LTspice IV User Guide

Contents

Introduction

SPICE (Simulation Program with Integrated Circuit Emphasis) is a general-purpose circuit simulator developed at the Electronics Research Laboratory in the University of California, Berkeley, by the doctoral student Laurence W. Nagel. SPICE's first version (SPICE1) was presented at the 16th Midwest Symposium on Circuit Theory in Canada in April of 1973. You can read the original SPICE paper by going to the following link:

http://docs.google.com/viewer?url=http://www.eecs.berkeley.edu/Pubs/TechRpts/1973/ERL-382.pdf.

The last version that UC Berkeley released was SPICE3f4, which is still available for download at this page but they are no longer providing support for any SPICE release.

Since circuit simulation has become an important stage in the analog electronics design, SPICE (originally an open source tool) has been adopted all over the world by both industry and academia as the standard simulation program. Nowadays, there is a wide variety of SPICE-based software available. Some versions have been made commercial (hence supported) and some others have been kept as open-source or freeware.

One of the most promising SPICE-based programs in academia available today is LTspice IV by Linear Technology Corporation. LTspice IV is a high-performance, general-purpose SPICE simulator based on SPICE3f4/5 release, which includes a built-in schematic capture interface and a waveform viewer tool. Also, it is possible to write a SPICE deck directly in the program or another text editor and run your simulations using this program. One of the advantages of this last characteristic is that LTspice IV can also be used as a SPICE engine to perform analyses using SPICE decks generated automatically by other schematic capture software, a PCB or a physical design tool.

Typically, a SPICE tool loads an input file (the SPICE deck file), which contains the circuit description, the type of analysis to be performed and the output variables to be printed or plotted; all this in a plain text format. This file is passed through the simulation engine and the desired output is generated in the form of an output text file or a graphic plot, depending on the analysis type you just performed. Recently developed tools (like LTspice IV) are capable of automatically generating the SPICE deck file using the schematic diagram (for the circuit description part) and the analysis setup interface (for the type of analysis and output variables to be presented). The software itself decides the best way to display the results.

LTspice IV, unlike some other freeware simulation tools, is free to use, although is not an open-source software. Users can not modify the source code or the software structure, but they can use it for as long as they want and can capture and simulate circuits as big as they want (limited only by the computer resources). There is no circuit size limitation either. Linear Technology Corporation maintains LTspice IV working, so it is possible to install updates when released directly from the built-in update tool or set the program to check for new updates often.

1. Hardware Requirements and Installation

LTspice IV is released as a pre-compiled Windows executable file (.exe) and it can be installed in almost any version of Windows running today (from Windows 98 to Windows 7). A free hard disk space of about 10 GB and RAM of 1 GB is recommended to avoid unfinished simulations for large circuits. LTspice IV can also be installed and executed over Linux using Wine (an open-source Windows emulator for Linux) as the installation interface.

You can download the self-extracting installation file of LTspice IV from the Linear Technology Corporation site. For Windows, you just have to run the installation file and follow the instructions to install the tool. To open the installed tool, just double-click on the LTspice IV icon on the desktop or in the Programs menu.

If you are a Linux user, you need to install Wine first (if you have not done it yet). Depending on your distribution, you can install it directly from your repositories or download a binary file from the Wine download section of the official website and install it. LTspice IV runs exactly the same on Linux as it does on Windows. There is a Mac distribution for Wine also, although it has not been tested by the authors of this manual.

2. LTspice IV Basics

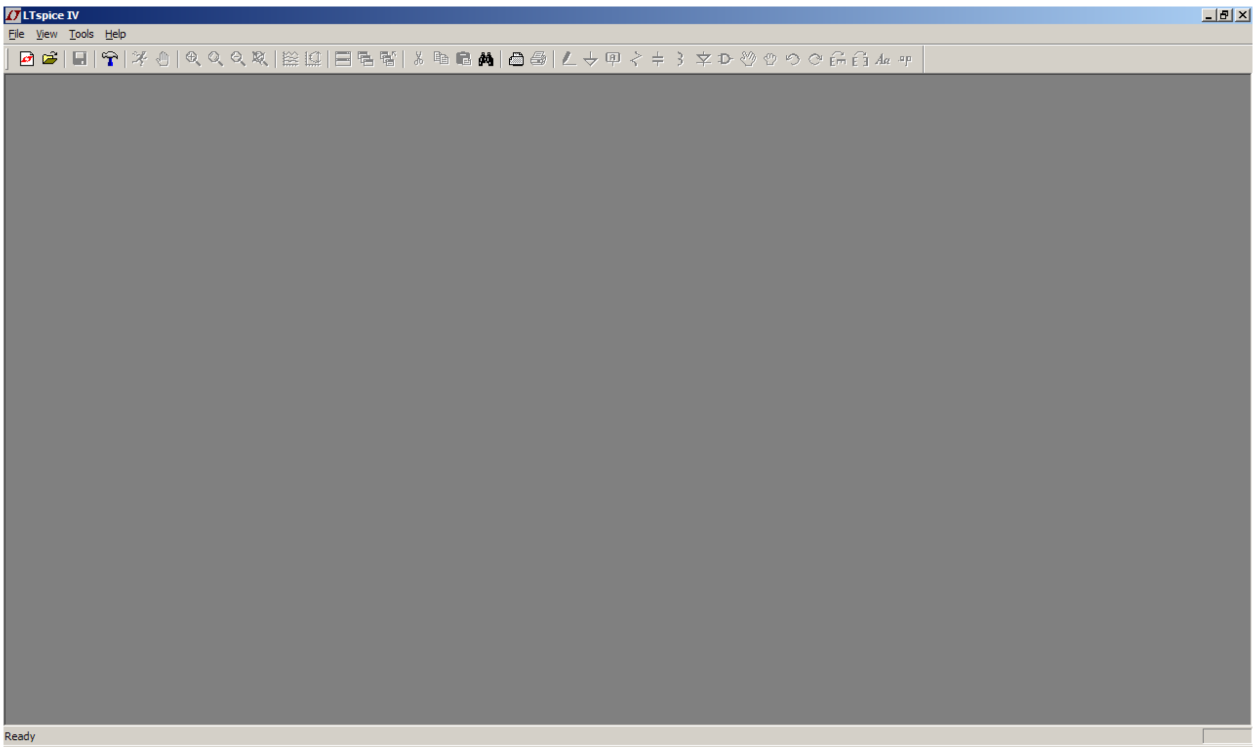
The great advantage of LTspice over other Spice-based tools available is that it is free and its capabilities are unrestricted. This means that the size of your circuits and the hierarchy levels are limited only by the resources of your computer. Also, you can simulate a circuit described by its Spice deck without worrying about the number of nodes present in your circuit.

