Appendix: PSPICE and EXCEL Using a Low-pass Filter Example

Appendix: PSPICE and EXCEL Using a Low-pass Filter Example

Introduction

Although an introduction to the detail of PSPICE and EXCEL is not an objective of the book, we need a description of a few features that play a prominent role. Just to be concrete, these features are described using a very simple low-pass filter. Here is the example.

Problem

For a source of known source resistance R_S, design a low pass RC filter to have a given corner frequency f_C and convey the most power possible to an output resistor load R for frequencies below the corner frequency. The circuit is in Figure 1 below.

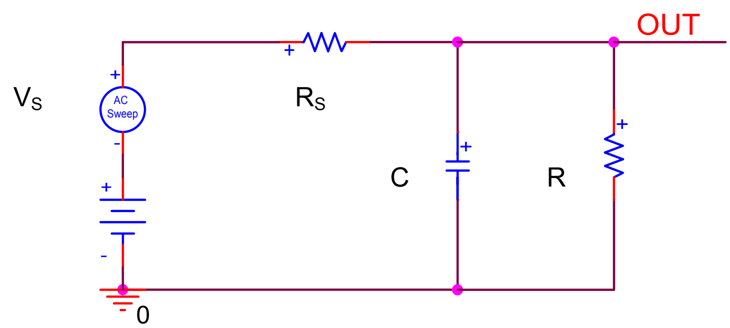


FIGURE 1

Schematic of our low-pass filter; V_S is the signal source, R_S = source resistor, R and C are to be selected

An example transfer characteristic as a function of frequency is shown in Figure 2 below. The corner frequency f_C is the frequency where the gain drops by 3 dB from its low frequency value.

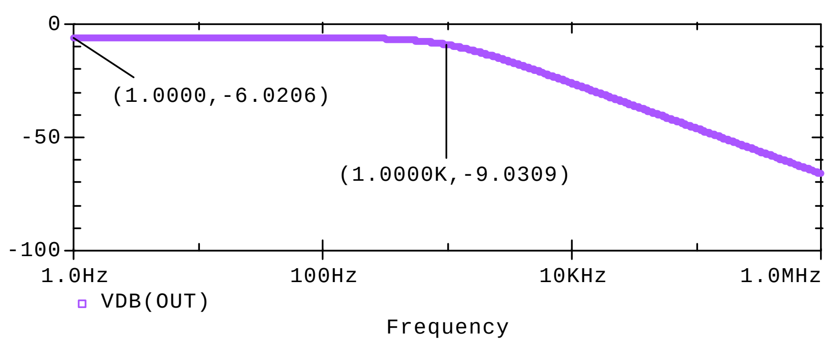


FIGURE 2

Transfer characteristic of the circuit in Figure 1; the corner frequency f_C = 1 kHz is 3 dB down from the value at f = 1 Hz

Analysis

Every design requires some hand analysis. Its purpose is to identify which parameter combinations affect the specifications, and to achieve a qualitative understanding of circuit operation. If we rely entirely on simulation to establish these combinations, it simply takes too many simulations, especially in examples with many parameters that cover a wide range of values.

The standard approach to this design is algebraic, based upon recognizing that this circuit is a voltage divider. EQ. 1 is the transfer function that results.

EQ. 1

$$\frac{V_{out}}{V_S} = \frac{R}{R_S + R}\left(\frac{1}{1 + j\,\omega C(R\,//\,R_S)}\right),$$

where V_{out} = AC output voltage, V_S = AC signal voltage, R_S = source resistance (given) and $\omega = 2\pi f$ = angular frequency in rad/s. The cut-off frequency f_C is defined as the frequency where the denominator of EQ. 1 has equal real and imaginary parts. At $f = f_C$ the bracketed term has a magnitude $1/|1 + j| = 1/\sqrt{2}$, corresponding to a transfer function that is 3 dB below its maximum (zero frequency) value. From EQ. 1 the corner frequency is determined by the design variables R and C as in EQ. 2,

EQ. 2

$$f_C = \frac{1}{2\pi C(R\,//\,R_S)}.$$

Normally, once an algebraic result is obtained, we try to understand it by looking at some limiting cases. For example, according to EQ. 2, as $C \to 0$, $f_C \to \infty$. Why? The answer is that $C \to 0$ is an open circuit, not a capacitor, so we know <u>intuitively</u> that the circuit is just a pair of resistors with the same transfer function at all frequencies, namely the DC value. Hence, there is no cut off and $f_C \to \infty$. As another example, as $C \to \infty$, according to EQ. 2, $f_C \to 0$. Why? The answer is that $C \to \infty$ is a short circuit, not a capacitor. In this case the output is shorted and we know intuitively there is zero output at all frequencies. That is, f_C is zero. Thus, we expect as C ranges between the limits of zero and infinity f_C will do the opposite, as EQ. 2 suggests. Intuition <u>suggests</u> these limiting cases, providing a check of EQ. 2. This interplay between intuition and circuit behavior is useful with hand analysis and with simulation.

EQ. 2 is readily rearranged to provide C in terms of any desired corner frequency f_C as

EQ. 3

$$C = \frac{1}{2\pi f_C (R\,//\,R_L)}.$$

The power to the load assuming sinusoidal input of amplitude V_{in} is $W = I^2 R/2$, or[1]

EQ. 4

$$W(R) = \frac{1}{2}\left(\frac{V_S}{R + R_S}\right)^2 R\,,$$

which has a maximum when $R = R_S$ (as determined setting the derivative with respect to R to zero). Consequently, the design is satisfied by the component values of EQ. 5 below.

EQ. 5

$$R = R_S$$
$$C = 1/(\pi f_C R_S).$$

EQ. 5 determines the circuit component values in terms of the specifications and so completes the analytical design.

Next we will discuss using PSPICE to simulate this circuit. To begin, we start with a description of CAPTURE and an efficient way to organize PSPICE files.

---

[1] The factor of one half results from the r.m.s. amplitude for a sinusoidal input, $I(\text{r.m.s.}) = I/\sqrt{2}$.

Organization of a project in CAPTURE

CAPTURE has three very useful functions: it is a file management tool, it is a drafting tool for circuit schematics, and it is a translation tool that converts the schematic to a list of components and connections readable by a simulation engine. The simulation engine of CAPTURE is PSPICE, and the graphics engine is PROBE. The results of simulation are accessible through PROBE, both as text in the OUTPUT FILE and as xy-plots. To learn the rudiments of CAPTURE, PSPICE and PROBE, please consult a handbook like that by Herniter referenced at the end of this chapter.

Because it is "only" bookkeeping, file management often does not receive enough attention, and headaches result. Here is a useful scheme for project organization.

PLACING FILES

When starting a new project in CAPTURE using the menu system FILE/NEW/PROJECT, as shown in Figure 3, the menu of Figure 4 appears.

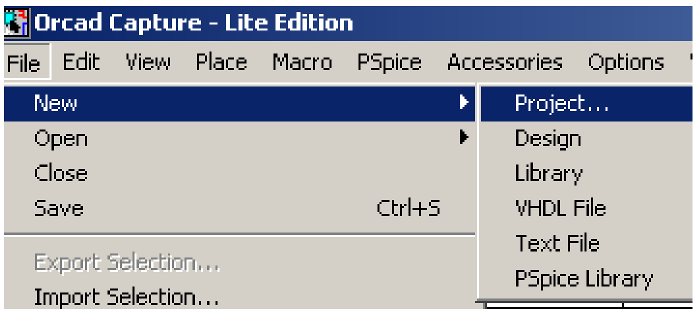


FIGURE 3

Starting a new project in CAPTURE

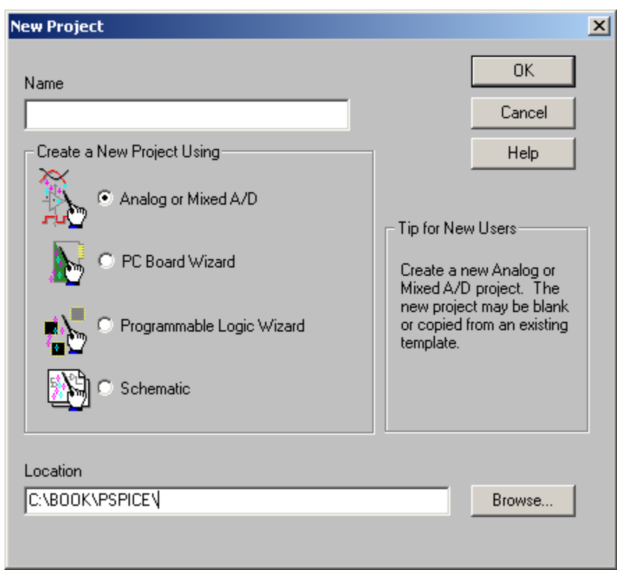


FIGURE 4

The NEW PROJECT menu for selecting the location for the new project

Notice the window at the bottom of the NEW PROJECT menu named LOCATION. By hitting the BROWSE button at the right of this window you can select where to put the new project file. The BROWSE button leads to the SELECT DIRECTORY menu shown in Figure 5 (see Herniter, p. 5).

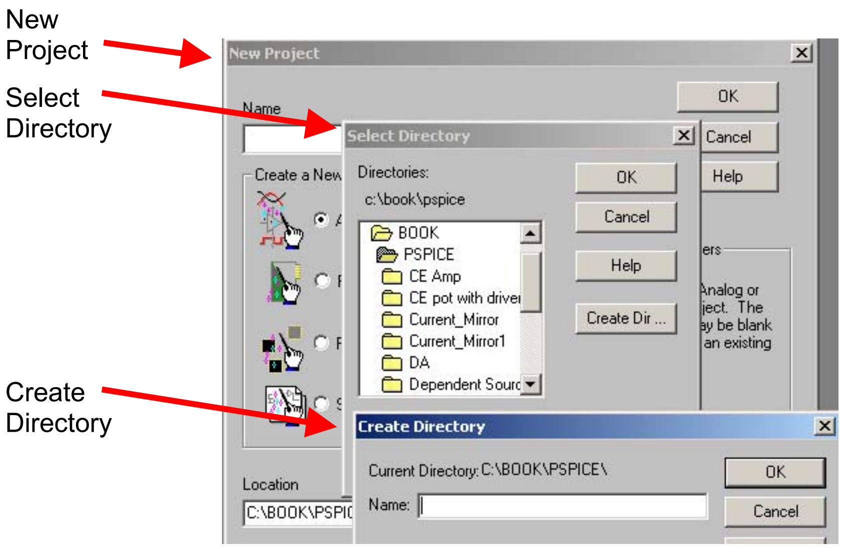


FIGURE 5

Creating a new folder for the project using the CREATE DIRECTORY button on the SELECT DIRECTORY menu.

I strongly recommend that each project be placed in its individual folder with a recognizable name. Following this procedure has the advantage that the dozens of files generated by this project are grouped together and not mixed up with other projects. As a result you can locate, copy, delete, or transfer the files for this project readily.

THE PROJECT MANAGER

As an example, we will use simulation of a low-pass filter as shown in Figure 1. Each project in CAPTURE has a PROJECT MANAGER accessed from the toolbar as shown in Figure 6.

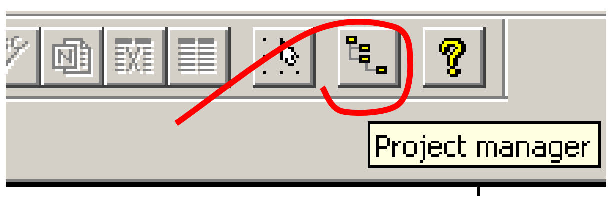


FIGURE 6

Icon for selecting the PROJECT MANAGER

The PROJECT MANAGER for the low-pass filter project is shown in Figure 7.

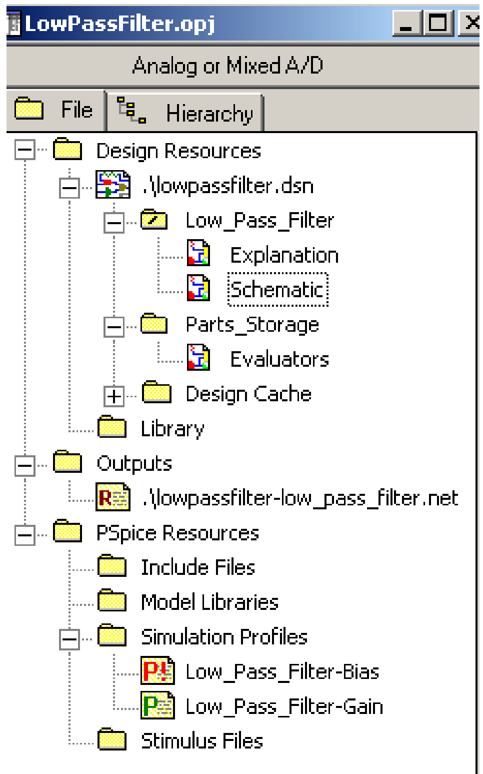


FIGURE 7

PROJECT MANAGER in CAPTURE

This facility is more useful than might first appear. For instance, the project manager enables addition of extra schematic pages by right clicking on the schematic branch label, in this case Low_Pass_Filter, as shown in Figure 8.

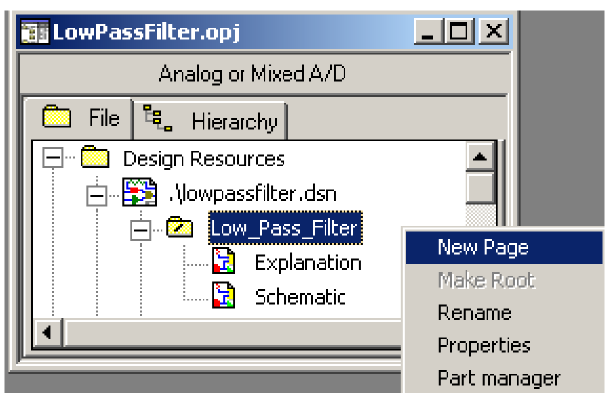


FIGURE 8

Insertion of a new schematic page using PROJECT MANAGER

For this example of a low-pass filter, the circuit schematic is contained on the page named SCHEMATIC. There is another page called EXPLANATION listed in Figure 7 and illustrated in Figure 9. While not necessary to the operation of the project, this page is useful if you return to the project when memory has become dim, or if you hand the project to a colleague. It also can be used to log revisions to the project, or to cross-reference related projects.

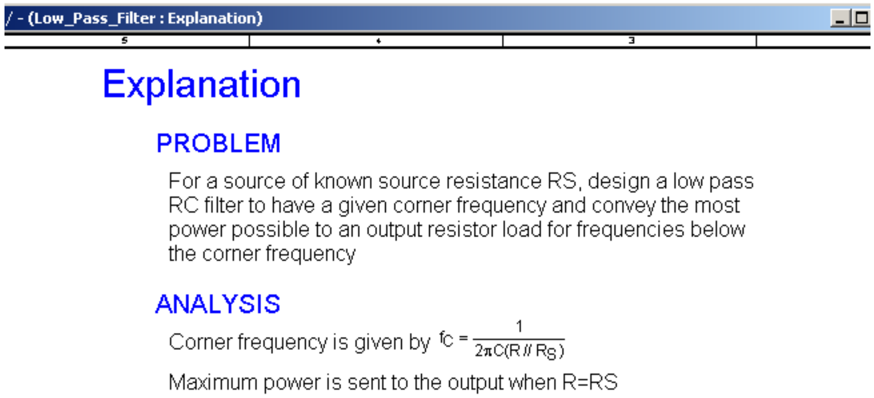


Figure 9

The Explanation page in the low-pass filter project reminds the user what the Project is about

Naming files

Although Capture recognizes names with spaces, I recommend against it. If you zip the files for the project, or mail the files to another computer, often the spaces in file names get removed. Then *inside* the files the file names have spaces, but their actual names no longer have spaces. When this happens, the project will not run.

Back-up of files

As with most software applications in Windows, system malfunction is a common occurrence, and back up is a necessary annoying activity. With Capture, these system failures take the form of freeze-up, or of failure to locate parts of a project, or automatic closure of a project just as it begins to open. See Figure 10.

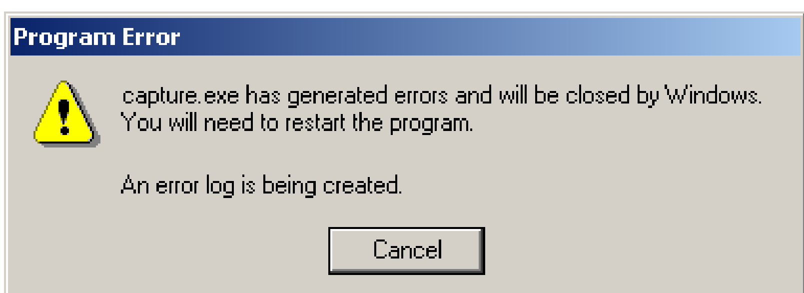


Figure 10

The fatal menu that tells you the project will not open[2]

One never knows when the entire project will become inaccessible. To create a back up, Capture has an Archive feature accessed by highlighting Design Resources in the Project Manager and selecting File/Archive Project. Selecting this option produces the Archive Project menu, which again has a Browse button, but identified only with an ellipsis "...". Hitting this Browse button reveals the Select Directory menu, which has a Create Directory button. The whole sequence is shown in Figure 11. In Figure 11 the archive is placed in a subfolder of the original project folder. Once the file is named, hit OK to obtain the Archive Project menu. I leave the default checking of all boxes and hit OK again.

