PSpice Advanced Analysis User Guide

Product Version 17.2-2016
February 2016

Document Last Updated: June 2019

Contents

Before you begin 11

Welcome 11

How to use this guide 12

Symbols and conventions 12

Related documentation 13

Accessing online documentation 15

1 Introduction 17

In this chapter 17

Advanced Analysis overview 17

Project setup 18

Validating the initial project 19

Advanced Analysis files 20

Workflow 20

Numerical conventions 22

2 Libraries 25

In this chapter 25

Overview 25

Parameterized components 25

Location of Advanced Analysis libraries 29

Assigning Tolerance using the Assigning Tolerance window 30

Using Advanced Analysis libraries 35

Using the library tool tip 35

Using Parameterized Part icon 35

Preparing your design for Advanced Analysis 36

Creating new Advanced Analysis-ready designs 36

Updating Existing Designs for Advanced Analysis 39

- Using the design variables table . . . 45
Modifying existing designs for Advanced Analysis . . . 47
Example . . . 48
- Selecting a parameterized component . . . 48
- Setting a parameter value . . . 50
Using the design variables table . . . 52
For power users . . . 54
- Legacy PSpice optimizations . . . 54

3 Sensitivity . . . 55

In this chapter . . . 55
Sensitivity overview . . . 55
Sensitivity strategy . . . 56
- Plan ahead . . . 57
- Workflow . . . 58
Sensitivity procedure . . . 58
- Setting up the circuit in the schematic editor . . . 58
- Setting up Sensitivity in Advanced Analysis . . . 59
- Running Sensitivity . . . 61
- Controlling Sensitivity . . . 65
- Sending parameters to Optimizer . . . 67
- Sensitivity calculations . . . 81

4 Optimizer . . . 85

In this chapter . . . 85
Optimizer overview . . . 85
Terms you need to understand . . . 87
Optimizer procedure overview . . . 94
- Setting up in the circuit in the schematic editor . . . 96
- Setting up Optimizer in Advanced Analysis . . . 97
- Running Optimizer . . . 112
- Assigning available values with the Discrete engine . . . 117
- Finding components in your schematic editor . . . 118

Examining a Run in PSpice 118
Example 119
Optimizing a design using measurement specifications 119
Optimizing a design using curve-fit specifications 137
For Power Users 143
What are Discrete Tables? 143
Adding User-Defined Discrete Table 144
Device-Level Parameters 145
Optimizer log files 146
Engine Overview 146

5
Smoke 149

In this chapter 149
Smoke overview 149
Smoke strategy 150
Plan ahead 150
Workflow 151
Smoke procedure 151
Setting up the circuit in the schematic editor 151
Running Smoke 152
Configuring Smoke 154
Example 156
Overview 156
Setting up the circuit in the schematic editor 156
Running Smoke 158
Configuring Smoke 163
For power users 166
Smoke parameters 166
Adding Custom Derate file 175
Supported Device Categories 189
Secondary Breakdown 190

6 Monte Carlo . . . 195

In this chapter . . . 195
Monte Carlo overview . . . 195
Monte Carlo strategy . . . 196
    Plan Ahead . . . 196
    Workflow . . . 198
Monte Carlo procedure . . . 199
    Setting up the circuit in the schematic editor . . . 199
    Setting up Monte Carlo in Advanced Analysis . . . 200
    Running Monte Carlo . . . 201
    Reviewing Monte Carlo data . . . 202
    Controlling Monte Carlo . . . 206
    Printing results . . . 208
    Saving results . . . 208
Example . . . 209
    Setting up the circuit in the schematic editor . . . 209
    Setting up Monte Carlo in Advanced Analysis . . . 212
    Running Monte Carlo . . . 216
    Reviewing Monte Carlo data . . . 217
    Controlling Monte Carlo . . . 225
    Printing results . . . 229
    Saving results . . . 229

7 Parametric Plotter . . . 231

In this chapter . . . 231
Overview . . . 231
Launching Parametric Plotter . . . 232
Sweep Types . . . 233
    Adding sweep parameters . . . 235
Specifying measurements . . . 238
    Adding measurement expressions . . . 238
    Adding a trace . . . 239

Running Parametric Plotter 240
Viewing results 240
Results tab 240
Analyzing Results 241
Plot Information tab 242
Adding plot 242
Viewing the plot 244
Measurements Tab 244
Example 244

8
Measurement Expressions 257
In this chapter 257
Measurements overview 257
Measurement strategy 258
Procedure for creating measurement expressions 258
Setup 258
Composing a measurement expression 259
Viewing the results of measurement evaluations 260
Example 260
Viewing the results of measurement evaluations. 264
Measurement definitions included in PSpice 265
For power users 269
Creating custom measurement definitions 269
Definition example 271
Measurement definition syntax 273
Syntax example 282

9
Optimization Engines 285
In this chapter 285
Modified LSQ engine 286
Configuring the Modified LSQ engine 286
Random engine 291
Configuring the Random Engine 292

Discrete engine 294
Commercially available values 295

10
Troubleshooting 297
In this chapter 297
Troubleshooting feature overview 297
Strategy 297
Workflow 298
Procedure 298
Example 300
Strategy 300
Setting up the example 300
Using the troubleshooting function 303
Analyzing the trace data 306
Resolving the optimization 308
Common problems and solutions 311

A
Property Files 321
Template property file 323
The model_info section 325
The model_params section 326
The smoke section 327
The device property file 333
The device_info section 334
Optional sections in a device property file 337

B
The Special Library 339
VECTOR1 340
VPRINT1 343
VPRINT2 344
IPRINT1 345

PRINT1 345
WATCH 347
IC1 347
IC2 347
NODESET1 347
NODESET2 348
Additional Symbols 348
Glossary 349

Index 359

Before you begin

Welcome

Advanced Analysis allows PSpice[1] and PSpice A/D users to optimize performance and improve quality of designs before committing them to hardware. Advanced Analysis' four important capabilities: sensitivity analysis, optimization, yield analysis (Monte Carlo), and stress analysis (Smoke) address design complexity as well as price, performance, and quality requirements of circuit design.

Advanced Analysis is integrated with Cadence's design entry tools[2], OrCAD Capture and Design Entry HDL. Refer to the Product Installation Guide for Windows for information on hardware and software requirements.

1. Depending on the license available, you will access either PSpice or PSpice Simulator.
2. In this guide, design entry tool is used for both OrCAD Capture and Design Entry HDL. Any differences between the two tools is mentioned, if necessary.

How to use this guide

This guide is designed to make the most of the advantages of onscreen books. The table of contents, index, and cross references provide instant links to the information you need. Just click on the text and jump.

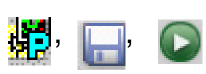
Each chapter about an Advanced Analysis tool is self-contained. The chapters are organized into these sections:

- Overview: introduces you to the tool
- Strategy: gives you tips on planning your project
- Procedure: lists each step you need to successfully apply the tool
- Example: lists the same steps with an illustrating example
- For power users: provides background information

If you find printed paper helpful, print only the section you need at the time. When you want an in-depth tutorial, print the example. When you want a quick reminder of a procedure, print the procedure.

Symbols and conventions

Our documentation uses a few special symbols and conventions.

Notation	Examples	Description
Bold text	**Import Measurements**, **Modified LSQ**, **PDF Graph**	Indicates that text is a menu or button command, dialog box option, column or graph label, or drop-down list option
Icon graphic		Shows the toolbar icon that should be clicked with your mouse button to accomplish a task
Lowercase file extensions	.aap, .sim, .drt	Indicates a file name extension

Related documentation

In addition to this guide, you can find technical product information in the embedded AutoHelp, in related online documentation, and on our technical website. The table below describes the type of technical documentation provided with Advanced Analysis.

This documentation component . . .	Provides this . . .
This guide— PSpice Advanced Analysis User Guide	A comprehensive guide for understanding and using the features available in Advanced Analysis.

This documentation component . . .	Provides this . . .
Help system (automatic and manual)	Provides comprehensive information for understanding the features in Advanced Analysis and using them to perform specific analyses. Advanced Analysis provides help in two ways: automatically (AutoHelp) and manually. AutoHelp is embedded in its own window and automatically displays help topics that are associated with your current activity as you move about and work within the Advanced Analysis workspace and interface. It provides immediate access to information that is relative to your current task, but lacks the complete set of navigational tools for accessing other topics. The manual method lets you open the help system in a separate browser window and gives you full navigational access to all topics and resources outside of the help system. Using either method, help topics include: ■ Explanations and instructions for common tasks ■ Descriptions of menu commands, dialog boxes, tools on the toolbar and tool palettes, and the status bar ■ Glossary terms ■ Reference information ■ Product support information
PSpice User Guide	An online, searchable user's guide
PSpice Reference Guide	An online, searchable reference manual for the PSpice simulation software products
PSpice Quick Reference	Concise descriptions of the commands, shortcuts, and tools available in PSpice
OrCAD Capture User Guide	An online, searchable user's guide

This documentation component . . .	Provides this . . .
OrCAD Capture Quick Reference Card	Concise descriptions of the commands, shortcuts, and tools available in Capture
Design Entry HDL User Guide	An online, searchable user's guide

Accessing online documentation

To access online documentation, you must open the Cadence Documentation window.

1. Do one of the following:

   a. From the Windows Start menu, choose the installed programs folder and then *Cadence Help*.

   b. From the *Help* menu in PSpice or PSpice Simulator, choose *Documentation*.

   The Cadence Documentation window appears.

2. Click the PSpice or PSpice Simulator category to show the documents in the category.

1

Introduction

In this chapter

- Advanced Analysis overview on page 17
- Project setup on page 18
- Advanced Analysis files on page 20
- Workflow on page 20
- Numerical conventions on page 22

Advanced Analysis overview

Advanced Analysis is an add-on program for PSpice[1] and PSpice A/D. Use these four Advanced Analysis tools to improve circuit performance, reliability, and yield:

- Sensitivity identifies which components have parameters critical to the measurement goals of your circuit design.
- The four Optimizer engines optimize the parameters of key circuit components to meet your performance goals.
- Smoke warns of component stress due to power dissipation, increase in junction temperature, secondary breakdowns, or violations of voltage / current limits.
- Monte Carlo estimates statistical circuit behavior and yield.

1. Depending on the license available, you will access either PSpice or PSpice Simulator.

Project setup

Before you begin an Advanced Analysis project, you need:

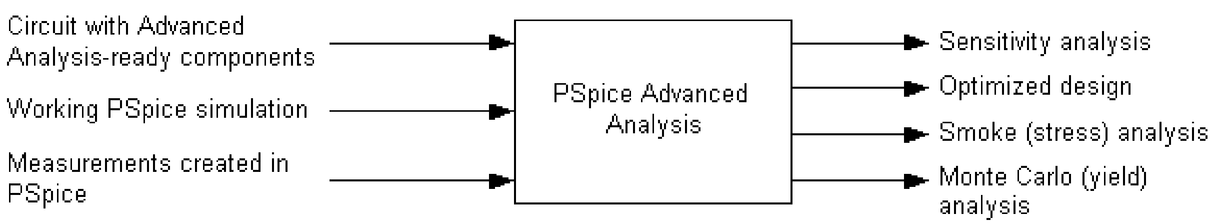
- Circuit components that are Advanced Analysis-ready

  Only those components that you want tested in Advanced Analysis have to be Advanced Analysis-ready. See Chapter 2, “Libraries.”

  Note: You can adapt passive RLC components for Advanced Analysis without choosing them from parameterized libraries. See Chapter 2, “Libraries.”

- A circuit drawn in Cadence’s design entry tools[1] and successfully simulated in PSpice.

- PSpice measurements that check circuit behavior critical to your design.



Creating measurement expressions

Sensitivity, Optimizer, and Monte Carlo require measurement expressions as input. You should create these measurements expressions in PSpice so you can test the results.

You can also create measurement expressions in Sensitivity, Optimizer, or Monte Carlo which can be exported to each other, but these measurements cannot be exported to PSpice for testing.

1. In this guide, design entry tool is used for both OrCAD Capture and Design Entry HDL. Any differences between the two tools is mentioned, if necessary.

Validating the initial project

Before you use Advanced Analysis:

1. Make your circuit components Advanced-Analysis ready for the components you want to analyze.

   See Chapter 2, "Libraries" for more information.

2. Set up a PSpice simulation.

   The Advanced Analysis tools use the following simulations:

This tool...	Works on these PSpice simulations...
Sensitivity	Time Domain (transient)
	DC Sweep
	AC Sweep/Noise
Optimizer	Time Domain (transient)
	DC Sweep
	AC Sweep/Noise
Smoke	Time Domain (transient)
Monte Carlo	Time Domain (transient)
	DC Sweep
	AC Sweep/Noise

3. Simulate the circuit and make sure the results and waveforms are what you expect.

4. Define measurements in PSpice to check the circuit behaviors that are critical for your design. Make sure the measurement results are what you expect.

Note: For information on setting up circuits, see your schematic editor user guide, Project setup on page 18, and Chapter 2, "Libraries."

For information on setting up simulations, see your *PSpice User Guide*.

For information on setting up measurements, see “Procedure for creating measurement expressions” on page 258.

Advanced Analysis files

The principal files used by Advanced Analysis are:

- PSpice simulation profiles (.sim)
- Advanced Analysis profiles (.aap)

Advanced users may also use these files:

- Device property files (.prp)

  For more information, see Appendix A, Property Files.

- Custom derating files for Smoke (.drt)

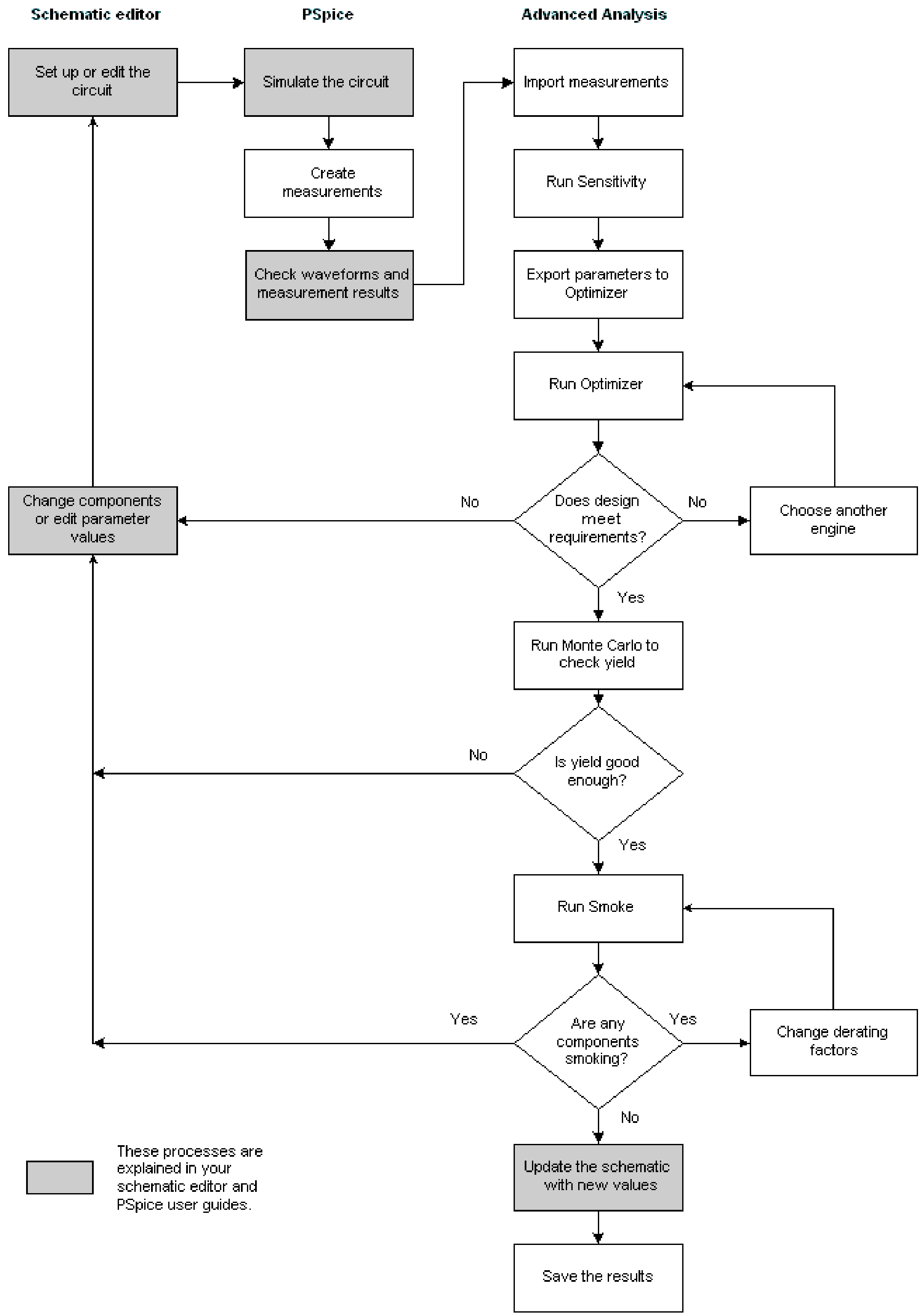
  For more information, see Adding Custom Derate file.

- Discrete value tables for Optimizer (.table)

  For more information, see “What are Discrete Tables?” on page 143.

Workflow

There are many ways to use Advanced Analysis. This workflow shows one way to use all four features.


Schematic editor
PSpice
Advanced Analysis
Set up or edit the circuit
Simulate the circuit
Import measurements
Create measurements
Run Sensitivity
Check waveforms and measurement results
Export parameters to Optimizer
Run Optimizer
Change components or edit parameter values
No
Does design meet requirements?
No
Choose another engine
Yes
Run Monte Carlo to check yield
No
Is yield good enough?
Yes
Run Smoke
Yes
Are any components smoking?
Yes
Change derating factors
No
These processes are explained in your schematic editor and PSpice user guides.
Update the schematic with new values
Save the results

Note: Temperature Sweep is not supported in the Advance Analysis Flow.

Numerical conventions

PSpice ignores units such as Hz, dB, Farads, Ohms, Henrys, volts, and amperes. It adds the units automatically, depending on the context.

Name	Numerical value	User types in:	Or:	Example Uses
femto-	10^{-15}	F, f	1e-15	2f
				2F
				2e-15
pico-	10^{-12}	P, p	1e-12	40p
				40P
				40e-12
nano-	10^{-9}	N, n	1e-9	70n
				70N
				70e-9
micro-	10^{-6}	U, u	1e-6	20u
	.000001			20U
				20e-6
milli-	$\mathbf{10^{-3}}$	**M, m**	**1e-3**	**30m**
	.001			**30M**
				30e-3
				.03

Name	Numerical value	User types in:	Or:	Example Uses
kilo-	10^3	K, k	1e+3	2k
	1000			2K
				2e3
				2e+3
				2000
mega-	**10^6**	**MEG, meg**	**1e+6**	**20meg**
	1,000,000			**20MEG**
				20e6
				20e+6
				20000000
giga-	10^9	G, g	1e+9	25g
				25G
				25e9
				25e+9
tera-	10^{12}	T, t	1e+12	30t
				30T
				30e12
				30e+12

2

Libraries

In this chapter

- Overview on page 25
- Using Advanced Analysis libraries on page 35
- Preparing your design for Advanced Analysis on page 36
- Example on page 48
- For power users on page 54

Overview

PSpice[1] ships with over 30 Advanced Analysis libraries containing over 4,300 components. Separate library lists are provided for Advanced Analysis libraries and standard PSpice libraries. See Using the library tool tip on page 35 for details.

The Advanced Analysis libraries contain parameterized and standard components. The majority of the components are parameterized. Standard components in the Advanced Analysis libraries are similar to components in the standard PSpice libraries and will not be discussed further in this document.

Parameterized components

A parameter is a physical characteristic of a component that controls behavior for the component model. A parameter value is either a number or a variable. When the parameter value is a variable, you

1. Depending on the license available, you will access either PSpice or PSpice Simulator.

have the option to vary its numerical solution within a mathematical expression and use it in optimization.

Note: A parameter is called a **property** in Capture and **attribute** in Design Entry HDL.

Design EntryWhen the parameter value is a variable, you have the option to vary its numerical solution within a mathematical expression and use it in optimization.In the Advanced Analysis libraries, components may contain one or more of the following parameters:

- Tolerance parameters

  For example, for a resistor the positive tolerance could be POSTOL = 10%.

- Distribution parameters

  For example, for a resistor the distribution function used in Monte Carlo analysis could be DIST = FLAT.

- Optimizable parameters

  For example, for an opamp the gain bandwidth could be GBW = 10 MHz.

- Smoke parameters

  For example, for a resistor the power maximum operating condition could be POWER = 0.25 W.

To analyze a circuit component with an Advanced Analysis tool, make sure the component contains the following parameters:

This Advanced Analysis tool...	Uses these component parameters...
Sensitivity	Tolerance parameters
Optimizer	Optimizable parameters
Smoke	Smoke parameters
Monte Carlo	Tolerance parameters,
	Distribution parameters (default parameter value is Flat / Uniform)

Tolerance parameters

Tolerance parameters define the positive and negative deviation from a component's nominal value. In order to include a circuit component in a Sensitivity or Monte Carlo analysis, the component must have tolerances for the parameters specified. Use the *Advanced Analysis library list* to identify components with parameter tolerances.

In Advanced Analysis, tolerance information includes:

- Positive tolerance

  For example, POSTOL for RLC is the amount a value can vary in the plus direction.

- Negative tolerance

  For example, NEGTOL for RLC is the amount a value can vary in the negative direction.

Tolerance values can be entered as percents or absolute numbers.

In Capture, you can assign positive or negative tolerance using the Assign Tolerance window to use parametrized components in Advanced Analysis. For more information on how to assign tolerance using Assign Tolerance window, see the Assigning Tolerance using the Assigning Tolerance window section.

Distribution parameters

Distribution parameters define types of distribution functions. Monte Carlo uses these distribution functions to randomly select tolerance values within a range.

For example, in a design entry tool property editor, a resistor could provide the following information:

Property	Value
DIST	FLAT

The distribution files are located at `<installation>\tools\pspice\library\distribution` by default. You can specify a different path by defining `DISTRIBUTION_DIRNAME` in the `PSPICE_ADVANCED ANALYSIS` section of the `pspice.ini` file. This path is taken by the tool as relative to `LIBPATH`. For example, if you define `DISTRIBUTION_DIRNAME=new_dir/file` and `LIBPATH=<install>\tools\PSpice\Library`, tool will search for distribution files from the following location:

`<install>\tools\PSpice\Library\new_dir\file`.

If `DISTRIBUTION_DIRNAME` is not defined, the default path will be assumed.

Optimizable parameters

Optimizable parameters are any characteristics of a model that you can vary during simulations. In order to include a circuit component in an Optimizer analysis, the component must have optimizable parameters. Use the *Advanced Analysis library list* to identify components with optimizable parameters.

For example, in Capture's property editor or Design Entry HDL's attribute editor, an opamp could provide the following gain bandwidth:

Property	Value
GBW	1e7

Note that the parameter is available for optimization only if you add it as a property on the schematic instance and assign it a value.

During Optimization, the GBW can be varied between any user-defined limits to achieve the desired specification.

Smoke parameters

Smoke parameters are maximum operating conditions for the component. To perform a Smoke analysis on a component, define the smoke parameters for that component. You can still use non-smoke-defined components in your design, but the smoke test ignores these components. Use the online *Advanced Analysis library list* to identify components with smoke parameters.

Most of the analog components in the standard PSpice libraries also contain smoke parameters.

See also Smoke parameters on page 166.

For example, in Capture's property editor or Design Entry HDL's attribute editor, a resistor could provide the following smoke parameter information:

Property	**Value**
POWER	RMAX
MAX_TEMP	RTMAX

Use the design variables table to set the values of RMAX and RTMAX to 0.25 Watts and 200 degrees Centigrade, respectively.

See Using the design variables table on page 45.

Location of Advanced Analysis libraries

The program installs the Advanced Analysis libraries to the following locations:

Capture symbol libraries

<Target_directory>\Capture\Library\PSpice\AdvAnls\

Design Entry HDL component libraries

<Target_directory> \ PSpice \ Library

PSpice Advanced Analysis model libraries

<Target_directory> \ PSpice \ Library

Assigning Tolerance using the Assigning Tolerance window

Assigning tolerance using the Assign Tolerance window in Capture allows you to use standard components or behavioral devices for Advanced Analysis. The global parameters can be assigned tolerance using the Assign Tolerance window, which allows us to use standard components or behavioral devices for Advanced Analysis.

The Assign Tolerance window in Capture can be launched from *PSpice – Advanced Analysis – Assign Tolerance*.

Note: Once tolerance is applied to the design using the Assign Tolerance window, the design will not be compatible with releases prior to SPB17.20.011.

You can assign tolerance to following devices using the Assign Tolerance window in Capture:

- Resistor (R),
- Inductor (L), and
- Capacitor (C)
- Voltage (V)
- Current (I)
- Insulated-Gate Bipolar Transistor (IGBT)
- Junction gate Field-Effect Transistor (JFET)
- Transistor (Q)
- Metal–Oxide–Semiconductor Field-Effect Transistor (MOSFET)
- Coupling Constant (K)
- Diode (D)
- Switch (S)

- Gallium Arsenide Field-effect Transistor (GaAsFET)

The following cases will let you understand how easily you can add tolerance to different types of parameters or devices in Capture using the Assign Tolerance window for sensitivity and monte carlo analysis:

- Case 1: Assigning Global Tolerance to Discrete Devices
- Case 2: Assigning Instance-level Tolerance to Discrete Devices
- Case 3: Assigning Tolerance to a Model Parameter
- Case 4: Assigning Tolerance to Sub-circuit Model Parameters
- Case 5: Assign Tolerance to Global Parameters

Case 1: Assigning Global Tolerance to Discrete Devices

To assign tolerance to discrete devices using the Assign Tolerance window, perform the following steps:

1. Select *PSpice – Advanced Analysis – Assign Tolerance* to open Assign Tolerance window.
2. The Assign Tolerance window displays R,C, L, V, and I in the Global List section of the Instance View tab.
3. Select the appropriate discrete device, such as, resistor.
4. Add positive tolerance (PosTol) or negative tolerance(NegTol) as percents. For example, 2% as PosTol and 1% as NegTol.
5. (Optional) Select the distribution type as one of the following: FLAT or GAUSS. If no distribution type is defined, FLAT will be used as the default distribution type.
6. Click Apply.

   Once tolerance is applied, a new entry is made for global tolerance in `PSpice.ini` under the `[PSPICE_ADVANCED_ANALYSIS]` section. The global tolerance is applicable not only to the current design, but to all the future designs also.

Case 2: Assigning Instance-level Tolerance to Discrete Devices

To assign tolerance to discrete devices using the Assign Tolerance window, perform the following steps:

1. Select *PSpice – Advanced Analysis – Assign Tolerance* to open Assign Tolerance window.
2. The Assign Tolerance window displays each instance of R,C, L, V, and I in the Instance List section of the Instance View tab. For example, if the design has 3 resistors and 2 capacitors. 3 different instances of resistors and 2 different instances of capacitors will be displayed.
3. Select the appropriate part, such as C2.
4. Add positive tolerance (PosTol) or negative tolerance(NegTol) as percents. For example, 2% as PosTol and 1% as NegTol.
5. (Optional) Select the distribution type as one of the following: FLAT or GAUSS. If no distribution type is defined, FLAT will be used as the default distribution type.
6. Click Apply.

   The instance level tolerance is applied to a particular instance of a device.

   Note: If global tolerance is assigned to a device and a instance tolerance is applied to an instance of same device. The instance level tolerance will override the global tolerance in a design.

Case 3: Assigning Tolerance to a Model Parameter

To assign tolerance to a model parameter using the Assign Tolerance window, perform the following steps:

1. Select *PSpice – Advanced Analysis – Assign Tolerance* to open the Assign Tolerance window.
2. The Assign Tolerance window displays each instance of the device used in the design in the Instance List section of the Instance View tab. For example, if a NPN Bipolar junction transistor (BJT) is added to a design, you will model name under the instance name of the device listed.

3. Select the model name, which is displayed under the instance name of the device. For example, if Q1 is the instance name and Q40235 is the model name. Q40235 will be displayed under Q1.
4. Click Edit PSpice Model to launch Model Editor.

   Using Model Editor, you will be able to add tolerance and distribution type to a model parameter.
5. In Model Text window of Model Editor, you will see the parameters' details mentioned. To add tolerance to a particular model parameter, add it after parameter details. For example, if parameter and it's value is mentioned as `Is = 69.28E-18`, you can add tolerance and distribution type as `Is = 69.28E-18 DEV/UNIFORM 10%/1%`.
6. Select *File – Save* or click the Save icon to save the PSpice model.

   Once the PSpice Model is saved, you can see the changes getting reflected in the Assign Tolerance window.
7. Click Apply on the Assign Tolerance window.

Case 4: Assigning Tolerance to Sub-circuit Model Parameters

To assign tolerance to model parameters of a sub-circuit, perform the following steps:

1. Select *PSpice – Advanced Analysis – Assign Tolerance* to open the Assign Tolerance window.
2. The Assign Tolerance window displays each instance of the device used in the design in the Instance List section of the Instance View tab. For example, if a sub-circuit is added to a design, you will see SubcktList listed in the Instance View tab. Inside SubcktList, the following hierarchy will be followed: Sub-circuit name - Model Instance Name - Model Name.
3. In the Model View tab, select the model name of the device. For example, if `X_D1.DFWD` is one sub-circuit and `X_D1.DLEAK` is the second sub-circuit. Then select the model name, `D180NQ045`, to edit it's PSpice model.
4. Click Edit PSpice Model to launch Model Editor.

Using Model Editor, you will be able to add tolerance and distribution type to a model parameter.

