SPICE Module
User’s Guide
Contents

1 Introduction

2 PSIM-SPICE Interface
   2.1 Overview 3
   2.2 SPICE Model Libraries 3
      2.2.1 SPICElib Folder 3
      2.2.2 Search Paths for SPICE Models 3
      2.2.3 Directives .LIB and .INCLUDE 4
      2.2.4 Set Path for LTspice Executable 4
      2.2.5 Find SPICE Models in Libraries 4
      2.2.6 Netlist Syntax Check 5
   2.3 Simulation Control Dialog Options for SPICE Simulation 6
      2.3.1 Transient Analysis 6
      2.3.2 AC Analysis 7
      2.3.3 DC Analysis 8
      2.3.4 Step Run Option 9
      2.3.5 Other Analysis Options 9
   2.4 PSIM Elements for SPICE Simulation 10
      2.4.1 Multi-Level Elements 10
      2.4.2 SPICE Directive Block 12
      2.4.3 SPICE Subcircuit Netlist Block 13
   2.5 Create PSIM Element from SPICE Netlist 14

3 SPICE Analysis Directives
   3.1 Overview 19
   3.2 Convergence 19
   3.3 SPICE Analysis Types 19
      3.3.1 .AC 19
      3.3.2 .DC 20
      3.3.3 .END 20
      3.3.4 .ENDS 20
      3.3.5 .FOUR 20
      3.3.6 .FUNC 20
      3.3.7 .GLOBAL 21
      3.3.8 .IC 21
      3.3.9 .INCLUDE 21
      3.3.10 .LIB 21
      3.3.11 .MODEL 22
      3.3.12 .NODESET 22
      3.3.13 .NOISE 23
      3.3.14 .OP 23
      3.3.15 .OPTIONS 23
      3.3.16 .PARAM 24
      3.3.17 .SAVE 25
4 SPICE Elements and Device Models

4.1 Overview 29
4.2 Passive Elements 29
  4.2.1 Resistor 29
  4.2.2 Capacitor 29
  4.2.3 Inductor 30
  4.2.4 Coupled Inductor 30
4.3 Transmission Lines 31
  4.3.1 Lossless Transmission Line 31
  4.3.2 Lossy Transmission Line 31
  4.3.3 Uniform Distributed RC Line 32
4.4 Active Elements 33
  4.4.1 Voltage-Controlled Switch 33
  4.4.2 Current-controlled switch 33
  4.4.3 Diode 34
  4.4.4 Bipolar Junction Transistor (BJT) 35
  4.4.5 MOSFET 35
  4.4.6 Junction Field-Effect Transistor (JFET) 36
  4.4.7 MESFET 36
4.5 Sources 38
  4.5.1 Independent voltage and current sources 38
    4.5.1.1 Pulse 39
    4.5.1.2 Sinusoidal 39
    4.5.1.3 Exponential 40
    4.5.1.4 Piece-Wise Linear 40
    4.5.1.5 Single-Frequency FM 40
    4.5.1.6 Amplitude Modulated Source (AM) 41
    4.5.1.7 Transient Noise Source 41
    4.5.1.8 Random Voltage Source 41
  4.5.2 Linear Dependent Sources 42
    4.5.2.1 Voltage-Controlled Voltage Source 42
    4.5.2.2 Voltage-Controlled Current Source 42
    4.5.2.3 Current-Controlled Voltage Source 43
    4.5.2.4 Current-Controlled Current Source 43
  4.5.3 Nonlinear Dependent Sources (Behavioral Sources) 43

5 Reference Document List
The SPICE Module is an add-on option of the PSIM software. It provides the convenience for users to utilize PSIM’s graphic circuit schematic interface for SPICE simulations and analyses.

The SPICE Module is powered by CoolSPICE\(^1\) from CoolCAD Electronics, LLC. The CoolSPICE engine provides the capability for mixed-level/mixed-signal SPICE circuit simulation. The SPICE engine in CoolSPICE is based on NGSPICE [1].

The SPICE Module provides the capability to capture the PSIM circuit schematic, convert it into SPICE netlist, and run SPICE simulation. It allows user to create the circuit schematic in PSIM, and run SPICE simulation by simply clicking on the button "Run SPICE Simulation". It also provide the option to read from other SPICE software, as long as the netlists are in standard SPICE or NGSPICE format.

Features of the SPICE Module include:

- Support of standard SPICE and NGSPICE models and analysis.
- Support of LTspice\(^2\) simulation with PSIM schematic in PSIM environment.
- Mixed-mode and behavior modeling.
- Accommodation of component models in other SPICE’s device database.
- Generation of LTspice netlist for LTspice simulation.

This manual describes how to use SPICE Module with PSIM schematic, and provides essential information for SPICE analysis, elements, and models.

**Run SPICE Simulation**

To run SPICE simulation, click on the Run SPICE Simulation button on tool bar or select "Simulate >> Run SPICE Simulation" from the pull-down menu, as indicated below.

The SPICE warning or error messages will be displayed in PSIM’s message window.

The simulation result waveforms for the output voltage and inductor current are displayed in the SIMVIEW window.

---

1. CoolSPICE is copyright by CoolCAD Electronics, LLC., 2011-2018
2. LTspice is copyright by Linear Technology Co., 1998-2018
Run LTspice Simulation

PSIM circuit schematics can run LTspice simulation. The simulation result will be displayed the same way as PSIM simulation result.

Before running LTspice simulation, user must install LTspice on the computer, and specify the location of LTspice executable file by clicking PSIM menu Options >> Set Path. At the bottom of this dialog, as shown in the picture below, user the Browse button to find and enter the path and filename for the LTspice executable file. Then, click Save and Close the dialog.

To run LTspice simulation, click on the Run LTspice Simulation button on tool bar or select Simulate >> Run LTspice Simulation from the pull-down menu, as indicated below.

When running LTspice simulation, only the content of the .log file generated from LTspice will be displayed in PSIM’s message window. During the time while LTspice simulation is running, there is no status or progress information passed from LTspice to PSIM for display.

At the end of LTspice simulation, PSIM’s message window will refresh the message from the .log file generated by LTspice, and the simulation result will be displayed in SIMVIEW.
2.1 Overview

The SPICE module captures the PSIM circuit schematic and runs SPICE simulation. It provides convenient ways to integrate the models in the form of SPICE netlists into PSIM environment.

2.2 SPICE Model Libraries

PSIM’s SPICE module can load device models in SPICE netlist form directly into PSIM schematics. Even the encrypted models can be used. The libraries of the SPICE models are managed in the following ways.

2.2.1 SPICElib Folder

PSIM’s SPICE module includes some models in SPICE netlist form. These models are stored in PSIM’s folder "Powersim\SPICElib". Users may add into this folder the netlist files containing models developed by themselves or by other developers.

2.2.2 Search Paths for SPICE Models

PSIM provides a convenient way for users to utilize pre-existing netlist files stored in other locations than SPICElib. Users may set the search path in PSIM.

Under PSIM menu Options >> Set Path, specify or add the locations of the SPICE library files, as shown below. Remember to Reload Models and Save the path settings before closing the dialog.
2.2.3 Directives .LIB and .INCLUDE

If user prefers to store the model netlist files in a location other than PSIM search paths, the “.include” command should be written in the SPICE Directive Block in the PSIM schematic:

```
.include <absolute path>/<filename>
```

If the models are in encrypted library files, such as "<filename>.lib", no matter whether the files are in the PSIM search paths or not, user must write the "lib" command in the SPICE Directive Block in the PSIM schematic:

```
.lib <absolute path>/<filename>.lib
```

This is because the name of the models and subcircuits are also encrypted in such files. PSIM can not un-encrypt those names, and hence, can not find them in the search paths.

2.2.4 Set Path for LTspice Executable

To run LTspice simulation from within PSIM schematic, LTspice must be installed on the computer, and the path to the LTspice executable file must be defined in PSIM.

As shown at the bottom of the graph in the above section, click the Browse button to find the path.

2.2.5 Find SPICE Models in Libraries

All the models in the files in the search path folders can be loaded into PSIM schematics. To check if the search path is set correctly and what models are available, user may click "View >> SPICE Model List" and "View >> SPICE Subcircuit List", as shown below.
### 2.2.6 Netlist Syntax Check

The SPICE Module uses the CoolSPICE engine, and the netlist formats are based on NGspice netlist syntax. It accepts most of but not all the Pspice and LTspice syntaxes. Some SPICE netlist files from other sources may contain syntaxes which are not accepted by PSIM-SPICE.

We have provided the function for users to check and convert the netlist syntaxes into what can be accepted by the SPICE Module. This function is in **Utilities >> SPICE Netlist Check**.

As shown in the picture below, user can load the netlist file, select the original format (either LTspice or PSpice), and click the "Check" button. The original netlist statements are on the left hand side and the converted PSIM-SPICE module accepted netlist statements are on the right hand sides.

- Lines highlighted in blue indicate those statements are converted.
- Lines highlighted in yellow indicate those statements have no compatible conversion. One must convert those statements manually.
2.3 Simulation Control Dialog Options for SPICE Simulation

For the three basic analyses: DC analysis, AC analysis, and transient analysis, the simulation controlling parameters and options are defined in PSIM’s Simulation Control dialog, under the SPICE tab, as explained in the following subsections.

For SPICE analyses and options not included in the Simulation Control dialog, user must write SPICE commands and define the options in PSIM’s "SPICE Directive Block". This block will be explained in Section 2.4.2 SPICE Directive Block.

2.3.1 Transient Analysis

For transient analysis, the Simulation Control dialog accepts the following parameters:

- **Use Initial Conditions**: If the box is checked, the "UIC" option is added to the .tran command.
  - Note: This setting also affects AC and DC Analysis.

