

# ECE 20007

LAB 15

## EXAMPLE REPORT

Author	Lab Partner
Example Author	Example Partner

December 10, 2021

### Contents

<b>15.1</b>	<b>A quick introduction</b>	<b>2</b>
15.1.1	Basic editing . . . . .	2
15.1.2	Document structure . . . . .	2
<b>15.2</b>	<b>Some more features</b>	<b>3</b>
15.2.1	Code blocks . . . . .	3
15.2.2	Circuit diagrams . . . . .	3
15.2.3	Plots . . . . .	4
15.2.4	. . . . .	5

### List of Circuits

1	Example Circuit . . . . .	4
---	---------------------------	---

### List of Plots

1	. . . . .	4
---	-----------	---

## 15.1 A quick introduction

### 15.1.1 Basic editing

Overleaf has written a much better guide than I would be able to – use `https://www.overleaf.com/learn/latex/Tutorials` to learn how to use L<sup>A</sup>T<sub>E</sub>X.

### 15.1.2 Document structure

This package provides task and subtask commands, which are analogous to (and implemented with) the section and subsection commands. For a typical paper, automatic section numbering is preferred. However, when following the lab manual, I preferred using the same section numbers. These commands make it easier to do that.

## 15.2 Some more features

**Objective** The `\objective` command provides an easy way to label objectives. There is also an equivalent `\conclusion` command.

### 15.2.1 Code blocks

`\inputcode[listings options]{filename.ext}`

---

`example.py`

---

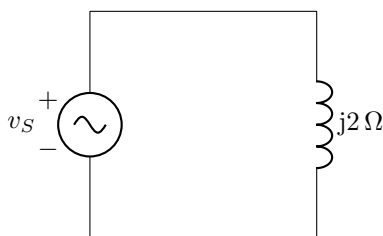
```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # thevenin voltage and impedance
5 v = 4/5 + 2/5*1j
6 z = 1 - 1j
7
8 # range to test input load angle at
9 t = np.linspace(-np.pi,np.pi,1000)
10 # input load
11 l = np.absolute(z) * (np.cos(t) + 1j * np.sin(t))
12
13 # power to load
14 p = .5 * np.absolute(v / (z + l)) ** 2 * np.real(l)
15
16 print("degrees at max:", np.degrees(np.angle(l[np.argmax(p)])))
17 print("max value:", np.max(p))
18
19 plt.plot(t*180/np.pi, p)
20 plt.show()
21
22 with open("power", 'w') as data:
23     for p in zip(t*180/np.pi, p):
24         data.write(f"{p[0]} {p[1]}\n")
```

---

The `\inputcode` command provides an easy way to input format code files. This is implemented with the listings package (<https://ctan.org/pkg/listings>), and can be tweaked using its key-value system. The default language is set to Python.

### 15.2.2 Circuit diagrams

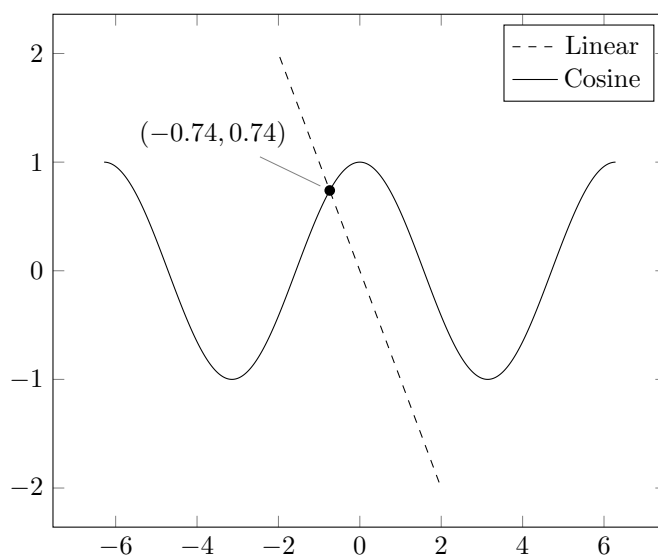
<https://ctan.org/pkg/circuitikz>

**Circuit 1:** Example Circuit

The `circuit` environment is a wrapper for the `circuitikz` environment from the corresponding package. The `circuit` environment sets the diagram to be centered, as well as adding support for `\caption[]{}{}`, `\label{}{}`, and `\autoref{}{}`. There is also a `\listofcircuits`. A starred environment is available that will not be added to the list. The links produced from `\autoref` appear as Circuit 1

### 15.2.3 Plots

<https://ctan.org/pkg/pgfplots>

**Plot 1**

The `plot` environment is a wrapper for the `tikzpicture` environment, intended to be used with the `axis` environment to produce plots. The environment does not directly wrap the `axis` environment, to allow more advanced features such as two axes on the same graph. Similar to the `circuit` wrapper above, this adds support for `\caption[]{}{}`, `\label{}{}`, and `\autoref{}{}`, as well as having its own `\listofplots` with a corresponding starred environment. Note that

the caption and label commands must be outside of the axis environment, but inside the plot environment. The autoref links appear as Plot 1.

#### **15.2.4**

Other useful features include an error calculator accessed with `\error{actual}{expected}` and an efficiency calculator accessed with `\efficiency{useful}{total}`. Both of these commands print a number from 0 to 100 with two decimal places.