

# Xcircuit and Spice

February 26, 2007

This week we are going to start with a new tool, namely *Spice*. *Spice* is a circuit simulator. The variant of *spice* we will use here is called *Spice-Opus*, and is a combined digital and analog circuit simulator. It is not as convenient as some other *spice* variants, which allow you to create and analyse circuits using purely graphical operations. In this version of *spice*, we must create programs called “netlists” using which we describe both the circuit to be analysed and the type of analysis we wish to do on it.

## 1 Invoking Spice

At a command prompt, type

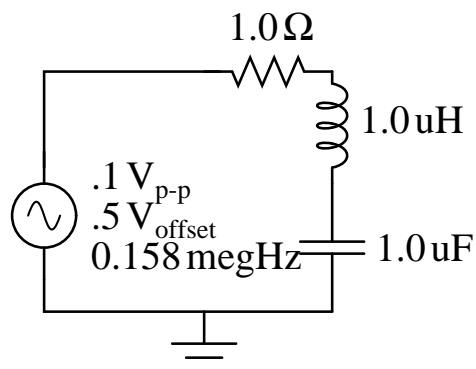
```
% spice3 &
```

This will start *spice* (*version 3*) and will open up an X-Windows command terminal at which you can give *spice* commands. The “&” tells Linux that you want *spice* to run in the background.

You need a *spice* program to use *spice*. So create the following file using your favourite editor:

```
* A Simple RLC network, at resonance
* The first line of a spice netlist describes the file.
* Always start start a netlist with a comment.
v0 1 0 0 ac sin (0.5 0.1 0.158meghz)
* we have asked for a voltage source between
* nodes 0 and 1. The source has a 0.5V offset and a 0.1V
* AC source oscillating at 158 KHz.
r1 1 2 1k
l1 2 3 1u
c1 3 0 1u
* These lines place a resistor of 1 $\Omega$ , a capacitor of
* 1 $\mu F$  and an inductor of 1 $\mu H$  between nodes 1 and 2,
* 2 and 3, and 3 and 0 respectively.
.end
* This ends the netlist
```

Save this file as *tut4.cir*. It describes the circuit in the following figure:



In the spice window, type

```
source tut4.cir
```

This reads in *tut4.cir* into spice. We now wish to see how the circuit behaves when the frequency is varied. The command for that is

```
ac dec N startf endf
```

where “dec N” stands for get “N” points per decade, and “startf” and “endf” are the starting and ending frequencies. For example, type

```
ac dec 10 1khz 100meghz
```

If you wish to see the values of the voltages across the resistor, type

```
print v(1)-v(2)
```

This will print out five columns. The first column is the row number, the second and third are the real and imaginary parts of the complex frequency, and the fourth and fifth columns are the real and imaginary parts of  $v_1 - v_2$ .

What we really want to do is to plot the result. We are driving the circuit at different frequencies and expect the voltage across the resistor to show a resonant peak at  $10^6/2\pi = 159155\ \text{Hz}$ . Type

```
plot v(1)-v(2)
```

This will bring up a plot of the voltage across series  $L$  and  $C$ . If you wish to change aspects of the plot, click the “other” mouse button on the plot to get a menu. The type of the plot can be changed, you can zoom in and out, and you can print your plot. Printing includes “print to file” which means that the plot is saved to a *postscript* file. If you wish to have a *pdf* file instead

```
ps2pdf graph.ps
```

where *graph.ps* should be replaced by whatever name you gave when saving the plot.

The plot showed the peak as desired. Which means that most of the voltage fell across the resistor and the series *L* and *C* combination had zero impedance at that frequency. Now let us see what happens if we suddenly start the circuit with a switch - this is called transient analysis. Type

```
tran .1u 20u
```

The *tran* command runs the simulation from the starting time (first argument) till the ending time (second argument) in steps specified by the third argument. Now plot the voltage across the *LC*. Type

```
plot v(2)
```

A new window pops up showing how the voltage initially swings back and forth and then settles to a voltage of 0.5 V.

