Department of Electrical and Electronic Engineering
Imperial College London

ELEC40002 – Analysis and Design of Circuits

An Introduction to LTspice

Paul D. Mitcheson, Dave C. Yates, Andrew Holmes and Nunzio Pucci

Course homepage: http://bb.imperial.ac.uk

Paul D. Mitcheson
paul.mitcheson@imperial.ac.uk
Room 1112, EEE

Imperial College London

Contents

1. What is SPICE?

SPICE (Simulation Program with Integrated Circuit Emphasis) is a well-known circuit simulator computer program which began with a mainframe program written at UC Berkeley by Lawrence Nagel (and others) during his PhD. The SPICE solver or "engine" (the core part of the program *i.e.* the part that solves the circuit equations) is free software but it is command line driven and is thus regarded as difficult to use. Consequently, several commercial versions (mostly based on the 2G7 release of the Berkeley program) are available which provide a graphical interface and up to date component libraries (models of commercially available semiconductor components provided by the semiconductor manufacturers). There are also some free versions of SPICE with a graphical user interface and the best known is LTSpice.

LTspice has a schematic GUI input and an oscilloscope-like output interface. The schematic input lets users draw circuits to be simulated on the screen and the oscilloscope-like output allows users to interactively plot traces after a simulation run. In between this input and output is what is basically the original SPICE solver (written by Nagel *et al*.) that reads the circuit definitions from "netlists" (a text file which describes the circuit by specifying components, their values and their interconnections) and runs the circuit simulation. In days gone by, users specified circuits by writing the netlist by hand in a text editor and the netlist file was effectively the input to the simulator. LTspice (and the other commercial SPICE implementations) create the netlist automatically from the circuit schematic drawn by the user (this process is called *schematic capture*).

An overview of the basic steps needed to run a simulation in LTspice is shown below (non-shaded bullet points are dealt with automatically by the LTspice program and most of the time the user does not need to concern themselves with these stages as (s)he would have done in the 1970s.):

- Draw schematic of circuit
- Set up Simulation
- Run Simulation
    - Build netlist from schematic
    - Run SPICE solver on the netlist
    - Text files written as output
- Plot output graphs

2. Using the software

This tutorial is written for version XVII of LTspice, downloadable at:
https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html

2.1. The Old-Fashioned Way

Before we learn how to use LTspice's graphical user interfaces we will first use the software as electrical engineers used to use it in the 1970s. You will find this exercise useful – it allows you to gain an understanding of what the software is actually doing underneath the graphical user interface. In the future when you start simulating large circuits (using the GUI) you may find it useful to be able to read and write netlists and this simple example guides you through writing a netlist.

Let's try to simulate the following circuit by manually creating the SPICE input file. We will perform the simplest type of simulation that SPICE can do – a **DC bias point simulation**. This is a simulation which works out the steady voltages and currents in a circuit when voltage and current sources in the circuit have constant values. This is essentially doing nodal analysis on a computer, rather than by hand as you have been taught in the lectures.

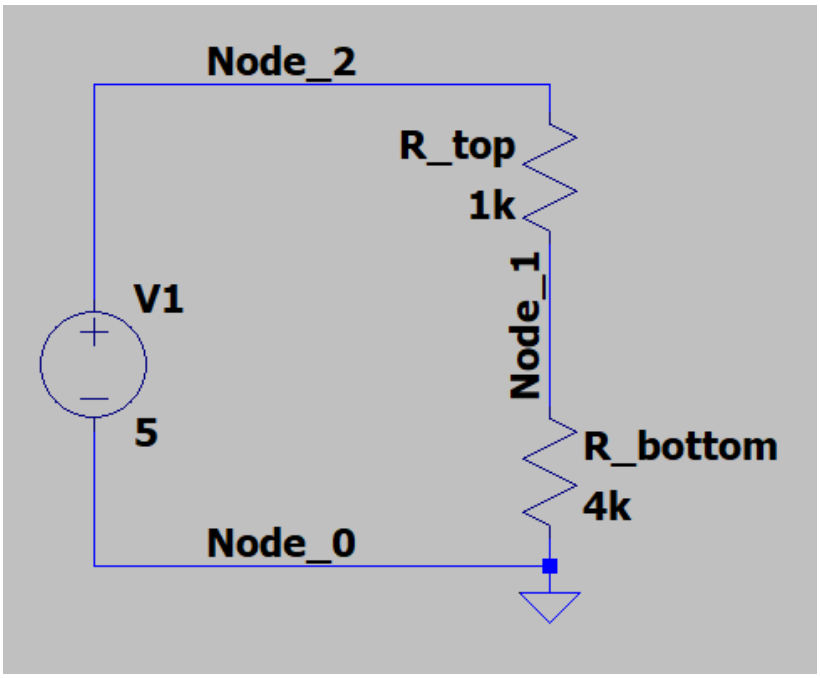


Figure 1: a first circuit

First we need to write a **netlist** for this circuit – *i.e.* a description of the circuit in a format that SPICE can understand. This is done by defining node numbers at the interconnections of components and, then, for each component, describing which nodes it connects to. We are free to define the node numbers however we like (as long as each node has a unique number (a positive integer) and that the ground node is labelled "0"). Let's use the node numbers as labelled on the circuit diagram above in this example. The following netlist describes the circuit:

```
VV1          N0002    0       5V
RR_top       N0001    N0002   1k
RR_Bottom    0        N0001   4k
```

Component type | Component name | Node to which pin 1 of the component connects | Node to which pin 2 of the component connects | Component properties

Note that:

- Each line describes a component in the circuit.
- The first letter in each line defines the type of each component. R is resistor and V is a DC voltage source.
- The other letters in the first column (after the first letter) form the name of the component.
- The next two columns describe the nodes that the component is connected to.
- The ground node is a special node in SPICE, and is just referred to as "0".
- The last column describes the properties of the component. So for the resistor, the final column is the value of the resistance, in ohms. For the voltage source it is the voltage, in volts.

Try to understand the format of this simple netlist before moving on.

