

Plotting script User manual

Written by
Vivekanand Dhakane

Contents

1. [Requirements](#)
2. [Plotting Single curve plot](#)
3. [Plotting First order derivative curve plot](#)
4. [Logarithmic plot](#)
5. [Plotting multiple curves in a single plot](#)
6. [Errors and their solutions](#)

This python script is used for plotting graphs using output file generating by ngspice

Requirements

For running this script, python needs to be installed on your computer

The following package needs to be installed for using plotting script

1. Matplotlib

To instal Matplotlibl, use the following command

```
pip install matplotlib
```

2. Numpy

To instal numpy, use the following command

```
pip install numpy
```

Plotting Single curve plot

The following design is used for plotting a single curve plot.

Print the value of the parameter that you want to plot. In the following design, we want to plot voltage at node 1 so we printed its value using 'print v(1)'

This is a voltage divider circuit

*Describe circuit

r1 1 2 1k

r2 2 0 1k

v 1 0

*analysis command

.dc v 0 5 0.1

.control

run

*display command

plot v(1)

print v(1)

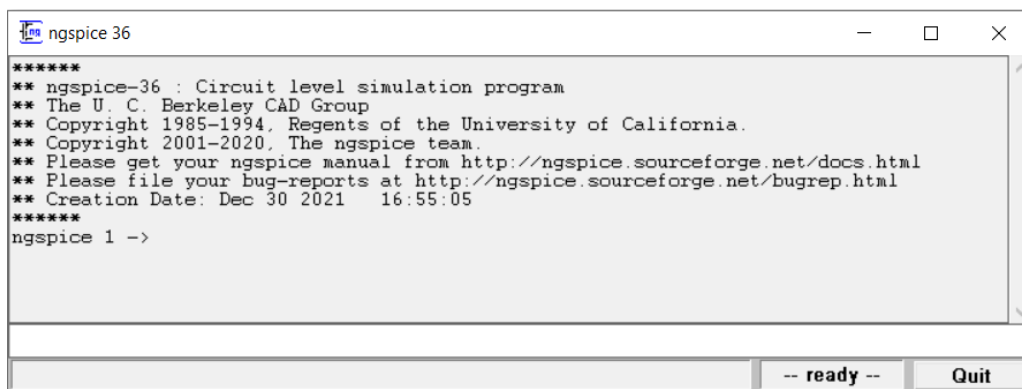
.endc

.end

Following are the steps for plotting a single curve plot

Step 1

Open ngspice

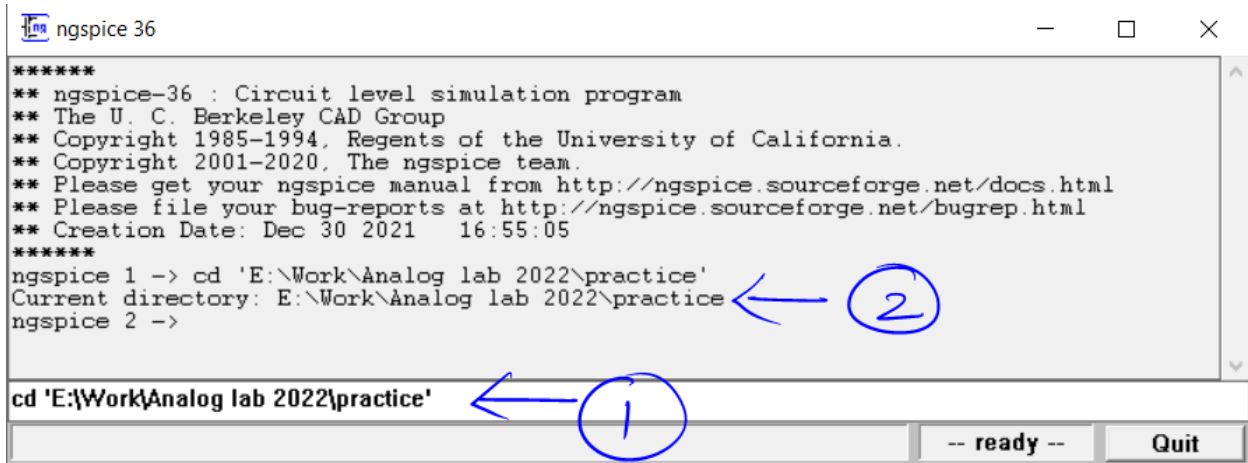


Step 2

Go to your project directory using the cd command

```
cd '<project_directory_path>'
```