## 2 Rules for describing Elements

### 2.1 Resistors, Capacitors and Inductors

Each element takes the form:

```
Name i j value
```

where *Name* is the name we give the element, *i* and *j* are the nodes between which the element has been connected and *value* is the numerical value for the element. Note the following:

1. The first letter of any element name indicates to *spice* the nature of that element. So names starting with *R* should be used to define resistors, names starting with *C* should be used to define capacitors and names starting with *L* should be used to define inductors.
2. *i* and *j* are names. Any name is acceptable, but node "0" is special. That node is defined to be ground. So to get useful voltages, you *must* have a ground node.
3. The *value* is a number without a unit. The unit is inferred from the element name. This value can be specified in scientific notation, and the following acronyms are allowed (all examples for resistor values):

**1k** 1K $\Omega$

**1m** 10<sup>-3</sup> $\Omega$

**1u** 10<sup>-6</sup> $\Omega$

**1n** 10<sup>-9</sup> $\Omega$

**1p** 10<sup>-12</sup> $\Omega$

**1meg** 10<sup>6</sup> $\Omega$

**1g** 10<sup>9</sup> $\Omega$

## 3 Xcircuit

Creating a spice netlist is laborious. So there are many tools that combine a GUI for drawing the circuit with spice to solve for the circuit behaviour. These all cost money, however. What *is* free though, are tools that only create netlists, so that you can combine them with spice to achieve the same thing. The best such tool is *xcircuit*, a professional quality schematic tool. This tool is used by professionals in the field to create publication quality circuit diagrams. By the way, it also creates spice netlists.

### 3.1 Using Xcircuit

At a shell prompt type

```
spice assign1
```

This will start up xcircuit with an empty sheet called *assign1*. We want to create a rectifier circuit driven by a sinusoidal voltage source:

1. Type “l” to bring up the first symbol library. Select “ground” by pressing “c” (for copy) while the cursor is over the symbol. The library will vanish and the drawing sheet will reappear. Move the symbol to where you want it and click to paste it. Click the other button to end the copy.
2. Type “l” to bring up the symbol library. Now we want the second page. Type “l” again. Select a sinusoidal voltage source by pressing “c” with the cursor on it. Paste as before.
3. Repeat this process with a diode. We would like the diode to be horizontal. Press “r” (for rotate right) repeatedly till the diode points in the desired direction. Paste as before.
4. Repeat with a resistor and a capacitor, oriented appropriately. Except for the ground symbol, all these symbols come from the second page.
5. At this point you should have the desired symbols in their proper positions and orientations, but with blank space between them. Now draw wires. Wires are drawn by simply clicking the mouse button anywhere and then moving to another point. A polygon is drawn, whose vertices are defined by successive mouse clicks. Finish the polygon by clicking with the third button (on two button mice, this means clicking both buttons together).
6. We now have a circuit with the correct types of elements, but with the wrong element values. We now need to change the voltage source to be a 1 Volt source at 10 KHz, the resistor to be 10K $\Omega$  and the capacitor to be 1 $\mu$ F. To do this, we need to edit the individual elements.
  - (a) Edit the element. For this, click on the blue down arrow on the right, standing for “push (edit object)” and then click on the AC source. The source appears enlarged, and more importantly with the code that goes with it.

(b) Move the cursor over the spice text and press “e”. Press “e”. The text is now available to edit.

(c) Use the left and right arrows to move the cursor till it reaches the “5” (look at the bottom edit field rather than the string on the screen). Delete it and replace with “2”. Similarly replace the 1 by 10. The string should read:

```
spice:V%i %ps.p %ps.m 0 ac sin(0 2 10k)
```

Press RETURN to complete the edit.

(d) Similarly edit and modify the resistor and capacitor values.

