

II. PSpice Tutorial

PSpice is a software tool for Spice simulation. This tutorial covers the use of OrCAD EE PSpice software package, i.e., Orcad Capture and PSpice Lite. You can download a student version of Orcad from <http://www.orcad.com/buy/orcad-educational-program>.

The tutorial will cover few examples in increasing degree of difficulty. The examples will cover many details of SPICE.

Example 1 Q-Point Analysis

In this example we will construct a simple resistive circuit and measure the currents in, and voltages across, the different resistors.

Start All Programs → cadence → OrCAD 16.6 Lite → OrCAD capture CIS Lite

You will be prompted with an error message saying License was not found.

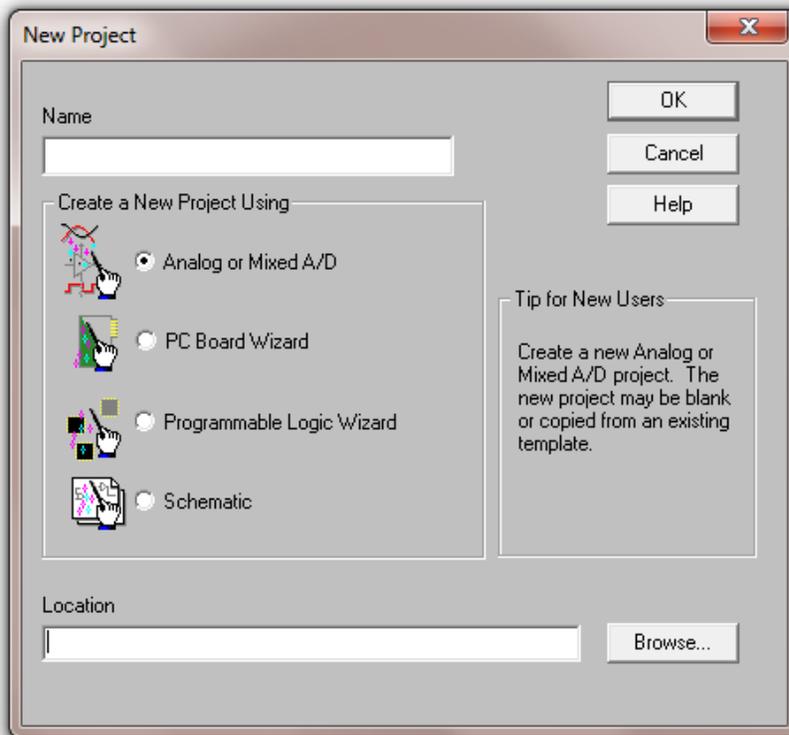
Would you like to launch the lite version

Click “Yes”. You can check don’t ask me again if you do not want to be bothered again.

OrCAD capture CIS Lite windows appears

Click on the project New in the getting started window, or File → New → Project.

The new project windows appear

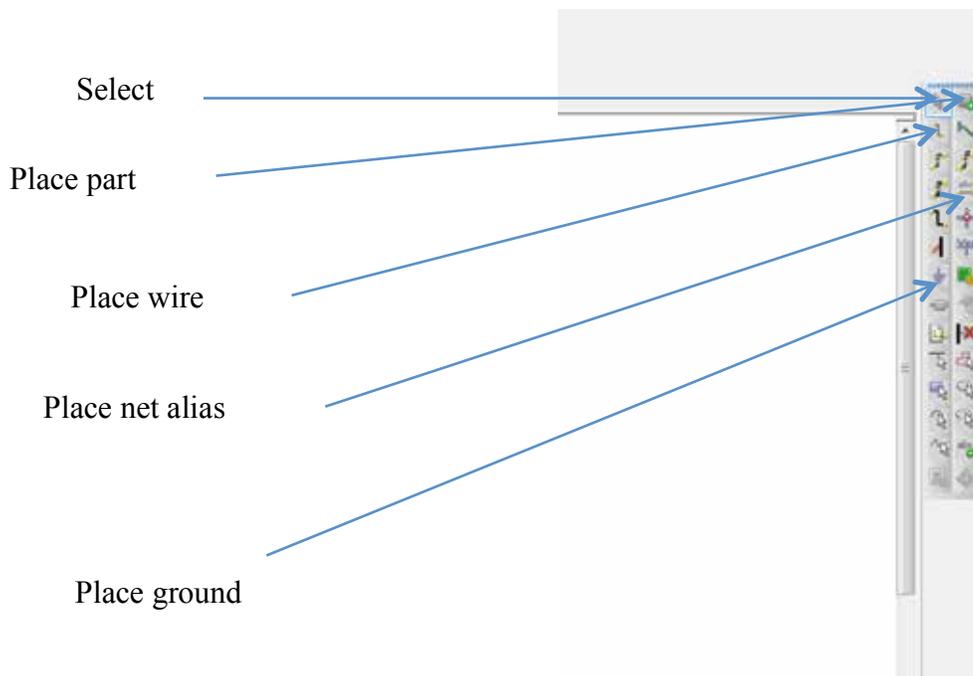
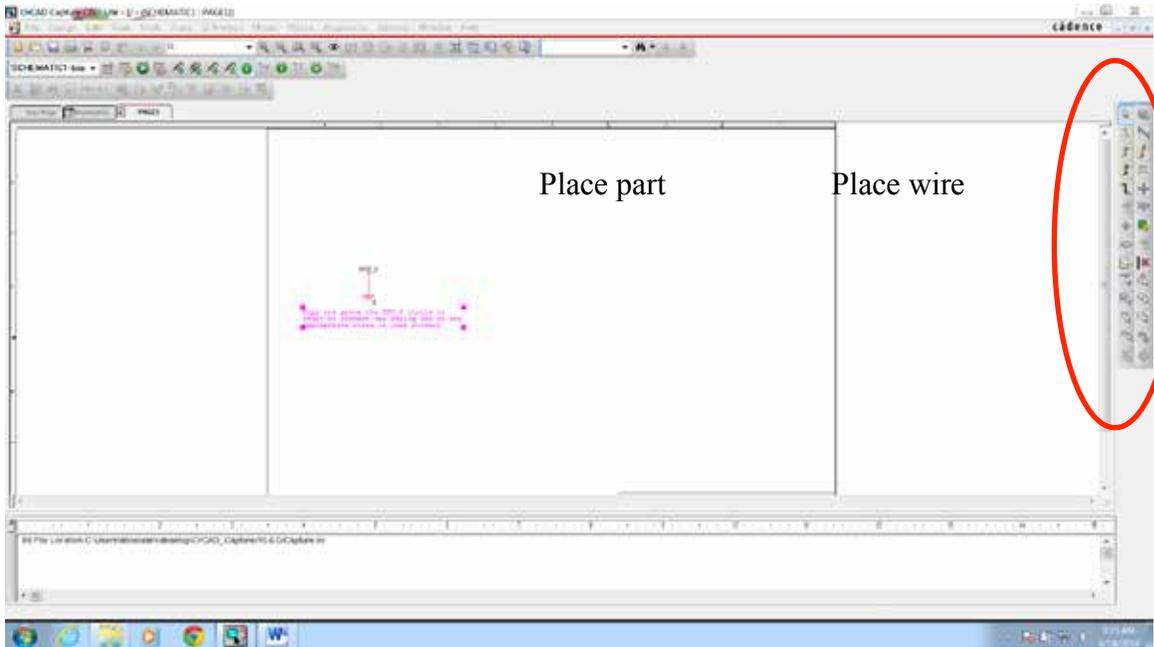


Choose Analog or Mixed A/D

Browse to the appropriate directory location, and name the project MySimpleCircuit Then OK.

Create PSpice Project Window appears, choose “Create a blank project”, then OK. The following screen appears

Expand mysimplecircuit.dsn then expand schematic and double click on page 1

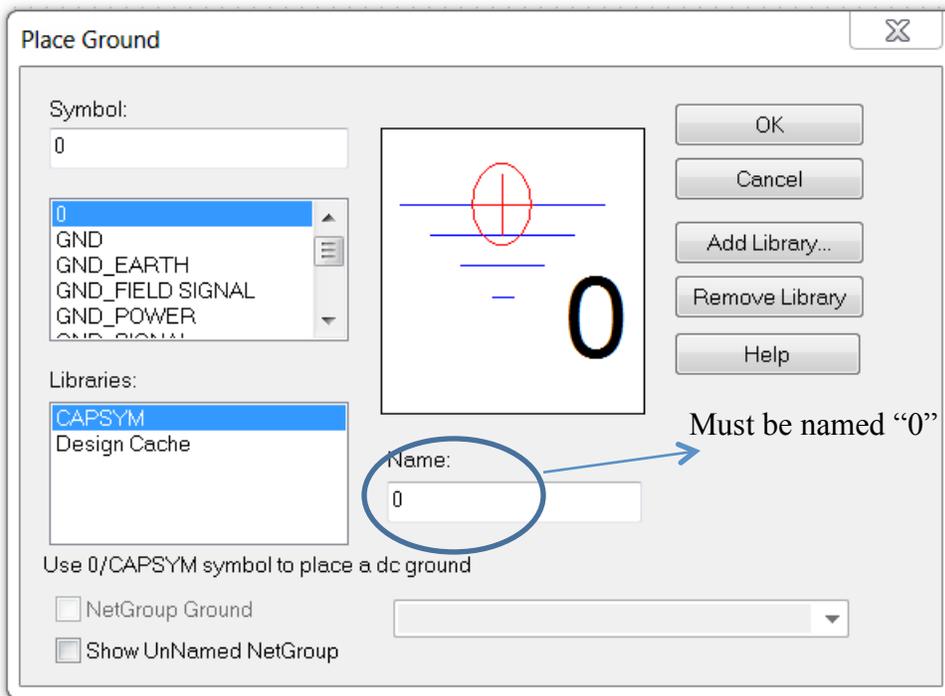


You will get the schematic page, with a ground node in it and text to tell you to copy and paste the ground in your design.

Every circuit must contain a ground and the name of the node is "0" that is the number zero. On the right there is a group of fast access keys, the most important is the place part (could also be accessed from the menu bar Place → part. The place ground button (ground is also a part you can access it from the place part button or menu. The place wire (also from the menu Place → Wire). Place net alias (also from the place menu, and the place ground as shown in the expanded figure

That schematic page is the window where you will build your circuit. The parts will be placed in the schematic one part at a time.

Go to Place → Part, the following window appears (or click on the place part in the side fast access buttons).



Probably it will not include all the libraries shown in the Figure.

The Figure here shows CAPSYM which include the node 0 (ground). You can also choose the node GND, but in this case you must right 0 in the field name.

Click on Add Library a new window appears, choose the PSpice directory.

Some libraries will appear, we need eval and source. A good idea is to add them all though, at least eval and source must be added.

The part windows may appear as a floating window or ducked between the schematic and the tool bar on the right.