The path must be written in single inverted commas as it may contain space



The screenshot shows the ngspice 36 command window. The title bar reads 'ngspice 36'. The main text area contains the following output:

```
*****
** ngspice-36 : Circuit level simulation program
** The U. C. Berkeley CAD Group
** Copyright 1985-1994, Regents of the University of California.
** Copyright 2001-2020, The ngspice team.
** Please get your ngspice manual from http://ngspice.sourceforge.net/docs.html
** Please file your bug-reports at http://ngspice.sourceforge.net/bugrep.html
** Creation Date: Dec 30 2021 16:55:05
*****
ngspice 1 -> cd 'E:\Work\Analog lab 2022\practice'
Current directory: E:\Work\Analog lab 2022\practice
ngspice 2 ->
```

Below the main text area, the command prompt shows the command `cd 'E:\Work\Analog lab 2022\practice'` being entered. A blue arrow points from the command to a blue circle containing the number 1. Another blue arrow points from the current directory output line to a blue circle containing the number 2. The status bar at the bottom shows '-- ready --' and a 'Quit' button.

As shown in the above image check whether the current directory is the same as the directory with your design file

Step 3

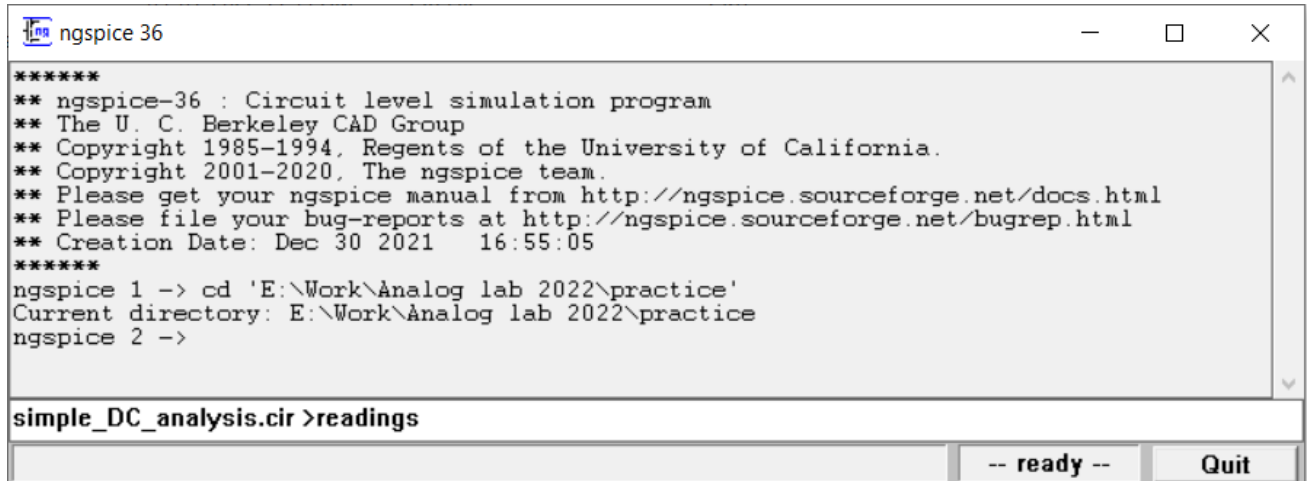
In this step, we are going to run the simulation.

