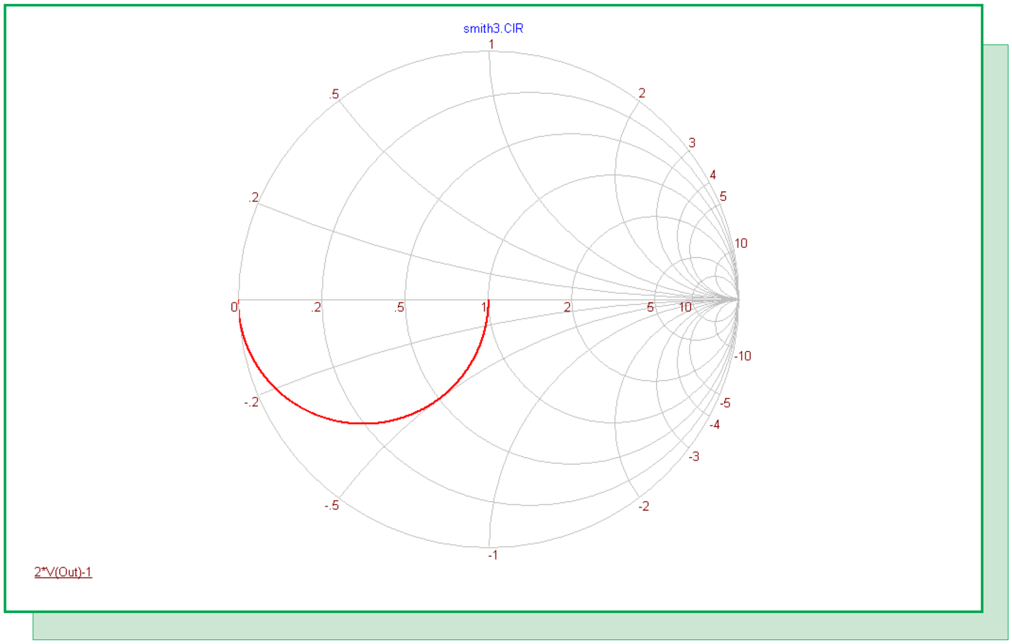
spectrum news

Applications for Micro-Cap™ Users

Spring 2005

News



Smith Chart Impedance Plots

Featuring:

- Using the Make Subcircuit Option
- Smith Charts and Impedance Plots
- Voltage Limiting Current Source Macro

News In Preview

This newsletter's Q and A section describes the reason why PGT won't necessarily be equivalent to the sum of PST and PDT when plotting them in AC analysis. The Easily Overlooked Features section describes the use of the .TR statement in setting a flexible maximum time step.

The first article describes how to use the Translate to SPICE option to create a subcircuit model.

The second article describes how to plot input impedance in a Smith chart.

The third article describes a voltage limiting current source macro that can be used in place of the ideal current source to provide better results for some circuit configurations.

Contents

Book Recommendations

General SPICE

- *Computer-Aided Circuit Analysis Using SPICE*, Walter Banzhaf, Prentice Hall 1989. ISBN# 0-13-162579-9

- *Macromodeling with SPICE*, Connelly and Choi, Prentice Hall 1992. ISBN# 0-13-544941-3

- *Inside SPICE-Overcoming the Obstacles of Circuit Simulation*, Ron Kielkowski, McGraw-Hill, First Edition, 1993. ISBN# 0-07-911525-X

- *The SPICE Book*, Andrei Vladimirescu, John Wiley & Sons, Inc., First Edition, 1994. ISBN# 0-471-60926-9

MOSFET Modeling

- *MOSFET Models for SPICE Simulation, William Liu, Including BSIM3v3 and BSIM4*, Wiley-Interscience, First Edition, ISBN# 0-471-39697-4

VLSI Design

- *Introduction to VLSI Circuits and Systems,* John P. Uyemura, John Wiley & Sons Inc, First Edition, 2002 ISBN# 0-471-12704-3

Micro-Cap - Czech

- *Resime Elektronicke Obvody,* Dalibor Biolek, BEN, First Edition, 2004. ISBN# 80-7300-125-X

Micro-Cap - German

- *Schaltungen erfolgreich simulieren mit Micro-Cap V,* Walter Gunther, Franzis', First Edition, 1997. ISBN# 3-7723-4662-6

Micro-Cap - Finnish

- *Elektroniikkasimulaattori,* Timo Haiko, Werner Soderstrom Osakeyhtio, 2002. ISBN# ISBN 951-0-25672-2

Design

- *Microelectronic Circuits High Performance Audio Power Amplifiers,* Ben Duncan, Newnes, First Edition, 1996. ISBN# 0-7506-2629-1

- *Microelectronic Circuits.,* Adel Sedra, Kenneth Smith, Fourth Edition, Oxford, 1998

High Power Electronics

- *Power Electronics,* Mohan, Undeland, Robbins, Second Edition, 1995. ISBN# 0-471-58408-8

- *Modern Power Electronics,* Trzynadlowski, 1998. ISBN# 0-471-15303-6

Switched-Mode Power Supply Simulation

- *SMPS Simulation with SPICE 3,* Steven M. Sandler, McGraw Hill, First Edition, 1997. ISBN# 0-07-913227-8

- *Switch-Mode Power Supply SPICE Simulation Cookbook*, Christophe Basso, McGraw-Hill 2001. This book describes many of the SMPS models supplied with Micro-Cap.