In LTspice it is easy to import sub-circuits like an OpAmp macro model and other circuits not included in the default LTspice library. Also, you can use a collection of symbols for your customized models or you can create new symbols.

2.1. Schematic capture

When you run LTspice IV, a screen similar to Figure 1 will appear. From this screen you can create a new schematic file or open an existing one. You are also able to open a previously created netlist or Log file. To create a new schematic, click the "New schematic" button or go to File => New schematic.

Figure 1. LTspice IV main screen



LTspice has some shortcuts that are very useful for the schematic capture process. Additionally, there are some other shortcuts for the most commonly used electric elements such as resistors, capacitors, inductors, etc. Table 1 below shows the main shortcuts available in LTspice IV.

Table 1. Most used shortcuts available in LTspice IV

Command	Shortcut	Button	Description
Help	F1		Open help.
Resistor	R		Place a new resistor on the schematic.
Inductor	L		Place a new inductor on the schematic.
Capacitor	C		Place a new capacitor on the schematic.

Diode	D		Place a new diode on the schematic.
Ground	G		Place a GROUND symbol. This is node “0”, the global circuit common.
Component	F2		Place a new component on the schematic. The command brings up a dialog that lets you browse and preview the symbol database.
Wire	F3		Click the left mouse button to start a wire. Each mouse click will define a new wire segment. Click on an existing wire segment to join the new wire. Right-click once to cancel the current wire. Right-click again to quit this command.
Label net	F4		Specify the name of a node so the netlister does not generate an arbitrary one for this node.
Delete	F5		Delete objects by clicking on them or dragging a box around them.
Duplicate	F6		Duplicate objects by clicking on them or dragging a box around them. You can copy from one schematic to another if they are both opened in the same invocation of LTspice IV. Start the Duplicate command in the window of the first schematic. Then make the second schematic the active window and type Ctrl-V.
Move	F7		Click on or drag a box around the objects you wish to move. Then you can move those objects to a new location.
Drag	F8		Click on or drag a box around the objects you wish to drag. Then you can move those objects to a new location; the attached wires move to the new location.
Rotate	Ctrl + R		Rotate the selected objects. Note that this is grayed out when there are no objects selected.
Mirror	Ctrl + E		Mirror the selected objects. Note that this is grayed out when there are no objects selected.
Undo	F9		Undo the last command.
Redo	↑ + F9		Redo the last Undo command.
Text	T		Place text on the schematic. This merely annotates the schematic with information. This text has no electrical impact on the circuit.

SPICE Directive	S	.op	Place text on the schematic that will be included in the netlist. This lets you mix schematic capture with a SPICE netlist. It lets you set simulation options, include files that contain models, define new models, or use any other valid SPICE commands. You can even use it to run a sub-circuit that you do not have a symbol for by stating an instance of the model (a SPICE command that begins with an 'X') on the schematic and including the definition.
SPICE Analysis			Enter/Edit the simulation command.

You can access some of these shortcuts and other important commands in the toolbar placed at the top of the schematic capture screen. Through this you are able to access the most commonly used components, manipulation tools, Spice directives and other standard options. Figure 2 shows the LTspice toolbar.

Figure 2. LTspice toolbar

2.2. Schematic capture procedure

On LTspice you can capture your schematics quickly by using the shortcuts available for this purpose. In this guide, we recommend you use the 4-step procedure, where each step makes use of its corresponding access key for a quicker capture and edition.

Step 1. The schematic capture procedure starts by adding the necessary components to the capturing screen. By using the F2 key you can quickly access the "Component" window to select and place all the components you need for your circuits. Of course, you can directly use the shortcuts available for the most common elements instead of using this key and save some time.

Step 2. After placing your circuit elements, hit F3 to wire your circuit. As you could have noticed when you place your circuit's elements, there is a grid on the screen useful to align the elements and wires.

Step 3. Once your circuit is drawn, it is important to label all the nodes in the circuit so when you run your analysis, you can identify those nodes quickly. Also, labeling your circuit becomes practical when organizing your schematics because, if you assign the same name to two or more nodes, you are actually shortening them together although no wire is placed. This is useful when working with large circuits and for connecting voltage sources to different nodes.

Step 4. For most cases, once you have wired and labeled your circuit, you define the type of analysis you want to perform and run the simulation. This can be done by going to Edit => SPICE analysis.

2.3. Analysis setup

LTspice includes all the analyses available in most Spice-based simulation tools. For a better understanding of the capabilities of the Spice analyses, Table 2 shows the most common applications for each analysis.