Back up often. If you want to use only one archive file, you will have to delete the old one before making the new archive because Capture will not overwrite an existing archive file.

---

[2] Often the failure to open is related to the Probe files in the project. By deleting all files in the project that are not ordinarily saved by the Archive feature, sometimes the project can be made to open.

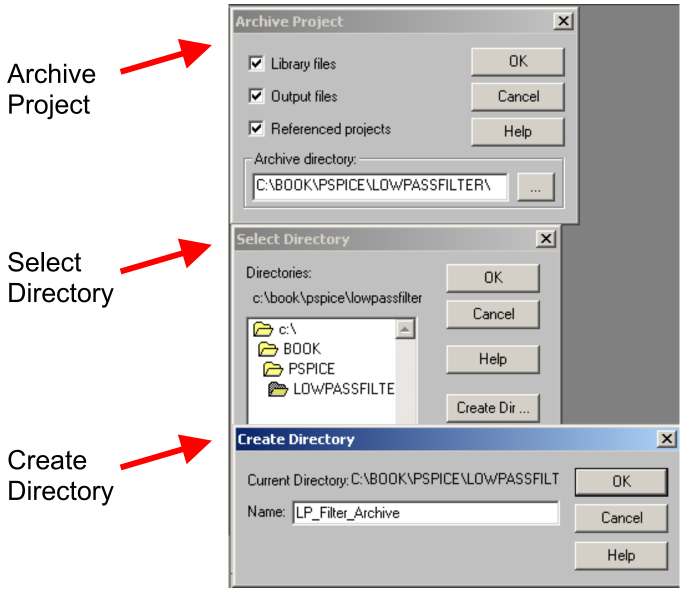


FIGURE 11

The sequence of menus for naming the back up file, named LP_Filter_Archive in this example

The files for the project now look like Figure 12. Notice that the ARCHIVE feature saves only a subset of the files, resulting in a substantial saving in memory. (Actually, the saving is usually much larger than in this example.)

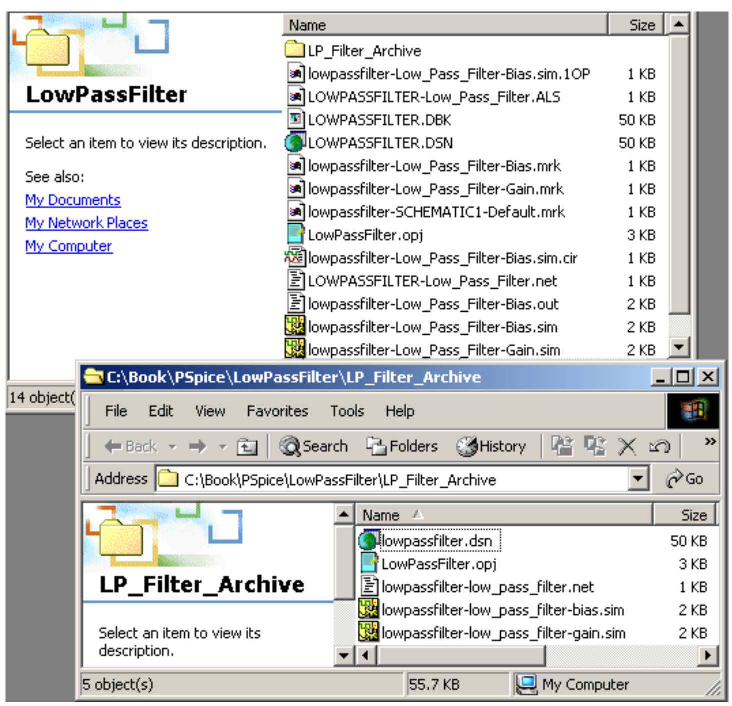


FIGURE 12

The original project files and the archived files for the low-pass filter project; the number of files is reduced from 13 to 5

E-MAIL OF FILES

The archive serves not only as back up, but also as a convenient abbreviated file for e-mail transfer. And if you keep only the archive when you are finished, and delete the original project, you save a lot of memory.

SIMULATION HIERARCHY AND PARTS STORAGE

The PROJECT MANAGER also allows insertion of an independent branch of schematic pages by right clicking on the •dsn label as shown in Figure 13.

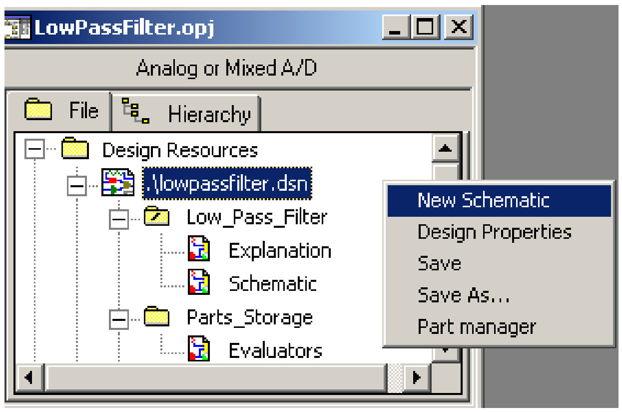


FIGURE 13

Inserting a new branch of schematics outside the simulation hierarchy

In this project example, the second branch of schematics in Figure 7 is called PARTS STORAGE. Because it is a second branch, these schematics are not run when the circuit schematic is run. As a result, one can store items in schematics of this branch for later use, even circuit fragments that are not functional by themselves and would lead to a run time error if executed. In this example, evaluator circuits are stored.[3] They can be copied from storage and pasted on the main SCHEMATIC when one wants them, and deleted when one doesn't want to clutter the main SCHEMATIC. Other occasionally useful items are voltage or current probes that one can copy and paste on the main schematic when wanted, and breakout parts.[4]

SIMULATION HIERARCHY AND SEPARATE SIMULATIONS

The hierarchical branch feature allows different versions of the circuit to be run separately and compared. To show how, we interject another project. For example, suppose we want to compare two versions of the filter circuit as shown in Figure 14.

[3] Evaluator circuits are described in the next main section and pictured in Figure 25.
[4] Breakout parts are custom devices with parameters selected by you, such as bipolar devices with particular choices for Early voltage or for capacitances. They are discussed later.

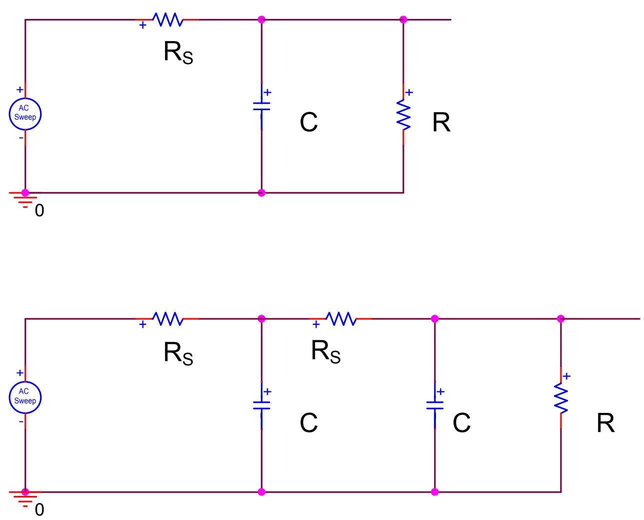


FIGURE 14

Two filters to be compared

We set up the file structure as shown in Figure 15.

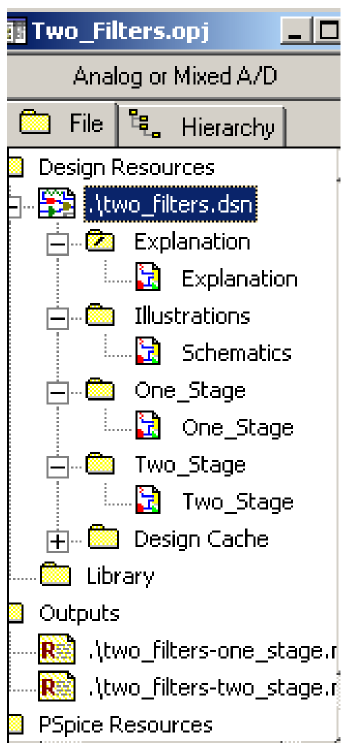


FIGURE 15

PROJECT MANAGER for two-filter example

The project manager shows four branches in the hierarchy: an Explanation branch (project objectives), an Illustrations branch (figures used when writing about the circuit), a One-stage filter branch (with the schematic of the one-capacitor filter) and a Two-stage filter branch (with the

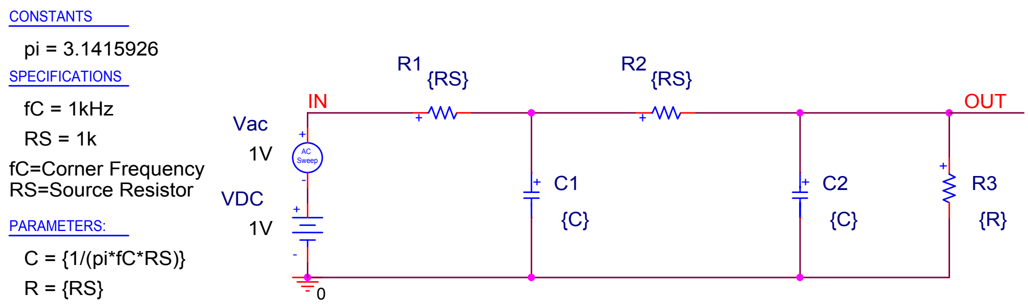


FIGURE 16
Schematic for the two-stage filter

schematic of the two-capacitor filter). The schematic for the two-stage filter circuit in PSPICE is shown in Figure 16. PSPICE schematics are discussed in the next section.

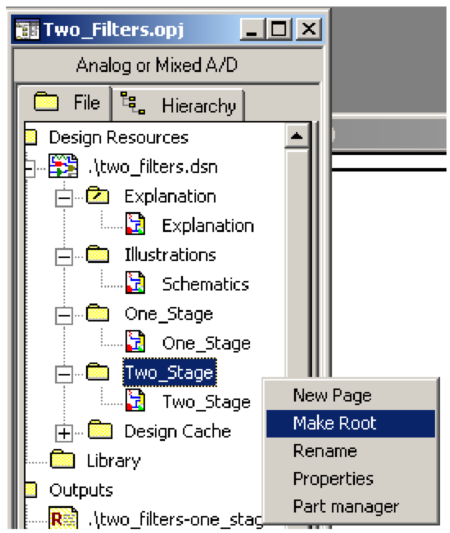


FIGURE 17
Using the MAKE ROOT menu to select the branch to be simulated

Only one branch in the simulation hierarchy is simulated, called the ROOT. To change the root, we highlight the branch we want to simulate and right-click to obtain the menu in Figure 17.

By running the One_Stage branch, changing the root, and running the Two_Stage branch we can make plots for each version of the circuit. If we make a corner frequency plot for the one-stage and two-stage filters, we find the plots of Figure 18.

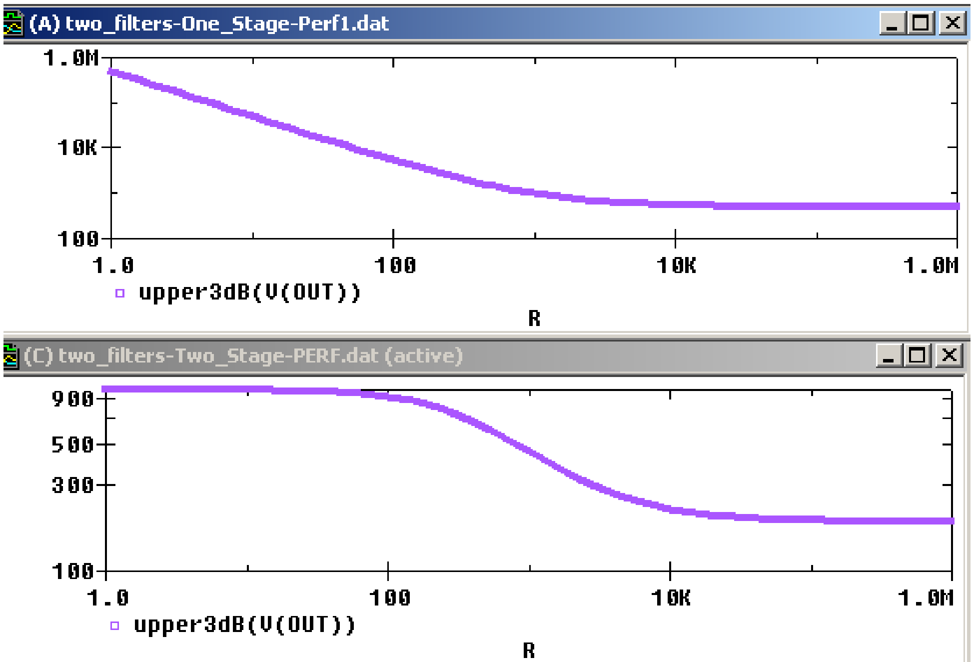


FIGURE 18

Corner frequency as a function of resistor value R for the two filters in Figure 14

This approach also can be convenient when you exceed the transistor count allowed for the free version of PSPICE. You may be able to simulate different stages in different branches. Some ingenuity is necessary to compensate for the stages inability to "talk" to each other.

SUPERPOSITION OF PLOTS FROM DIFFERENT SIMULATION PROFILES

A great feature of CAPTURE is that each simulation profile stores its settings and its graphical output, so even if the circuit is changed, we have preserved the earlier results for comparison. We can put the plots of Figure 18 together in PROBE using the menu FILE/APPEND as shown in Figure 19.

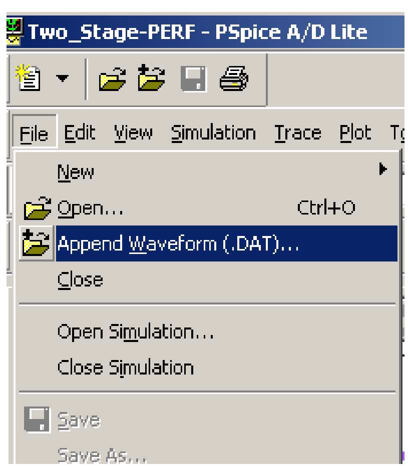


FIGURE 19

Appending a waveform; the APPEND menu results

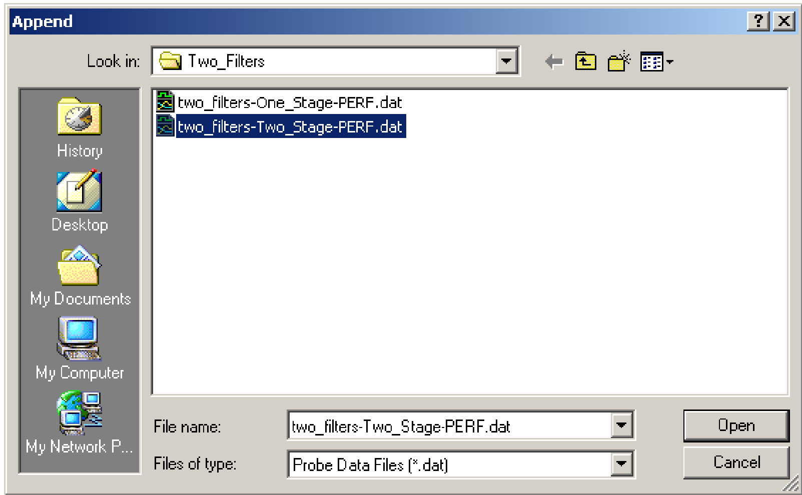


FIGURE 20

Appending the two-stage file to the one-stage plot using the APPEND menu

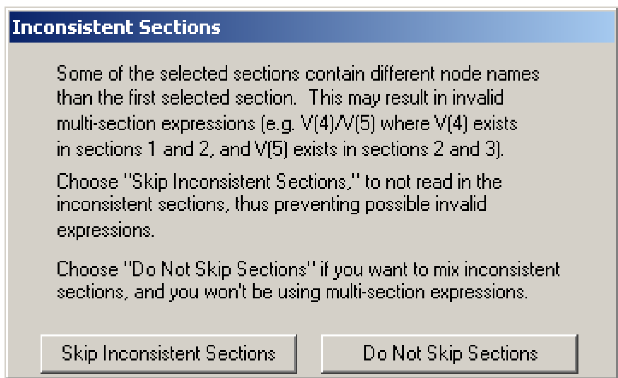


FIGURE 21

We elect do not skip sections

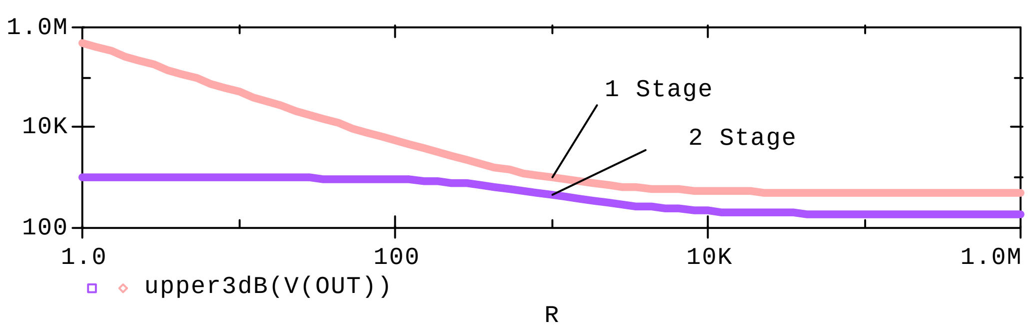


FIGURE 22

The two plots on the same graph

Of course, this example is very simple, and the same results could be obtained without using the append option.[5] The purpose here is to show how to use the append option, so you can do it when it is advisable.