This netlist is not sufficient by itself to run a simulation. Some other statements are needed in the SPICE input file. The first line in any SPICE file is the title – and must be present (although the title can be anything we like, such as "My first SPICE simulation"). We also need to tell SPICE what sort of simulation we want it to run. In this case we want to perform a DC analysis (also known as the operating point) – and so we must add the "`.OP`" statement to the file. We must also add a "`.END`" statement so that SPICE knows when it has read the whole file. The completed SPICE input file for performing a DC bias point simulation on this circuit therefore looks like this:

```
My first SPICE simulation

.OP

VV1                 N0002     0         5V
RR_top              N0001     N0002     1k
RR_Bottom           0         N0001     4k

.END
```

Now we can try to run SPICE on this file.

Open Windows notepad (or any plain text editor of your choosing – do not use rich text format) and type in the above netlist. You can separate the columns with tabs or spaces. Save the file as first_spice.cir into your home directory into a suitable directory (such as h:\spice_tutorial).
Now you can use **right click** | **Open with…** | **SPICE Simulator w/ Schematic Capture**.

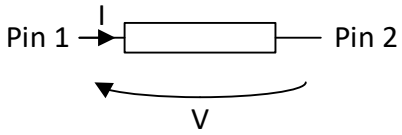
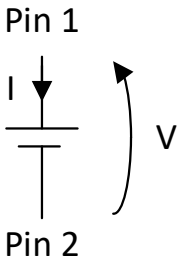
Two windows will pop up, showing the netlist and the output of the simulation. You should be able to find a line within the file which tells you the voltage at nodes 1 and 2 of the circuit. Are they what you expect from this potential divider? (Are the signs right? If you are not sure, check the next section of the notes on sign conventions…)

You have now run your first SPICE simulation and have some idea of how circuits are described in the SPICE netlist format. This day is almost as important as the day that you learnt Newton's second law!

2.2. A note on Sign Conventions in SPICE

SPICE uses the passive sign convention, and hence calculation of power a resistor is positive, and a voltage source supply power to a circuit will have negative power (note that a source which is absorbing energy – like charging a battery – will have positive power).

The convention for voltage and current maps on to pin numbering as follows:

	
R, L, C conventions (resistor shown)	Source conventions (voltage source shown)

When you write a netlist, and say what nodes each component is connected to, you specify the connection in ascending order of pin numbers, i.e.:

```
RR_top              N0001     N0002     1k
```

Means that a component of type resistor (R), labelled R_top, has pin 1 connected to circuit node 1 and pin 2 connected to circuit node 2, and is of value 1 kΩ.

2.3. Using the GUI

Now that we have run a first simulation by writing the SPICE input file by hand we shall start using the software using the modern LTspice GUI interface. This is the recommended way of using the software.

To create a new project you must start the **LTspice** application**.**

To create a new project use the menu shortcut **File | New Schematic** (or just **CRTL+N**). The New Project dialogue will pop up. Choose a name and save the schematic in the desired folder (you can still use the previously created h:\spice_tutorial): **File | Save As**

You are now ready to start editing your circuit. Before we do this we will take a look at the types of simulation we can run in LTspice.

2.4. Types of Simulation

SPICE is capable of doing several types of circuit simulation. The three below are the most common:

- Bias point (or DC) analysis (find the steady state condition of a circuit)
- Transient analysis (a time domain simulation – *i.e.* to show how the voltages and currents in the circuit behave over time)
- AC sweep (look at the performance of the circuit at different input frequencies)

We shall look at these three types of analysis in turn.

2.5. A First Circuit – Simple Bias point (DC) analysis

In this section you will create the same circuit that you simulated when you wrote your own netlist file (a potential divider) but this time you will use the **LTspice** GUI instead of writing the netlist yourself. The circuit is shown in figure 2. We first need to draw the circuit.

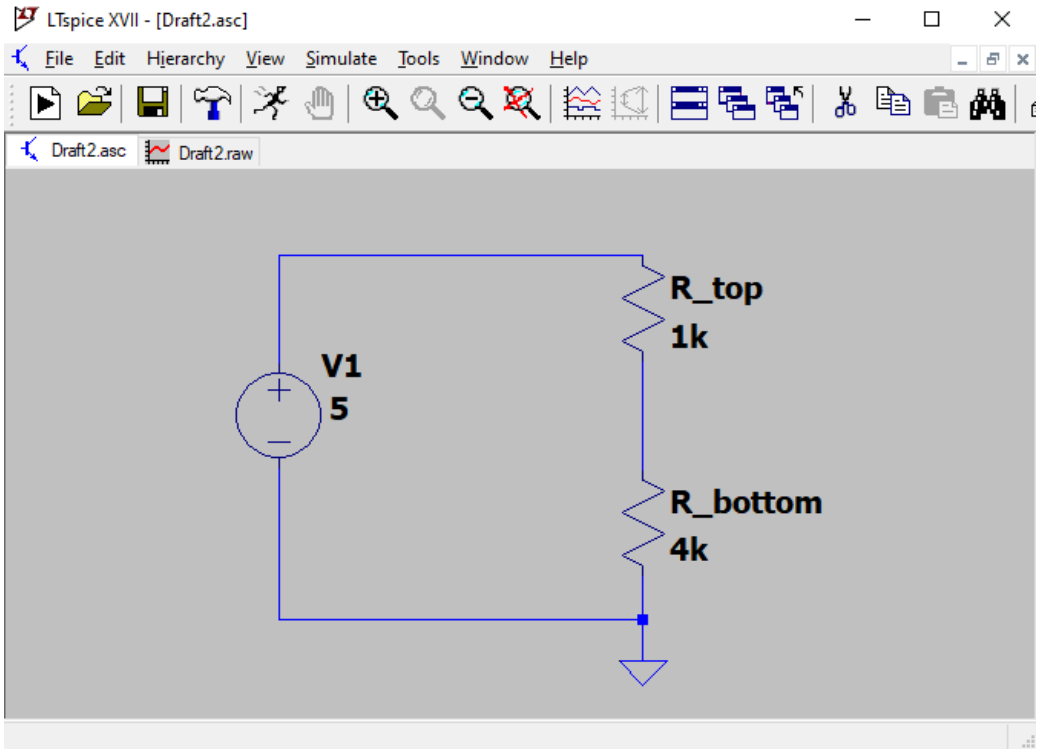
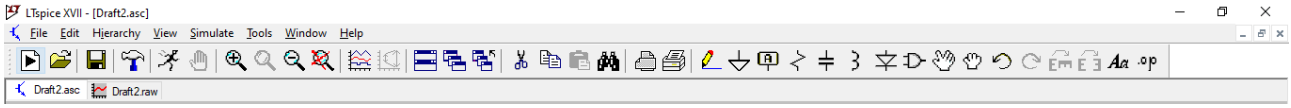


Figure 2 - A first circuit

2.5.1. Placing a Part

We will place the first component on our circuit, a resistor.