Table 2. Main applications for each type of Spice analysis

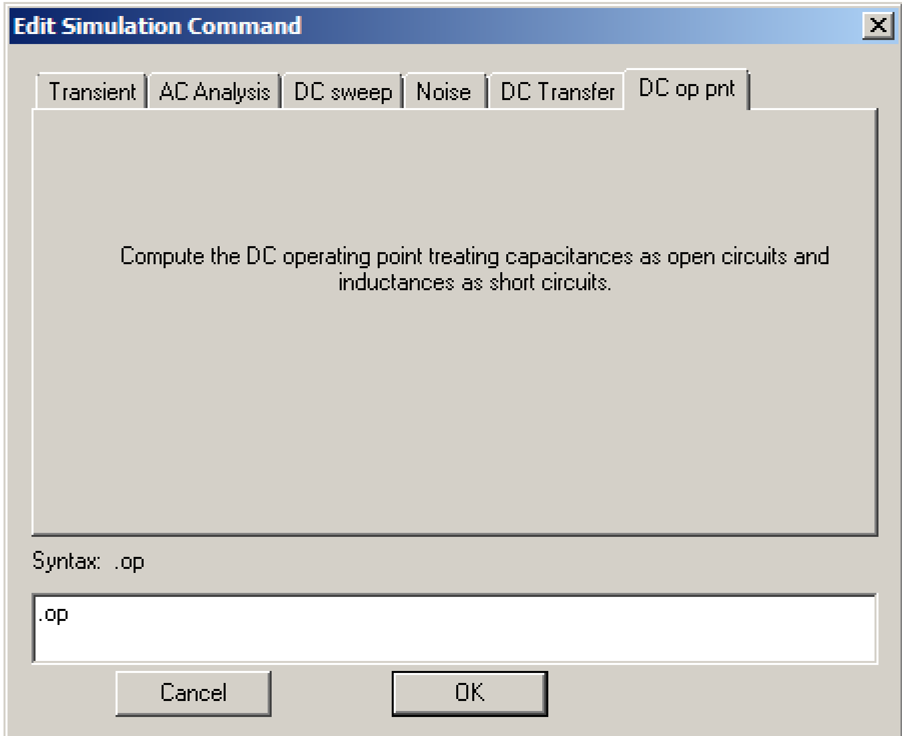
Analysis	**Application**
DC operation point	Determine the DC conditions of a biased transistor (i. e. the operation region). DC node voltages and loop currents of an electric network.
Transient	The time response of any circuit.
DC sweep	Transfer function of an amplifier, DC characteristic curves of a transistor.
AC	Frequency response (gain and phase) of a passive or active filter. Bandwidth of an amplifier.
Noise	Test the response of an audio amplifier under noise conditions.
DC transfer	Input and output resistance of an electric network. Output impedance of an amplifier. Output voltage of a network given for a determined input value.

A short description of each Spice analysis and its corresponding Spice syntax is presented next.

1.3.1. DC operation point

The `.OP` command replaces all capacitors with an open circuit and all the inductors with a short circuit and calculates the DC solution for the circuit. The results are displayed in a dialog box that pops up after the simulation is complete. Figure 3 shows the analysis setup where there is no need to enter any argument. AC sources are disconnected.

Figure 3. DC operation point analysis' setup



2.3.2. Transient analysis

A non-linear, time domain analysis is performed, so results are displayed in such a way that the independent axis is timed in seconds, and the dependent axis can be any of the electrical variables interpreted by Spice or a mathematical expression of them.

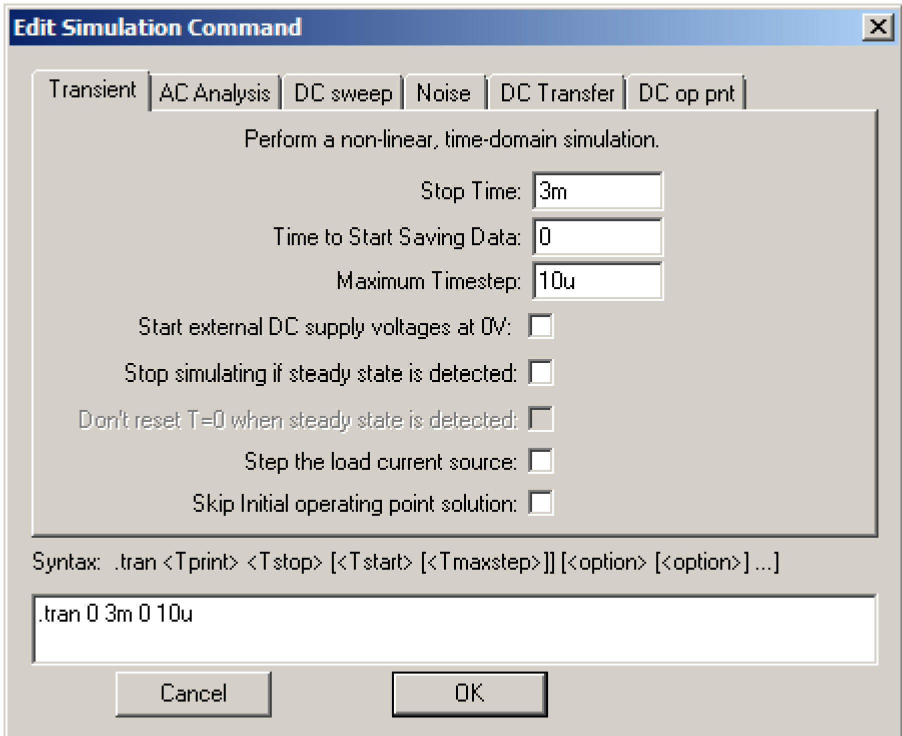
The syntax of the transient analysis is as follows:

```
.TRAN <Tstep> <Tstop> [Tstart [dTmax]] [modifiers]
```

where `<Tstep>` is the plotting increment of the waveforms and `<Tstop>` is the duration of the simulation. You can specify `Tstart` if you wish to start the simulation at a time different from zero.

Figure 4 shows the transient analysis setup window, where you have the option of writing the Spice directive to set such analysis at the bottom of the window.

Figure 4. Transient analysis setup



2.3.3. DC sweep

With this analysis, a DC source is swept in a determined voltage range, so the response of a variable as function of such voltage sweep can be plotted. This means that we can find the transfer function from the input source to some output variable.