- **Time Step**: Suggested computing increment and printing or plotting increment, in second.

- **Max Step**: The maximum step size that SPICE uses, in second. For default, the program chooses either tstep or (tend-tstart)/50.0, whichever is smaller. tmax is useful when one wishes to guarantee a computing interval which is smaller than tstep.

- **Start Time**: The initial time, in second. The transient analysis always begins at time zero. In the interval from zero to tstart, the circuit is analyzed (to reach a steady state), but no outputs are stored.

- **End Time**: The final time, in second.

- **Integration Method**: The numerical integration methods used by SPICE. Choices are: Trapezoidal, Modified Trapezoidal, and Gear.
  - Note: This setting also affect AC and DC Analysis.

- **Order**: The order for the numerical integration method. Available values for Trapezoidal method are 1 and 2. Available values of Gear method are from 2 to 6.
2.3.2 AC Analysis

For ac analysis, the Simulation Control dialog accepts the following parameters:

**Sweep Type**

Defines the sweep types. The options are: Octave, Decade, Linear, or List.

Parameters for the Octave or Decade option:

- **Start Freq**: Starting frequency, in Hz.
- **End Freq**: Final frequency, in Hz.
- **Points/oct (or dec)**: Number of points per octave or decade.

Parameters for the Linear option:

- **Start Freq**: Starting frequency, in Hz.
- **End Freq**: Final frequency, in Hz.
- **Points**: Total number of points.

Parameters for the List option:

- **Freq List**: A list of frequencies to be analyzed, in Hz. Each value is separated with a space.
2.3.3 DC Analysis

For dc analysis, the Simulation Control dialog accepts the following parameters:

**Name**
Name of the source for DC sweep. Source 1 is by default the x-axis. Source 2 can be enabled. The DC sweep sources can be voltage, current, or temperature.

**Sweep Type**
Defines the sweep types. The options are: Octave, Decade, Linear, or List.

Parameters for the Octave and Decade options:

- **Start**: Starting value.
- **End**: Final value.
- **Points/oct (or dec)**: Number of points per octave or decade.

Parameters for the Linear option:

- **Start**: Starting value.
- **End**: Final value.
- **Increment**: Incremental step size.

Parameters for the List option:

- **List**: A list of values to be analyzed. Each value is separated with a space.
2.3.4 Step Run Option

If enabled, SPICE simulation would run a parameter sweep. Note that this function is not implemented in CoolSPICE for this version of PSIM. But it is available for LTspice netlist generation.

Parameter Name of the parameter for Step Run. If the step run is NOT for a voltage or a current source, or temperature, the box "PARAM" must be checked.

Sweep Type Defines the sweep types. The options are: Octave, Decade, Linear, or List.

Parameters for the Octave and Decade options:
- **Start**: Starting value.
- **End**: Final value.
- **Points/oct (or dec)**: Number of points per octave or decade.

Parameters for the Linear option:
- **Start**: Starting value.
- **End**: Final value.
- **Step**: Incremental step size.

Parameters for the List option:
- **List**: A list of values to be analyzed. Each value is separated with a space.

2.3.5 Other Analysis Options

The Simulation Control dialog also accepts the following analysis options:

Operating Point: If enabled, SPICE simulation will determine the DC operating point of the circuit with inductors shorted and capacitors opened.

Thermal Analysis (CoolSPICE only) If enabled, SPICE simulation will perform thermal analysis by including the heating effect in device models. All CoolSPICE SiC and GaN models support thermal analysis. With the thermal analysis, the extra node of SiC/GaN models will show the junction temperature of the devices in °C.

Note that this option is for CoolSPICE simulation only

Step Run Option: If enabled, SPICE simulation would run a parameter sweep. Note that this function is not implemented in CoolSPICE for this version of PSIM. But it is available for LTspice netlist generation.
- **Parameter**: The name of the parameter to sweep
- **Start**: Starting value
- **Step**: Incremental value
- **End**: Final value

Error Tolerance Option: If enabled, the tolerance values for SPICE simulation can be set manually.
- **RELTOL**: Relative error tolerance of the program.
- **TRTOL**: Transient error tolerance.
- **VNTOL**: Absolute voltage error tolerance of the program.
- **ABSTOL**: Absolute current error tolerance of the program.
- **CHGTOL**: Charge tolerance of the program

For SPICE analyses and options not included in the Simulation Control dialog, user must write SPICE commands and define the options in PSIM’s "SPICE Directive Block". This block will be explained in Section 2.4.2 SPICE Directive Block.

In such cases, SPICE simulation can not be run directly from PSIM’s menu. Users must use PSIM’s "Generate SPICE Netlist" function to generate the netlist from PSIM schematic. Then, before running SPICE simulation, remove from the netlist all the undesired analysis commands.
2.4 PSIM Elements for SPICE Simulation

Many PSIM schematic elements are supported for SPICE simulation. User can set PSIM’s display option to see which elements are supported. In the menu **Options >> Settings >> Advanced**, check the box **Show image next to elements that can be used for SPICE**. With this box checked, the elements supported by SPICE will be marked with the image 📈.

The SPICE netlist implementation of PSIM elements imitates the PSIM element characteristics. For example, if the PSIM element Resistor’s Model Level is selected as Level 1, its corresponding SPICE netlist is a single resistor; if it is selected as Level 2, its corresponding SPICE netlist would contain a resistor with a series inductor and a parallel capacitor.

2.4.1 Multi-Level Elements

Most of the 📈 marked elements can be used both in PSIM simulation and in SPICE simulation. However, some elements have multi-level options. Some levels may be used for both PSIM and SPICE simulation; some levels are for PSIM only while others are for SPICE only.

The advantage of multi-level elements is that, user may specify the level for desired simulation, and hence, the same schematic can run both PSIM and SPICE simulation.

For example, the element **Power >> Switches >> MOSFET** is a multi-level element. It has the following model levels:

- Ideal: may be used in both PSIM and SPICE simulation.
- Level 1: for SPICE simulation only.
- Level 2: for PSIM simulation only.
- SPICE Model: for SPICE simulation only.
- SPICE Subcircuit: for SPICE simulation only, 3-node: drain, gate, source
- SPICE Subcircuit (4-pin): for SPICE simulation only, 4-nodes: drain, gate(+), gate(-), and source
- SPICE Subcircuit (5-pin): for SPICE simulation only, 5-nodes: drain, gate, source, temp(+), and temp(-).

One can check the box "Select different element models for simulation" in the "Simulation Models" tab in this MOSFET’s parameter dialog, and specify different levels for "PSIM Model" and for "SPICE Model".

To help users checking if correct model level is selected for desired simulation, a function is provided. This function is in **Simulate >> Check Multi-Level Elements**. It will list elements in the circuit, as shown below.
The list can show different types of elements by selecting the options from the drop-down menu:

- **Show all**: List all the elements in the schematic.
- **Show only multi-level elements**: List only multi-level elements, and show the model level selected for PSIM simulation or SPICE simulation.
- **Show only elements that are not compatible with PSIM simulation**: List only the elements which are not supported for PSIM simulation.
- **Show only elements that are not compatible with SPICE simulation**: List only the elements which are not supported for SPICE simulation

If the checkbox **Highlight Elements** is checked, all the elements listed in display will be highlighted in the schematic.

The model levels of multi-level elements can be changed here directly, and it has the same effect as if they were changed through the property dialog window.

Elements can be enabled/disabled by checking/unchecking the **Enable** checkbox.

With this function, one can quickly identify and replace the elements which are not supported by the desired simulation engine.
2.4.2  SPICE Directive Block

It is not possible to implement all the SPICE elements and controls in PSIM schematic capture. PSIM’s SPICE module provides an element "SPICE Directive Block" for users to write the SPICE commands, options, models, subcircuit netlist, parametric attributes, and other directives which are not implemented by PSIM schematic capture. This block can be found in the menu Elements >> SPICE >> SPICE Directive Block.

Only one SPICE Directive block is allowed in each PSIM schematic. Therefore, all the SPICE directives must be collected together in one block. In the netlist captured from PSIM schematic, the content of this block will be placed at the top of the file.

The syntax in this block must follow NGspice netlist format. User should always check the PSIM-SPICE generated netlist to for SPICE netlist syntax errors before running simulation.

For the example shown in the picture below, the PSIM-SPICE captured SPICE netlist would include the content of the SPICE Directive Block:

```
.model AP9465GEM NMOS
  + Rg=2 Vto=1.8 Rd=0 Rs=16m Rb=10m
  + Kp=30 lambda=0.04 Cgs=600p Cjo=1.2n
  + Is=3p Vds=40 Ron=25m Qg=8.5n)
```
2.4.3 SPICE Subcircuit Netlist Block

PSIM’s SPICE Module provides an element "SPICE Subcircuit Netlist Block" for users to create or to use existing SPICE subcircuit netlist in PSIM. This block can be found in the menu Elements >> SPICE >> SPICE Subcircuit Netlist Block.

This block would generate a calling statement to the subcircuit. In this block, a user may specify the subcircuit’s name, number of nodes, parameter names, and parameter values. The definition and content of the subcircuit must be provided by either browsing the subcircuit files or writing in the SPICE Directive Block.

The subcircuit syntax in the SPICE Directive Block must follow SPICE netlist format for subcircuit, starting with .SUBCKT and ending with .ENDS. User should always check the PSIM-SPICE generated netlist for SPICE netlist syntax errors before running simulation.