1. You can find common components in the components toolbar shown below, or choose a specific one from the "Components" button:



2. Choose the **resistor** from the toolbar (or use **R** as a shortcut).
3. The dialogue will disappear and the mouse pointer will be replaced with a resistor symbol.
4. Click anywhere on the grid and the part will be placed.
5. Press the escape key, **Esc**, and the mouse pointer will turn back from a resistor to a normal mouse pointer.
6. Move new component using the Move button (to ignore any previous connections) or the Drag button (to keep the component's pins connected to any existing node). While you move the component you can rotate it or flip it using **CTRL+R** and **CRTL+E**).
7. The **F6** key lets you copy and paste components and **F5** lets you delete them.
8. Note that the UNDO button is not CTRL+Z as many might have already noticed. The shortcut is instead **F9** (**SHIFT+F9** for REDO)
9. You can find other useful shortcuts under Edit.

2.5.2. Specifying a Component Name and Value

1. Right-click on the part name (R1 by default) and change it to something more meaningful. **R_bottom,** for example.
2. Right-click on the component value (1k by default) and change it to **4k.**

2.5.3. Completing The Circuit

1. Add a second resistor and ensure its value is 1K. Call it **R_top**.
2. Add a DC voltage source (component **voltage** from the Components button). You can now define this source by right-clicking on the symbol to open the property editor and set the DC voltage to **5**.
3. Join the components up using the wire tool (**F3**). Click at the point where you want to start a wire and left click with the mouse when you need to turn a corner. To end the wire press ESC.
4. A least one node in the circuit must be declared as ground and all other nodes must have a path to it. Click (or use **G**) to place your ground. Your circuit should look as below – and is ready to simulate.

2.5.4. Preparing for a simulation

1. Choose **Simulate | Edit Simulation Cmd**
2. The simulation options dialogue box will open.
3. Change the analysis type to: **DC op pnt**.
4. Click OK and position the generated line **.op V1** anywhere on the schematic (preferably in a place where you can find it).

2.5.5. Running a Simulation

1. Click on the Run icon to start the simulation. The circuit will be checked for major errors (unconnected components and missing values). The SPICE input file will then be written (including the netlist and other simulation statements of which you are now familiar with) and a window will open similarly to section 2.1.
2. Yu can now go back to the schematic window and update the results of the DC bias point simulation by right-clicking the node you are interested in and choosing **Place .op Data Label**.

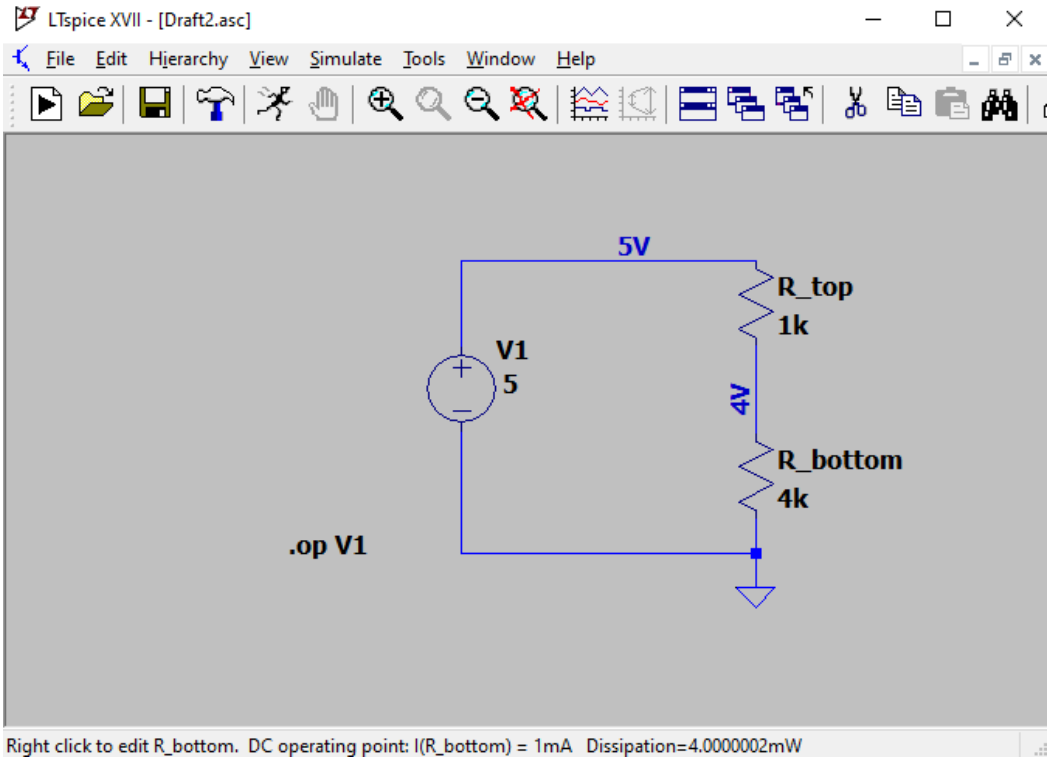


Figure 3 - Results of bias point simulation

You can also see steady state currents, and power dissipations by placing the mouse on top of a component (the result will be displayed at the bottom of your GUI, similarly to figure 3).