Micro-Cap Questions and Answers

Question: I am running an AC analysis on my schematic. In the AC analysis limits, the three Y expressions that I have specified are PDT, PST, and PGT in order to view the total power dissipated, stored, and generated in the schematic. I would expect that:

PGT = PDT + PST

When I run the simulation, the outputs of PDT and PST produce a sum greater than the value of PGT. However, if I plot the sum directly as:

PDT + PST

then the output of that expression will match exactly with the PGT waveform. What is happening?

Answer: In AC analysis, the internal calculations are performed using complex mathematical operators. An expression such as PDT is actually calculated as a complex number, but when it is plotted to the screen, it is displayed as a magnitude unless an operator such as RE() or IM() is used in conjunction with it. When PDT, PST, and PGT are all plotted individually, the magnitude of PGT will not equal the sum of the magnitudes of PDT and PST as the sum would be equivalent to the following:

SQRT(RE(PDT)^2 + IM(PDT)^2) + SQRT(RE(PST)^2 + IM(PST)^2)

whereas the PGT magnitude would be equivalent to:

SQRT((RE(PDT) + RE(PST))^2 + (IM(PDT) + IM(PST))^2)

The reason that the PDT + PST waveform plot produces a match to the PGT waveform is that the magnitude operation is the last calculation done before plotting to the screen. When the addition operation occurs, the power values are still in complex form and the addition is therefore a complex operation where:

(RE(PDT) + RE(PST)) + (IM(PDT) + IM(PST))

This case is also applicable to voltages and currents when plotted in an AC analysis.

Easily Overlooked Features

This section is designed to highlight one or two features per issue that may be overlooked because they are not made visually obvious with a toolbar button.

.TR Command Statement

In a transient analysis, the Maximum Time Step field controls the maximum amount of time that will be present between two data point calculations. Decreasing the Maximum Time Step will produce more data points in a simulation but also increase the duration of the simulation run. For some simulations, a critical part of the transient analysis may require a smaller time step, but setting this smaller time step over the entire run will create a very long run time. The purpose of the .TR statement is to provide flexibility in setting the time step for these types of simulations. The format of the .TR statement is as follows:

```
.TR <s1 t1> [s2 t2...sn tn]
```

where the s entries are the time step values and the t entries are the corresponding time values that the time step will be set for. An example .TR statement is:

```
.TR 30n 150n 1n 400n 30n 2u
```

In this example, the time step is limited to 30ns from 0 to 150ns. Between 150ns and 400ns, it is limited to 1ns. Between 400ns and 2us, it is limited to 30ns. A transient analysis is shown below that has this .TR statement implemented. Note the increased data point sampling between 150ns and 400ns versus the rest of the simulation.

The .TR statement should be placed in one of the schematic text pages or the schematic itself. It will have priority over any value set in the Maximum Time Step field.

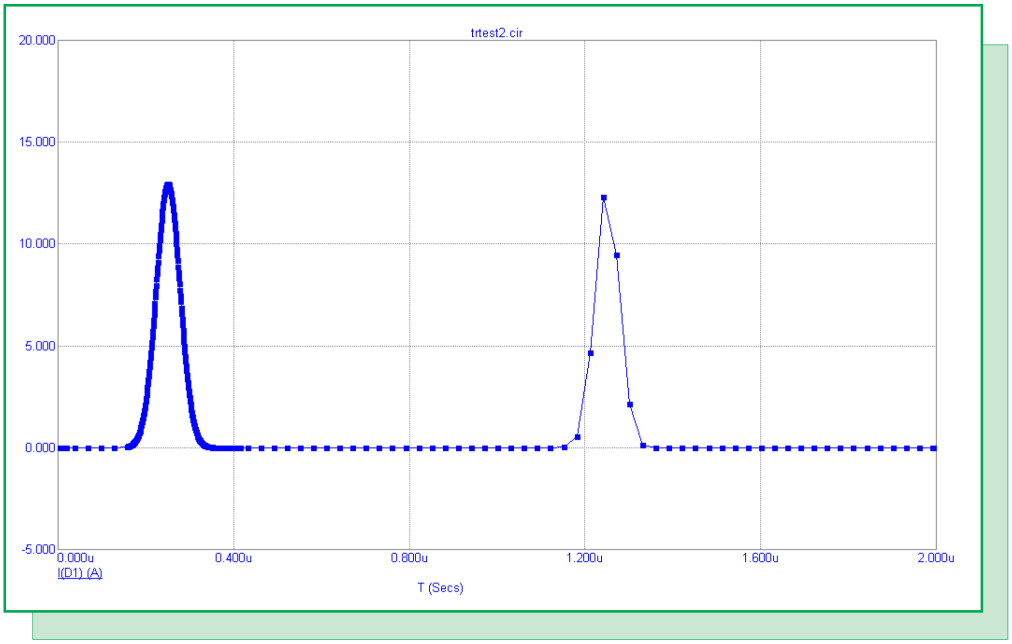


Fig. 1 - Transient analysis using a .TR statement

Using the Make Subcircuit Option

There are two model types within Micro-Cap that let the behavior of a complex circuit block be incorporated into and represented by a single component in a schematic. These model types are the macro and the subcircuit. The main difference between these two types is that a macro has its circuitry defined through a schematic whereas the subcircuit has its circuitry defined through a SPICE text netlist.

While the macro is typically the easier model type to create, it is limited to use in only the Micro-Cap simulator. Though the text description of a subcircuit model is more difficult to develop, a subcircuit can be transferred to any other application that is capable of handling the SPICE netlist format. This portability is the main advantage to using a subcircuit model.

