would include:

```
1 \usepackage{fullpage}
2 \usepackage{graphicx}
3 \usepackage{amsmath}
4 \usepackage{hyphens}{url}
5 \usepackage{tikz}
6 \usepackage{siunitx}{circuitikz}
7 \usepackage{hidelinks}{hyperref}
```

## 3 Starting Tips and Tricks

There are a few useful tips and tricks that you would probably find out after using L<sup>A</sup>T<sub>E</sub>X for a

while. I'll add the really useful ones here!

### 3.1 Commenting

You can comment in your document using the % sign. That means that when you actually want a % sign in your document you need to use a backslash before the symbol:

```
1 \%
```

### 3.2 No Indentation

Each time you start a new paragraph  $\LaTeX$  automatically indents the line.

Like this! But you might not always want the line to be indented. You can either disable the indentation completely or when you start a new paragraph you can add:

```
1 \noindent
```

### 3.3 Double Inverted Commas

Adding inverted commas is NOT as simple as using the " character. You need to enclose the information you would like to quote using the ' and ' characters:

```
1 'My Quote'
```

will then give "My Quote".

### 3.4 Math Mode

Math mode is the environment you use to input mathematical equations in your document. An example of using the equation environment is given below:

```
1 %
2 \begin{equation}
3 f(t) = \alpha e^{\beta t}
4 \end{equation}
5 %
```

Notice the % sign before and after the equation? That's to prevent additional spacing, don't forget to put that in otherwise there will be a lot of whitespace between your text and your equations.

#### 3.4.1 Superscript and Subscript

To add a superscript or subscript in math mode, we use the ^ and \_ respectively. For instance the variables:

$$V_i$$
$$n^2$$
$$x_n^3$$

Would look like this in the  $\LaTeX$  equation environment:

```
1 V_i
2 n^2
3 x_n^3
```

#### 3.4.2 Symbols

It'll help to have a cheat sheet of the mathematical symbols that you will most likely use in your  $\LaTeX$  reports. Here are examples of some of the most common symbols:

$$\alpha$$
$$\backslash alpha$$

$$\beta$$
$$\backslash beta$$

$$\gamma$$
$$\backslash gamma$$

$$\omega$$
$$\backslash omega$$

$$\zeta$$
$$\backslash zeta$$

$$\pi$$
$$\backslash pi$$

$$\sum$$
$$\backslash sum$$

$$\sum_{i=0}^N$$
$$\backslash sum\limits_{i=0}^N$$

$$\frac{1}{x}$$
$$\backslash frac{1}{x}$$

$$\left(\frac{1}{x}\right)^2$$
$$\left(\backslash frac{1}{x}\right)^2$$

$$\left(\frac{1}{x}\right)^2$$
$$\left(\backslash frac{1}{x}\right)^2$$

$$\sin()$$
$$\backslash sin ()$$

$$\cos()$$
$$\backslash cos ()$$

$$\tan()$$
$$\backslash tan ()$$

>	>	You use the \$ \$ signs to make it in-line, for example:
<	<	<code>1 \$\sin (2 \pi f t)\$</code>

$\geq$	<code>\geq</code>	<b>3.6 No Tags for Math Mode</b>
$\leq$	<code>\leq</code>	When you create an equation it often gives a tag (an equation number) for example:

$$f(t) = t^2 e^{2t} \tag{2}$$

$\approx$	<code>\approx</code>	If you don't want the "(2)" after the equation you use the following:
-----------	----------------------	---

$\pm$	<code>\pm</code>	<pre>1 \begin{equation} 2   f(t) = t^2 e^{2t} \notag 3 \end{equation}</pre>
-------	------------------	---

$\int$	<code>\int</code>	And you'll now get:
--------	-------------------	---------------------

$$f(t) = t^2 e^{2t}$$

$\int_{t=0}^{t=1} dt$	<code>\int^{t=1}_{t=0} dt</code>	An alternative is to use the shorthand equation environment:
-----------------------	----------------------------------	--

```
1 \[
2   f(t) = t^2 e^{2t}
3 \]
```

This environment doesn't put tags automatically:

$\sqrt{x}$	<code>\sqrt{x}</code>	$f(t) = t^2 e^{2t}$
------------	-----------------------	---------------------

$\max(a, b)$	<code>\text{max}(a,b)</code>	<b>3.7 Matrices</b>
--------------	------------------------------	---------------------

Matrices are relatively straightforward to use in L<sup>A</sup>T<sub>E</sub>X using the math mode. The only package that you need for this is *amsmath*. Within the equation you can change the matrix type by changing the environment:

$A \cdot B$	<code>A \cdot B</code>	$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ <code>pmatrix</code>
$n = 1, \dots, 5$	<code>n = 1, \dots, 5</code>	

$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$	<code>A = \left(\begin{array}{cc} a &amp; b \\ c &amp; d \end{array}\right)</code>	$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ <code>bmatrix</code>
--	--	---

$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$	<code>a &amp; b \\ c &amp; d</code>	$\begin{vmatrix} a & b \\ c & d \end{vmatrix}$ <code>vmatrix</code>
--	-------------------------------------	---

<code>\end{array}\right)</code>	The code for the pmatrix would look as follows:
---------------------------------	---

```
1 \begin{equation}
2   \begin{pmatrix}
3     a & b \\
4     c & d
5   \end{pmatrix}
6 \end{equation}
```

To create the bmatrix or vmatrix you'd just replace pmatrix with the different environment name.