5. In Model Text window of Model Editor, you will see the parameters' details mentioned. To add tolerance to a particular model parameter, add it after parameter details. For example, if parameter and it's value is mentioned as `Is = 69.28E-18`, you can add tolerance and distribution type as `IS=10.05152E-6 DEV/GAUSS 2%/1%`. For more information on the syntax, see PSpice Reference Guide.
6. Select *File – Save* or click the Save icon to save the PSpice model.

   Once the PSpice Model is saved, you can see the changes getting reflected in the Assign Tolerance window.
7. Click Apply on the Assign Tolerance window.

Case 5: Assign Tolerance to Global Parameters

To assign tolerance to global parameters (.PARAM) in Capture, perform the following steps:

1. Select *PSpice – Advanced Analysis – Assign Tolerance* to open the Assign Tolerance window.
2. In the global list of the Instance View tab, you will see global parameters under the PARAM group.

   The global parameter can be used as an expression in any of the model parameter of a device, which can be either behavioral or standard device.
3. Select one of the global parameters under the PARAM group to add tolerance.
4. Add positive tolerance (PosTol) or negative tolerance(NegTol) as percents. For example, 2% as PosTol and 1% as NegTol.
5. (Optional) Select the distribution type as one of the following: FLAT or GAUSS. If no distribution type is defined, FLAT will be used as the default distribution type.
6. Click Apply.

Using Advanced Analysis libraries

There are three ways to quickly identify if a component is from an Advanced Analysis library:

- Using the library tool tip in the **Place Part** dialog box
- Using the Parameterized Part icon in the **Place Part** dialog box (Capture only)

Using the library tool tip

One easy way to identify if a component comes from an Advanced Analysis library is to use the tool tip in the **Place Part** dialog box.

1. From the Place menu, select **Part**.

   The **Place Part** dialog box appears.

2. Enter a component name in the **Part** text box.
3. Hover your mouse over the highlighted component name.

   A library path name appears in a tool tip.

4. Check for **ADVANLS** in the path name.

   If ADVANLS is in the path name, the component comes from an Advanced Analysis library.

Using Parameterized Part icon

Another easy way to identify if a component comes from an Advanced Analysis library is to use the Parameterized Part icon in the **Place Part** dialog box.

1. From the Place menu, select **Part**.

   The **Place Part** dialog box appears.

2. Enter a component name in the **Part** text box.

   Or:

   Scroll through the **Part List** text box

3. Look for  in the lower right corner of the dialog box.

This is the Parameterized Part icon. If this icon appears when the part name appears in the Part text box, the component comes from an Advanced Analysis library.

Preparing your design for Advanced Analysis

You may use a mixture of standard and parameterized components in your design, but Advanced Analysis is performed on only the parameterized components.

Note: From 17.2-2016 (HotFix 011), all advanced analysis, except smoke analysis, can be performed with only standard components in a Capture design.

You may create a new design or use an existing design for Advanced Analysis. There are several steps for making your design Advanced Analysis-ready.

See “Modifying existing designs for Advanced Analysis” on page 47.

Creating new Advanced Analysis-ready designs

Select parameterized components from Advanced Analysis libraries.

1. Open the online *Advanced Analysis library list* found in Cadence Online Documentation.
2. Find a component marked with a **Y** in the **TOL**, **OPT**, or **SMK** columns of the *Advanced Analysis library list*.

   Components marked in this manner are parameterized components.
3. For that component, write down the **Part Library** and **Part Name** from the *Advanced Analysis library list*.
4. Add the library to your design in your schematic editor.
5. Place the parameterized component on your schematic.

   For example, select the **resistor** component from the **pspice_elem** Advanced Analysis library.

Setting a parameter value

For each parameterized component in your design, set the parameter value individually on the component using your schematic editor.

A convenient way to add parameter values on a global basis is to use the design variable table.

See Using the design variables table on page 45.

Note: If you set a value for POSTOL and leave the value for NEGTOL blank, Advanced Analysis will automatically set the value of NEGTOL equal to the value of POSTOL and perform the analysis.

Note: As a minimum, you must set a value for POSTOL. If you set a value for NEGTOL and leave the POSTOL value blank, Advanced Analysis will not include the parameter in Sensitivity or Monte Carlo analyses.

Adding additional parameters

If the component does not have Advanced Analysis parameters visible on the symbol, add the appropriate Advanced Analysis parameters using your schematic editor.

For example: For RLC components, the parameters required for Advanced Analysis Sensitivity and Monte Carlo are listed below. The values shown are those that can be set using the design variables table.

See Using the design variables table on page 45.

Part	Tolerance Property Name	Value
Resistor	POSTOL	RTOL%
Resistor	NEGTOL	RTOL%
Inductor	POSTOL	LTOL%
Inductor	NEGTOL	LTOL%

Part	Tolerance Property Name	Value
Capacitor	POSTOL	CTOL%
Capacitor	NEGTOL	CTOL%

For RLC components, the parameter required for Advanced Analysis Optimizer is the value for the component. Examples are listed below:

Part	Optimizable Property Name	Value
Resistor	VALUE	10K
Inductor	VALUE	33m
Capacitor	VALUE	0.1u

For example: For RLC components, the parameters required for Advanced Analysis Smoke are listed below. The values shown are those that can be set using the design variables table.

See Using the design variables table on page 45.

Part	Smoke Property Name	Value
Resistor	MAX_TEMP	RTMAX
Resistor	POWER	RMAX
Resistor	SLOPE	RSMAX
Resistor	VOLTAGE	RVMAX
Inductor	CURRENT	DIMAX
Inductor	DIELECTRIC	DSMAX
Capacitor	CURRENT	CIMAX
Capacitor	KNEE	CBMAX
Capacitor	MAX_TEMP	CTMAX
Capacitor	SLOPE	CSMAX
Capacitor	VOLTAGE	CMAX

If you use RLC components from the “analog” library, you will need to add parameters and set values; however, instead of setting values for the POSTOL and NEGTOL parameters, you set the values for the TOLERANCE parameter. The positive and negative tolerance values will use the value assigned to the TOLERANCE parameter.

Updating Existing Designs for Advanced Analysis

You can simulate existing designs and legacy PSpice models using PSpice Advanced Analysis. To simulate legacy PSpice or existing designs in PSpice Advanced Analysis, perform the following steps:

1. Update the design to use parts that have model associations.

   Note: Cores are not supported in this flow.

2. Open the parts using Model Editor and assign DEV or LOT tolerances. DEV values are used for simulation if both DEV and LOT values are assigned.

3. Create a PSpice simulation profile as shown in Figure 2-1 on page 40.

   a. Ensure that you select the Monte Carlo option in the Analysis tab of the Simulation Settings dialog box.

   b. To enable legacy support, select Enable PSpice AA support for legacy

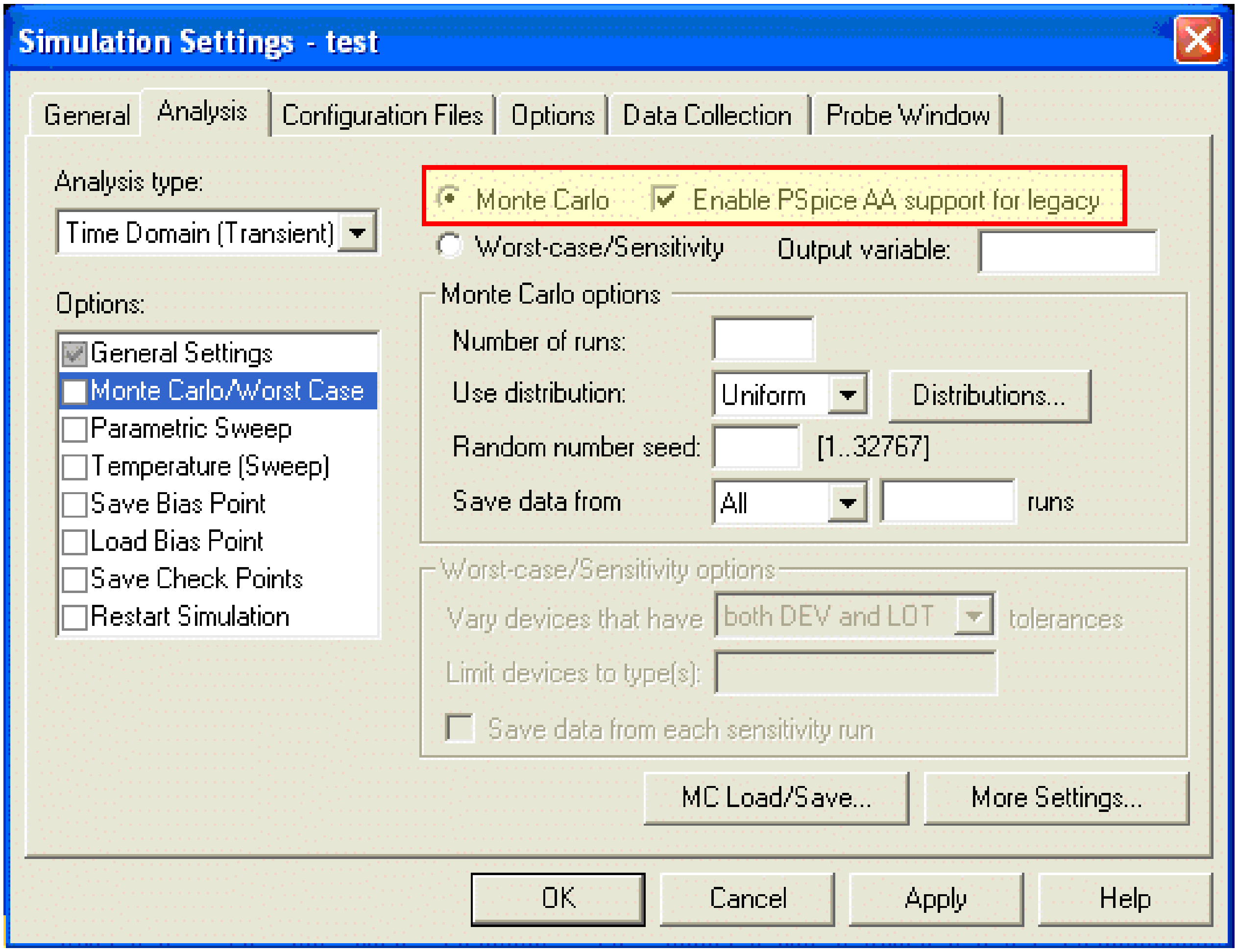


Figure 2-1 Simulation Settings dialog box with Monte Carlo selected

4. Create netlist for the design.

   Note: The netlisting process might be slower if legacy support is selected. However, the simulation performance is not affected.

5. Optionally, run the simulation in PSpice.
6. Run Advanced Analysis tools.

Limitations

The legacy PSpice model support has the following limitations:

- Cores are not supported in the flow.
- Although the performance of simulation is not affected, there might be a reduction in performance while netlisting.
- The formula used to calculate tolerance values for PSpice AA from DEV or TOL is an approximation. Therefore, the results might differ.
- Although the trends will be similar to PSpice AA results in comparison to basic PSpice analyses, they might not match exactly.

Example: Legacy Support

In this example you will simulate an active band-pass filter circuit with a legacy capacitor component, `Cgauss`, as shown in Figure 2-2 on page 42.

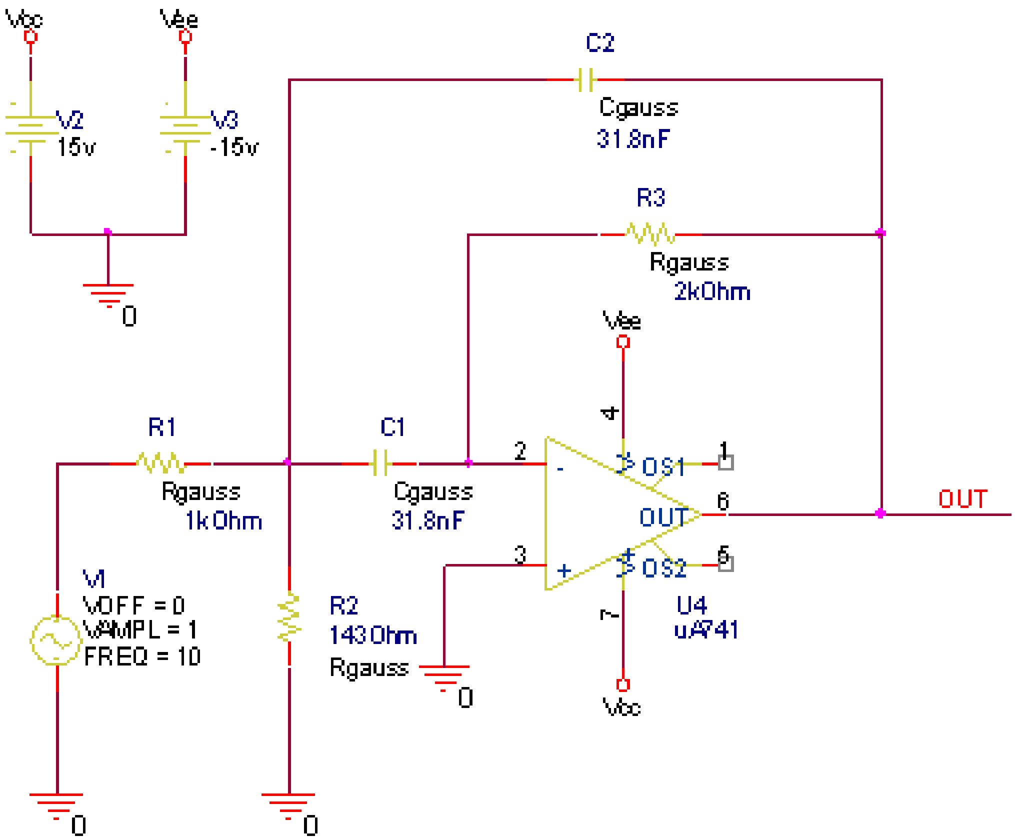


Figure 2-2 Band-Pass Filter Circuit with Legacy Component

Specify a DEV value of 5% to the Cgauss capacitor model as:

```
.model Cgauss CAP C=1 DEV/truegauss=5%
```

Specify the different parameters in the Simulation Settings dialog box, as shown in Figure 2-3 on page 43 and then create a netlist.

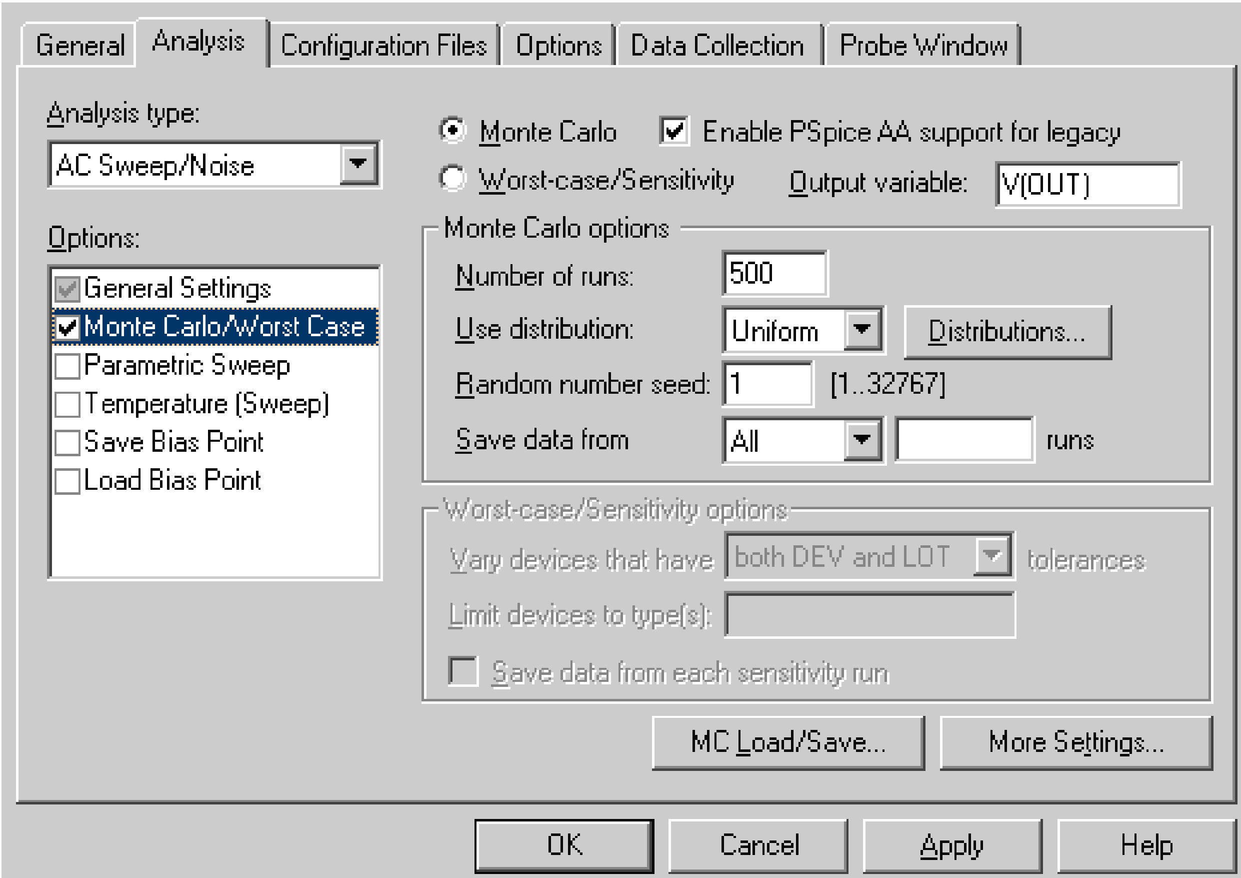


Figure 2-3 Enable PSpice AA support for legacy Selected

Simulate the circuit in PSpice. Figure 2-4 on page 44 shows the simulation results.

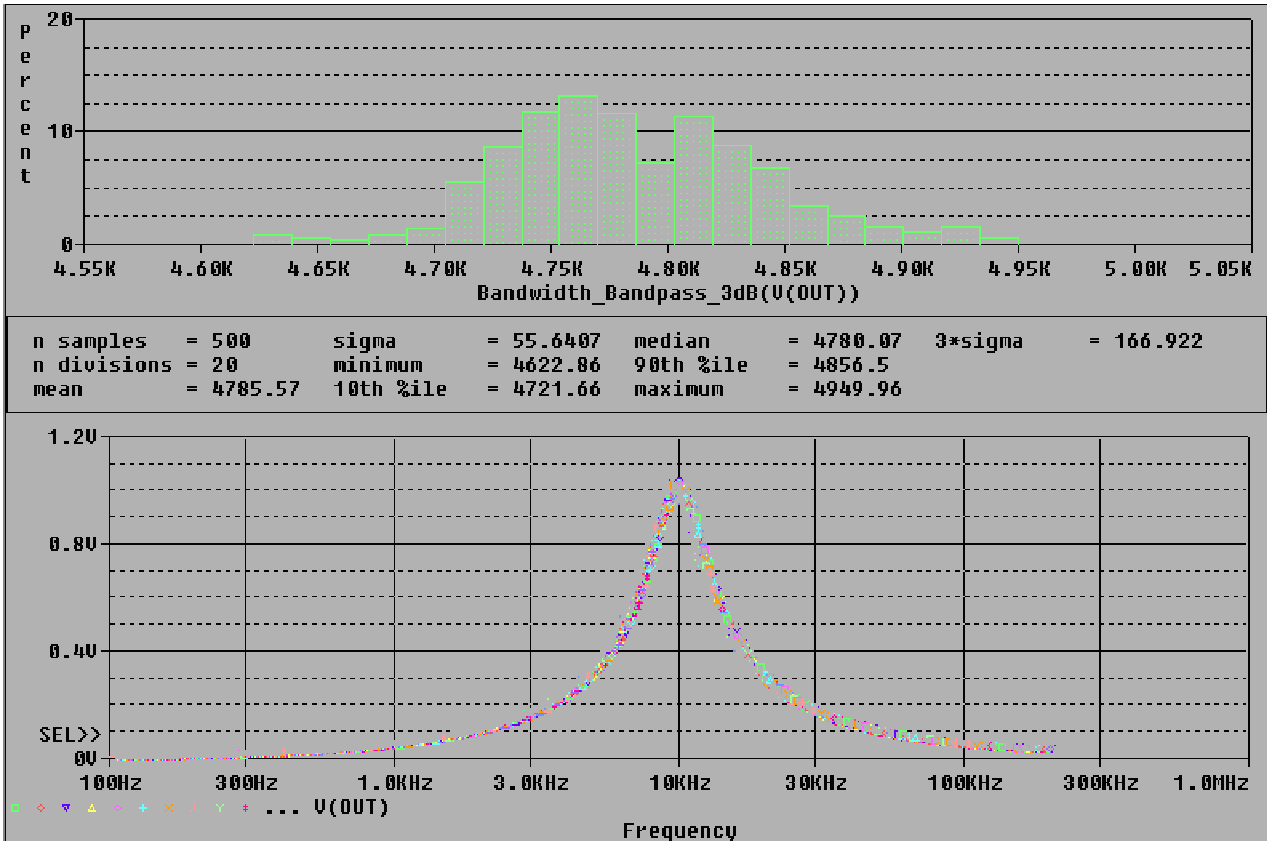


Figure 2-4 Result Simulation Run in PSpice

You can now simulate the circuit in Advanced Analysis to get the result shown in Figure 2-5 on page 45.

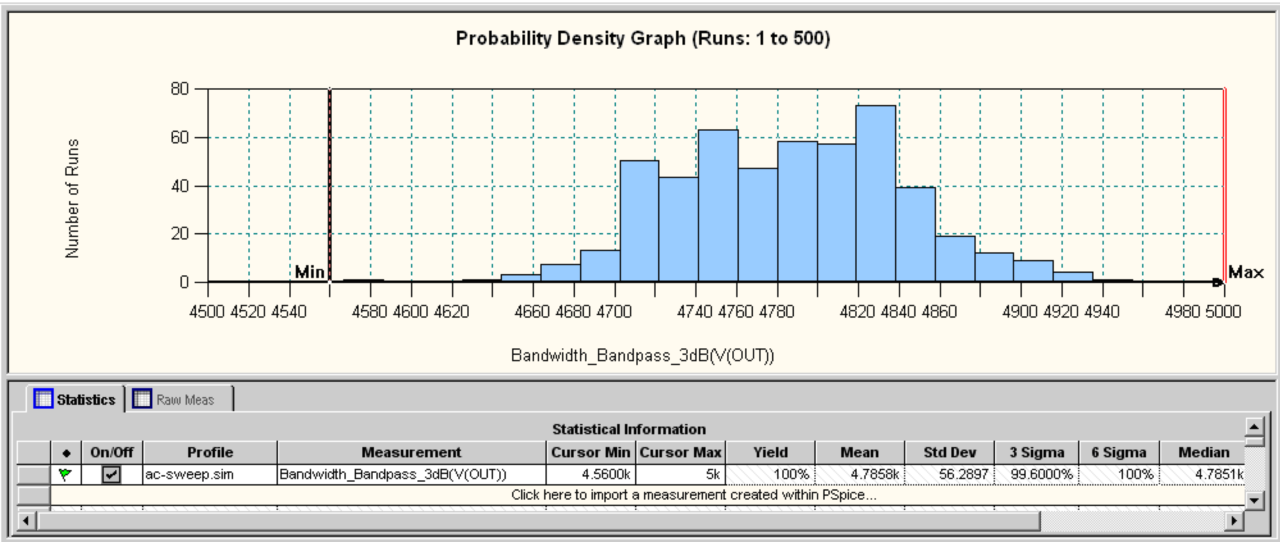


Figure 2-5 Simulation of Circuit with Legacy component in Advanced Analysis

Using the design variables table

The design variables table is a component available in the installed libraries that allows you to set global values for parameters. For example, using the design variables table, you can easily set a 5% positive tolerance on all your circuit resistors. The default information available in the design variables table includes variable names for tolerance and smoke parameters. For example, RTOL is a variable name in the design variables tables, which can be used to set POSTOL (and NEGTOL) tolerance values on all your circuit resistors.

In Capture:

1. Choose *Place* - *Part*.
2. Add the PSpice `pspice_elem` library to your design libraries.
3. Select the *Variables* component from the PSpice pspice_elem library.
4. Click *OK*.

A design variable table of parameter variable names will appear on the schematic.

5. Double click on a number in the design variable table.

   The **Display Properties** dialog box will appear.

6. Edit the value in the *Value* text box.

7. Click *OK*.

   The new numerical value will appear on the design variables table on the schematic and be used as a global value for all applicable components.

Parameter values set on a component instance will override values set in the design variables table.

In Design Entry HDL:

1. Choose *Component* - *Add* to open the Component Browser.
2. Under Browse Libraries, from the libraries list, select `pspice_elem`.
3. From the Cells list, select *variables*.
4. Click *Add*.
5. Place the design variable table of parameter variable names on the schematic.
6. Open the Attributes dialog box for a variable.
7. Edit the value in the *Value* column.
8. Click *OK*.

Modifying existing designs for Advanced Analysis

Existing designs that you construct with standard components will work in Advanced Analysis; however, you can only perform Advanced Analysis on the parameterized components. To make sure specific components are Advanced Analysis-ready (parameterized), do the following steps:

- Set tolerances for the RLC components

  Note: For standard RLC components, the TOLERANCE property can be used to set tolerance values required for Sensitivity and Monte Carlo. Standard RLC components can also be used in the Optimizer.

- Replace active components with parameterized components from the Advanced Analysis libraries

- Add smoke parameters and values to RLC components

Example

This example is a simple addition of a parameterized component to a new design.

We'll add a parameterized resistor to a schematic and show how to set values for the resistor parameters using the property editor and the design variables table in capture and using the Attributes dialog box and the Variables component in Design Entry HDL.

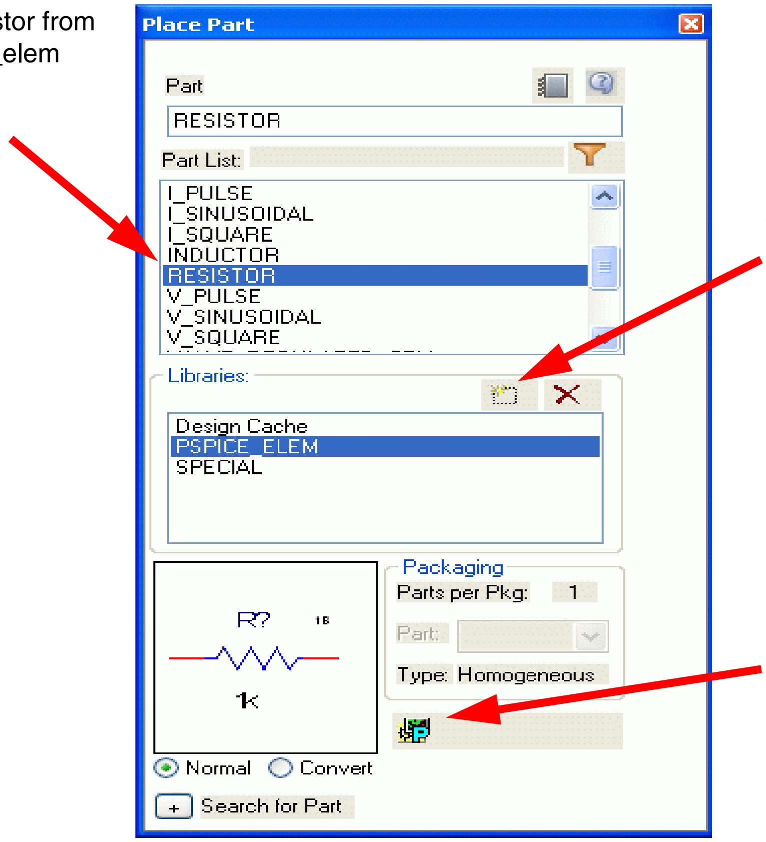
We know the `pspice_elem` library on the *Advanced Analysis library list* contains a resistor component with tolerance, optimizable, and smoke parameters. We'll use that component in our example.

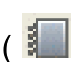
Selecting a parameterized component

In Capture:

1. In Capture, from the Place menu, select **Part.**

The **Place Part** dialog box appears.



2. Use the Add Library browse button to add the **pspice_elem** library from the advanls folder to the **Libraries** text box.
3. Select **Resistor** and click **Place Part** ( .).

   The resistor appears on the schematic.

In Design Entry HDL:

Add the pspice_elem library to your example project.

1. From the *Component* menu, select *Add*.

   The Add Component dialog box will appear.
2. Select the *pspice_elem* library in the *Library* drop-down list.
3. Select the Resistor component in the Cells text box.
4. Click Close.

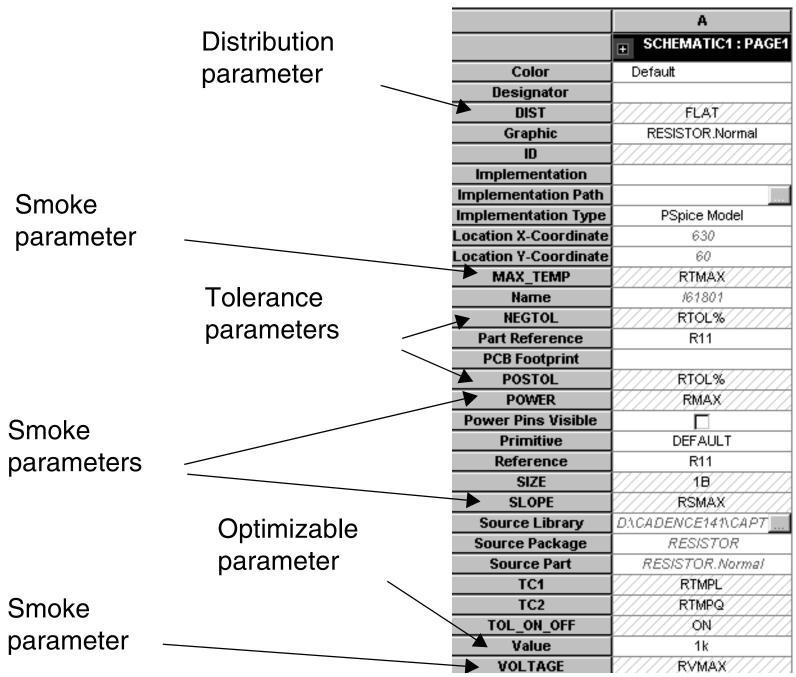
The resistor appears on the schematic.