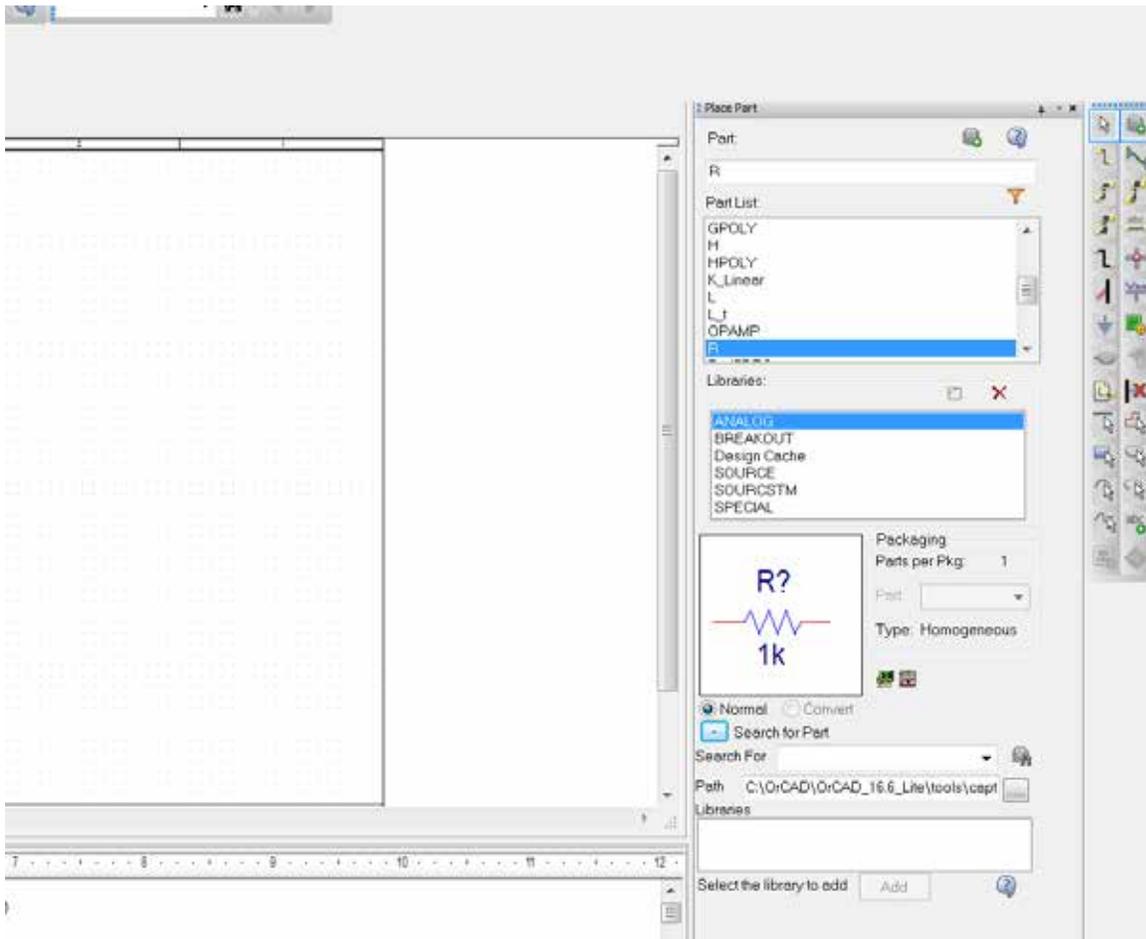
Now we start constructing a simple circuit. Click on Place Part, the following window appears (again, the part window may be floating or ducked).

Choose the analog library

Type R in the part field,

The part list appear as shown, choose R and the resistor picture appear in the box below the library as shown in the window below.

Hit Enter and move the mouse to the schematic part, click to insert a resistor (you can click as many times as you want, when you finish his the Esc key).



Put 3 resistors as shown below, the resistors are automatically named R1, R2, ... with a value of 1K.

DoubleClick on the 1K value of R1, a window will appear with the value set to 1K, you can change this to any value you want. Change it to 5K. In the font part, click change and change it to size 10 (size 5 seems too small). Repeat the font change with the R1 name.

You can set the font in your design this way

Go to MySimpleCircuit tab.

Right click on mysimplecircuit.dsn file

Choose Design properties

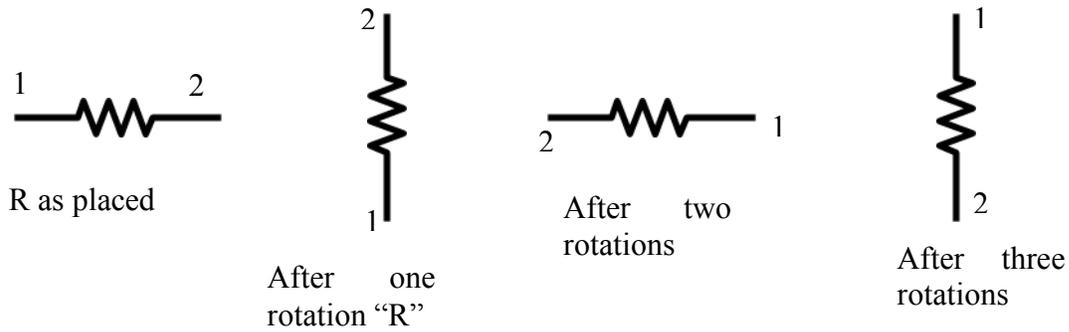
A window appear

Choose Schematic Desig tab

Click on design properties

A window appears where you can choose the fonts type and size for all new parts (font Ariel 10 seems to work fine).

Highlight the second resistor, click R (rotate) each click will rotate it by 90 degrees counter clockwise (this is important, since when you place a resistor, nodes 1 and 2 are automatic to the left and right of the resistor respectively, the current is always measured as going from 1 to 2, if the current is actually from 2 to 1 the current will be negative).



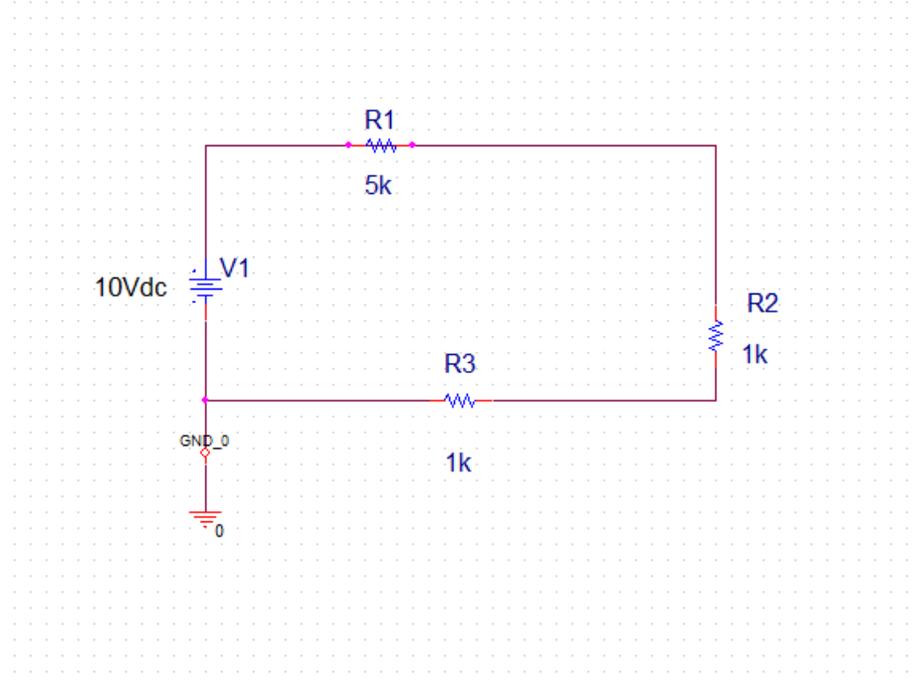
Highlight R3 and hit the R key twice to rotate it 180 degrees.

In the part window, choose the “source” library.

In the part type Vdc a batter symbol appears, highlight it and hit enter and place a battery (source) in the schematic. Double Click on the value (0Vdc) and change it to 10V.

Click on place wire (or Place → Wire)

Connect the parts as shown in the Figure below. Connect also the ground as shown below

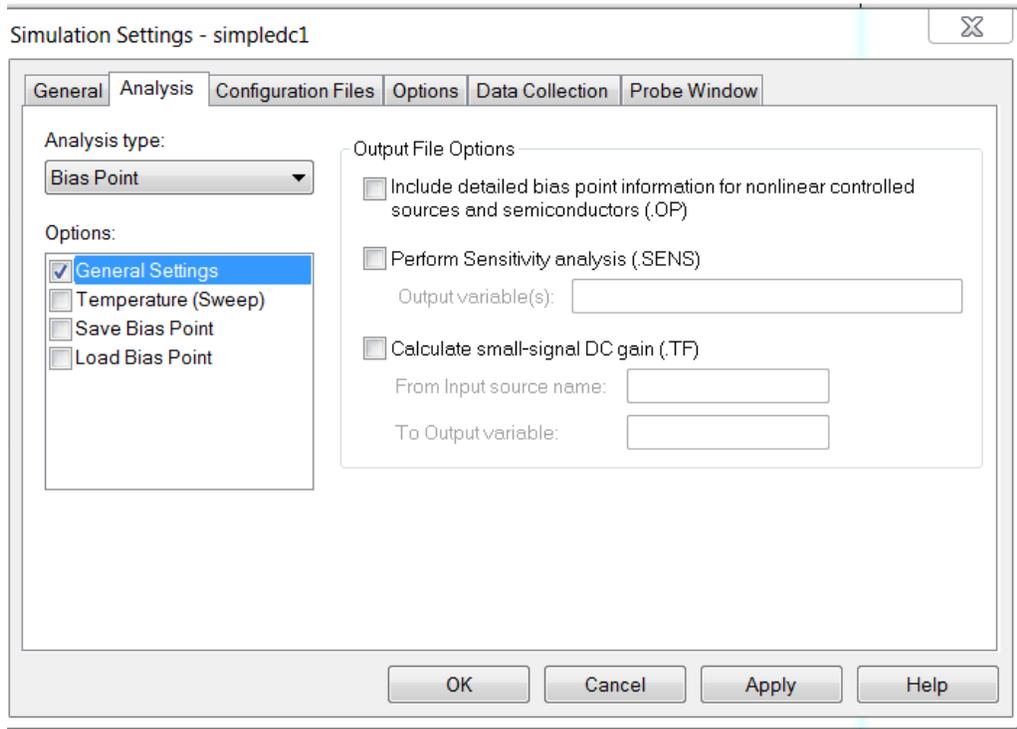


Now, we are ready to simulate the circuit

Go to the top menu, Pspice → New simulation profile.

Choose a suitable name, and inherit from “none”

The following window appears



In the Analysis choose “Bias Poin”, then O.K.
 Then click on the green arrow, or choose PSpice → Run
 The simulation runs.

The simulation window appears, but since we are not using this window (we are only interested in the bias point analysis), we will not be using it.

In the schematic window, right below the main menu bar you find 3 buttons each one is a green circle with “V”, “I”, or “W” inside. These represent the voltage, current, and power respectively. If you click on any one of them, the corresponding value (voltage at every node, current in every circuit element, and power consumed in every circuit element) appears in the circuit schematic.