[5] For example, we could put both circuits in the same branch so they simulate simultaneously. We would have to be careful to use different part numbers in the two schematics, for example, resistor R_1 cannot occur twice. Also the output nodes could not both be named OUT.

SUMMARY OF FILE MANAGEMENT

There are many files in a project, as exampled in Figure 12, which suggests the importance of keeping each project in a file separate from other projects. Besides the convenience of separate files for normal maneuvers like copying and so forth, there is the extra problem that every so often a project becomes detached detritus, and has to be deleted.

Including an EXPLANATION page can prod a fading memory when a project is revisited, log its history and cross-reference related projects.

Archiving a project is valuable for back up, for e-mail, and for storing a project in a small memory space.

Using multiple branches for the simulation hierarchy, the PROJECT MANAGER makes possible the storage of parts that can be copied and pasted into active schematics. It also makes possible the separation of different phases of the project, for example small-signal analysis can be separated form large signal, or one circuit variation can be separated from another. This hierarchical separation often is more convenient than using multiple projects. Results from different branches can be superposed in PROBE using the FILE/APPEND option.

PSPICE simulation of the design

We are not looking for any revelations about circuit operation here, but aim to illustrate some features of PSPICE. First we put the circuit of Figure 1 into CAPTURE, as shown in Figure 23.

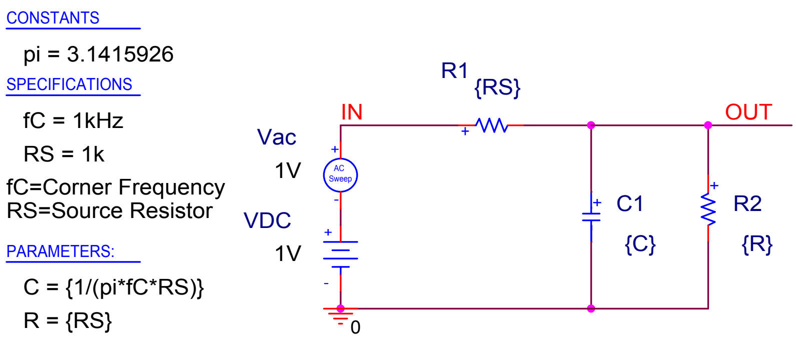


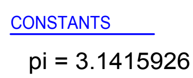
FIGURE 23

The schematic of Figure 1 in PSPICE; EQ. 5 has been incorporated in Figure 1 in the PARAMETERS box.

USING VARIABLES

In Figure 23 the component values all are expressed in terms of variables. The use of variables is indicated by enclosure in curly braces {} (for example, the value of C1 is {C}). These variables in turn are interpreted in terms of formulas using a PARAMETER box (for example, C={1/(pi*fC*RS)}). In the PARAMETER box only the formula on the right is enclosed in {}, and not the variable on the left.

The parameters like that below



also are PARAMETER boxes, but the label has been changed to CONSTANTS by editing the PARAMETER part. This editing is done by highlighting the part, right clicking, and selecting the menu EDIT/PART. Whether this extra clarity is worthwhile depends on who may use this circuit.

The analysis is short in this example and, because the circuit is simple and linear, no approximations are used. One has confidence in the results. Nonetheless, and definitely in the case of more complicated problems, one likes to check the results. In fact, for a more complex circuit one may wish to test one's entire concept of how the circuit works. Often, however, this lofty goal is cut down to a check at only a few points using PSPICE. In this example, for instance, we might run a gain plot like Figure 2 and check the real corner frequency for a few choices of our selected input value of f_C; both frequencies should turn out to be the same.

When formulas from the analysis are used in PSPICE as is done in Figure 23, a more convincing test of the derived formulas is possible. We run a PERFORMANCE ANALYSIS, use PSPICE to find the actual 3dB frequency, and plot it against our selected input frequency f_C as shown in Figure 24.

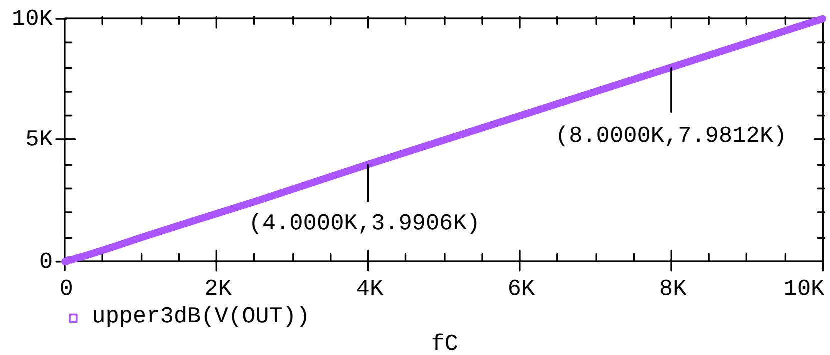


FIGURE 24

A PERFORMANCE ANALYSIS of the actual corner frequency of the design vs. expected corner f_C with component values determined using the formulas shown in Figure 1[6]

Figure 24 shows that for any value of corner frequency 1Hz $\leq f_C \leq$ 10 kHz (and presumably other frequencies as well), the design formulas of EQ. 5 result in the correct corner frequency. Figure 24 is a more complete verification of the design approach than a spot check for a few particular values like Figure 2. Shortly, we will see how such a plot is generated. First, here is a little convenience I stumbled upon that makes PSPICE more like a spreadsheet: the evaluator circuit.

USE OF EVALUATOR CIRCUITS

When formulas are used in PSPICE, the component values that are expressed as formulas, like C in Figure 23, are not displayed on the schematic. Of course they can be calculated, but it is more convenient to use evaluator circuits like those in Figure 25.

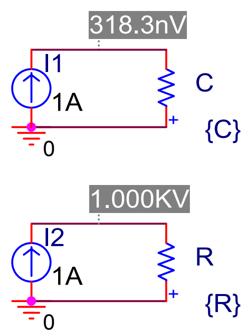


FIGURE 25

Evaluator circuits used to display the values of C and R for the circuit of Figure 23

An evaluator circuit drives a DC current of 1A through a resistor of numerical value given by the formula for the component being evaluated. The voltage across the resistor provides the numerical value of the component. The evaluators in Figure 25 are exercised using a BIAS POINT simulation profile (described next). In Figure 25 we find C = 318 nF, which satisfies the formula $C = 1/[2\pi f_C(R//R_S)]$, and R = 1 kΩ, which satisfies the formula $R = R_S$ (= 1 kΩ in Figure 23).

Another use of the evaluator circuit is that it makes the component value available for plotting in PROBE. For example, using the evaluator in Figure 25, we could plot C as a function of f_C.

SIMULATION PROFILES

Now let us see how to generate a plot like Figure 24. A SIMULATION PROFILE is a simulation with its various options. CAPTURE allows four types of simulation. The possibilities are listed in Figure 26.

---

[6] Generation of this plot requires the use of the GOAL FUNCTION named UPPER3DB(V(OUT)). The use of performance analysis and goal functions is described later.

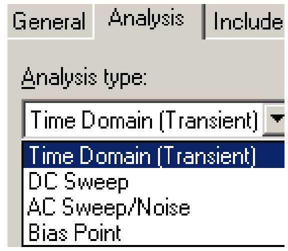


FIGURE 26

The four types of SIMULATION PROFILE

Their names are descriptive, but some details are listed below:

1. BIAS POINT: this simulation determines the DC operating or Q-point of the circuit. (Herniter, pp. 157-192).
2. AC SWEEP: this simulation makes a <u>linearized version of the circuit</u> at the Q-point and does a small-signal analysis of the circuit behavior. For example, one can plot the small-signal voltage gain or its phase as a function of frequency. Figure 2 is an example. (Herniter, pp. 278-326).
3. DC SWEEP: this simulation allows a DC analysis of the Q-point variables as a function of any circuit parameter or device parameter. For example, one could plot DC output voltage as a function of bipolar transistor parameter B_F, which is related to the transistor beta. (Herniter, pp. 193-277).
4. TIME DOMAIN (TRANSIENT): this simulation allows the <u>large-signal</u> analysis of the circuit as a function of time. For example, one could plot the output voltage response to a ramp input and determine whether the circuit can follow the input, or input a sine wave and see whether the output clips. (Herniter, pp. 327-428).

PLOTS IN Probe

Each of the simulation profiles has options. In particular, multiple curves can be generated using a second parameter. For example, for the AC SWEEP simulation profile the main SIMULATION SETTINGS menu is shown in Figure 27, which sets the range of frequencies to be swept.

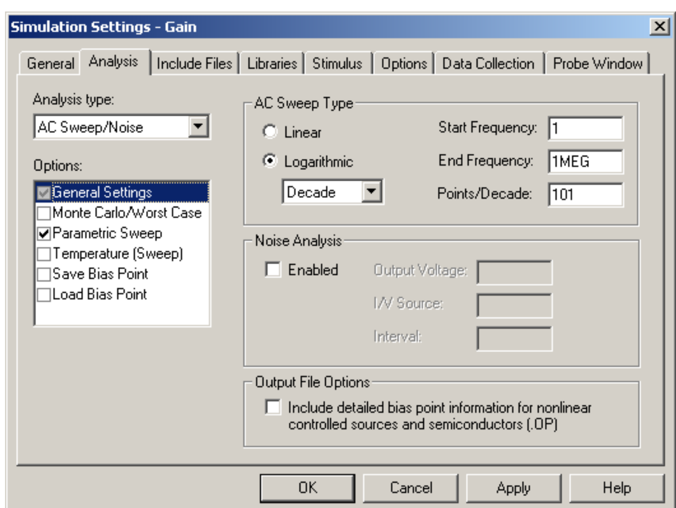


FIGURE 27

GENERAL SETTINGS menu for the AC SWEEP simulation profile; frequency range $1 \leq f \leq$ 1MHz is selected

In Figure 27 the PARAMETRIC SWEEP box has been checked. Clicking on this heading results in the PARAMETRIC SWEEP selection menu of Figure 28. There are five types of parameter. In this book we will focus upon the GLOBAL PARAMETER, which refers to a variable named in a PARAMETER BOX. Examples of global parameters from Figure 23 are f_C, R_S, C and R.

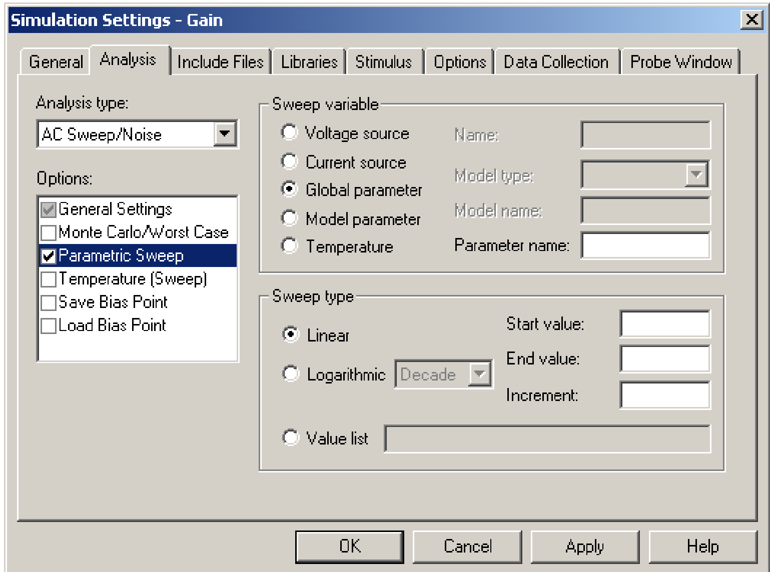


FIGURE 28

The PARAMETRIC SWEEP option of the AC SWEEP simulation profile

SINGLE-FREQUENCY PERFORMANCE ANALYSIS

A very useful option that makes use of the parameter sweep is to choose only one frequency on the GENERAL SETTINGS menu of Figure 27, and to select a range of values for a parameter of interest on the PARAMETRIC SWEEP menu of Figure 28. Example profile settings are shown in Figure 29 and Figure 30.

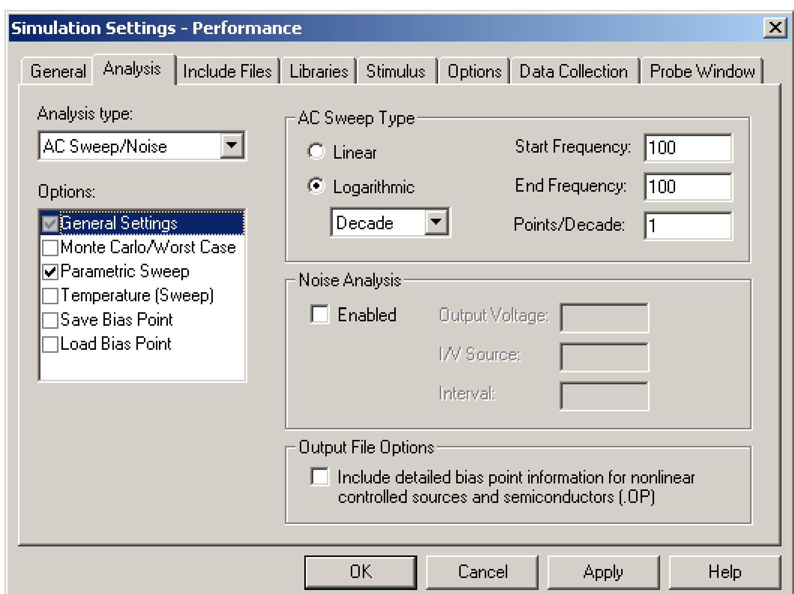


FIGURE 29

Setting up a PERFORMANCE ANALYSIS using only <u>one</u> frequency point in POINTS/DECADE

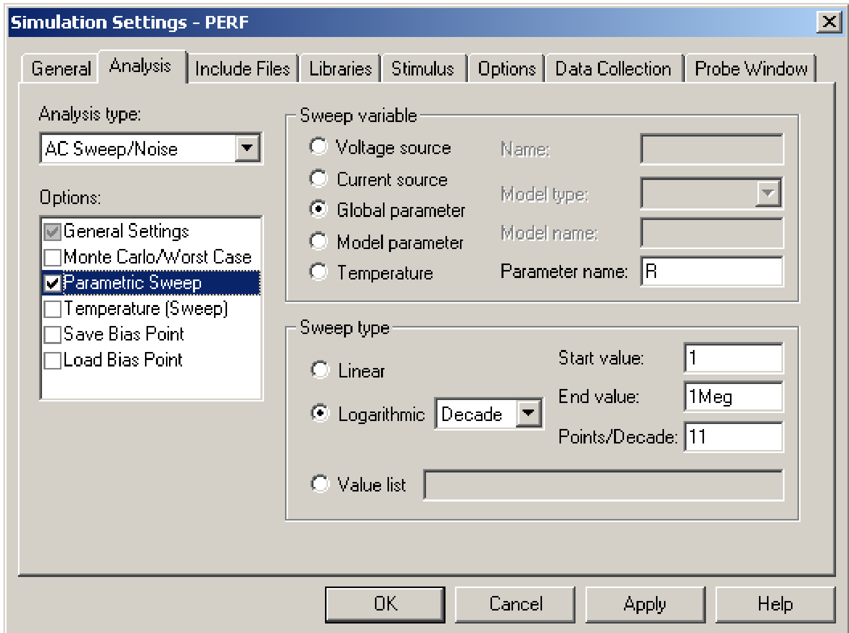


FIGURE 30
The PARAMETRIC SWEEP option with a range of values for variable R

The result of this combination of settings is a PERFORMANCE ANALYSIS, that is, one obtains the circuit variables as a function of the swept variable <u>at the selected frequency</u>.[7] To obtain the plot in PROBE, the simulation profile is run and in PROBE the TRACE/ADD TRACE menu is selected as shown in Figure 31. We can choose to plot anything from the list of OUTPUT VARIABLES, as shown in the left panel of Figure 32.[8]