Setting a parameter value

In Capture:

1. Double click on the Resistor symbol.

   The Property Editor appears. Note the Advanced Analysis parameters already listed for this component.



2. Verify that all the parameters required for Sensitivity, Optimizer, Smoke, and Monte Carlo are visible on the symbol.

   Refer to the tables in Adding additional parameters on page 37.

3. Set the resistor **VALUE** parameter to 10k.

4. Set the resistor **POSTOL** parameter to **RTOL%**.

In Design Entry HDL:

1. From the Text menu, select *Attributes*, then click on the Resistor component instance.

The Attributes dialog box appears. Note the Advanced Analysis parameters already listed for this component.

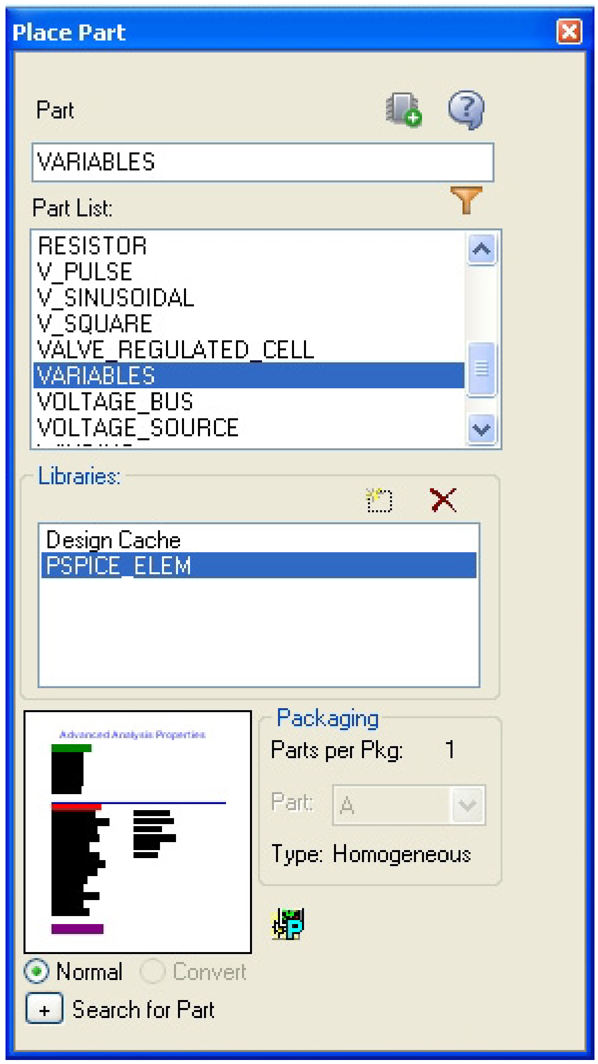
2. Set the resistor VALUE parameter to `10K`.
3. Note the resistor POSTOL parameter is already set to *RTOL%*.
4. Click *OK*.

Using the design variables table

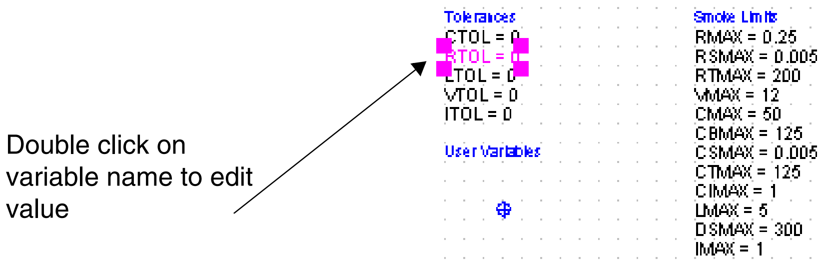
Set the resistor parameter values using the design variables table.

We'll do one parameter for this resistor.

1.Select the **Variables** part from the PSpice pspice_elem library.

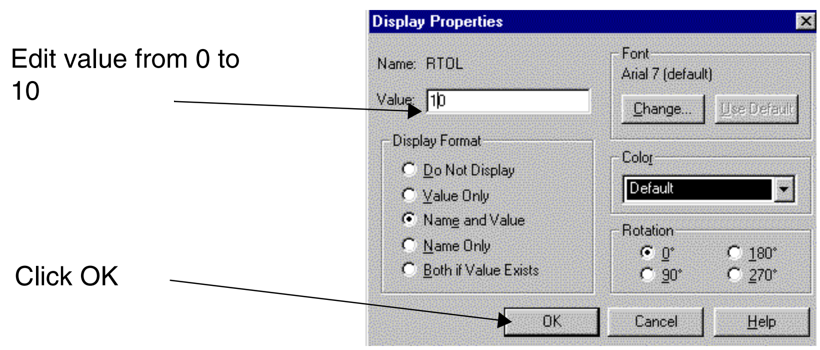


The design variables table appears on the schematic.



2. Double click on the RTOL number **0** in the design variables table.

   The **Display Properties** dialog box appears.



3. Edit the value in the **Value** text box.
4. Click **OK**.

   The new numerical value will appear on the design variable table on the schematic.

Advanced Analysis will now use the resistor with a positive tolerance parameter set to 10%. If we added more resistors to this design, we could then set the POSTOL resistor parameter values to RTOL% and each resistor would immediately apply the 10% value from the design variables table.

Note: Values set on the component instance override values set with the design variables table.

For power users

Legacy PSpice optimizations

For tips on importing legacy PSpice Optimizations into Advanced Analysis Optimizer, see our technical note on importing legacy PSpice optimizations.

3

Sensitivity

In this chapter

- Sensitivity overview on page 55
- Sensitivity strategy on page 56
- Sensitivity procedure on page 58
- Example on page 68
- For power users on page 81

Sensitivity overview

Note: Sensitivity analysis is available with the following products:

- PSpice[1] Advanced Optimizer Option
- PSpice Advanced Analysis

Sensitivity identifies which components have parameters critical to the measurement goals of your circuit design.

The Sensitivity Analysis tool examines how much each component affects circuit behavior by itself and in comparison to the other components. It also varies all tolerances to create worst-case (minimum and maximum) measurement values.

You can use Sensitivity to identify the sensitive components, then export the components to Optimizer to fine-tune the circuit behavior.

You can also use Sensitivity to identify which components affect yield the most, then tighten tolerances of sensitive components and

1. Depending on the license available, you will access either PSpice or PSpice Simulator.

loosen tolerances of non-sensitive components. With this information you can evaluate yield versus cost trade-offs.

Absolute and relative sensitivity

Sensitivity displays the absolute sensitivity or the relative sensitivity of a component. Absolute sensitivity is the ratio of change in a measurement value to a one unit positive change in the parameter value.

For example: There may be a 0.1V change in voltage for a 1 Ohm change in resistance.

Relative sensitivity is the change in a measurement based on one percent positive change of component parameter value.

For example: For each 1 percent change in resistance, there may be 2 percent change in voltage.

Since capacitor and conductor values are much smaller than one unit of measurement (Farads or Henries), relative sensitivity is the more useful calculation.

For more on how this tool calculates sensitivity, see Sensitivity calculations on page 81.

Absolute sensitivity should be used when the tolerance limits are not tight or have wide enough bandwidth. Where as relative sensitivity should be used when the tolerance limits are tight enough or have less bandwidth. The tolerance variations are assumed to be linear in this case.

Sensitivity strategy

If Sensitivity analysis shows that the circuit is highly sensitive to a single parameter, adjust component tolerances on the schematic and rerun the analysis before continuing on to Optimizer.

Optimizer works best when all measurements are initially close to their specification values and require only fine adjustments.

Plan ahead

Sensitivity requires:

- Circuit components that are Advanced Analysis-ready

  See Chapter 2, Libraries for more information.

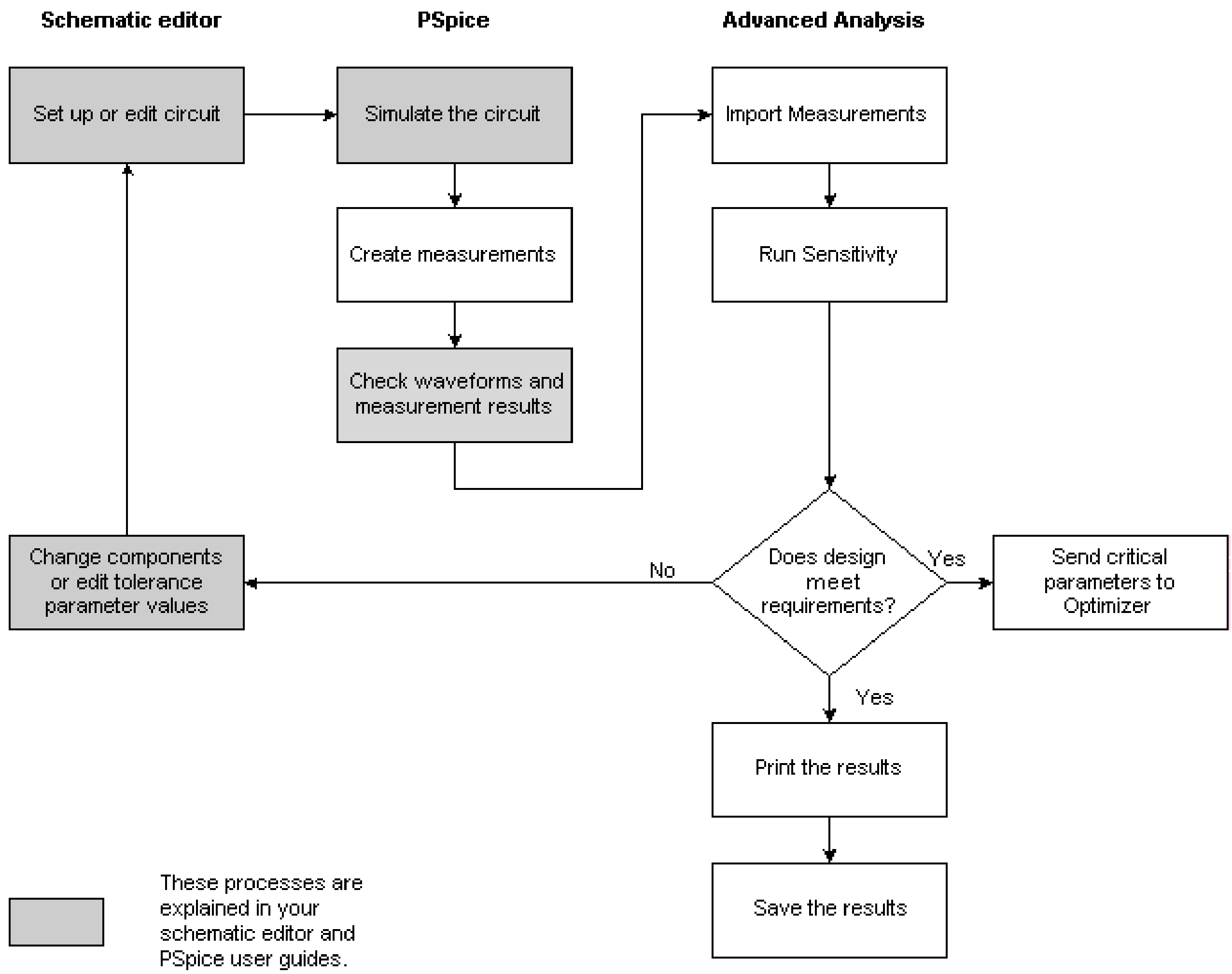
- A circuit design, that is working and can be simulated in PSpice
- Measurements set up in PSpice

  See Procedure for creating measurement expressions on page 258

Any circuit components you want to include in the Sensitivity data need to be Advanced Analysis-ready, with their tolerances specified.

See Chapter 2, Libraries for more information.

Workflow



Sensitivity procedure

Setting up the circuit in the schematic editor

Start with a working circuit in the schematic editor. Circuit components you want to include in the Sensitivity data need to have the tolerances of their parameters specified. Circuit simulations and measurements should already be set up.

The simulations can be Time Domain (transient), DC Sweep, and AC Sweep/Noise analyses.

1. Open your circuit from your schematic editor.
2. Run a PSpice simulation.
3. Check your key waveforms in PSpice and make sure they are what you expect.
4. Check your measurements and make sure they have the results you expect.

Note: For information on circuit layout and simulation setup, see your schematic editor and PSpice user guides.

For information on components and the tolerances of their parameters, see Preparing your design for Advanced Analysis on page 36.

For information on setting up measurements, see Procedure for creating measurement expressions on page 258.

For information on testing measurements, see Viewing the results of measurement evaluations on page 260.

Setting up Sensitivity in Advanced Analysis

1. From the **PSpice** menu in your schematic editor, select **Advanced Analysis** / **Sensitivity**.

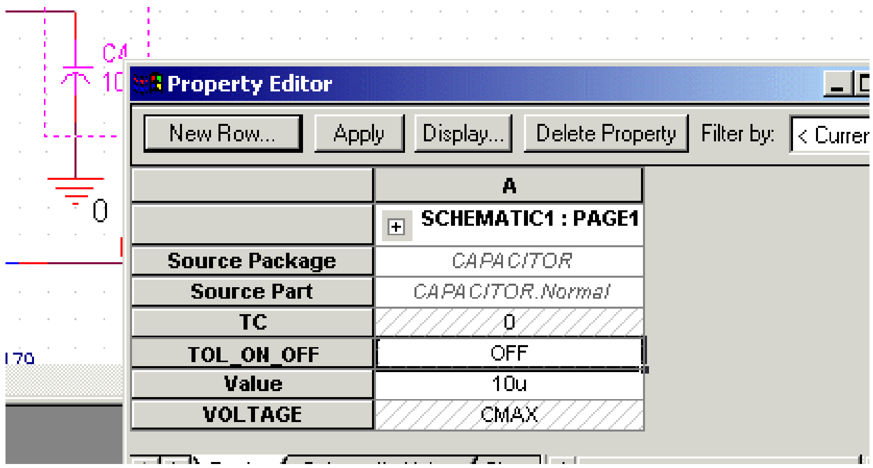
   The Advanced Analysis Sensitivity tool opens.

Parameters Window

In the Parameters window, a list of component parameters appears with the parameter values listed in the Original column. Only the parameters for which tolerances are specified appear in the Parameters window.

Note: Sensitivity analysis can only be run if tolerances are specified for the component parameters.

In case you want to remove a parameter from the list, you can do so by using the TOL_ON_OFF property. In the schematic design, set the value of TOL_ON_OFF property attached to the instance as OFF. If there is no TOL_ON_OFF property attached to the instance of the device, attach the property and set its value to OFF. This is so, because if the tolerance value is specified for a parameter and TOL_ON_OFF property is not attached to the component, by default Advanced Analysis assumes that the value of TOL_ON_OFF property is set to ON.



In case of hierarchical designs, the value of the TOL_ON_OFF property attached to the hierarchical block has a higher priority over the property value attached to the individual components. For example, if the hierarchical block has the TOL_ON_OFF property value set to OFF, tolerance values of all the components within that hierarchical design will be ignored.

Specifications Window

In the Specifications window, add measurements for which you want to analyze the sensitivity of the parameters. You can either import the measurements created in PSpice or can create new measurements in Advanced Analysis.

To import measurements:

1. In the Specifications table, click on the row containing the text “Click here to import a measurement created within PSpice.”

   The **Import Measurement(s)** dialog box appears.

2. Select the measurements you want to include.

To create new measurements:

1. From the **Analysis** drop-down menu, choose **Sensitivity** / **Create New Measurements**.

   The **New Measurement** dialog box appears.

2. Create the measurement expression to be evaluated and click OK.

Running Sensitivity

- Click  on the top toolbar.

  The Sensitivity analysis begins. The messages in the output window tell you the status of the analysis.

  For more information, see Sensitivity calculations on page 81.

Displaying run data

Sensitivity displays results in two tables for each selected measurement:

- Parameters table
  - Parameter values at minimum and maximum measurement values
  - Absolute / Relative sensitivities per parameter
  - Linear / Log bar graphs per parameter
- Specifications table
  - Worst-case min and max measurement values

Sorting data

➡ Double click on column headers to sort data in ascending or descending order.

Reviewing measurement data

➡ Select a measurement on the Specifications table.

A black arrow appears in the left column on the Specifications table, the row is highlighted, and the **Min** and **Max** columns display the worst-case minimum and maximum measurement values.

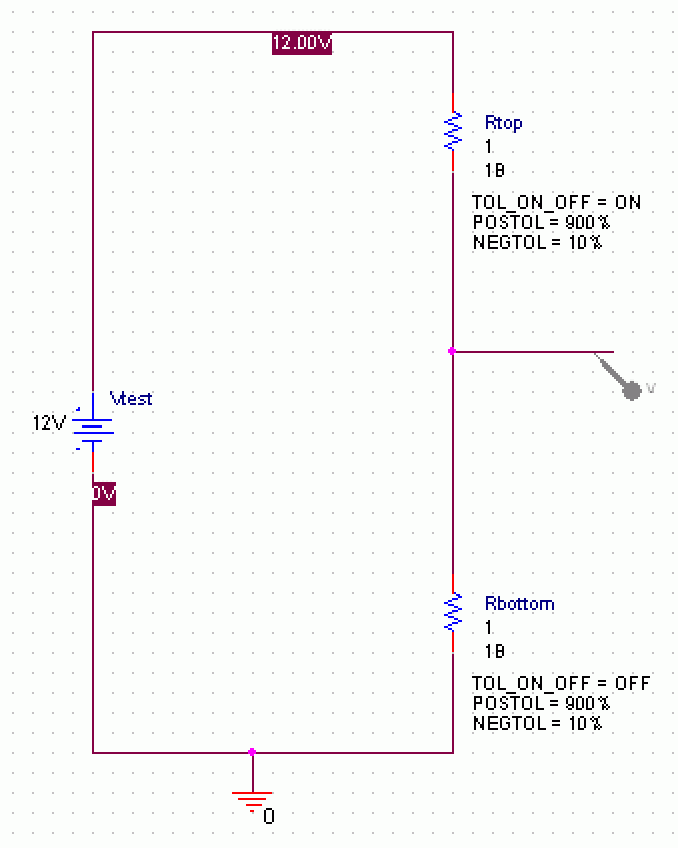
The Parameters table will display the values for parameters and measurements using the selected measurement only.

Interpreting @min and @max

Values displayed in the **@min** and **@max** columns are the parameter values at the measurement's worst-case minimum and maximum values.

If a measurement value is insensitive to a component, the sensitivity displayed for that component will be zero. In such cases, values displayed in the @Min and @Max columns will be same and will be equal to the Original value of the component.

The @Min and @Max columns display values depending upon which defined goal (Measurement) specification has the minimum and maximum value respectively. These minimum and maximum bound of component value can be either of the tolerances, POSTOL and NEGTOL, depending upon measurement specifications.For example, in the circuit shown below, maximum voltage exists across the resistor Rbottom exists when Rbottom has maximum value and TOLERANCE is set to OFF for Rtop and set to ON for Rbottom. As a result, @Min has the lower value given by NEGTOL for defined goal specification's minimum value and @Max bound has a higher value given by POSTOL for defined goal specification's maximum value. Similarly, if TOLERANCE is ON for Rtop and OFF for Rbottom, then Rbottom has maximum voltage when Rtop has minimum value. As a result, @Min bound has higher value given by POSTOl and @MAX bound has lower value given by NEGTOL.



Negative and positive sensitivity

If the absolute or the relative sensitivity is negative it implies that for one unit positive increase in the parameter value, the measurement value increases in the negative direction.

For example, if for a unit increase in the parameter value, the measurement value decreases, the component exhibits negative sensitivity. It can also be that for a unit decrease in the parameter value, there is an increase in the measurement value.

On the other hand, positive sensitivity implies that for a unit increase in the component value, there is an increase in the measurement value.

Changing from Absolute to Relative sensitivity

1. Right click anywhere in the Parameters table.
2. Select **Display** / **Absolute Sensitivity** or **Relative Sensitivity** from the pop-up menu.

Note: See Sensitivity calculations on page 81.

Changing bar graph style from linear to log

Most of the sensitivity values can be analyzed using the linear scale. Logarithmic scale is effective for analyzing the smaller but non-zero sensitivity values.

To change the bar graph style,

1. Right-click anywhere in the Parameters table.
2. Select **Bar Graph Style** / **Linear** or **Log** from the pop-up menu.

Important

If 'X' is the bar graph value on a linear scale, then the bar graph value on the logarithmic scale is not log (X). The logarithmic values are calculated separately.

Interpreting <MIN> results

Sensitivity displays <MIN> on the bar graph when sensitivity values are very small but nonzero.

Interpreting zero results

Sensitivity displays zero in the absolute / relative sensitivity and bar graph columns if the selected measurement is not sensitive to the component parameter value.

Controlling Sensitivity

Data cells with cross-hatched backgrounds are read-only and cannot be edited. The graphs are also read-only.

Pausing, stopping, and starting

Pausing and resuming

1. Click on the top toolbar.

   The analysis stops, available data is displayed, and the last completed run number appears in the output window.

2. Click the or to resume calculations.

Stopping

➡ Click on the top toolbar.

If a Sensitivity analysis has been stopped, you cannot resume the analysis.

Sensitivity does not save data from a stopped analysis.

Starting

➡ Click to start or restart.

Controlling measurement specifications

- To exclude a measurement specification from Sensitivity analysis: click on the applicable measurement row in the Specifications table.

  This removes the check and excludes the measurement from the next Sensitivity analysis.

- To add a new measurement: click on the row containing the text “Click here to import a measurement created within PSpice.”

  The **Import Measurement(s)** dialog box appears.

Or:

Right click on the Specifications table and select **Create New Measurement**.

The **New Measurement** dialog box appears.

See Procedure for creating measurement expressions on page 258.

- ❑ To export a new measurement to Optimizer or Monte Carlo, select the measurement and right click on the row containing the text "Click here to import a measurement created within PSpice."

  Select **Send To** from the pop-up menu.

Adjusting component values

Use **Find in Design** from Advanced Analysis to quickly return to the schematic editor and change component information.

For example: You may want to tighten tolerances on component parameters that are highly sensitive or loosen tolerances on component parameters that are less sensitive.

1. Right click on the component's critical parameter in the Sensitivity Parameters table and select **Find in Design** from the pop-up menu.
2. Change the parameter value in the schematic editor.
3. Rerun the simulation and check results.
4. Rerun Sensitivity.

Varying the tolerance range

During Sensitivity analysis, by default Advanced Analysis varies parameter values by 40% of the tolerance range. You can modify the default value and specify the percentage by which the parameter values should be varied within the tolerance range.

To specify the percentage variation:

1. From the **Edit** drop-down menu in Advanced Analysis, choose **Profile Settings**.
2. In the Profile Settings dialog box, select the **Sensitivity** tab.
3. In the Sensitivity Variation text box, specify the percentage by which you want the parameter values to be varied.
4. Click OK to save the modifications.

If you now run the Sensitivity analysis, the value specified by you would be used for calculating the absolute and relative sensitivity.

Sending parameters to Optimizer

1. Select the critical parameters in Sensitivity.
2. Right click and select **Send to Optimizer** from the pop-up menu.
3. Select **Optimizer** from the drop-down list on the top toolbar.

   This switches the active window to the Optimizer view where you can double check that your critical parameters are listed in the Optimizer Parameters table.
4. Click the **Sensitivity** tab at the bottom of the Optimizer Specifications table.

   This switches the active window back to the Sensitivity tool.

Printing results

➡ Click .

Or

From the File menu, select Print.

Saving results

➡ Click .

Or

From the File menu, select Save.

The final results will be saved in the Advanced Analysis profile (.aap).

Example

The Advanced Analysis examples folder contains several demonstration circuits. This example uses the RFAmp circuit.

The circuit contains components with the tolerances of their parameters specified, so you can use the components without any modification.

Two PSpice simulation profiles have already been created and tested. Circuit measurements, entered in PSpice, have been set up and tested.

Note: See Chapter 2, Libraries for information about setting tolerances for other circuit examples.

Setting up the circuit in the schematic editor

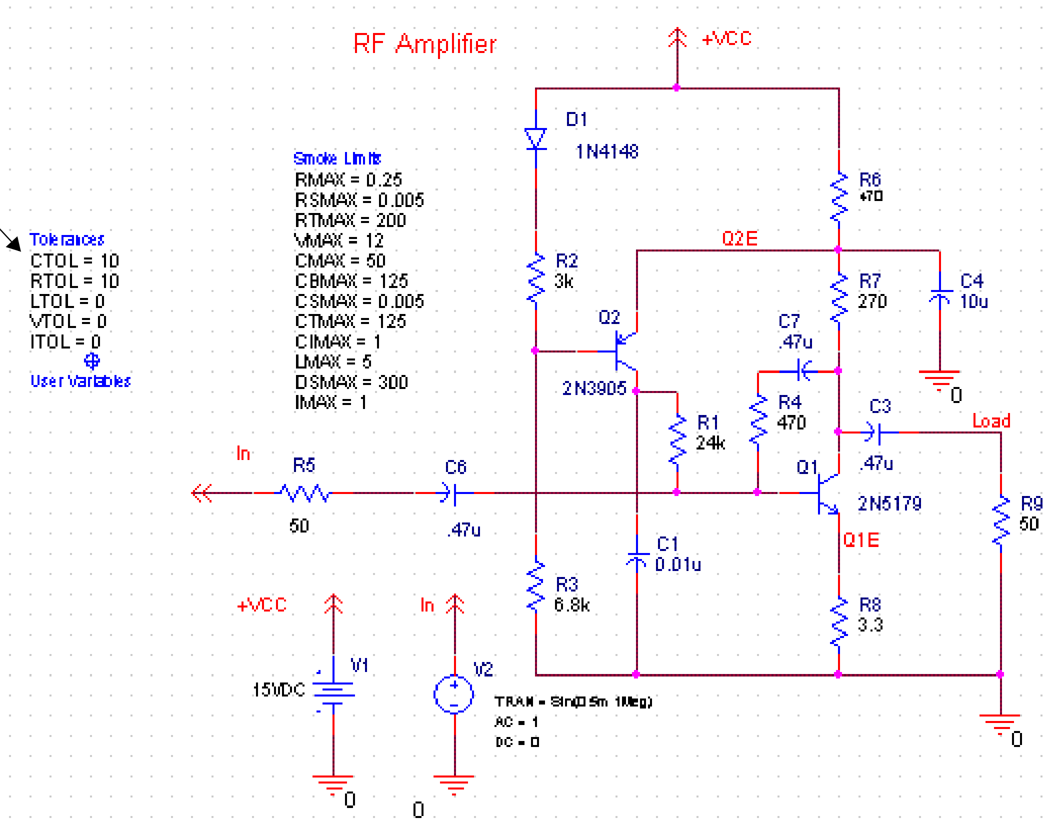
1. In your schematic editor, browse to the RFAmp tutorials directory.

   `<target_directory>\PSpice\tutorial\Capture\pspiceaa\rfamp`

   `<target_directory>\PSpice\tutorial\Concept\pspiceaa\rfamp`

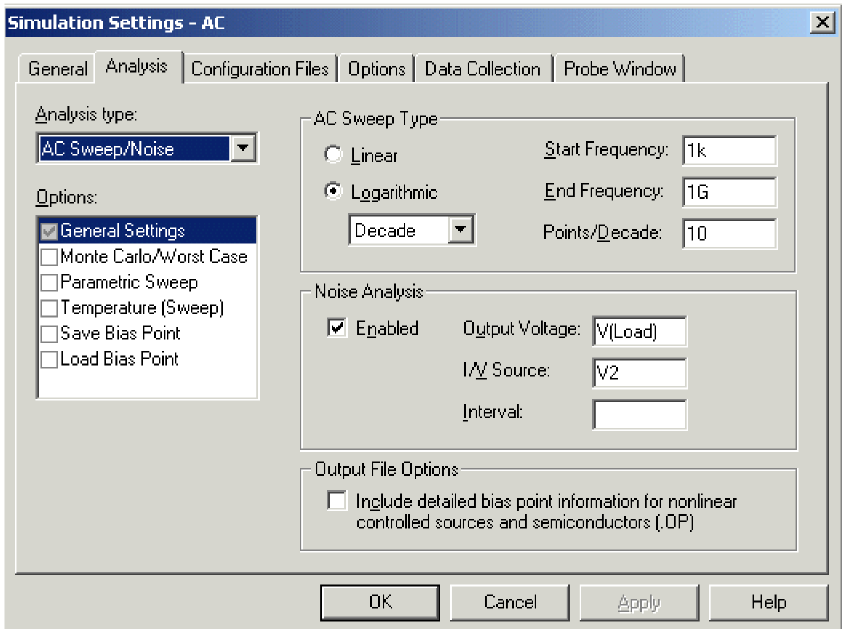
2. Open the RFAmp project.