For the example shown in the picture below, in the SPICE netlist, the calling of the subcircuit in PSIM-SPICE captured SPICE netlist would be:

```
XSPS2 3 5 10 Si4628DY
```

And the subcircuit content would be the same as written in the SPICE Directive Block:

```
.SUBCKT Si4628DY D G S
X1 D G S Si4628DY_nmos
X2 S D Si4628DY_schottky
.ENDS Si4628DY
```

... ...
2.5 Create PSIM Element from SPICE Netlist

Users may have accumulated many subcircuit netlists from previous works or from manufacturers’ model database. PSIM provides a convenient way for users to utilize those pre-existing netlists. One may create and maintain a SPICE element library as part of PSIM schematic element library.

Before the SPICE subcircuit elements can be entered into a PSIM element library, the path for the SPICE subcircuit netlist files must be included in PSIM’s search path.

Once the SPICE netlist file is entered into PSIM search path, one should remove it from the SPICE Directive Block in the PSIM circuit schematic.

A SPICE subcircuit element library can be created and maintained in the same way as an element in PSIM library. The following information are needed to create a PSIM library element from a SPICE subcircuit netlist:

<table>
<thead>
<tr>
<th>Name</th>
<th>The name of the subcircuit, must be the same as in the .subckt line in the netlist.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>A short description of the subcircuit.</td>
</tr>
<tr>
<td>Image</td>
<td>User may use PSIM’s Image Editor and define the subcircuit's image size and port positions connecting to main circuit. PSIM provides a few standard semiconductor images. User must make sure the port sequence is the same as in the subcircuit netlist.</td>
</tr>
<tr>
<td>Help File</td>
<td>The link to the optional html help file of the subcircuit.</td>
</tr>
<tr>
<td>Ports</td>
<td>A list of port names/numbers as defined in the netlist's .subckt statement.</td>
</tr>
<tr>
<td>Parameters</td>
<td>List of parameters and default values, must be the same as in the .subckt or .param statement.</td>
</tr>
<tr>
<td>File</td>
<td>The filename of the subcircuit file.</td>
</tr>
</tbody>
</table>

For example, if a SPICE netlist file "My SPICE Subckt.txt" contains some SPICE subcircuit netlists, and it is saved in the folder "C:\PSIM_SPICE Tutorial\SPICE Subs". To create a new library which contains those subcircuits:

- Include the folder "C:\PSIM_SPICE Tutorial\SPICE Subs" in the PSIM’s SPICE model search path.
- In the menu Edit >> Edit Library >> Edit Library Files, select New Library to create a new library for SPICE subcircuit elements.
- Type the library name to be displayed in PSIM's "Elements" menu: "User SPICE Elements".
- Type the library file name: "My SPICE File". Click OK to add this new file into the library list.
- Select this new library name "My SPICE File.lib", then, click the button for "Edit Selected Library". The PSIM's image library editor would open as shown below.
• Click the button for "New SPICE Element". All the models and subcircuits in the files which already are put into the PSIM searching path would be displayed.
• Double click the file name "My SPICE Subckt.txt" and all the subcircuits in this file would show.
• Double click on "Si4628DY", the editor for SPICE Library Element would open. In this editor, the subcircuit's definition, name, nodes, and parameters and their default values are parsed automatically.

• Write an optional brief description "SPICE Tutorial example MOSFET".
• Select the MOSFET image from the image list. Please make sure the node sequence in the image is the same as in the subcircuit definition.
• If a new image is desired, click "Edit Image" to edit this element's image: to set the size, to locate the nodes, to add texts, and to draw graphic designs for the image.
• It is optional to create a html formatted help file name "SPICE Tutorial MOSFET Help.html" in the folder "Powersim\Help", type the help file name in the space provided, and click the button "Test Help Page" to verify the link.
• Click "Save" button to save the element in the library, and click "OK" to close the SPICE element editor.
• In the Image Library editor, the new element "Si4628DY" is shown now.
• Click "Save Image Library" to update the library, then, click "Update Menu" button to update the PSIM "Element" menu for the display of this newly created element.

• Now this new PSIM element is ready to be used for in PSIM schematics for SPICE simulation.

• To insert this newly created PSIM element in a PSIM schematic for SPICE simulation, simply click on Elements >> User SPICE Elements >> Si4628DY, place it in the proper location in the schematic, and connect the wires.
3.1 Overview

The SPICE engine in CoolSPICE is based on NGSPICE [5]. PSIM captures circuit schematics and creates
netlists in standard SPICE syntax for which extensive documentation is available online [4]. The schematic
generates the netlist in the syntax set down here for each analysis type, described here. As described in Chapter
3, these commands can also be directly added to the netlist to run the relevant analyses.

This chapter describes SPICE analysis types, commands, and options.

3.2 Convergence

Sometimes SPICE simulation does not converge. When this occurs, try to change the type of the numerical
integration algorithm, or adjust the error tolerance values, or add snubber circuits to switches.

3.3 SPICE Analysis Types

All the analysis types and options supported in NGSPICE are supported by the SPICE module, as listed below.
More details and extensive documentation is available online [4].

The simulator supports the following different types of analysis:
- DC Analysis (Operating Point and DC Sweep)
- AC Small-Signal Analysis
- Transient Analysis
- Pole-Zero Analysis
- Small-Signal Distortion Analysis
- Sensitivity Analysis
- Noise Analysis

3.3.1 .AC

Perform a small signal ac analysis linearized based on a dc operating point.

Syntax:
.ac dec/oct/lin Nsteps Start_f End_f

Examples:
.ac dec 10 1 10k
.ac lin 100 1 100

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>oct/dec/lin</td>
<td>Specify if the analysis is in octave, decade, or linear variation</td>
</tr>
<tr>
<td>Nsteps</td>
<td>Number of steps in each octave, decade, or total number of linearly spaced steps between the starting and final frequency.</td>
</tr>
<tr>
<td>Start_f</td>
<td>Starting frequency</td>
</tr>
<tr>
<td>End_f</td>
<td>Final frequency</td>
</tr>
</tbody>
</table>
3.3.2 .DC

Perform dc analysis while sweeping the dc value of a source. It is useful for plotting the characteristic curves of an electronic component.

**Syntax:**
```
.dc src_name Vstart Vstop Vstep [src2 Vstart2 Vstop2 Vstep2]
```

**Examples:**
```
.dc Vin 0.25 5.0 0.25
.dc VDS 0 10 0.5 Vgs 0 5 1
```

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>src_name</td>
<td>Name of the independent source that is to be swept. It can be a voltage or current source.</td>
</tr>
<tr>
<td>Vstart</td>
<td>Starting value for the sweep</td>
</tr>
<tr>
<td>Vstop</td>
<td>Final value for the sweep</td>
</tr>
<tr>
<td>Vstep</td>
<td>Incremental value for the sweep</td>
</tr>
</tbody>
</table>

3.3.3 .END

It marks the end of the netlist. All data and every other command must come before it. All lines after this is ignored.

**Syntax:**
```
.END
```

3.3.4 .ENDS

It marks the end of a subcircuit definition. See SUBCKT for more information

**Syntax:**
```
.ENDS
```

3.3.5 .FOUR

This command controls whether SPICE performs a Fourier analysis as a part of the transient analysis. The Fourier analysis is performed over the one period interval before the transient analysis' final time.

**Syntax:**
```
.FOUR Freq Ov1 <Ov2 OV3 ... >
```

**Example:**
```
.FOUR 100K v(5)
```

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Freq</td>
<td>Fundamental frequency</td>
</tr>
<tr>
<td>Ov1, Ov2, ...</td>
<td>Desired vector(s) to be analyzed.</td>
</tr>
</tbody>
</table>

3.3.6 .FUNC

This directive allows the creation of user-defined functions for behavioral sources.

**Syntax:**
```
.FUNC Fname(args) { <expression> }
```

**Example:**
```
.FUNC icos(x) {cos(x)-1}
```
.FUNC f(x,y) {x*y}

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fname</td>
<td>Function name</td>
</tr>
<tr>
<td>args</td>
<td>Arguments in the function</td>
</tr>
<tr>
<td>expression</td>
<td>Mathematical expression of the function</td>
</tr>
</tbody>
</table>

3.3.7 .GLOBAL

This statement defines the nodes which are available to all circuit and subcircuit blocks independent from any circuit hierarchy.

**Syntax:**
```
.GLOBAL node1 <node node3 ... >
```

**Example:**
```
.GLOBAL VDD VCC
```

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>node1, node2, node3 ...</td>
<td>The nodes defined as global, to be accessible from the top level.</td>
</tr>
</tbody>
</table>

3.3.8 .IC

This line sets initial conditions for transient analysis.

**Syntax:**
```
.IC V(node1)=val <V(node2)=val I(inductor)=curr ... >
```

**Example:**
```
.IC V(in)=2 V(out)=5 V(12)=2.2 I(L4)=300m
```

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>V(node1)=val, V(node2)=val ...</td>
<td>Initial node voltage settings</td>
</tr>
<tr>
<td>I(inductor)=curr</td>
<td>Initial inductor current setting</td>
</tr>
</tbody>
</table>

3.3.9 .INCLUDE

This directive includes the named file as if that file has been typed into the netlist. There is no restriction on the file name beyond those imposed by the Windows operating system.

The absolute path should be entered for the file name. Otherwise, PSIM will look in the paths set by the "Set Path" option.

**Syntax:**
```
.INCLUDE path\filename
```

**Example:**
```
.INCLUDE C:\PSIM_SPICE Tutorial\SPICE Subs\LC_FILTER.spicesub
```

3.3.10 .LIB

This directive includes the named library as if that file has been typed into the netlist. There is no restriction on the file name beyond those imposed by the local operating system.

