

SAPICE v.0.1  
Symbolic Analysis from sPICE netlists in sagemath  
SAge+ngsPICE

Alessandro Bernardini

June 22, 2013

Version: 0.1 (initial commit).

Author: Alessandro Bernardini

`alessandro.bernardini.tum@gmail.com`

License: GNU GPL.

<http://www.gnu.org/licenses/gpl.html>

Disclaimer: THERE IS NO WARRANTY FOR THE PROGRAM (SAPICE and all its provided components), TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS (Alessandro Bernardini) AND/OR OTHER PARTIES PROVIDE THE PROGRAM AS IS WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

In the present version the program was NOT well tested, so bugs are expected: please report all kind of bugs and issues you will find.

Project home: <https://github.com/alessandro-bernardini/SAPICE>  
Requires: ngspice revision 24; sage version 5.6. Other version of ngspice or sage should work well too.

<http://www.sagemath.org/>  
<http://ngspice.sourceforge.net/download.html>  
Date: June 2013

## 1 Overview

The program SAPICE for now consists of the python module `sage_circuit_analysis.py`

This python package is intended for use in SAGE. ngspice is required too.

It provides the class `small_signal_linear_circuit` which reads an NGSPICE netlist and computes in symbolic form the nodal equations for the linearized small signal circuit in the Laplace domain. Those nodal equations in the Laplace domain can be fully symbolic or the numerical values read from the NGSPICE netlist can be substituted for the circuit parameters.

The so obtained equations can be solved with SAGE.

Project homepage: <https://github.com/alessandro-bernardini/SAPICE>  
Works with: ngspice revision 24; sage version 5.6  
Download link for SAGE:  
<http://www.sagemath.org/>  
Download link for NGSPICE:  
<http://ngspice.sourceforge.net/download.html>  
or via `sudo apt-get install ngspice`

## 2 Usage

Start with a description of your circuit's netlist in an ngspice circuit file `circuitfile.cir`.

Only resistor, capacitors, inductors, current sources, voltage controlled current sources and BJTs are supported for now. *The netlist must be flat.*

Independent voltage sources are not supported and an independent voltage sources has to be replaced with an electrically equivalent current source (source transformation). In doing so the internal resistance of the source has to be considered.

This is because for now only nodal analysis is provided. In future versions modified nodal analysis is planned and this will support voltage sources and short circuits.

A short circuit ( $R = 0\Omega$ ) is also not supported for now and for dealing with a short circuit the original ngspice netlist has to be modified, identifying both nodes of the short with the same node identifier in the netlist. In alternative a sufficiently low resistance can approximate the short.

The ground node must be named 0 or *gnd* or *GND*.

The netlist must be processed by ngspice and a batch output file `batchoutput` is required for the `.OP` data and the description of BJT models (so put an `.OP`

line in your ngspice file - no other analyses should be activated except the operating point computation with *.OP*).

For this run the command (from the unix prompt):

```
ngspice -b circuitfile.cir -o batchoutput.log
```

Then start SAGE.

From the sage prompt load the file containing the `small_signal_linear_circuit` class description with<sup>1</sup>:

```
load("sage_circuit_analysis.py")
```

Now you can use the classes provided in the module. From SAGE prompt:

```
circuitinstance = small_signal_linear_circuit("circuitfile.cir", "batchoutput.log")
```

`circuitinstance` will be an object that symbolically describes the circuit in SAGE.

You can provide additional options in the constructor of the `small_signal_linear_circuit`: `check_operating_region=False` will assume that the transistors are in the forward active region of operation without checking if this is true; `set_default_IC_to_zero=False` will not consider a default zero initial condition when initial conditions are *not* provided; `ignore_all_IC=True` will set to zero every initial condition, even those explicitly specified and this both for symbolic nodal equations and for nodal equations with substituted numerical values regarding the circuit parameters.

### 3 Description of methods and data members

We will describe the main methods and data members of the class `small_signal_linear_circuit`. Clearly there are invoked with `circuitinstance.member()` or `circuitinstance.data`

`.nodal.equations`

prints the nodal equations for the small signal linearized circuit in the Laplace domain as a dictionary that associates to each node the relative nodal equation in symbolic form in SAGE

`.nodal.equations.substitutions`

prints the nodal equations for the small signal linearized circuit in the Laplace domain as a list for SAGE, substituitng the numerical values given in the netlist for the circuit parameters. The indipendent current sources are always in symbolic form a function of the variable *s*.

`.print.information()`

prints informations about the circuit and the equations.

`.solve.nodal.equations()`

solves the nodal equations for nodal analysis of the small signal linearized circuit in the Laplace domain in symbolic form. The computation can take time. The result is a list containing a single element which is a list which contains the

---

<sup>1</sup>here we assume the file beeing in the current directory, see `pwd`

nodal voltages in symbolic form<sup>2</sup>.

`.solve_nodal_equations_with_substitutions()`

solves the nodal equations for the nodal analysis of the small signal linearized circuit in the Laplace domain replacing the numerical values for the circuit parameters. The type of the result is as for the symbolic case.

`.additional_equations`

prints additional equations linking the linearized small signal circuit quantities to operating point values and other parameters.

`.additional_equation_explicit`

prints additional equations linking the linearized small signal circuit quantities to operating point values and other parameters in the form of a dictionary that can be used for substitutions in SAGE.

`.prin_lin_circuit`

prints the linearized small signal linear circuit. Usefull for further manipulation of the circuit. The output can then be feed back to SAPICE.

`.default_substitutions`

prints the default substitutions as a dictionary. Usefull for substitutions in SAGE.

`.default_substitutions_values`

prints the default substitutions as a dictionary. Usefull for substitutions in SAGE. Numerical values are considered for the small signal equivalent circuit elements too that depends on the operating point of semiconductor devices.

`.initial_conditions`

retuns the initial conditions for (small signal equivalent) circuit elements.

`.nodal_voltages`

retuns the nodal voltages as a list.

`.print_symbol()`

prints an explanation of used symbols.

## 4 Known issues

The software is only in the initial stage and was NOT really tested. This is only the first commit.

Please contribute in testing the software and report bugs.

---

<sup>2</sup>more precisely the nodal voltages are given in the form  $V_{nodename} == expression$ .