Assign global tolerances using this table

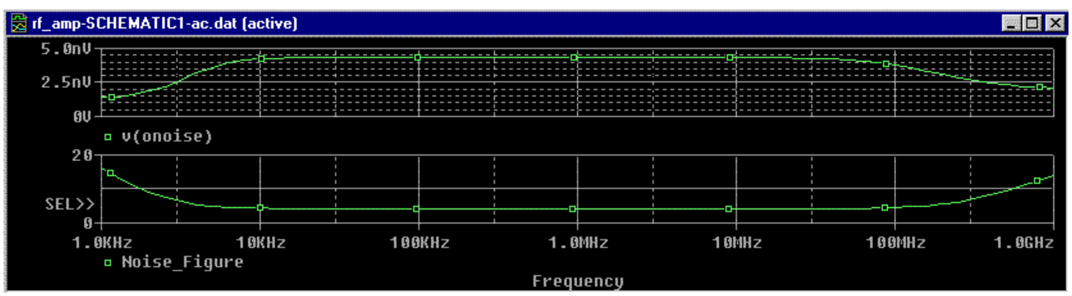


3. Select the SCHEMATIC1-AC simulation profile.

The AC simulation included with the RF example



1. Click  to run the simulation.
2. Review the results.



The waveforms are what we expected.

In the simulator, view measurement results

	Evaluate	Measurement	Value	Measurement
	☑	max(db(v(load)))	9.41807	
	☑	bandwidth(v(load),3)	150.57877meg	
	☑	min(10*log10(v(inoise)*v(inoise)/8.28...	4.14805	
	☑	max(v(onoise))	4.33832n	

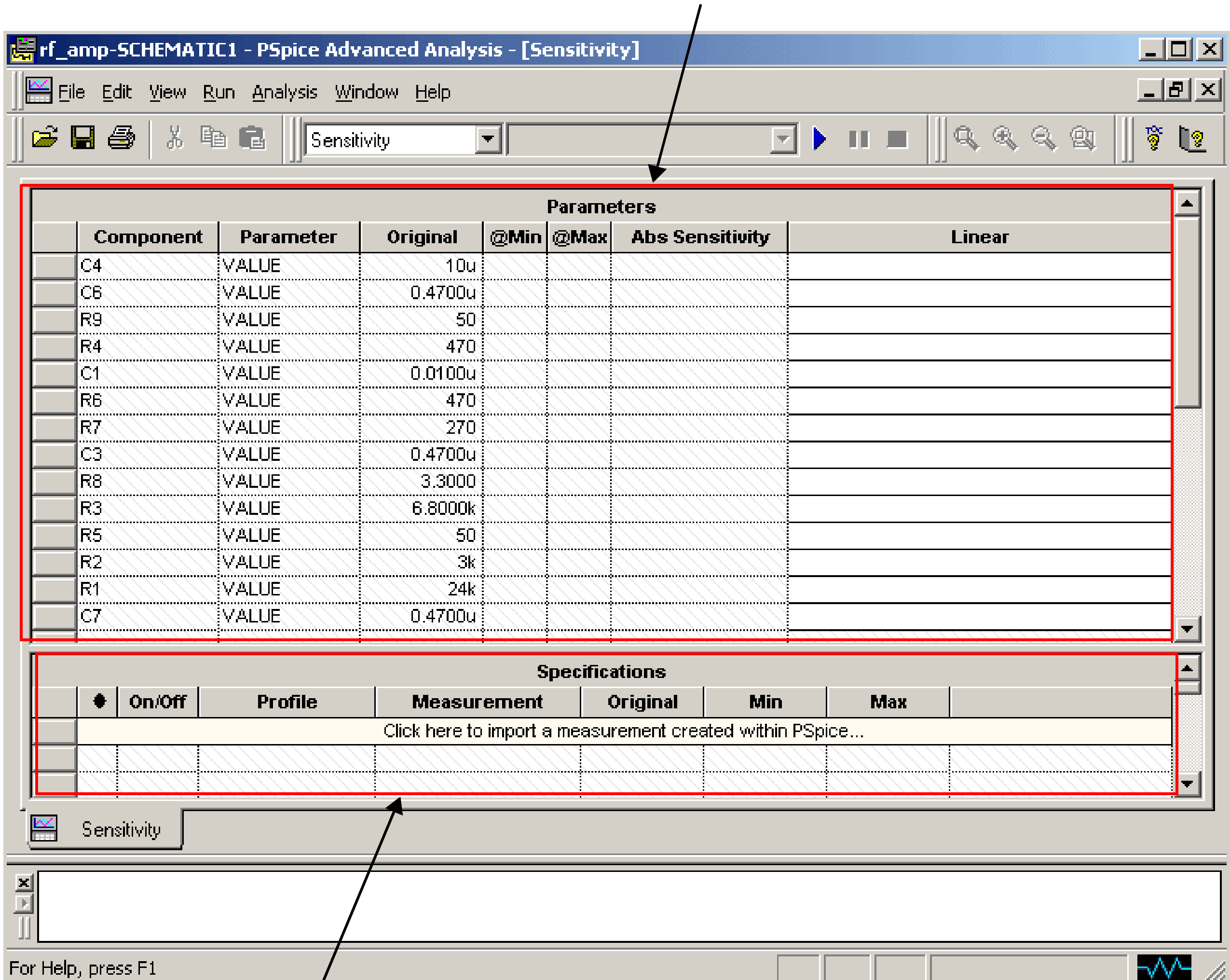
The measurements in PSpice give the results we expected.

Setting up Sensitivity in Advanced Analysis

1. From the **PSpice** menu in your schematic editor, select **Advanced Analysis** / **Sensitivity**.

   The Advanced Analysis window opens, and the Sensitivity tool is activated. Sensitivity automatically lists component parameters for which tolerances are specified and the component parameter original (nominal) values.

Sensitivity Parameters table prior to the first run

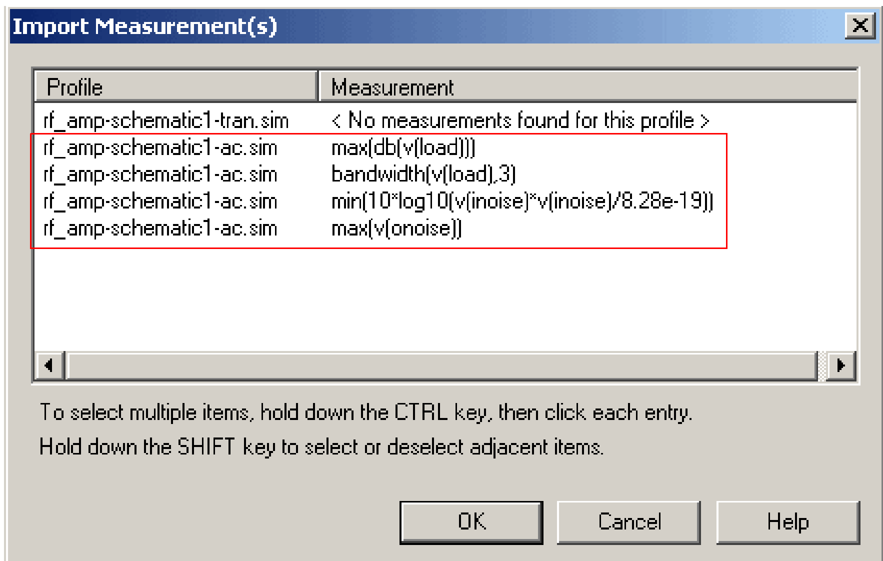


Sensitivity Specifications table before a project is set up and run

In case you want to remove some parameters from the Parameters list, you can do so by modifying the parameter properties in the schematic tool.

2. In the Specifications table, right click the row titled, “Click here to import a measurement created within PSpice.”

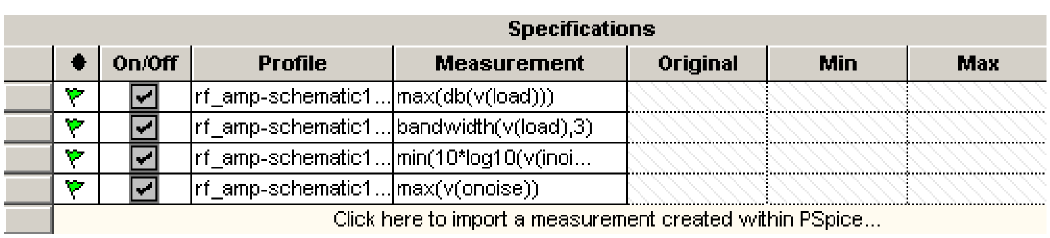
   The **Import Measurement(s)** dialog box appears with measurements configured earlier in PSpice .



3. Select the four ac.sim measurements.

4. Click **OK**.

   The Specifications table lists the measurements.

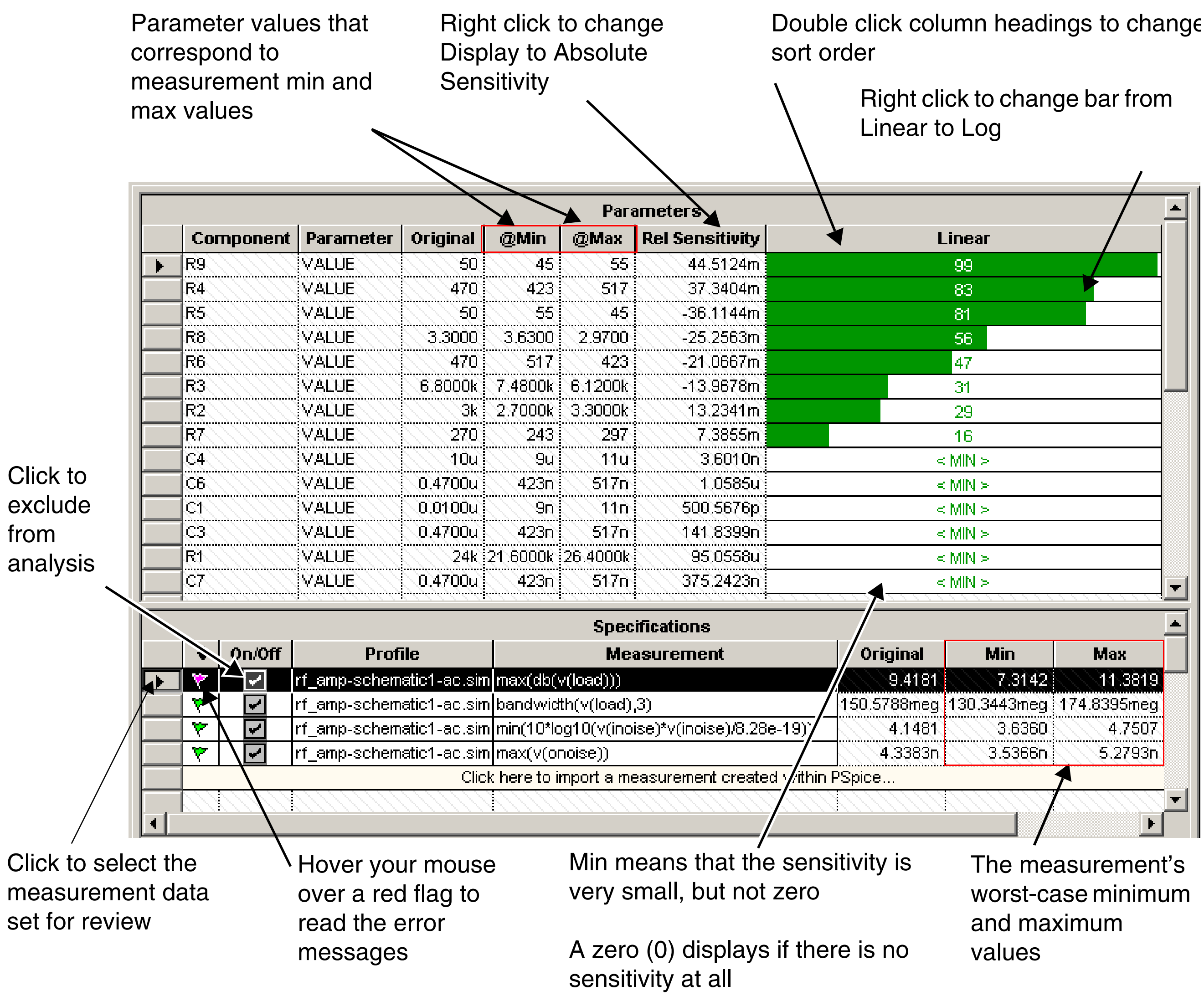


Running Sensitivity

➡ Click  on the top toolbar.

Displaying run data

Results are displayed in the Parameters and Specifications tables according to the selected measurement.



Component	Parameter	Original	@Min	@Max	Rel Sensitivity	Linear
R9	VALUE	50	45	55	44.5124m	99
R4	VALUE	470	423	517	37.3404m	83
R5	VALUE	50	55	45	-36.1144m	81
R8	VALUE	3.3000	3.6300	2.9700	-25.2563m	56
R6	VALUE	470	517	423	-21.0667m	47
R3	VALUE	6.8000k	7.4800k	6.1200k	-13.9678m	31
R2	VALUE	3k	2.7000k	3.3000k	13.2341m	29
R7	VALUE	270	243	297	7.3855m	16
C4	VALUE	10u	9u	11u	3.6010n	< MIN >
C6	VALUE	0.4700u	423n	517n	1.0585u	< MIN >
C1	VALUE	0.0100u	9n	11n	500.5676p	< MIN >
C3	VALUE	0.4700u	423n	517n	141.8399n	< MIN >
R1	VALUE	24k	21.6000k	26.4000k	95.0558u	< MIN >
C7	VALUE	0.4700u	423n	517n	375.2423n	< MIN >

On/Off	Profile	Measurement	Original	Min	Max
✓	rf_amp-schematic1-ac.sim	max(db(v(load)))	9.4181	7.3142	11.3819
✓	rf_amp-schematic1-ac.sim	bandwidth(v(load),3)	150.5788meg	130.3443meg	174.8395meg
✓	rf_amp-schematic1-ac.sim	min(10*log10(v(inoise)*v(inoise)/8.28e-19))	4.1481	3.6360	4.7507
✓	rf_amp-schematic1-ac.sim	max(v(onoise))	4.3383n	3.5366n	5.2793n

Sorting data

➡ Double click on the **Linear** column header to sort the bar graph data in ascending order. Double click again to sort the data in descending order.

Selecting the measurement to view

➡ Select a measurement in the Specifications table.

The data in the Parameters table relates to the measurement you selected.

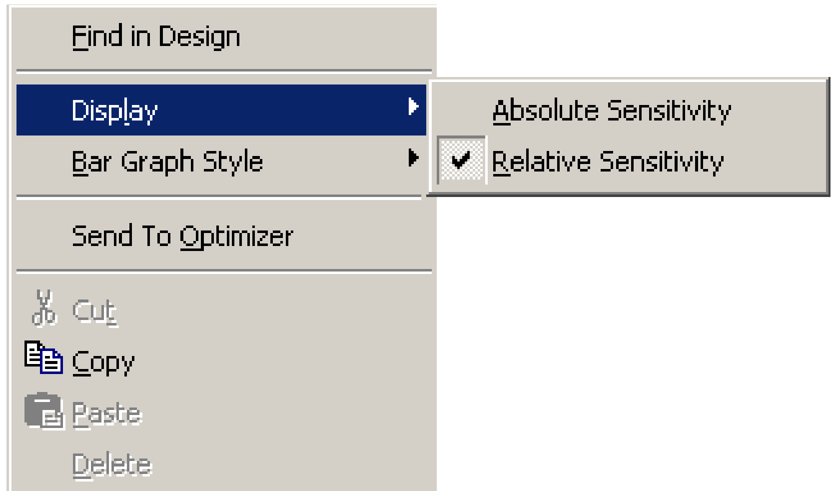
Table...	Column heading...	Means...
Parameters	Original	The nominal component parameter values used to calculate nominal measurement.
	@Min	The parameter value used to calculate the worst-case minimum measurement.
	@Max	The parameter value used to calculate the worst-case maximum measurement.
	absolute sensitivity	The change in the measurement value divided by a unit of change in the parameter value.
	relative sensitivity	The percent of change in a measurement value based on a one percent change in the parameter value.
Specifications	Original	The nominal value of the measurement using original component parameter values.
	Min	The worst-case minimum value for the measurement.
	Max	The worst-case maximum value for the measurement.

Note: To see all the parameter and measurement values used in Sensitivity calculations: from the View menu, select Log File.

Changing from Absolute to Relative sensitivity

1. Right click anywhere on the Parameters table.

A pop-up menu appears



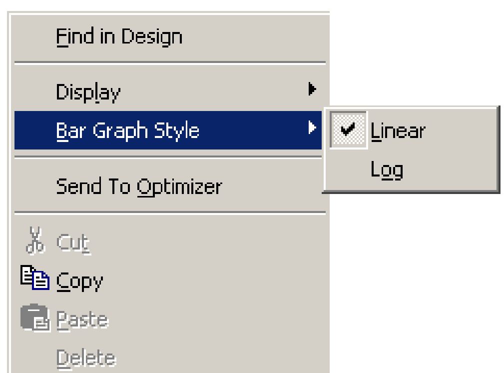
2. Select **Relative Sensitivity**.

Note: See Sensitivity calculations on page 81.

Changing the bar graph to linear view

1. Right click anywhere on the Parameters table.

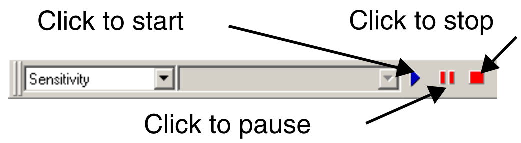
   A pop-up menu appears.



2. Select **Linear**.

Controlling Sensitivity

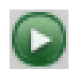
Pausing, stopping, and starting



Pausing and resuming

1. Click  on the top toolbar.

   The analysis stops, available data is displayed, and the last completed run number appears in the output window.

2. Click the depressed  or  to resume calculations.

Stopping

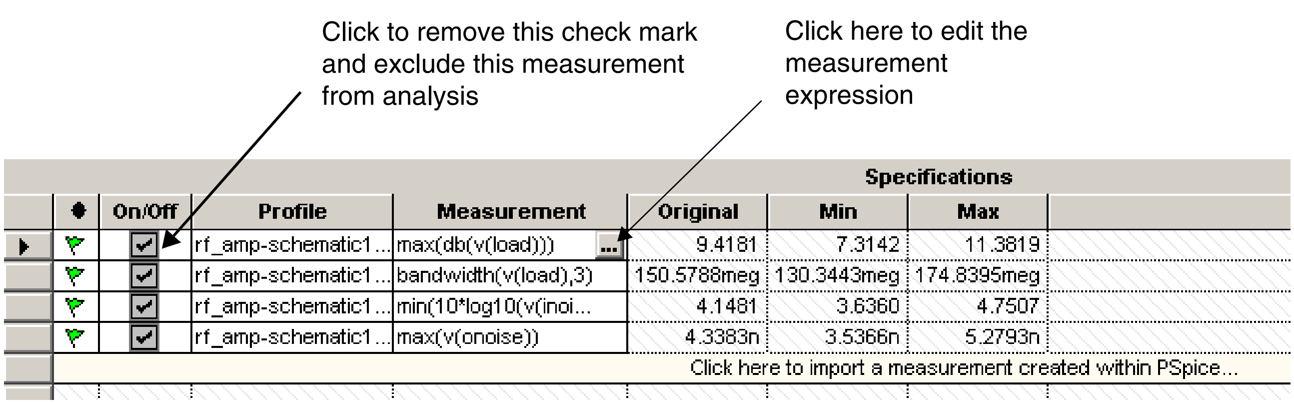
➡ Click  on the top toolbar.

If a Sensitivity analysis has been stopped, you cannot resume the analysis.

Starting

➡ Click  to start or resume.

Controlling Measurements



					Specifications		
		On/Off	Profile	Measurement	Original	Min	Max
▶		✔	rf_amp-schematic1...	max(db(v(load)))	9.4181	7.3142	11.3819
		✔	rf_amp-schematic1...	bandwidth(v(load),3)	150.5788meg	130.3443meg	174.8395meg
		✔	rf_amp-schematic1...	min(10*log10(v(inoi...	4.1481	3.6360	4.7507
		✔	rf_amp-schematic1...	max(v(onoise))	4.3383n	3.5366n	5.2793n
	Click here to import a measurement created within PSpice...						

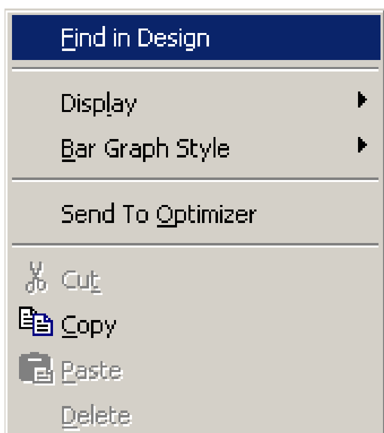
Adjusting component values

In the RF example, we will not change any component parameters.

With another example you may decide after reviewing sensitivity results that you want to change component values or tighten tolerances. You can use **Find in Design** from Advanced Analysis to return to your schematic editor and locate the components you would like to change.

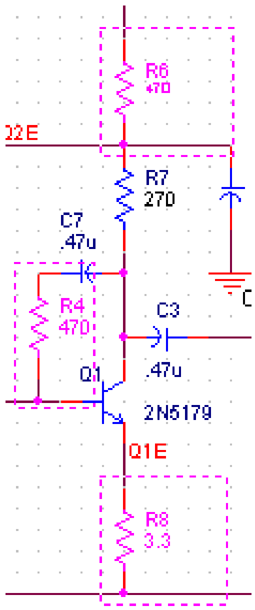
1. In the Parameters table, highlight the components you want to change.
2. Right click the selected components.

   A pop-up menu appears.



3. Left click on **Find in Design**.

The schematic editor appears with the components highlighted.



4. Change the parameter value in the schematic editor.
5. Rerun the PSpice simulation and check results.
6. Rerun Sensitivity.

Sending parameters to Optimizer

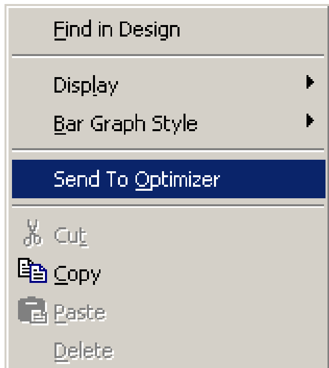
Review the results of the Sensitivity calculations. We need to use engineering judgment to select the sensitive components to optimize:

- We won't change R5 or R9 because they control the input and output impedances.
- We won't change R2 or R3 because they control transistor biasing.

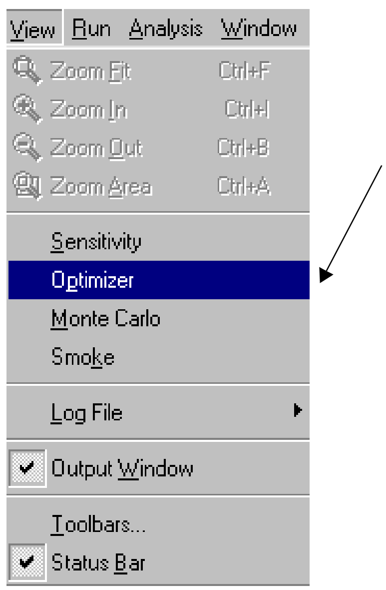
The linear bar graph at the Relative Sensitivity setting shows that R4, R6, and R8 are also critical parameters. We'll import these parameters and values to Optimizer.

1. In the Parameters table, hold down the Ctrl key and select R4, R6, and R8.
2. Right click the selected components.

A pop-up menu appears.



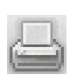
3. Select **Send to Optimizer**.
4. From the **View** menu, select **Optimizer**.



Select Optimizer view to switch to the Optimizer window and see the parameters you sent over from Sensitivity

Optimizer becomes the active window and your critical parameters are listed in the Optimizer Parameters table.

Printing results

➡ Click .

Or

From the File menu, select Print.

Saving results

➡ Click .

Or

From the File menu, select Save.

The final results will be saved in the Advanced Analysis profile (.aap).

For power users

Sensitivity calculations

Absolute sensitivity

Absolute sensitivity is the ratio of change in a measurement value to a one unit positive change in the parameter value.

For example: There may be a 0.1V change in voltage for a 1 Ohm change in resistance.

The formula for absolute sensitivity is:

$$[(M_s - M_n) / (P_n * S_v * Tol)]$$

Where:

M_s = the measurement from the sensitivity run for that parameter

M_n = the measurement from the nominal run

Tol = relative tolerance of the parameter

P_n = Nominal parameter value

S_v = Sensitivity Variation. (Default = 40%)

By default, the parameter value is varied within 40% of the set tolerance.

You can change this value to any desired percentage using the Profile settings dialog box.

1. From the Edit drop-down menu, choose Profile Settings.

2. In the Profile setting dialog box, select the Sensitivity tab.
3. In the Sensitivity Variation dialog box, specify the value by which you want to vary the parameter value.
4. Click OK to save your settings.

The values entered by you in the Profile Setting dialog box, are stored for the future use as well. Every time you load the project, old values are used for advanced analysis simulations.

Example

For example, if you specify the Sensitivity Variation as 10%, the parameter values will be varied within 10% of the tolerance value.

Consider that you want to test a resistor of 100k for sensitivity. The tolerance value attached to the resistor is 10%.

By default, for sensitivity calculations, the value of resistor will be varied from 96K to 104K. But if you change the default value of Sensitivity Variation to 10%, the resistor values will be varied from 99K to 101K for sensitivity calculations.

Relative sensitivity

Relative sensitivity is the percentage of change in a measurement based on a one percent positive change of a component's parameter value.

For example: For each 1 percent change in resistance, there may be 2 percent change in voltage.

The formula for relative sensitivity is:

$$[(Ms - Mn) / (S_v * Tol)]$$

Where:

M_s = the measurement from the sensitivity run for that parameter

M_n = the measurement from the nominal run

Tol = relative tolerance of the parameter

S_v = Sensitivity Variation. (Default = 40%)

Relative sensitivity calculations determine the measurement change between simulations with the component parameter first set at its original value and then changed by `Sv` percent of its positive tolerance. Linearity is assumed. This approach reduces numerical calculation errors related to small differences.

For example, assume that an analysis is run on a 100-ohm resistor which has a tolerance of 10 percent. The maximum value for the resistor would be 110 ohms. Assuming the default value of `Sv`, which is 40%, the analysis is run with the value of the resistor set to 104 ohms (40 percent of the 10 ohm tolerance) and a measurement value is obtained. Using that value as a base, Sensitivity assumes that the resistance change from 100 to 104 ohms is linear and calculates (interpolates) the measured value at 1 percent tolerance (101 ohms).

Worst-case minimums and maximums

For each measurement, Sensitivity sets all parameters to their tolerance limits in the direction that will increase the measurement value, runs a simulation, and records the measurement value. Sensitivity then sets the parameters to the opposite tolerance limits and gets the resulting value.

If worst-case measurement values are within acceptable limits for the design, the measurements can in most cases be ignored for the purpose of optimization.

Sensitivity assumes that the measured quantity varies monotonically throughout the range of tolerances. If not (if there is an inflection point in the curve of output function values), the tool does not detect it. Symptoms of this include a maximum worst-case value that is less than the original value, or a minimum value greater than the original value.

Sensitivity analysis runs

Sensitivity performs the following runs:

- A nominal run with all parameters set at original values
- The next run with one parameter varied within tolerance

  Values are obtained for each measurement. View the Log File for parameter values used in each measurement calculation.

- Subsequent runs with one parameter varied within tolerance
- A minimum worst-case run for each measurement
- A maximum worst-case run for each measurement

For our example circuit with 4 measurements and 12 parameters with tolerances, Sensitivity performs 21 runs.

There is one worst-case minimum and one worst-case maximum run per measurement

1 + 12 + (2 x 4) = 21 runs

The nominal run using the original parameter values

There are four measurements used in this example

There is one run for each parameter varied within tolerance. We use 12 parameters

To see the details of parameter and measurement calculations: from the **View** menu select **Log File**.

4

Optimizer

In this chapter

This chapter introduces you to Optimizer, its function, and the optimization process.

- Optimizer overview on page 85
- Terms you need to understand on page 87
- Optimizer procedure overview on page 94
- Example on page 119
- For Power Users on page 143

Optimizer overview

Note: Advanced Analysis Optimizer is available with the following products:

- PSpice[1] Advanced Optimizer Option
- PSpice Advanced Analysis
- PSpice Optimizer

Optimizer is a design tool for optimizing analog circuits and their behavior. It helps you modify and optimize analog designs to meet your performance goals.

1. Depending on the license available, you will access either PSpice or PSpice Simulator.

Optimizer fine tunes your designs faster and automatically than trial and error bench testing can. Use Optimizer to find the best component or system values for your specifications.

Advanced Analysis Optimizer can be used to optimize the designs that meet the following criteria:

- Design should simulate with PSpice.

  You can optimize a working circuit design that can be simulated using PSpice and the simulation results are as desired.

- Components in the design must have variable parameters, each of which relates to an intended performance goal.

Optimizer cannot be used to:

- Create a working design
- Optimize a digital design or a design in which the circuit has several states and small changes in the variable parameter values causes a change of state. For example, a flip-flop is on for some parameter value, and off for a slightly different value.

You can use the Advanced Analysis Optimizer to import legacy Optimizer projects.

Terms you need to understand

Optimization

Optimization is the process of fine-tuning a design by varying user-defined design parameters between successive simulations until performance comes close to (or exactly meets) the ideal performance.

The Advanced Analysis Optimizer solves four types of optimization problems as described in the table shown below.

Problem Type	Optimizer Action	Example
Unconstrained minimization	Reduces the value of a single goal	Minimize the propagation delay through a logic cell
Constrained minimization	Reduces the value of a single goal while satisfying one or more constraints	Minimize the propagation delay through a logic cell while keeping the power consumption of the cell less than a specified value
Unconstrained least squares1	Reduces the sum of the squares of the individual errors (difference between the ideal and the measured value) for a set of goals	Given a terminator design, minimize the sum of squares of the errors in output voltage and equivalent resistance
Constrained least squares	Reduces the sum of squares of the individual errors for a set of goals while satisfying one or more constraints	Minimize the sum of squares of the figures of merit for an amplifier design while keeping the open loop gain equal to a specified value

1 Use unconstrained least squares when fitting model parameters to a set of measurements, or when minimizing more than one goal.