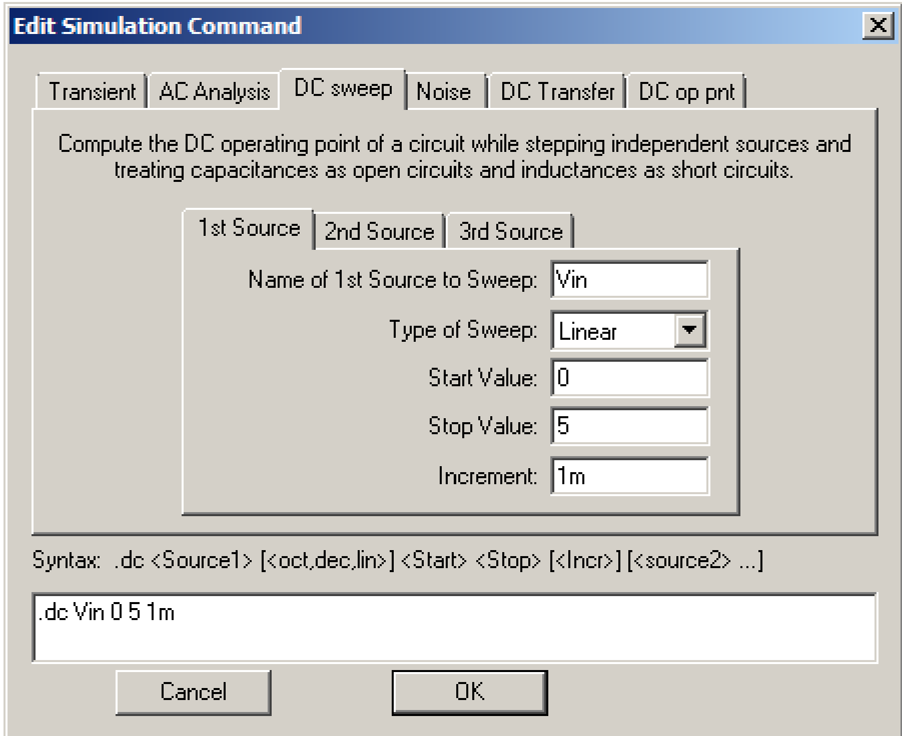
The syntax of the DC sweep analysis is as follows:

```
.dc <srcnam> <Vstart> <Vstop> <Vincr>
+ [<srcnam2> <Vstart2> <Vstop2> <Vincr2>]
```

where <srcnam> is the name of the DC source to be swept starting from <Vstart> and stoping at <Vstop> by increments of <Vincr>.

Figure 5 shows the setup window of this type of analysis. As you can see, you can sweep up to three sources, which is useful when plotting parametric curves.

Figure 5. DC sweep analysis' setup



2.3.4. DC transfer function

With this analysis you are able to find the small-signal DC transfer function of a voltage node or loop current, due to small variations of an independent source. Results shown includes the input voltage and resistance and output voltage or current and output resistance.

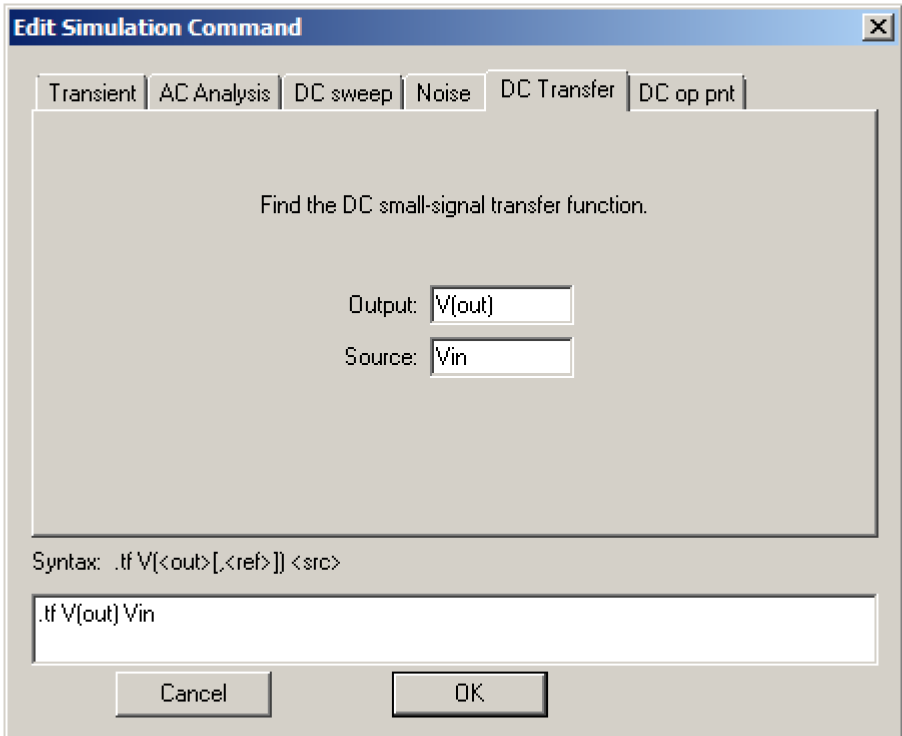
The syntax of the transfer function analysis is as follows:

```
.TF V(<node>[, <ref>]) <source>
.TF I(<voltage source>) <source>
```

where V(<node>[,<ref>]) and I(<voltage source>) are the voltage node and current through the source, respectively, and those values will be displayed as a function of the voltage source <source>.

Figure 6 shows the setup of a DC transfer function analysis.

Figure 6. DC transfer function analysis' setup



2.3.5. AC analysis

This analysis computes the small-signal AC response of a circuit as a function of frequency. This analysis finds the DC operation point of the circuit first, so you can use DC sources to bias your circuit to find the small signal response under a DC bias condition.

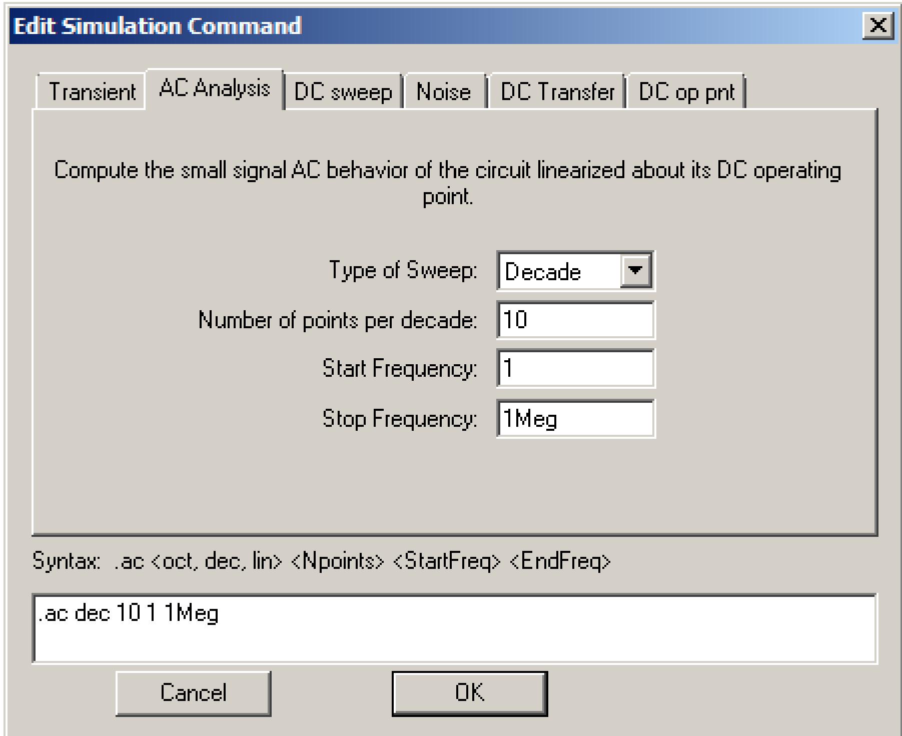
The syntax of the AC analysis is as follows:

```
.ac <oct, dec, lin> <Nsteps> <StartFreq> <EndFreq>
```