### 3.5 In-line Math Mode

Sometimes you may not want the full equation environment, for example, you might prefer  $\sin(2\pi ft)$  instead of:

$$\sin(2\pi ft) \tag{1}$$

### 3.8 SI Units

The “siunits” package is very useful to write out units after a number with the correct spacing. Examples are as follows:

10 mV  $\backslash\text{SI}\{10\}\{\backslash\text{milli}\backslash\text{volt}\}$

60  $\mu$ A  $\backslash\text{SI}\{60\}\{\backslash\text{micro}\backslash\text{ampere}\}$

1 V<sup>2</sup> m<sup>-1</sup>  $\backslash\text{SI}\{1\}\{\backslash\text{square}\backslash\text{volt}\backslash\text{per}\backslash\text{metre}\}$

1 J C<sup>-1</sup>  $\backslash\text{SI}\{1\}\{\backslash\text{joule}\backslash\text{per}\backslash\text{coulomb}\}$

kg m s<sup>-1</sup>  $\backslash\text{SI}\{\}\{\backslash\text{kilogram}\backslash\text{metre}\backslash\text{per}\backslash\text{second}\}$

Check out <ftp://ftp.dante.de/tex-archive/macros/latex/exptl/siunitx/siunitx.pdf> for a list of all of the units it supports.

### 3.9 Tables

Inserting tables can get a little bit tricky in L<sup>A</sup>T<sub>E</sub>X! However, there are tools that greatly assist in creating the code for these tables. Take a look at <http://www.tablesgenerator.com/> which provides a great interface to creating tables with formatting. Table 1 demonstrates a table in L<sup>A</sup>T<sub>E</sub>X. An example of creating a table is presented below. It may be complicated now but after using the generator a few times you’ll be able to create tables directly in your document!

```
1 \begin{table}[h!]
2 \caption{Table caption}
3 \label{tab:table_label}
4 \begin{tabular}{|c|r|l|}
5 \hline
6 First Word & Number & Last Word\\ \hline
7 Hello & 1.00 & World\\ \hline
8 Hello & 2.00 & Worlds\\ \hline
9 \end{tabular}
10 \end{table}
```

Table 1: My Measurements.

Voltage (V)	Current (mA)	Resistance ( $\Omega$ )
1.0	20.2	49.5
2.0	40.1	49.9
3.0	60.0	50.0
4.0	79.2	50.5
5.0	99.6	50.2

You may be wondering what the options are after the `\begintable` command is called. These are

placement options and each character has a specific meaning:

**h** Put the table here.

**t** Put the table at the top of the column.

**b** Put the table at the bottom of the column.

**!** Do your VERY BEST to place it where I’ve specified.

### 3.10 Booktabs

Booktabs is an additional package that allows you to make your tables look even better! If you ever see the `\toprule`, `\cmidrule`, `\midrule`, or `\bottomrule`, on a tutorial the maker of the tutorial is using Booktabs. Check out Table 2, it looks pretty cool right?

Table 2: Circuit Price Table using Booktabs.

Item			
Name	Cost (R)	Quantity	Total (R)
LM555	5.50	2	11.00
1k $\Omega$ Res	0.30	5	1.50
10nF Cap	0.20	1	0.20
White LED	4.20	2	8.40
<b>Total</b>			21.10

### 3.11 Wide Tables

It is often very tough to fit a table into a single column. A lot of the time, the table would fit if we could just stretch it over both columns. Use the “\*” in the table environment to make it split over two columns. The environment now looks like this:

```
1 \begin{table*}
2 ...
3 \end{table*}
```

Check out Table 3 to see how it stretches across both columns.

### 3.12 Inserting Figures

To insert a figure into L<sup>A</sup>T<sub>E</sub>X you use the figure environment:

```
1 \begin{figure}[h!]
2 \begin{center}
3 \includegraphics[width=\columnwidth]{image.png}
4 \end{center}
5 \caption{Caption here}
6 \label{fig:myFigure}
7 \end{figure}
```

Table 3: My Wide Measurements.

Voltage (V)	Current (mA)	Resistance ( $\Omega$ )	Confidence (%)	Temperature ( $^{\circ}\text{C}$ )
1.0	20.2	49.5	95.3	25.3
2.0	40.1	49.9	95.8	25.4
3.0	60.0	50.0	96.1	25.3
4.0	79.2	50.5	95.5	25.3
5.0	99.6	50.2	95.6	25.4

The options for the figure environment, [h!], mean that you would like the image to be placed as close to where you've created this code as possible. The `includegraphics` command allows you to set the width of your image and specify the image name. L<sup>A</sup>T<sub>E</sub>X supports png, jpg, pdf and many more!

### 3.13 Automatically Resizing an Image

L<sup>A</sup>T<sub>E</sub>X is able to resize an image to fit in the column. To do this you use the following option in the `includegraphics` command:

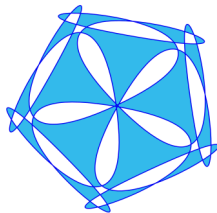
```
1 \includegraphics[width=\linewidth]{
  imagename.png}
```

You can also scale it to fit a percentage of your column width, for instance, if you want 80% you'd use:

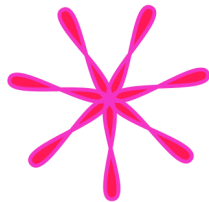
```
1 \includegraphics[width=0.8\linewidth]{
  imagename.png}
```

### 3.14 Subfigures

Subfigures can be very useful when images are closely related. Figure 1 below demonstrates how I have two figures (Figure 1a and Figure 1b) and I can reference them as a group or each figure individually.



(a) This is my first image.



(b) This is my second image.

Figure 1: Two of my finest works of art.