The absolute path name should be entered for the file name. Otherwise, PSIM will look in the paths set by the "Set Path" option.
Syntax:
.LIB path\filename

Example:
.LIB C:\LTC\lib\cmp\standard.bjt

3.3.11 .MODEL

This directive defines a model for SPICE components.

Syntax:
.MODEL Mname Type (param1=val param2=val ...)

Example:
.MODEL QT1 npn (bf=50 is=1e-13 vbf=50)

Keyword Description
Mname Model name, must be unique for each type of circuit element. For example, a diode and a transistor can not have the same model name.
Type Model type as listed in the table below.
Param1=val Parameters and their values for the model. Some models are highly complicated with a long list of parameters. Please refer to the reference documents for comprehensive and detailed model parameter descriptions and definitions.
param2=val ...

<table>
<thead>
<tr>
<th>Model Type</th>
<th>Associated Circuit Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>Semiconductor resistor model</td>
</tr>
<tr>
<td>C</td>
<td>Semiconductor capacitor model</td>
</tr>
<tr>
<td>L</td>
<td>Inductor model</td>
</tr>
<tr>
<td>SW</td>
<td>Voltage controlled switch</td>
</tr>
<tr>
<td>CSW</td>
<td>Current controlled switch</td>
</tr>
<tr>
<td>URC</td>
<td>Uniform distributed RC line</td>
</tr>
<tr>
<td>LTRA</td>
<td>Lossy transmission line</td>
</tr>
<tr>
<td>D</td>
<td>Diode</td>
</tr>
<tr>
<td>NPN</td>
<td>NPN bipolar transistor</td>
</tr>
<tr>
<td>PNP</td>
<td>PNP bipolar transistor</td>
</tr>
<tr>
<td>NJF</td>
<td>N-channel JFET model</td>
</tr>
<tr>
<td>PJF</td>
<td>P-channel JFET model</td>
</tr>
<tr>
<td>NMOS</td>
<td>N-channel MOSFET</td>
</tr>
<tr>
<td>PMOS</td>
<td>P-channel MOSFET</td>
</tr>
<tr>
<td>NMF</td>
<td>N-channel MESFET</td>
</tr>
<tr>
<td>PMF</td>
<td>P-channel MESFET</td>
</tr>
<tr>
<td>VDMOS</td>
<td>Vertical double diffused power MOSFET</td>
</tr>
</tbody>
</table>

3.3.12 .NODESET

This directive is used to hold specified node at given voltage values. This would help the SPICE program to find the dc or initial transient solution. The holding is released for subsequent iterative passes.

Syntax:
.NODESET V(node1)=val <V(node2) ... >
.NODESET ALL=val
Example:
  .NODESET V(in)=2 V(out)=5 V(12)=2.2 I(L4)=300m
  .NODESET ALL=1.5

Keyword | Description
--------|-----------------|
V(node1=val, V(node2)=val ...) | Initial node voltage settings
ALL=val | Initial voltage setting for all nodes except the ground node

### 3.3.13 .NOISE

This function is used in frequency domain analysis to compute the noise spectral density.

**Syntax:**
  .NOISE V(Nout Nref) source dec/lin/oct Nstep Fstart Fend

**Example:**
  . NOISE V(5) VIN dec 10 1kHZ 100Mhz
  . NOISE V(5,3) V1 oct 8 1.0 1.0e6 1

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>V(Nout Nref)</td>
<td>Noise voltage V(Nout)-V(Nref). Nout is the output node, and Nref is the reference node which is ground by default.</td>
</tr>
<tr>
<td>source</td>
<td>Independent source to which input noise is referred.</td>
</tr>
<tr>
<td>dec/oct/lin</td>
<td>Specify if the analysis is in octave, decade, or linear variation</td>
</tr>
<tr>
<td>Nstep</td>
<td>Number of points in the range from Fstart to Fend</td>
</tr>
<tr>
<td>Fstart</td>
<td>Starting frequency</td>
</tr>
<tr>
<td>Fend</td>
<td>Final frequency</td>
</tr>
</tbody>
</table>

### 3.3.14 .OP

Operating point analysis. This directive determines the dc operating point of a circuit with all inductors shorted and all capacitors opened.

**Syntax:**
  .OP

**Example:**
  .OP

### 3.3.15 .OPTIONS

This command sets simulator options. It allows users to set options for specified simulation purpose.

**Syntax:**
  .OPTIONS opt1 opt2 ...
  .OPTIONS opt1=val opt2=val ...

**Example:**
  .OPTIONS TELTOL=0.005 TRTOL=8

The table below lists some often used options. Many of these options can be set in PSIM’s Simulation Control dialog under the tab SPICE.
For a complete option list, please refer to the reference documents listed at the end of this document.

### 3.3.16 .PARAM

This directive allows the creation of user-defined variables. It is useful for associating a name with a value for the sake of clarity and parameterizing subcircuits.

**Syntax:**

```
.PARAM param1=expr1 param2=expr2...
```

**Example:**

```
.PARAM po=6 pp=7.8 pop=10k
```

The table below lists the built-in operators available to be used in .PARAM lines.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Precedence</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>~</td>
<td>~</td>
<td>Unary ~</td>
</tr>
<tr>
<td>!</td>
<td>1</td>
<td>Unary not</td>
</tr>
<tr>
<td>**</td>
<td>2</td>
<td>Power, same as PWR</td>
</tr>
<tr>
<td>^</td>
<td>2</td>
<td>Power, same as PWR</td>
</tr>
<tr>
<td>*</td>
<td>3</td>
<td>Multiply</td>
</tr>
<tr>
<td>/</td>
<td>3</td>
<td>Divide</td>
</tr>
<tr>
<td>%</td>
<td>3</td>
<td>Modulo</td>
</tr>
<tr>
<td>\</td>
<td>3</td>
<td>Integer divide</td>
</tr>
<tr>
<td>+</td>
<td>4</td>
<td>Add</td>
</tr>
<tr>
<td>-</td>
<td>4</td>
<td>Subtract</td>
</tr>
<tr>
<td>==</td>
<td>5</td>
<td>Equality</td>
</tr>
<tr>
<td>!=</td>
<td>5</td>
<td>Not equal</td>
</tr>
<tr>
<td>&lt;&gt;</td>
<td>5</td>
<td>Not equal</td>
</tr>
<tr>
<td>&lt;=</td>
<td>5</td>
<td>Less or equal</td>
</tr>
<tr>
<td>=&gt;</td>
<td>5</td>
<td>Greater or equal</td>
</tr>
<tr>
<td>&gt;</td>
<td>5</td>
<td>Less than</td>
</tr>
<tr>
<td>&lt;</td>
<td>5</td>
<td>Greater than</td>
</tr>
<tr>
<td>&amp;&amp;</td>
<td>6</td>
<td>Boolean AND</td>
</tr>
<tr>
<td>&amp;</td>
<td>6</td>
<td>Boolean AND</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTOL</td>
<td>Absolute current error tolerance</td>
<td>1 pA</td>
</tr>
<tr>
<td>CHGTOL</td>
<td>Absolute charge tolerance</td>
<td>10 fC</td>
</tr>
<tr>
<td>MAXSTEP</td>
<td>Maximum step size for transient analysis</td>
<td>Infinity</td>
</tr>
<tr>
<td>METHOD</td>
<td>Numerical integration method. The choices are trapezoidal, modified trapezoidal, and Gear.</td>
<td>Trapezoidal</td>
</tr>
<tr>
<td>RELTOL</td>
<td>Relative error tolerance</td>
<td>0.001</td>
</tr>
<tr>
<td>TEMP</td>
<td>Default temperature for circuit element instances that do not specify temperature</td>
<td>27 degree C</td>
</tr>
<tr>
<td>TRTOL</td>
<td>Transient error tolerance. This parameter is an estimate of factor by which the actual truncation error is overestimated.</td>
<td>1</td>
</tr>
<tr>
<td>VNTOL</td>
<td>Absolute voltage error tolerance.</td>
<td>1uV</td>
</tr>
</tbody>
</table>
The table below lists the built-in functions available to be used in .PARAM lines:

<table>
<thead>
<tr>
<th>Built-in Function</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>sqr(x)</td>
<td>y=x*x</td>
</tr>
<tr>
<td>sqrt(x)</td>
<td>y=sqrt(x)</td>
</tr>
<tr>
<td>sin(x), cos(x), tan(x)</td>
<td></td>
</tr>
<tr>
<td>asin(x), acos(x), atan(x)</td>
<td></td>
</tr>
<tr>
<td>sinh(x), cosh(x), tanh(x)</td>
<td></td>
</tr>
<tr>
<td>asinh(x), acosh(x), atanh(x)</td>
<td></td>
</tr>
<tr>
<td>arctan(x)</td>
<td></td>
</tr>
<tr>
<td>exp(x)</td>
<td></td>
</tr>
<tr>
<td>ln(x), log(x)</td>
<td>ln(x) is not recognized. must use log(x)</td>
</tr>
<tr>
<td>abs(x)</td>
<td></td>
</tr>
<tr>
<td>nint(x)</td>
<td>Nearest integer, half integers towards even</td>
</tr>
<tr>
<td>int(x)</td>
<td>Nearest integer rounded towards 0</td>
</tr>
<tr>
<td>floor(x)</td>
<td>Nearest integer rounded towards negative infinity</td>
</tr>
<tr>
<td>ceil(x)</td>
<td>Nearest integer rounded towards positive infinity</td>
</tr>
<tr>
<td>pow(x,y)</td>
<td>Same as x**y or x^y</td>
</tr>
<tr>
<td>pwr(x,y)</td>
<td>power(fabs(x),y)</td>
</tr>
<tr>
<td>min(x,y)</td>
<td></td>
</tr>
<tr>
<td>max(x,y)</td>
<td></td>
</tr>
<tr>
<td>sgn(x)</td>
<td>1.0 for x&gt;0; 0 for x==0; -1 for x&lt;0</td>
</tr>
</tbody>
</table>

The scaling suffixes (any decorative alphanumeric string may follow):

<table>
<thead>
<tr>
<th>Suffix</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>g</td>
<td>10e9</td>
</tr>
<tr>
<td>meg</td>
<td>1e6</td>
</tr>
<tr>
<td>k</td>
<td>1e3</td>
</tr>
<tr>
<td>m</td>
<td>1e-3</td>
</tr>
<tr>
<td>u</td>
<td>1e-6</td>
</tr>
<tr>
<td>n</td>
<td>1e-9</td>
</tr>
<tr>
<td>p</td>
<td>1e-12</td>
</tr>
<tr>
<td>f</td>
<td>1e-15</td>
</tr>
</tbody>
</table>

### 3.3.17 .SAVE

This directive names the vector(s) to be saved in raw data file.