Use the following command

```
design_filename.cir > output_file
```

For example, if my design file name is '*simple_DC_analysis.cir*' and I want the output file name in a file called '*readings*' then the command will be

```
simple_DC_analysis.cir > readings
```

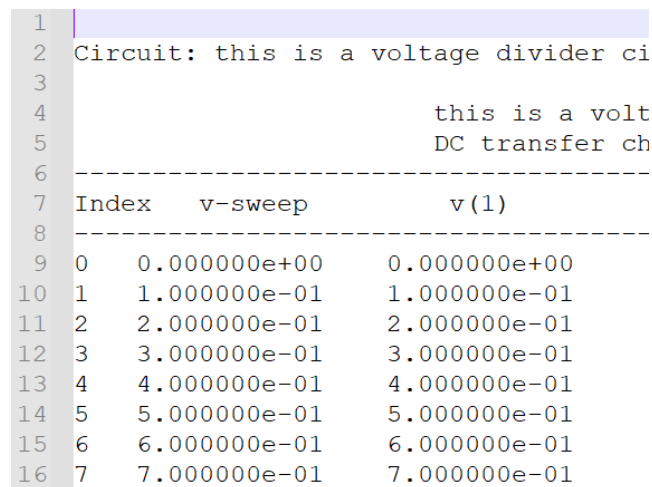


```
ngspice 36
*****
** ngspice-36 : Circuit level simulation program
** The U. C. Berkeley CAD Group
** Copyright 1985-1994, Regents of the University of California.
** Copyright 2001-2020, The ngspice team.
** Please get your ngspice manual from http://ngspice.sourceforge.net/docs.html
** Please file your bug-reports at http://ngspice.sourceforge.net/bugrep.html
** Creation Date: Dec 30 2021 16:55:05
*****
ngspice 1 -> cd 'E:\Work\Analog lab 2022\practice'
Current directory: E:\Work\Analog lab 2022\practice
ngspice 2 ->

simple_DC_analysis.cir >readings
-- ready -- Quit
```

Step 4

After running the simulation using the command in the previous step. The output file will be generated in the same directory. You can open that file using notepad and. The following image shows the same output file.



1			
2	Circuit: this is a voltage divider ci		
3			
4			this is a volt
5			DC transfer ch
6	-----		
7	Index	v-sweep	v(1)
8	-----		
9	0	0.000000e+00	0.000000e+00
10	1	1.000000e-01	1.000000e-01
11	2	2.000000e-01	2.000000e-01
12	3	3.000000e-01	3.000000e-01
13	4	4.000000e-01	4.000000e-01
14	5	5.000000e-01	5.000000e-01
15	6	6.000000e-01	6.000000e-01
16	7	7.000000e-01	7.000000e-01

Step 5

Copy-paste the output file in the directory containing the plotting script

Step 6

Open script and output filename. 'Readings' in my case.

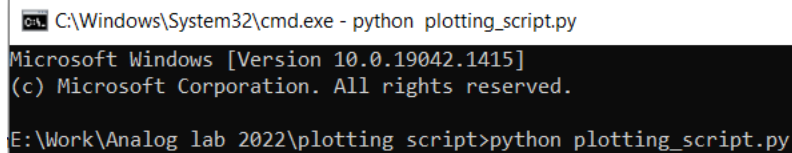
Fill other details like Plot_Title, x_axis_label, y_axis_label

```
filename      = "readings"  
Plot_Title   = "This is a plot title"  
x_axis_label = "x_axis_label"  
y_axis_label = "y_axis_label"  
number_of_curves_in_plot = 1  
legend_list  = ["a"]  
LOG_OF_X     = False  
first_order_derivative = False
```