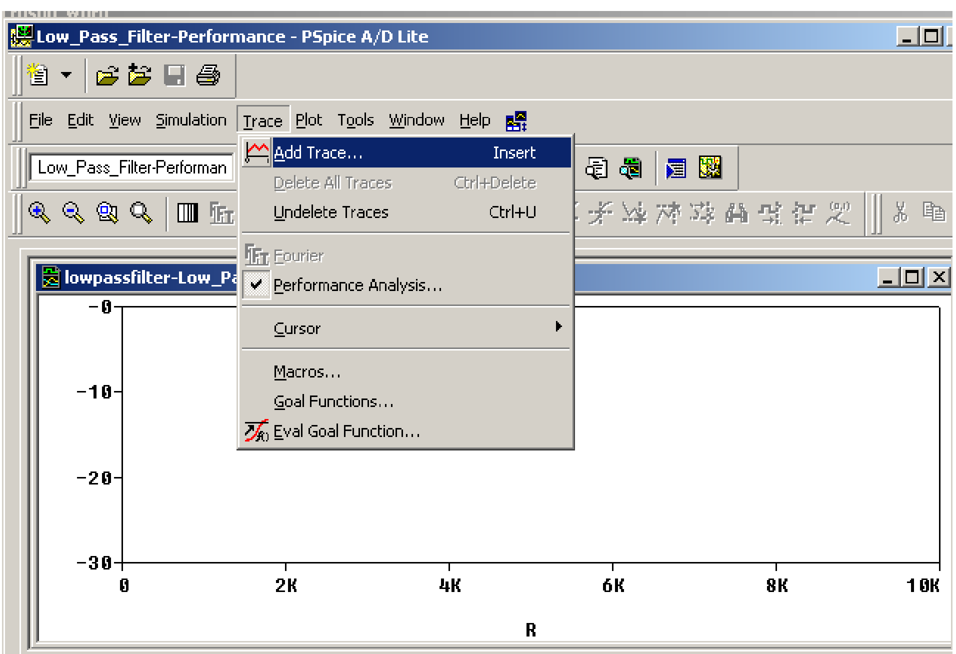


FIGURE 31
Adding a trace to be plotted in PROBE using the TRACE/ADD TRACE menu

[7] The specification of R by the sweep command overrides the value set in the PARAMETERS box.
[8] If you want to plot voltage or current, a MARKER can be placed on the schematic instead. (Herniter, p. 106).

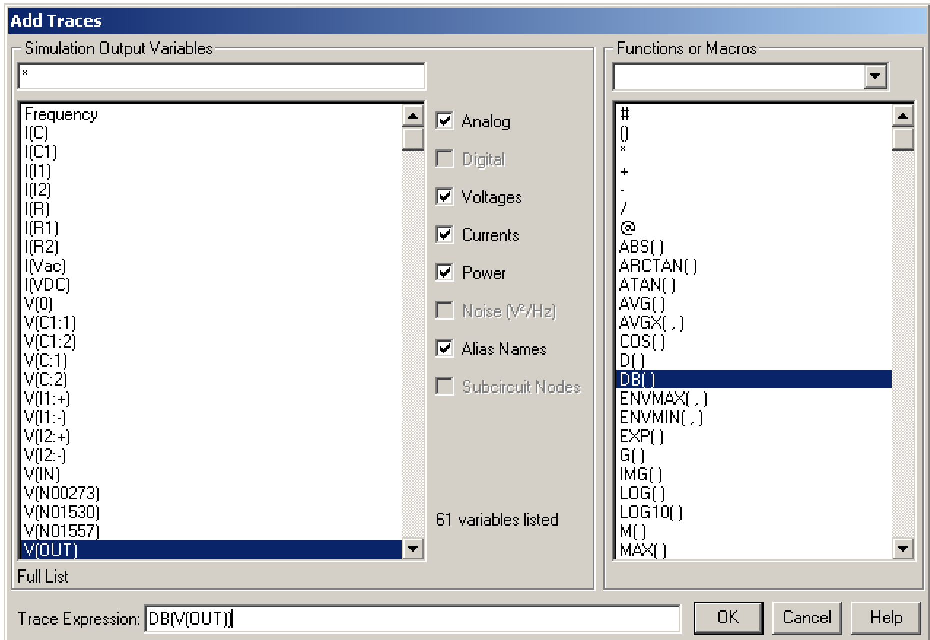


FIGURE 32

Adding the trace of output voltage in dB applying the provided function DB() as DB(V(OUT))

A number of built-in functions are available as well. In Figure 32, the decibel function **DB()** has been applied to the output voltage variable V(OUT) to obtain the trace in Figure 33.

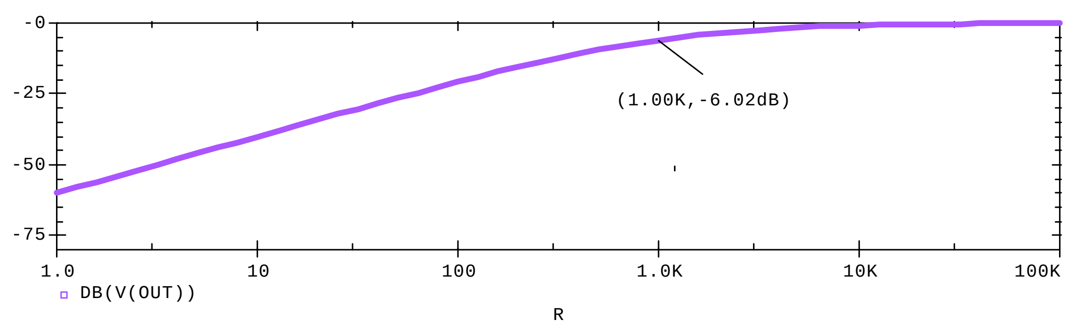


FIGURE 33

Single-frequency performance analysis; adding a function of a variable DB() as a trace in PROBE; the output voltage is plotted in dB vs. the resistor value R of the low filter circuit at a frequency of 10 Hz

MANY-FREQUENCY PERFORMANCE ANALYSIS

A different type of performance analysis uses both a frequency sweep and a parameter sweep. Figure 24 is an example. For a many-frequency performance analysis, instead of Figure 29 we use the settings of Figure 34.

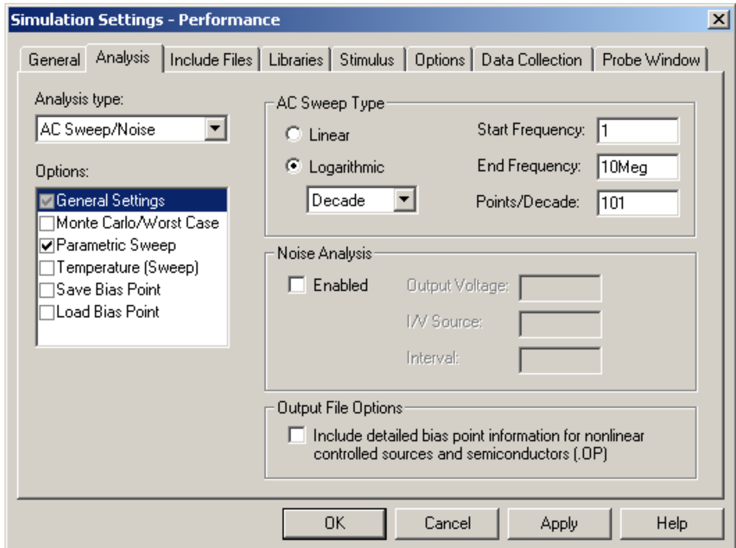


FIGURE 34

Setting a frequency sweep along with a parameter sweep

In this case we obtain a complete frequency sweep for every case requested in the parameter sweep. To obtain a performance analysis in this case we must select it on the AXIS SETTINGS menu in PROBE, as shown in Figure 35. The result is a blank plot with the parameter as x-axis, as shown in Figure 36. We then must use TRACE/ADD TRACE to get a plot. In this case we again have a set of predefined functions to choose from called GOAL FUNCTIONS, shown in Figure 37.[9]
We choose the goal function **upper3dB()** (upper 3dB frequency) to get the corner frequency of the amplifier as a function of the circuit resistor value R. The result is Figure 38

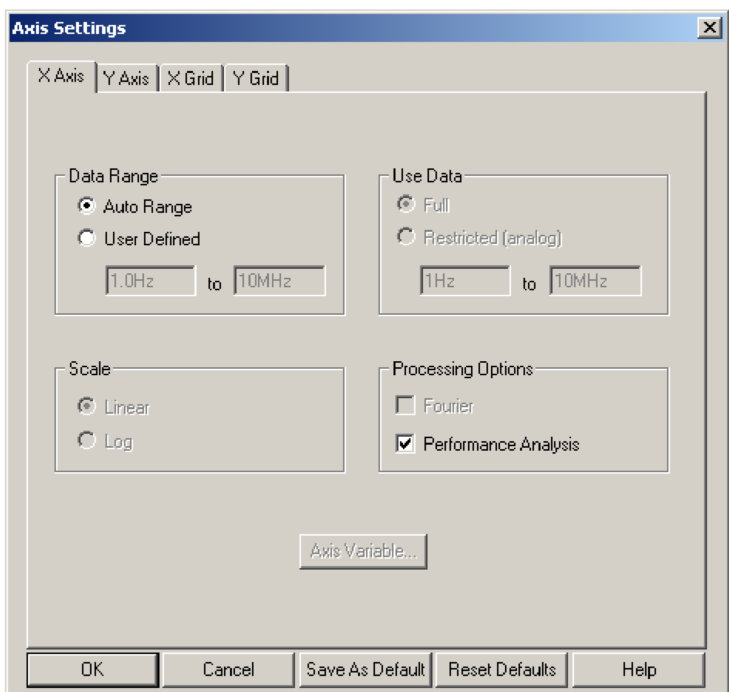


FIGURE 35

Choosing a PERFORMANCE ANALYSIS by checking the PERFORMANCE ANALYSIS box

---

[9] Herniter, pp.312-316

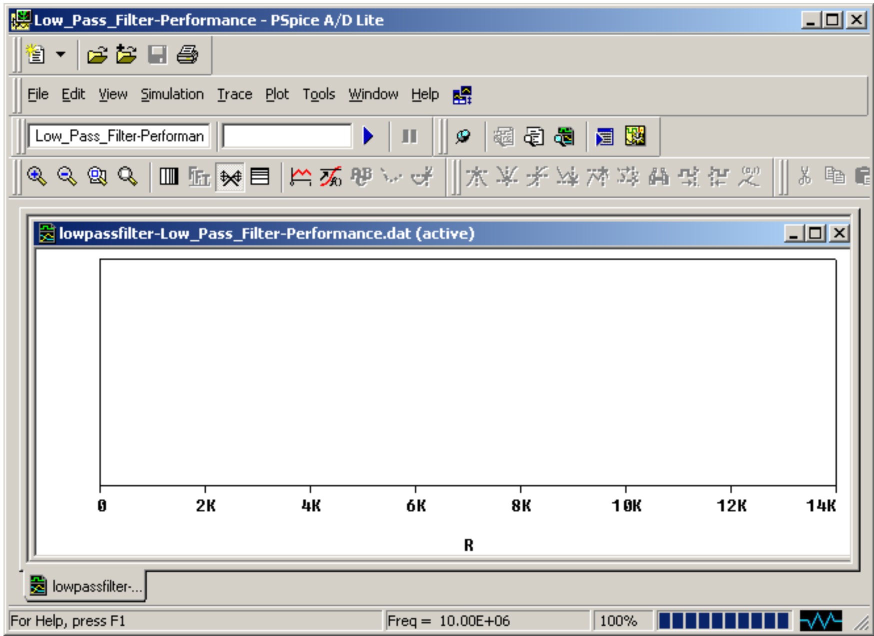


FIGURE 36

Blank plot with R as x-axis before adding a trace

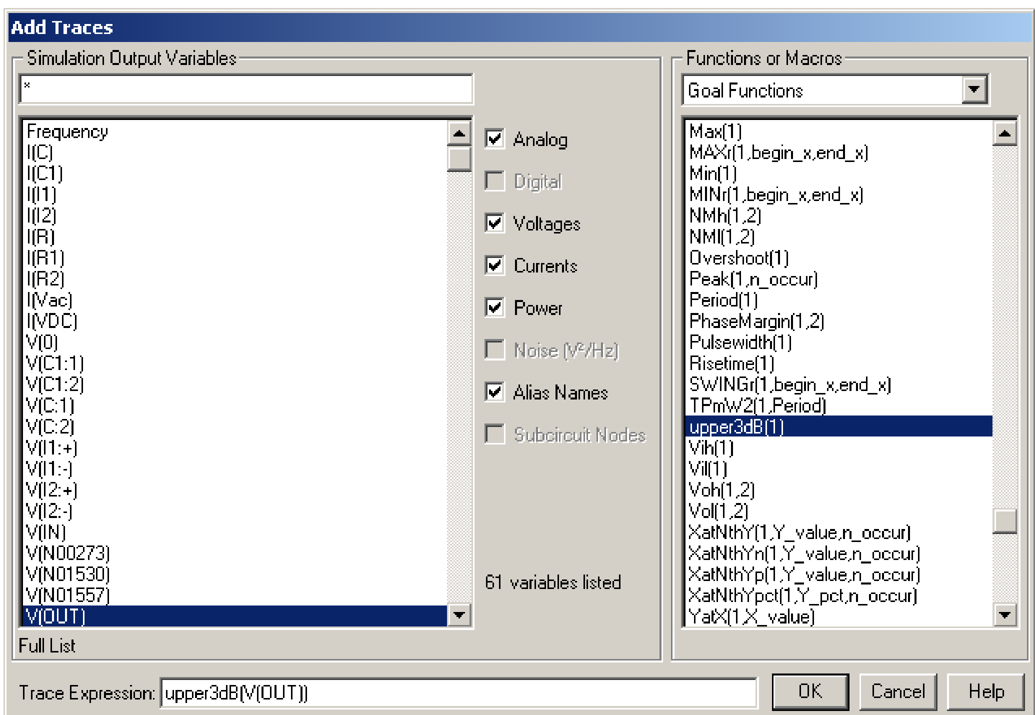


FIGURE 37

Choosing the goal function upper3dB(V(OUT)) to obtain the corner frequency of the filter

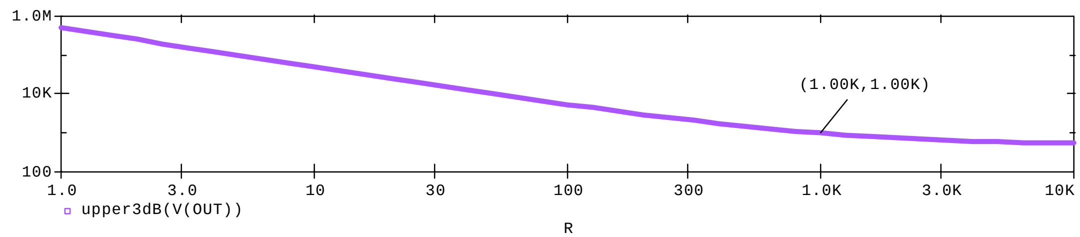


FIGURE 38

Many-frequency performance analysis plot of the corner frequency as a function of resistor value R

If we follow the same steps and use f_C as the swept parameter in the menu of Figure 30 we obtain the plot of Figure 24.

Performance analysis also can be done using TRANSIENT ANALYSIS simulation profiles, where frequency is replaced by time, and small-signal behavior by large-signal behavior.

SUMMARY OF PSpice SIMULATION

The use of variables and formulas in PSPICE to determine component values has been illustrated. A convenient evaluator circuit for formulas was proposed. The use of variables allows incorporation of hand analysis in PSPICE and makes possible a thorough verification of the hand analysis assumptions and results. Such verification requires the use of a number of PROBE plots, chief among them single- and multiple-frequency PERFORMANCE ANALYSIS plots, whose set up has been demonstrated.

Simulation using PSPICE with EXCEL

We now return to the example of Figure 23 and look at the second specification, namely, power transfer to a resistor load. This time PSPICE is used in combination with EXCEL to explore the results.