Note: All four cases allow simple bound constraints; that is, lower and upper bounds on all of the parameters. Optimizer also handles nonlinear function as constraints.

Curve fitting

Curve fitting is a method of optimizing a model to a waveform. In this method, the specifications are represented using a collection of x-y points. These points describe the response of a system or a part of it.

Parameter

A parameter defines a property of the design for which the Optimizer attempts to determine the best value within specified limits.

A parameter can:

- Represent component values (such as resistance, R, for a resistor).
- Represent other component property values (such as slider settings in a potentiometer).
- Participate in expressions used to define component values or other component property values.
- Be a model parameter, such as IS for a diode.

  Example: A potentiometer part in a schematic uses the SET property to represent the slider position. You can assign a parameterized expression to this property to represent variable slider positions between 1 and 0. During optimization, the Optimizer varies the parameterized value of the SET property.

Specification

A specification describes the desired behavior of a design in terms of goals and constraints.

For example: For a given design, the gain shall be 20 dB ±1 dB; for a given design, the 3 dB bandwidth shall be 1 kHz; for a given design, the rise time must be less than 1 usec.

A design must *always* have at least one goal. You can have any number of goals and constraints in any combination, but it is recommended that the number of goals should be less. You can easily change a goal to a constraint and vice-versa.

The Advanced Analysis Optimizer can have two types of specifications: internal and external.

Internal specifications

An internal specification is composed of goals and constraints that are defined in terms of target values and ranges. These specifications are entered using the Standard tab of the Advanced Analysis Optimizer.

External specifications

An external specification is composed of measurement data defined in an external data file, which is read by the Advanced Analysis Optimizer. The external specifications are entered using the Curve Fit tab of the Advanced Analysis Optimizer.

Goal

A goal defines the performance level that the design *should* attempt to meet (for instance, minimum power consumption). A goal specification includes:

- The name of the goal.
- An acceptable range of values.
- A circuit file to simulate, a simulation profile.
- An expression or a measurement function for measuring performance.

Constraint

A constraint defines the performance level that the design *must* fulfill. For example, an expression indicating that the output voltage that must be greater than a specific level can be a constraint. The constraint specification includes:

- The name of the constraint.
- An acceptable range of values.
- A circuit file to simulate or a simulation profile.
- An expression or a measurement function for measuring performance.
- An allowed relationship between measured values and the target value, which can be one of the following:

<=	measured value must be less than or equal to the target value
=	measured value must equal the target value
>=	measured value must be greater than or equal to the target value

It is recommended that in a design, nonlinear functions of the parameters should be treated as constraints and not as goals.

For example: Bandwidth can vary as the square root of a bias current and as the reciprocal of a transistor dimension.

Performance

The performance of a design is a measure of how closely the calculated values of its specifications approach their target values for a given set of parameter values.

Each aspect of a design's performance is found by evaluating Optimizer expressionsL

In many cases (particularly if there are multiple conflicting specifications), it is possible that the Optimizer will not meet all of the goals and constraints. In these cases, optimum performance is the best *compromise* solution—that is, the solution that comes closest to satisfying each of the goals and constraints, even though it may not completely satisfy any single one.

Evaluation

An evaluation is an algorithm that computes a single numerical value, which is used as the measure of performance with respect to a design specification.

The Optimizer accepts evaluations in one of these three forms:

- Single-point PSpice A/D trace function
- PSpice A/D measurement expression
- Expression based on a combination of functions. For example `max(X)+max(Y)`

Given evaluation results, the Optimizer determines whether or not the changes in parameter values are improving performance, and determines how to select the parameters for the next iteration.

Trace function

A trace function defines how to evaluate a design characteristic when running a single-point analysis (such as a DC sweep with a fixed voltage input of 5 V). For example: V(out) to measure the output voltage; I(d1) to measure the current through a component.

> **Note:** Refer to the online PSpice A/D Reference Guide for the variable formats and mathematical functions you can use to specify a trace function.

Measurement Expression

A measurement expression defines how to evaluate a design characteristic when running any kind of analysis other than a single-point sweep analysis. A measurement expression computes a single number from a waveform. This can be done by finding a characteristic point (e.g., time of a zero-crossing) or by some other operation.

For example, you can use a measurement expressions to:

- Find maxima and minima in a trace.
- Find distance between two characteristic points (such as peaks).

- Measure slope of a line segment.
- Derive aspects of the circuit's performance which are mathematically described (such as 3 dB bandwidth, power consumption, and gain and phase margin).

To write effective measurement expressions, determine what you are attempting to measure, then define what is mathematically special about that point (or set of points).

Note: Be sure that the measurement expressions accurately measure what they are intended to measure. Optimization results highly depend on how well the measurement expressions behave. Discontinuities in measurement expressions (i.e., sudden jumps for small parameter changes) can cause the optimization process to fail.

Optimizer expression

An expression defines a design characteristic. The expression is composed of optimizer parameter values, constants, and the operators and functions shown in Table 4-1.

For example: To measure the sum of resistor values for two resistors with parameterized values named R1val and R2val, respectively, use the expression *R1val + R2val*.

Table 4-1 Valid Operators and Functions for Advanced Analysis Optimizer Expressions.

Operator	Meaning
+	addition
-	subtraction
*	multiplication
/	division
**	exponentiation
exp	e^x
log	ln(x)
log10	$\log_{10}(x)$
sin	sine

Table 4-1 Valid Operators and Functions for Advanced Analysis Optimizer Expressions., *continued*

Operator	Meaning
cos	cosine
tan	tangent
atan	arctangent

Note: Unlike trace functions and measurement expressions, Optimizer expressions are evaluated without using a simulation.

Derivative

A derivative can be defined as the rate of change of specification value with the change in parameter value.

Simulation profile

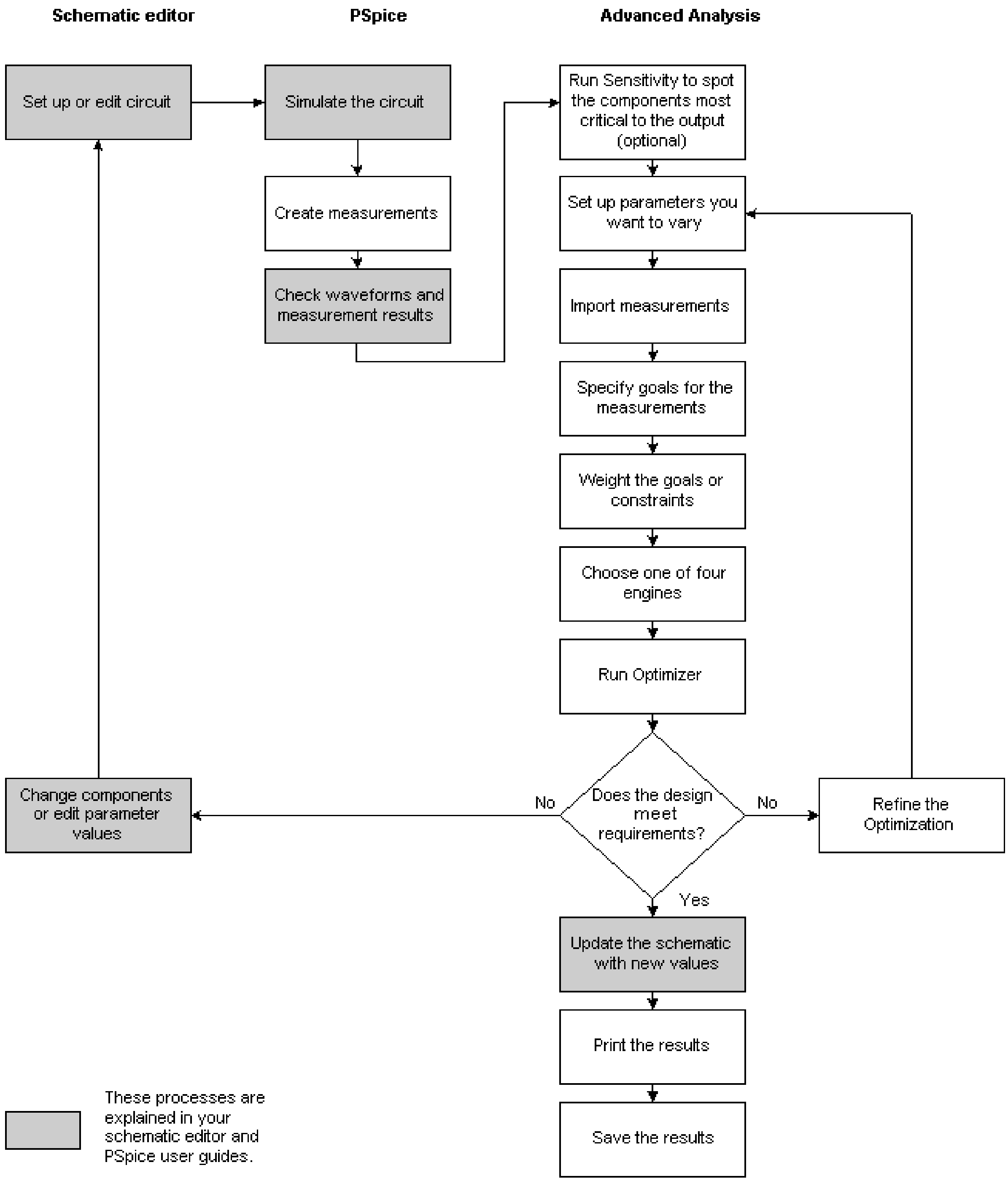
A simulation profile is used in the basic simulation flow. A simulation profile contains and saves the simulation settings for an analysis type so that it can be reused.

Advanced Analysis Profile

An advanced analysis profile contains and saves the advanced analyses (optimizer/sensitivity) settings so it can be reused.

Optimizer procedure overview

Workflow



To obtain meaningful optimization results, the Optimizer requires a problem description that consists of a circuit design, a list of optimization parameters, number of problem constraints derived from the design specification, and a set of performance goals to optimize.

To optimize a circuit,

1. Create or edit a circuit using a schematic tool.
2. Simulate and define circuit measurements.
3. Determine and set up optimization parameters that you want to vary during optimization.

   You can set up parameters from the Optimizer or on the schematic.

4. Specify the optimization specification.

   For Internal specifications use the standard tab.

   a. Define the goal for the circuit measurements.

      You can define goals such as rise time, phase margin, or entire response curves.

   b. Weight goals as needed.

   For External specifications use the Curve Fit tab.

   a. Specify the trace expression and the location of the reference file.

   b. For each of the Trace expression, specify the reference waveform, tolerance, and weight.

5. Select the Optimizer engine.

   You can use either the Modified Least Squares Quadratic or Random engines for optimization. The Discrete engine should be used after optimization to convert optimization parameters into discrete values only.

6. Start the analysis from the Optimizer.
7. Analyze or review the data in the Optimizer and refine the circuit design.
8. Save and print out your results.

Setting up in the circuit in the schematic editor

Start with a circuit in the design entry tool. The circuit simulations and measurements should be already defined.

The simulation can be a Time Domain (transient), a DC Sweep, or an AC Sweep/Noise analysis.

1. From your schematic editor, open your circuit.
2. Simulate the circuit.
3. Check your key waveforms in PSpice and make sure they are what you expect.

Test your measurements and make sure they have the results you expect.

For information on circuit layout, and simulation setup, see your schematic editor or PSpice user guides.

For information on setting up measurements, see “Measurement Expressions."

Setting up Optimizer in Advanced Analysis

Setting up the Optimizer consists of the following tasks:

- Opening Optimizer in Advanced Analysis
- Selecting an engine
- Defining Optimization parameters (Selecting Component Parameters)
- Setting up circuit measurements or specifications.
- Specifying optimization goals

Optimization goals are design specifications that you want to meet. Therefore, while defining optimization parameters, you need to determine parameters that affect your goals the most.

Opening Optimizer in Advanced Analysis

➡ From the **PSpice** menu in your schematic editor, select **Advanced Analysis / Optimizer.**

The Advanced Analysis Optimizer tool opens.

Selecting an engine

Optimizer in advanced analysis supports multiple engines. These are Modified LSQ (MLSQ), Random, and Discrete engines. In an optimization cycle, a combination of these engines is used.

Use these Optimizer engines for these reasons:

- Modified LSQ engine: to rapidly converge on an optimum solution.
- Random engine: to pick a starting point that avoids getting stuck in local minima when there is a problem converging.
- Discrete engine: to pick commercially available component values and run the simulation one more time with the selected commercial values.

The normal flow in which these engines are used is Random engine, followed by MLSQ engine, and finally the Discrete engine.

To know more about the Optimizer engines see Engine overview.

➡ From the top toolbar engine drop-down list, select one of the four optimizing engines.

Note: The Discrete engine is used at the end of the optimization cycle to round off component values to commercially available values.

Setting up component parameters

In this step, you identify the components or the parts in the circuit, whose parameter values you need to vary. Though the Optimizer in Advanced Analysis can support any number of components, it is recommended that the number of components with the variable parameter values should be kept to minimum.

You can specify parameters using:

- Schematic Editor
- Optimizer
- Sensitivity

Schematic Editor

1. In the schematic editor, select the component, whose parameter values you want to vary.
2. Select **PSpice > Advanced Analysis > Export Parameters to Optimizer**.

   The component gets added in the *Parameters* table.

Note: After you select the component, you can right-click and select **Export Parameters to Optimizer** from the pop-up menu. This command is enabled only if the selected component is based on PSpice-provided templates.

Optimizer

1. In the Parameters table in Advanced Analysis, click on the row containing the text “Click here to import.”

The **Parameters Selection** dialog box appears.

2. Highlight the components you want to vary and click **OK**.

   The components are now listed in the *Parameters* table.

Sensitivity

1. After you run the sensitivity analysis, select the most sensitive components and right-click.

2. From the pop-up menu, select **Send to Optimizer**.

   Selected components are listed in the *Parameters* table.

When you add a component to the *Parameters* table, the parameter name, the original value of the parameter, and the minimum and maximum values of the parameter are also listed in the *Parameters* table. The **Min** and **Max** values sets the range the engine will vary the component's parameters. These values are calculated by the Optimizer based on the original value. By default, **Min** value is one-tenth of the **Original** value and **Max** value is ten times the **Original** value.

You can use your engineering judgment to edit the Parameters table **Min** and **Max** values for the Optimization.

Caution

If you reimport any of the parameter that is already present in the *Parameters* ***table, the entries in the*** **Original*****,*** **Min*****, and*** **Max** ***columns are overwritten by the new values.***

Guidelines for selecting components

Optimization parameters need to carefully selected to ensure quicker optimizations and the best results.

- Vary your specification's most sensitive components. Run a sensitivity analysis to find them.
- Use good engineering judgment. Don't vary components whose values need to stay the same for successful circuit operation.

For example: if the input and output resistors need to be 50 ohms for impedance matching, do not choose those components to optimize.

- Vary just one component if varying other components can cause the same effect.

  For example: in an RC filter combination, both the resistor and capacitor affect the bandwidth. Selecting one parameter simplifies the problem. If your goal cannot be met with one parameter, you can add the second parameter.

Guidelines for setting up Parameters

- Make sure that ranges you specify take into account power dissipation and component cost.

  For example: a resistor with a small value (low ohms) could require a larger, more expensive power rating.

- Start with a small set of parameters (three or four) and add to the list during your optimization process.
- Aim for parameters with initial values near the range midpoints. Optimizer has more trouble finding solutions if parameter values are close to the endpoint of the ranges.
- Keep optimization parameter ranges within 1 or 2 orders of magnitude.

Setting up specifications

Using the Advanced Analysis Optimizer you can set two types of specifications:

- Measurement specifications - should be used in cases where circuit performance is measurable in terms of variable parameter values, such as gain margin for the circuit.
- Curve-fit specifications - should be used in cases where circuit output is a waveform, such as in wave shaping circuits.

Setting up measurement specifications

In the Advanced Analysis Optimizer, you can specify the measurement specification in the Standard tab.

1. In the Specifications table, click on the row containing the text "Click here to import..."

   The **Import Measurements** dialog box appears with measurements configured earlier in PSpice.

2. Highlight the measurements you want to vary and click **OK**.

   The components are now listed in the Specifications table.

3. Specify the acceptable minimum and maximum measurement values in the Specifications table **Min** and **Max** columns.

4. If you are using the Modified LSQ engine, mark the measurement as a goal or constraint by clicking in the **Type** column.

   The engine strives to get as close as possible to the goals while ensuring that the constraints are met.

5. Weigh the importance of the specification using the **Weight** column.

   Change the number in the weight column if you want to emphasize the importance of one specification with respect to another. Use a positive integer greater than or equal to one.

Note: Trial and error experimenting is usually the best way to select an appropriate weight. Pick one weight and check the Optimizer results on the Error Graph. If the results do not emphasize the weighted trace more than the rest of the traces on the graph, pick a higher weight and rerun the Optimization. Repeat until you get the desired results.

Guidelines for setting up measurement specifications

- Determine your requirements first, then how to measure them.
- Don't set conflicting goals.

  For example: Vout > 5 and Vout < 2 when the input is 3V.

- Make sure enough data points are generated around the points of measurements. Good resolution is required for consistent and accurate measurements.
- Simulate only what's needed to measure your goal.

  For example: for a high frequency filter, start your frequency sweep at 100 kHz instead of 1 Hz.

Setting up curve-fit specifications

Use curve fitting for following:

1. To optimize a model to one or more sets of data points. Using curve fitting, you can optimize multiple model parameters to match the actual device characteristic represented either waveforms from data sheets or measured data.
2. When the measurement expressions are specified as values at particular points, YatX().
3. To optimize circuits that need a precise AC or impulse response. For example, you can use curve fitting for optimizing signal shaping circuits, where the circuit waveform must match the reference waveform.

To use curve fitting for optimizing a design, you need to specify the following in the Curve Fit tab of the Advanced Analysis Optimizer:

1. A curve-fit specification

   You can either import a specification from an existing `.opt` file or can create a new specification.

   Creating a new specification includes specifying a trace expression, a reference file containing measured points and the corresponding measurement values, and a reference waveform.

   To know the details about the New Trace Expression dialog box, see *Advanced Analysis Online Help*.

   To see the detailed procedure for creating a new curve-fit specification, see "Creating a Curve-fit Specification" on page 103.

   To know more about the reference files, see Reference file on page 103.

2. List of parameters to be changed

   All the optimizable parameters in a circuit are listed in the property map file. This file is created when you netlist the design, and has information of each of the device used in the circuit design.

Creating a Curve-fit Specification

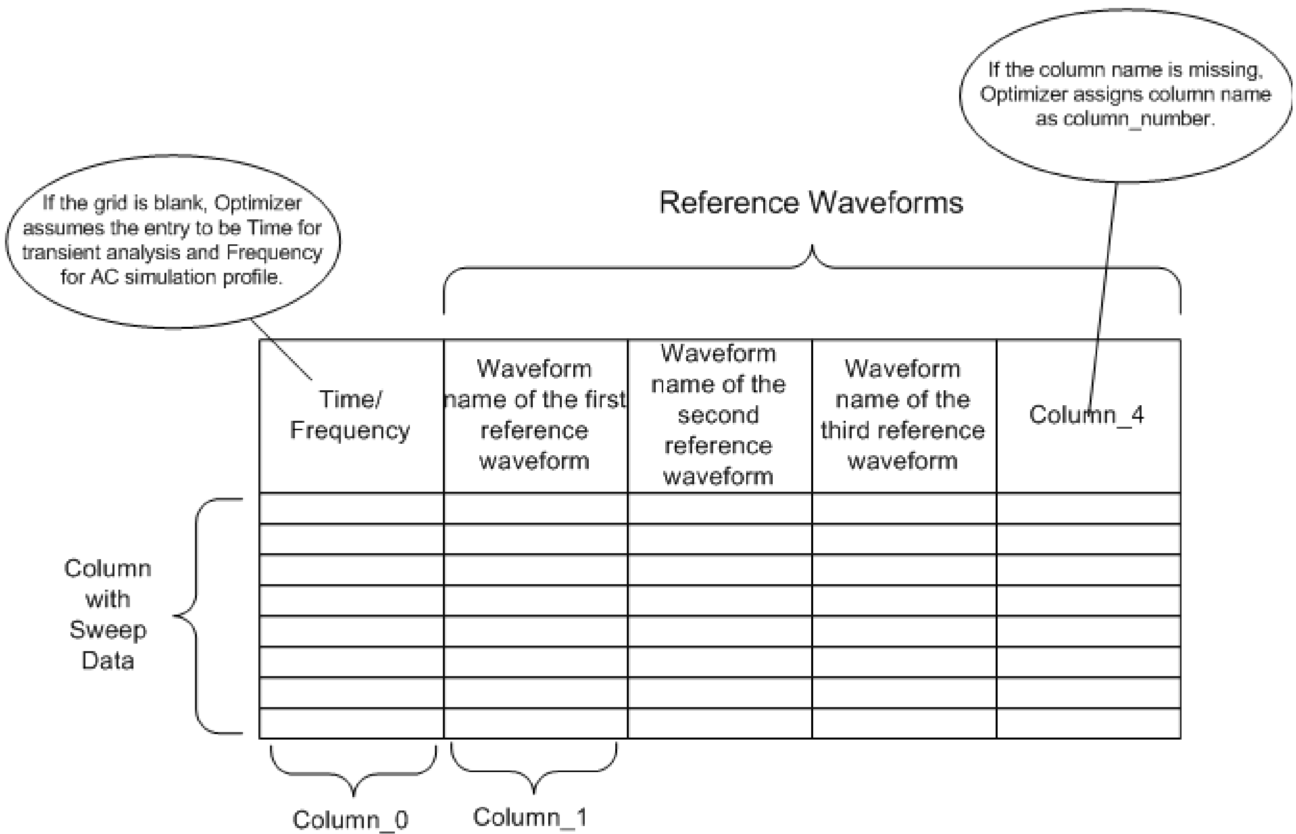
1. Specify the Trace Expression.
   a. In the Specifications area, click the row stating “Click here to enter a curve-fit specification”.
   b. In the New Trace Expression dialog box, select the simulation profile from the Profile drop-down list, and also specify the trace expression or the measurement for which you want to optimize the design.
2. Specify the reference file.
3. Specify the reference waveform. The Ref. Waveform drop-down list box lists all the reference waveforms present in the reference file that is specified in the previous step.
4. Specify the Weight for the specification.
5. Specify the relative tolerance.

Reference file

To be able to use curve fitting for optimizing your circuit, you must have a reference waveform. In Advanced Analysis Optimizer, the reference waveform is specified in form of multiple data points stored in a reference file. A reference file is a text file that contains the reference waveform with respect to a sweep in the tabular form with the data values separated by *white spaces*, *blanks*, *tabs* or *comma*.

An reference file has to have a minimum of two columns, one for the sweep data and one for the reference waveform. A reference file can have multiple columns. Each extra column represents a different reference waveform.

The format of a multiple column reference file is shown below:



A sample MDP file with one reference waveform is shown below.

```
    Time               V(D4:2)
                     0     1.35092732941686e-022
                2e-010    0.119616948068142
  2.17331331036985e-010    0.129942461848259
  2.51993993110955e-010    0.150499030947685
  3.21319317258894e-010     0.19108946621418
  4.59969965554774e-010    0.270239174365997
  7.37271262146533e-010    0.420916199684143
  1.14672723207623e-009    0.627191662788391
  1.52335408125073e-009    0.802674531936646
  2.27660777959973e-009     1.13146245479584
  3.77361568603665e-009     1.87895023822784
  6.76763149891049e-009      3.6644229888916
  1.27556631246582e-008     7.35082197189331
```

```
2.46214577833191e-008     14.6913433074951
 4.1200489727594e-008      24.834680557251
6.12008282819763e-008     36.7118606567383
8.12011668363586e-008     48.0069961547852
1.01201505390741e-007     58.5374412536621
1.21201843945123e-007     68.1351776123047
1.41202182499506e-007     76.6477890014648
1.61202521053888e-007     83.9403915405273
 1.8120285960827e-007     89.8975143432617
2.01203198162653e-007     94.4249801635742
2.21203536717035e-007     97.4511413574219
2.41203875271417e-007     98.9281539916992
2.61204213825799e-007      98.832633972168
2.81204552380182e-007     97.1660690307617
3.01204890934564e-007     93.9547653198242
```

First column of the reference file contains the sweep data, which is plotted on the X-axis. The first element in the header row indicates the type of analysis. For transient analysis the entry should be **Time**, for ac analysis it is **Freq** (frequency). For the DC-analysis there is no special entry. In case you leave the column header of the first column blank, the Advanced Analysis Optimizer assumes the entries in the sweep column to be time or frequency depending on whether the simulation profile is ac or transient, respectively.

The remaining entries in the header row indicate the names of the reference waveform in each column. These entries are displayed in the Reference Waveform drop-down list of the Curve Fit tab.

Creating Reference Files

You can create a reference file using one of the following.

- Manually

  Write the x,y points of the reference waveform in a text file. Save the text file with either `.mdp`, `.csv`, or `.txt` extension.

- Using the Export command in the PSpice File menu.

  a. Load a `.dat` file in PSpice.

  b. In the PSpice File menu, choose Export. Select Text (.txt file).

c. The Export Text Data dialog box appears.

The Output Variable to Export list displays the list of existing traces. You can add or delete traces from this list.

d. In the File name field, specify the name of the reference file and the location where the reference file is to be saved.

e. Click OK to generate the reference file.

To know the details about the Export Text Data dialog box, see *PSpice AD online help*.

The reference file generated using the Export menu command, has data values separated by tab.

Error Calculation

The error displayed in the Error column of the Curve Fit tab is influenced by the following factors:

- Relative Tolerance, specified by the user in the Tolerance column of the Curve Fit tab.
- Curve Fit Gear, specified by the user in the Optimizer tab of the Profile Settings dialog box. Curve fit gears are the methods used for error calculations.

  Note: The Profile Settings dialog box is displayed when you choose Profile settings from the Advanced Analysis Edit menu.

The error displayed is the difference between Root Mean Square Error (E_{rms}) and the tolerance specified by the user.

The Root Mean Square Error (E_{rms}) is calculated using the following formula:

$$E_{rms} = 100 \times \frac{\sqrt{\sum (R_i - S_i)^2}}{\sqrt{\sum (R_i)^2}}$$

Where

$$R_i = Y_{at}X(R,X_i)$$

V_i represents the reference value at the same sweep point.

and

$$S_i = Y_{at}X(S,X_i)$$

Y_i is the simulated data value.

X_i indicates the set of sweep values considered for the error calculation. The value of X_i depends on the gear type selected by the user.

Legacy gear

In this case, each point in the reference waveform is treated as an individual specification (goal) by the Optimizer. In this method, every data point is optimized. Therefore, the error at each data point should be zero. The Optimizer calculates error at each of the reference point and the final error is the RMS of the error at all reference points.

> **Note:** The legacy gear works only if the number of data points to be optimized is less than 250. If the number of data points is more than 250, next gear selected automatically.

Weighted reference gear

In this case, the Advanced Analysis Optimizer considers a union of the reference data points as well as simulation data points in the common interval of time or frequency values. A weight factor is multiplied to the error at each X_i. In this case, Xi will contain both, the reference file points and the simulation sweep points, but the error is calculated by multiplying the weight factor to the error at each point. Therefore, the error is:

$$E_{rms} = 100 \times \frac{\sqrt{\sum W_i \times (R_i - S_i)^2}}{\sqrt{\sum W_i \times (R_i)^2}}$$

Where W_i is the weight that is calculated using the following formula.

- For data points appearing only in the simulation data.

$$W_i = 1$$

- For data points appearing in the reference waveform.

$$W_i = \left\lceil \frac{b}{a} \right\rceil^2$$

Where

$$b = \text{sizeof}\{X_{ref+sim}\}$$

and

$$a = \text{sizeof}\{X_{ref}\}$$

The `sizeof` function returns the size of the vector.

$X_{ref + sim}$ indicate the union of the reference data points as well as simulation data points in a common interval.

Note: The weighted reference gear is same as Reference data points only gear for cases where $\frac{b}{a} \to \infty$.

Reference only gear

In this case, the Advanced Analysis Optimizer tries to fit in the simulation curve to the curve specified by the reference waveform, and the goal is to minimize the (RMS_{error}/RMS_{ref}) below the tolerance level specified by the user. The error is calculated only at the reference data points. Therefore, X_i will only contain the points on the reference waveform.

The error calculation formula is same as used in the Weighted reference gear, except that W_i is zero for all data points that are not on the reference waveform.

Simulation also gear

In this case, the Advanced Analysis Optimizer considers a union of the reference data points as well as simulation data points in the common interval of Time or frequency values.

Therefore, the error is calculated using the following formula:

$$E_{rms} = 100 \times \frac{\sqrt{\sum (R_i - S_i)^2}}{\sqrt{\sum (R_i)^2}}$$

Note: Notice that if W_i is equal to 1 for all X_i, then the Weighted reference gear is same as the Simulation and reference data points alike gear.

Example

Consider a situation in which the reference sweep or the value of X for the reference waveform, ranges from 30u to 110u. The value of X for the simulation waveform ranges from 0u to 100u. In this case, sweep value for error calculation (Xi) will range from 30u to 100u. This is so because the common interval between ranges 0-100u and 30u-110u is 30u to 100u. Lets assume that in the above-mentioned range, there are 100 reference data points and a total of 400 data points (simulation plus reference) on which error is being calculated. The Erms will be calculated for all the 400 data points.