To ease the difficulties in creating a SPICE netlist, Micro-Cap provides a translation feature to convert schematic files into SPICE netlists. One of the options within this translate to SPICE feature is the ability to create a subcircuit model out of the schematic rather than the full SPICE circuit file that is normally produced. This article will describe the process of taking a schematic and converting it into its subcircuit equivalent.

The first step is to create the schematic that will represent the appropriate model. The schematic should only contain the components that will be part of the model as all of the components within the schematic will be incorporated into the subcircuit. For this example, the schematic to be used appears in Figure 2. This schematic models a basic Wien bridge oscillator that produces about a 9kHz output sine wave. The desired input and output nodes for the model have been labelled with text. The power supplies for the opamp have been called Vcc and Vee, and the output is simply labelled Out.

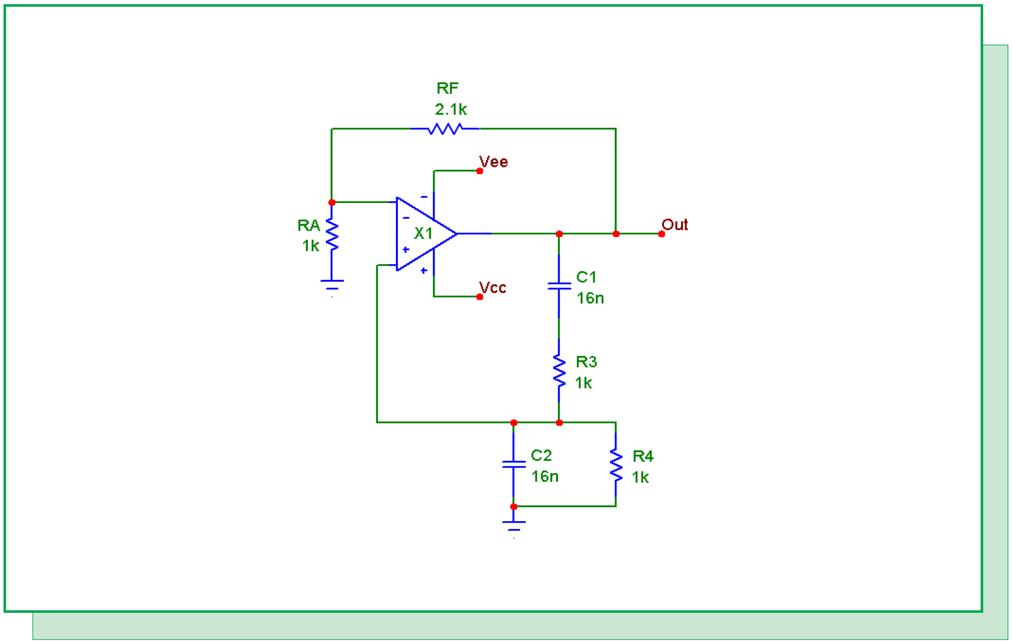
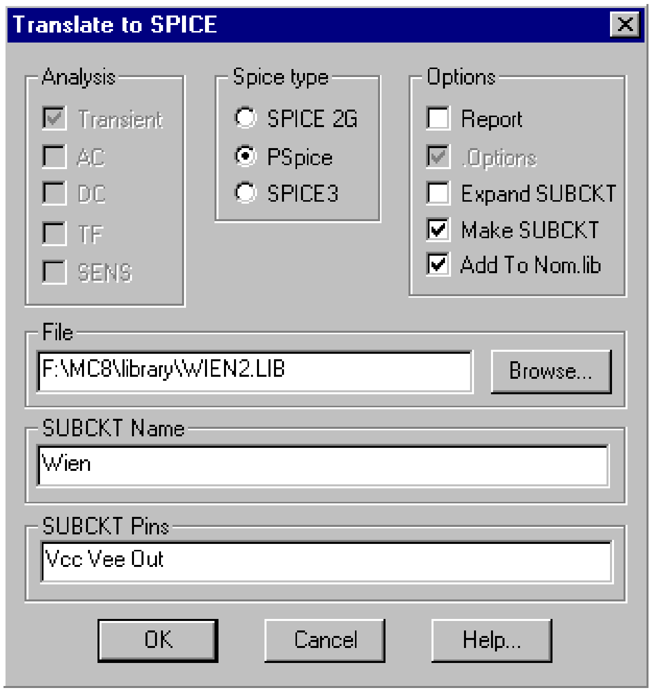


Fig. 2 - Wien bridge oscillator

Once the schematic is ready, click on the File menu, select Translate and choose the Schematic to SPICE Text File option. The Translate to SPICE dialog box that is invoked is displayed in the figure below.



Translate to SPICE

Analysis
☑ Transient
□ AC
□ DC
□ TF
□ SENS

Spice type
○ SPICE 2G
● PSpice
○ SPICE3

Options
□ Report
☑ .Options
□ Expand SUBCKT
☑ Make SUBCKT
☑ Add To Nom.lib

File: F:\MC8\library\WIEN2.LIB

SUBCKT Name: Wien

SUBCKT Pins: Vcc Vee Out

Fig. 3 - Translate to SPICE dialog box

Enabling the Make SUBCKT in the Options section will force the translation to create a subcircuit model in a library file. Both the Analysis section and the .Options option will be disabled since these are only available for a full circuit netlist.

The Spice type section determines the SPICE format that the subcircuit will be created in. The three options are: SPICE 2G, PSpice, and SPICE3. The selection here would depend on the type of syntax that any other programs that may access the subcircuit can handle. Micro-Cap is capable of handling all three of these formats.