You can plot your results using octal, decade or linear frequency ranges using the keywords `oct`, `dec` or `lin`, respectively. With `<Nsteps>` you define the number of steps for each octave or decade of the range from `<StartFreq>` to `<EndFreq>`.

Figure 7 shows the setup of an AC analysis.

Figure 7. AC analysis' setup



2.3.6. Noise analysis

This analysis performs a frequency domain analysis to compute Johnson, shot and flicker noise types. The output is noise spectral density per unit square root bandwidth.

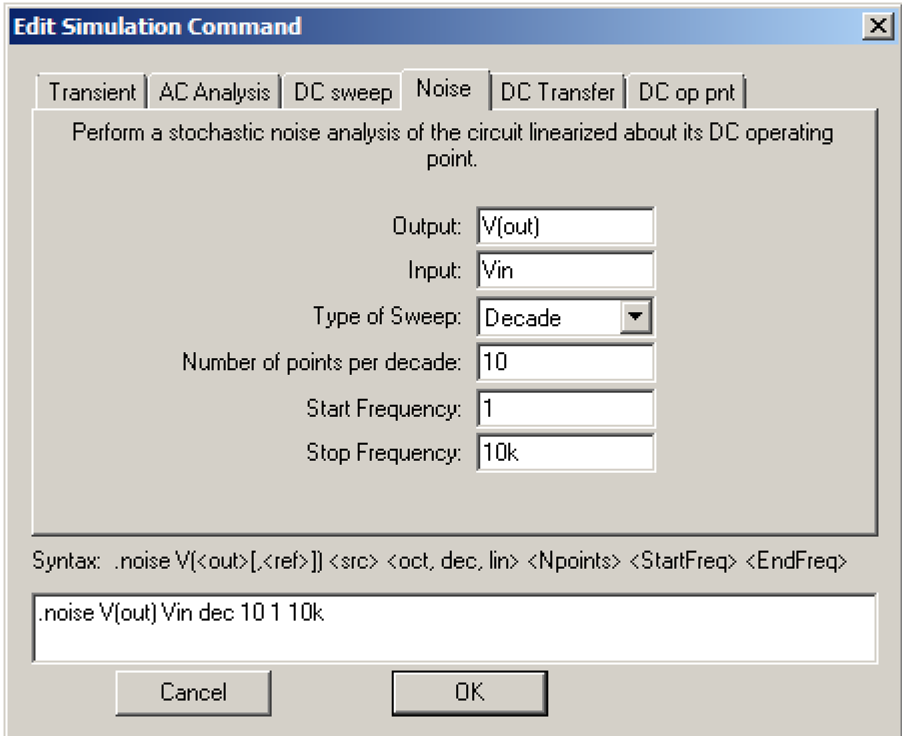
The syntax of the noise analysis is

```
.noise V(<out>[,<ref>]) <src> <oct, dec, lin>
      + <Nsteps> <StartFreq> <EndFreq>
```

`V(<out>[,<ref>])` is the node at which the output noise is calculated. `<src>` is the name of an independent source to which input noise is referred. `<src>` is the noiseless input signal. The parameters `<oct, dec, lin>`, `<Nsteps>`, `<StartFreq>`, and `<EndFreq>` define the frequency range of interest and resolution in the manner used in the .ac directive.

Figure 8 shows the setup of a noise analysis.

Figure 8. Noise analysis' setup

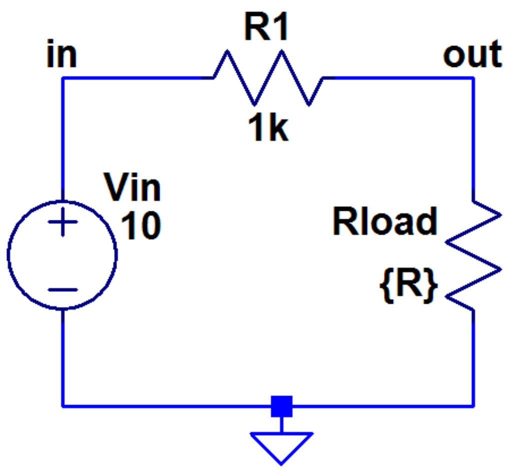


2.3.7. Parametric analysis

With a parametric analysis you are able to sweep the value of a component while performing any type of analysis available on LTspice IV such as transient, DC bias point, AC analysis, DC sweep. Basically, a parametric analysis is a multi-run process where you set a main analysis and specify a series of values to be swept for a component. When you run the analysis, LTspice IV sets the first value of the parametric variable and performs the simulation. When finished, the next value is set automatically on the circuit and the simulations run again. This process is repeated until the list of values for the component that you selected is completed and the results are plotted.

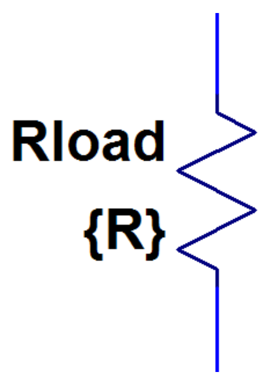
For instance, Figure 9 shows a simple voltage divider where the value of R_{load} will be swept and the voltage at node Out will be calculated for each resistance value.