For each value of Xi, Si, which is the simulated value at Xi, can either be an exact value specified in the simulation data (.dat) file, or it can be the interpolated value at Xi. Similarly, Ri, which is the reference value at Xi, can either be an exact value specified in the reference file, or it can be the interpolated value at Xi.

Thus, for the simulation also curve-fit error gear, Xi contains both the reference file points and the simulation sweep points (a total of 400 data points). The error between the Ri and Si is calculated at each of the 400 points and the RMS of this error waveform is calculated. The ratio of RMS of the error waveform and the RMS of the reference waveform R is calculated and normalized to the equivalent percentage.

For the weighted reference curve-fit error gear, the weighted RMS error is calculated at each of the 400 points (Xi). In this case there is one reference point for every four simulation data points (assuming linear distribution of reference and simulated data points). So each of the reference points is weighted by a scale factor of four (400/100).

Note: In all gears except the legacy gear, error is calculated for all the sweep points that are overlapping between the output wave form and the reference waveform.

Using curve fitting to optimize a design

1. Open a project and simulate it.

   Verify that circuit is complete and is working fine.

2. Invoke Advanced Analysis Optimizer, select the Curve Fit tab.

3. Create a curve-fit specification.
   Specify the following:

   a. Trace Expression

      Select a simulation profile and add a trace expression.

   b. Name and location of the Reference file

   c. Reference waveform as specified in the reference file.

   d. Tolerance

   e. Weight

4. Select the optimizable parameters.

   For each parameter, the original value, the min value (original value/10), and the max value (original value*10) displays automatically. You can change the min-max range as per the requirement.

5. Specify the method for error calculation.

   a. From the *Edit* menu, choose *Profile Settings*.

   b. From the *Curve-Fit Error* drop-down list in the *Optimizer* tab of the *Profile Settings* dialog box, select the method to be used for the error calculation.

      To know more about error calculation methods, see Error Calculation on page 106.

6. Specify whether or not you want to store simulation data.

a. In the *Profile Settings* dialog box, select the Simulation tab.

b. From the Optimizer drop-down list, select *Save All Runs*, if you want the simulation data to be stored, and select *Save None* if you do not want the simulation data to be stored.

7. Select an engine and start the Advanced Analysis Optimizer.

Running Optimizer

Starting a run

➡ Click  on the top toolbar.

The optimization analysis begins. The messages in the output window tell you the status of the analysis.

A nominal run is made with the original component parameter values.

As the optimization proceeds, the Error Graph shows a plot with an error trace for each measurement. Data in the Parameters and Specifications tables is updated.

Displaying run data

➡ Place your cursor anywhere in the Error Graph to navigate the historical run data.

The Parameters and Specifications tables display the corresponding data calculated during that run. The optimization engine used for each run is displayed in the Optimization Engine drop-down list box. Though the engine name is displayed, the list box is disabled indicating that you can only view the engine used for the optimizer run selected in the Error Graph.

Note: The Advanced Analysis Optimizer saves only the engine name associated with the simulation run. Engine settings are not saved.

Clearing the Error Graph history

Selecting the Clear error graph history, retains the value of parameters at the last run. Simulation information for all previous simulation runs is deleted.

For example, if the Optimizer has information stored for N number of simulation runs then select Clear Error graph history will delete all the simulation information from 0 to N-1 runs. The values in the current column of the Parameters window are used as the starting point for the next simulation run.

To get back the original parameter values, you need to delete all parameters and import again.

➡ Right click on the Error Graph and select **Clear History** from the pop-up menu.

This removes all historical data and restores the current parameter values to last parameter value.

Controlling optimization

You can stop an analysis to explore optimization trends in the Error Graph, reset goals when results are not what you expected, or change engines.

Pausing, stopping and starting

➡ To start or continue, click on the top toolbar.

➡ To pause, click on the top toolbar.

The analysis pauses at an interruptible point and displays the current data.

➡ To stop, click on the top toolbar.

Note: Starting after pause or stop resumes the analysis from where you left off.

Controlling component parameters

The range that Optimizer varies a component's parameter is controlled by the Max and Min values.

Default component values are supplied. For resistors, capacitors, and inductors the default range is one decade in either direction.

For more efficient optimization, tighten up the range between the Min and Max values.

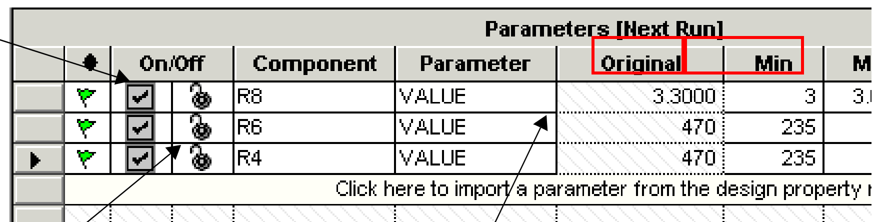
- To change the minimum or maximum value a parameter is varied: click in the **Min** or **Max** column in the Parameters table and type in the change.

- To use the original parameter value (with no change) during the next optimizing run: click in the Parameters table to toggle the check mark off.
- To lock in the current value (with no change) of a parameter for the next optimizing run: click on the lock icon in the Parameters table to toggle the lock closed .

  Note: If you cannot edit a value, and this is not the first run, you may be viewing historical data. To return to current data, click to the right of the horizontal arrow in the Error Graph.

Click to remove the check mark, which tells Optimizer to use the Original value without variation during the next optimizing run.

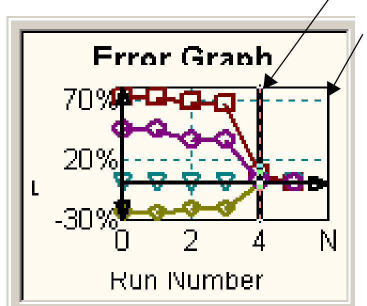
Click to lock in the current value without variation during the next optimizing run.



Click a Min or Max value to type in a change.

Default component values are supplied.
For resistors, capacitors, and inductors the default range is one decade in either direction.

Note:
If you can't edit a value, you might be viewing the historical data (if you have already run an optimization).



Click here to make changes which will affect the next run.

Controlling measurement specifications

Cells with cross-hatched backgrounds are read-only and cannot be edited.

- To exclude a measurement from the next optimization run, click the ☑ in the Specifications table, which removes the check mark.
- To hide a measurement's trace on the Error Graph, click the graph symbol icon ( ) in the Specifications table, which toggles the symbol off.
- To edit a measurement, click on the measurement you want to edit, then click on ... .
- To add a new measurement, click on the row that reads "Click here to import a measurement..."

  Note: For instructions on setting up new measurements, see "Procedure for creating measurement expressions" on page 258.
- To export a new measurement to Optimizer or Monte Carlo, select the measurement and right click on the row containing the text "Click here to import a measurement created within PSpice."

  Select **Send To** from the pop-up menu.

The example for this topic comes with measurements already set up in PSpice.

Copying History to Next Run

During optimization, you might want to modify an Optimizer run by copying parameter values from a previous optimization run into the current run database. You can then modify optimization specifications or engine settings, and run the Optimizer again to see the effects of varying certain parameters.

The Copy History to Next Run command allows you to copy the parameter values of the selected run to the last run which is also the starting point for the next simulation run.

Caution

Using Copy History To Next Run, you can only copy the parameter values of the selected run. The specifications, engine, and engine settings are not copied.

Use the following procedure to copy history.

1. In the Error Graph, select a run that you want to copy.

   The history marker appears positioned on the selected run.

2. Right-click on the Error Graph.

3. Select **Copy History To Next Run** from the pop-up menu.

   The parameters values are copied from the current marker run, for example, Run 1 to the end run.

Note: The **Copy History To Next Run** command is available only when you stop the Optimizer. Selecting **Pause** does not enable this menu command.

Consider a case where during optimization, parameter values do not converge after a particular point. In such cases, you can stop the Optimizer, copy the parameter values to the last run, select a different Optimizer engine and run the optimizer again.

Assigning available values with the Discrete engine

The Discrete engine is used at the end of the optimization cycle to round off components to commercially available values.

1. From the top toolbar engine field, select **Discrete** from the drop-down list.

   A new column named **Discrete Table** appears in the Parameters table.

2. For each row in the Parameters table that contains an RLC component, click in the **Discrete Table** column cell.

   An arrow appears, indicating a drop-down list of discrete values tables.

3. Select from the list of discrete values tables.

   A discrete values table is a list of components with commercially available numerical values. These tables are available from manufacturers, and several tables are provided with Advanced Analysis.

4. Click .

   The Discrete engine runs.

   The Discrete engine first finds the nearest commercially available component value in the selected discrete values table.

   Next, the engine reruns the simulation with the new parameter values and displays the measurement results.

   At completion, the **Current** column in the Parameters table is filled with the new values.

5. Return to your schematic editor and put in the new values.

   See Finding components in your schematic editor.

6. While you are still in your schematic editor, rerun the simulation.

   Check your waveforms and measurements in PSpice and make sure they are what you expect.

Finding components in your schematic editor

You can use the **Find in Design** feature to return to your schematic editor and locate the components you would like to change.

1. In the Parameters table, highlight the components you want to change.
2. With the components selected, right click the mouse button.

   A pop-up menu appears.
3. Select **Find in Design.**

   The schematic editor appears with the components highlighted.

Saving results

➡ Click .

Or

From the File menu, select Save.

The final results will be saved in the Advanced Analysis profile (.aap).

Examining a Run in PSpice

During the optimization process, one or more optimizer runs can fail. To investigate optimization failures,

■ Select **Analysis > Optimizer > Troubleshoot in PSpice**.

The simulation profile associated with the selected measurement opens in PSpice. PSpice then automatically opens the waveform viewer and shows a comparison of the last Optimizer simulation to a nominal PSpice simulation. PSpice lists results for both runs in the Measurement spreadsheet for easy comparison.

Example

This section, covers two design examples. The first example domesticates optimizing a design using measurement specifications. The second design example covers optimizing a design using curve-fit specifications.

Optimizing a design using measurement specifications

This example uses the tutorial version of RFAmp located at:

`<target_directory>\PSpice\tutorial\capture\pspiceaa\rfamp`

`<target_directory>\PSpice\tutorial\concept\pspiceaa\rfamp`

The circuit is an RF amplifier with 50-ohm source and load impedances. It includes the circuit schematic, PSpice simulation profiles, and measurements.

For a completed example see:

`<target_directory>\PSpice\Capture_Samples\AdvAnls\RFAmp directory`

For a completed example see:

`<target_directory>\PSpice\Concept_Samples\AdvAnls\RFAmp directory`

The example uses the goals and constraints features in the Modified LSQ engine. The engine strives to get as close as possible to the goals while ensuring that the constraints are met.

When designing an RF circuit, there is often a trade-off between the bandwidth response and the gain of the circuit. In this example we are willing to trade some gain and input and output noise to reach our bandwidth goal.

Optimizer goal:

- Increase bandwidth from 150 MHz to 200 MHz

Note: Enter meg or e6 for MHz when entering these values in the Specifications table.

Optimizer constraints:

- Gain of at least 5 dB (original value is 9.4 dB)
- Max noise figure of 5 (original value is 4.1)
- Max output noise of 3 nano volts per root Hz (original value is 4.3 nano volts per root Hz)

Setting up the circuit in the schematic editor

1. In your schematic editor, browse to the RFAmp tutorials directory.

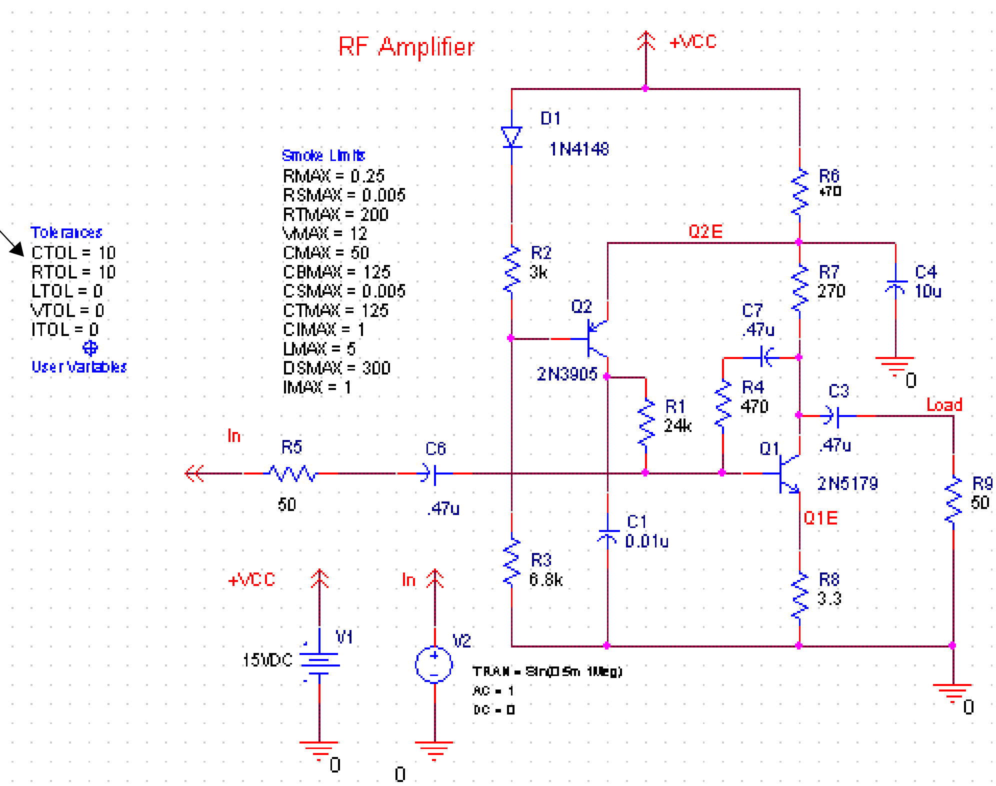
   ```
   <targe_directory>\PSpice\tutorial\Capture\pspiceaa\rfamp
   ```

   ```
   <target_directory>\PSpice\tutorial\Concept\pspiceaa\rfamp
   ```

2. Open the RFAmp project.

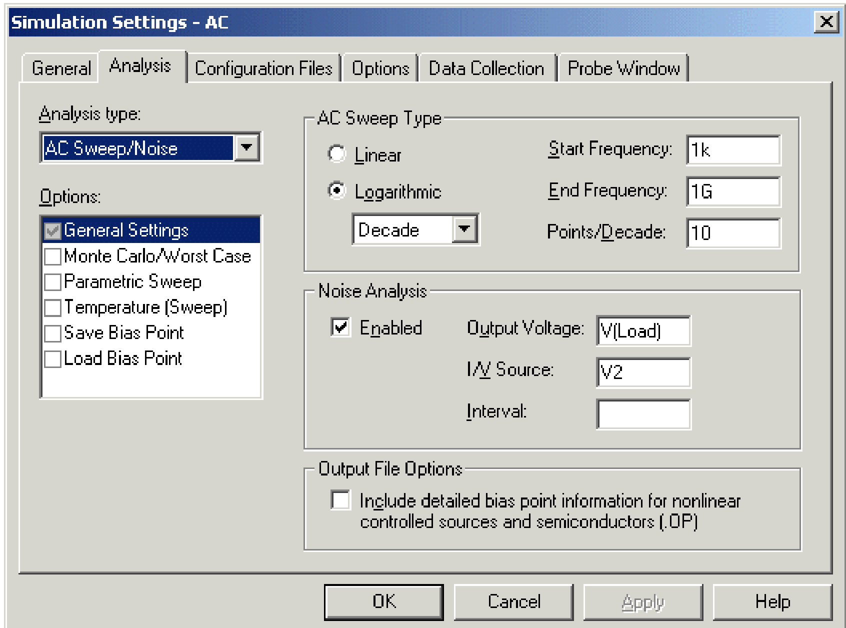
The RF amplifier circuit example

Assign global tolerances using this table



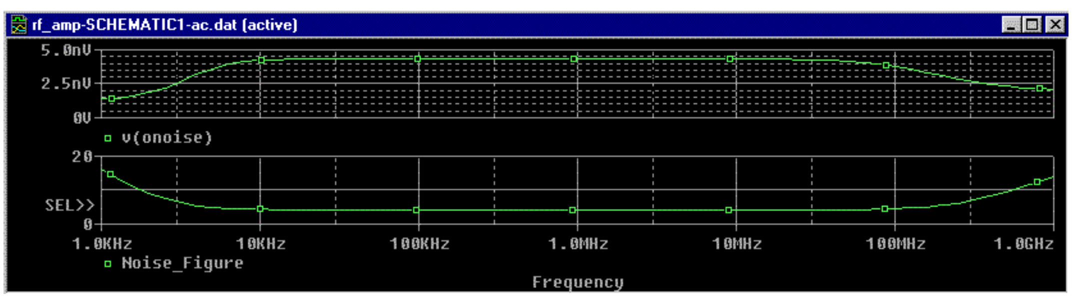
3. Select the SCHEMATIC1-AC simulation profile.

The AC simulation included in the RFAmp example



4. Click  to run the PSpice simulation.

5. Review the results.



The waveforms in PSpice are what we expected.

In PSpice, view measurement results

				Measurement Results
	Evaluate	Measurement	Value	
	☑	max(db(v(load)))	9.41807	
	☑	bandwidth(v(load),3)	150.57877meg	
▶	☑	min(10*log10(v(inoise)*v(inoise)/8.28...	4.14805	
	☑	max(v(onoise))	4.33832n	

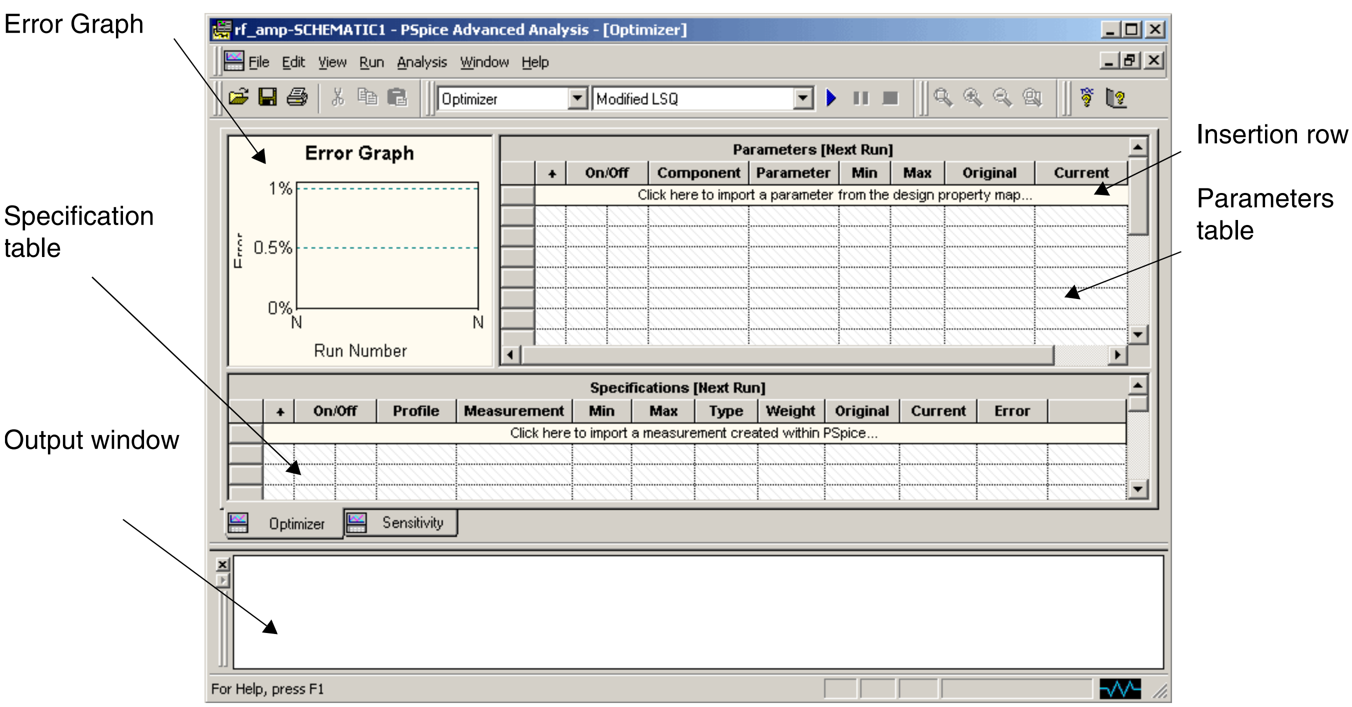
The measurements in PSpice give the results we expected.

Setting up Optimizer in Advanced Analysis

Opening Optimizer in Advanced Analysis

➡ From the **PSpice** menu in your schematic editor, select **Advanced Analysis / Optimizer**.

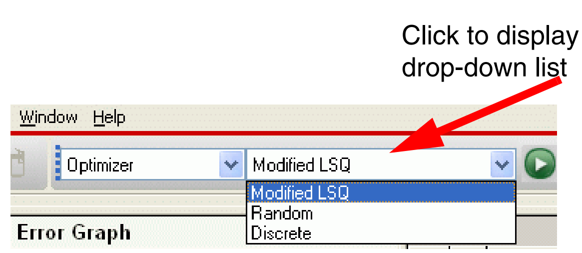
The Optimizer tool opens.



Selecting an engine

1. Click on the drop-down list to the right of the Optimizer tool name.

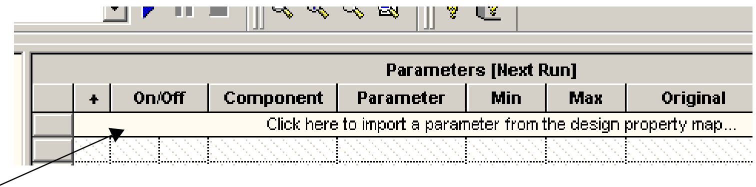
   A list of engines appears.



2. Select the Modified LSQ engine.

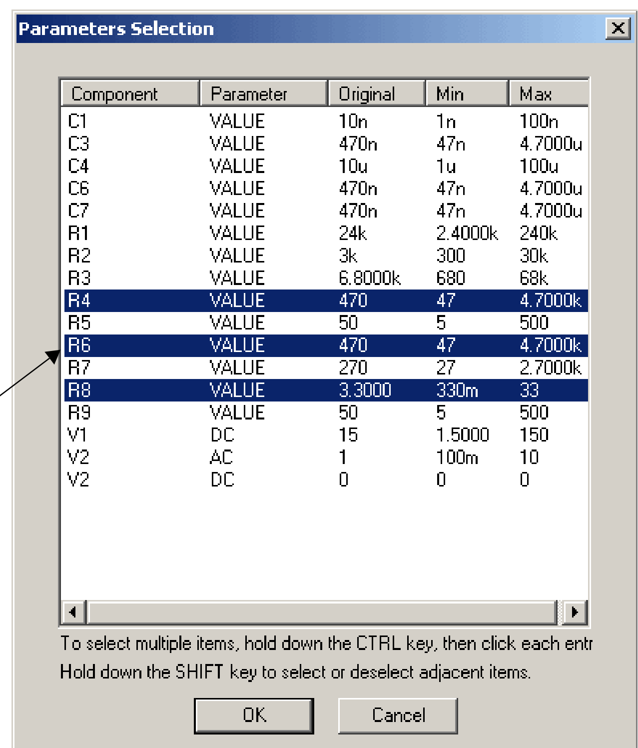
Setting up component parameters

1. In the Parameters table, click on the row containing the text "Click here to import..."



Click

The **Parameters Selection** dialog box appears.



Hold down the CTRL key and click to add multiple components

2. Highlight these components in the **Parameters Selection** dialog box:

- ❑ R6, the 470 ohm resistor
- ❑ R4, the 470 ohm resistor
- ❑ R8, the 3.3 ohm resistor

3. Click **OK**.

   The components are now listed in the Parameters table

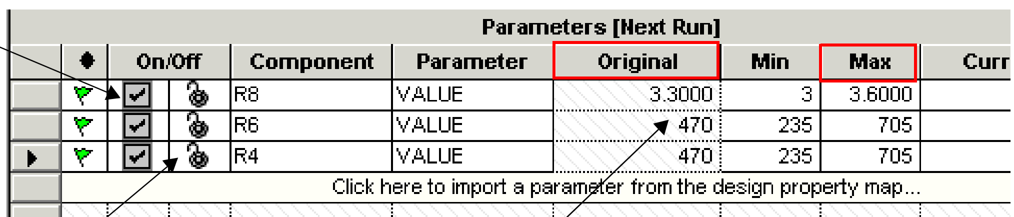
4. In the Parameters table **Min** and **Max** columns, make these edits:

   - R8: min value 3, max value 3.6
   - R6: min value 235, max value 705
   - R6: min value 235, max value 705

   This tightens the range the engine will vary the resistance of each resistor, for more efficient optimization.

Click to remove the check mark, which tells Optimizer to use the Original value without variation during the next optimizing run.

Click to lock in the current value without variation during the next optimizing run.



Click a Min or Max value to type in a change.

Default component values are supplied.
For resistors, capacitors, and inductors the default range is one decade in either direction.

Setting up measurement specifications

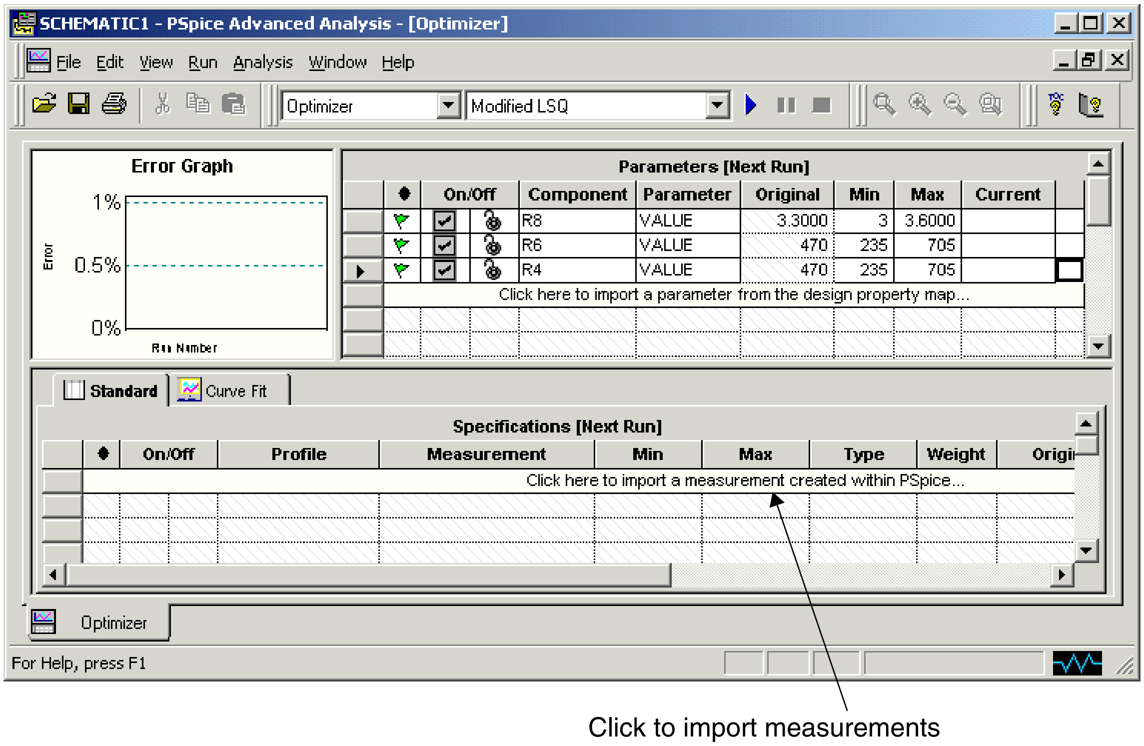
Measurements (set up earlier in PSpice) specify the circuit behavior we want to optimize. The measurement specifications set the min and max limits of acceptable behavior.

When using the Modified LSQ engine, you can also weigh the importance of the measurement specifications and mark them as constraints or goals.

The engine strives to get as close as possible to the goals while ensuring that the constraints are met.

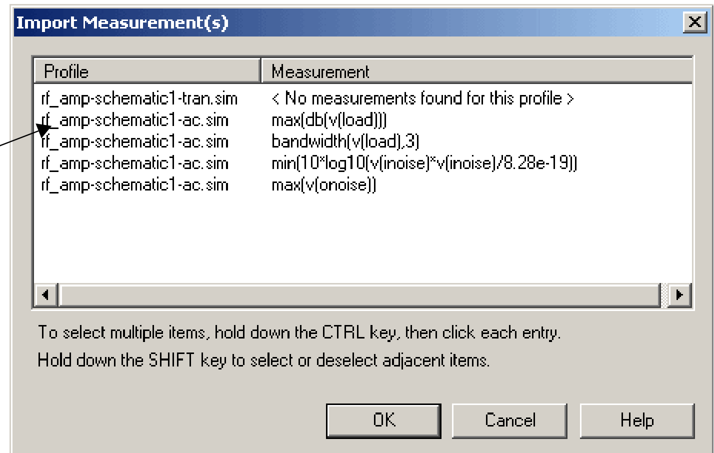
When there is more than one measurement specification, change the number in the weight column if you want to emphasize the importance of one specification with respect to another.

1. In the Specifications table, click on the row containing the text "Click here to import...."



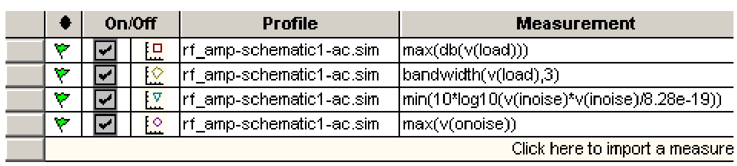
The **Import Measurements** dialog box appears with measurements configured earlier in PSpice.

