NGSPICE- Usage and Examples

Debapratim Ghosh **Mohamed Jabir**

Wadhwani Electronics Laboratory (WEL)
Department of Electrical Engineering
Indian Institute of Technology Bombay
July 2017

Introduction

Q: What is electronic design?

A: Given some desired **specifications** to be achieved, we want to have a system which can be made by interconnecting **known elements**.

Specifications- We want an amplifier with a gain of 100.

Known Elements- This amplifier can be made using a transistor (BJT or MOSFET), or an op-amp, along with some resistors. Known elements are those whose behaviour can be represented by means of algorithms, equations or specific models. In short, known elements are those whose behaviour is familiar to us.

If we wish to design complex circuits, a circuit simulator is a useful tool.

What is NGSPICE?

- SPICE is an acronym for for **S**imulation **P**rogram with **I**ntegrated **C**ircuit **E**mphasis. First developed at UC Berkeley, it is the origin of most modern simulators.
- NGSPICE is an open source mixed-signal circuit simulator. It is the result of combining existing SPICE features with some extra analyses, modeling methods and device simulation features.
- It is freely available for use in Linux and Windows. It is recommended to use Linux for NGSPICE.
- NGSPICE requires you to describe your circuit as a **netlist**. A netlist is defined as a set of circuit components and their interconnections.

NGSPICE provides you with....

Basic Circuit Elements

- Passive components- resistors, capacitors, inductors, etc.
- Sources- independent voltage and current sources, controlled sources

Semiconductor Devices

- Pre-defined circuit elements such as diodes and transistors
- Allows you to define or include models of specific devices e.g. specialized transistors and op-amps

Circuit Analysis Techniques

- DC and AC analyses
- Transient and Steady-state analyses
- Pole-Zero, Noise analyses and more

Getting Started with NGSPICE (Linux)

- You can use any text editor (say, gedit) to write your circuit netlist. The first line in an NGSPICE file is **not executed**. It is used to describe the aim of the circuit being simulated.
- All NGSPICE comments start with an asterix, i.e. ‘*’
- The NGSPICE file comprises of the circuit netlist followed by the details of the analysis the user wishes to do.
- NGSPICE files are usually saved with the extension ‘.cir’ or ‘.spice’.
- Circuit components are identified by their first letter of naming, called **prefix**, i.e. resistors begin with `r`, capacitors with `c`, bipolar transistors with `q`, MOSFETs with `m`, voltage sources with `v` and so on.
- All circuit nodes are named/numbered. The netlist requires one ground node (zero potential).

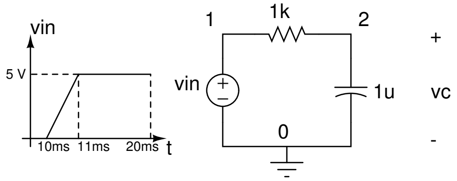
Inside the NGSPICE Shell

- Once any netlist is run by NGSPICE, the terminal is hooked to an NGSPICE shell, with a prompt such as `ngspice 1 ->`. You can exit the NGSPICE shell anytime by typing `exit` or `quit`.
- It is not always necessary to quit NGSPICE every time to run a new netlist. You can use the command `source <filename>.cir` to simulate a new netlist file using the NGSPICE prompt.
- The commands specified between `.control` and `.endc` in the netlist file may be used in the NGSPICE prompt separately.
- The waveforms may be saved as a postscript file by clicking on the hardcopy icon on the waveform window. However, this saves it as a default filename in the root directory. A better way to save the waveform in the working directory would be to use the following commands (say for the R-C circuit discussed above)

  ```
  set hcopydevtype=postscript
  hardcopy rcPlot.ps v(1) v(2)
  ```
- You can even save the plots using different colours. Read the NGSPICE manual for that, and much more!

Example I- Transient Analysis of an RC Circuit

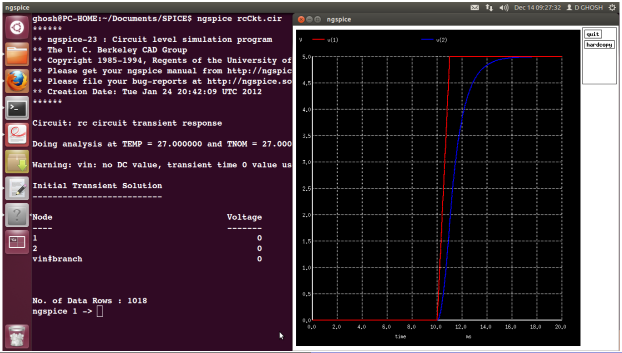
A simple RC circuit, excited by a user-defined signal V_{in}. We want to find the capacitor voltage i.e. $V_c(t)$. The netlist is as shown.



```
RC Circuit Transient Response
*resistor connected between nodes 1 and 2
r1 1 2 1k
*capacitor connected between nodes 2 and 0
c1 2 0 1u
*piecewise linear input voltage
vin 1 0 pwl (0 0 10ms 0 11ms 5v 20ms 5v)
*transient analysis for 20ms, step size 0.02ms
.tran 0.02ms 20ms
*defining the run-time control functions
.control
run
*plotting input and output voltages
plot v(1) v(2)
.endc
.end
```