The code for a subfigure is very similar to a normal figure:

```
1 \begin{figure}[h!]
2   \centering
3   \begin{subfigure}[b]{0.4\columnwidth}
4     \centering
5     \includegraphics[width=\
      linewidth]{abstract1}
6     \caption{This is my first image
      .}
7     \label{fig:abstract1}
8   \end{subfigure}
9   \quad\quad
10  \begin{subfigure}[b]{0.4\columnwidth}
11    \centering
12    \includegraphics[width=\
      linewidth]{abstract2}
13    \caption{This is my second image
      .}
14    \label{fig:abstract2}
15  \end{subfigure}
16  \caption{Two of my finest works of
      art .}
17  \label{fig:abstract_art_subfigure}
18 \end{figure}
```

### 3.15 Labelling and Referencing Figures

Figures, tables and sections can be referenced at any point in your report as long as you assign it a label. Labels are assigned within the environment, for figures:

```
1 \begin{figure}[h]
2   % Your figure
3   \label{fig:myFigure}
4 \end{figure}
```

For tables:

```
1 \begin{table}[h]
2 \label{tab:myTable}
3   % Your table
4 \end{table}
```

And for sections:

```
1 \section{mySection}
2 \label{sec:mySection}
```

Now when you'd like to reference these you'd use the `ref` command:

```
1 As shown in Figure~\ref{fig:myFigure}.
2 As shown in Table~\ref{tab:myTable}.
3 Explained in Section~\ref{sec:mySection}
```

### 3.16 Using Cleveref

Cleveref automatically determines the type of object you're referencing. Instead of using the `\ref` command you'd use the `\cref` command. Now it will not only fill in the figure number but it'll also fill in the type such as table, figure, section, or equation. If you'd like the type to be capitalized (Table vs table) you use `\Cref`.

The following references to figures are using Cleveref: Table 2, Figure 1, Section 3.16, Equation (2), Figure 1a. Check the difference between these two references: Table 2 vs table 2.

```
1 \Cref{tab:measurements_booktabs} vs \
   cref{tab:measurements_booktabs}.
```

### 3.17 Non-Breaking Space

In the previous section you may have noticed the `~` sign before the `\ref`. The `~` character ensures that the text before and after it remains on the same line. You'll need to use this OFTEN! Especially when it comes to naming figures, tables, sections, and authors. For instance, I'll use it when I write Kerren Ortlepp to ensure that "Kerren" and "Ortlepp" split over two lines.

### 3.18 Footnotes

If you'd like to add a footnote to your document you use the following code:

```
1 \footnote{...}
```

You'll now see a small number where you've placed your footnote and the text will be at the bottom of the page<sup>1</sup>.

### 3.19 Creating Lists - Numbered or Bullets

To create a list we either use an enumeration or itemize environment. The lists below are generated using the following code:

```
1 \begin{enumerate}
2   \item Point 1.
3   \item Point 2.
4   \item Point 3.
5 \end{enumerate}
6 \begin{itemize}
7   \itemsep0em
8   \item Point 1.
9   \item Point 2.
10  \item Point 3.
11 \end{itemize}
```

<sup>1</sup>Don't forget to add a full stop to the footnote, it should be a full sentence.

1. Point 1.
2. Point 2.
3. Point 3.

- Point 1.
- Point 2.
- Point 3.

The "`\itemsep0em`" removes the additional spacing between each point in the list.

### 3.20 Single Column Abstract on a Double Column Document

In some cases you may want your abstract to go over both columns instead of being in the double column format. To do this you use the following code:

```
1 \begin{document}
2 \twocolumn[
3 \begin{@twocolumnfalse}
4   \maketitle
5   \vspace{-1cm} % Reduces the space
                   between the title and the
                   abstract.
6   \begin{abstract}
7     \noindent Write your abstract
                   here!
8   \end{abstract}
9 \end{@twocolumnfalse}]
```

### 3.21 Page Size and Margins

The margins and page size can be modified using the "geometry" package. The code is added to the preamble<sup>2</sup> and would look something like this:

```
1 \geometry{
2   a4paper ,
3   total={170mm,250mm} ,
4   left=25mm,
5   right=25mm,
6   top=22mm,
7   bottom=35mm,
8   columnsep=8mm,
9   textwidth=172mm
10 }
```

Please refer to the Blue Book **bluebook** to find the margins and font sizes that should be used in your reports.

### 3.22 Verbatim Mode and Listings

Verbatim mode writes out everything you type to screen. It doesn't see any characters as "special". For instance, if you're in verbatim mode and you

<sup>2</sup>The preamble is the code that exists before the `\begin{document}`.

type the `$` character, it'll output the character as opposed to going into math mode.

In most cases you'd need verbatim mode for writing out code or some form of syntax that uses a lot of special characters. However, there's a better package to use for that known as "listings". This is the same package that has been used to show the  $\text{\LaTeX}$  code in previous examples. Verbatim mode would look as follows:

```
This is verbatim text      it doesn't
ignore spaces or special characters
like:
!@#$ %^&*()-_+'`\"|/<>
Often we need these characters when
showing some form of syntax. We use
listings for that instead of verbatim
mode because it offers a lot more
functionality!
```

### 3.23 Vertical and Horizontal Spacing

Sometimes you need to create some whitespace between objects and text. You can use the `\vspace` and `\hspace` commands which allow you to add vertical and horizontal space respectively. Take a look at the code below and what it produces:

```
1 \noindent This sentence has a 2cm gap
   between it and the next sentence.
2
3 \vspace{2cm}
4
5 \noindent The next sentence.
```

This sentence has a 2cm gap between it and the next sentence.

The next sentence.