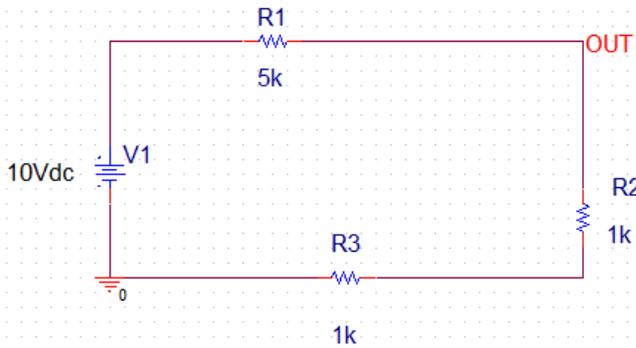
DC Sweep

Now, we will perform a DC sweep. We change one (or more) of the DC sources and check the voltage across, or current, in an element.

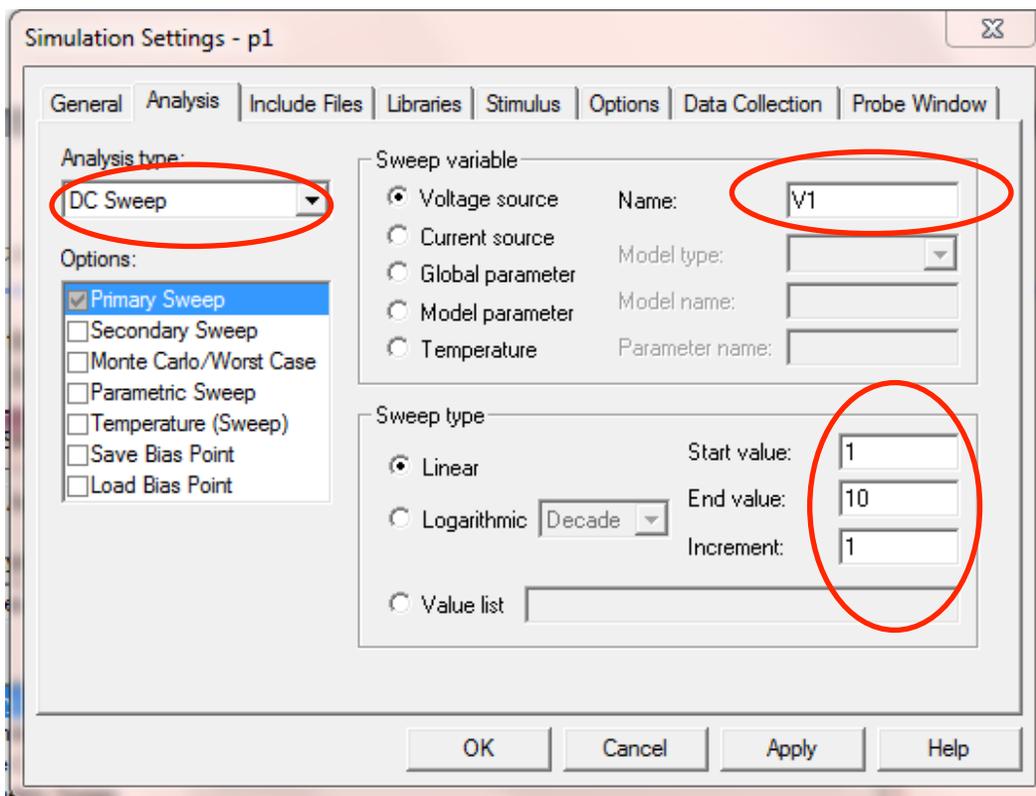
Go back to the schematic diagram. From the right hand fast access keys, choose net alias (a horizontal line with abc written above it), or Place → Net Alice. A windows pops up. In the filed “alias” write OUT. Then OK.

A small square appears attached to the mouse cursor. Move it to the intersection of the horizontal line connecting R1 to the vertical line connecting R1, then click.

The label (net alias) “OUT” is shown there as shown below. It is just a handy way of describing this node.



From the top menu. Choose PSpice → edit simulation profile, you get the following window



In the Analysis tab set the “Analysis type” to be DC Sweep. Choose primary sweep from “Options”.

In Sweep variable, choose Voltage source

Set name to V1 (The name of the voltage source in your schematic).

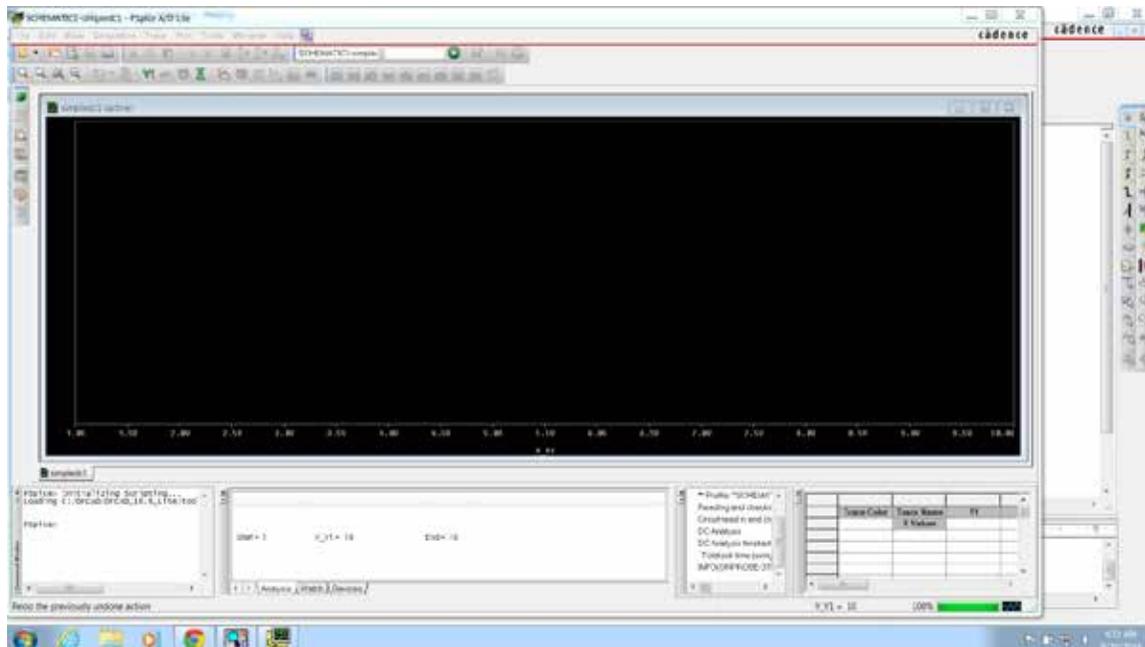
In Sweep type, choose linear, start value = 1, End value = 10, increment = 1.

PSpice will change the voltage source V1 from 1 to 10 in linear increment of 1.

Now click Apply and O.K..

In Schematic choose PSpice → run

A window appear



That window indicates that the simulation is completed; a linear sweep was done on the voltage source V1 from 1 to 10.

Note that the upper window contains the X-axis with V1 varying from 1 to 10 as we set it. Nothing appears in the window, we have to choose what we want to see.

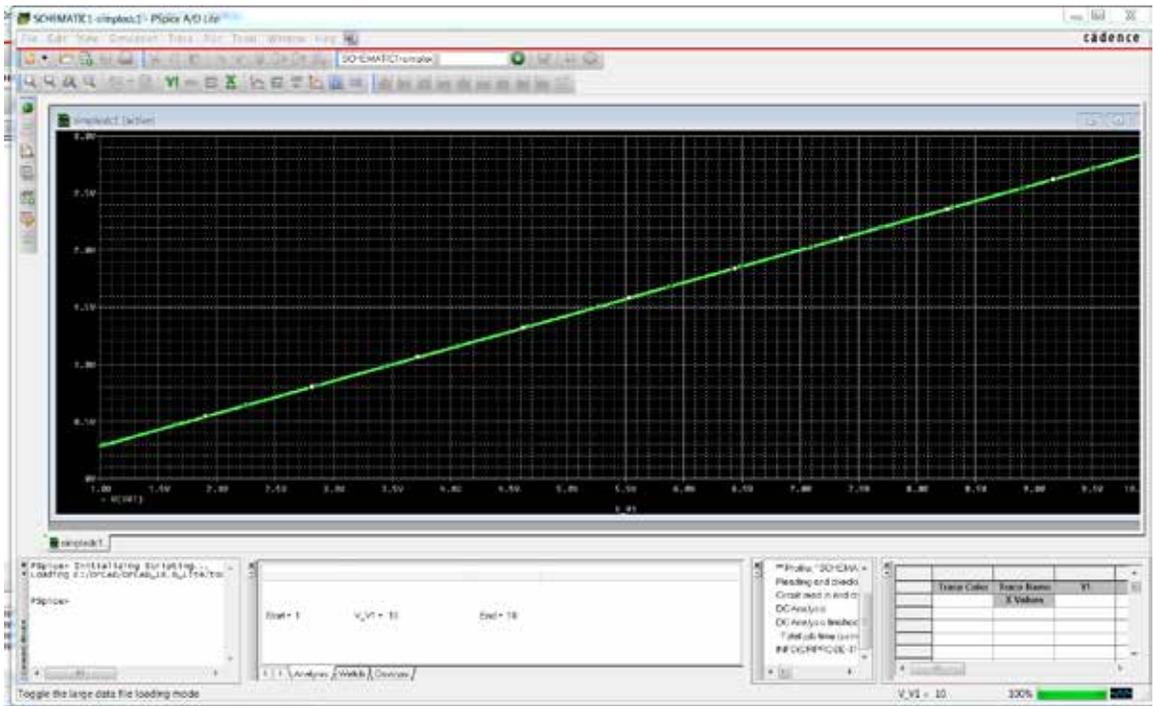
Go to Trace → add trace, a window appears

The right panel include all signals that you can trace. The left panel includes macros and operations.

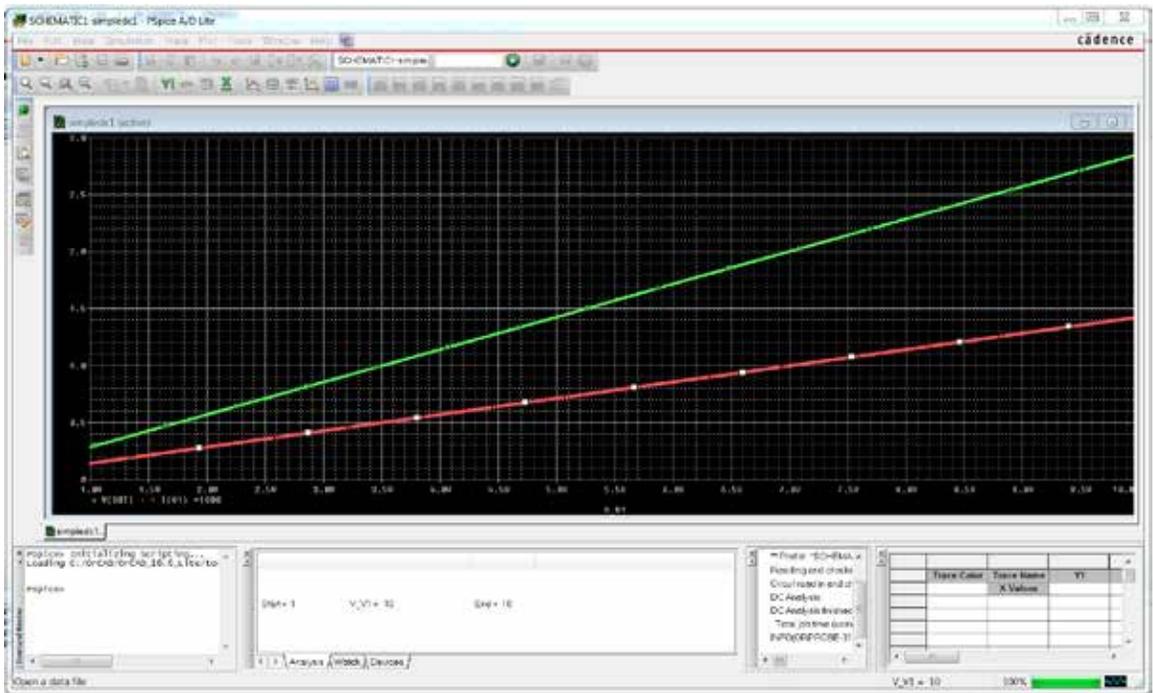
From the right panel, choose V(OUT), note that is the net label we choose for our output voltage V(OUT) appears in the trace expression field, click O.K.