7. *xcircuit* assigns arbitrary node names to the various interconnects in your circuit, which is inconvenient. So, let us add our own names to the nodes we care about.

(a) Move the cursor over the corner above the voltage source and press “T”. Type “in” and then press “7” on the numeric keypad. The various directions on the numeric keypad control the *justification* of the text. We want the text to the top and to the left of the corner, so we press “7”. Press <ENTER>.

(b) Repeat for the node between the diode and the  $10K\Omega$  and call it “d”. Use “2” on the numeric keypad to place the text below the node.

(c) Repeat for the node above the capacitor and call it “out”. Use “9” on the numeric keypad this time.

8. Save the sheet by writing the *xcircuit* ps file (in the File menu).

9. Exit *xcircuit*.

As soon as you exit *xcircuit*, *spice* will pop up along with a pretty picture of your circuit (courtesy *xcircuit* - remember that it is a professional circuit drawing tool).

### 3.2 Using the Xcircuit Netlist

The netlist created by Xcircuit is deficient in some ways. Some of the items need further definition. To solve this problem, the following command is automatically run when you exit form *xcircuit*

```
spc2cir assign1
```

This takes *assign1.spc* and converts it into a more acceptable file *assign1.cir* (acceptable, that is, to spice).

*spice* starts up and lists the circuit for your reference. In addition it also shows you the *xcircuit* diagram to help you understand the netlist.

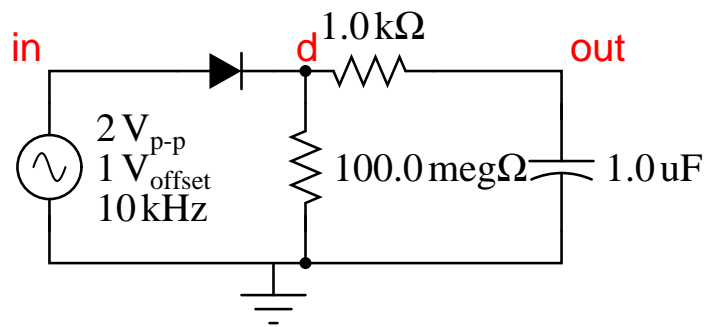
## 4 Problems for the Lab Class

The following problems have to be carried out during this class:

1. Create the RLC circuit of the first scilab session using the *xcircuit* - *spice* approach. Determine the frequency response between 500 and 1500 k-rad/sec. Vary  $R$  to different values by using “alter r1=value” and repeat the “ac” analysis. Note that “ac” analysis creates a vector called *frequency*. When plotting, you can plot against omega by using:

```
let omega=2*3.14159*frequency
plot v(out) vs omega
```

2. Create the following circuit and analyse it:



The purpose of the  $100M\Omega$  resistor is to allow the node “d” to have a connection to ground when the diode is off. Unless it is present, *spice* will object with “singular matrix” or other equally cryptic message. Do a transient analysis upto 10msec in steps of  $10\mu\text{sec}$ . Plot the voltages at nodes “in”, “d” and “out”. Zoom into the plot using “z” and look at the detailed response.

The diode is a nonlinear element. A 10 KHz sinusoidal source generates harmonics in the output when a diode is present. Look at the fourier series of the output waveform using the “fourier” command:

```
fourier 10k v(out)
```

The .ps and the .cir files should be submitted as part of your lab. The .cir file should contain the analysis commands inside a .control block, for example:

```
.control
ac lin 500k 1500k
omega=2*3.14159*frequency
plot v(out) vs omega
.endc
```

It is worth noting that an easy way to go back and forth between spice3 and the script is to execute “edit” in the spice window. That brings up xedit with the script. Everytime you save in xedit, you can run the script in spice by entering “source FILENAME”. Anything between “.control” and “.endc” is executed by spice on startup. Put all your assignment commands there. Once everything is working, submit it (or show it to the TAs).