It's important to note the empty line before and after the `\vspace` command. This has to be there otherwise it doesn't add the vertical space!

### 3.24 Section Numbering

Sometimes we don't want to number a section heading, for instance, if we want an acknowledgement section. In that case we use the `"*` charac-

ter when creating the environment and it doesn't give the section a number:

```
1 \section*{Acknowledgements}
```

Would give the following:

## Acknowledgements

This section is created to look the same as others but it doesn't add to the section count!

## 4 Bibliographies

Referencing in  $\text{\LaTeX}$  is extremely easy! Using software that creates a Bibtex file (such as JabRef) you can easily create a reference list. To insert the references, at the end of the document you add:

```
1 \bibliographystyle{unsrt}
2 \bibliography{myReferences.bib}
```

Now whenever you need to cite someone (or a number of people) in your text you just use:

```
1 Blah blah blah \cite{ref1}.
2 Blah blah blah \cite{ref1,ref2,ref3}
```

Where `ref1`, `ref2` and `ref3` are the unique identifiers that you give to each reference (you'll understand what I mean when you open JabRef).

## 5 Tikz Pictures

Tikz creates high quality images directly in your document! It's similar to a vector based system where you specify the co-ordinates of points and draw lines between them. I'll leave it up to you to read all of the documentation but Figure 2 demonstrates the quality of a Tikz image.

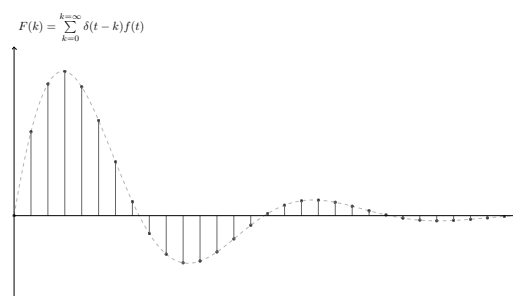


Figure 2: My Tikz Plot.

In most cases I find it easiest to create a standalone document with your Tikz image:



```

1 \documentclass{standalone}
2 \usepackage{tikz}
3 \usepackage{amsmath}
4 \usepackage{pgfplots}
5
6 \begin{document}
7 \begin{tikzpicture}
8
9     % Your Image Goes Here!
10
11 \end{tikzpicture}
12 \end{document}

```

The standalone document allows you to build the file and view it without having to build the whole project. It's not a requirement and does add overhead but it allows you to quickly edit images and view them and import them into your project later!

You then include the image you've just made into your report using:

```

1 \begin{figure}[ht!]
2 \begin{center}
3 \includestandalone[width=0.9\columnwidth]{tikzimagefile}
4 \end{center}
5 \caption{Caption here}
6 \label{fig:myFigure}
7 \end{figure}

```

This will automatically scale the image for you to fit in the column!

## 5.1 Tikz Plotting CSV Files

Sometimes we need to plot CSV data points directly. Figure 3 demonstrates a plot of a .csv file using Tikz. The axis environment is used and for each plot that should be added the following code is used:

```

1 \addplot table [x=x col name, y=y col
  name, col sep=comma] {file.csv};

```

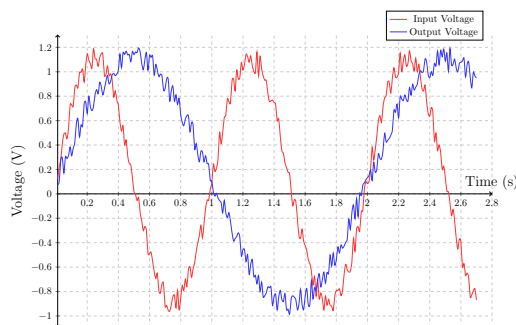


Figure 3: Plotting from a CSV.

Take a look at <http://pgfplots.sourceforge.net/gallery.html> for more examples of plots

of co-ordinates and check out <http://pgfplots.sourceforge.net/pgfplots.pdf> for the full documentation. Check out the Section titled “Markers, Linestyles, (Background-) Colors and Colormaps” for useful information that you'll often use in your plots. Figure 4 demonstrates the use of different markers (squares and triangles) in a scatter plot.

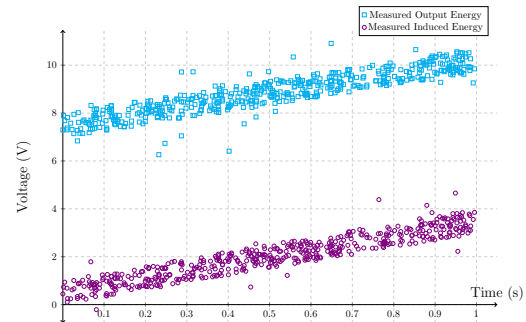


Figure 4: Scatter Plot from a CSV.

One thing to keep in mind is that you're using figures to portray information to the reader. If you can show more information in a single plot you should! Figure 5 demonstrates how you can show more information in a single plot by using two axes instead of one.

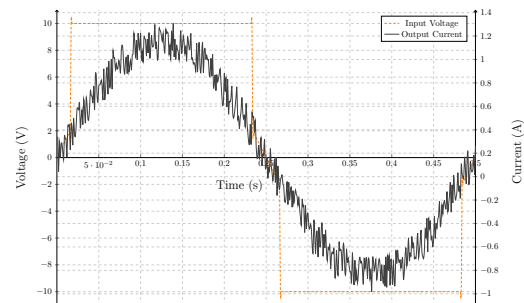


Figure 5: Multiple Axes to Show More Information in a Single Plot.