The functional dependence on R_S, and the power transfer requirement can be verified using PSPICE. Power transfer to the resistor R for a few values of R_S is shown in Figure 39. Figure 39 supports the conclusion that maximum power transfer occurs when $R=R_S$ as found from an analysis based on differentiation of the power expression of EQ. 4,

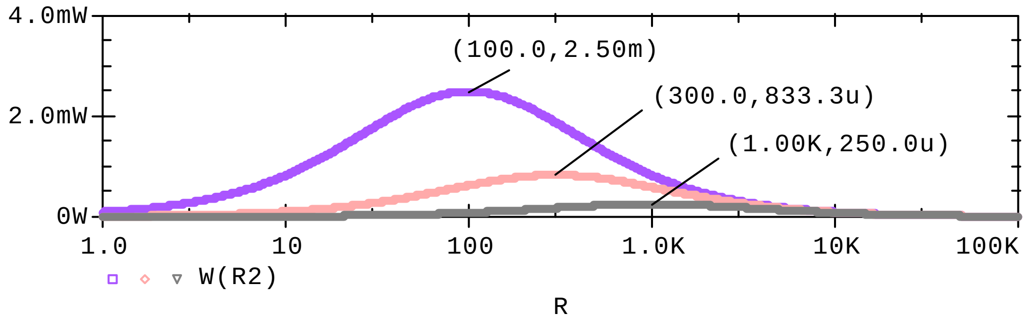


FIGURE 39

A DC SWEEP of R in the design showing that maximum power is transferred to the load when $R = R_S$; here R_S is fixed at 100 Ω, 300 Ω, 1 kΩ and R varies; frequency is held at f = 0 Hz (DC), an arbitrary value chosen below the corner frequency

VERIFICATION USING EXCEL WITH PSPICE

Figure 39 is only a spot check of the power transfer condition $R=R_S$. A more complete check would compare the maximum power transfer from PSPICE against the analytical value from EQ. 4 with $R=R_S$, namely

EQ. 6

$$W(R_S) = \frac{1}{2}\left(\frac{V_S}{2R_S}\right)^2 R_S = \frac{V_S^2}{8R_S} .$$

The PSPICE results for the power actually transferred[10] by the circuit are shown in Figure 40. Figure 40 shows the power transferred to the load for a 1V input voltage as a function of R_S for the circuit of Figure 23. When sweeping R_S, the formula in the PARAMETERS box of Figure 23 forces the condition $R=R_S$.[11]

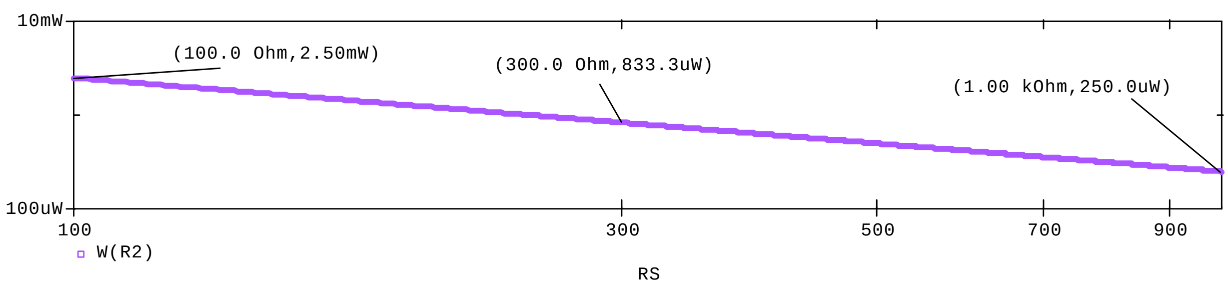


FIGURE 40

PSPICE power transferred for any value of R_S when the formulas of Figure 23 are used including, in particular, R_S = R; an AC performance analysis at f = 10 Hz was used; note that at the labeled points the power at 10Hz is the same as at DC from Figure 39

[10] That is, the power in the load R is calculated from the circuit, not from a formula.

[11] Because R_S is swept, not R, the PARAMETERS box enforces the specification $R=R_S$.

To compare EQ. 6 with PSPICE, we use EXCEL.[12] A spreadsheet is set up that employs EQ. 6 and a chart is generated. Then the PSPICE results are pasted into EXCEL for direct comparison on the same chart, as shown in Figure 41. [13] Examination of Figure 41 shows that PSPICE does not include the factor of 1/2 included in EQ. 6; that is, for power calculations PSPICE makes no r.m.s. correction for a sinusoidal input, but instead assumes the input amplitude V_{ac} = 1V already is r.m.s. corrected. If V_{ac} is really the amplitude, not the r.m.s. amplitude, PSPICE predicts double the actual power. Sometimes a check is useful, eh?

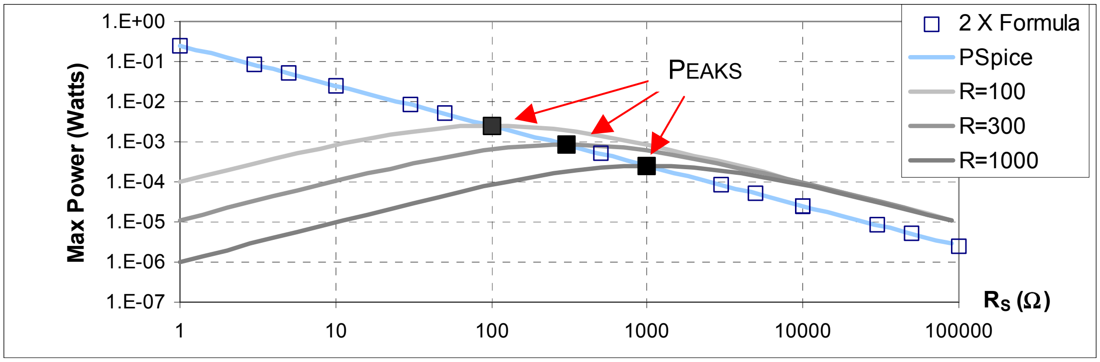


FIGURE 41

Comparison of maximum power to load from formula EQ. 6 with power actually transferred by the designed circuit according to PSPICE; PSPICE makes no r.m.s. corrections. The curves of Figure 39 are included; the peaks of these curves coincide with the maximum power transfer curve, as they should.

MAKING COMPARISONS BETWEEN FORMULAS AND PSpice USING Excel

We digress again to explain some useful features of EXCEL. First we describe making a plot of a formula in EXCEL. Then we explain how to paste PROBE data into EXCEL for comparison on the same chart. If you need more detail on EXCEL, please refer to the references at the end of the chapter.

NAMED VARIABLES IN Excel

Figure 42 shows a section of a spreadsheet used to plot the power formula of EQ. 6. The variables listed under the headings Constants, Specifications, and Parameters are named variables, that is, they are specified in EXCEL using the menu Insert/Name/Create as shown in Figure 43.

	A	B	C	D	E	F	G
6				Formula			
7	Constants			RS	R_	W	2*W
8	pi=	3.141593		1	1	1.25E-01	0.25
9				3	3	4.17E-02	0.083333
10	Specifications			5	5	2.50E-02	0.05
11	fC=	1000		10	10	1.25E-02	0.025
12				30	30	4.17E-03	0.008333
13	Parameters			50	50	2.50E-03	0.005
14	C_=	(1/(pi*fC*RS)		100	100	1.25E-03	0.0025
15				300	300	4.17E-04	0.000833
16				500	500	2.50E-04	0.0005
17				1000	1000	1.25E-04	0.00025
18				3000	3000	4.17E-05	8.33E-05
19				5000	5000	2.50E-05	0.00005
20				10000	10000	1.25E-05	0.000025
21				1.00E+04	10000	1.25E-05	0.000025
22				3.00E+04	30000	4.17E-06	8.33E-06
23				5.00E+04	50000	2.50E-06	0.000005
24				1.00E+05	100000	1.25E-06	2.5E-06

FIGURE 42

Section of spreadsheet for graphing EQ. 6; the W-column uses EQ. 4 with column R_

[12] We could use PROBE for this comparison, but for more complicated formulas EXCEL is easier.
[13] How to copy PROBE data into a spreadsheet and filter it for EXCEL charts is discussed later.

Highlighted cell to be labeled (right) and its label C_ (left)

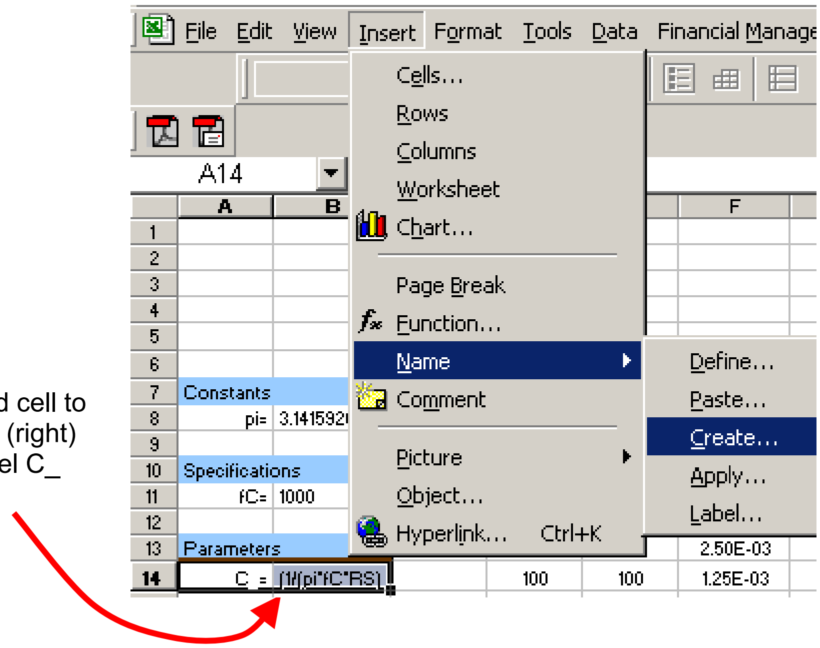
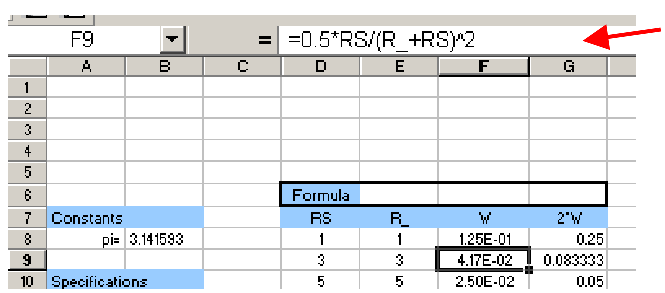


FIGURE 43

Naming the variable C_ in EXCEL using INSERT/NAME/CREATE

The procedure is to highlight the cell to be named and its label (C_= in this case) and then select INSERT/NAME/CREATE. In this example the variable is named C_ instead of C because EXCEL keeps "C" as a proprietary name. We use the same approach to name the columns RS, R_ and so forth. The entire columns and their column headings are highlighted in this case. The advantage of named variables is that all formulas look like algebra and so are easily checked. Using the named variables and clicking on a cell in the power column, for example, the algebraic formula appears in the formula box, as shown in Figure 44.



Formula Box

FIGURE 44

With named variables, the algebraic formula for the power W appears in the FORMULA BOX when a cell in the power column W is highlighted, which makes checking the formula easy

MAKING A PLOT IN Excel

To plot one column as y-axis as a function of another as x-axis, the x-axis column is placed to the left of the y-axis column and both columns are highlighted. First the x-axis column is highlighted and next the Ctrl key is held while the y-column is highlighted.[14] Once they are highlighted the CHART TYPE icon is selected on the toolbar[15] and the xy-chart option is selected as shown in Figure 45. The result is shown in Figure 46, and the data is almost invisible. To correct this problem, right click in an axis border to obtain the FORMAT AXIS option, as shown in Figure 47.

[14] Highlight only the numbers in the columns, not the column titles.
[15] Or, use CHART/CHART TYPE from the main menu.

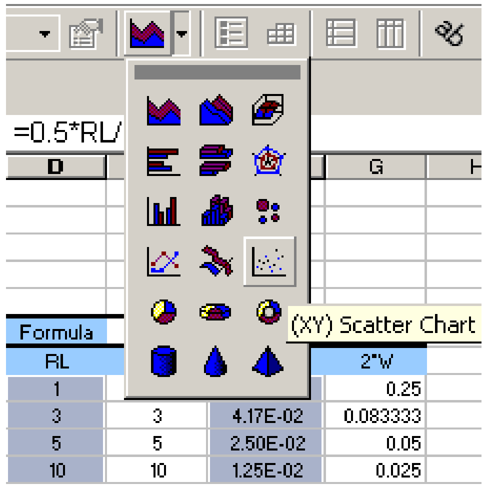


FIGURE 45

Selecting an xy-graph in EXCEL to plot W vs. R_S

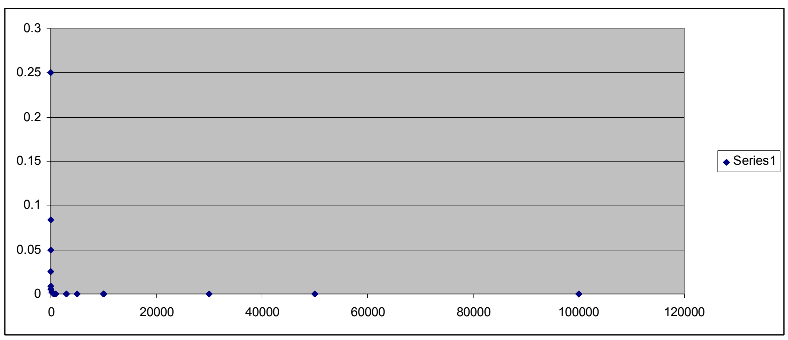


FIGURE 46

The initial plot; data points almost merge with the axes using these scales

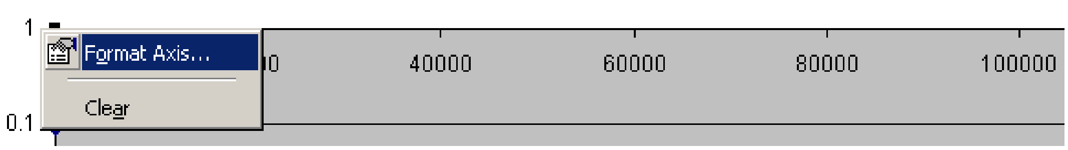


FIGURE 47

Obtaining the FORMAT AXIS menu where we can change the scale to logarithmic

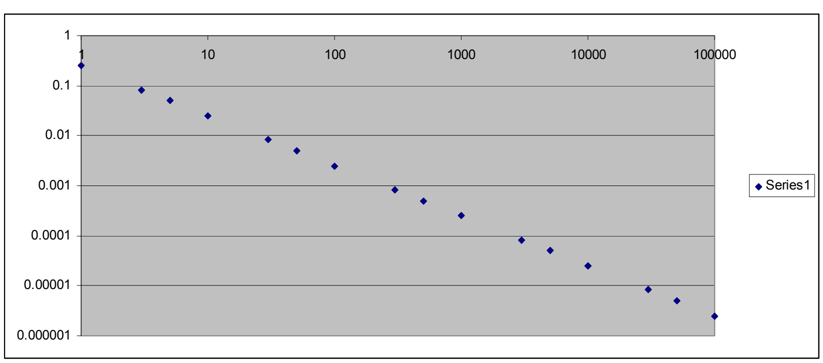


FIGURE 48

Initial chart when scales are made logarithmic

Then change the scale to logarithmic. Repeat for the other axis. The result is the chart of Figure 48.

The chart of Figure 48 needs some sprucing up. Much of this is a question of taste, but I will illustrate my prejudices. First, the background can be made white. Right click with the pointer in the background of the chart to obtain the FORMAT PLOT AREA option. Choosing this option leads to the menu of Figure 49. Select the white background.

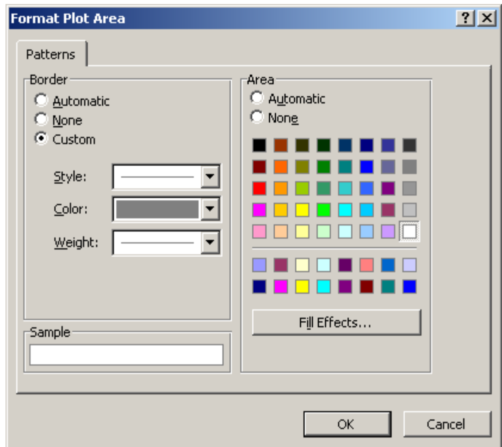


FIGURE 49

Selecting a white background using the FORMAT PLOT AREA menu

Next, put the pointer on the y-axis and right click to get the FORMAT AXIS option. This option leads to the FORMAT AXIS menu in Figure 50. Here the LOGARITHMIC SCALE option previously was set. To move the x-axis down to the bottom of the chart, set the x-axis to cross at a ridiculously small value by filling in 1E-10 in the VALUE (X) AXIS CROSSES AT box.