Step 7

Double click on the script file or Run the script using the following command

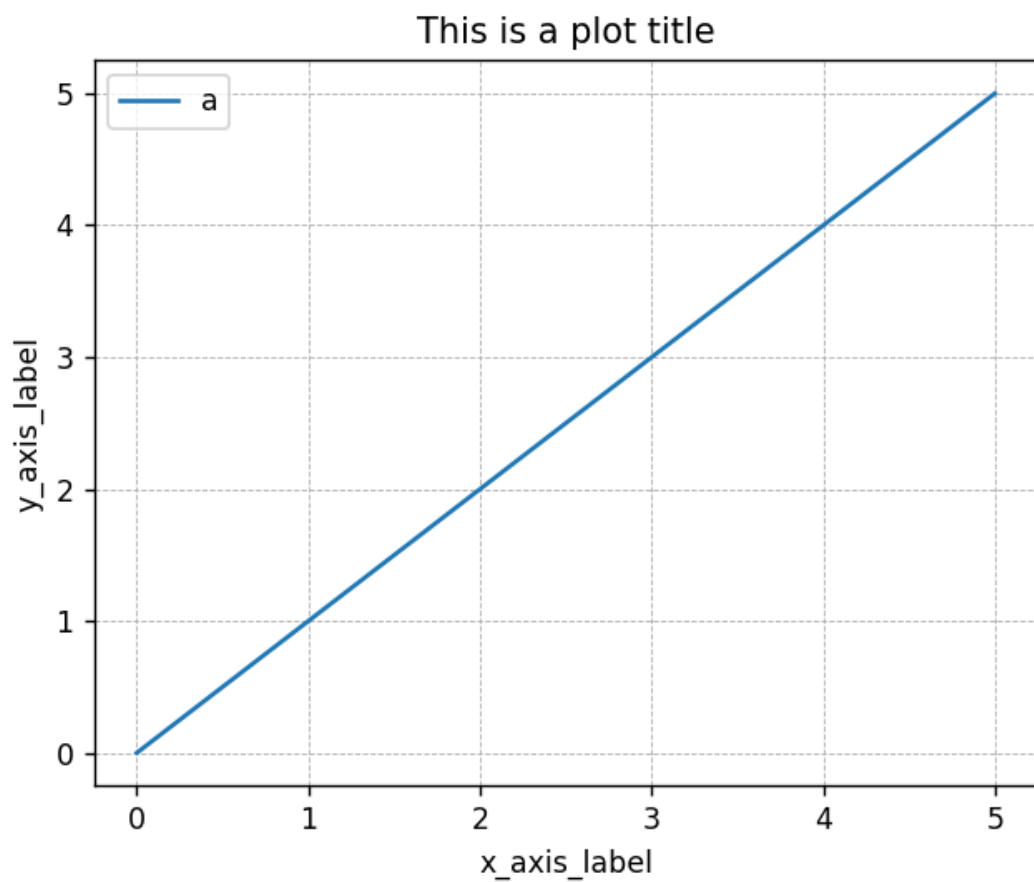
```
python plotting_script.py
```



```
C:\Windows\System32\cmd.exe - python plotting_script.py  
Microsoft Windows [Version 10.0.19042.1415]  
(c) Microsoft Corporation. All rights reserved.  
E:\Work\Analog lab 2022\plotting script>python plotting_script.py
```