## 5.2 Tikz Circuits

You can also create full circuits in Tikz by using the following package:

```

1 \usepackage[siunitx]{circuitikz}

```

Now you can create a standalone document with the following structure:

```

1 \documentclass{standalone}
2 \usepackage{tikz}
3 \usepackage{amsmath}
4 \usepackage{pgfplots}

```

```

5 \usepackage[siunitx]{circuitikz}
6
7 \begin{document}
8 \begin{circuitikz}
9
10 \end{circuitikz}
11 \end{document}

```

Again you get a high quality circuit output as demonstrated in Figure 6. The circuit components are all defined in the document at <http://texdoc.net/texmf-dist/doc/latex/circuitikz/circuitikzmanual.pdf>.

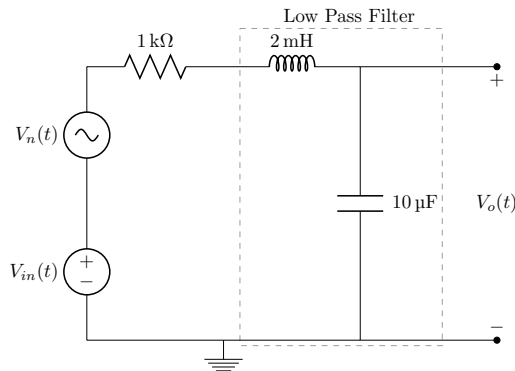


Figure 6: My Tikz Circuit.

### 5.3 Tikz Flow Diagrams

As an engineering student you WILL need flow diagrams! Check out Figure 7 below. Tikz provides a great way to create flow diagrams in your document. The Tikz code still needs some work in terms of shaping and positioning of blocks but it's easy to move them around.

### 5.4 Tikz 3D Models

Tikz also supports 3D models, in order to use the 3D functionality you just need to write your co-ordinates in the form  $(x, y, z)$ . The output is a 2D view of your 3D model, as demonstrated in Figure 8 below<sup>3</sup>. Check out this URL <http://tex.stackexchange.com/questions/42812/3d-bodies-in-tikz> to see how to plot spheres, cones, cubes, and more.

## 6 Creating New Commands

L<sup>A</sup>T<sub>E</sub>X allows you to create your own commands. This allows you to adhere to the “Do Not Repeat

<sup>3</sup>Thanks to Harish Vallabhapurapu for the code and permission to use it!

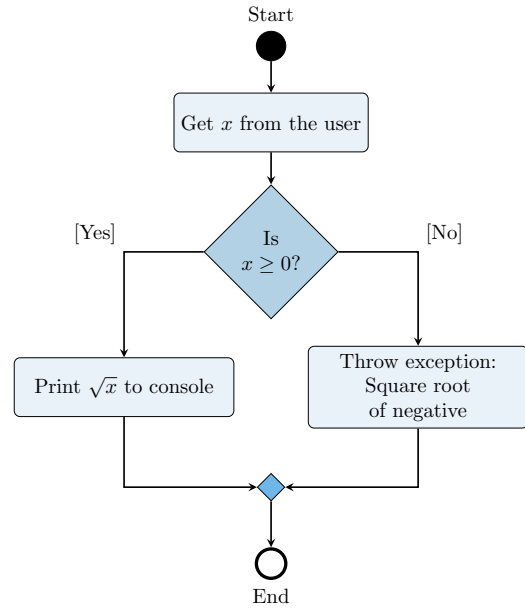


Figure 7: My Tikz Flow Diagram.

Yourself” (DRY) principle. For instance, if you always add figures in the same way, you could create a command that takes care of most of the information for you.

The code that creates Figure 9 image is as follows:

```

1 \tikzimage{0.8}{circuit}{This Image has
  Been Imported in a Single Line!}

```

I’m able to do this because I created a command in the following way:

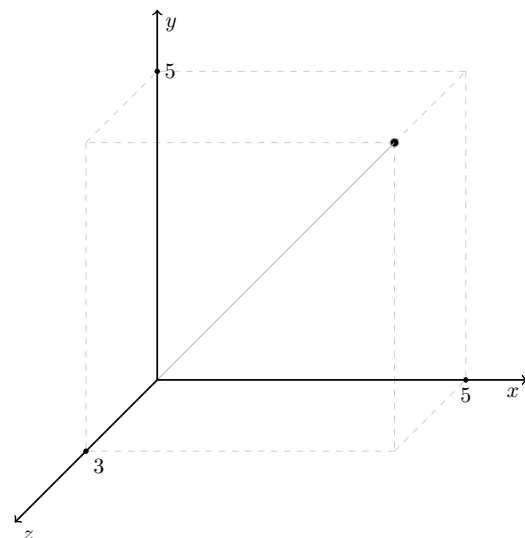


Figure 8: An Example of a Tikz 3D Model.

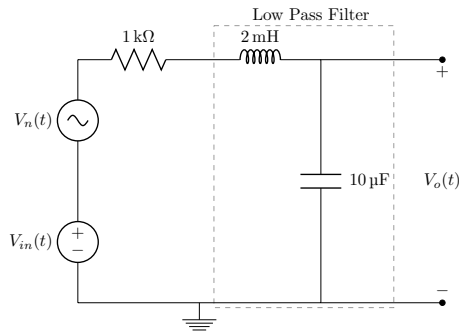


Figure 9: This Image has Been Imported in a Single Line!

```

1 \newcommand{\tikzimage}[3]{
2   \begin{figure}[h!]
3     \centering
4     \includestandalone[width=.#1\linewidth]{#2}
5     \caption[#3]{#3}
6     \label{fig:#2}
7   \end{figure}
8 }

```

This is a pretty advanced step and if you're interested in learning how to do it properly, check out the following link, <https://www.sharelatex.com/learn/Commands>.