Now the diagram showing the relation between Vout and V1 appears.

In the graph below I made the line thicker to make is easier to see. You can do that by placing the mouse on the line, right click, and choose “Trace property” where you can change the line type, thickness and color.



Again, go to top menu Trace → add trace
 The previous window appears.
 Click on “-” from functions or Macros(or type -, minus).
 Click on I(V1) from simulation output variables
 Click on “*” from functions or macros
 type 1000 in the trace expression box.
 Now the trace expression box contains $-I(V1)*1000$
 Click O.K.
 The following graph appears



The plot shows two graphs, the green one is V(OUT) and the red one is I(V1).
 Now, why did we do $-I(V1)*1000$?

I(V1) is the current flowing in the v1 source, by definition the current in a voltage source is flowing from the positive terminal to the negative terminal. That is a negative quantity (since the actual current is leaving the positive terminal of V1).

That should be O.K., however the value of the current will be negative, and the graph will be resized to show both positive and negative values. We prefer to see the current in the positive direction as we imagine the current will be flowing in the circuit.

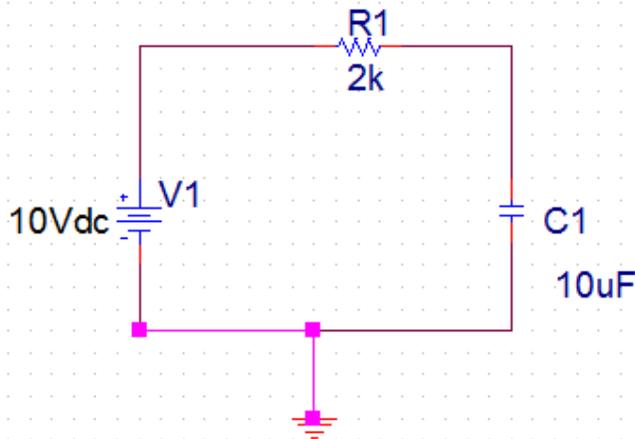
Also, the value of the current will be in the milli ampere range. If we leave it as is it will be a line almost on the X-axis. If we multiply it by 1000 the value will be in the ampere range and will show as a decent size graph as we see in the Figure above.

That concludes our first circuit.

Transient analysis

Now start a new project called RC1 to simulate the step response of a simple RC Circuit. Draw the following circuit. Do not forget to change the name of the ground from GND to 0 or SPICE will not work. Save the file.

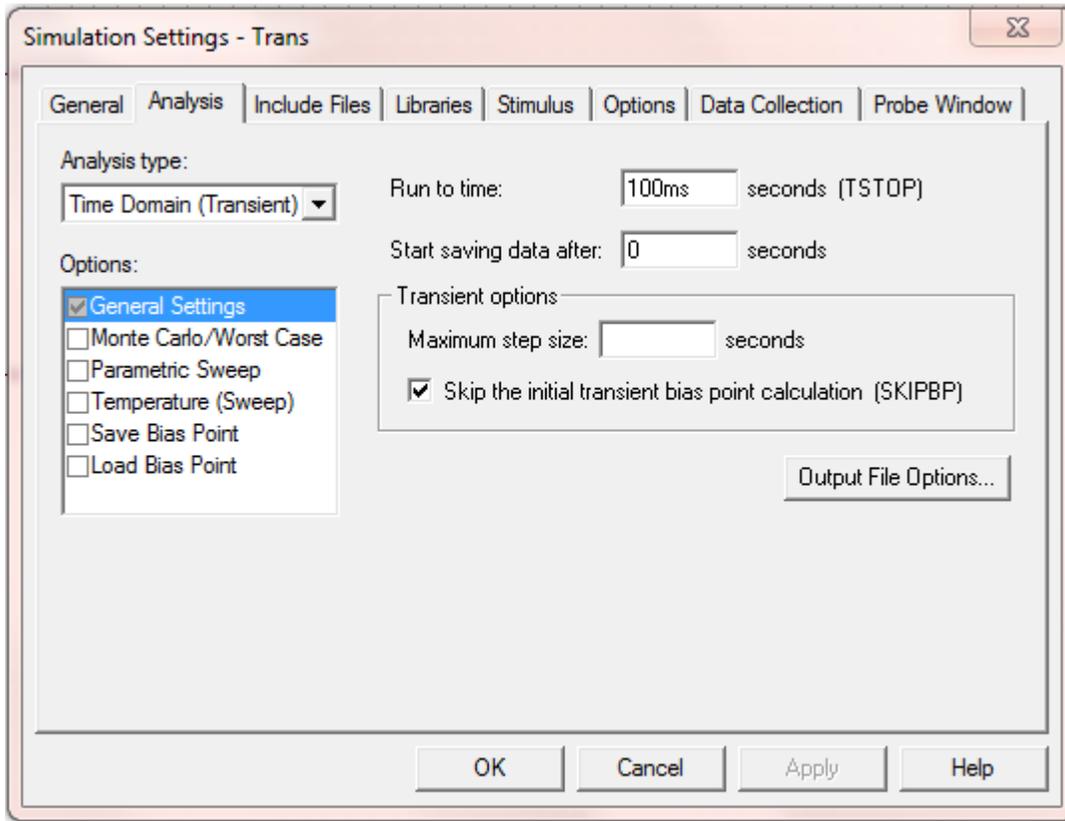
Change the R value to 2K and the capacitance to “10uF”



Now, we have to set the simulation profile.

Top menu PSpice → new simulation profile.

Call it tran, no inheritance from other profiles, then O.K.



In the Analysis tab choose analysis type to be Time domain (transient), general setting
 Choose the run to time 100 msec. (the time constant in the circuit is 20 ms, we will run it five times that). Also start saving data after set it to 0.

Click on “Skip the initial transient bias point calculation”. This is very important if you want to see the transient response. Without it SPICE will calculate the bias point (Capacitor voltage is 10V) and use it to start simulation. In this case you will not see the transient period and the voltage will be a constant 10V. Click Apply and OK.

Go to the top menu PSpice → run

The output window appears as before without any traces.

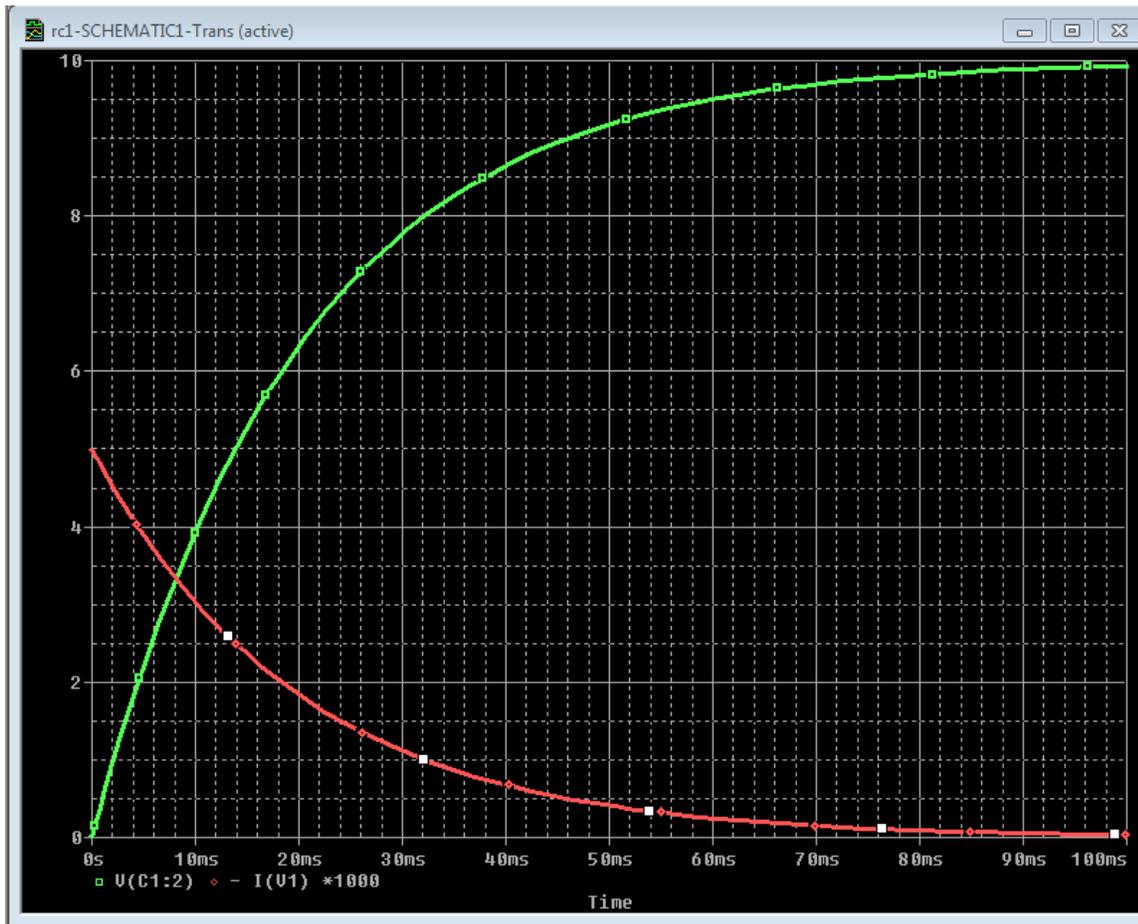
Go to Trace → add trace

The add the following traces

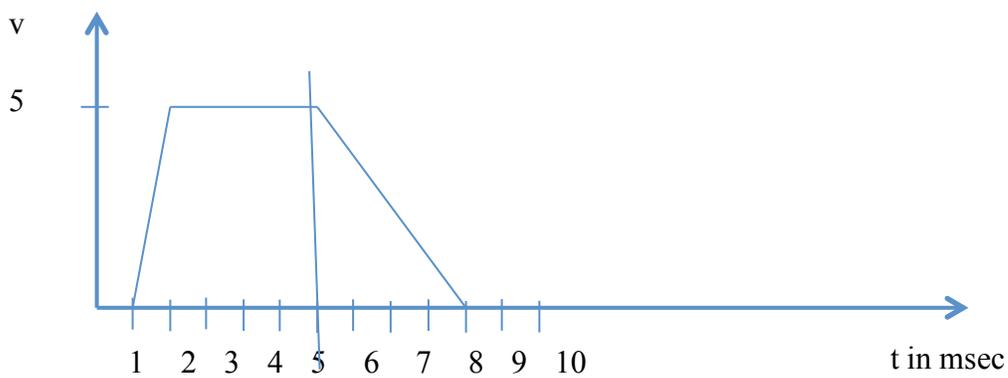
V[C1:2] Note this is the upper terminal of the capacitor, if you choose V[C1:1} that is connected to ground and always 0.