Note that for PSIM, if no .SAVE is in the netlist, no result will be recorded in the raw file. User should use either the probes in PSIM schematic or write .SAVE statement in the netlist in order to keep the simulation result data.

**Syntax:**

```
.SAVE vector1 vector2 ...
```

**Example:**
3.3.18 .STEP

This command is used for parameter sweep. It causes an analysis to be repeatedly performed while stepping the specific parameter.

In PSIM, this command must be used together with .PARAM in order to define the sweep parameter. The is done in the Simulation Control dialog under SPICE tab by checking the box for "Step Run Option" and fill in the parameters.

Syntax:

```
.PARAM Param_name=0
.STEP Param_name Vstart Vend Vstep
```

Example:

```
.PARAM Rswp=0
.STEP Rswp 10 20 1
...R1 3 0 {Rswp}
...R2 4 5 {Rswp}
...R3 6 7 {Rswp}
```

Keyword Description

- **Param_name**: Name of the parameter to be swept. In the example, it is `Rswp`.
- **Vstart**: Starting value.
- **Vend**: Final value.
- **Vstep**: Increment value.

3.3.19 .SUBCKT

This directive starts a SPICE subcircuit definition netlist. The end of a subcircuit definition must be a .ENDS directive.

Syntax:

```
Xsub_calling node1 node2 ...Sub_name param1=val param2=val ...
.SUBCKT Sub_name node1 node2 ... param1=dval param2=dval ...
```

Example:

```
* The following line is the calling for the subcircuit:
xdiv1 10 7 0 vdivide
* The following lines are the subcircuit definition:
.SUBCKT vdivide 1 2 3
  r1 1 2 10K
  r2 2 3 5K
  .ENDS vdivider
```

Keyword Description

- **Xsub_calling**: Circuit component which calls for the subcircuit.
- **param1=val**: Subcircuit parameters. In the `Xsub_calling` line, they are the values used in circuit for the simulation. If one or more parameters are omitted, their default values defined in the subcircuit definition will be used.
- **Sub_name**: Name of the subcircuit.
- **node1 node2 ...**: Subcircuit nodes.
- **param1=dval**: Parameters. In the subcircuit definition, they are default values.
3.3.20 .TRAN

This command performs transient analysis of the circuit.

**Syntax:**

```
.TRAN Tstep Tstop <Tstart <Tmax> > <UIC>
```

**Example:**

```
.TRAN 10n 1m
.TRAN 1n 100n UIC
```

<table>
<thead>
<tr>
<th><strong>Keyword</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Tstep</td>
<td>Suggested computing increment.</td>
</tr>
<tr>
<td>Tstop</td>
<td>Final time of the simulating</td>
</tr>
<tr>
<td>Tstart</td>
<td>Initial time of the simulation. Optional. If omitted, it is assumed to be zero. The transient analysis always starts at time zero. In the interval from zero to Tstart, the circuit is analyzed but no results are stored.</td>
</tr>
<tr>
<td>Tmax</td>
<td>Maximum time step size for simulation. If omitted, as default, (Tstop-Tstart)/50 is used.</td>
</tr>
<tr>
<td>UIC</td>
<td>Use initial conditions. This option indicates that the user does not want SPICE to solve for the quiescent operating point before beginning the transient analysis. Then, SPICE uses the values specified with .IC control line of IC=val in various elements as the initial condition.</td>
</tr>
</tbody>
</table>
4.1 Overview

This chapter describes the most common netlist format of the elements and device models in SPICE simulation. For more details and complex elements, such as how to specify temperature dependency or how to define a resistor’s semiconductor model, the information can be found in the SPICE manuals in the references listed at the end of this document.

For the SPICE elements which has not yet implemented with PSIM schematic elements, user may insert them in PSIM schematics as subcircuit blocks using PSIM’s "SPICE Subcircuit Netlist Block" and "SPICE Directive Block".

4.2 Passive Elements

The following passive elements are described in this section:

<table>
<thead>
<tr>
<th>SPICE Element</th>
<th>PSIM Schematic Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resistor</td>
<td>Resistor (Level 1)</td>
</tr>
<tr>
<td>Inductor</td>
<td>Inductor (Level 1)</td>
</tr>
<tr>
<td>Capacitor</td>
<td>Capacitor (Level 1), Capacitor (electrolytic)</td>
</tr>
<tr>
<td>Coupled Inductor</td>
<td>Coupled Inductor (2)</td>
</tr>
</tbody>
</table>

4.2.1 Resistor

This element defines a linear resistor between nodes N1 and N2.

**General Form:**

Rname N1 N2 Rvalue

**Examples:**

- R1 1 2 25
- R2 3 4 10k

**Arguments**

- **Rname**: Name of the resistor in the circuit
- **N1**: Positive element node
- **N2**: Negative element node
- **Rvalue**: Resistance, in Ohm. It must not be zero.

4.2.2 Capacitor

This element defines a linear capacitor between nodes N1 and N2.

**General Form:**

Cname N1 N2 Cvalue <IC=value>

**Examples:**

- C1 1 2 1u
- C2 3 4 10u IC=4V
4.2.3 Inductor

This element defines a linear inductor between nodes N1 and N2.

**General Form:**

```
Lname N1 N2 Lvalue <IC=value>
```

**Examples:**

```
L1 1 2 1m
L2 3 4 5m IC=2
```

<table>
<thead>
<tr>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lname</td>
<td>Name of the inductor in the circuit.</td>
</tr>
<tr>
<td>N1</td>
<td>Positive element node</td>
</tr>
<tr>
<td>N2</td>
<td>Negative element node</td>
</tr>
<tr>
<td>Lvalue</td>
<td>Inductance, in Henry, must not be zero.</td>
</tr>
<tr>
<td>&lt;IC=value&gt;</td>
<td>Optional initial condition of inductor current, in ampere.</td>
</tr>
</tbody>
</table>

4.2.4 Coupled Inductor

This element defines a coupled (mutual) inductor between inductors Lname1 and Lname2.

**General Form:**

```
Kname Lname1 Lname2 Coupling_value
```

**Examples:**

```
K1 L1 L2 0.9
```

<table>
<thead>
<tr>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kname</td>
<td>Name of the coupling inductors</td>
</tr>
<tr>
<td>Lname1</td>
<td>Name of the first coupled inductor</td>
</tr>
<tr>
<td>Lname2</td>
<td>Name of the second coupled inductor</td>
</tr>
<tr>
<td>Coupling_value</td>
<td>Coefficient of coupling. It must be greater than 1 and less than or equal to 1.</td>
</tr>
</tbody>
</table>
4.3 Transmission Lines

The following transmission line elements are described in this section:

- Lossless transmission line
- Lossy transmission line
- Uniform distributed RC line

Although these transmission lines are not represented with PSIM schematic elements, user may insert them in PSIM schematics as subcircuit blocks using PSIM’s "SPICE Subcircuit Netlist Block" and "SPICE Directive Block".

4.3.1 Lossless Transmission Line

This element defines a lossless transmission line between port 1 and port 2.

**General Form:**

```
Tname N1 N2 N3 N4 Z0=value <TD=value> <F=value> <NL=value> <IC=V1,I1,V2,I2>
```

**Examples:**

```
T1 1 2 0 20 Z0=50 TD=5ns
```

**Arguments**

<table>
<thead>
<tr>
<th><strong>Description</strong></th>
<th><strong>Name</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of the lossless transmission line</td>
<td>Tname</td>
</tr>
<tr>
<td>Nodes at port 1</td>
<td>N1, N2</td>
</tr>
<tr>
<td>Nodes at port 2</td>
<td>N3, N4</td>
</tr>
<tr>
<td>Characteristic impedance, in ohm</td>
<td>Z0=value</td>
</tr>
<tr>
<td>Transmission delay, in second (optional)</td>
<td>&lt;TD=value&gt;</td>
</tr>
<tr>
<td>Frequency to calculate the wavelength (optional)</td>
<td>&lt;F=value&gt;</td>
</tr>
<tr>
<td>Normalized electrical length of the transmission line with respect to the wavelength in the line at the frequency given above (optional)</td>
<td>&lt;NL=value&gt;</td>
</tr>
<tr>
<td>Initial condition of the voltage and current at each of the transmission line ports (optional)</td>
<td>&lt;IC=V1, I1, V2, I2&gt;</td>
</tr>
</tbody>
</table>

4.3.2 Lossy Transmission Line

This element defines a lossy transmission line between port 1 and port 2.