2.5.6. Dealing with Errors

You **will** encounter errors when using LTspice. The two main types of error are:

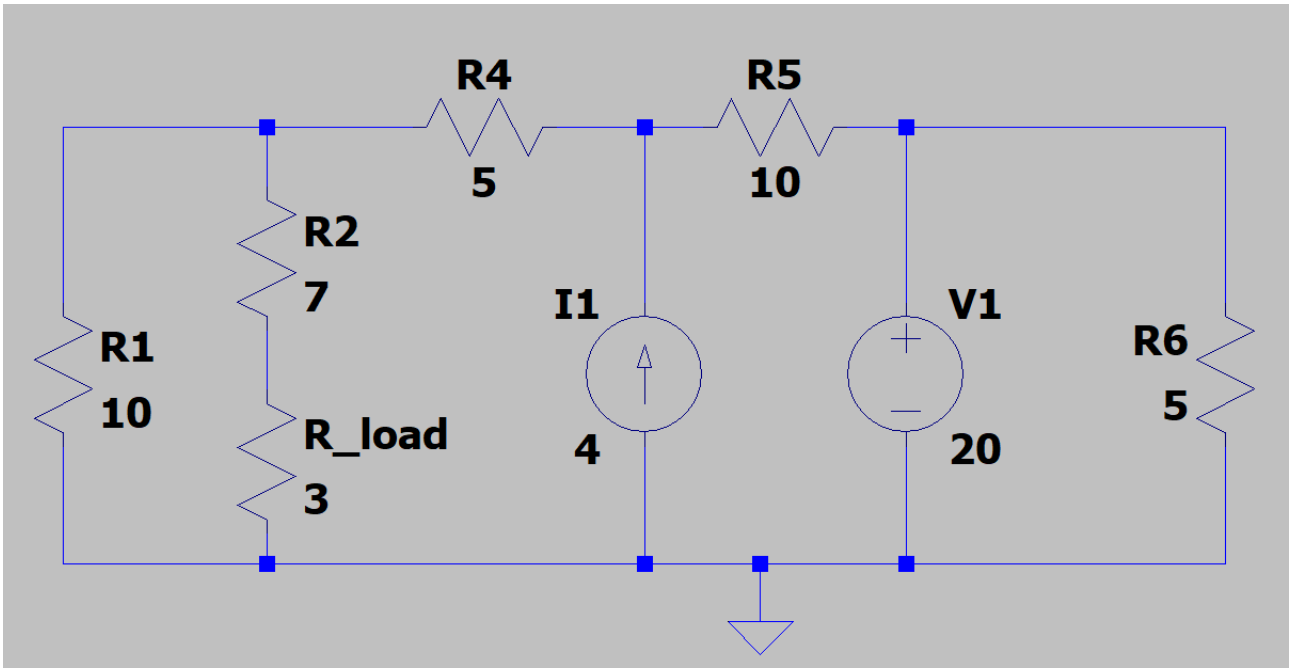
1. User error – most likely a netlist error or simulation setup. This will occur because something is missing from the circuit, or a wire which you think is connected is actually not connected (try dragging components to check that the wires at their pins drag with them to check for connections). You will also get a netlist error if a component value is not specified.
2. A simulation error – probably a simulation convergence error. This often occurs if SPICE is running a transient simulation (but can occur also on a simple DC bias point analysis) and on calculating the circuit voltages and currents for the next time step (using the voltages and currents from the previous time step as starting points) it cannot find a solution. SPICE uses Newton-Raphson to numerically solve the circuit equations and the Newton-Raphson algorithm is not guaranteed to converge to a solution. This type of error will be uncommon for basic use of SPICE (small, low frequency circuits) but it may rear its ugly head if you try and simulate more complex circuits. Be aware of this point for the future.

Try forcing an error (so you know what to expect when you come across a real error) by deleting the ground connection in the current circuit and rerunning the simulation. You should see an error message in the SPICE window warning that nodes are floating (because the ground node is no longer defined).

A convergence error will show up in the LTspice window during the simulation run (after the netlist has been sucesfully generated). Ways of achieving convergence are to reduce the minimum time step, reduce the maximum frequencies in the circuit (which includes reducing the rise and fall time of square waves) and reduce the impedances of each node to ground. This is the sort of skill which is best learnt by doing (if you chose to study lecture courses which use SPICE later in your undergraduate programme you will learn this skill). General tips on how to avoid convergence errors can be found at http://ltwiki.org/index.php?title=Convergence_problems%3F

2.5.7. Exercise: Superposition

Let's now try to link what you are doing in lectures with the simulator, by investigating superposition as a way to solve circuits. Open LTspice and draw the following circuit schematic. (You can find the current source in the components section as **current**).



Firstly, using superposition of the two sources, hand calculate the expected current in, and voltage across, R_load for the circuit above.

Now, let's verify this using the simulator.

We are interested in three different measurements taken under DC bias point simulation type:

- Current through R_load
- Voltage drop across R_load
- Power dissipated in R_load

Again, using the principle of superposition simulate current through R_load and voltage drop across R_load. You will need to run 2 simulations to do this.

What can you say about the DC voltage across, the current through, and power in R_load?

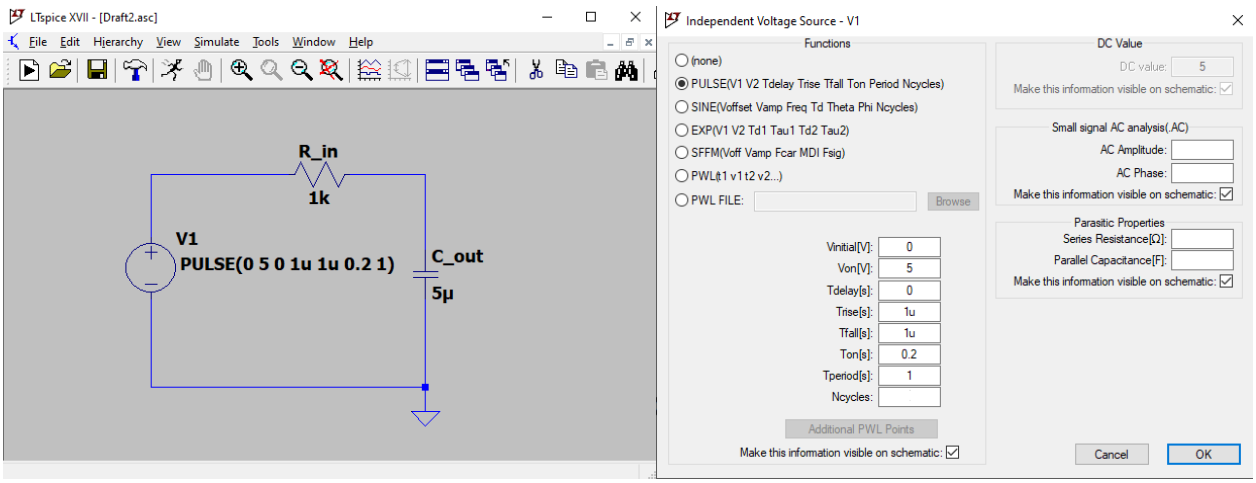
Is the power through R_load the value you expected? Does this obey superposition? Explain what you find.