Choose also trace $-I[V1]*1000$ as before

To make the graph lines thicker, right click on the graph curve, choose property and set the thickness as you wish. Here is the output



Now for a little bit more complicated input source. Instead of a DC voltage source, use PWL (piece wise linear) source. A piece wise linear source is defined by the voltage values at different times. For example consider the following source.

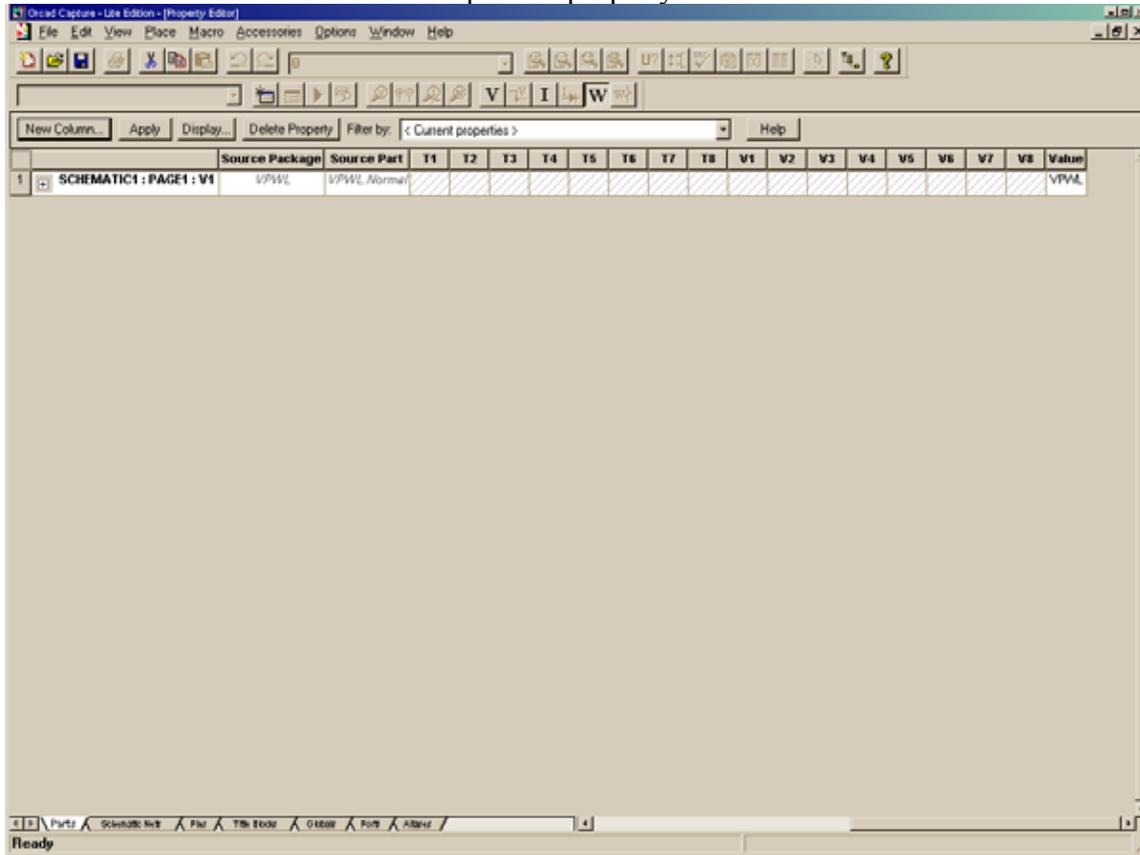


The above waveform is piece linear waveform. It stays at 0 V till 1 msec. Then it, goes linearly from 0 to 5 V for another millisecond. Stays at 5 V for 4 msec. then it goes linearly to 0 in 4 msec.

Replace the DC source in your circuit with a PWL source.
Highlight V1 in the schematic and delete it

Insert part, from library source choose VPWL

Insert it in the schematic
 Connect it to the circuit
 Now double click on the source to open the property editor

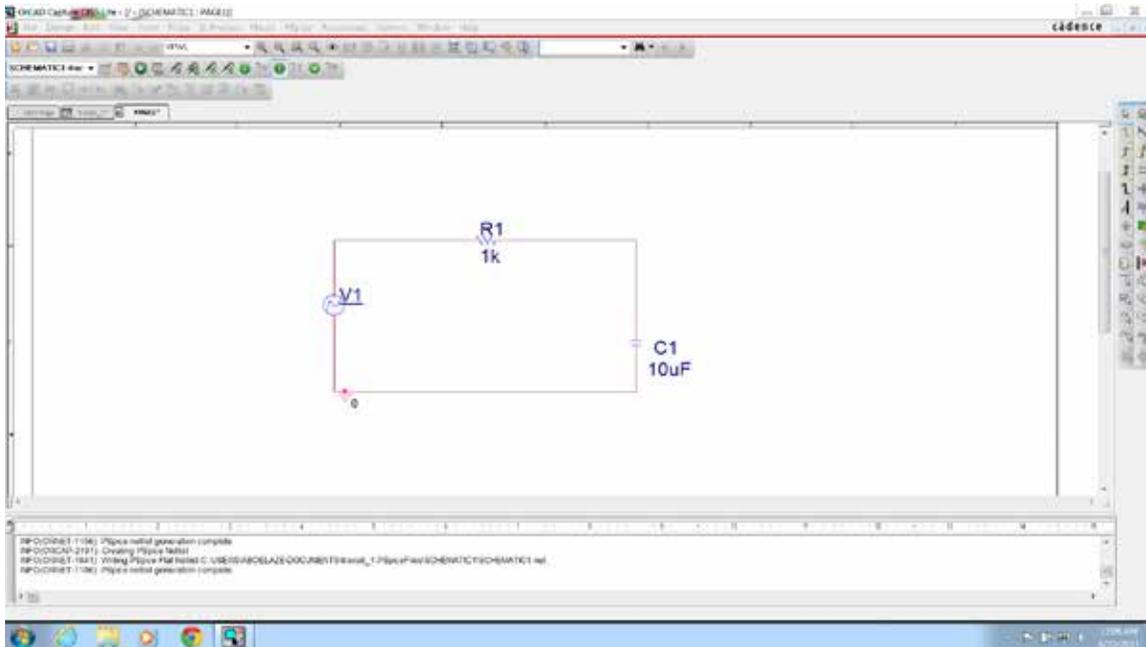


In the Parts tab (bottom part of the window) start inserting columns. Insert columns with

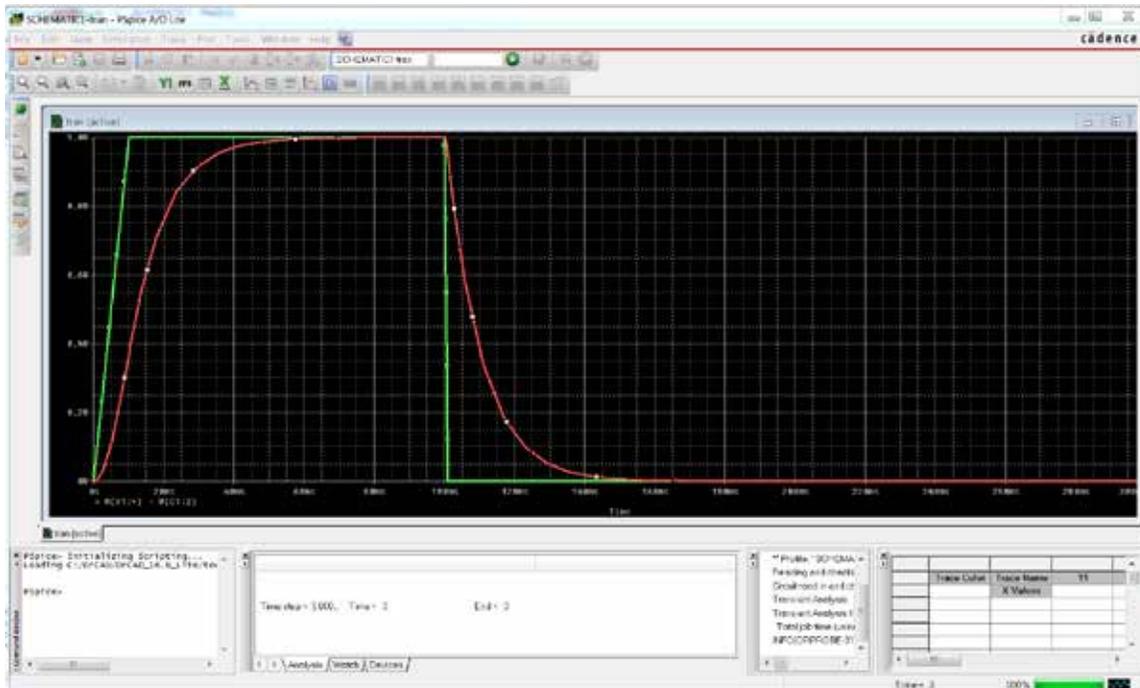
Name	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
Value	0e-3	10e-3	100e-3	101e-3	200e-3	210e-3	220e-3	230e-3	240e-3	250e-3
Name	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
Value	0	1	1	0	0	0	0	0	0	0

This creates a piecewise voltage signal with value of 1V from 10ms-100ms and zero elsewhere.
 Apply and exit.