**General Form:**

```
Oname N1 N2 N3 N4 model_name
```

**Examples:**

```
O12 3 5 4 5 Model_Lossy
```

**Arguments**

<table>
<thead>
<tr>
<th><strong>Description</strong></th>
<th><strong>Name</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of the lossy transmission line</td>
<td>Oname</td>
</tr>
<tr>
<td>Nodes at port 1</td>
<td>N1, N2</td>
</tr>
<tr>
<td>Nodes at port 2</td>
<td>N3, N4</td>
</tr>
<tr>
<td>Model name of the lossy transmission line. The model description is in the reference list at the end of this document.</td>
<td>model_name</td>
</tr>
</tbody>
</table>
4.3.3 Uniform Distributed RC Line

This element defines a uniformly distributed RC line between nodes N1 and N2, with the capacitance connected to node N3.

**General Form:**

\[ \text{Uname N1 N2 N3 model\_name } l=\text{len} <n=\text{lumps}> \]

**Examples:**

\[ \text{U1 1 2 3 Model\_UniRC} \]

<table>
<thead>
<tr>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uname</td>
<td>Name of the uniform distributed RC line</td>
</tr>
<tr>
<td>N1, N2</td>
<td>Nodes connect RC line</td>
</tr>
<tr>
<td>N3</td>
<td>Node connects the capacitance</td>
</tr>
<tr>
<td>model_name</td>
<td>Model name of the lossy transmission line. The model parameters and descriptions can be found in the reference list at the end of this document.</td>
</tr>
<tr>
<td>l=\text{len}</td>
<td>Length of the RC line, in meter.</td>
</tr>
<tr>
<td>&lt;n=\text{lumps}&gt;</td>
<td>Lumped segments to use in modeling the RC line (optional). The model description is in the reference list at the end of this document.</td>
</tr>
</tbody>
</table>
4.4 Active Elements

The following active elements are described in this section:

<table>
<thead>
<tr>
<th>SPICE Element</th>
<th>PSIM Schematic Element</th>
<th>PSIM Model Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voltage-controlled switch</td>
<td>Diode</td>
<td>SPICE Model</td>
</tr>
<tr>
<td>Current-controlled switch</td>
<td>Diode</td>
<td>SPICE Model</td>
</tr>
<tr>
<td>Diode</td>
<td>npn Transistor (model)</td>
<td>SPICE Model</td>
</tr>
<tr>
<td>BJT</td>
<td>pnp Transistor (model)</td>
<td>SPICE Model</td>
</tr>
<tr>
<td>MOSFET</td>
<td>MOSFET (model)</td>
<td>SPICE Model</td>
</tr>
<tr>
<td>JFET</td>
<td>p-MOSFET (model)</td>
<td>SPICE Model</td>
</tr>
</tbody>
</table>

For the SPICE elements which do not yet have corresponding PSIM element representation, one may insert them in PSIM schematics as subcircuit blocks using PSIM’s "SPICE Subcircuit Netlist Block" and "SPICE Directive Block".

If a desired semiconductor device model is not included in PSIM’s SPICE model library, one may either write the model in "SPICE Directive Block" or include the model library file as described in Section 2.5 Create PSIM Element from SPICE Netlist.

4.4.1 Voltage-Controlled Switch

This element defines a voltage controlled switch between nodes N1 and N2 controlled by the voltage between nodes NC1 and NC2.

**General Form:**

\[ \text{Sname N1 N2 NC1 NC2 model}\_\text{name } \text{<ON/OFF>} \]

**Examples:**

S1 1 2 3 4 Smod ON

**Arguments** | **Description**
---|---
Sname | Name of the switch
N1, N2 | Nodes at the two terminals of the switch
NC1, NC2 | Positive and negative terminals of the controlling voltage
model\_name | Name of the model for the switch
<ON/OFF> | Initial state of the switch. Required when the controlling voltage starts inside the range of hysteresis loop. Optional otherwise.

4.4.2 Current-controlled switch

This element defines a current controlled switch between nodes N1 and N2 controlled by the current flowing through the voltage source Vname.

**General Form:**

\[ \text{Wname N1 N2 Vname model}\_\text{name } \text{<ON/OFF>} \]

**Examples:**

W1 1 2 3 4 Wmod ON

**Arguments** | **Description**
---|---
Sname | Name of the switch
N1, N2 | Nodes at the two terminals of the switch
4.4.3 Controlled Switch Model

A switch model defines a nearly ideal switch. In SPICE simulation, a switch cannot be ideal to switch from 0 to infinite resistance. A finite positive value must be assigned for on and off conditions.

Voltage Controlled Switch Model General Form:
.model model_name sw (vt=value1 vh=value2 ron=value3 roff=value4)

Current Controlled Switch Model General Form:
.model model_name csw (it=value1 ih=value2 ron=value3 roff=value4)

Example:
Vm3 14 0 dc 0 ; voltage source for W1 control current
S1 10 1 0 Sw1 off
W1 20 21 Vm3 Wsw1 off
.model Sw1 sw vt=1 vh=0.2 ron=1m roff=10meg; model for voltage controlled switch S1
.model Wsw1 csw it=1m ih=0.2m ron=0.01 roff=10meg; model for current controlled switch W1

The model parameters are listed in the table below.

<table>
<thead>
<tr>
<th>Name</th>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>vt</td>
<td>Threshold voltage</td>
<td>V</td>
<td>0.0</td>
<td>SW</td>
</tr>
<tr>
<td>vh</td>
<td>Hysteresis voltage</td>
<td>V</td>
<td>0.0</td>
<td>SW</td>
</tr>
<tr>
<td>it</td>
<td>Threshold current</td>
<td>A</td>
<td>0.0</td>
<td>CSW</td>
</tr>
<tr>
<td>ih</td>
<td>Hysteresis current</td>
<td>A</td>
<td>0.0</td>
<td>CSW</td>
</tr>
<tr>
<td>ron</td>
<td>ON resistance</td>
<td>ohm</td>
<td>1.0</td>
<td>SW, CSW</td>
</tr>
<tr>
<td>roff</td>
<td>Off resistance</td>
<td>ohm</td>
<td>1.0e-12</td>
<td>SW, CSW</td>
</tr>
</tbody>
</table>

4.4.4 Diode

This element defines a PN junction diode between nodes N1 and N2.

General Form:
Dname N1 N2 model_name <area=value> <ON/OFF>

Examples:
D1 1 2 Dmod

Arguments Description
Dname Name of the diode
N1 Positive (anode) node
N2 Negative (cathode) node
model_name Name of the model for the diode
<ON/OFF> Initial state of the device for DC operating point analysis (optional)

Diode Model General Form:
.model model_name d (param1=pval1 param2=pval2 ... )

Example:
.model DMOD D (bf=50 is=1.2e-13 vbf=50)

The junction diode is the basic switching device and the simplest one modeled in SPICE. However, its model is
highly complicated. Please refer to the reference documents for comprehensive and detailed model parameter descriptions and definitions.

4.4.5 Bipolar Junction Transistor (BJT)

This element defines either a npn junction transistor or a pnp junction transistor.

**General Form:**

Qname NC NB NE <NS> model_name <area=value> <OFF>

**Examples:**

Q1 1 2 3 Qmod

<table>
<thead>
<tr>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Qname</td>
<td>Name of the transistor</td>
</tr>
<tr>
<td>NC</td>
<td>Transistor collector node</td>
</tr>
<tr>
<td>NB</td>
<td>Transistor base node</td>
</tr>
<tr>
<td>NE</td>
<td>Transistor emitter node</td>
</tr>
<tr>
<td>&lt;NS&gt;</td>
<td>Transistor substrate node (optional)</td>
</tr>
<tr>
<td>model_name</td>
<td>Name of the transistor model.</td>
</tr>
<tr>
<td>&lt;area=value&gt;</td>
<td>Area factor. Default value is 1 (optional)</td>
</tr>
<tr>
<td>&lt;ON/OFF&gt;</td>
<td>Initial state of the device for DC operating point analysis (optional)</td>
</tr>
</tbody>
</table>

**BJT Model General Form:**

.model model_name npn (param1=pval1 param2=pval2 ... )

.model model_name pnp (param1=pval1 param2=pval2 ... )

**Example:**

.model QMOD NPN (level=2)

The SPICE models for bipolar junction transistors are highly complicated. Please refer to the reference documents for comprehensive and detailed model parameter descriptions and definitions.

4.4.6 MOSFET

This element defines either an N-channel MOSFET or a P-channel MOSFET.

**General Form:**

Mname ND NG NS NB model_name <instance parameters>

**Examples:**

M1 1 2 3 0 Mmod L=1u W=2

<table>
<thead>
<tr>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mname</td>
<td>Name of the MOSFET</td>
</tr>
<tr>
<td>ND</td>
<td>MOSFET drain node</td>
</tr>
<tr>
<td>NG</td>
<td>MOSFET gate node</td>
</tr>
<tr>
<td>NS</td>
<td>MOSFET source node</td>
</tr>
<tr>
<td>NB</td>
<td>MOSFET bulk node</td>
</tr>
<tr>
<td>model_name</td>
<td>Name of the MOSFET model.</td>
</tr>
<tr>
<td>&lt;instance parameters&gt;</td>
<td>Instance parameters defining the MOSFET instance (optional)</td>
</tr>
</tbody>
</table>

**MOSFET Model General Form:**

.model model_name nmos (param1=pval1 param2=pval2 ... )

.model model_name pmos (param1=pval1 param2=pval2 ... )
Example:
   .model Mmod NMOS (level=3)

The SPICE models for MOSFET devices are highly complicated. Please refer to the reference documents for comprehensive and detailed model parameter descriptions and definitions.

### 4.4.7 Junction Field-Effect Transistor (JFET)

This element defines either an N-channel JFET or a P-channel JFET.