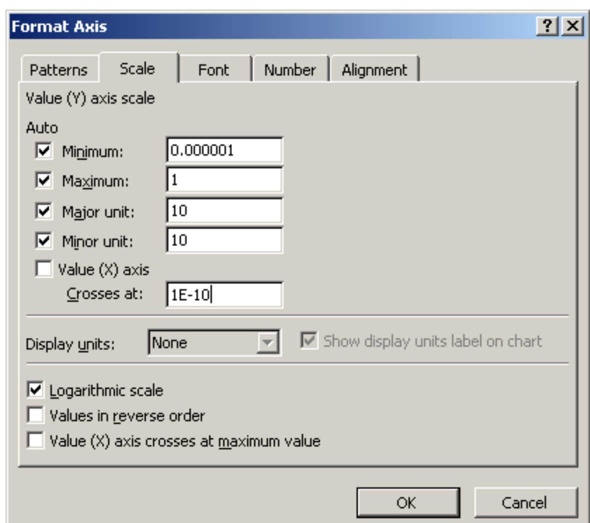


FIGURE 50

Selecting the x-axis options using the FORMAT AXIS menu

Now right-click on the curve on the chart to get the FORMAT DATA SERIES option shown in Figure 51.

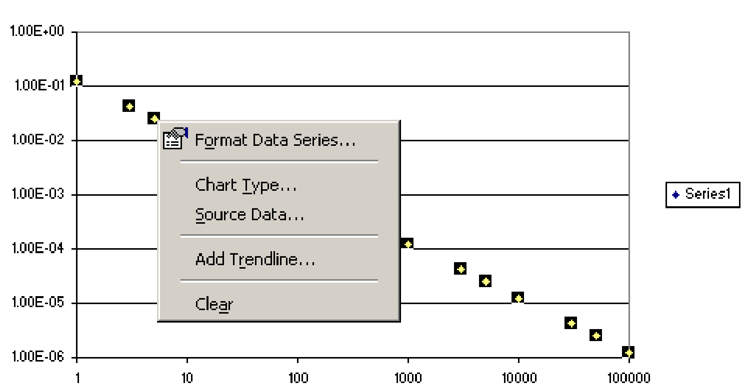


FIGURE 51

Obtaining the FORMAT DATA SERIES option

Select the options shown in Figure 52. I selected empty symbols because later I want to add the PSPICE data and these symbols show the comparison between the results more clearly than using two solid lines, for example.

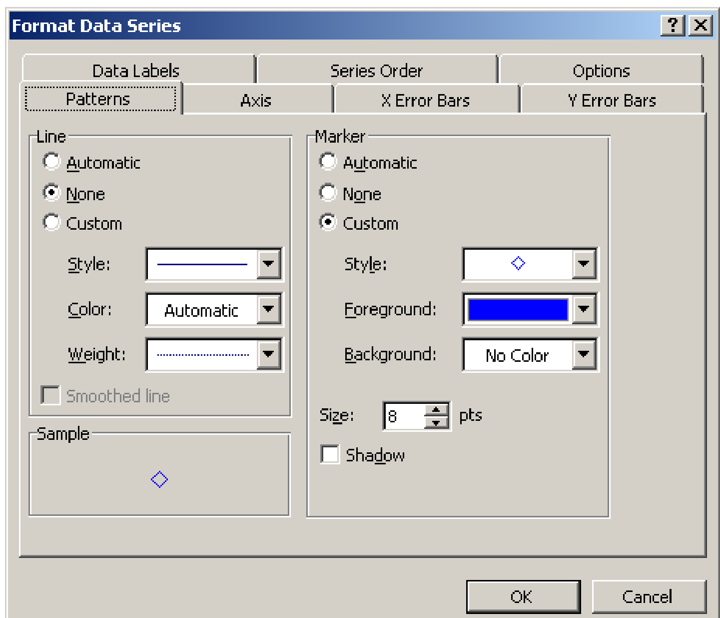


FIGURE 52

Formatting the plot using the FORMAT DATA SERIES menu

Right click with the pointer in the background of the chart to obtain the CHART OPTIONS option. Choosing this option leads to the menu of Figure 53. Choose the GRIDLINES tab. Check both major and minor gridlines.

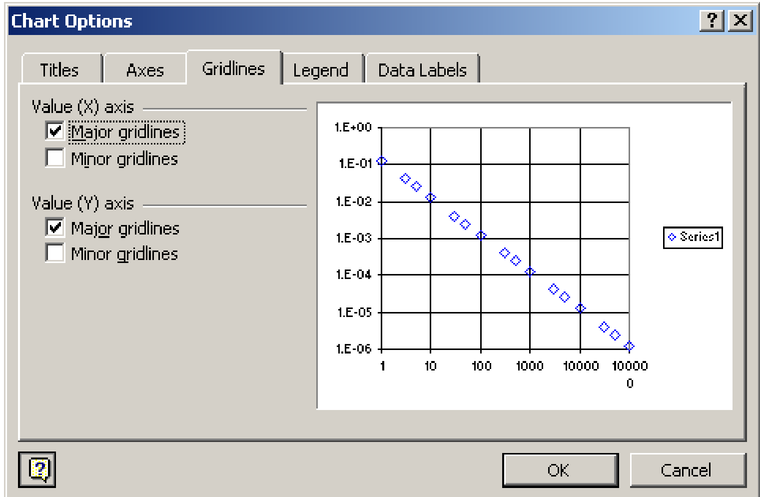


FIGURE 53

Selecting display of gridlines in both x- and y-directions

Once the gridlines are displayed on the chart, right click on them to obtain the FORMAT GRIDLINES option. Select a dashed line format. Then right click on the chart background again and select CHART OPTIONS to find the menu of Figure 54. Fill in the axes labels.

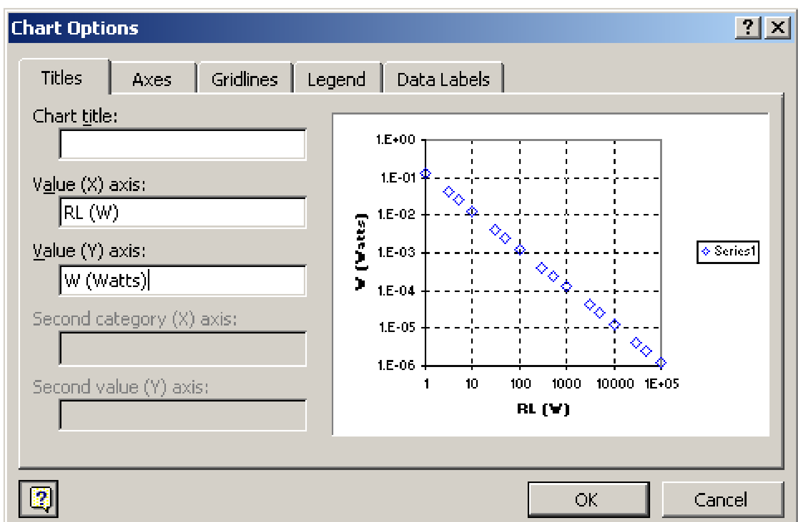


FIGURE 54

Labeling the axes using the menu CHART OPTIONS

On the chart, next highlight the W in the RS(W) label and right click to obtain the FORMAT AXIS TITLE menu. Change the font to SYMBOL as shown in Figure 55. This change makes W become Ω. Then highlight the "L" in the RL label and right click. The same menu allows making "L" a subscript.

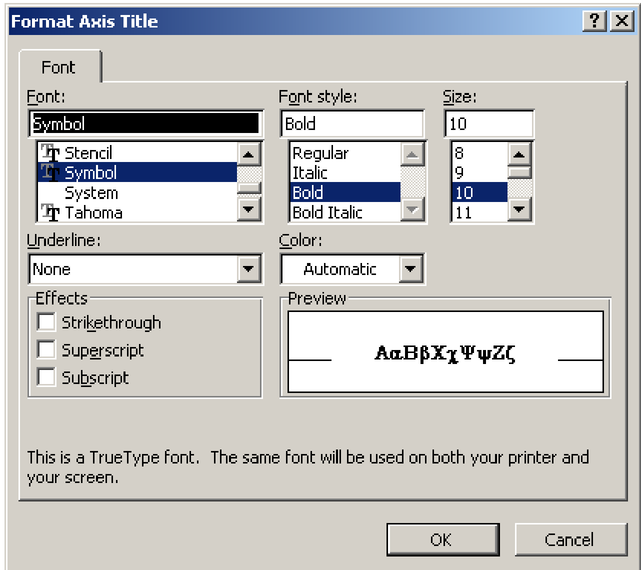


FIGURE 55

Changing W to Ω in the axis label

Then right click again on the background of the chart and select the SOURCE DATA menu shown in Figure 56.

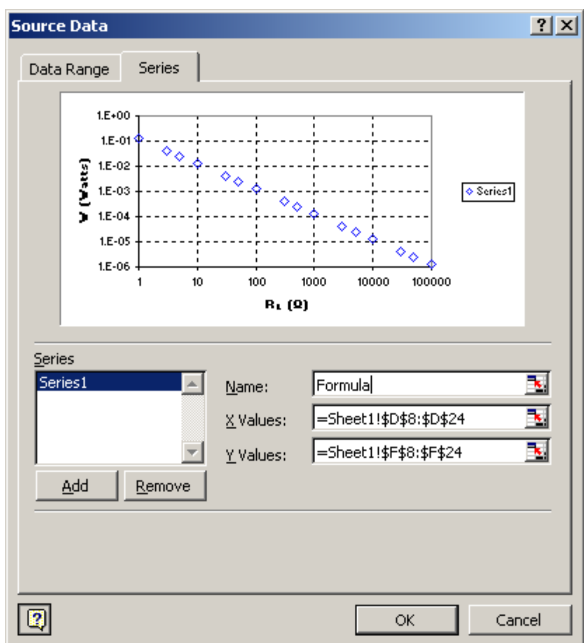


FIGURE 56

Naming the data set "Formula" using the SOURCE DATA menu

Setting the NAME box to "Formula" sets the name in the legend on the chart to "Formula". The chart now looks like Figure 57.

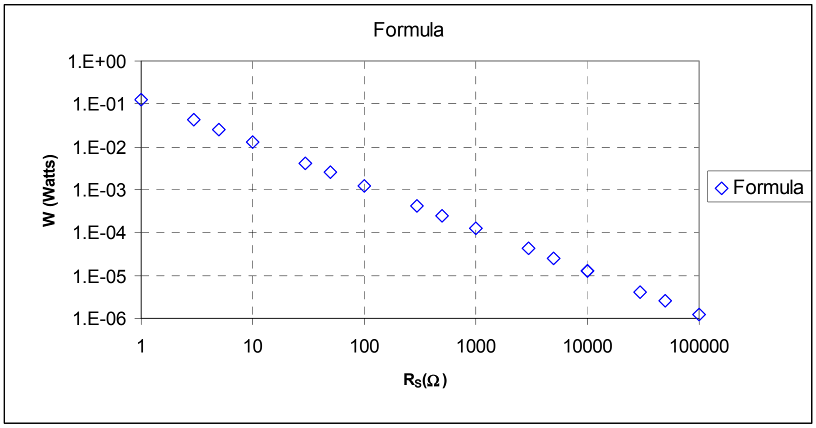


FIGURE 57
The chart after the above modifications

ADDING PSpice DATA TO AN Excel PLOT

Next, we add the PSPICE data. We run the PSPICE plot of Figure 40. In PROBE we click on the W(R2) label in the bottom border of the PROBE graph, and choose EDIT/COPY from the PROBE toolbar. Then we go to EXCEL and select a cell where we want to start pasting the data. Right clicking on this cell produces a PASTE option. Selecting PASTE puts the data into the spreadsheet. We go to the EXCEL chart, right click on the background, and select SOURCE DATA. Choose the ADD button on the SOURCE DATA menu. SERIES 2 appears. Type in the name of the new data series as PSPICE, as shown in Figure 58.

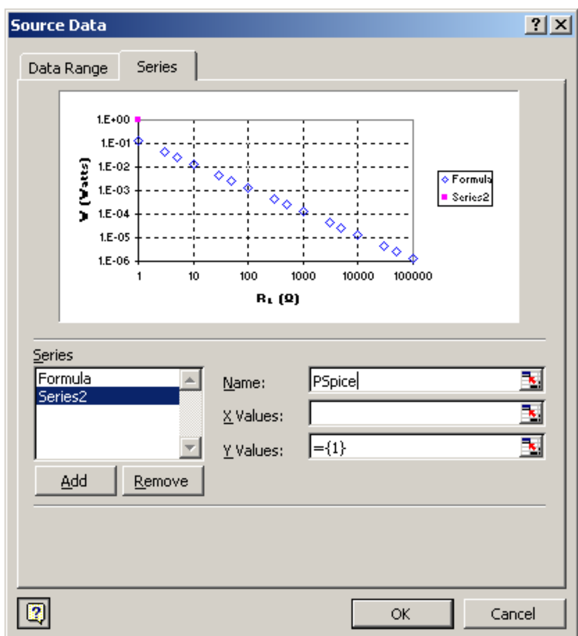


FIGURE 58
Adding the PSPICE data to the chart

Put the cursor in the X-values box and highlight the RS-data column from PSPICE. Then do the same for the Y-values box. The chart now looks like Figure 59.

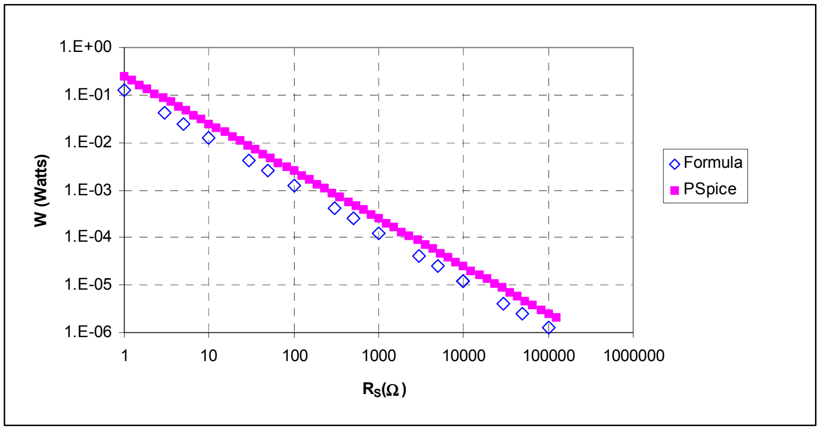


FIGURE 59

Chart with formula and PSPICE data plotted

It is apparent that the formula and PSPICE do not agree. The reason is that PSPICE does not include the factor of 1/2 needed for a sine wave input. We can check this point by looking at the DC case, for which PSPICE calculates the same power as it does at 10 Hz. That is, regardless of AC or DC, the power consumption as calculated by PSPICE is simply I^2R, and not the sine wave result $I^2R/2$. So, to compare against PSPICE we multiply our power by two and plot 2W instead of W. The result, after formatting the PSPICE results as a solid line, is Figure 41.

THE "FREEZE-DATA" FEATURE AND MULTIPLE PLOTS

Often we want to compare curves on an EXCEL chart for a several parameter values. We can plot the curve we want for one value, but if we change this value, the plot automatically updates. To compare plots for two values we can "freeze" the first curve, that is, unlink it from the spreadsheet. Here is a convenient way to do this.
As an example, we use the spreadsheet shown in Figure 60. It is the same as Figure 42, except the value of R is held fixed so we get EQ. 4 power transferred to R, instead of setting $R=R_S$ to find the maximum power of EQ. 6.