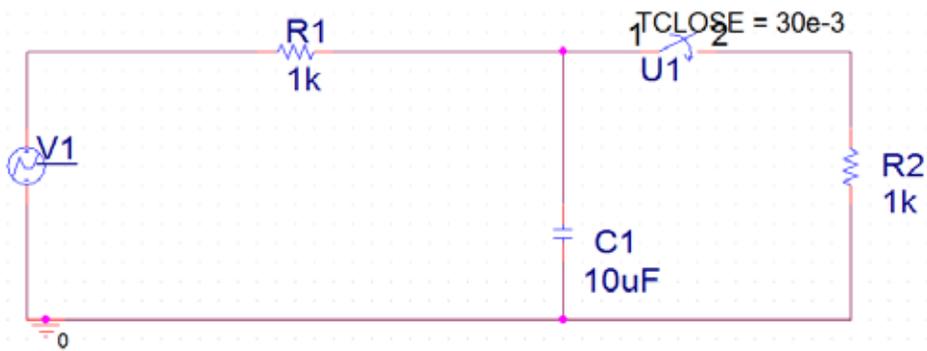
Now, the circuit looks like this



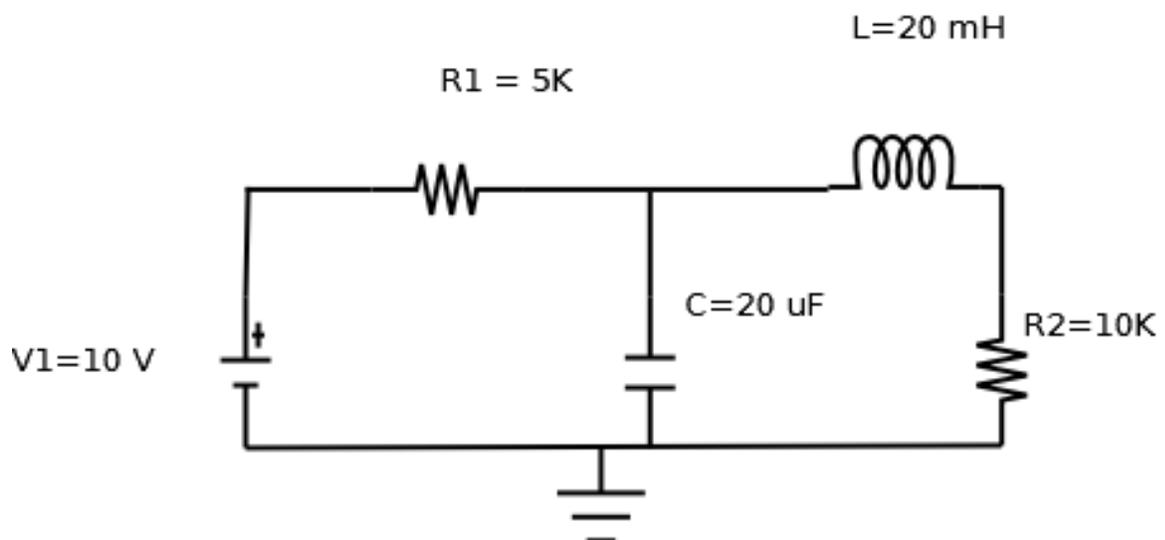
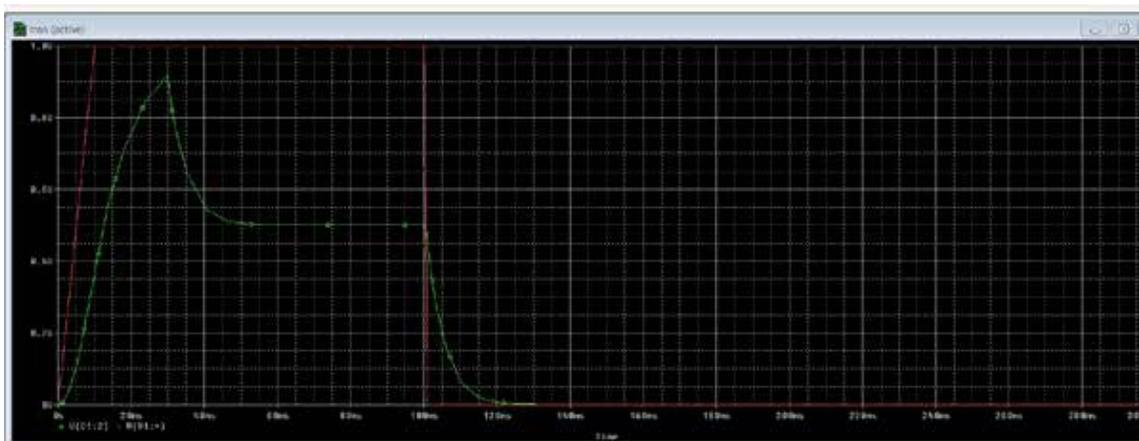
Edit the simulation profile to make the simulator runs for 300 msec. Run the simulator. Using Trace → add trace add V[C1:2] and V[V1:+]. Note that V6 is the name of the PWL source in my simulation. The result should look like that



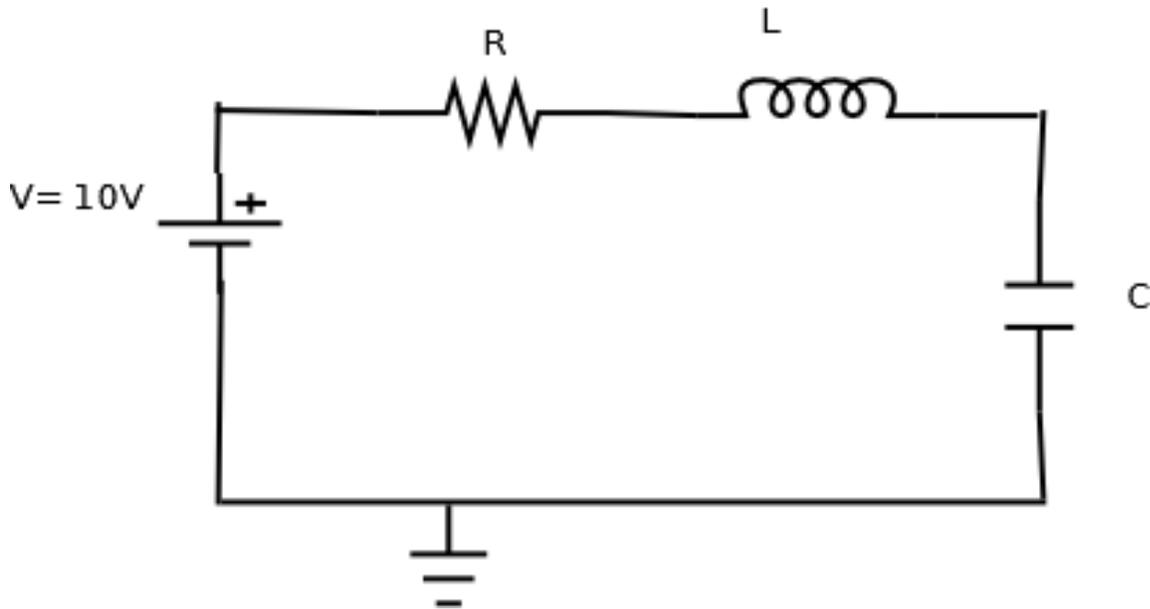
Exercise
Simulate the following circuit



The answer should look like that, not ethat the switch is sw_tClose in the library EVAL and set it to close at 30msec.



INSERT p10



Simulate three cases as follows

1. $R=10\ \Omega$ $C=10\ \mu\text{F}$ $L=0.25\ \text{mH}$
2. $R=2\ \Omega$ same C and L
3. $R=25\ \Omega$ same L and C

Submit the following

The schematic of the circuit (one schematic is for the three cases)

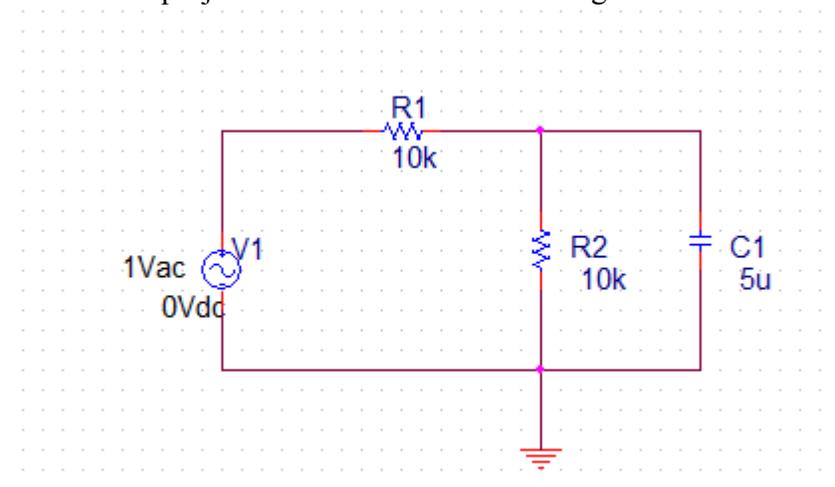
The SPICE code (one code is enough)

Graph showing the voltage across the capacitor and the coil **for every case**.

AC Analysis

In this section, we illustrate how to use SPICE for AC Analysis

Start a new project and construct the following circuit



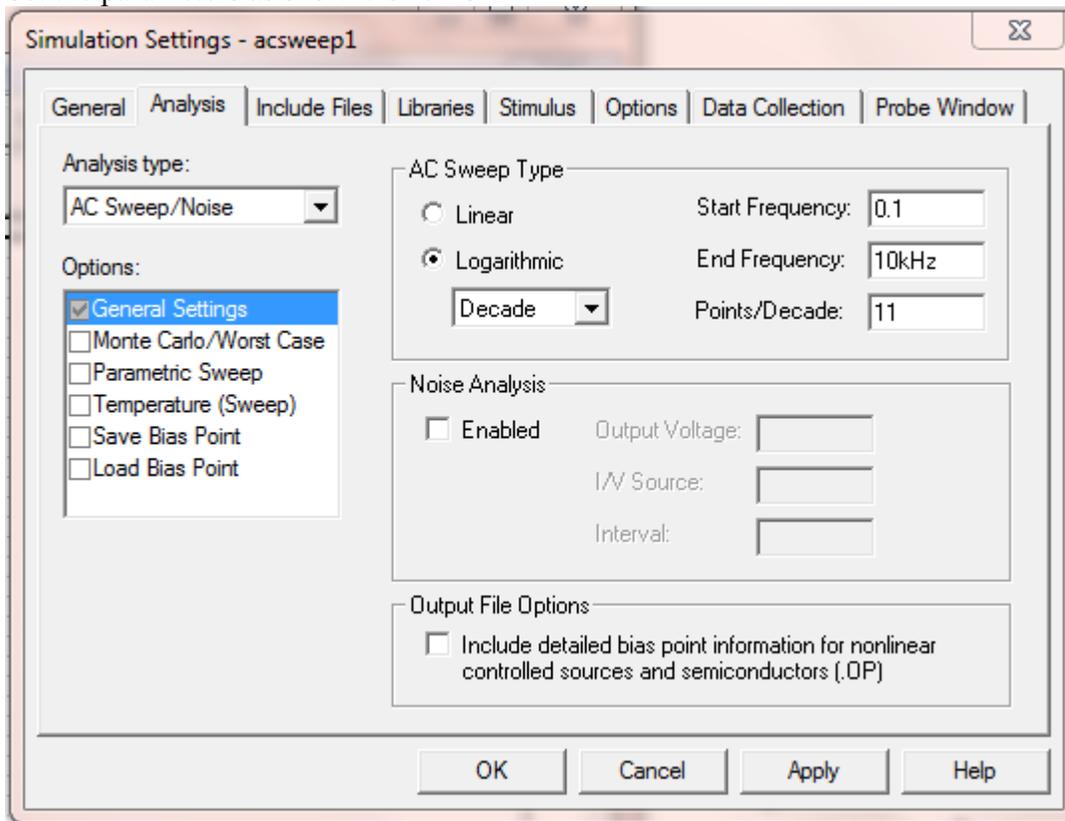
Choose the parameters as shown.

In the Vac source, double click on 1Vac and choose a value of 1 (there is no change here, just to know how to set it).

Save the file,

In capture, top menu PSpice → New simulation profile, choose a suitable name, no inheritance.