**General Form:**

\[ \text{Jname ND NG NS model\_name <area> <OFF>} \]

**Examples:**

   J1 1 2 3 0 Jmod off

<table>
<thead>
<tr>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jname</td>
<td>Name of the JFET</td>
</tr>
<tr>
<td>ND</td>
<td>JFET drain node</td>
</tr>
<tr>
<td>NG</td>
<td>JFET gate node</td>
</tr>
<tr>
<td>NS</td>
<td>JFET source node</td>
</tr>
<tr>
<td>model_name</td>
<td>Name of the JFET model</td>
</tr>
<tr>
<td>&lt;area&gt;</td>
<td>JFET area factor (optional)</td>
</tr>
<tr>
<td>&lt;OFF&gt;</td>
<td>Initial condition for DC operating point analysis (optional)</td>
</tr>
<tr>
<td>&lt;instance parameters&gt;</td>
<td>Instance parameters defining the JFET instance (optional)</td>
</tr>
</tbody>
</table>

**JFET Model General Form:**

   .model model\_name NJF (param1=pval1 param2=pval2 ... )
   .model model\_name PJF (param1=pval1 param2=pval2 ... )

**Example:**

   .model JMOD NJF (RD=80)

The SPICE models for JFET devices are highly complicated. Please refer to the reference documents for comprehensive and detailed model parameter descriptions and definitions.

### 4.4.8 MESFET

This element defines either an N-channel MESFET or a P-channel MESFET.

**General Form:**

\[ \text{Zname ND NG NS model\_name <OFF>} \]

**Examples:**

   Z1 1 2 3 Zmod

<table>
<thead>
<tr>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zname</td>
<td>Name of the MESFET</td>
</tr>
<tr>
<td>ND</td>
<td>MESFET drain node</td>
</tr>
<tr>
<td>NG</td>
<td>MESFET gate node</td>
</tr>
<tr>
<td>NS</td>
<td>MESFET source node</td>
</tr>
<tr>
<td>model_name</td>
<td>Name of the JFET model</td>
</tr>
<tr>
<td>&lt;OFF&gt;</td>
<td>Initial condition for DC operating point analysis (optional)</td>
</tr>
<tr>
<td>&lt;instance parameters&gt;</td>
<td>Instance parameters defining the JFET instance (optional)</td>
</tr>
</tbody>
</table>
MESFET Model General Form:

```
.model model_name NMF (param1=pval1 param2=pval2 ... )
.model model_name PMF (param1=pval1 param2=pval2 ... )
```

**Example:**

```
.model Zmod NMF (level=1 rd=46)
```

The SPICE models for MESFET devices are complicated. Please refer to the reference documents for comprehensive and detailed model parameter descriptions and definitions.
4.5 Sources

The following sources are described in this section:

- Independent voltage source
- Independent current source
- Voltage-controlled voltage source
- Current-controlled current source
- Voltage-controlled current source
- Current-controlled voltage source

The following passive elements are described in this section:

<table>
<thead>
<tr>
<th>SPICE Elements</th>
<th>PSIM Schematic Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Independent Sources:</strong></td>
<td></td>
</tr>
<tr>
<td>Pulse</td>
<td>Square-wave and Triangular-wave voltage and current sources, Sawtooth-wave voltage source</td>
</tr>
<tr>
<td>Sinusoidal</td>
<td>Sinusoidal voltage and current sources</td>
</tr>
<tr>
<td>Exponential</td>
<td></td>
</tr>
<tr>
<td>Piece-wise linear</td>
<td>Piece-wise linear, Piece-wise linear (in pair), Step, and Step (2 level) voltage and current sources</td>
</tr>
<tr>
<td>Single frequency FM</td>
<td></td>
</tr>
<tr>
<td>Amplitude modulated source</td>
<td></td>
</tr>
<tr>
<td>Transient noise source</td>
<td></td>
</tr>
<tr>
<td>Random voltage source</td>
<td>Random voltage and current sources</td>
</tr>
<tr>
<td><strong>Linear Dependent Sources</strong></td>
<td></td>
</tr>
<tr>
<td>Linear voltage-controlled current source</td>
<td>Voltage-controlled current source</td>
</tr>
<tr>
<td>Linear voltage-controlled voltage source</td>
<td>Voltage-controlled voltage source</td>
</tr>
<tr>
<td>Linear current-controlled current source</td>
<td>Current-controlled (flowing through) current source</td>
</tr>
<tr>
<td>Linear current controlled voltage source</td>
<td>Current-controlled (flowing through) current source</td>
</tr>
<tr>
<td>Polynomial source</td>
<td></td>
</tr>
<tr>
<td><strong>Nonlinear Dependent Sources (Behavioral Sources)</strong></td>
<td></td>
</tr>
<tr>
<td>Nonlinear dependent source</td>
<td>Variable-gain voltage-controlled, Nonlinear (multiplication), Nonlinear (division), and Nonlinear (square root) voltage and current sources. Power and Math expression voltage sources. Polynomial and Polynomial (1) current sources</td>
</tr>
<tr>
<td>Nonlinear voltage source</td>
<td></td>
</tr>
<tr>
<td>Nonlinear current source</td>
<td></td>
</tr>
</tbody>
</table>

For the SPICE elements which do not yet have corresponding PSIM element representations, one may insert them in PSIM schematics as subcircuit blocks using PSIM’s "SPICE Subcircuit Netlist Block" and "SPICE Directive Block".

4.5.1 Independent voltage and current sources

The independent voltage source is between node N1 and N2 while the independent current source flows from N1 to N2.

**General Form:**

\[
\text{Vname N1 N2 <DC DC_TRvalue> <AC ACamplitude ACphase> <other options>}
\]

\[
\text{Iname N1 N2 <DC DC_TRvalue> <AC ACamplitude ACphase> <other options>}
\]
Examples:
Vcc 10 0 DC 12
Vin 1 2 AC 110 120

Arguments | Description
---|---
Vname, Iname | Name of the voltage and current sources
N1 | Positive voltage source node
N1 | Negative voltage source node
<DC DC.TRvalue> | Optional keyword for the DC value of the source. DC.TRvalue is the DC voltage value for DC and transient analysis, in V
<AC AC.amplitude AC.phase> | Optional keyword for the AC value of the source. AC.amplitude is the AC peak amplitude of the source used during AC analysis, in V. AC.phase is the AC phase of the source used only in AC analysis, in degrees.
<other options> | Options to define the following types of the voltage sources for transient analysis:
- Pulse
- Exponential
- Sinusoidal
- Piecewise linear
- Single-frequency FM
- AM
- Transient noise
- Random, and
- External data (only with NGSPICE shared library)

If the voltage source must have the options which do not yet have corresponding PSIM element representations, user must use a subcircuit block to define the voltage source in PSIM schematics using PSIM's "SPICE Subcircuit Netlist Block" and "SPICE Directive Block".

4.5.1.1 Pulse

General Form:
PULSE(V1 V2 TD TR TF PW PER)

Example:
Vin 3 0 PULSE(0 10 2u 1u 1u 50u 100u)

<table>
<thead>
<tr>
<th>Name</th>
<th>Parameter</th>
<th>Default Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>Initial value</td>
<td>--</td>
<td>V, A</td>
</tr>
<tr>
<td>V2</td>
<td>Pulsed value</td>
<td>--</td>
<td>V, A</td>
</tr>
<tr>
<td>TD</td>
<td>Delay time</td>
<td>0.0</td>
<td>second</td>
</tr>
<tr>
<td>TR</td>
<td>Rise time</td>
<td>TSTEP</td>
<td>second</td>
</tr>
<tr>
<td>TF</td>
<td>Fall time</td>
<td>TSTEP</td>
<td>second</td>
</tr>
<tr>
<td>PW</td>
<td>Pulse width</td>
<td>TSTEP</td>
<td>second</td>
</tr>
<tr>
<td>PER</td>
<td>Period</td>
<td>TSTEP</td>
<td>second</td>
</tr>
</tbody>
</table>

4.5.1.2 Sinusoidal

General Form:
SIN(VO VA FREQ TD THETA PHASE)

Example:
Vin 3 0 SIN(0 110 60 0 120)
4.5.1.3 Exponential

**General Form:**

\[ \text{EXP}(V1 \ V2 \ \text{TD1} \ \text{TAU1} \ \text{TD2} \ \text{TAU2}) \]

**Example:**

\[ \text{Vin} \ 3 \ 0 \ \text{EXP}(-4 \ -1 \ 2n \ 30n \ 60n \ 40n) \]

<table>
<thead>
<tr>
<th>Name</th>
<th>Parameter</th>
<th>Default Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>VO</td>
<td>Offset</td>
<td>--</td>
<td>V, A</td>
</tr>
<tr>
<td>VA</td>
<td>Amplitude</td>
<td>--</td>
<td>V, A</td>
</tr>
<tr>
<td>FREQ</td>
<td>Frequency</td>
<td>1/TSTOP</td>
<td>Hz</td>
</tr>
<tr>
<td>TD</td>
<td>Delay</td>
<td>0.0</td>
<td>second</td>
</tr>
<tr>
<td>THETA</td>
<td>Damping factor</td>
<td>0.0</td>
<td>1/second</td>
</tr>
<tr>
<td>PHASE</td>
<td>Initial phase</td>
<td>0.0</td>
<td>degree</td>
</tr>
</tbody>
</table>

4.5.1.4 Piece-Wise Linear

**General Form:**

\[ \text{PWL}(T1 \ V1 <T2 \ V2 \ T3 \ V3 \ldots>) <r=value> <td=value> \]

**Example:**

\[ \text{V osc} \ 3 \ 0 \ \text{PWL}(0 -1 \ 10u -1 11u 0 20u 0 21u 1 50u 1) \ r=0 \ td=15u \]

Each pair of value (Ti Vi) specifies that the value of source is Vi (in V or A) at time Ti. The value of the source at intermediate values of the time is determined by using linear interpolation on the input values.