## 7 Conclusion

You have now seen both the basic and advanced aspects of  $\text{\LaTeX}$ . But seeing is not enough! The only way you'll learn how to use  $\text{\LaTeX}$  is through practical applications, running into errors and troubleshooting them. One thing that you should always remember is that **GOOGLE IS YOUR FRIEND**. Troubleshooting  $\text{\LaTeX}$  on your own definitely improves your problem solving abilities as an Engineer. So don't just ask people for help when you're struggling - first try to solve the error yourself.

## License

Learning LaTeX is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

You should have received a copy of the license along with this work. If not, see <http://creativecommons.org/licenses/by-nc-sa/4.0/>.

## A Extra Stuff to Look Up

If you're still interested in learning  $\text{\LaTeX}$  I'd suggest you find you about a few more things:

- Alignment in a mathematical equation
- Spacing in a mathematical equation
- Page numbering
- Appendices
- TabularX to control the widths of the columns in a table

## B Tikz Images

The code that creates the Tikz image in Figure 2 is as follows:

```

1 \documentclass{standalone}
2 \usepackage{tikz}
3 \usepackage{amsmath}
4 \usepackage{pgfplots}
5 \usepackage{ifthen}
6 \ifthenelse{\isundefined{\samplesize}}{
7 {
8     \newcommand{\samplesize}{500}
9 }
10 {
11 }
12
13 \begin{document}
14 \begin{tikzpicture}
15     %Axes
16     \draw[>, thick] (0, 0) -- (15, 0) node[below] {$t$};
17     \draw[<-, thick] (0, 5) node[above right] {$F(k) = \sum\limits_{k=0}^{\infty} \delta(t-k) f(t)$} -- (0, -2.5);
18     \draw[smooth, draw opacity = 0.3, dashed] plot[samples=\samplesize, domain=0:14.8]
19         (\x, { 7 * exp(-0.3*\x) * sin( 2*pi*2/14.8*\x r) });
20     \foreach \i in {0,...,29}{
21         \draw[fill, draw opacity = 0.6, ultra thin] (\i/2, 0) -- plot[domain=\i/2:\i/2+1]
22             (\x, { 7 * exp(-0.3*\i/2) * sin( 2*pi*2/14.8*\i/2 r) }) circle (1pt);
23     };
24 \end{tikzpicture}
25 \end{document}

```

The code that creates the Tikz image in Figure 3 is as follows:

```

1 \documentclass{standalone}
2 \usepackage{tikz}
3 \usepackage{amsmath}
4 \usepackage{pgfplots}
5 \usepackage{siunitx}
6 \usepackage{ifthen}
7 \ifthenelse{\isundefined{\samplesize}}{
8 {
9     \newcommand{\samplesize}{500}
10 }
11 {
12 }
13
14 \begin{document}
15 \pgfmathsetmacro{\newPageScale}{0.8}%
16 \pgfmathsetmacro{\axiswidth}{21*\newPageScale}%
17 \pgfmathsetmacro{\axisheight}{14.8*\newPageScale}%
18 % These macros are used to shape the graph to fit nicely in a report.
19 \begin{tikzpicture}
20     \pgfplotsset{every non boxed x axis/.append style={x axis line style=>}}%
21     \pgfplotsset{every non boxed y axis/.append style={y axis line style=<-, thick}}%
22     \begin{axis}[
23         width=\axiswidth*1cm, height=\axisheight*1cm,%
24         axis y line=left,%
25         axis x line=middle,%
26         xlabel=\Large{Time (s)},%
27         ylabel=\Large{Voltage (V)},%
28         xmax = 2.8,
29         enlarge y limits=0.05,%
30         x label style={%
31             %at={(axis description cs:0,-0.01)},%
32             anchor=south%
33         },%
34         y label style={%
35             at={(axis description cs:-0.01,.5)},%
36             },%
37         legend style={%

```

```

38         anchor= south east%
39     },%
40     grid style={dashed,black!20},%
41     grid = both%
42 ]%
43 \addplot[mark=.,red!80,thick,smooth] table [x=t, y=v, col sep=comma] {plot.csv};
44 \addplot[mark=.,blue!80,thick,smooth] table [x=t, y=v, col sep=comma] {plot2.csv};
45 \legend{Input Voltage, Output Voltage};
46 \end{axis}
47 \end{tikzpicture}
48 \end{document}

```

The code that creates the scatter plot in Figure 4 is as follows:

```

1 \documentclass{standalone}
2 \usepackage{tikz}
3 \usepackage{amsmath}
4 \usepackage{pgfplots}
5 \usepackage{siunitx}
6 \usepackage{ifthen}
7 \ifthenelse{\isundefined{\samplesize}}{
8 {
9     \newcommand{\samplesize}{500}
10 }
11 {
12 }
13
14 \begin{document}
15 \pgfmathsetmacro{\newPageScale}{0.8}%
16 \pgfmathsetmacro{\axiswidth}{21*\newPageScale}%
17 \pgfmathsetmacro{\axisheight}{14.8*\newPageScale}%
18 % These macros are used to shape the graph to fit nicely in a report.
19 \begin{tikzpicture}
20     \pgfplotsset{every non boxed x axis/.append style={x axis line style=>}}%
21     \pgfplotsset{every non boxed y axis/.append style={y axis line style=<->, thick}}%
22     \begin{axis}[
23         width=\axiswidth*1cm, height=\axisheight*1cm,%
24         axis y line=left,%
25         axis x line=middle,%
26         xlabel=\Large{Time (s)},%
27         ylabel=\Large{Voltage (V)},%
28         xmax = 1.05,
29         enlarge y limits=0.05,%
30         x label style ={%
31             %at={{(axis description cs:0,-0.01)}},%
32             anchor=south%
33         },%
34         y label style={%
35             at={{(axis description cs:-0.01,.5)}},%
36         },%
37         legend style={%
38             anchor= south east%
39         },%
40         grid style={dashed,black!20},%
41         grid = both%
42     ]%
43     \addplot[only marks,mark=square,scale=0.1,cyan] table [x=t, y=inenergy, col sep=comma] {scatter.csv};
44     \addplot[only marks,mark=o,scale=0.1,violet] table [x=t, y=inducedenergy, col sep=comma] {scatter.csv};
45     \legend{Measured Output Energy, Measured Induced Energy};
46 \end{axis}
47 \end{tikzpicture}
48 \end{document}