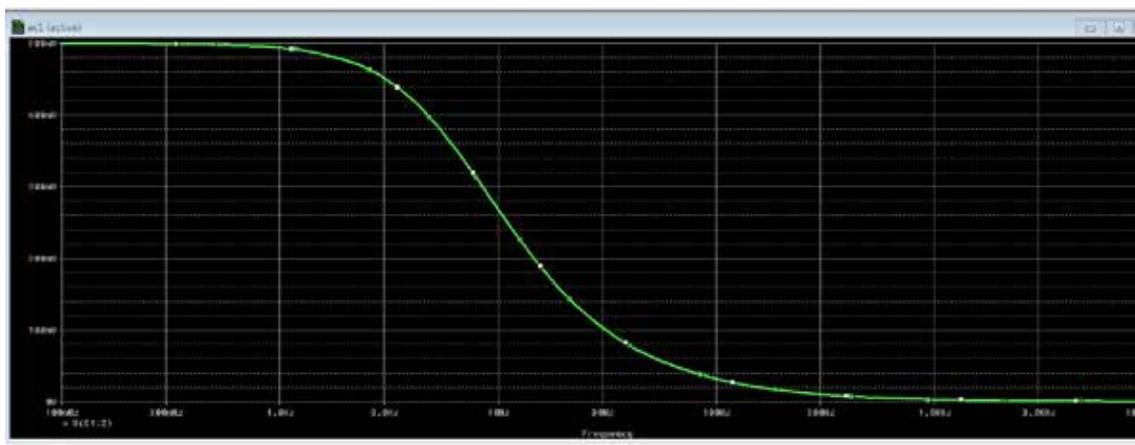
Set the parameters as shown. Click OK



PSpice → Run, the probe widow appears.

Now, top menu Trace → add trace add V(C1:2)

add P!11



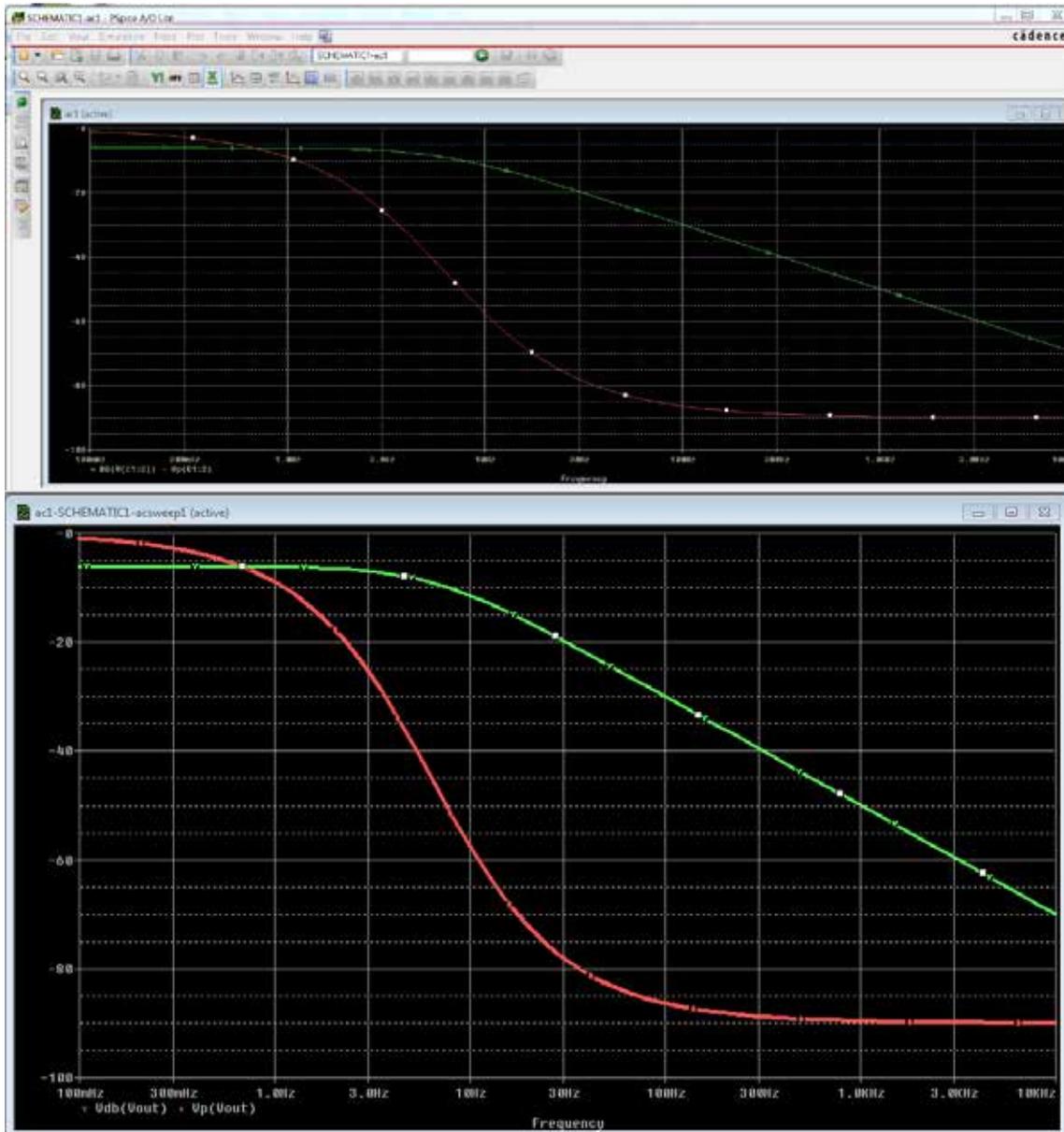
Now we can see Vout vs frequency.

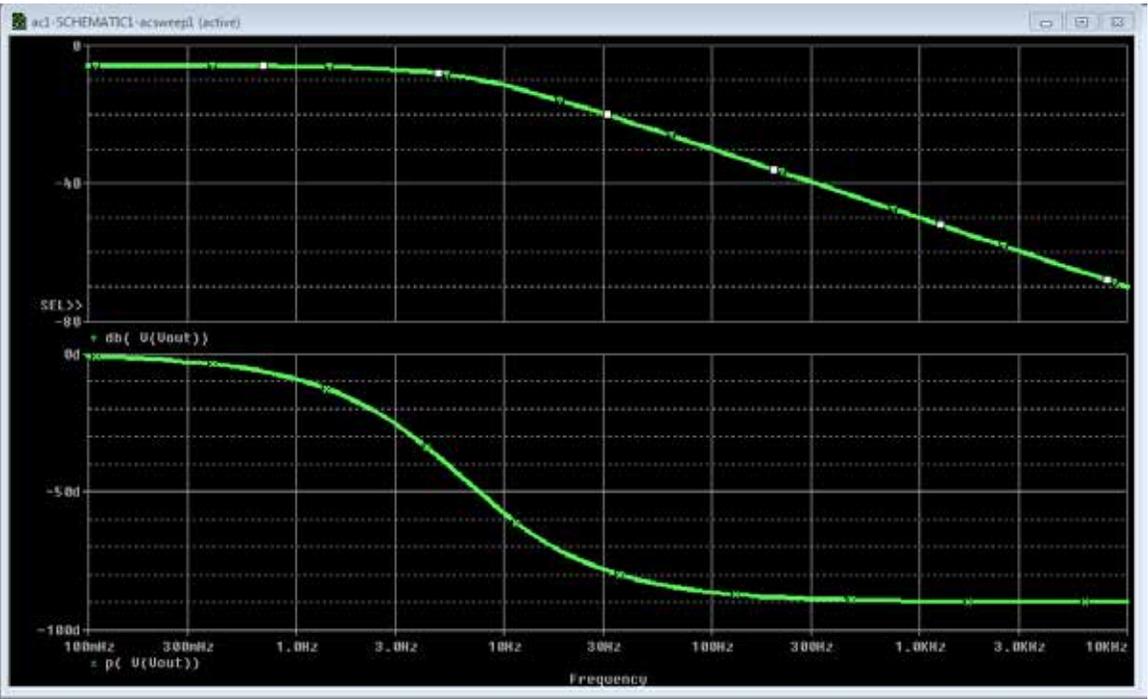
For Bode plot, we would like to plot Vout in dB. In that case Trace → Delete all traces.

Then add a new trace. From the output variable choose Vdb(C1:2) (choose V(out) then insert db after V). Another way to do this is to choose DB() from Functions or macros from the right hand side of the Add Trace window, then click on V(C1:2) to get DB(V(C1:2)).

Add another trace Vp(C1:2) (again insert p after V in V(C1:2)). This is to show the phase of V_{C1} . Or you may use the Functions/macros P(V(C1:2)).

You can also divide the display into 2 plots. From the top menu choose Plot → add plot to window. Another plot will be inserted. The active plot will have “SEL >>” to its left. If you click in the inactive plot, then it becomes active and “SEL >>” moves next to it.





II. SPICE Tutorial

Introduction:

SPICE (Simulation Program with Integrated Circuits Emphasis) is a general-purpose open source analog electric circuit simulator. It started at the University of California Berkeley [1-3] and many implementations exist now, but all are based on the original Berkeley SPICE.

Spice needs a description of the circuit (*netlist*) to run the simulation and commands on what exactly to do. It performs the simulation and displays the results. SPICE can simulate many circuits with elements such as resistors, capacitors, inductors, transmission lines, many different types of transistors, etc.

As mentioned above, there are many versions of SPICE. Although they all share the basic idea, some difference may exist between the versions. In this tutorial we will concentrate on two versions of SPICE, i.e.

1. ngspice: a GNU simulator
2. HSPICE: ORCAD simulator.

If there are differences, it will be mentioned in the context.

To run SPICE, you need to supply a netlist and control commands.

Netlist: the netlist of a circuit is a description of the circuits, the circuit elements and how they are connected. There are two ways to do this. The first is to use SPICE language to describe the circuit. The second is to use a schematic capture program to generate the netlist. A schematic capture program is a GUI interface that lets you draw the circuit, then it transform the circuit you draw into a netlist that could be used by SPICE.

The simulation engine itself can be an interactive program, that reads the netlist, and then you provide commands to the program on what you want to do and display the simulation results (ngspice). Or, the engine could be integrated with the schematic capture, so you draw the circuit, choose the simulation profile, and perform the simulation (HSPICE). It is important that you know both techniques.

This tutorial is a very basic introduction to SPICE. It covers only the parts necessary to use in this course. As you progress in your study, especially if you consider a hardware related career, you will be introduced more and more to SPICE and some other CAD tools.

SPICE Input File Syntax

The SPICE program consists of three parts:

Data Statement: describes the circuit to simulate

Control Statement: tells SPICE what type of analysis to perform

Output statement; specifies the output to be displayed

For example in ngspice, the input file consists of the netlist, that is the first part describing the circuit. Control and output display is performed interactively after the circuit is processed.

A general description of the input file.

- The first line is the title of the circuit. It will be displayed with the results.
- Any line starts with (*) is a comment line, will be ignored by the simulator
- Any line starts with (+) is a continuation line. That means it is a part (continuation) of the previous line.
- Control and command lines starts with a (.).

Netlist

All nodes must have labels (positive integer is compatible with all versions of SPICE).