The Report option will place a section of comments in the library file detailing the components that the subcircuit is comprised of. This will have no effect on the actual operation of the subcircuit. The Expand SUBCKT option will place all subcircuit or macro models that the active schematic references into the library file as separate subcircuit models. This option localizes all the model information for the main subcircuit model into the specified library file. The Add To Nom.lib option will add a reference for the library file that is being created into the NOM.LIB file. The NOM.LIB file contains a list of library files that Micro-Cap will automatically access. This option should be enabled if the new subcircuit model may be used within Micro-Cap.

The File field contains the path and name of the library file to be created. For Micro-Cap use, the library file should be created with a .LIB extension.

The SUBCKT Name field specifies the name of the subcircuit model. The name must be a single string with no spaces.

The SUBCKT Pins field defines the nodes that the subcircuit will use for its input and output connection pins. The pins should be node names or node numbers from the schematic and should be separated by commas or spaces in the list. If you have named the node in the schematic with a text label, you must use the name rather than the corresponding node number since the text name has priority when the translation occurs.

Clicking on the OK button initiates the translation with the specified settings. The new library file will then be loaded in Micro-Cap. For the Wien Bridge Oscillator example, the subcircuit produced is:

```
.SUBCKT Wien Vcc Vee Out
C1 Out 4 16N
C2 3 0 16N
R3 3 4 1K
R4 0 3 1K
RA 0 1 1K
RF 1 Out 2.1K
X1 3 1 Vcc Vee Out UA741_TI
.ENDS Wien
```

Note that in the .SUBCKT line above the name of the model is Wien and the pin names are Vcc, Vee, and Out as was specified in the Translate to SPICE dialog box. To include the UA741_TI model that the Wien subcircuit references in the same library file, the Expand SUBCKT options would have needed to be enabled.

When the Make SUBCKT option is selected, the translation will also create an entry for the new subcircuit in the component library so that this part will be available for placement in a schematic. The new entry is placed in the SUBCKT.CMP file in the main MC8 directory, and the component will now be accessible through a Subckt group on the first level of the Component menu.

Since Micro-Cap does not inherently know the function of the subcircuit, the new component entry uses the AutoMacro shape. This shape provides a resizeable rectangle where the pins can be moved through a drag operation. Typically, one would want to edit the entry so that it uses a more recognizable shape that exists in the shape library prior to placing this part in a schematic. To do this, just enter the Component Editor, and select the entry that was created for this subcircuit. It will be present in the Subckt.cmp section of the component tree on the right hand side of the dialog box. Once selected, just change the Shape reference for the part and then move the pin connections to the appropriate parts of the shape. If none of the existing shapes are appropriate for the subcircuit, a new shape would need to be created in the Shape Editor.

Assuming the Add To Nom.lib option was enabled for the translation, this subcircuit is now ready to be used in a Micro-Cap simulation. If the Add To Nom.lib was disabled, the Nom.lib file would then need to be manually edited or an explicit reference to the new library, such as with a .lib statement, would need to be defined in the schematic that is using the subcircuit component before this subcircuit would be available for simulation.

A simple test circuit has been set up to test the Wien bridge oscillator. Fixed Analog components have been used to place twelve volts at the Vcc pin and minus twelve volts at the Vee pin. The transient analysis output of the oscillator is displayed in Figure 4.

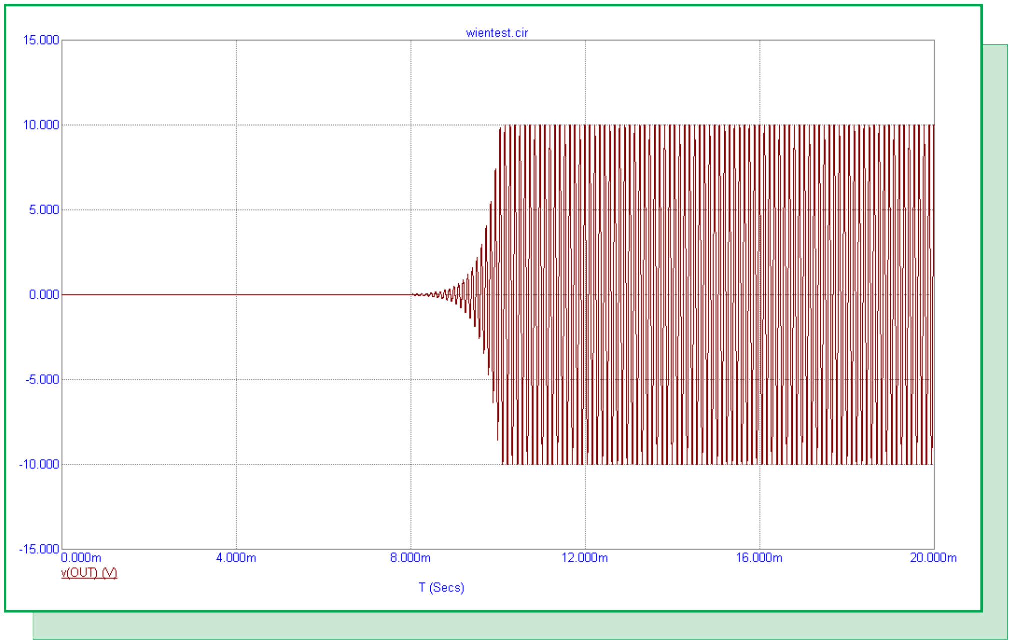


Fig. 4 - Wien bridge oscillator output

The transient simulation has been run over a time range of 20ms. The output produces a sine wave shape whose peaks and valleys are flattened due to saturation at the output rails of the UA741 opamp referenced within the oscillator. The output frequency of the sine wave is measured to be approximately 8.74kHz.