Check your answers by setting both sources to be active at the same time and simulate the circuit.

2.6. Transient Analysis

We will now create our first transient simulation in LTspice. This is a time domain simulation (i.e. we will simulate the operation of a circuit over, say 10s, for a time varying input). We will then use **.raw** file to plot oscilloscope-like traces of voltages and currents around the circuit once the simulation is complete.

Create a new project and give it a suitable name. Draw out the following circuit into the schematic:



a) **b)**

Figure 4 - First transient simulation

Insert the capacitor component (the shortcut is **C**) and the voltage source similarly to the previous section. To make the DC voltage source into an AC source, right click on the inserted symbol in the schematic and select **Advanced.** You will be prompted to the same window displayed in Figure 4 b).

Make sure you have set all the attributes of the voltage source as shown in the above diagram. These attributes specify the two different voltages that the source will switch between, the delay before the source starts switching, the rise and fall time, the time the source spends in state **Von**, and the time period.

Now edit the simulation command: **Simulate | Edit Simulation Cmd | Transient**.

1. Enter 10s in the "Stop time" box.
2. Click on the "Skip initial operating point solution" checkbox. This allows us to set our own initial voltage for capacitors and initial current for inductors (default values are zero). Without this option SPICE will run a bias point analysis in an attempt to find the steady state conditions to use as a starting point for the transient analysis. This isn't always appropriate.
3. Press OK and place the generated line in a visible location. Run the simulation .
4. Click on the node you want to analyse to show the voltage waveform or on a component pin to display the current going through it. **ALT+Left Click** allows you to monitor power through a component.
5. Alternatively you can use **Plot Settings** to add traces (**CTRL+A**) or edit your plot (for instance you can add a grid or set the y-axis to logarithmic through the **Manual Limits** prompt). Another useful functions accessible through **Plot Settings** is **Add Plot Plane**, allowing you to display more traces without mixing them up in the same graph.

The **Add Traces to Plot** option allows you to display expressions of the different waveforms. For example we can find the voltage drop across R_in by typing V(n001)-V(n002) as shown below:

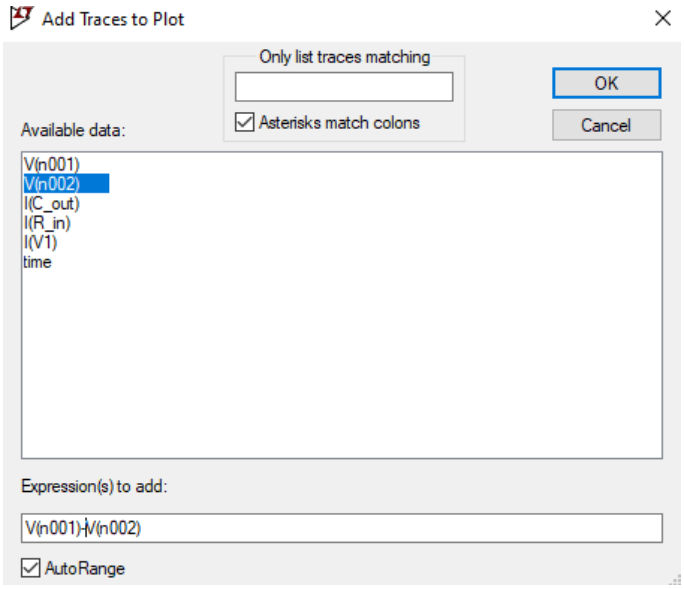


Figure 5 – Display Waveforms Expressions

Having run the simulation you can check if you find your results reasonable. Choose **Trace | Add Trace** from the menu bar. The add traces dialogue box will open.

The **.raw** window should now look something like the screen shot below:

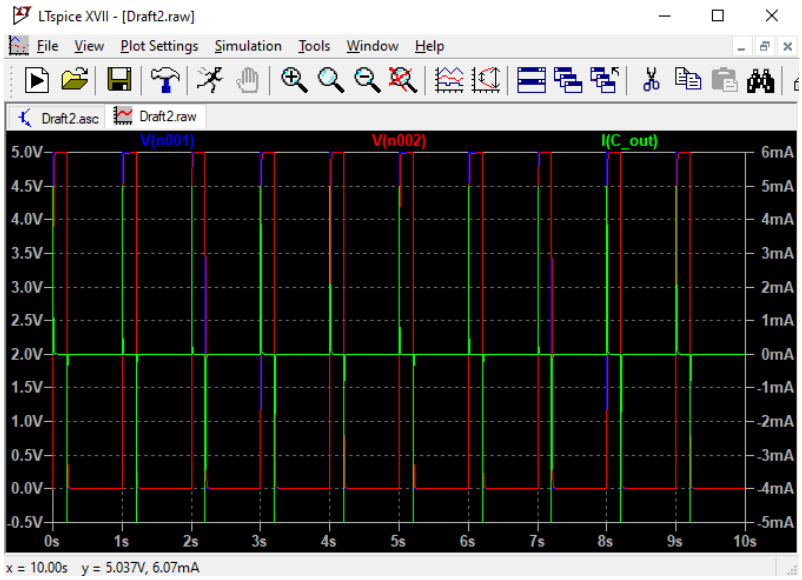


Figure 6 - Results of transient simulation in probe

You should see that the capacitor charges and discharges quickly with respect to the pulse period. The detail can be observed by changing the x-axis scaling (right-click on the x-axis to change the displayed time window). Zoom in and look at the detail of the charging and discharging of the capacitor. You can zoom to specific rectangular sections of your graph by selecting them with left-click.