There must be a ground node (node 0).

All circuit elements must have a text label; the first letter of the label specifies the type of the circuit element. Here is a partial list

V	Independent voltage source
I	Independent current source
R	Resistor
C	Capacitor
L	Inductor
D	Diode
E	Voltage controlled voltage source
G	Voltage controlled current source
H	Current controlled voltage source
F	Current controlled current source

Each element is described by the element name, the two nodes it is connected to, and value.

Example:

```
RES1 1 0 3500
```

This is a resistor labelled RES1 connected between nodes 1 and 0 and have a value of 3500 ohms. There could be more optional fields, but will not cover them now.

The value could be written as
3500, 3.5E3 3.5K 0.0035M, etc.

Where T is tera, G is giga, M is Mega, K is kilo, m is milli, u is micro, n is nano, p is pita, and f is femto. Original SPICE accepts only capital letters, recent versions accepts both small and capital letters.

Example:

```
CIN1 3 0 47PF <IN>
```

The previous statement describes a capacitor connecting nodes 3 and 0 with a value of 47 Pico Farads. <IN> is an optional initial condition. The first node 3 is the positive node when calculating the voltage on the capacitor.

Voltage and current sources:

DC voltage

The DC voltage is defined as: Vname n1 n2 Type value

Example:

Vin n5 n7 DC 10

A voltage source called Vin between nodes n1 (+ve) and n2 (-ve) DC source with 10Volts

Note that the current flows from the positive node to the negative node inside the source.

For sinusoidal sources: Vname n1 n2 SIN(V0 VA f t_D Theta Phase)

where

$$V_{name} = V_0 + V_A e^{-\theta(t-t_D)} \sin[2\pi f(t - t_D) + (Phase/360)]$$

V0 = offset voltage in volt

VA = amplitude in volt

f = frequency in Hz

Theta = damping factor per second

Phase = phase in degrees

Vsin 1 2 SIN(0 5 50)

is a sine wave with 0 offset, magnitude of 5 and frequency of 50 Hz (since T_d, theta and phase are zeros, we did not include them)

Commands (Control)

.OP

.OP stands for operating point, just calculate DC operating point of the circuit. Some versions of SPICE will do this always, some need to be instructed to do so by using the .OP

DC sweep analysis

The command .DC changes the specified independent source (voltage or current) over the interval specified by the user and calculates the OP analysis

Example:

```
.DC vin 0.25 5.0 0.25
```

Changes the value of the independent voltage source vin from 0.25V to 5.0V in increments of 0.25V

AC sweep analysis

In AC sweep analysis, we change the frequency of all AC sources in the circuit according to the control statement

```
.AC increment_type number_of_points start end
```

Examples:

```
.AC DEC 10 100 10KHz
```

Performs an AC sweep, DEC stands for decade, 10 points per decade, starting at 100HZ and ending at 100KHz.

```
.AC OCT 5 100 100KHz
```

Oct for octave (a factor of 8) with 5 points per octave, starting at 100 ending at 100KHz.

```
.AC LIN 50 100 100KHz
```

LIN for linear, with 50 points in total starting from 100Hz to 100KHz.

Transient analysis

```
.TRAN Tstep Tstop [Tstart [ Tmax ] ] [ UIC ]
```

Tstep is the time step in calculating the transient response. Tstop is the end time.

Tstart is an optional for the start time, if not mentioned it is zero.

Tmax is the maximum step size that ngspice uses;

UIC (use initial conditions) is an optional keyword which indicates that the user does not want ngspice to solve for the quiescent operating point before beginning the transient analysis. If this keyword is specified, ngspice uses the values specified in the circuit description.

Output statements

.PRINT and .PLOT are used to print and plot some values. The examples given below will illustrate this point.

The control statement, could be included in a .control .endc block in the netlist file. In ngspice if included in that block, do not use capital letters or "." before the command

Here is a simple example using ngspice, that netlist describes the circuit shown below

Here is a file called ex1.net

Example 1

```
V1 n0 0 DC 10
```

```
R1 n0 n1 10K
```

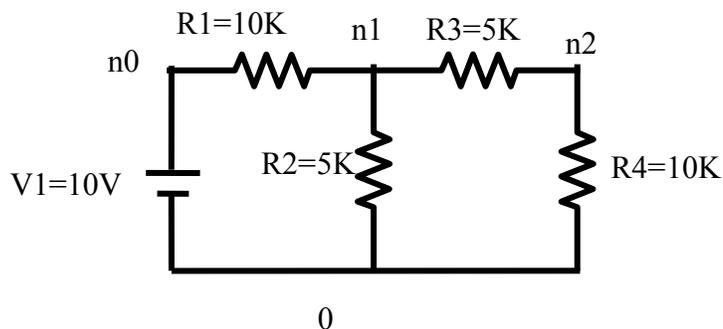
```
R2 n1 0 5K
```

```
R3 n1 n2 5K
```

```
R4 n2 0 10K
```

```
.END
```

Save this program in a file called ex1.net



Start ngspice

```
tigger 136 % /tmp/cse/local/bin/ngspice
*****
** ngspice-24 : Circuit level simulation program
** The U. C. Berkeley CAD Group
** Copyright 1985-1994, Regents of the University of California.

** 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: Tue Jul 17 12:01:33 EDT 2012

*****

ngspice 115 -> source ex1.net

Circuit: example 1

ngspice 116 -> op
Doing analysis at TEMP = 27.000000 and TNOM = 27.000000

No. of Data Rows : 1

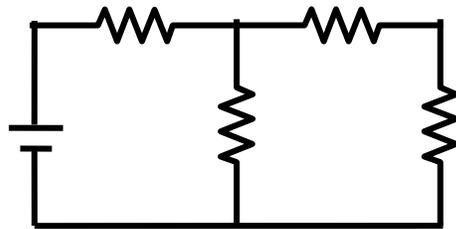
ngspice 117 -> print n1
n1 = 2.727273e+00
ngspice 118 -> print n2
n2 = 1.818182e+00
ngspice 119 -> print i(v1)
i(v1) = -7.27273e-04
ngspice 120 ->
```

Note that in ngspice you can print the currents through a voltage sources only. However you can insert a zero voltage source in series with any load and you can print the current through that voltage source (same current in the load).

The control statement we entered (shown in red) can be included in the netlist file. This is included between .control and .endc statements.
Write this file and store it in ex2.net

Note that the “dc V1 0.0 5.0 0.5” sweeps the voltage source V1 from 0 to 5V with an increment of 0.5 and the print v(2) prints the voltage at v2 for every value of V1.

```
Example 2
V1 1 0 DC 10
R1 1 2 10K
R2 2 0 5K
R3 2 3 5K
R4 3 0 10K
.END
.control
dc v1 0.0 5.0 0.5
print v(2)
.endc
```



Run ngspice. The figure below shows the result

```

tigger 57 % ngspice
*****
** ngspice-24 : Circuit level simulation program
** The U. C. Berkeley CAD Group
** Copyright 1985-1994, Regents of the University of California.
** 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: Thu Sep 6 12:00:52 EDT 2012
*****
ngspice 313 -> source ex2.net

Circuit: example 2

Doing analysis at TEMP = 27.000000 and TNOM = 27.000000

No. of Data Rows : 11
example 2
DC transfer characteristic Fri Sep 14 10:00:35 2012
-----
Index  v-sweep      v(2)
-----
0      0.000000e+00 0.000000e+00
1      5.000000e-01 1.363636e-01
2      1.000000e+00 2.727273e-01
3      1.500000e+00 4.090909e-01
4      2.000000e+00 5.454545e-01
5      2.500000e+00 6.818182e-01
6      3.000000e+00 8.181818e-01
7      3.500000e+00 9.545455e-01
8      4.000000e+00 1.090909e+00
9      4.500000e+00 1.227273e+00
10     5.000000e+00 1.363636e+00
ngspice 314 ->

```

Next, we will simulate an RC circuit. Write the following file and store it in a file named ex_rc.net. Run ngspice, source the file and run it.

Simple example for spice3 tutorial

*Sources

V1 1 0 DC 10V

*Network Elements

R1 1 2 500

C1 2 3 1uF IC=2V

R2 3 0 500

.END

.control

run

tran 0.1ms 5ms UIC

plot v(2,3)

.endc

Screen capture of ngspice

ngspice 314 -> source ex5.net

Circuit: simple example for spice3 tutorial

ngspice 315 -> run

Doing analysis at TEMP = 27.000000 and TNOM = 27.000000

ngspice 316 -> source ex_rc.net

Circuit: simple example for spice3 tutorial

Doing analysis at TEMP = 27.000000 and TNOM = 27.000000

Doing analysis at TEMP = 27.000000 and TNOM = 27.000000

Initial Transient Solution

Node	Voltage
1	0
2	0
3	0
v1#branch	0

No. of Data Rows : 59

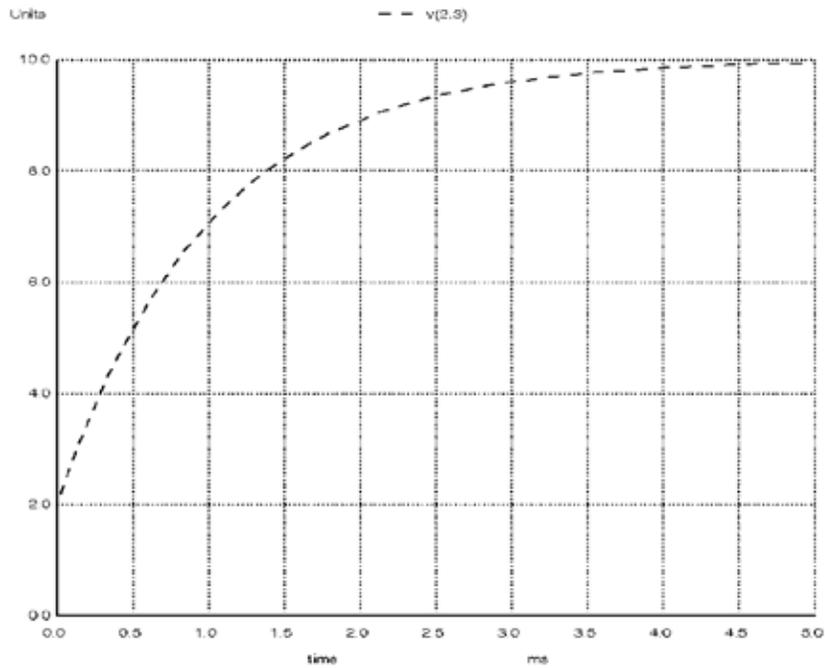
Warning: Missing charsets in String to FontSet conversion

Warning: Unable to load any usable fontset

ngspice 317 ->

The file "/tmp/hc600" may be printed on a postscript printer.

ngspice 317 ->



Reference

4. Nagel, L. W, and Pederson, D. O., *SPICE (Simulation Program with Integrated Circuit Emphasis)*, Memorandum No. ERL-M382, University of California, Berkeley, Apr. 1973.
5. Nagel, Laurence W., *SPICE2: A Computer Program to Simulate Semiconductor Circuits*, Memorandum No. ERL-M520, University of California, Berkeley, May 1975.
6. <http://ngspice.sourceforge.net/docs.html>.