If r is not given, the whole sequence of values (Ti Vi) is issued once, and then the output stays at its final value.

If r is given, the value of r has to be either 0 or one of the time points Ti.

- If r=0, the whole sequence from time=0 to time=Tn is repeated forever.
- If r=Ti, the sequence between time=Ti and time=Tn will be repeated forever.

If td is given, the whole PWL sequence is delayed by time=td second.

4.5.1.5 Single-Frequency FM

**General Form:**

\[ \text{SFFM}(\text{VO} \ \text{VA} \ \text{FC} \ \text{MDI} \ \text{FS}) \]

**Example:**

\[ \text{V1} \ 4 \ 0 \ \text{SFFM}(0 \ 1m \ 20K \ 5 \ 1K) \]

<table>
<thead>
<tr>
<th>Name</th>
<th>Parameter</th>
<th>Default Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>VO</td>
<td>Offset</td>
<td>--</td>
<td>V, A</td>
</tr>
<tr>
<td>VA</td>
<td>Amplitude</td>
<td>--</td>
<td>V, A</td>
</tr>
</tbody>
</table>
Chapter 4: SPICE Elements and Device Models

4.5.1.6 Amplitude Modulated Source (AM)

**General Form:**
AM(V A VO MF FC TD)

**Example:**
V1 5 0 AM(0.5 1 20K 5MEG 1m)

<table>
<thead>
<tr>
<th>Name</th>
<th>Parameter</th>
<th>Default Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>VA</td>
<td>Amplitude</td>
<td>--</td>
<td>V, A</td>
</tr>
<tr>
<td>VO</td>
<td>Offset</td>
<td>--</td>
<td>V, A</td>
</tr>
<tr>
<td>MF</td>
<td>Modulating frequency</td>
<td>--</td>
<td>Hz</td>
</tr>
<tr>
<td>FC</td>
<td>Carrier frequency</td>
<td>1/TSTOP</td>
<td>Hz</td>
</tr>
<tr>
<td>TD</td>
<td>Signal delay</td>
<td>--</td>
<td>second</td>
</tr>
</tbody>
</table>

FC Carrier frequency 1/TSTOP Hz
MDI Modulation index
FS Signal frequency 1/TSTOP Hz

4.5.1.7 Transient Noise Source

**General Form:**
TRNOISE(NA NT NALPHA NAMP RTSAM RTSCAPT RTSEMT)

**Examples:**
Vwhitenoise 6 0 DC 0 TRNOISE(20n 0.5n 0 0)
V1ofnoise 7 0 DC 0 TRNOISE(0 10p 1.1 12p)

<table>
<thead>
<tr>
<th>Name</th>
<th>Parameter</th>
<th>Default Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>NA</td>
<td>Gaussian noise rms voltage amplitude</td>
<td>--</td>
<td>V, A</td>
</tr>
<tr>
<td>NT</td>
<td>Time step</td>
<td>--</td>
<td>second</td>
</tr>
<tr>
<td>NALPHA</td>
<td>1/f exponent</td>
<td>0&lt;alpha&lt;2</td>
<td>--</td>
</tr>
<tr>
<td>NAMP</td>
<td>1/f amplitude</td>
<td>--</td>
<td>V, A</td>
</tr>
<tr>
<td>RTSAM</td>
<td>Random telegraph signal (RTS) amplitude</td>
<td>--</td>
<td>V, A</td>
</tr>
<tr>
<td>RTSCAPT</td>
<td>RTS capture time</td>
<td>--</td>
<td>second</td>
</tr>
<tr>
<td>RTSEMT</td>
<td>RTS emission time</td>
<td>--</td>
<td>second</td>
</tr>
</tbody>
</table>

4.5.1.8 Random Voltage Source

**General Form:**
TRRANDOM(TYPE TS <TD> < PARAM1 PARAM2>)

**Example:**
Vrandom 8 0 DC 0 TRRANDOM (2 10m 0 1)

<table>
<thead>
<tr>
<th>Name</th>
<th>Parameter</th>
<th>Default Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>TYPE</td>
<td>Type of random variates</td>
<td>Uniform</td>
<td></td>
</tr>
</tbody>
</table>
The parameters for each type of random source are:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>PARAM1</th>
<th>Default</th>
<th>PARAM2</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Uniform</td>
<td>Range</td>
<td>1</td>
<td>Offset</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>Gaussian</td>
<td>Std. Dev.</td>
<td>1</td>
<td>Mean</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>Exponential</td>
<td>Mean</td>
<td>1</td>
<td>Offset</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>Poisson</td>
<td>Lambda</td>
<td>1</td>
<td>Offset</td>
<td>0</td>
</tr>
</tbody>
</table>

### 4.5.2 Linear Dependent Sources

Four types of linear dependent sources are described in this section:
- Voltage controlled voltage source
- Voltage controlled current source
- Current controlled voltage source
- Current controlled current source

#### 4.5.2.1 Voltage-Controlled Voltage Source

This element defines a linear voltage controlled voltage source.

**General Form:**

\[ \text{Ename N1 N2 NC1 NC2 Gain} \]

**Examples:**

\[ \text{E1 2 3 4 5 2.5} \]

<table>
<thead>
<tr>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ename</td>
<td>Name of the controlled voltage source</td>
</tr>
<tr>
<td>N1, N2</td>
<td>Positive and negative voltage source nodes</td>
</tr>
<tr>
<td>NC1, NC3</td>
<td>Positive and negative controlling voltage nodes</td>
</tr>
<tr>
<td>Gain</td>
<td>Voltage gain</td>
</tr>
</tbody>
</table>

#### 4.5.2.2 Voltage-Controlled Current Source

This element defines a linear voltage controlled current source.

**General Form:**

\[ \text{Gname N1 N2 NC1 NC2 Gain} \]

**Examples:**

\[ \text{G1 2 3 4 5 10} \]

<table>
<thead>
<tr>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gname</td>
<td>Name of the controlled current source</td>
</tr>
<tr>
<td>N1, N2</td>
<td>Current source nodes. Positive current flows from N1 to N2</td>
</tr>
<tr>
<td>NC1, NC3</td>
<td>Positive and negative controlling voltage nodes</td>
</tr>
<tr>
<td>Gain</td>
<td>Transconductance, in mohs.</td>
</tr>
</tbody>
</table>
### 4.5.2.3 Current-Controlled Voltage Source

This element defines a linear current controlled voltage source.

**General Form:**

\[ \text{Hname N1 N2 Vname Gain} \]

**Examples:**

\[ \text{H1 4 0 Vload 200} \]

<table>
<thead>
<tr>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hname</td>
<td>Name of the controlled voltage source</td>
</tr>
<tr>
<td>N1, N2</td>
<td>Positive and negative voltage source nodes</td>
</tr>
<tr>
<td>Vname</td>
<td>Name of the voltage source through which the controlling current flows. The direction of positive controlling current flow is from the positive node, through the source, to the negative node of Vname</td>
</tr>
<tr>
<td>Gain</td>
<td>Trans-resistance, in ohms</td>
</tr>
</tbody>
</table>

### 4.5.2.4 Current-Controlled Current Source

This element defines a linear current controlled current source.

**General Form:**

\[ \text{Fname N1 N2 Vname Gain} \]

**Examples:**

\[ \text{F1 3 4 Vsense 0.2} \]

<table>
<thead>
<tr>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fname</td>
<td>Name of the controlled current source</td>
</tr>
<tr>
<td>N1, N2</td>
<td>Current source nodes. Positive current flows from N1 to N2</td>
</tr>
<tr>
<td>Vname</td>
<td>Name of the voltage source through which the controlling current flows. The direction of positive controlling current flow is from the positive node, through the source, to the negative node of Vname</td>
</tr>
<tr>
<td>Gain</td>
<td>Current gain</td>
</tr>
</tbody>
</table>

### 4.5.3 Nonlinear Dependent Sources (Behavioral Sources)

The nonlinear dependent sources described in this section allow voltage and current sources resulting from evaluation of math expressions.

**General Form:**

\[ \text{Bname N1 N2 V=Vexpression} \]
\[ \text{Bname N1 N2 I=Iexpression} \]

**Examples:**

\[ \text{BV1 1 0 V= 100*\text{sin(\text{V(1)})}} \]
\[ \text{BI2 2 3 I= (\text{V(1)}<-1.0) ? -1.0 : (\text{V(1)}>1.0) ? 1.0 : \text{V(1)}} \]

<table>
<thead>
<tr>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bname</td>
<td>Name of the voltage or current source</td>
</tr>
<tr>
<td>N1, N2</td>
<td>Positive and negative voltage source nodes. For current source, positive current flows from N1 to N2.</td>
</tr>
<tr>
<td>Vexpression, I expression</td>
<td>Math expressions which determines the voltage or current source output values.</td>
</tr>
</tbody>
</table>
The following math functions can be used in the math expressions for the nonlinear dependent sources:

- Standard operators: +, -, *, /, **
- Logical operators: ==, ! =, <, >, <=, >=, ||, &&
- Trigonometric functions: SIN, COS, TAN, ASIN, ACOS, ATAN
- Hyperbolic functions: COSH, SINH, ACOSH, ASINH, ATANH
- Exponential and logarithmic: EXP, LN, LOG
- Other: ABS, SQRT
- Two variable functions: MIN, MAX, POW
- Ternary function: A ? B : C means if A, then B, else C.
- Special variables: time, temper, hertz