

Circuit Design and Simulation with ngspice

Dinesh Sharma, J. John, P.C. Pandey, Kushal Tuckley

EE Department
IIT Bombay, Mumbai

March 10, 2023

What is ngspice?

SPICE is an acronym for “Simulation Program with Integrated Circuit Emphasis”. It was first developed at UC Berkeley. Most modern circuit simulators are based on it.

ngspice is an open source mixed-signal circuit simulator. (New Generation SPICE). It combines existing SPICE features with some extra analyses, modeling methods and device simulation features.

- It is freely available for use with Linux, Windows and MAC OS.
- Input to the program is a description of the circuit as a “net list”, kinds of analyses to be performed and the format of output.
- It solves the network equations and presents results as a print or a plot.

What is a net list?

- ngspice is familiar with the behaviour of a large variety of circuit components like resistors, capacitors, inductors, diodes . . .
- You name each component in your circuit uniquely, with the first letter of the name identifying the type of the component (R for resistor, L for inductor, C for capacitors, D for diodes, Q for bipolar transistors, V for voltage sources, I for current sources – and the list goes on . . .)
- Similarly, name each node uniquely. This could be a number or a more descriptive name like “Input”.
- Now you insert each component, giving its name (for example R1), the nodes between which it is connected: (for example Input, 0) and its value (for example 10KOhms).

That's it. This list of components with connectivity information is the net list.

ngspice Inputs

- Components like transistors and diodes don't have a simple "value". For these we provide a model, which carries the equations and parameters describing the behaviour of these components.
- We can add independent and dependent voltage and current sources as well.

We also specify the kind of computation we want to do –

- Computing DC values (as we sweep some voltage – say the input, over a range),
- Computing Transient values (voltages/currents as a function of time)
- or amplitudes of sinusoidal signals as a function of frequency (AC analysis).

ngspice Outputs

Finally, we tell ngspice what output we need – a tabular printout, or a plot or time difference between marked events etc.

Now we are ready to run ngspice. We give the input file to ngspice and it produces the desired output ...

Or (Ugh ...!) a list of errors.

In that case we correct the errors and re-run ngspice.

We'll now describe how ngspice may be installed on different operating systems.

ngspice Installation

On MS Windows If you are on MS Windows (say, 64 bit Windows 10/11), download ngspice-38_64.zip from:
https://sourceforge.net/projects/ngspice/files/ng-spice-rework/38/ngspice-38_64.zip/download

Expand the contents of the zip file to some folder on your computer.

(You should have read and write access to this location, if not run in admin mode), e.g. to D:\

You may also download a GUI from
https://ngspice.sourceforge.io/experimental/ngspice_start.7z
Expand its content into the Spice64\bin folder.
(Just search for ngspice windows download for more information).

ngspice Installation

On Linux Pre-compiled versions may be available for your distribution (fedora/Ubuntu/ . . . etc). Just install using the utilities of your OS (dnf for fedora, aptget for Ubuntu etc.).

You can also download the source code from sourceforge and compile it yourself.

- You will need to install development tools first if you don't have these already.
- Next, you should download the source code in tar format from:

<https://sourceforge.net/projects/ngspice/files/ngspice-rework/39/ngspice-39.tar.gz>

ngspice Installation on Linux

- Extract the source code in some appropriate directory (with root privileges) – say in /usr/local/src.
- Then follow the standard unix procedure for installing . . .

Most Linux programs are installed from source in three steps:

- 1 Configure using “./configure” in the source directory,
 - 2 Compile using “make”
 - 3 Install using “make install”
- You can optionally do “make clean” after installation to remove intermediate files.

ngspice Installation on MAC

ngspice compiled for MAC is not distributed by sourceforge as of now.

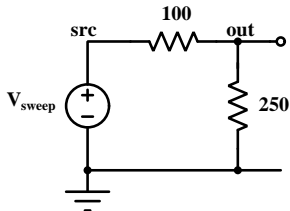
However, ngspice port to MAC may be available at ...

<https://ports.macports.org/port/ngspice>

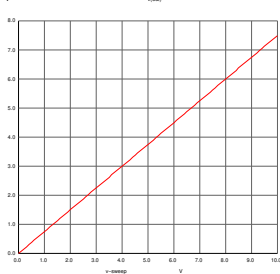
It may also be possible to compile ngspice from source on MAC as it is done for Linux.

Advantage of compiling from source is that one can install the latest version of the software. Pre-compiled binaries are typically several versions behind the source code.