```

The code that creates a multi-axis plot in Figure 5 is as follows:

```

1 \documentclass{standalone}

```

```

2 \usepackage{tikz}
3 \usepackage{amsmath}
4 \usepackage{pgfplots}
5 \usepackage{siunitx}
6 \usepackage{ifthen}
7 \ifthenelse{\isundefined{\samplesize}}{
8 {
9 \newcommand{\samplesize}{500}
10 }
11 {
12 }
13
14 \begin{document}
15 \pgfmathsetmacro{\newPageScale}{0.8}%
16 \pgfmathsetmacro{\axiswidth}{21*\newPageScale}%
17 \pgfmathsetmacro{\axisheight}{14.8*\newPageScale}%
18 % These macros are used to shape the graph to fit nicely in a report.
19 \begin{tikzpicture}
20 \pgfplotsset{every non boxed x axis/.append style={x axis line style=>} }%
21 \pgfplotsset{every non boxed y axis/.append style={y axis line style=<->, thick}
22 }%
23 \pgfplotsset{set layers}
24 \begin{axis}[
25 scale only axis,
26 width=\axiswidth*1cm, height=\axisheight*1cm,%
27 axis y line=left,%
28 axis x line*=middle,%
29 xlabel=\Large{Time (s)},%
30 ylabel=\Large{Voltage (V)},%
31 xmax = 0.5, xmin = 0,
32 enlarge y limits=0.05,%
33 x label style ={%
34 at={(axis description cs:0.5,0.48)},%
35 anchor=north east%
36 },%
37 y label style={%
38 at={(axis description cs:-0.01,.5)}},%
39 },%
40 legend style={%
41 anchor= south east%
42 },%
43 grid style={dashed, black!20},%
44 grid = both%
45 ]%
46 \addplot[mark=., scale=0.05, orange, smooth, densely dashed, thick] table [x=t, y=
47 input, col sep=comma] {2Axes.csv}; \label{plot_voltage}
48 \end{axis}
49 \begin{axis}[
50 scale only axis,
51 width=\axiswidth*1cm, height=\axisheight*1cm,%
52 axis y line=right,%
53 axis x line=none,%
54 xlabel=\Large{Time (s)},%
55 ylabel=\Large{Current (A)},%
56 xmax = 0.5, xmin = 0,
57 enlarge y limits=0.05,%
58 x label style ={%
59 anchor=south%
60 },%
61 y label style={%
62 at={(axis description cs:1.01,.4)},%
63 anchor=north west
64 },%
65 legend style={%
66 anchor= north east%
67 },%
68 grid style={dashed, black!20},%
69 grid = both%
70 ]%
71 \addlegendimage{/pgfplots/refstyle=plot_voltage}\addlegendentry{Input Voltage}

```

```

70 \addplot[mark=.,scale=0.05,darkgray,smooth,thin] table [x=t, y=output, col sep
      =comma] {2Axes.csv}; \label{plot_current}
71 \addlegendentry{Output Current}
72 \end{axis}
73 \end{tikzpicture}
74 \end{document}

```

The code that creates the Tikz image in Figure 6 is as follows:

```

1 \documentclass{standalone}
2 \usepackage{tikz}
3 \usepackage{amsmath}
4 \usepackage{pgfplots}
5 \usepackage[siunitx]{circuitikz}
6
7 \usepackage{ifthen}
8 \ifthenelse{\isundefined{\samplesize}}{
9 {
10 \newcommand{\samplesize}{500}
11 }
12 {
13 }
14
15 \begin{document}
16 \begin{circuitikz}[american voltages]
17 \draw
18 (0, 0) to [V=$V_{in}(t)$] (0, 2.5)
19 to [sV=$V_n(t)$] (0, 5)
20 to [R, l=1<\kilo\ohm>] (2.5, 5)
21 to [L, l=2<\milli\henry>] (5, 5)
22 to [C, l=10<\micro\farad>] (5, 0) — (0, 0);
23 \draw (2.5, 0) node[ground] {};
24 \draw[dashed, draw opacity=0.4] (2.8, 5.7) — (6.5, 5.7) — (6.5, -0.1) — (2.8,
      -0.1) — cycle;
25 \draw (5, 5) — (7.5, 5) node[circ,] {};
26 \draw (5, 0) — (7.5, 0) node[circ,] {};
27 \draw (7.5, 5) node[anchor=north] {$+$};
28 \draw (7.5, 0) node[anchor=south] {$-$};
29 \draw (7.5, 2.5) node {$V_o(t)$};
30 \draw (5-0.2,5.7) node[anchor=south] {Low Pass Filter};
31 \end{circuitikz}
32 \end{document}