Example I- Transient Analysis of an RC Circuit (cont'd)

- Once you have typed your netlist, save it with an appropriate name, say `rcCkt.cir`.
- Open the Linux terminal, and change the working directory to the folder where your netlist file is saved. e.g. if the file is in the `Documents` folder, type `cd ~/Documents` in the the command prompt.
- Run the netlist file using the command `ngspice rcCkt.cir`
- And that's it- your netlist should run! A snapshot is as shown.



```
ghosh@PC-HOME:~/Documents/SPICE$ ngspice rcCkt.cir
******
** ngspice-23 : Circuit level simulation program
** The U. C. Berkeley CAD Group
** Copyright 1985-1994, Regents of the University of
** Please get your ngspice manual from http://ngspic
** Please file your bug-reports at http://ngspice.so
** Creation Date: Tue Jan 24 20:42:09 UTC 2012
******

Circuit: rc circuit transient response

Doing analysis at TEMP = 27.000000 and TNOM = 27.000

Warning: vin: no DC value, transient time 0 value us

Initial Transient Solution
--------------------------

Node                                       Voltage
----                                       -------
1                                                0
2                                                0
vin#branch                                       0



No. of Data Rows : 1018
ngspice 1 -> 
```

Example II- AC Analysis of RC Circuit

For the same R-C circuit discussed in Example I, let us do the small-signal AC analysis, i.e. find its frequency response.After running this, you should be able to see two plot windows- a magnitude (dB) plot and a phase (degrees) plot.

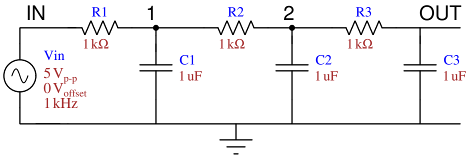
```
RC Circuit Frequency Response
r1 1 2 1k
c1 2 0 1u
*Specifying an AC source with zero dc
vin 1 0 dc 0 ac 1
*AC analysis for 1 Hz to 1MHz, 10 points per decade
.ac dec 10 1 1Meg
.control
run
*Magnitude dB plot for v(2) on log scale
plot vdb(2) xlog
*Phase degrees plot for v(2) on log scale
plot {57.29*vp(2)} xlog
.endc
.end
```

Subcircuits

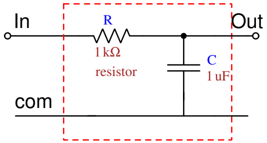
- We have seen how to use electronic devices to build a circuit and test it using NGSPICE.
- Various electronic devices have their own existing model files that represent the electrical behaviour of that device, which we can use in a netlist. What if we now have an existing circuit, and want to use it to build bigger circuits?
- A typical example is using an op-amp (operational amplifier) to design a simple amplifier or a filter. Note that, an op-amp is a pre-existing **circuit** and not a device. It is made of many transistors.
- NGSPICE allows us to define an op-amp as a **subcircuit**. A subcircuit is much like an IC- we know its pins to interface with the outside world, but we need not be familiar with the inside circuit!
- A subcircuit is a collection of devices familiar to SPICE. A subcircuit is identified by the prefix `x`. The usage is very similar to that of a model file.

Example - Subcircuits

See the circuit shown below.



- The netlist of the circuit will have redundant initiations of resistors and capacitors.
- Since all resistors and capacitors are in the same orientation, a subcircuit can be created with these components as shown below.



- The above subcircuit has 3 terminals, In, Out and com. It can be invoked multiple times to form the complete circuit.

Example - Subcircuits (cont'd)

Let us do a transient analysis for the circuit discussed previously.

```
RC Circuit Transient Analysis using Subcircuits
*Defining the Subcircuit 'RC_subcircuit'
.subckt RC_subcircuit In Out com
r In Out 1K
c Out com 1u
.ends
*Subcircuit definition ends here
vsin IN gnd sin(0 2.5 1K 0 0)
Invoking RC_subcircuit
xrc_1 IN 1 gnd RC_subcircuit
xrc_2 1 2 gnd RC_subcircuit
xrc_3 2 OUT gnd RC_subcircuit
.control
tran 0.02m 40m
plot v(IN)
plot v(OUT)
.endc
.end
```

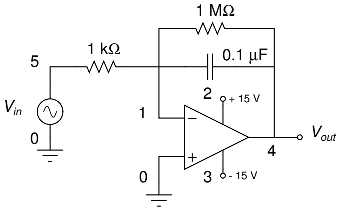
Example - Subcircuits (cont'd)

Thus, a subcircuit can be invoked multiple times to form the complete circuit. We can also initiate the components of the subcircuit with desired values, while its being invoked.

```
RC Circuit Transient Analysis using Subcircuits
.subckt RC_subcircuit In Out com r_1=1K c_1=1u
r In Out r_1
c Out com c_1
.ends
vsin IN gnd sin(0 2.5 1K 0 0)
xrc_1 IN 1 gnd RC_subcircuit
xrc_2 1 2 gnd RC_subcircuit r_1=100 c_1=0.1u
xrc_3 2 OUT gnd RC_subcircuit
.control
tran 0.02m 5m
plot v(IN) v(OUT)
.endc
.end
```