The plot will be generated as shown in the following image

Figure 1

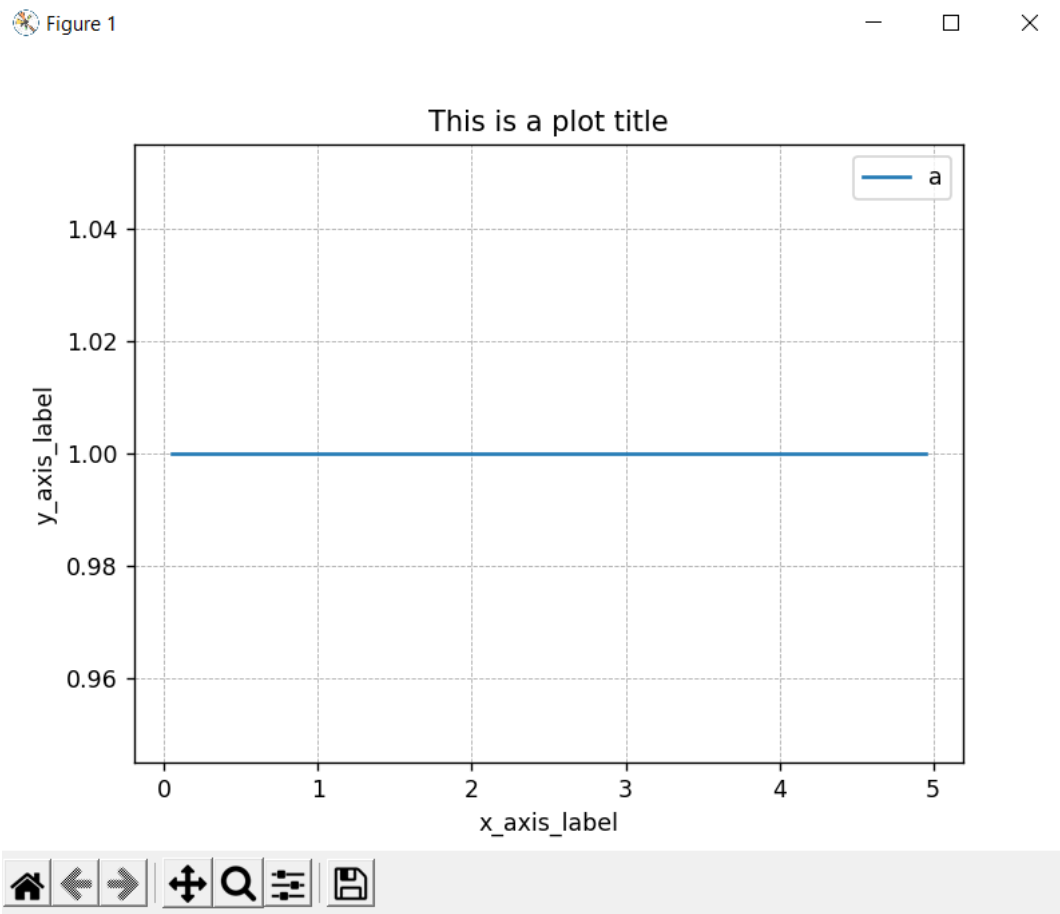


Plotting First order derivative curve plot

For plotting first-order derivative open python script and make ***is_first_order_derivative = True*** as shown in the following image. We are using same output file generated in last tutorial. Now run the python script

```
filename      = "readings"  
Plot_Title   = "This is a plot title"  
x_axis_label = "x_axis_label"  
y_axis_label = "y_axis_label"  
number_of_curves_in_plot = 1  
legend_list  = ["a"]  
LOG_OF_X     = False  
is_first_order_derivative = True  
..
```

Following is the first-order derivative curve of the previous plot



Logarithmic plot

The following design is used for getting the logarithmic plot

```
This is an RC circuit

*Describe circuit
r1 1 2 1k
c 2 0 1u
v 1 0 dc 0 ac 1 $ac analysis
*      ^      ^      ^
* DC_offset AC_Amp for_freq_analysis

*analysis command
.ac dec 10 1 1k
*      ^      ^
*time_step End_time

.control

run

*display command
plot vdb(2)

print vdb(2)

.endc

.end
```

Step 1

Repeat all steps in the previous tutorial and generate readings in an output file

Step 2

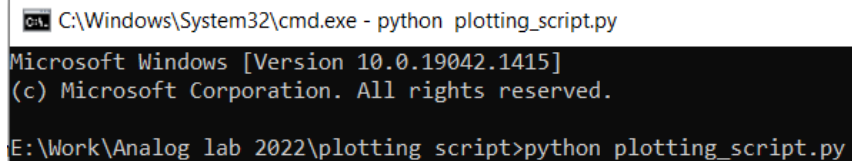
Then for making x-axis logarithmic (with base 10), open python script make **LOG_OF_X = True** as shown in the following image

```
filename      = "db_plot"
Plot_Title    = "This is a plot title"
x_axis_label  = "x_axis_label"
y_axis_label  = "y_axis_label"
number_of_curves_in_plot = 1
legend_list   = ["a"]
LOG_OF_X      = True
is_first_order_derivative = False
```