	A	B	C	D	E	F
6				Formula		
7	Constants			RS	W	2*W
8	pi=	3.1415926		1	4.99E-07	0.25
9				3	1.49E-06	0.083333
10	Specifications			5	2.48E-06	0.05
11	fC=	1000		10	4.90E-06	0.025
12				30	1.41E-05	0.008333
13	Parameters			50	2.27E-05	0.005
14	C_=	(1/(pi*fC*RS)		100	4.13E-05	0.0025
15	R_=	1000		300	8.88E-05	0.000833
16				500	1.11E-04	0.0005
17	Label	R_=1000		1000	1.25E-04	0.00025
18				3000	9.38E-05	8.33E-05
19				5000	6.94E-05	0.00005
20				10000	4.13E-05	0.000025
21				1.00E+04	4.13E-05	0.000025
22				3.00E+04	1.56E-05	8.33E-06
23				5.00E+04	9.61E-06	0.000005
24				1.00E+05	4.90E-06	2.5E-06

FIGURE 60

Section of spreadsheet for power transferred to resistor R; the W-column uses EQ. 4 but variable R is now set under the PARAMETERS listing in cell B15

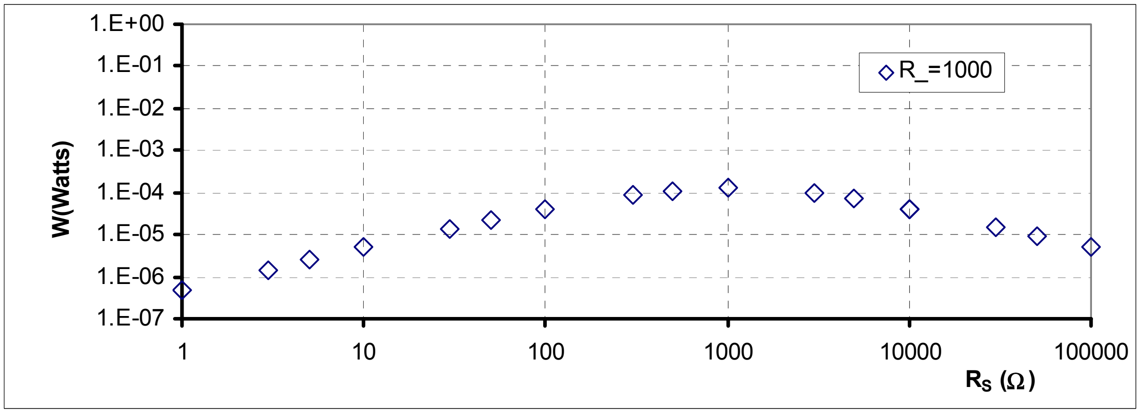


FIGURE 61

Plot of transferred power versus resistor value R_S for R=1 kΩ

Figure 61 shows a plot of the transferred power for one value of resistor R = 1 kΩ. We want to add a plot for a different value of R.

To do this, first create a label for the curve using the CONCATENATE function, as shown in Figure 62. A cell is selected to store the label (cell B17). Then the cell is highlighted and menu choice INSERT/FUNCTION is selected. The function CONCATENATE is picked, and the label R_= (cell A15) and the adjacent cell with the value for the resistor R are chosen for concatenation. Next, use INSERT/NAME/CREATE to name the cell "label".

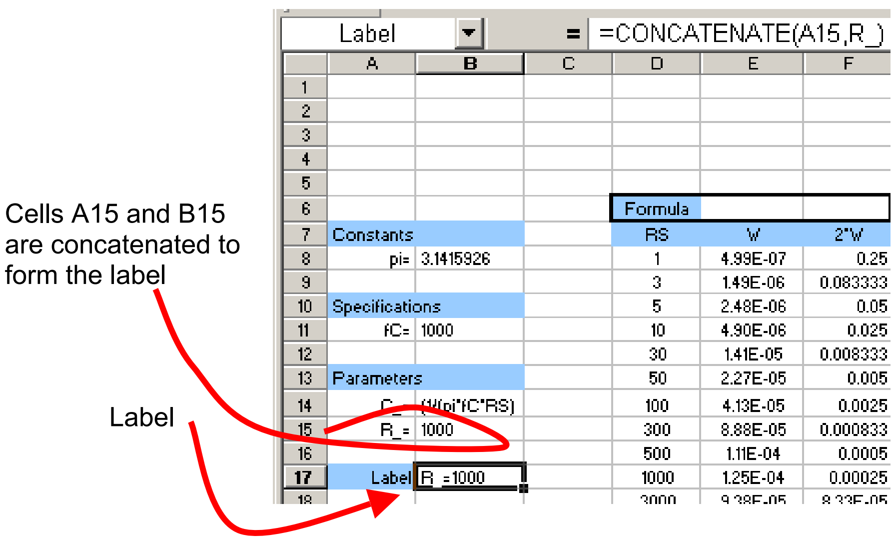


FIGURE 62

Creating a curve label using INSERT/FUNCTION and selecting CONCATENATE

Next, click on the background of the chart to get the SOURCE DATA menu. Insert the name of the series as shown in Figure 63 as =Sheet1!label.

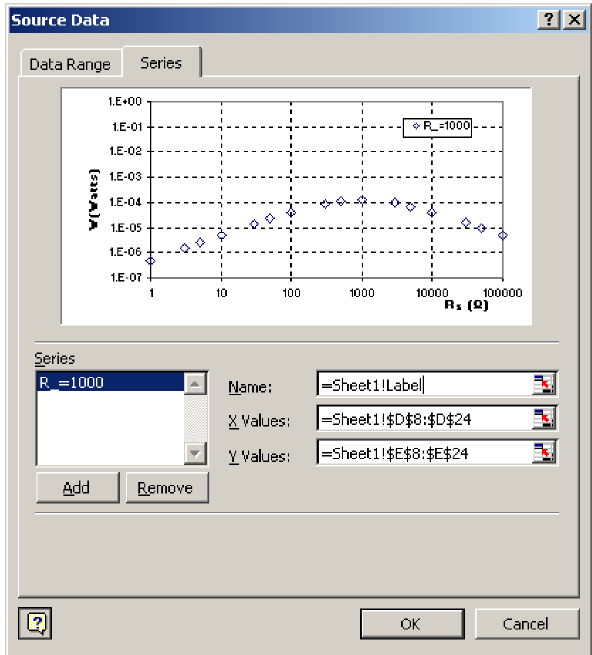


FIGURE 63

Inserting "label" as the Series name

Now we simply add a new series that simply duplicates this series. We click on the ADD button in Figure 63. The name Series 2 appears. We click on the R_=1000 label in the SERIES box. We highlight the entry in the NAME box and on the main EXCEL toolbar select EDIT/COPY. Then we click on the Series 2 entry in the SERIES box. Place the cursor in the NAME box and on the main EXCEL toolbar select EDIT/PASTE. Repeat this process for the X VALUES entry and the Y VALUES entry. When we are finished, the menu appears as in Figure 64.

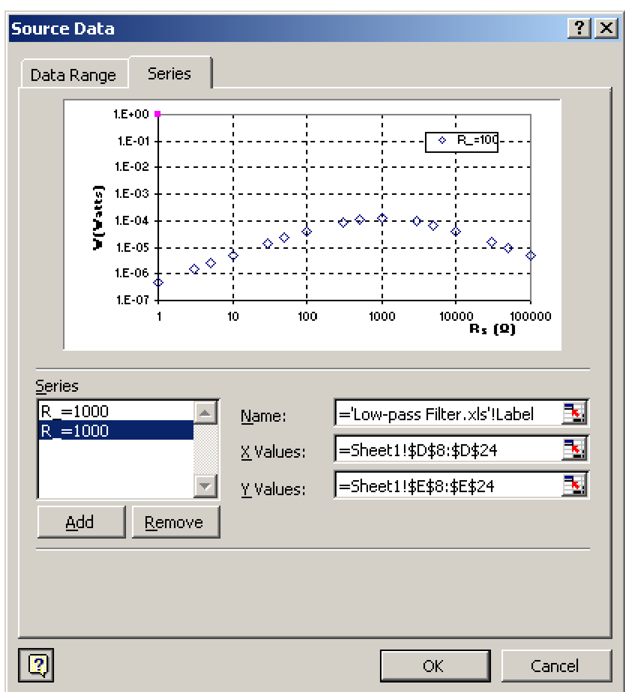


FIGURE 64

Menu with second identical series added

The chart appears as in Figure 65.

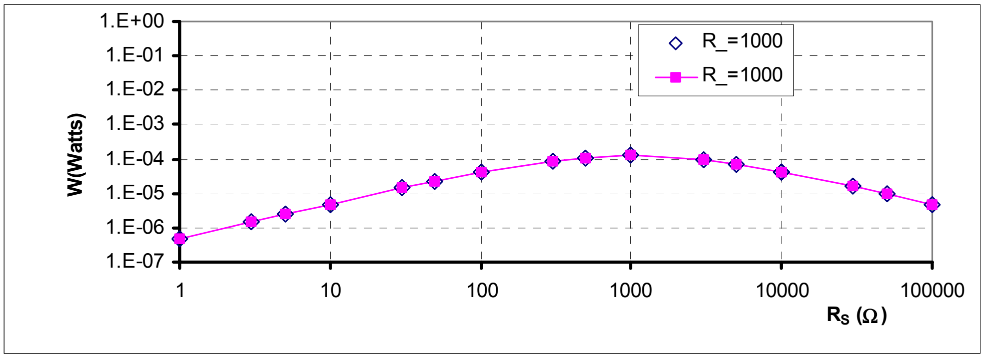


FIGURE 65

Chart with two identical series

We now "freeze" the data by right clicking on the plotted points, causing the FORMULA box to show the data labels as shown in Figure 66.

FIGURE 66

Formula box after right clicking on plotted points in chart

We now place the cursor in the FORMULA box and hit the keys Ctrl...+ simultaneously or use key F9. The result is the formula changes to a data listing. We select the check box as shown in Figure 67. This series is now retained in the chart as data points only, and is no longer coupled to the spreadsheet.

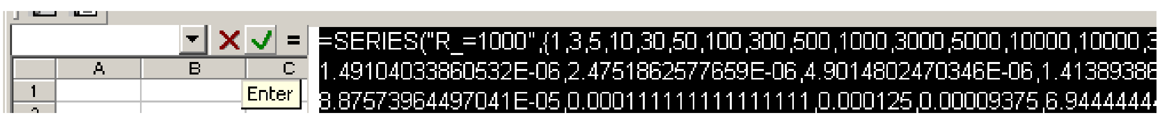


FIGURE 67

Selecting the check box after Ctrl...+

Now we can change the value of R_ on the spreadsheet. The data point curve stays put, but the other curve moves. Also, the label of the curve tracking the spreadsheet stays current. See Figure 68.

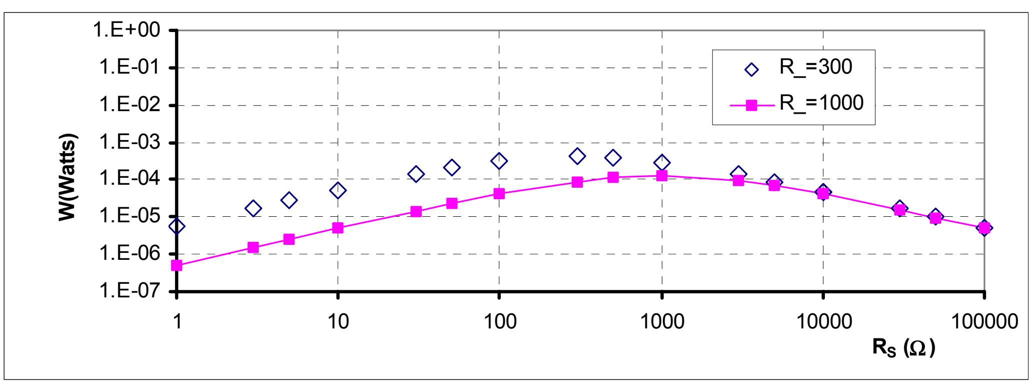


FIGURE 68

Changing R to R=300 on the spreadsheet, one curve is frozen, the other follows the spreadsheet

FILTERING PSpice DATA

Often a PSPICE run has a lot of data points, and we may not want to plot all of them on our chart – it takes a lot of memory. It also means that the "freeze data" feature will not work because there are too many points (only 1024 characters are allowed). Here is one way to select a subset of points. First we number the points using the index POINT, say. We name the POINT column. Then we define the number of points we will jump, say JUMP and name this variable. Then we set up two new columns for the filtered x- and y-values, say FILTERED RS and FILTERED W. The spreadsheet looks like Figure 69.

Jump variable

	A	B	C	D	E	F	G	H
4			Jump=	3				
5								
6				Formula			Filtered	Filtered
7	Constants		Point	RS	W	2*W	RS	W
8	pi=	3.1415926	1	1	5.52E-06	0.25	5	2.69E-05
9			2	3	1.63E-05	0.083333	50	0.000204
10	Specifications		3	5	2.69E-05	0.05	500	0.000391
11	fC=	1000	4	10	5.20E-05	0.025	5000	8.9E-05
12			5	30	1.38E-04	0.008333	30000	1.63E-05
13	Parameters		6	50	2.04E-04	0.005	0	0
14	C_=	(1/(pi*fC*RS))	7	100	3.13E-04	0.0025	0	0
15	R_=	300	8	300	4.17E-04	0.000833	0	0
16			9	500	3.91E-04	0.0005	0	0
17	Label	R_=300	10	1000	2.96E-04	0.00025	0	0
18			11	3000	1.38E-04	8.33E-05	0	0
19			12	5000	8.90E-05	0.00005	0	0
20			13	10000	4.71E-05	0.000025	0	0
21			14	1.00E+04	4.71E-05	0.000025	0	0
22			15	3.00E+04	1.63E-05	8.33E-06	0	0
23			16	5.00E+04	9.88E-06	0.000005	0	0
24			17	1.00E+05	4.97E-06	2.5E-06	0	0

FIGURE 69

The spreadsheet with the named JUMP variable, the named POINT number column, and the columns for the filtered data

To create the filtered data we use the OFFSET function as shown in Figure 70. The cell $D$7 in the OFFSET function refers to the start of the x-column, namely R_S in this example. The second argument tells the OFFSET function to report the contents of the cell a distance (Jump*Point) cells below this starting cell. The last argument tells the OFFSET function to report the contents of the cell in the same column as the starting cell (zero column offset).

G8 = =OFFSET($D$7,Jump*Point,0)

	A	B	C	D	E	F	G	H
1								
2								
3								
4			Jump=	3				
5								
6				Formula			Filtered	Filtered
7	Constants		Point	RS	W	2*W	RS	W
8	pi=	3.1415926	1	1	5.52E-06	0.25	5	2.69E-05

FIGURE 70

Using the OFFSET function to report only cells at intervals JUMP from each other

For the FILTERED W column instead of the starting cell $D$7 we use $E$7. The results of filtering can be seen in Figure 69. By plotting the filtered data, we can adjust variable JUMP to control just how many points to plot.[16]

---

[16] As a fine point, to avoid plotting points corresponding to (0, 0) like (G13, H13) in Figure 69, we can use an IF statement as the filtering function. For example, instead of the entry in Figure 70, we could use =IF (OFFSET($D$7,Jump*Point,0)=0,NA(),OFFSET($D$7,Jump*Point,0)). This change substitutes #N/A instead of zero, and #N/A will not plot. The advantage of this choice is that the plotted ranges can be made large and do not have to be hand tailored to exclude the filtered regions with zero entries.

Making a worksheet into a function

It may happen that we have put together a worksheet that computes a particular result as a function of many parameters. Then we decide we would like to see how the function changes as one or more of the parameters are changed. Of course, we can change the parameter and the spreadsheet updates, but we would like to see a chart to compare the value for a range of parameter changes. For example, let us take the spreadsheet of Figure 60. Let us suppose we want to construct a chart showing W as a function of R_ for R_S = 500 Ω, instead of W as a function of R_S, which is the way the worksheet was set up originally, where column E was plotted against column E. As shown in Figure 71, we create an x-column for R_ and a y-column for W. We do not name these columns. The first cell in the R_ column is ignored, and we begin in the second cell to list the R_ values. In the first cell of the new W-column, we insert =$E$16, which is the value of W corresponding to R_S = 500 Ω (highlighted in Figure 71). We then fill in this entry all through the W-column, and the $-signs make all these entries read =$E$16. Then in the named R_ cell, namely B15, we enter =H8, the location of the second cell in the new R_ column. Next we highlight the new R_ and W columns and select the menu DATA/TABLE. The set up is shown in Figure 72. The ROW INPUT CELL box is left blank because we are using R_ as the entry for a column, not a row. Hit OK.