```

The code that creates the flow diagram in Figure 7 is as follows:

```

1 \documentclass{standalone}
2 \usepackage{tikz}
3 \usepackage{amsmath}
4 \usepackage{pgfplots}
5 \usepackage{siunitx}
6 \usepackage{xcolor}
7
8 \usepackage{ifthen}
9 \ifthenelse{\isundefined{\samplesize}}{
10 {
11 \newcommand{\samplesize}{500}
12 }
13 {
14 }
15
16 % Check out https://www.sharelatex.com/blog/2013/08/29/tikz-series-pt3.html for
      details on the flow diagram building blocks!
17 \definecolor{decisionColour}{HTML}{B2D1E5}
18 \definecolor{processColour}{HTML}{E9F2F9}
19 \definecolor{extraNodeColour}{HTML}{6FB7E9}
20 \usetikzlibrary{shapes.geometric, arrows}
21 \tikzstyle{start} = [circle, minimum width=0.5cm, minimum height=0.5cm, text centered,
      draw=black, fill=black!100, yshift=-0.3cm]
22 \tikzstyle{end} = [circle, minimum width=0.5cm, minimum height=0.5cm, text centered,
      draw=black, fill=black!0, yshift=-0.3cm, ultra thick]
23 \tikzstyle{process} = [rectangle, minimum width=3cm, minimum height=1cm, text centered
      , draw=black, fill=processColour, rounded corners=0.1cm]

```

```

24 \tikzstyle{decision} = [diamond, text badly centered, draw=black, fill=decisionColour,
    text width = 3cm]
25 \tikzstyle{extraNode} = [diamond, minimum width=0.25cm, minimum height=0.25cm, text
    centered, draw=black, fill=extraNodeColour]
26 \tikzstyle{arrow} = [thick,->,>=stealth]
27
28 \begin{document}%
29 \begin{tikzpicture}
30     \node (start) [start] {} node[anchor=south] {Start};
31     \node (procInput) [process, below of = start, yshift=-0.3cm] {Get  $x$  from the
        user};
32     % Change the text width and minimum height to get it to look good, it's best to
        keep them the same so that the decision block is symmetric!
33     \node (if1) [decision, below of = procInput, text width = 1.1cm, minimum height =
        1.1cm, yshift=-1.2cm] {Is  $x \geq 0$ ?};
34     \node (procXGreater) [process, left of = if1, yshift=-2.3cm, xshift=-1.5cm, text
        width = 3.5cm] {Print  $\sqrt{x}$  to console};
35     \node (procXLower) [process, right of = if1, yshift=-2.3cm, xshift=1.5cm, text
        width = 3.5cm] {Throw exception:  $\sqrt{\text{Square root of negative}}$ ;};
36     \node (finalNode) [extraNode, below of = if1, yshift = -3cm] {};
37     \node (end) [end, below of = finalNode] {};
38     \draw (end) node[anchor=north, yshift=-0.3cm] {End};
39     \draw [arrow] (start) — (procInput);
40     \draw [arrow] (procInput) — (if1);
41     \draw [arrow] (if1) —| node[anchor=south east] {[Yes]} (procXGreater);
42     % The —| means horizontal then vertical line
43     % That means that |— would draw a vertical line first and then a horizontal.
        Change them around to see what they do!
44     \draw [arrow] (if1) —| node[anchor=south west] {[No]} (procXLower);
45     \draw [arrow] (procXGreater) |— (finalNode);
46     \draw [arrow] (procXLower) |— (finalNode);
47     \draw [arrow] (finalNode) — (end);
48 \end{tikzpicture}
49 \end{document}

```

The code that creates the 3D Tikz image in Figure 8 is as follows:

```

1 \documentclass{standalone}
2 \usepackage{tikz}
3 \usepackage{amsmath}
4 \usepackage{pgfplots}
5 \usepackage{ifthen}
6 \ifthenelse{\isundefined{\samplesize}}{
7 {
8     \newcommand{\samplesize}{500}
9 }
10 {
11 }
12 \begin{document}
13 \begin{tikzpicture}
14     \draw[thick,->] (0,0,0) — (6,0,0) node[anchor=north east]{ $x$ };
15     \draw[thick,->] (0,0,0) — (0,6,0) node[anchor=north west]{ $y$ };
16     \draw[thick,->] (0,0,0) — (0,0,6) node[anchor=north west]{ $z$ };
17     \draw[fill, draw opacity = 0.3] (0,0,0) — (5,5,3) circle (2pt);
18     \draw[dashed, draw opacity = 0.2] (5,5,3) — (5,0,3);
19     \draw[dashed, draw opacity = 0.2] (5,5,3) — (0,5,3);
20     \draw[dashed, draw opacity = 0.2] (5,5,3) — (5,5,0);
21     \draw[dashed, draw opacity = 0.2] (5,0,3) — (5,0,0);
22     \draw[dashed, draw opacity = 0.2] (5,0,3) — (0,0,3);
23     \draw[dashed, draw opacity = 0.2] (0,5,3) — (0,5,0);
24     \draw[dashed, draw opacity = 0.2] (0,5,3) — (0,0,3);
25     \draw[dashed, draw opacity = 0.2] (5,5,0) — (5,0,0);
26     \draw[dashed, draw opacity = 0.2] (5,5,0) — (0,5,0);
27     \draw[fill] (0,0,3) node[anchor=north west] {$z$} circle (1pt);
28     \draw[fill] (0,5,0) node[anchor=west] {$y$} circle (1pt);
29     \draw[fill] (5,0,0) node[anchor=north] {$x$} circle (1pt);
30 \end{tikzpicture}
31 \end{document}

```