Smith Chart and Impedance Plots

The Smith chart provides a means to view both the complex reflection and the complex impedance of a port. It is most applicable when designing in the RF range for applications involving transmission lines, amplifiers, and antennas among others. One common usage for the Smith chart is to view the input impedance at a port. This article describes the method for correctly plotting the input impedance to a Smith chart in Micro-Cap.

For this example, we will use a simple parallel RC combination since the impedance waveform for that configuration is well known. The initial circuit is displayed in Figure 5. It consists of the RC combination and a Voltage Source component whose AC magnitude parameter is defined as 1V.

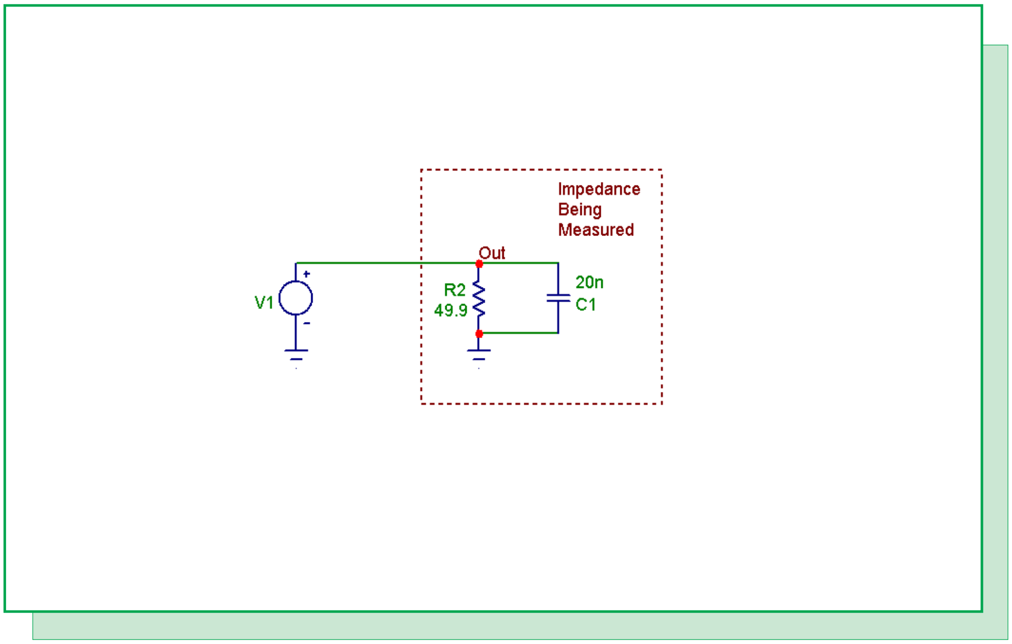


Fig. 5 - Standard impedance measuring circuit

For a standard Cartesian plot, the V/I equation can be used to plot the impedance. In this case, the impedance would be measured using the following expression:

-V(OUT)/I(V1)

which divides the voltage across the input port by the current going into the port. The negative sign is to account for the fact that the I(V1) is calculated from the positive node to the negative node of the source which would be flowing out of the port. The AC analysis of this plot is displayed in Figure 6. At low frequencies, the impedance is dominated by the resistance, and at high frequencies, the capacitor impedance dominates. The most common error when trying to plot the impedance in a Smith chart is to use this same technique.

In order to plot the complex impedance in a Smith chart, the expression that needs to be plotted is the S parameter equivalent. The S11 parameter represents the input impedance and the S22 parameter represents the output impedance. To measure the S11 parameter, the circuit will need

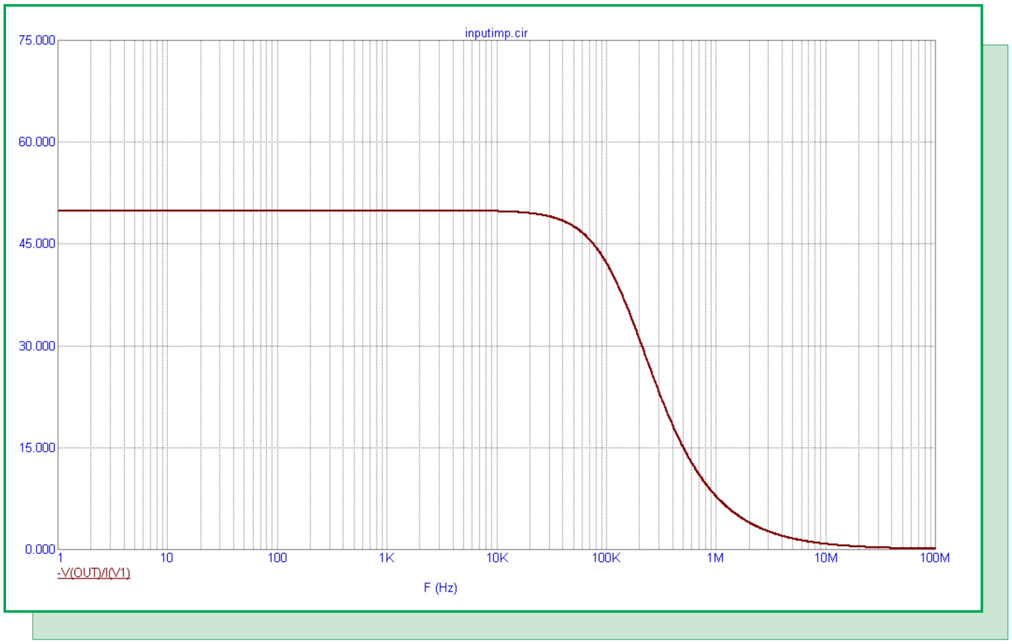


Fig. 6 - Cartesian impedance plot

to be slightly modified as shown in Figure 7. A resistor has been added between the voltage source and the impedance to be measured. The value of this resistance will determine the normalization factor that is used when creating the Smith chart. For this circuit, the S11 parameter is determined by the following equations:

b1 = S11*a1 + S12*a2
a1 = Normalized incident voltage at port 1=(V(Out)+R1*I(R1))/(2*sqrt(R1))
b1 = Normalized reflected voltage at port 1=(V(Out)-R1*I(R1))/(2*sqrt(R1))
a2 = 0 when measuring S11

This reduces the equation to:

S11 = (V(Out)-R1*I(R1)) / (V(Out)+R1*I(R1))

From Kirchoff's law, we determine the voltage loop at the input as:

-V(V1) + R1*I(R1) +V(Out) = 0 where V(V1) is 1 volt for the AC analysis

Plugging this back into the S11 equation, the final expression becomes:

S11 = 2*V(Out) - 1

The resultant Smith chart when plotting this expression is displayed in Figure 8. Note that this plot has been normalized to the source resistance of 50 ohms. The impedance plot starts approximately at the real, imaginary values of 1,0. It then arcs through the capacitive half and ends at the short circuit point of the Smith chart as would be expected by a parallel RC impedance.

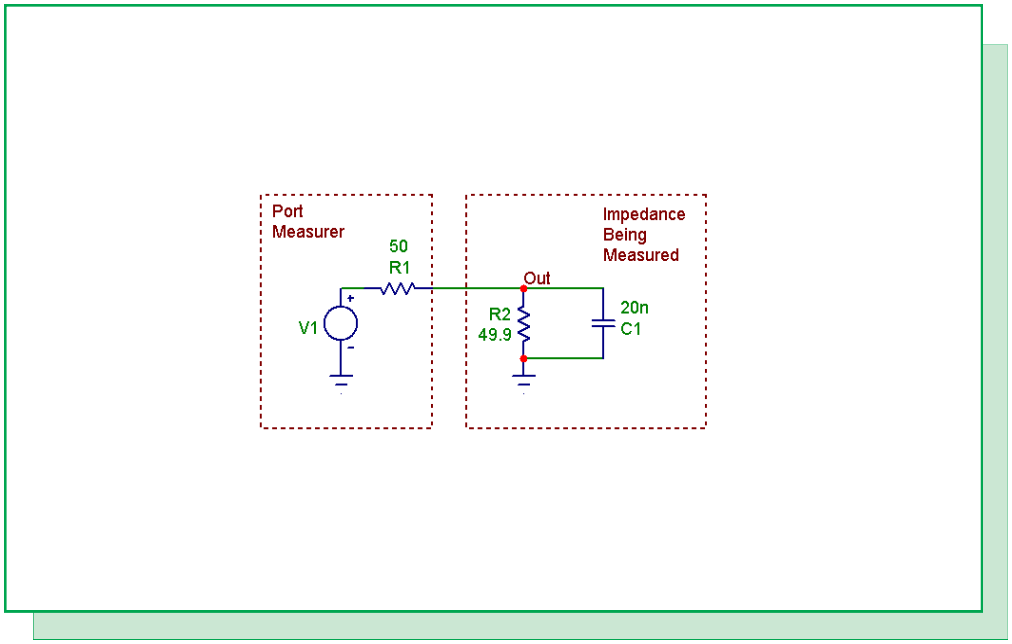


Fig. 7 - Smith chart impedance circuit

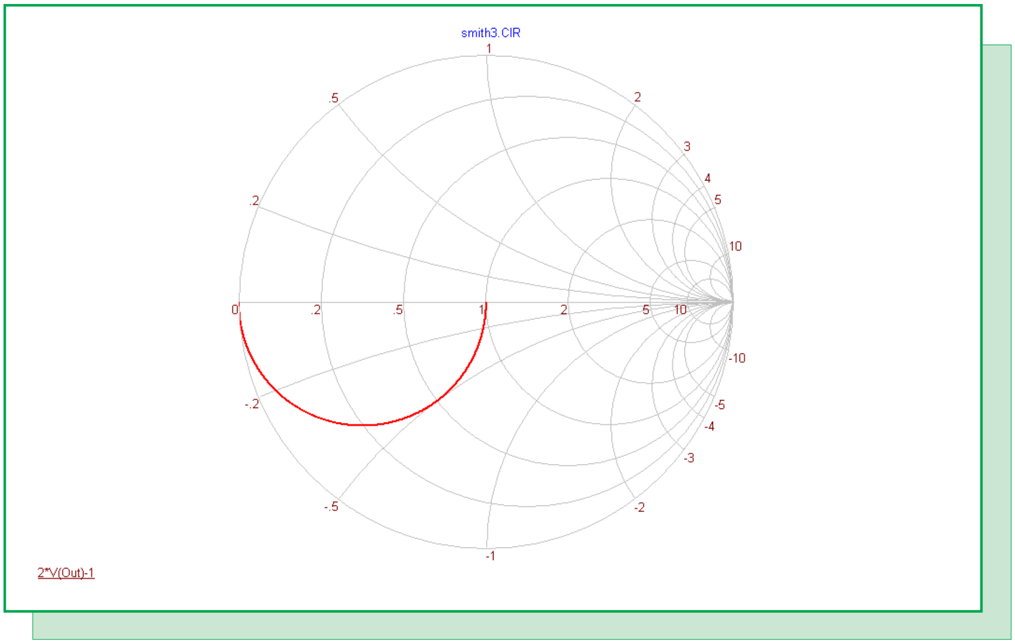


Fig. 8 - Smith chart impedance plot

Voltage Limiting Current Source Macro

The SPICE constant current source available in Micro-Cap is an ideal source. It will produce the specified current throughout the simulation run. This means that the voltage across the source needs to have an infinite range since the specified current must be generated for any topology. This feature of the current source has the potential of creating unrealistic voltages across the source. For some applications, introducing a current model that has voltage limitation capabilities can be a good improvement. The voltage limiting current source macro derived below is based on a model created by Ray Kendall which is described in the article "SPICE model mimics realistic current source" in the EDN Design Ideas Supplement of June 22, 1995.

The voltage limiting current source macro circuit is shown in Figure 9. This macro takes a single parameter, IValue, which defines the typical current that will flow through the source when the voltage is not being limited. The macro circuit consists of two nonlinear function voltage (NFV) sources and two diodes. The current between the Plus and Minus pins is determined through an iterative process between the NFV sources EL and ED. The ED function source is defined with the equation:

1e6*(I(EL)-{IValue})

which looks at the difference between the current flowing between the Plus and Minus pins and the current value specified as the input parameter. The source then multiplies this difference by a large gain in order to forward bias the DD diode. The EL function source is defined with the equation:

-I(ED)

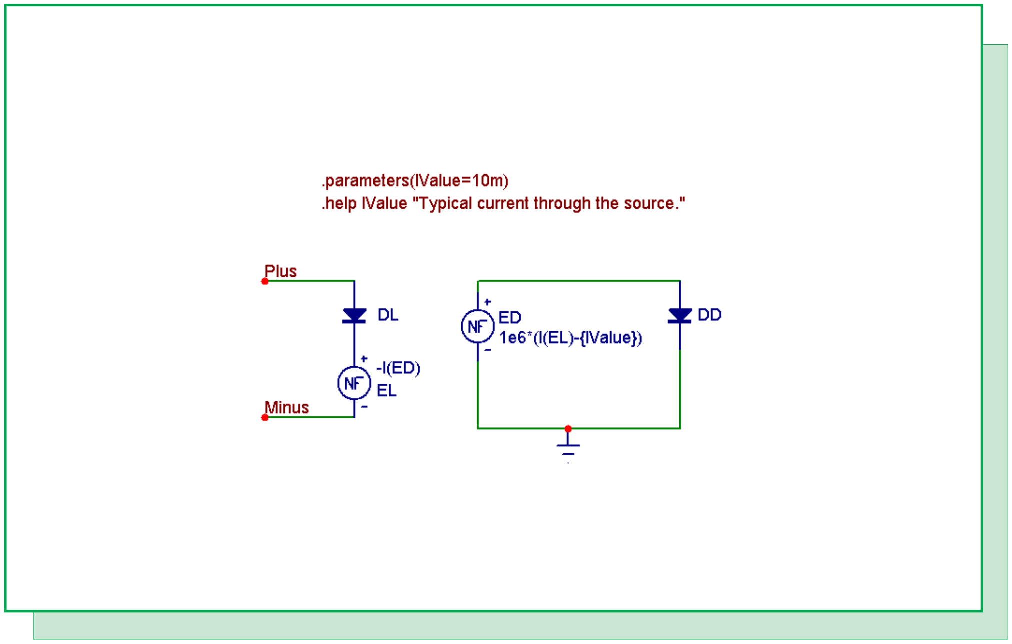


Fig. 9 - Voltage limiting current source macro circuit

which produces a voltage in the EL source equivalent to the negative value of the current flowing through the ED source. The iterative process works such that the current produced in the ED source and DD diode loop will be equivalent to the voltage offset needed for the EL source to create a voltage drop across the DL diode that will cause a current of IValue amps to flow through the diode.

The actual current that will flow through the source will be slightly different than the IValue specified. The size of the difference depends on the gain value used in the ED source equation. For a 20mA source, with the gain set to 1e6, the actual current in the source will be around 20.001mA. To accurately simulate smaller currents with this source, the gain value should be increased accordingly to minimize the difference.

The voltage limitation capabilities of the source are due to the presence of the diodes. The DD diode ensures that the current in the loop will flow in only one direction. This prevents the EL source from producing a negative voltage, so that the current source will be shut down if the differential voltage across it becomes smaller than the diode drop needed to produce the specified current value.

An example schematic for the macro appears in Figure 10. The schematic contains two circuits. One uses the current source macro and the other uses the ISource component. Both of these sources have been defined as a 20mA current source. Each circuit is powered with a 1000V battery. The sources are used to charge a 100uF capacitor at the output.