Hold down the CTRL key and click to add multiple measurements



2. Select all the AC sim measurements and click **OK**.

   The measurements are now listed in the Specifications table.



On/Off	Profile	Measurement
✓	rf_amp-schematic1-ac.sim	max(db(v(load)))
✓	rf_amp-schematic1-ac.sim	bandwidth(v(load),3)
✓	rf_amp-schematic1-ac.sim	min(10*log10(v(inoise)*v(inoise)/8.28e-19))
✓	rf_amp-schematic1-ac.sim	max(v(onoise))
		Click here to import a measure

3. In the **Max(DB(V(Load)))** row of the Specifications table:
   - ❑ Min column: type in a minimum dB gain of **5**.
   - ❑ Max column: type in a maximum dB gain of **5.5**.
   - ❑ Type column: click in the cell and change to **Constraint**
   - ❑ Weight column: type in a weight of **20**
4. In the **Bandwidth(V(Load),3)** row:
   - ❑ Min column: type in a minimum bandwidth response of **200e6**
   - ❑ Max column: leave empty (unlimited)
   - ❑ Type column: leave as a **Goal**
   - ❑ Weight column: leave the weight as **1**
5. In the **Min (10*log10(v(in...** row:

- ❑ Min column: leave empty
- ❑ Max column: type in a maximum noise figure of **5**
- ❑ Type column: click in the cell and change to **Constraint**
- ❑ Weight column: leave the weight as **1**

6. In the **Max(V(onoise))** row:
   - ❑ Min column: leave empty
   - ❑ Max column: type in a maximum noise gain of **3n**
   - ❑ Type column: click in the cell and change to **Constraint**
   - ❑ Weight column: type in a weight of **20**

Note: For information on numerical conventions, Numerical conventions on page 22.

Click a cell to get a drop-down list and select Goal

Specifications [Next Run]

	+	On/Off	Profile	Measurement	Min	Max	Type	Weight
▶		✔	rf_amp-schematic1 ...	max(db(v(load)))	5	5.5000	Constraint	20
		✔	rf_amp-schematic1 ...	bandwidth(v(load),3)	200000000		Goal	1
		✔	rf_amp-schematic1 ...	min(10*log10(v(inoise)*v(inoise)/8.28e-19))		5	Constraint	1
		✔	rf_amp-schematic1 ...	max(v(onoise))		3n	Constraint	20
	Click here to import a measurement created within PSpice...							

Click a cell to type in a value

Select number and edit

Tip

It is recommended that you complete the steps for setting up component parameters and measurement specifications. In case you choose not to perform the steps, you can use the `SCHEMATIC1_complete.aap` file located at `..\tools\pspice\tutorial\capture\pspiceaa\rfamp\rf_amp-PSpiceFiles\SCHEMATIC` `for Capture and`

`..\tools\pspice\tutorial\concept\pspiceaa\rfamp\rf_amp-PSpiceFiles\SCHEMATIC` for Design Entry HDL

`1`. To use the aap file provided with the design example, rename `SCHEMATIC1_complete.aap to SCHEMATIC1.aap`.

Running Optimizer

Starting a run

➡ Click  on the top toolbar.

Click to start optimization

The optimization analysis begins. The messages in the output window tell you the status of the analysis.

A nominal run is made with the original component parameter values.

As the optimization proceeds, the Error Graph shows a plot with an error trace for each measurement.

Data in the Parameters and Specifications tables is updated.

Optimizer finds a solution after five runs.

Displaying run data

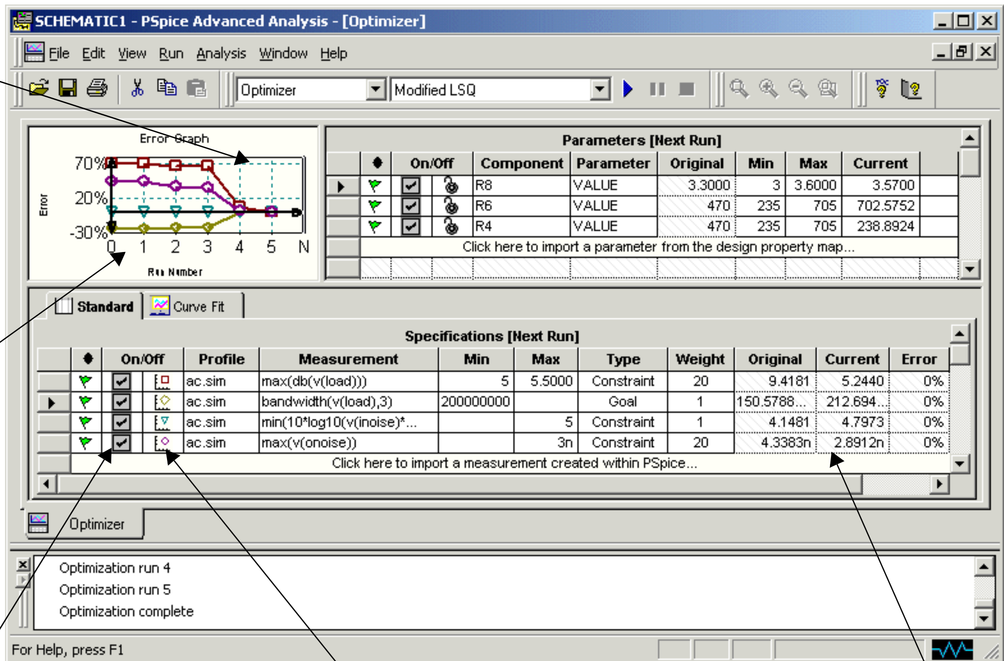
➡ Place your cursor anywhere in the Error Graph to navigate the historical run data.

The Parameters and Specifications tables display the corresponding data calculated during that run. Historical run data cannot be edited. It is read-only, as indicated by the cross-hatched background.

Click a run line to see data for that run The data in the Parameters and Specifications tables will change to reflect the values of that run

To clear the Error Graph and remove all historical data: right click on the Error Graph and select Clear History from the pop-up menu.

Click to remove the check mark, which excludes the measurement from the next optimization run



Click the graph symbol to toggle the symbol off, which hides the measurement's trace on the Error Graph

Cells with cross-hatched backgrounds are read-only and cannot be edited.

Controlling optimization

Pausing, stopping and starting

You can stop and resume an analysis to explore optimization trends in the Error Graph, to reset goals, or to change engines when results are not what you expected. The analysis will stop, saving the optimization data. You can also use pause and resume to accomplish the same thing.

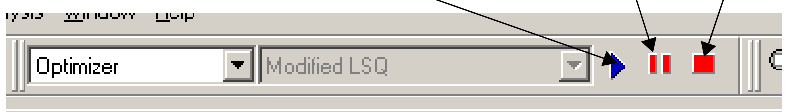
- To start or resume, click on the top toolbar.
- To pause, click on the top toolbar.

- To stop, click ■ on the top toolbar.

Click to start optimization

Click to pause optimization

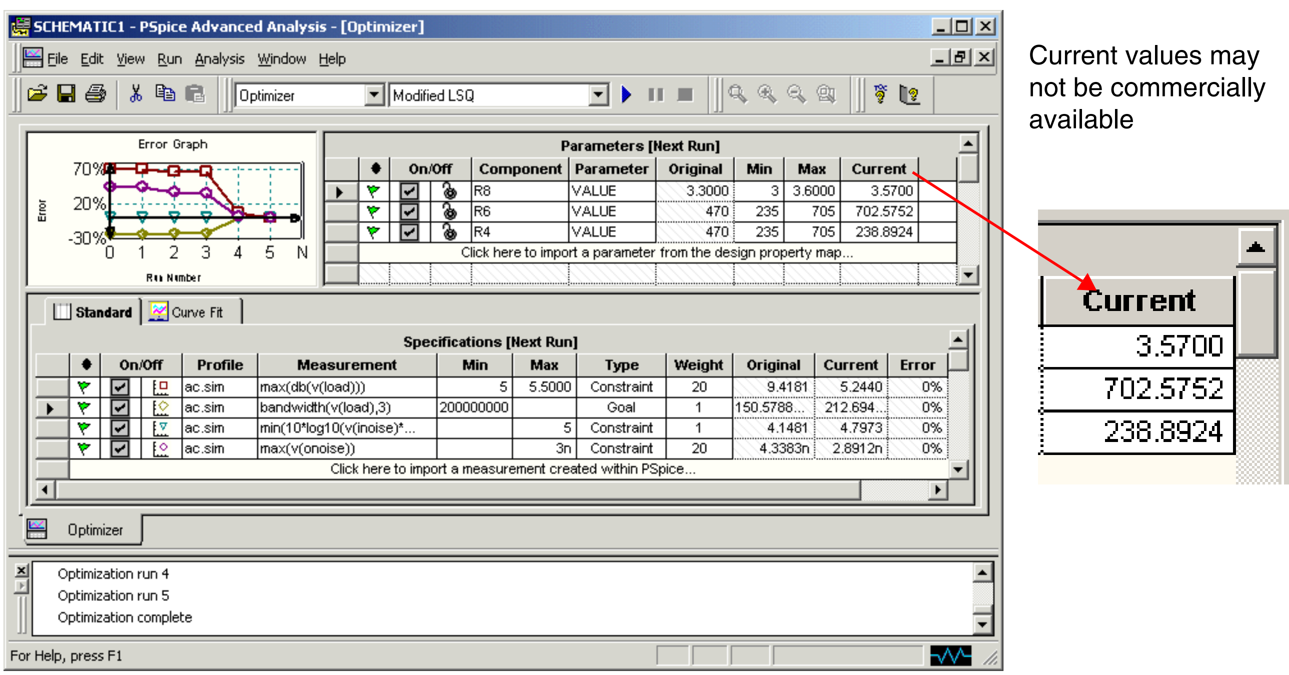
Click to stop optimization



Assigning available values with the Discrete engine

The Discrete engine is used at the end of the optimization cycle to round off component values to the closest values available commercially.

At the end of the example run, Optimization was successful for all the measurement goals and constraints. However, the new resistor values may not be commercially available values. You can find available values using the Discrete engine.



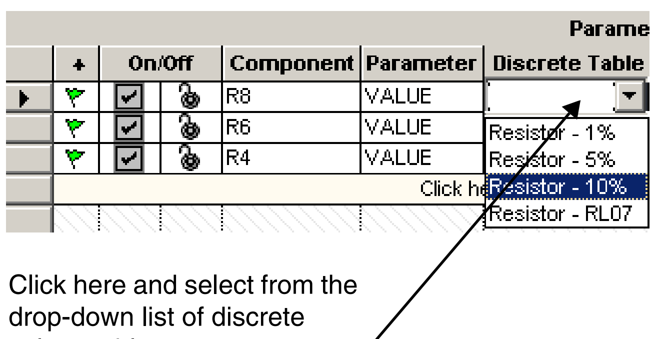
Current values may not be commercially available

1. From the top toolbar engine text box, select Discrete from the drop-down list.

   A new column named **Discrete Table** appears in the Parameters table. Discrete values tables for RLC components are provided with Advanced Analysis.

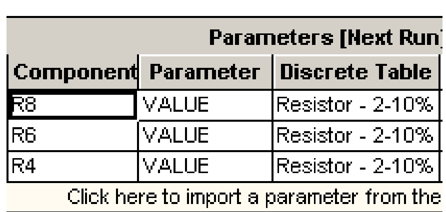
2. To select a discrete values table, click on any RLC component's **Discrete Table** column.

   You will get a drop-down list of commercially available values (discrete values tables) for that component.



Click here and select from the drop-down list of discrete values tables

3. Select the 10% discrete values table for resistor R8. Repeat these steps to select the same table for resistors R6 and R4.



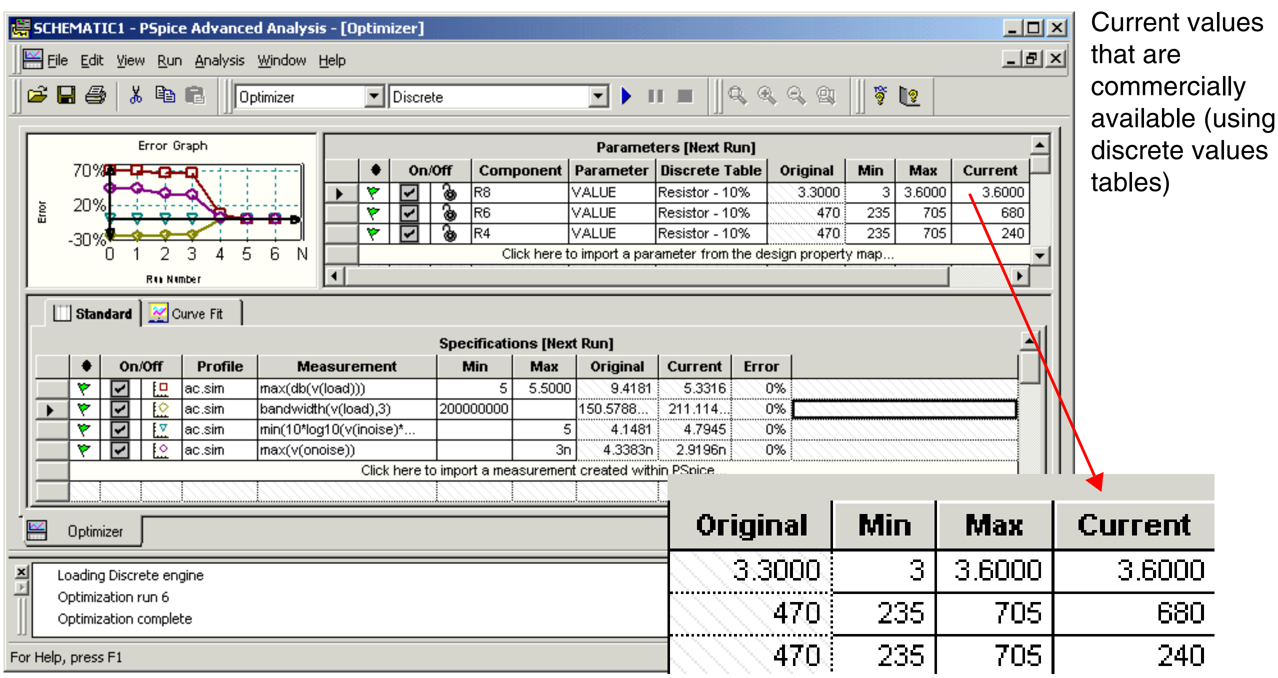
4. Click .

The Discrete engine runs.

First, the Discrete engine finds the nearest commercially available component.

Next, the engine reruns the simulation with the new parameter values and displays the measurement results.

At completion, the **Current** column in the **Parameters** table is filled with the new values.



5. Return to your schematic editor and change:

- ❑ R8 to 3.6 ohms
- ❑ R6 to 680 ohms
- ❑ R4 to 240 ohms

Note: You can use **Find in Design** to locate components in your schematic editor. See Finding components in your schematic editor.

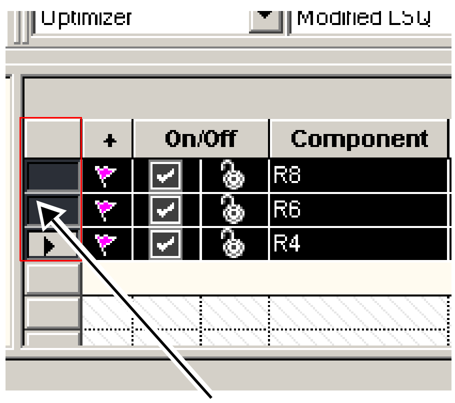
6. While you are still in your schematic editor, rerun the simulation titled AC.

   Check your waveforms and measurements in PSpice and make sure they are what you expect.

Finding components in your schematic editor

You can use **Find in Design** from Advanced Analysis to return to your schematic editor and locate the components you would like to change.

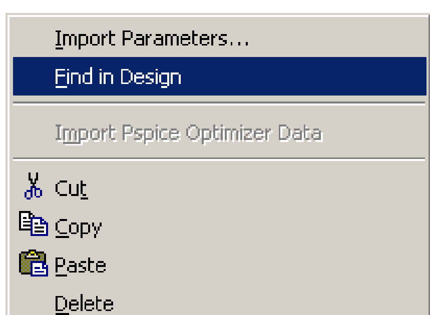
1. In the Parameters table, highlight the components you want to change.



Click here to select components (hold down shift key to select several)

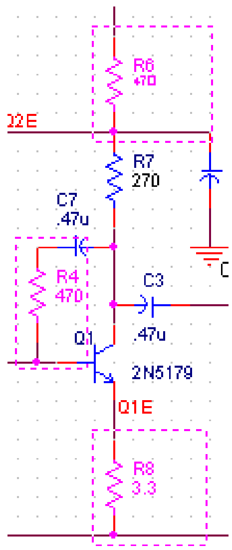
2. With the components selected, right click the mouse button.

   A pop-up menu appears.



3. Left click on **Find in Design.**

   The schematic editor appears with the components highlighted.

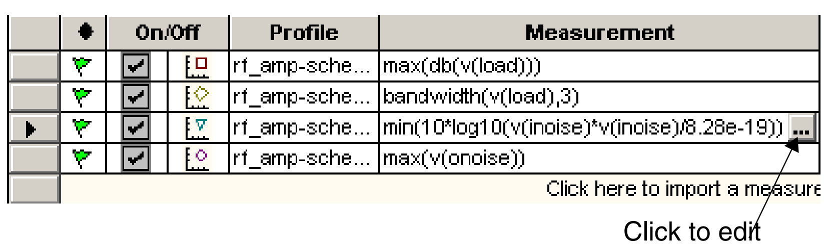


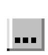
Editing a measurement within Advanced Analysis

At some point you may want edit a measurement. You can edit from the Specifications table, but any changes you make will not appear in measurements in the other Advanced Analysis tools or in PSpice.

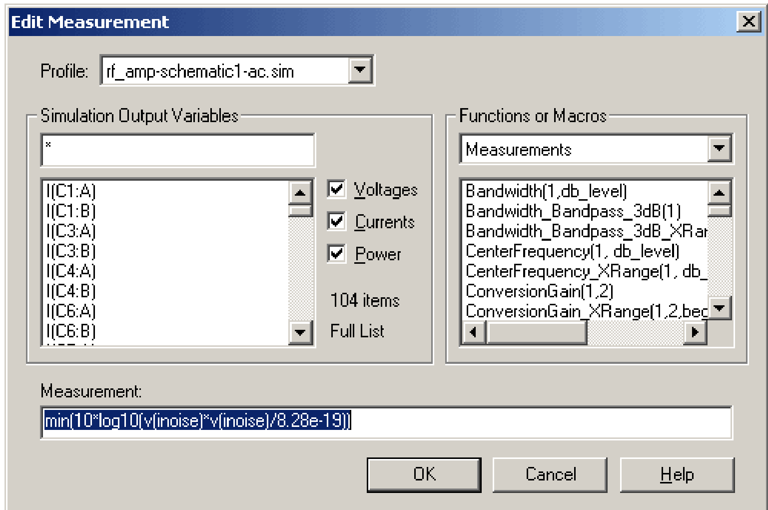
1. Click on the measurement you want to edit.

   A tiny box containing dots appears.



2. Click .

The Edit Measurement dialog box appears.



3. Make your edits.

It's a good idea to edit and run your measurement in PSpice and check its performance before running Optimizer.

4. Click **OK**.

Printing results

➡ Click .

Or

From the File menu, select Print.

Saving results

1. Click .

Or

From the File menu, select Save.

The final results will be saved in the Advanced Analysis profile (.aap).

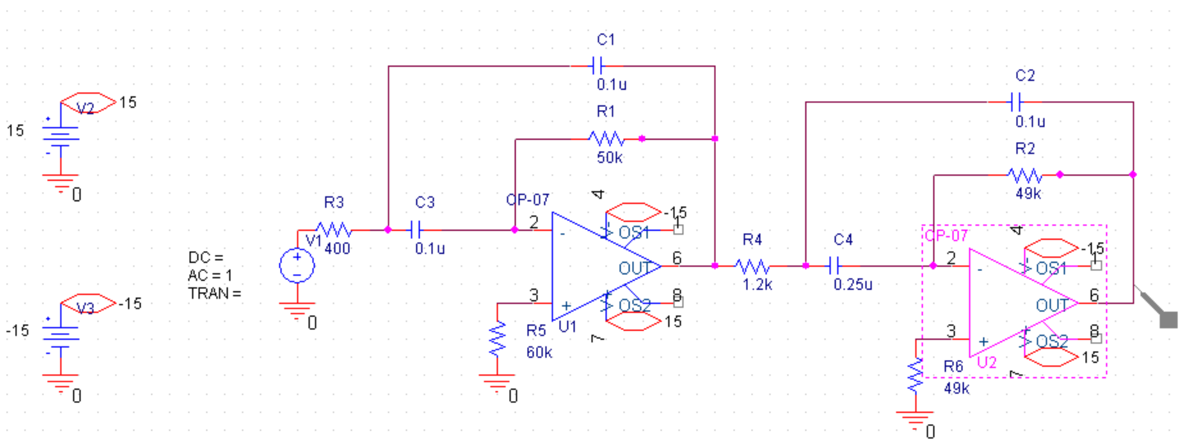
Optimizing a design using curve-fit specifications

The design example covered in this section, explains how you can use curve fitting to achieve desired response from a multiple feedback two pole active bandpass filter.

This bandpass filter uses two, 7-pin operational amplifiers. A plot window template marker, Bode Plot dB - dual Y axes is added at the output of the second operational amplifier (before R7). This marker is used to plot the magnitude and the phase gain of the output voltage.

The design example is available at
`..\tools\pspice\tutorial\capture\pspiceaa\bandpass`
for Capture or
`..\tools\pspice\tutorial\concept\pspiceaa\bandpass`
for Design Entry HDL

Figure 4-1 Bandpass Filter



1. Draw the circuit as shown in Figure 4-1.
2. Simulate the circuit.

   From the PSpice menu, choose Run.
3. The PSpice probe window appears displaying the simulation results. Two traces, one for phase gain of the output voltage and

another for the voltage gain(dB) of the output voltage are displayed.

We will now optimize the values of the component parameters in the circuit, such that the output waveform matches the waveform described in the reference file. For this design example, we will use `reference.txt` for specifying the reference waveform for `DB(V(`V_{out}`)` and `P(V(`V_{out}`)`.

Note: In a real life scenario, you will have to create a reference file containing the reference waveform, before you can use the curve fitting in Advanced Analysis Optimizer.

Opening Optimizer in Advanced Analysis

From the PSpice menu, choose Advanced Analysis Optimizer.

Selecting an engine

1. Click on the drop-down list to the right of the Optimizer tool name.
2. From the drop-down list, select the Modified LSQ engine.

Setting up component parameters

1. In the Parameters window, add the parameters that you want to optimize to obtain the desired output.

   Select the *Click here to import a parameter from the design property file* row.

2. In the Parameter Selection dialog box, select `C1`,`C2`,`C3`,`C4`,`R1`,`R2`,`R3`, and `R4`, and click OK.

   The selected components, their original values, and the min and max values that are calculated using the original values, appear in the Parameters window.

   For example, in the circuit, value of R4 is 1.2K. Therefore, the value displayed in the Original column against R4 is 1200. The min value displayed is 120 (1200/10) and the max value displayed is 12000 (1200*10).

3. In the Parameters tab, if you do not want the value of a particular parameter to change, you can do so by locking the parameter value. Lock the parameter values for R6 and R5.

4. You can also ignore some of the parameter values.

   Though we added the parameter R3, we will ignore it for this optimizer session. To do this, clear the check mark next to the message flag.

Setting up curve-fit specification

1. Select the *Curve Fit* tab in the Optimizer window.

2. In the *Curve Fit* tab, add specifications. Select the *Click here to enter a curve-fit specification* row.

3. In the *New Trace Expression* dialog box, first select P() from the list of Analog operators and Functions, and then select V(out) from the list of Simulation Output Variables.

   The Measurement text box should read *P(V(out))*.

4. Click *OK* to save the new trace expression.

5. In the Reference File text box, specify the location of `reference.txt`.

6. Click the *Ref.Waveform* list box. From the drop-down list that appears, select *PHASE*.

   Note: The entries in the drop-down list are the column headings in the reference file. If you open the reference file, `reference.txt,` you will see that *PHASE* is the heading of the second column and the third column has no heading. When the column headers are blank in the reference file, the reference waveform drop-down list displays entries such as, Column_2 and Column_3, instead of a name.

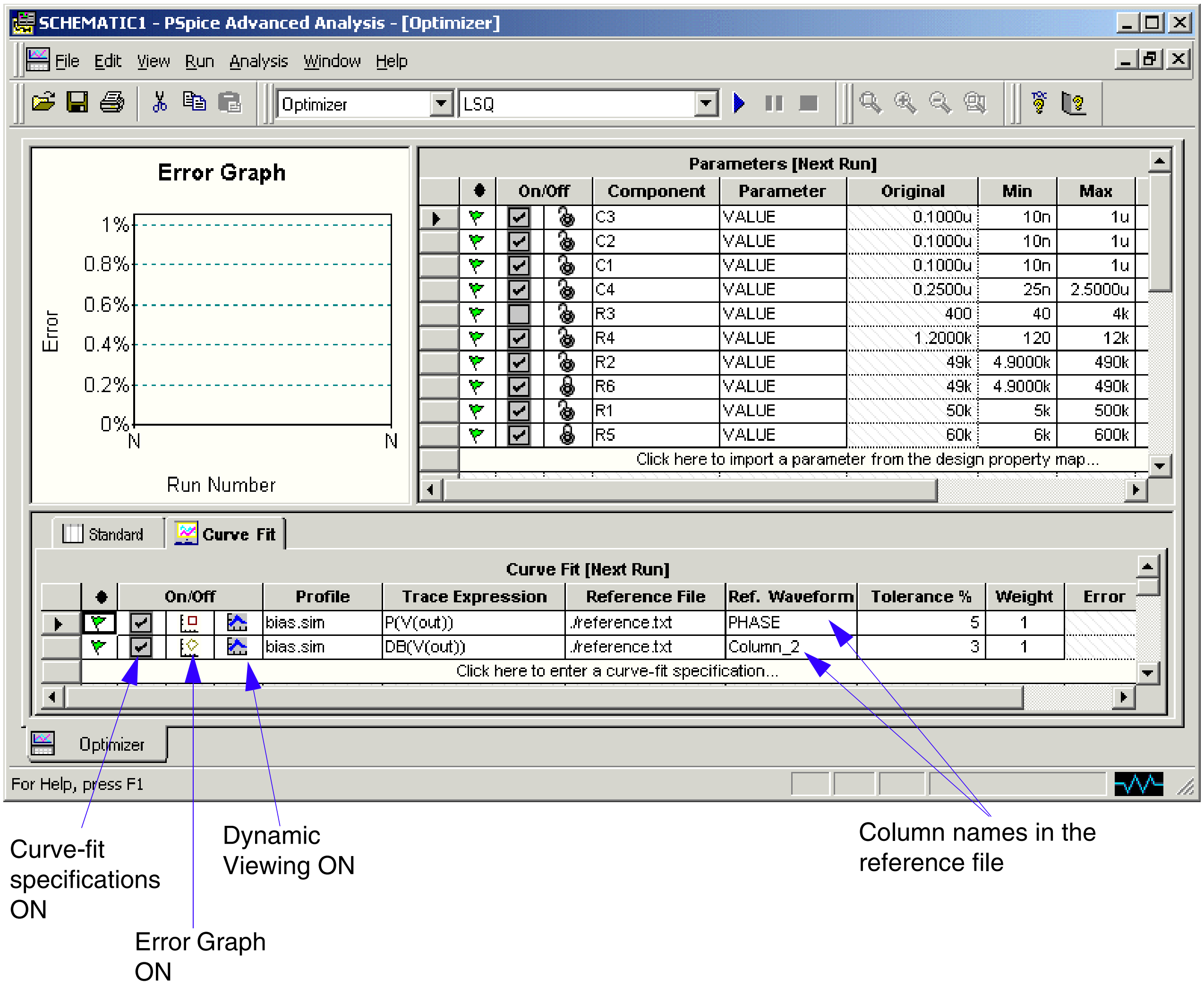
7. Specify the tolerance and weight at 5 and 1, respectively.

   This completes the process of creating a new curve-fit specification. In case you want to enable dynamic viewing of the output waveform, select the third field in the On/Off column.

8. Similarly, add another specification. Specify the trace expression as *DB(V(out))*, reference file as `reference.txt`, reference waveform as *Column_2*, tolerance as 3, and weight as 1.

9. Turn the dynamic viewing on.

The snapshot of the Optimizer, after you have modified the settings, is shown below:



10. In case you want that the simulation data should be available to you even after the optimization session is complete, you need to modify the Optimizer settings. From Advanced Analysis the Edit menu, choose Profile settings.

Tip

It is recommended that you complete the steps for setting up component parameters and curve-fit specifications. In case you choose not to perform the steps, you can use the `SCHEMATIC1_complete.aap` file located at `..\tools\pspice\tutorial\capture\pspiceaa\bandpass\bandpass-PSpiceFiles\SCHEMATIC1` or `..\tools\pspice\tutorial\concept\pspiceaa\bandpass\bandpass-PSpiceFiles\SCHEMATIC1`. To use the aap file provided with the design example, rename `SCHEMATIC1_complete.aap to SCHEMATIC1.aap`.

11. Select the *Simulation* tab in the *Profile Settings* dialog box, and ensure that Optimizer data collection is set to *Save All Runs*.

12. Run the Optimizer.

The PSpice UI comes up displayed the changes in the output waveform for each Optimizer run. The PSpice UI comes up only if you have turned the dynamic viewing on.