If ever you plot a trace and find that you are not getting a trace that you expect, then ensure you are plotting the voltage at the correct pin of a component.

A good practice for identifying the correct nodes voltages is to label your nets before running the simulation. Not only this helps you debug your schematic more easily, but it also allows you to perform connections between components without using long wires: see example below. Labelling is done by clicking on the **Label Net** button (shortcut is **F4**), inserting a net name and placing it on the desired net.

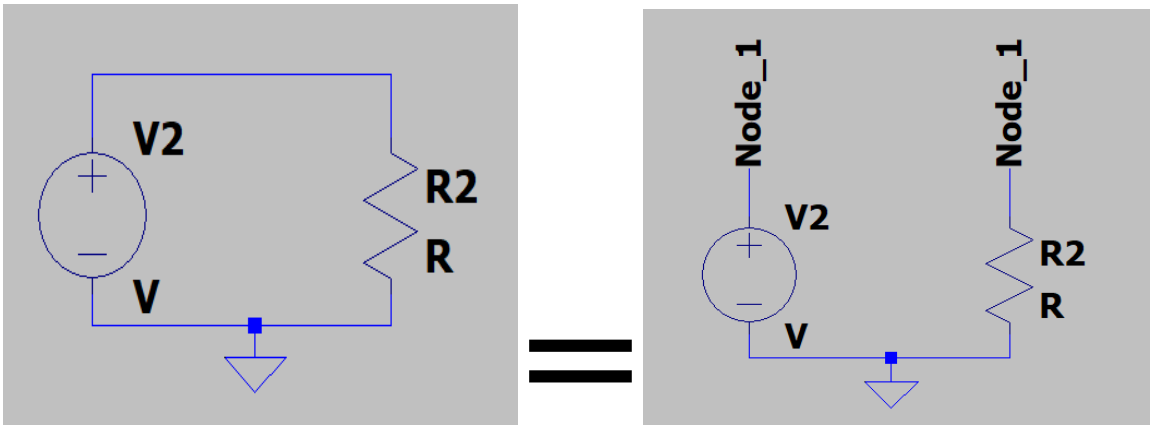


Figure 7 – Two equivalent ways of connecting components through Net Labelling

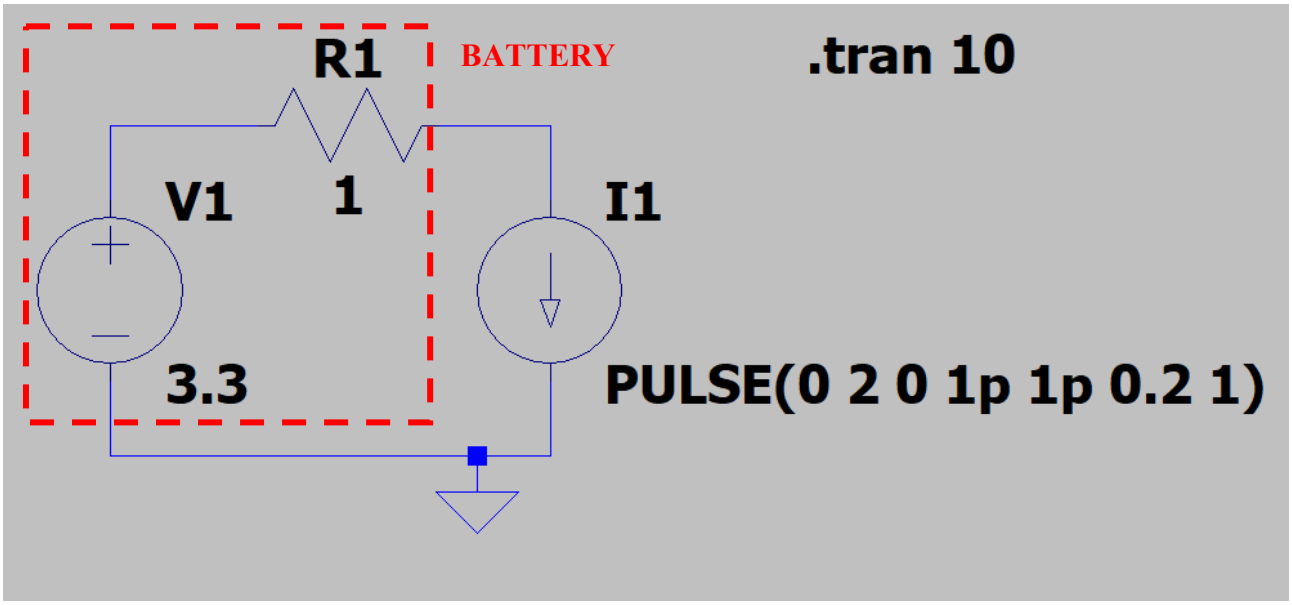
2.6.1. Using initial conditions

1. Change the IC attribute of the capacitor to -12. To do this, do **CTRL+Right Click** on the capacitor to bring up the property editor. In the SpiceLine cell, type: *IC= -12*. This means terminal 1 will by at -12V with respect to terminal 2. You've probably got terminal 1 at ground and so terminal 2 will start the simulation at 12V.
2. Click **OK**.
3. Rerun the simulation.
4. The graph plotter should restore the same traces you were viewing last time.
5. You should now see that the initial voltage on the capacitor is 12V (or -12V depending upon which pin you have connected to ground!). Make sure you see this effect of forcing the initial condition.

2.6.2. Exercise: Output Resistance of a Battery

You are given a battery (represented by V1 in series with R1 as in the circuit below) and you are asked to find the series resistance of the battery through simulation. Of course, from the circuit below you can see the series resistance is 1Ω, but what if this was a real battery, and you didn't know the value? How could we measure it? We are now going to simulate an experiment we could do in the lab.