An Example: Potential divider



dc1: "potential divider
V



* Potential divider

* Net list

Vsweep src 0 DC 0V

R1 src out 100

R2 out 0 300

* Analysis

.DC Vsweep 0 10 0.1

* Output

.control

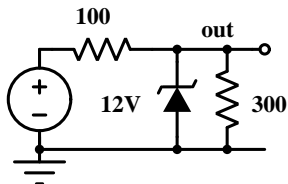
run

plot V(out)

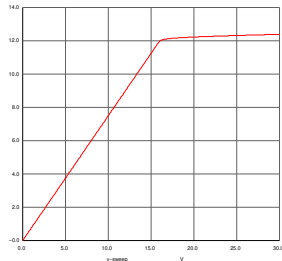
.endc

.end

Example: 12V Zener Regulator



dc1: * 12v zener regulator
V



* 12V Zener Regulator

* Net list

Vsweep src 0 DC 0V

Rs src out 100

DZ 0 out Dzener

.model Dzener D bv=12.0 Rs=1

IKR=1.0E-3

RL out 0 300

* Analysis

.DC Vsweep 0 30 0.1

* Output

.control

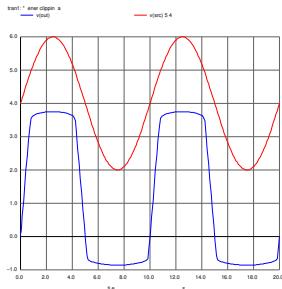
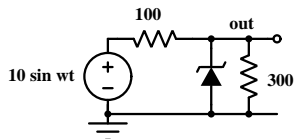
run

plot V(out)

.endc

.end

Example: Zener clipping



* Zener clipping

* Net list

Vsupply src 0 Sin (0 10 100)

Rs src out 100

DZ 0 out Dzener

.model Dzener D bv=3.5 Rs=1

IKR=1.0E-3

RL out 0 300

* Analysis

.tran 1E-4 2E-2 0

* Output

.control

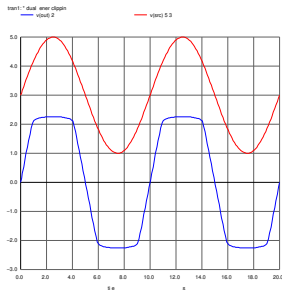
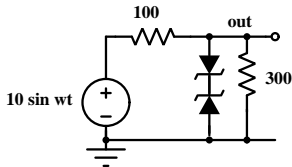
run

plot V(src)/5+4 V(out)

.endc

.end

Example: Dual Zener clipping



* Dual Zener clipping

* Net list

Vsupply src 0 Sin (0 10 100)

Rs src out 100

DZ1 0 midp Dzener

DZ2 out midp Dzener

.model Dzener D

+ bv=3.5 Rs=1 IKR=1.0E-3

RL out 0 300

* Analysis

.tran 1E-4 2E-2 0

* Output

.control

run

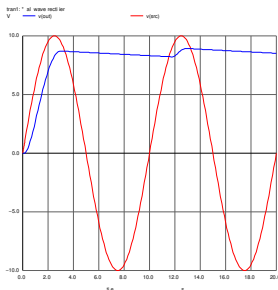
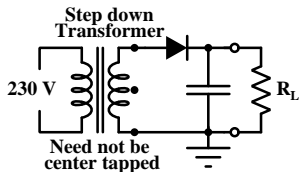
plot V(src)/5+3

V(out)/2

.endc

.end

Half Wave Rectifier



* Half Wave rectifier

* Net list

Vsupply src 0 Sin (0 10 100)

D1 src out Drect

.model Drect D

+bv=100 Rs=1 IKR=1.0E-3

RL out 0 300

Csmooth out 0 500u

* Analysis

.tran 1E-4 2E-2 0

* Output

.control

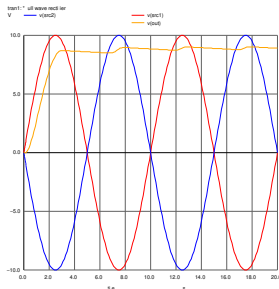
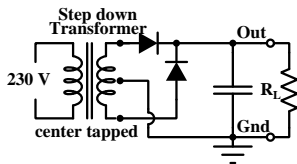
run

plot V(src) V(out)

.endc

.end

Full Wave Rectifier



* Full Wave rectifier

* Net list

```
Vsec1 src1 0 Sin (0 10 100)
```

```
Vsec2 src2 0 Sin (0 -10 100)
```

```
D1 src1 out Drect
```

```
D2 src2 out Drect
```

```
.model Drect D
```

```
+ bv=100 Rs=1 IKR=1.0E-3
```

```
RL out 0 300
```

```
Csmooth out 0 500U
```

* Analysis

```
.tran 1E-4 2E-2 0
```

* Output

```
.control
```

```
run
```

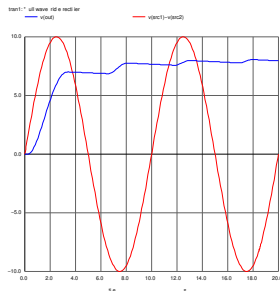
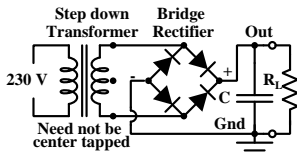
```
plot V(src1)
```

```
V(src2) V(out)
```

```
.endc
```

```
.end
```

Full Wave Bridge Rectifier



* Full Wave Bridge rectifier

* Net list

D1 src1 out Drect

D2 src2 out Drect

D3 0 src1 Drect

D4 0 src2 Drect

Vsec src1 src2 Sin (0 10 100)

.model Drect D

+ bv=100 Rs=1 IKR=1.0E-3

RL out 0 300

Csmooth out 0 500U

* Analysis

.tran 1E-4 2E-2 0

* Output

.control

run

plot

V(src1)-V(src2)

V(out)

.endc

.end