Figure 9. Simple voltage divider



To sweep the value of a resistor for a DC operation point analysis, a parametric variable has to be declared instead of a constant value for the resistor (see Figure 10).

Figure 10. Parametric variable for a resistor



Then, through a Spice directive, the values of the resistor have to be declared following the syntax below:

```
.step param <var> <initial> <final> <step>
```

For example, for the resistor on Figure 10

```
.step param R 1k 3k 1k
```

With this, the resistor will sweep from 1 k to 3 k in increments of 1 k.

You can set up any type of analysis and declare a parametric variable. The results will be displayed as a family of curves, one for each value of the parametric variable.

2.3.8. *Temperature analysis*

A temperature analysis is a parametric analysis that uses the intern variable TEMP; this analysis performs an analysis run for each temperature listed in its arguments. The syntax is as follows:

```
.TEMP <T1> <T2> ...
```

where `<T1> <T2> ...` are the temperature values to be swept.

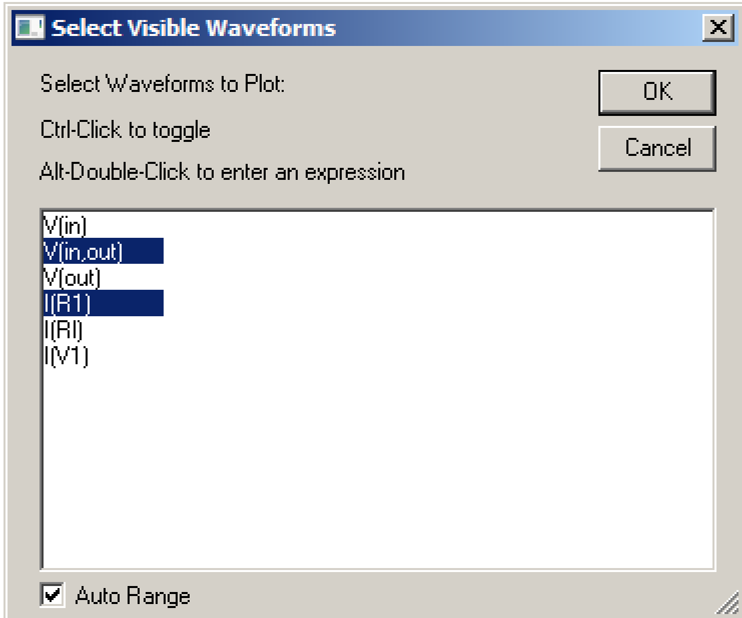
2.4. Plot window

The graphical results of your simulations can be displayed in the built-in waveform viewer of LTspice. All types of analysis, except the DC bias point analysis, return graphical information for a better interpretation of the results, so it is important to be aware of the capabilities of this tool.

There are three different ways of selecting data to be plotted in the waveform viewer after running an analysis:

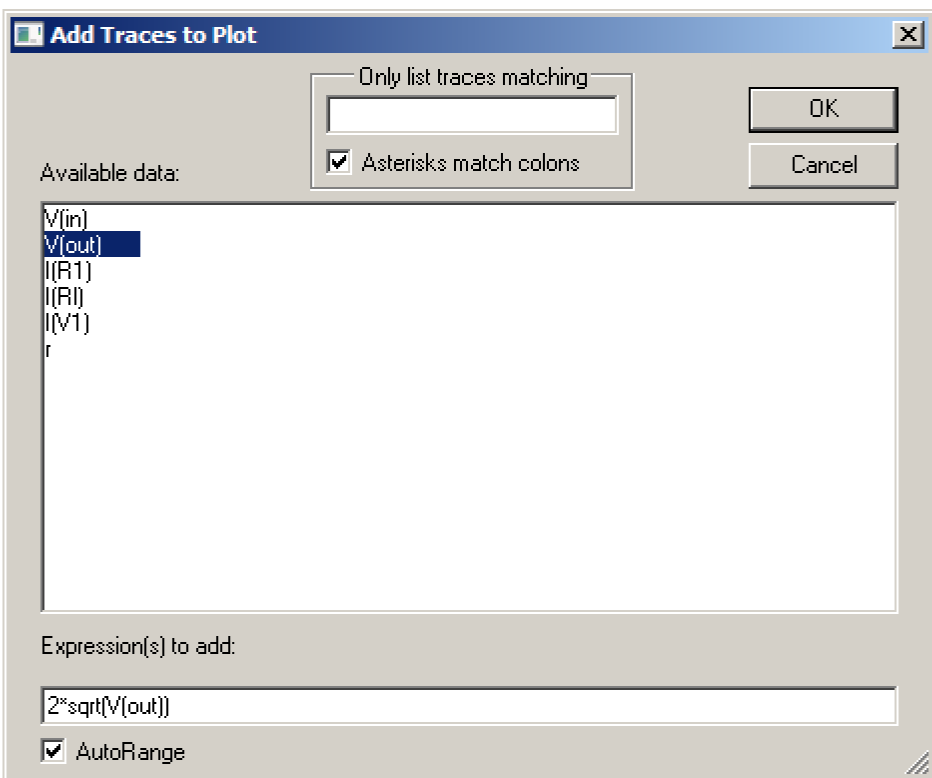
1. **Select the trace to plot directly on the schematic diagram.** You can plot the voltage across any node and ground by simply clicking over the desired wire. If you want to plot a voltage difference, just click the first node, hold the mouse button down and drag the pointer to another node. To plot the current through an element, just click the symbol of such element.
2. **Menu "Plot settings => Visible traces".** We you do this, a list of all node voltages and loop currents will appear (see Figure 11). By holding the Ctrl key you can select different labels and plot them all together in the same pane. Every time you use this method you will have to select all the traces you want to be plotted.

Figure 11. List of all node voltages and loop currents in the Plot settings =>Visible traces data selection



3. **Menu "Plot settings => Add trace"**. With this, the traces already plotted will not be replaced by the new trace, which will just be added to the older traces (Figure 12).