	A	B	C	D	E	F	G	H	I
6				Formula				R_	W
7	Constants		Point	RS	W	2*W			9.96E-04
8	pi=	3.1415926	1	1	1.25E-01	0.25		1	9.96E-04
9			2	3	9.38E-02	0.083333		2	9.96E-04
10	Specifications		3	5	6.94E-02	0.05		3	9.96E-04
11	fC=	1000	4	10	4.13E-02	0.025		5	9.96E-04
12			5	30	1.56E-02	0.008333		8	9.96E-04
13	Parameters		6	50	9.61E-03	0.005		10	9.96E-04
14	C_=	(1/(pi*fC*RS))	7	100	4.90E-03	0.0025		20	9.96E-04
15	R_=	1	8	300	1.66E-03	0.000833		30	9.96E-04
16			9	500	9.96E-04	0.0005		50	9.96E-04
17	Label	R_=1	10	1000	4.99E-04	0.00025		80	9.96E-04
18	Label2	RS=500	11	3000	1.67E-04	8.33E-05		100	9.96E-04
19			12	5000	1.00E-04	0.00005		200	9.96E-04
20			13	10000	5.00E-05	0.000025		300	9.96E-04
21			14	1.00E+04	5.00E-05	0.000025		500	9.96E-04
22			15	3.00E+04	1.67E-05	8.33E-06		800	9.96E-04
23			16	5.00E+04	1.00E-05	0.000005		1000	9.96E-04
24			17	1.00E+05	5.00E-06	2.5E-06		2000	9.96E-04
25								3000	9.96E-04
26								5000	9.96E-04
27								8000	9.96E-04
28								10000	9.96E-04

FIGURE 71

Spreadsheet for set up as a function

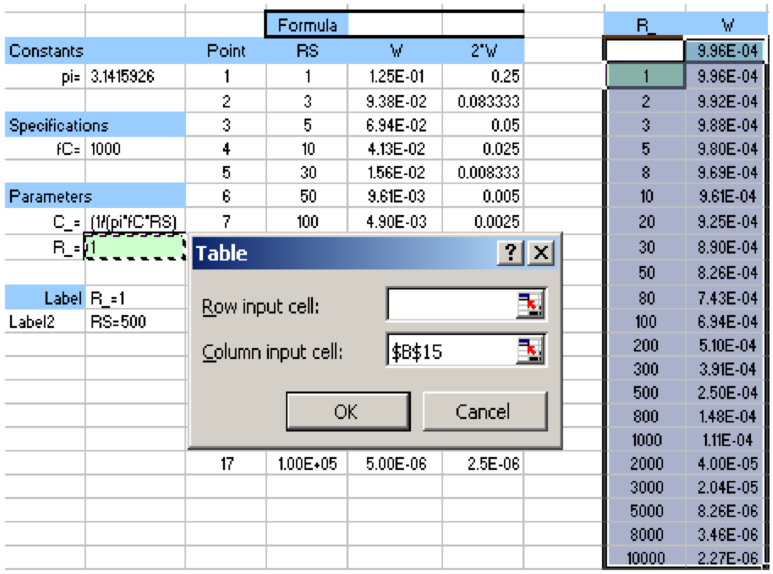


FIGURE 72

Inserting the data table with R_ and W columns highlighted

The table will now calculate and a chart can be made based on these columns, as shown in Figure 73.

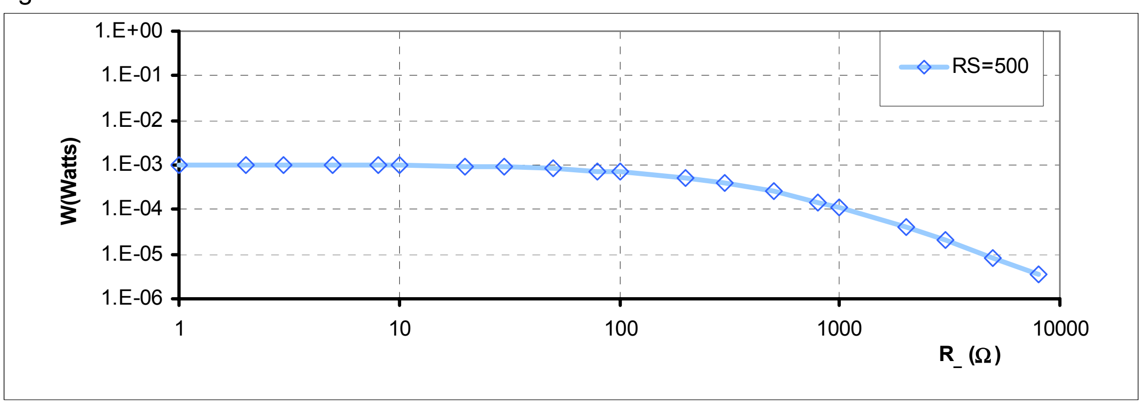


FIGURE 73

Chart based on Figure 72 shows power W as a function of R_ for fixed R_S = 500 Ω

SUMMARY OF USE OF PSpice WITH Excel

For EXCEL the use of named variables and xy-plots has been demonstrated. In particular, it was shown how to paste PROBE data into EXCEL and use an EXCEL chart to compare the formula of a hand analysis with PROBE output. Some conveniences in making multiple plots and filtering data also were discussed.

Adjusting PSpice Accuracy

PSpice is not intended to provide tremendous accuracy and, indeed, for practical purposes high accuracy makes little sense because the transistor models used are not exact, and even if they were, the transistor parameters are only estimates and vary from one nominally identical device to another because of manufacturing variations. However, for idealized modeling one does like to know whether errors are conceptual or just numerical.

Accuracy can be changed from the default values using the Capture menu PSpice/Edit Simulation Profile and selecting the Options tab, as shown in Figure 74.

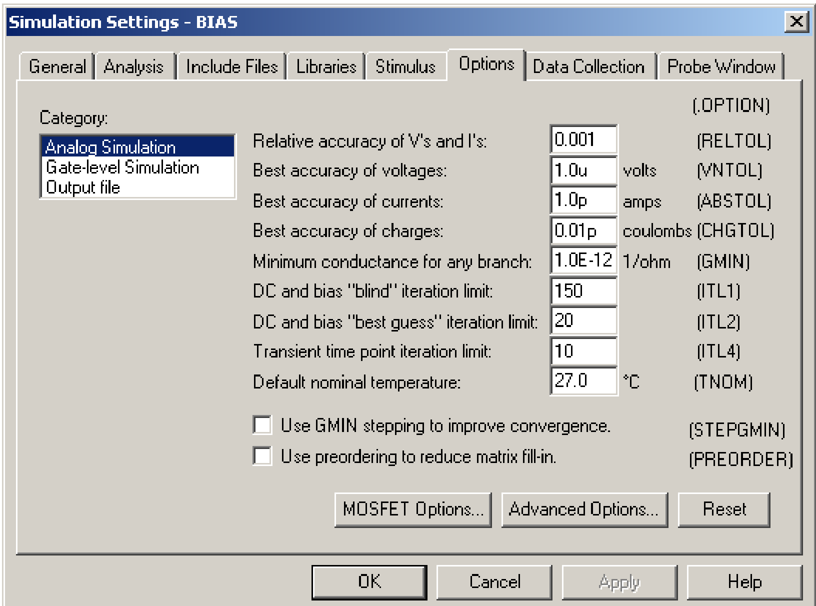


Figure 74

Setting accuracy levels in a simulation with the Options tab of Simulation Settings menu

We also can set the displayed number of figures on the schematic using the Capture menu PSpice/Bias Points/Preferences and changing the displayed precision, as shown in Figure 75. At most one can set this box to 10.

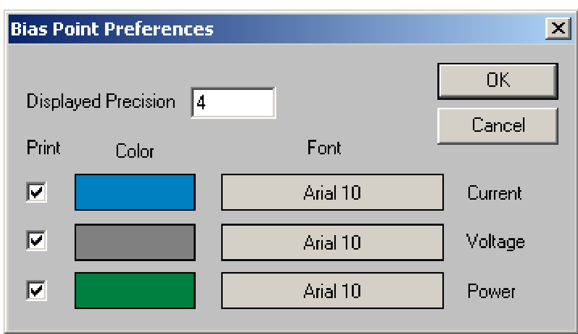


Figure 75

Changing the displayed precision on the schematic

Likewise, the number of figures displayed using the cursor in Probe can be adjusted using the Probe menu Tools/Options and setting the Number of Cursor Digits box, as shown in Figure 76.

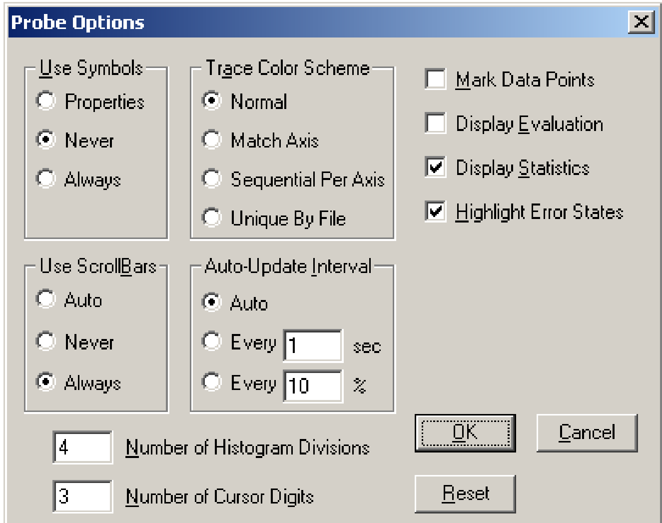


FIGURE 76

Setting the number of figures displayed using the cursor in PROBE

EXPLORING ACCURACY OF THE THERMAL VOLTAGE

With these tools we can explore the accuracy of PSPICE for the thermal voltage.[17] The fundamental definition of thermal voltage corresponding to a given temperature in degrees Celsius is based on the equations in Figure 77.

Thermal Voltage

kB = 1.3806503E-23_J/K

q = 1.602176462E-19_C

T_ice = 273.15

Celsius → T_C = 27

Kelvin → T = {T_ice+T_C}

Vth = {kB*T/q}

FIGURE 77

Fundamental definition of thermal voltage corresponding to 27° Celsius

A discussion of the Celsius scale and the role of the temperature of melting ice (T_ice) can be found at http://www.wikipedia.org/wiki/Physical_constants, and the values of the constants are found on the web site http://physics.nist.gov/cuu/Constants . We can compare the result of this calculation of V_{th} with PSPICE by comparing the built-in voltage of a diode from PSPICE with the diode formula

EQ. 7

$$V_D = V_{th} \ell n\,(I_D/I_S),$$

where I_D = diode current and I_S = diode scale current. The set up is shown in Figure 78.

---

[17] This example is selected because close comparison between ideal circuits and PSPICE sometimes is limited by inaccuracy in the value of V_{th}.

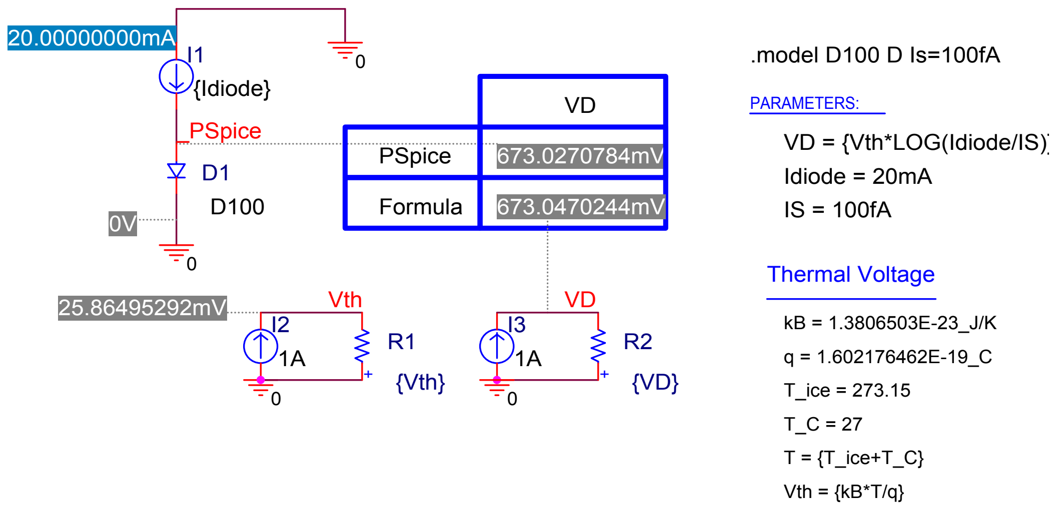


FIGURE 78

Set up to compare built-in diode voltage in PSPICE with formula; default accuracy tolerances were used; the diode obeys the simple diode law, as indicated by its dot-model statement

We also can run a range of current levels to see if the error changes with current level, as shown in Figure 79. It does, so no single correction of V_{th} will lead to zero error at all currents.

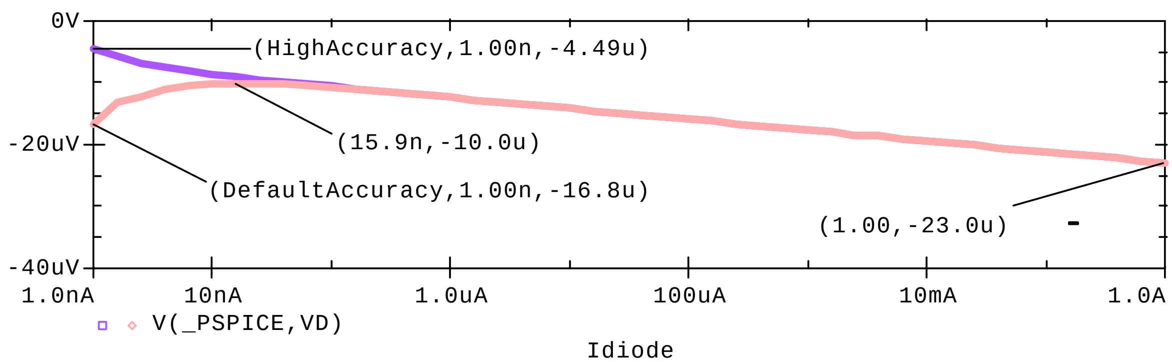


FIGURE 79

Difference between PSPICE value for diode voltage V_D and value based upon EQ. 7

Figure 79 compares the error in diode voltage using the default options in Figure 74 with that when the first five option boxes are all set at 1f.

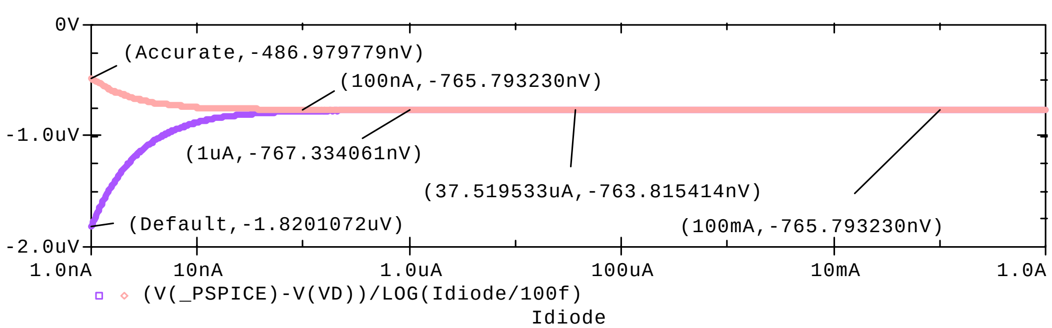


FIGURE 80

Correction to V_{th} as a function of diode current for currents between 1nA and 1A

Figure 80 shows the difference between PSPICE (voltage V(_PSPICE)) and the formula based upon the evaluator (voltage V(VD)) as a function of current level. Like Figure 79, Figure 80 also shows that the PSPICE value of V_{th} varies with current level, especially at lower current levels, where it becomes sensitive to the default settings. For higher current levels the correction to V_{th} is about 764–767nV, or an error of about 3/1000 %.

It is unclear where the error originates. To this author it appears likely that the error is not actually in the determination of V_{th}, although there may be a small current-independent component of the error from such a cause. It seems likely that the error is in the determination of V_{BE}. This determination may be based on a general numerical algorithm that works even for devices that do not obey the simple diode law, and so incurs a small error when applied to an ideal diode law example.

SUMMARY OF ACCURACY

Menus for adjusting the accuracy and its display in PSPICE were shown. Evaluator circuits were used to check the accuracy of thermal voltage V_{th} in PSPICE. Numerical accuracy checks on the solution methods themselves have not been discussed.

Overall summary

This appendix has introduced many features of CAPTURE, PSPICE, PROBE, and EXCEL that prove useful. It looks like a lot of overhead, but it is quickly mastered and becomes second nature, like using a calculator or riding a bike. If you can do the Exercises below, you have the software know-how to go through this book. If you cannot, please consult the References to fill in the gaps.

With these software skills, exploration of circuit behavior becomes much more interesting than trial-and-error iteration of PSPICE or dogged attempts at getting pages of algebra to yield their secrets.

References

M. E. Herniter, "Schematic Capture with Cadence PSPICE", Second Edition, Prentice Hall, 2003

William J. Orvis, "EXCEL for Scientists and Engineers", Second Edition, Sybex, 1996

Craig Stinson and Mark Dodge, "Microsoft EXCEL Version 2002 Inside Out", Microsoft Press, 2001

Exercises

1. Create Figure 24.
2. Create Figure 41.
3. Carry out the procedure used to generate Figure 22 and then do it again by putting both circuits in the same branch of the simulation hierarchy so they execute simultaneously.
4. Compare the PSPICE data of Figure 39 with the corresponding EXCEL plots in Figure 68 by pasting the PSPICE curves into the EXCEL chart.
5. Create Figure 79. To do this use the breakout diode part Dbreak and edit its dot-model statement by highlighting it on the schematic and right clicking to obtain the choice EDIT PSPICE MODEL. Paste the dot model statement of Figure 78 over the default dot model statement and SAVE.