Step 3

Run the script using the following command

```
python plotting_script.py
```

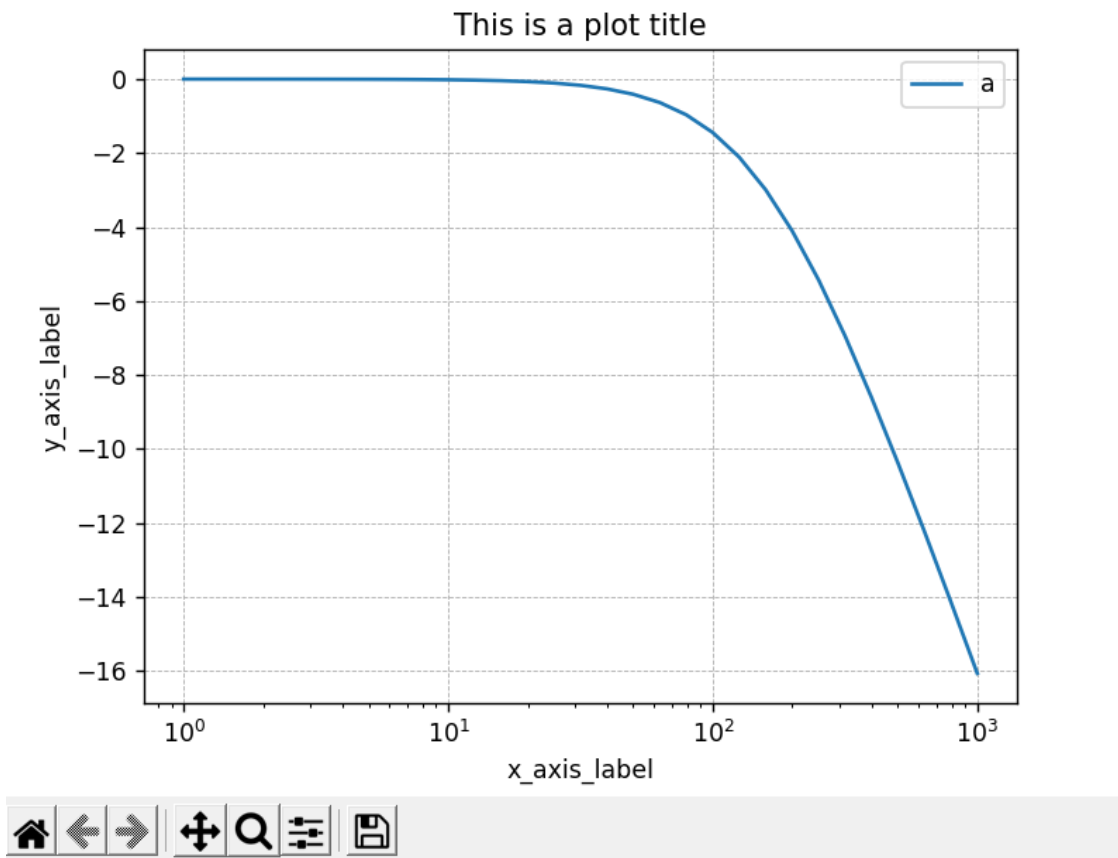


```
C:\Windows\System32\cmd.exe - python plotting_script.py
Microsoft Windows [Version 10.0.19042.1415]
(c) Microsoft Corporation. All rights reserved.

E:\Work\Analog lab 2022\plotting script>python plotting_script.py
```

Then plot will be generated as shown in the following image

Figure 1



Plotting multiple curves in a single plot

The following design is used to plot three curves in a single plot

This is a voltage divider circuit with three resistors

```
*Describe circuit
*<element-name> <nodes> <value/model>
r1 1 2 1k
r2 2 3 1k
r3 3 0 1k
v 1 0 pwl(0 0 10m 0 11m 5 20m 5)
*      ^ ^ ^ ^ ^ ^ ^ ^
*      time value time value time value

*analysis command
.tran 10u 20m
*      ^      ^
*      time_step end_time

.control

run

*display command
plot v(1) v(2) v(3)

print v(1) v(2) v(3)

.endc

.end
```

Step 1

Repeat all steps in the previous tutorial and generate readings in an output file

Step 2

In the above design, we are printing values of V(1), V(2) and V(3). if we want to plot all 3 curves in a single plot open python script and do the following changes