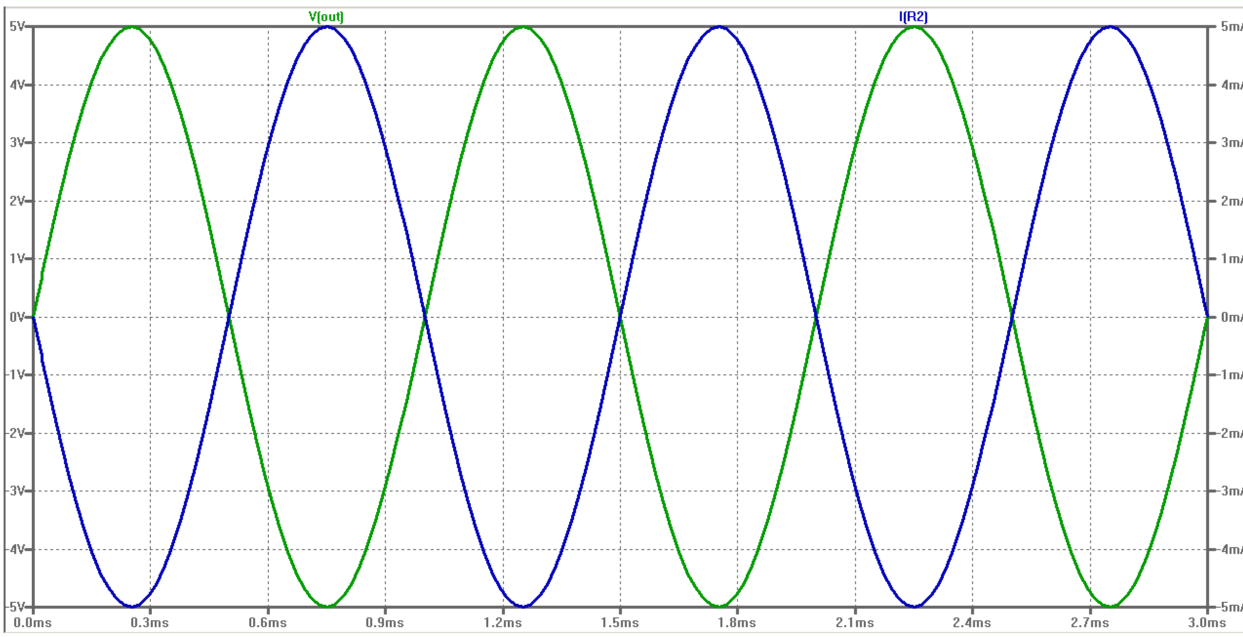
Figure 12. Add traces to plot window



2.4.1. Multiple traces

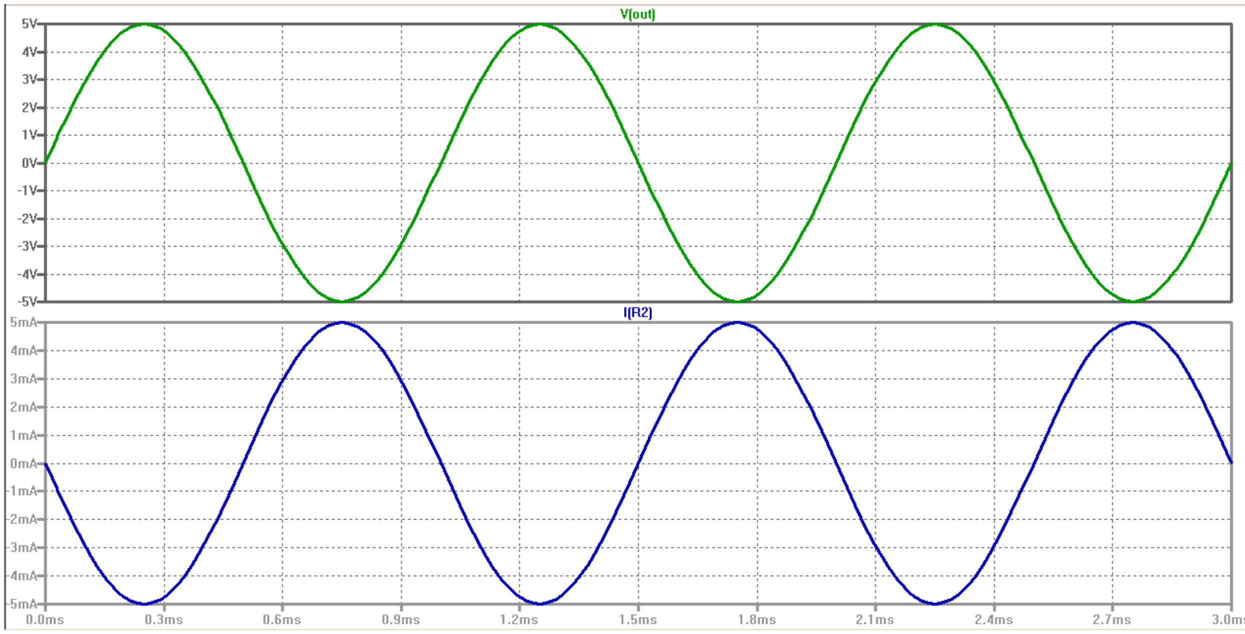
When working with multiple traces, you can plot all your curves on the same pane or you can divide them into separate panes for a better organization of the information. This way, in one pane you can plot a voltage curve and in a separate pane plot a current curve and avoid multiple scales on the same pane. In Figure 13 you can see a current and a voltage curve displayed on the same pane.

Figure 13. Current and voltage curves in the same pane



To add a new plot pane go to "Plot settings => Add plot pane". With this, you can separately display the two curves, as shown in Figure 14.

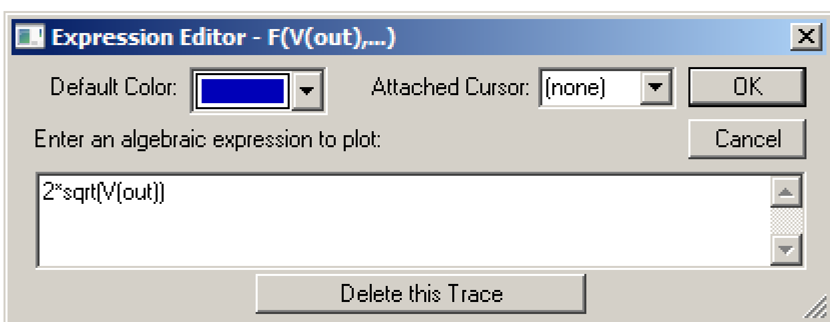
Figure 14. Two curves in two panes



2.4.2. Mathematical expressions

You are able to enter a mathematical expression through the “Visible traces” and “Add trace” options using all the signals present in the circuit. You can also do this by right-clicking on the label of a trace already plotted, so a dialog window similar to the one in Figure 15 appears.