By appropriately setting a pulsed current source and a transient simulation **estimate the series resistance** of the given battery. Note that you cannot simply set the value of V1 to zero in the simulatoin… you could not do this on a real battery in real life and we want to try to simulate something that can be physically achieved.



2.7. AC Sweep - Frequency Response Simulation

The final simulation type that we will look at is the AC analysis (frequency response). In this mode, SPICE lets us see the performance of a circuit as the input frequency is altered – for example looking at the frequency response of transistor amplifiers and filters.

1. Return to the last circuit we simulated in **LTspice**.
2. Right click on the voltage source, **V1** and select the **(none)** option instead.
3. Set the **AC Amplitude** parameter (under small signal AC Analysis) so that it equals 1. This provides a 1V sinusoid at a phase angle of 0° which will be swept across the frequency range specified in step 4 below. This is therefore considered the input of the filter (the RC circuit we now have is a low pass filter).
4. Choose **Simulate | Edit Simulation Cmd** and change the Analysis **Type** to **AC Analysis**. Set the parameters as follows:

   Type of sweep: Decade
   Number of points per decade: 21
   Start Frequency: 0.1
   Stop Frequency: 100k

5. Click OK and place the line on the schematic.

We set the output voltage label to **OUT**, to make it easier to identify later on. The circuit should now look like this:

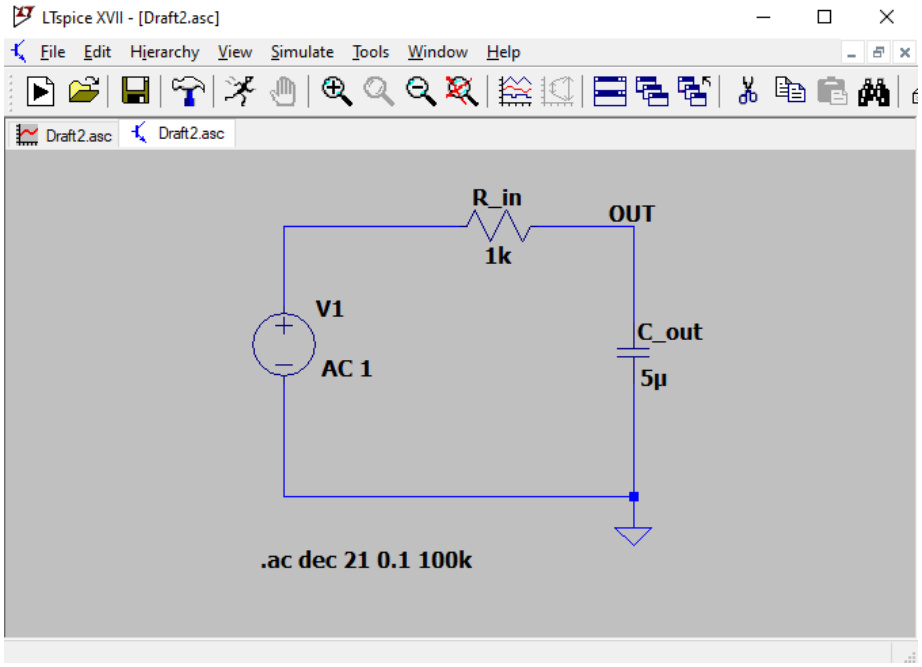


Figure 8 - circuit on which we look at frequency response

1. Run the simulation.
2. Obtain a trace of capacitor voltage V(out) and ensure that it corresponds to the expected filter output. The x-axis is already log scaled but the y-axis will need to be switched to log scaling (Right-click on the y-axis and tick the **Logarithmic** box).
3. You can observe that the dotted line represents the phase response (scaled along the y-axis values on the right) and the continuous line represents the magnitude response (scaled along the y-axis values on the left).

The graph plot window should now look something like the screen shot below.

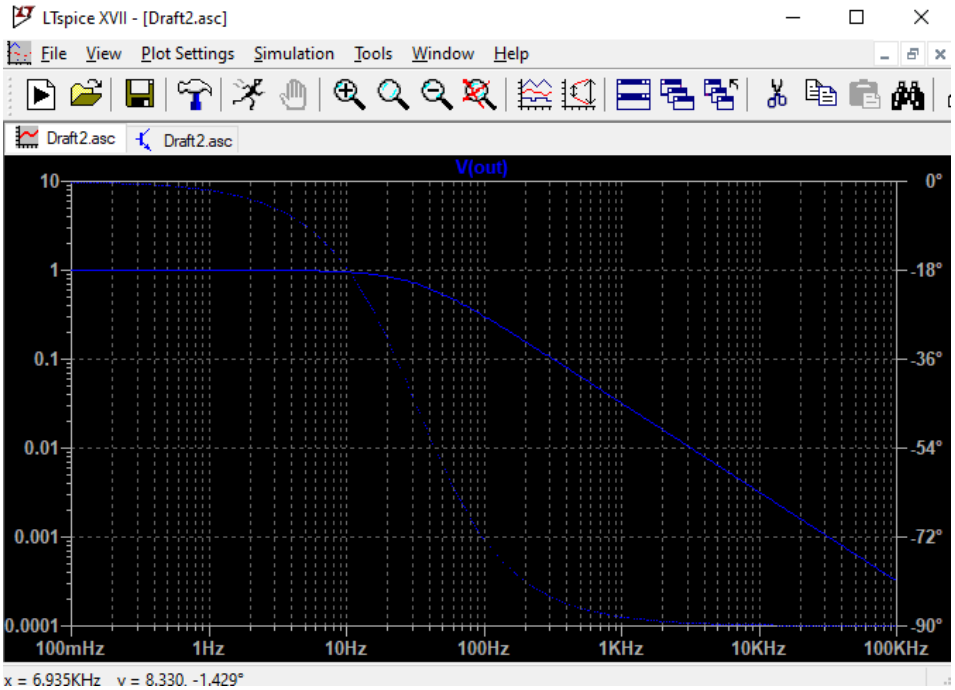
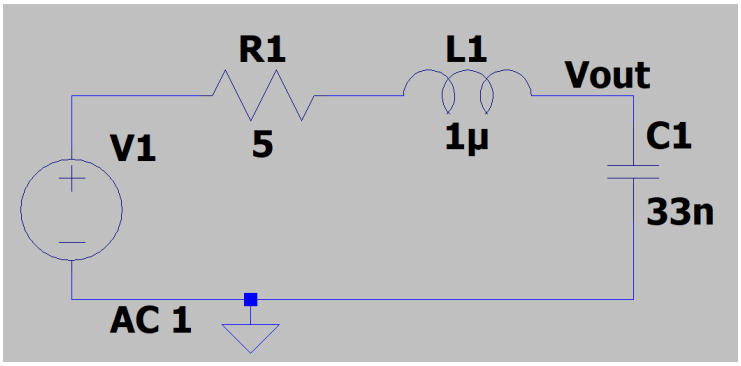


Figure 9 - Results of frequency response

2.8. Exercise: Frequency Response of an RLC Circuit

Draw the following circuit schematic. Set the voltage V1 to pulse and set the values as shown in the figure below.