1. Write filename in filename variable
2. ***number_of_curves_in_plot = 3*** ←this can be any value less than or equal to 3
3. ***legend_list = ["v1","v2","v3"]*** ← These are legend that will be displayed in graph
number of legents must be equal to variable ***number_of_curves_in_plot*** set above

Example :

```
filename      = "piecewise_linear_3_resistors"
Plot_Title    = "This is a plot title"
x_axis_label  = "x_axis_label"
y_axis_label  = "y_axis_label"
number_of_curves_in_plot = 3
legend_list   = ["v1", "v2", "v3"]
LOG_OF_X      = False
is_first_order_derivative = False
```

Step 3

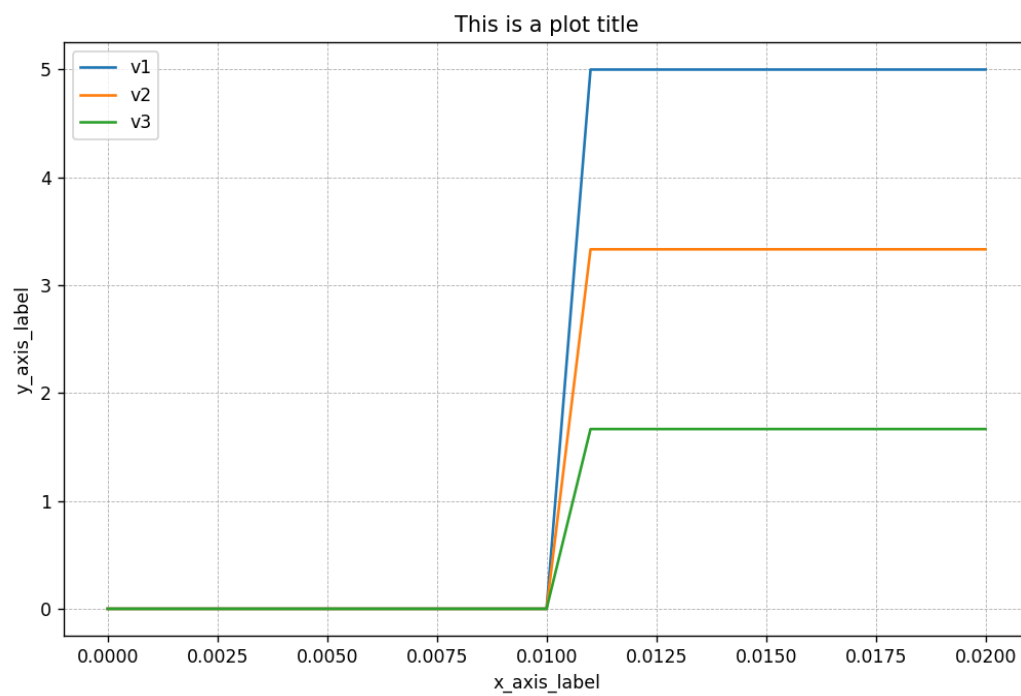
Run the script using the following command

```
python plotting_script.py
```

```
C:\Windows\System32\cmd.exe - python plotting_script.py
Microsoft Windows [Version 10.0.19042.1415]
(c) Microsoft Corporation. All rights reserved.

E:\Work\Analog lab 2022\plotting script>python plotting_script.py
```

The following plot will be generated after running the script



Errors and their solutions

1. **FileNotFoundError: [Errno 2] No such file or directory:**

Check whether the filename entered in the script is correct with its extension(if any, like filename.txt)

2. **Length mismatch Error: number_of_graphs_in_plot and length of legend_list must be the same**

Check whether variable **number_of_curves_in_plot** and number of strings in **legend_list** are the same.

*For example, for plotting 3 curves in a single plot, the following lines will give the above error

```
number_of_curves_in_plot = 3
legend_list = ["a", "b"]
```

* for plotting 3 curves in a single plot, the Correct way is as follows

```
number_of_curves_in_plot = 3
legend_list = ["a", "b", "c"]
```

3. **Column number mismatch Error: Number of curves in the given output file are less than number_of_curves_in_plot**

In ngspice code, check whether you have printed the same number of curves entered in python script in variable **number_of_curves_in_plot**