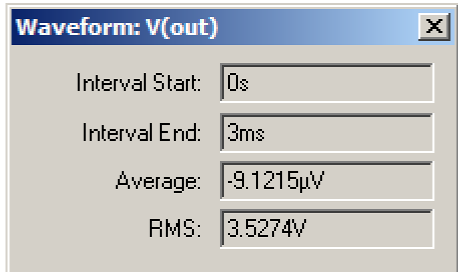
Figure 15. Window to enter a mathematical expression



Besides the mathematical expression you are also able to change the color of the selected trace.

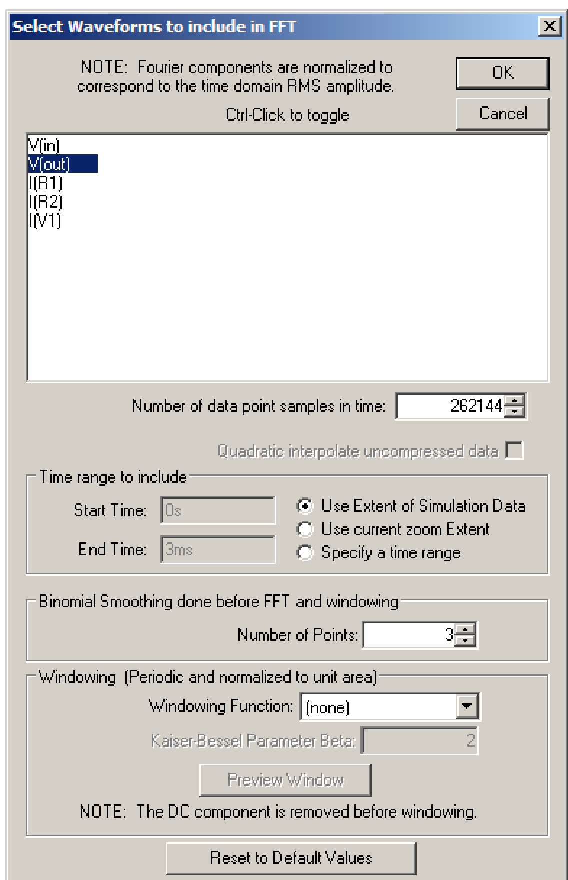
There is an easy way to find average and RMS values on a plot. To do this, place the cursor of the mouse over the label of interest, hold down the Ctrl key on your keyboard and left-click your mouse. A window similar to Figure 16 will display the information.

Figure 16. Average and RMS values of a 5V sine wave



You can also display the Fast Fourier transform of your traces. By clicking on the "View => FFT" menu button a dialog window will pop up indicating the place where you can select the traces you want to obtain its FFT. Figure 17 shows this dialog window.

Figure 17. Fast Fourier transform dialog window



The following functions are available to be used within the expressions:

Function	Description
`abs(x)`	Absolute value of x
`acos(x)`	Arc cosine of x
`arccos(x)`	Synonym for acos()
`acosh(x)`	Arc hyperbolic cosine
`asin(x)`	Arc sine
`arcsin(x)`	Synonym for sin()
`asinh(x)`	Arc hyperbolic sine
`atan(x)`	Arc tangent of x
`arctan(x)`	Synonym for atan()
`atan2(y, x)`	Four quadrant arc tangent of y/x
`atanh(x)`	Arc hyperbolic tangent
`buf(x)`	1 if x > .5, else 0
`ceil(x)`	Integer equal or greater than x
`cos(x)`	Cosine of x
`cosh(x)`	Hyperbolic cosine of x
`d()`	Finite difference-based derivative
`exp(x)`	e to the x
`floor(x)`	Integer equal to or less than x
`hypot(x,y)`	sqrt(x**2 + y**2)
`if(x,y,z)`	If x > .5, then y else z
`int(x)`	Convert x to integer
`inv(x)`	0. if x > .5, else 1
`limit(x,y,z)`	Intermediate value of x, y, and z
`ln(x)`	Natural logarithm of x
`log(x)`	Alternate syntax for ln()
`log10(x)`	Base 10 logarithm
`max(x,y)`	The greater of x or y
`min(x,y)`	The lesser of x or y
`pow(x,y)`	x**y
`pwr(x,y)`	abs(x)**y
`pwrs(x,y)`	sgn(x)*abs(x)**y
`rand(x)`	Random number between 0 and 1 depending on the integer value of x
`random(x)`	Similar to rand(), but with smoother transitions among values.
`round(x)`	Nearest integer to x
`sgn(x)`	Sign of x
`sin(x)`	Sine of x
`sinh(x)`	Hyperbolic sine of x

`sqrt(x)`	Square root of x
`table(x,a,b,c,d,...)`	Interpolate a value for x based on a look up table given as a set of pairs of points.
`tan(x)`	Tangent of x
`tanh(x)`	Hyperbolic tangent of x
`u(x)`	Unit step, i.e. 1 if x > 0., else 0
`uramp(x)`	x if x > 0., else 0
`white(x)`	Random number between -.5 and .5 that smoothly transitions among values even more smoothly than random().

You can also use some of the most common constants that are defined on LTspice.

Name	Value
E	2.7182818284590452354
Pi	3.14159265358979323846
K	1.3806503e-23
Q	1.602176462e-19

You can define your own functions to be used with your traces using the already defined functions and constants. Go to "Plot settings => Edit Plot Defs File" and enter your functions using the following syntax:

```
.func function_name(arguments) {mathematical expressions}
```

For example, to define the function that converts Hertz to Radians you write

```
.func radians(freq) {2*Pi*freq}
```

which takes a value of frequency in Hertz and obtains its corresponding value in radians/sec.

For a complete list of the Spice directives available on LTspice, refer to the Linear Technology Corporation website (http://www.linear.com/designtools/software/)