Here, `r_1` and `c_1` are the parameters of the subcircuit, with default values of 1KΩ and 1μF respctively. The circuit for the above netlist will have R2 and C2 as 100Ω and 0.1μF respectively. All other resistors and capacitors will have the default values of

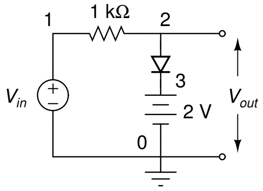
Example- Integrator using op-amp (741)



```
Differentiator using op-amp 741
*Including the predefined op-amp subcircuit file
.include ua741.txt
*Connections as mentioned in subcircuit file
x1 0 1 2 3 4 UA741
r1 5 1 1k
c1 4 1 0.1u
rf 4 1 1Meg
vcc 2 0 dc 15v
vee 3 0 dc -15v
*Giving a sinusoidal input
vin 5 0 sin (0 1v 1k 0 0)
.tran 0.02ms 6ms
.control
run
plot v(5) v(4)
.endc
```

Example III- DC Analysis of a Shunt Clipper

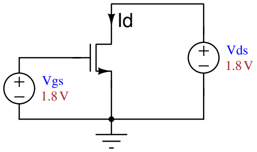
A DC analysis involves varying a voltage or a current source output throughout a range of values. Consider the following shunt clipper circuit. We wish to find the V_{out} v/s V_{in} characteristic for this circuit, say for -5 V $\leq$ $V_{in} \leq$ +5 V.



```
Shunt Clipper DC analysis
r1 1 2 1k
*Specifying a default diode p n
d1 2 3
*Independent DC source of 2V
vdc 3 0 dc 2
*Independent DC source whose voltage is to be varied
vin 1 0 dc 0
*DC Analysis on source vin, to vary from -5 to +5V
.dc vin -5 5 0.01
.control
run
plot v(2) vs v(1)
.endc
```

MOSFET Characterization- Family of Curves

Here, we will look at how to characterise a MOSFET. We will do it by generating a family of curves. Consider the circuit shown below,



To generate the family of curves, we change one of the voltages, say Vgs in steps, while the other, Vds is varied its entire range for each step of Vgs. Thus, we can plot the MOSFET current, Id (for various step values of Vgs) versus Vds.

MOSFET Characterization- Family of Curves (cont'd)

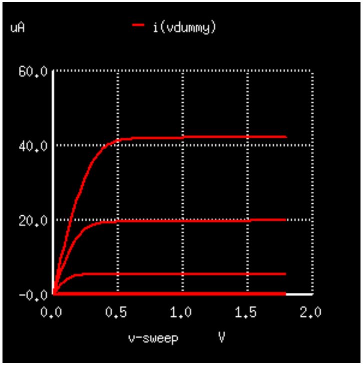
Let us generate the id v/s vds characteristics for various values of vgs. You should be getting the I-V characteristics as shown below.

```
Family of Curves

*Model file for the MOSFET
.include tsmc_spice_180nm.txt

*Usually, the order of terminals are
*drain, gate, source and body respectively

m1 d g gnd gnd CMOSN
vgs g gnd dc 1.8V
vds dummy gnd dc 1.8V
vdummy dummy d dc 0V
.control
dc vds 0 1.8 0.02 vgs 0 1 0.2
plot i(vdummy)
.endc
.end
```



Rules for Nomenclature

Table: Element Nomenclature

Element Type	First letter	Element Type	First letter
Resistor	r	Voltage(independent)	v
Capacitor	c	Current(independent)	i
Inductor	l	V (dependent on Vx)	e
Diode	d	I (dependent on Ix)	f
BJT	q	I (dependent on Vx)	g
JFET	j	V (dependent on Ix)	h
MOSFET	m	Sub circuit (ex. Op-Amp)	x

- For Nodes we can assign any numerical or alphabetical name but for reference node either 0 or 'gnd'.

Rules for Node sequence

Element Type	Node Sequence
R,L,C,D,V,I	positive-node negative-node
V (dependent on Vx)	p-node(V) n-node(V) p-node(Vx) n-node(Vx)
I (dependent on Vx)	p-node(I) n-node(I) p-node(Vx) n-node(Vx)
V (dependent on Ix)	p-node(V) n-node(V) Vx
I (dependent on Ix)	p-node(I) n-node(I) Vx
BJT	collector base emitter
JFET or MOSFET	drain gate source
Sub-circuit	according to defined in model file

Different sources

- sin(offset amplitude FREQ delay damping-factor)
- v node-p node-n dc 0 ac 1 (for ac analysis)
- pulse(initial-value pulsed-value TD TR TF PW period)
- pwl (t1 value@t1 t2 value@t2 t3 value@t3 ..)
- exp(initial pulsed rise-delay rise-time fall-delay fall-time)