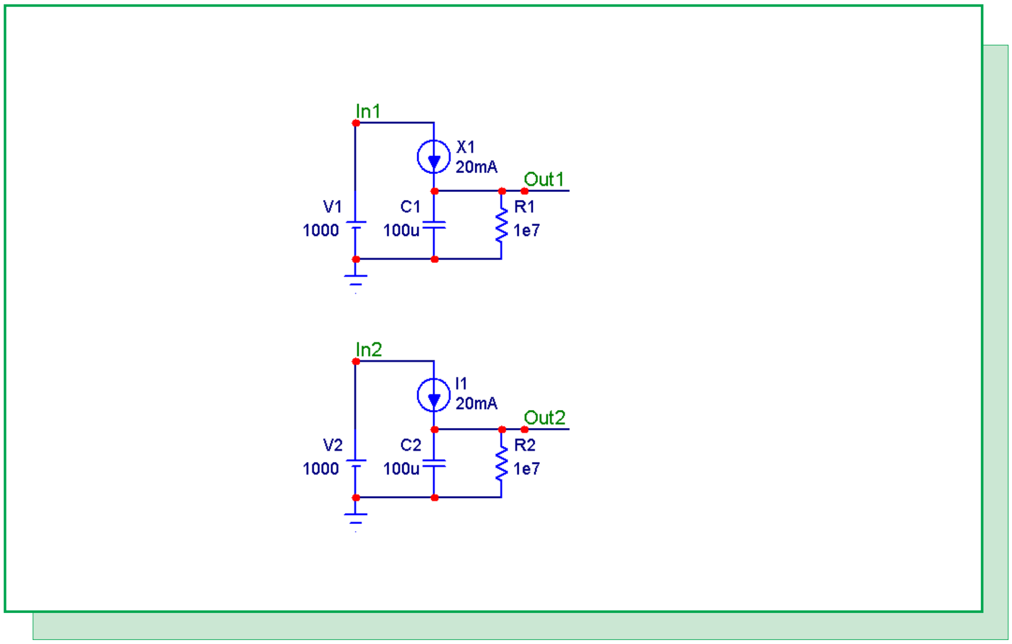


Fig. 10 - Voltage limiting current source example circuit

The transient analysis results for this schematic is displayed in Figure 11. The blue waveforms are from the circuit using the current source macro, and the red waveforms are from the circuit using the ISource component.

For the first half of the simulation, the waveforms match as each capacitor charges up to 1000 volts. As soon as the capacitors hit the 1000 volt mark halfway through the simulation, the waveforms diverge as shown in the plots. At this point, the current source macro shuts itself off as there is essentially zero voltage across it. Note in the middle plot that the current through the source drops from 20mA to approximately 0A when this occurs, and in the bottom plot, the voltage across the current source macro, V(In1,Out1), limits at approximately 0V.

For the ideal source, since it will continually produce the 20mA through the whole simulation, the capacitor continues to charge even though it has exceeded the battery voltage. By the end of the simulation, the C2 capacitor actually has 2kV across it. In order to do this, the ideal current source has to produce a negative voltage across itself.

While the ideal current source is fine for most applications, it is simulations such as this one in which the voltage limiting current source macro provides an analysis that would be more in line with real world expectations.

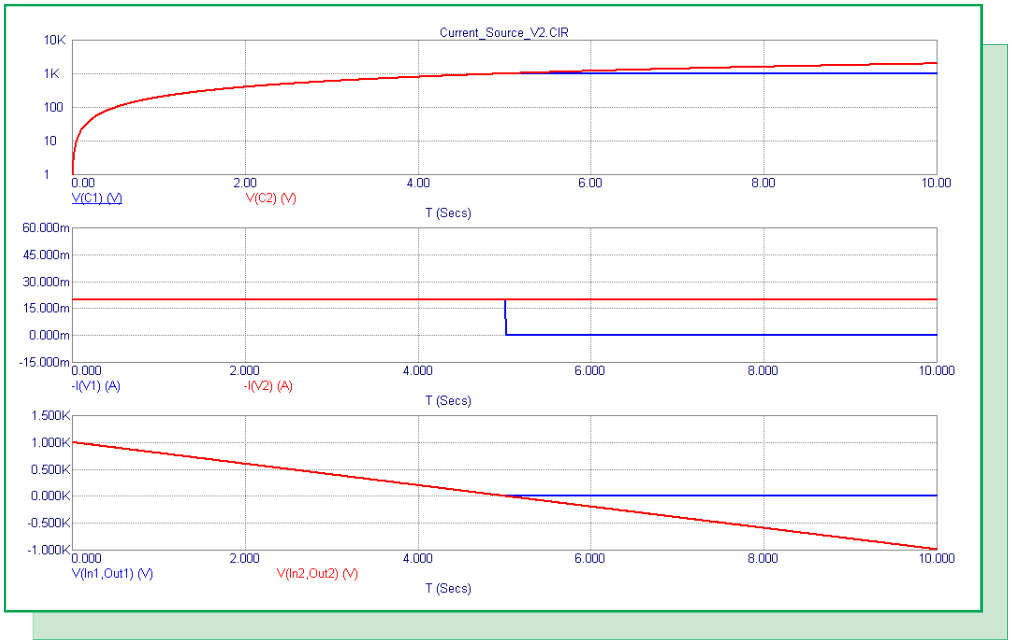


Fig. 11 - Transient analysis results for the current source macro

Product Sheet

Latest Version numbers

Micro-Cap 8	Version 8.0.9
Micro-Cap 7	Version 7.2.4
Micro-Cap 6	Version 6.3.3
Micro-Cap V	Version 2.1.2

Spectrum's numbers

Sales	(408) 738-4387
Technical Support	(408) 738-4389
FAX	(408) 738-4702
Email sales	sales@spectrum-soft.com
Email support	support@spectrum-soft.com
Web Site	http://www.spectrum-soft.com
User Group	micro-cap-subscribe@yahoogroups.com