Perform an AC analysis (setting the AC voltage of V1 to 1). Obtain a plot for the AC sweep at the vout node.

- Determine the resonant frequency of the circuit.
- Change the L1 and C1 combination to obtain a resonant frequency of around 6.78 MHz.
- Make sure the corner frequency is right by analysing the plot.
- Analyze the effect of changing R1 (between 0 and 10 Ohms), both with the original L1 and C1 values and with the new values you have found.
- Comment on the shape of the magnitude response of the cicuit.

3. A Note on Finding Parts in LTspice

You may want to search and simulate specific component. A wide number of commercial components' libraries are already available to you by simply clicking on the Components button . As you can ee in Figure 10 they are already divided into categories.

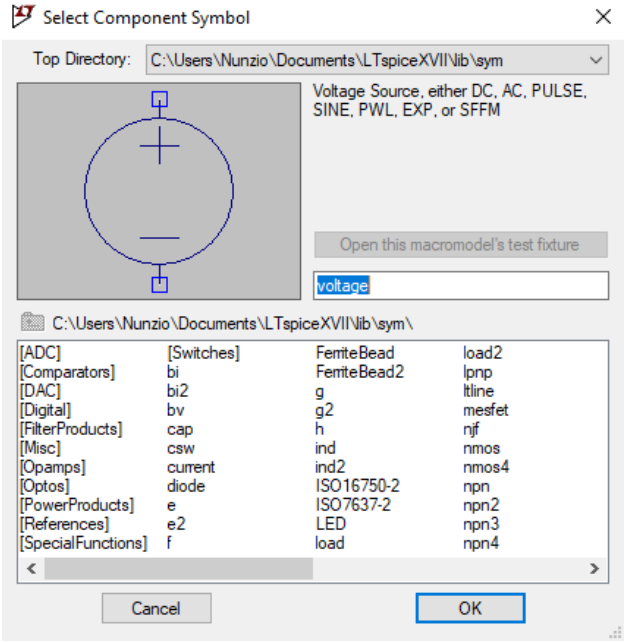


Figure 10 – Components Selection

If you do not find the component you are looking for, do not panic. You can find out how to import Third-Party Models through the following link:

https://www.analog.com/en/technical-articles/ltspice-simple-steps-to-import-third-party-models.html

4. Additional information

4.1. Looking at the frequency components of a signal (Fourier Analysis)

Until you have studied Fourier Analysis in your mathematics course, you may not understand this section. Do not panic! You will soon be able to understand it…

In order to see the frequency components of a time domain waveform (from a transient analysis) simply select **View | FFT** and choose the signal you want to plot.

Note that the data is processed using a non-windowed Fourier technique that only gives reliable results for periodic data.

4.2. What if my graph looks jagged - like it is missing some detail?

Sometimes when you plot a time-domain graph in the graph plotter**,** the graph will appear to be missing some detail – as if the simulation was not done with enough time steps for you to see all the complexity of the waveforms. For instance, you input a sine wave into a circuit and when you plot that sine wave the trace does not look completely sinusoidal (it is a few samples joined by straight lines.) Do not panic! SPICE has actually simulated the circuit accurately at the points that are being displayed. We just need to get it to calculate and display more points. In order to do this, go to **Simulate | Edit Simulation Cmd**. Set the maximum step size to something you think it reasonable so that you will see all the detail (if you have a signal at 100 Hz then set the sampling frequency to 1 kHz, which corresponds to a maximum step size of 1 ms). This will cause SPICE to take more data points during a transient analysis and thus be able to plot a graph which shows more detail.

5. **Shortcuts for LTSpice**

Command	Shortcut
Undo	F9
Redo	SHIFT+F9
Text	T
Insert Spice Directive	S
Components Tab	F2
Rotate	CTRL+R
Flip	CTRL+F
Draw Wire	F3
Label Net	F4
Place GND	G
Delete	F5
Copy	F6
Move	F7
Drag	F8
Save	CTRL+S
Open	CTRL+O
New Schematic	CTRL+N
Halt Simulation	CTRL+H
Pause Simulation	CTRL+P
Find	CTRL+F

6. Apple Users

We do not recommend that you use the LTSpice on MacOS and the GTAs won't be able to help you with it. If you use an Apple computer, you should run a virtual machine with a Windows 10 installed on it, and run LTSpice on that. This will be the case for other types of engineering software throughout your degree so it is worth setting up a VM now if you don't have one running.

The Mac version of LTspice has, shall we say, a minimalist user interface, and for example, does not even have the option to the **Edit Simulation Cmd** tab in the menus. You can insert the simulation type by writing a spice directive (**S**).

For this step you need to know the synthax of that simulation command as a spice directive. You can find a guide on the three main ones below:

DC Bias

Syntax: .op

Example: ***.op***

AC Analysis

Syntax: .ac <oct, dec, lin> <Npoints> <StartFreq> <EndFreq>

Example (Decade sweep from 10 Hz to 1 KHz, 40 points per decade) : ***.ac dec 40 10 1000***

Transient Analysis

Syntax: .tran <Tprint> <Tstop> [<Tstart> [<Tmaxstep>]] [<option> [<option>] ...]

Example (10s simulation starting to save data at 3s. Maximum timestep 0.2s): ***.tran 0 10 3 0.2***