After the optimization is complete, you can view any of the Optimizer runs, provided you had selected the Save All Runs option in the Profile Settings dialog box.

Viewing an Optimizer run

1. Select run 4 in the Error Graph section.
2. Select the curve-fit specification for which you want to view the run. Select the first specification.
3. Right-click and select View[Run #4] in PSpice.

   The trace for the selected run opens in PSpice.

For Power Users

What are Discrete Tables?

After you have run Advanced Analysis optimizer and obtained the optimum values for your parts, it is possible that those values may not be commercially available. Optimizer has a Discrete engine feature that finds the closest available manufacturer's values for your parts. These values are discrete values and can be selected from a drop-down list of discrete values tables in Advanced Analysis Optimizer.

Discrete tables provided by the Advanced Analysis Optimizer are located at `<your_installation_dir>\tools\pspice\library\discretetables`. Under this directory, you find the following subdirectories, that contain the discrete value tables corresponding to each part.

- Capacitance
- Inductance
- Resistance

With Advanced Analysis, you get six discrete values tables. These tables are available on a global level. The range of values for each of the discrete value table is listed below.

Part	**Discrete table alias**	**Range of values**
Resistor	Resistor – 1%	10.0e0 to 20.0e5
Resistor	Resistor – 5%	10.0e-1 to 24.9e5
Resistor	Resistor – 10%	10.0e-1 to 27e5
Resistor	Resistor – RL07	51 to 150000
Capacitor	Capacitor	1.0e-12 to 1.0
Inductor	Inductor	3.9e-6 to 1.8e-2

Adding User-Defined Discrete Table

You can create your own discrete value table for components and variables that you want the Discrete Engine to read in your project directory where you run the Advanced Analysis Optimizer. Tasks to be completed for setting up a discrete value table for a user-defined variable are:

- Creating a new discrete value table
- Associating the table with the discrete engine
- Using the table

Creating a new discrete value table

1. Create a file called *xyz.table*.
2. Enter the table as shown.

```
START
1
1.2
1.4
....{fill in other values}
5.8
6.0
END
```

Associating the table with the discrete engine

After creating the table, the next step is to add the new discrete table to Advanced Analysis. You can create custom derate files at any location and then associate these with the Advanced Analysis discrete engine using the Advanced Analysis Optimizer Settings dialog box.

1. From the Advanced Analysis **Edit** menu, select **Profile Settings**.
2. Click the **Optimizer** tab.
3. Select **Discrete** from the **Engine** drop-down list.
4. Click the new file button in the **Discrete Files** text box, browse to the new table file you have created and select it.

5. Click in the **Discrete Table Alias** text box.

   Advanced Analysis places the file alias name in the text box.

6. In the **Part Type** text box, select the part for which the new discrete value table is created. In case the discrete value table is not for Resistor, inductor, or capacitor, select **Other** from the drop-down list.

7. Click **OK**.

   You can now use the information stored in the new table file.

Using the table

1. In the Advanced Analysis Optimizer view, select **Discrete** from the toolbar engine drop-down list.

   The Discrete Table column gets added in the Parameters table.

2. In the Discrete Table column, select the required table from the drop-down list.

Device-Level Parameters

What are device-level parameters?

The devices are constructed as parameterized models, that allow passing of parameters from the instance of the device.

During optimization, you can also vary the device-level parameters of a component.

To add device-level parameters of a component to the parameter table in the Advanced Analyses Optimizer, complete the following tasks:

- Add device-level parameters as instance properties.
- Export these properties to Advanced Analysis optimizer.

Adding Device-level parameters as instance properties

1. Select the component.

2. Select **PSpice > Advanced Analyses > Import Optimizable Parameters**.

   The **Import Optimizable Parameters** dialog box appears.

   Note: After you select the component, you can right-click and select Import Optimizable Parameters from the pop-up menu.

3. Select the parameters you want to vary and click **OK**.

   The parameter name and the default value is now displayed in the schematic editor.

4. Save the schematic.

Exporting Device-level parameters to Optimizer

1. Select the component.

2. Select **PSpice > Advanced Analyses > Export Parameters to Optimizer**.

   The component and parameters gets added in the *Parameters* table.

Note: This feature of exporting device-level parameters to Optimizer is available only for components based on PSpice-provided templates.

Optimizer log files

After every optimization run, Optimizer generates log files. This file contains information that can be used at instances where optimization failed to converge.

To view the optimizer log files choose **View > Log Files > Optimizer**.

The Optimizer log file opens in the text editor.

Engine Overview

Optimizer includes three engines:

- Modified LSQ engine

  The Modified LSQ engine uses both constrained and unconstrained minimization algorithms, which allow it to optimize goals subject to nonlinear constraints.

  When using the Modified LSQ engine, you can set your measurement specifications as goals or constraints.The engine strives to get as close to the goals as possible while ensuring that the constraints are met.

- Random engine

  The Random engine randomly picks values within the specified range and displays misfit error and parameter history.

- Discrete engine

  The Discrete engine is used at the end of the optimization cycle to round off component values to the closest values available commercially. Typically, once you have optimized your circuit, you will most likely want to convert your component values into “real-world” parts.

  For example, the Optimizer determines that the 3K resistor in the RF amplifier circuit should be 2.18113K, but you cannot use this value in your manufactured design. You can then specify a discrete table and switch to the Discrete Engine. The Discrete engine determines a new value for this resistor depending on the table used. For a one percent table, the new value is 2.21K.

  The Optimizer in Advanced Analysis provides discrete value defaults for resistors, capacitors, and inductors.

5

Smoke

In this chapter

- Smoke overview on page 149
- Smoke strategy on page 150
- Smoke procedure on page 151
- Example on page 156
- For power users on page 166

Smoke overview

Note: Smoke analysis is available with the following products:

- PSpice[1] Smoke Option
- PSpice Advanced Analysis

Long-term circuit reliability

Smoke warns of component stress due to power dissipation, increase in junction temperature, secondary breakdowns, or violations of voltage / current limits. Over time, these stressed components could cause circuit failure.

Smoke uses Maximum Operating Conditions (MOCs), supplied by vendors and derating factors supplied by designers to calculate the Safe Operating Limits (SOLs) of a component's parameters.

1. Depending on the license available, you will access either PSpice or PSpice Simulator.

Smoke then compares circuit simulation results to the component's safe operating limits. If the circuit simulation exceeds the safe operating limits, Smoke identifies the problem parameters.

Use Smoke for Displaying Average, RMS, or Peak values from simulation results and comparing these values against corresponding safe operating limits

Safe operating limits

Smoke will help you determine:

- Breakdown voltage across device terminals
- Maximum current limits
- Power dissipation for each component
- Secondary breakdown limits
- Junction temperatures

Smoke strategy

Smoke is useful as a final design check after running Sensitivity, Optimizer, and Monte Carlo, or you can use it on its own for a quick power check on a new circuit.

Plan ahead

Smoke requires:

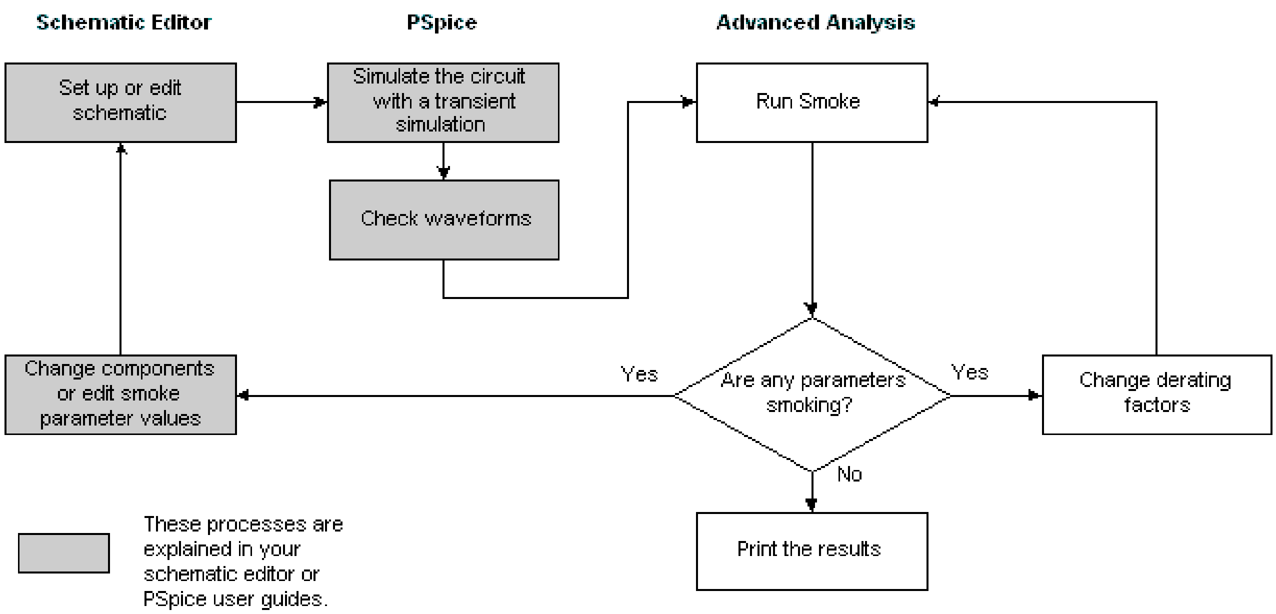
- Components that are Advanced Analysis-ready

  See Chapter 2, Libraries.

  See Smoke parameters on page 166 for lists of parameter names used in Advanced Analysis Smoke.
- A working circuit schematic and transient simulation
- Derating factors

  Smoke uses "no derating" as the default.

Workflow



Smoke procedure

Setting up the circuit in the schematic editor

Advanced Analysis requires:

- A circuit schematic and working PSpice simulation

Smoke analysis also requires:

- Any components included in a Smoke analysis must have smoke parameters specified.

  For more information see Chapter 2, Libraries.

- Time Domain (transient) analysis as a simulation

  Smoke does not work on other types of analyses, such as DC Sweep or AC Sweep/Noise analyses.

a. From your schematic editor, open your circuit.

b. Run a PSpice simulation.

c. Check your key waveforms in PSpice and make sure they are what you expect.

Note: For information on schematic design and simulation setup, see your schematic editor and PSpice user guides.

See Smoke parameters on page 166.

Running Smoke

Starting a run

➡ In your schematic editor, from the **PSpice** menu, select **Advanced Analysis** / **Smoke**.

Smoke automatically runs on the active transient profile.

Smoke calculates safe operating limits using component parameter maximum operating conditions and derating factors.

The output window displays status messages.

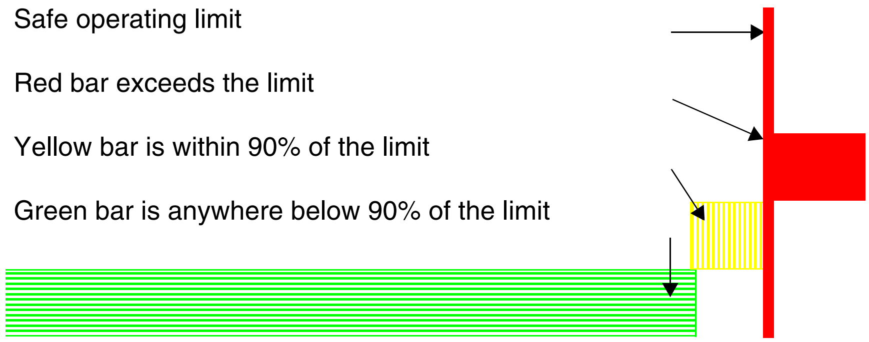
Viewing Smoke results

- To see **Average**, **RMS**, and **Peak** values, right click and from the pop-up menu select the values you want to review.

  Check the bar graph:

  - Red bars show values that exceed safe operating limits.
  - Yellow bars show values getting close to the safe operating limits: between 90 and 100 percent of the safe operating limits.
  - Green bars show values within safe operating limits: less than 90 percent of the safe operating limits.

- ❑ Grey bars indicate the limit is not valid for the parameter.



- To decipher the acronym for a parameter, right click and from the pop-up menu select **Parameter Descriptions**.
- To view temperature parameters only, right click and from the pop-up menu select **Temperature Only Parameters**.

  Only average and peak values are useful when viewing temperature parameters.
- To change the sort order of a column, click on the column header.
- To locate a problem component in your schematic, right click on a component parameter and select **Find in Design** from the pop-up menu.

  This returns you to the schematic editor with the component selected.
- To view components with similar set of reference designators, right-click in the smoke result spreadsheet and choose *Component Filter*. Specify the search criterion in the *Find what* box, such as `R*` to search for all components with reference designator starting with R.

  Alternatively, you can choose *Analysis - Smoke - Component Filter*.

Printing results

➡ Click  .

OR

From the File menu, select Print.

Configuring Smoke

Changing components or parameters

Smoke results are read-only. To modify the circuit:

1. Make your changes in your schematic editor.
2. Rerun the PSpice simulation.

Follow the steps for Setting up the circuit in the schematic editor on page 151 and Running Smoke on page 152.

Controlling smoke on individual design components

You can use the SMOKE_ON_OFF property to control whether or not you want to run smoke analysis on individual devices or blocks in a schematic.

If you attach the SMOKE_ON_OFF property to the device instance for which you do not want to perform the smoke analysis, and set the value to OFF, the smoke analysis would not run for this device.

This property can also be used on hierarchical blocks. The value of the SMOKE_ON_OFF property attached to the parent block has a higher priority over the property value attached to the individual components.

Selecting other deratings

To select other deratings:

1. Right click and from the pop-up menu select **Derating**.
2. Select one of the three derating options on the pull-right menu:
    - No Derating
    - Standard Derating
    - Custom Derating Files

3. Click on the top toolbar to run a new Smoke analysis with the revised derating factors.

   New results appear.

For information on creating a custom derating file, see Adding Custom Derate file.

Example

Overview

This example uses the tutorial version of RFAmp located at:

`<target_directory>\PSpice\tutorial\capture\pspiceaa\rfamp`

`<target_directory>\PSpice\tutorial\concept\pspiceaa\rfamp`

The circuit is an RF amplifier with 50-ohm source and load impedances. It includes the circuit schematic, PSpice simulation profiles, and measurements.

For a completed example see:

`<target_directory>\PSpice\Capture_Samples\AdvAnls\RFAmp directory.`

For a completed example see:

`<target_directory>\PSpice\Concept_Samples\AdvAnls\RFAmp directory.`

Setting up the circuit in the schematic editor

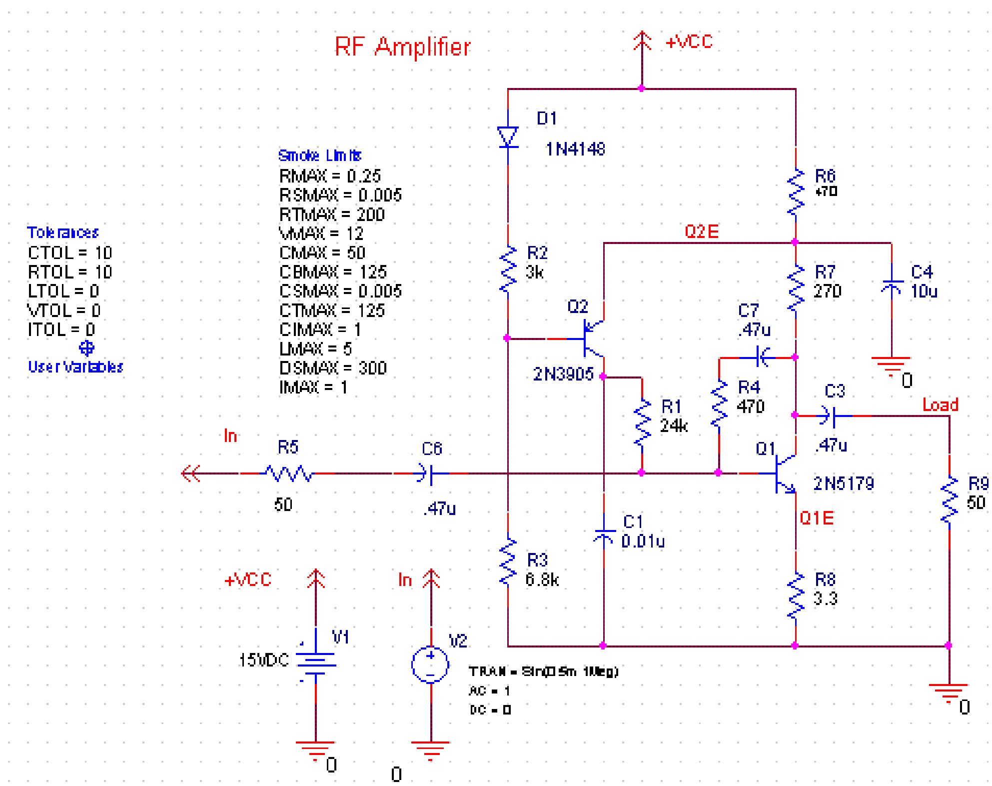
1. In your schematic editor, browse to the RFAmp tutorials directory.

   `<target_directory>\PSpice\tutorial\Capture\pspiceaa\rfamp`

   `<target_directory>\PSpice\tutorial\Concept\pspiceaa\rfamp`

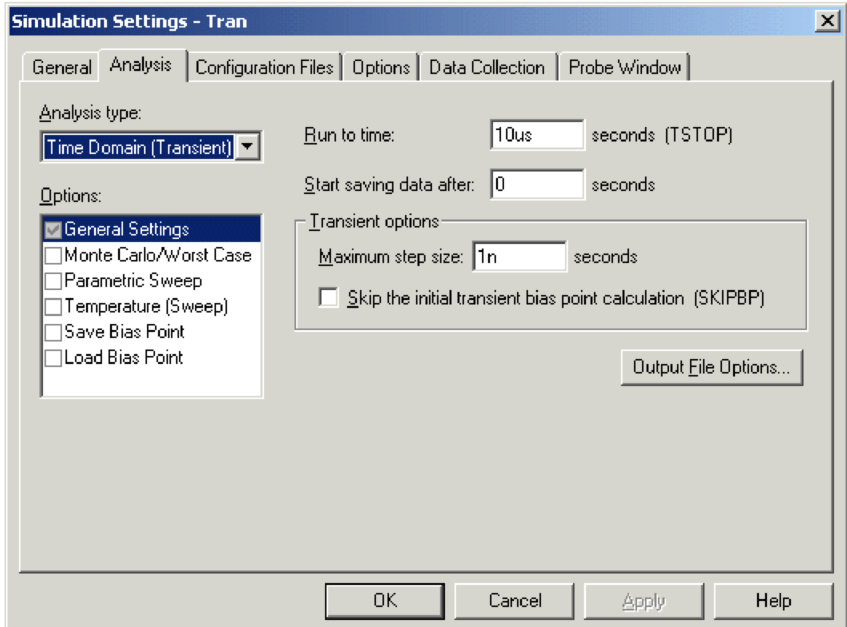
2. Open the RFAmp project.

The RF amplifier circuit example

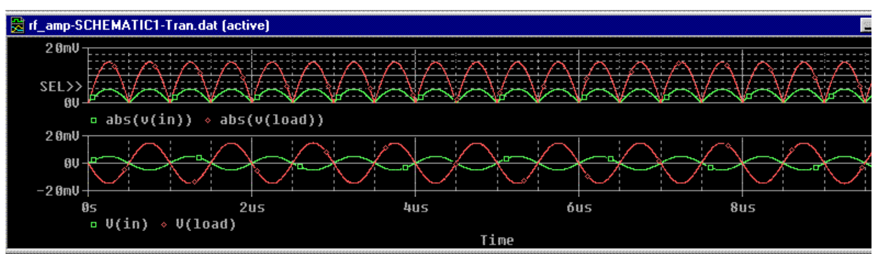


3. Select SCHEMATIC1-Tran.

The Transient simulation included in the RF Amp example



1. Click on the top toolbar to run the PSpice simulation.
2. Review the results.



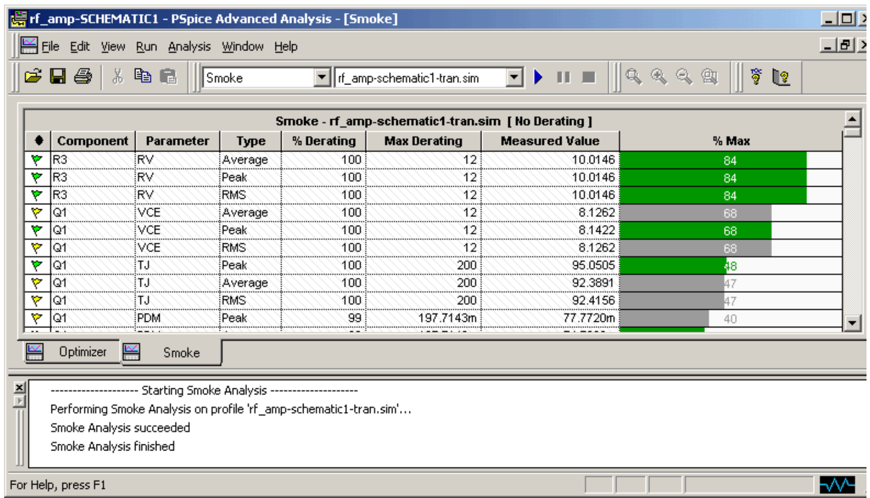
The key waveforms in PSpice are what we expected.

Running Smoke

Starting a run

➡ From the **PSpice** menu in your schematic editor, select **Advanced Analysis** / **Smoke**.

The Smoke tool opens and automatically runs on the active transient profile.

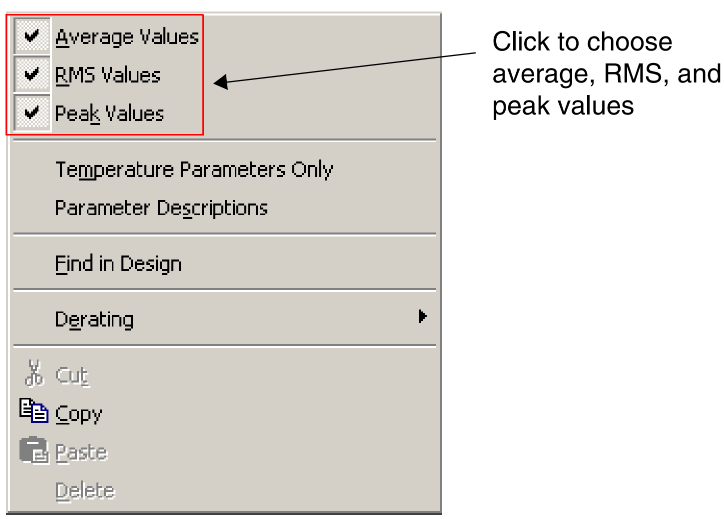


Smoke calculates safe operating limits using component parameter maximum operating conditions and derating factors.

The output window displays status messages.

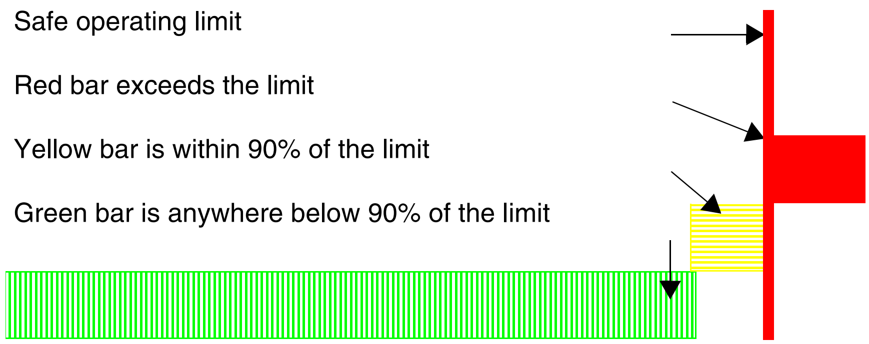
Viewing Smoke results

1. Right click and from the pop-up menu select **Average**, **RMS**, and **Peak Values**.



In the %Max column, check the bar graphs.

- Red bars show values that exceed safe operating limits.
- Yellow bars show values getting close to the safe operating limits: between 90 and 100 percent of the safe operating limits.
- Green bars show values well within the safe operating limits: less than 90 percent of the safe operating limits.
- Grey bars indicate that limits are not valid for the parameters.



The value in the % Max column is calculated using the following formula:

(5-1) %Max=Actual operating Value/Safe operating limit *100

Where:

Actual operating value

- is displayed in the *Measured Value* column.
- is calculated by the simulation controller.

Safe operating limit	■ is displayed in the *Max Derating* column. ■ is `MOC*derating_factor`. ■ MOC or the Maximum Operating Condition is specified is the vendor supplied data sheet ■ derating factor, is specified by the users in the *% Derating* column.

The value calculated using the Equation 5-1 on page 160 is rounded off to the nearest integer, larger than the calculated value, and then displayed in the %Max column.

For example, if the calculated value of %Max is 57.06, the value displayed in the %Max column will be 58.

2. Right click on the table and select **Temperature Parameters Only** from the pop-up menu.

   Only maximum resistor or capacitor temperature (TB) and maximum junction temperature (TJ) parameters are displayed. When reviewing these results, only average and peak values are meaningful.

In this example, none of the parameters are stressed, as indicated by the green bars.

Right click to select average, RMS, and peak values

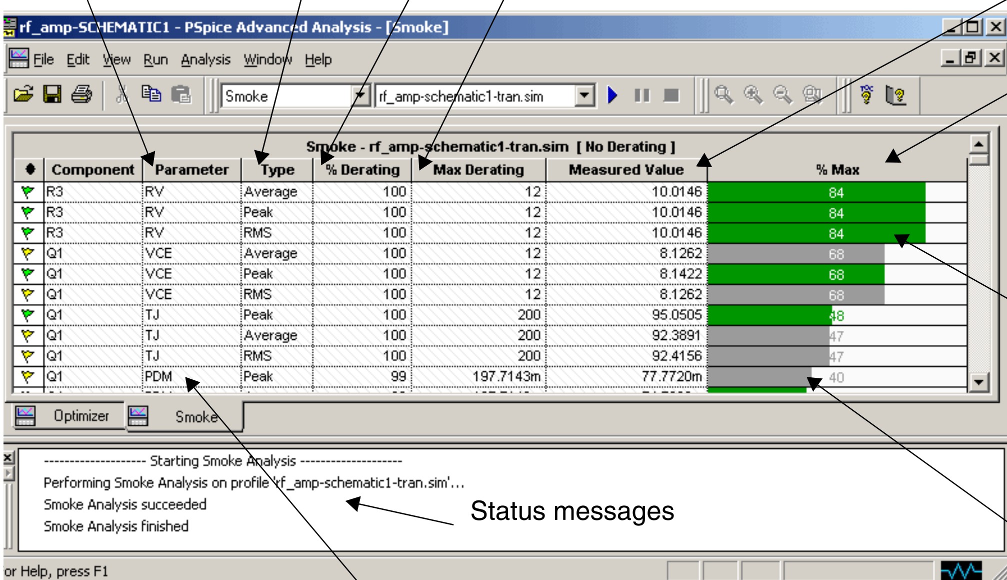
Right click to select temperature-only parameters

Shows the derating factor percentage

The calculated SOL is the max derating =
SOL = MOC x % derating

Shows the parameter measurement (for example: voltage, power, current, or temperature)

% Max is the (actual operating value / SOL) x 100



Green bars show values within the safe operating limits

Status messages

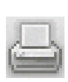
Grey bars indicate that limits are not valid for the parameters

Right click and select Parameter Descriptions to decipher the parameter acronyms

Note: The Junction Temperature TJ is calculated as $\mathrm{TJ} = \mathrm{PDM} \times (\mathrm{RCA} + \mathrm{RJC}) + \mathrm{Sim}_{\mathrm{temp}}$ where:

- PDM is the maximum power dissipation taken from analysis result
- $\mathrm{Sim}_{\mathrm{temp}}$ is the simulated temperature
- RCA is case-to-ambient thermal resistance
- RJC is the junction-to-case thermal resistance

Printing results

➡ Click  .

OR

From the File menu, select Print.

Configuring Smoke

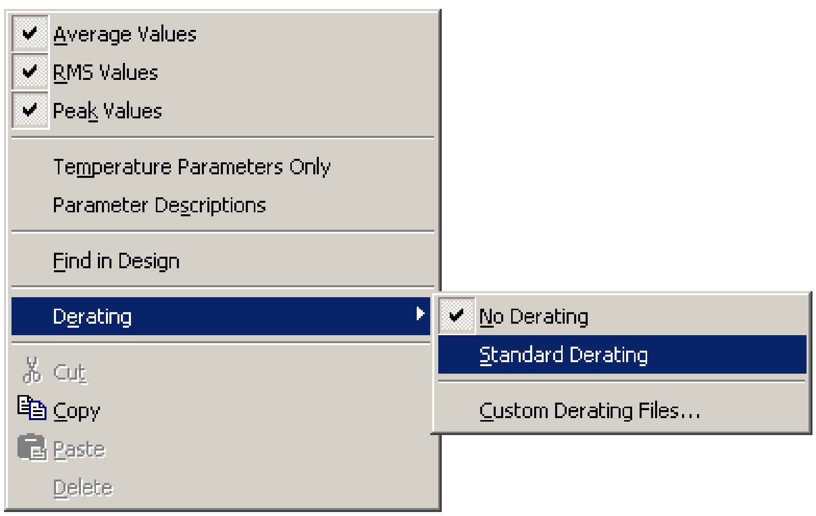
Selecting another derating option

The default derating option uses 100% derating factors, also called No Derating.

We'll now run the circuit with standard derating and examine the results.

Selecting standard derating

1. Right click and from the pop-up menu select **Derating**.
2. Select **Standard Derating** from the pull-right menu.



3. Click  on the top toolbar to run a new Smoke analysis.

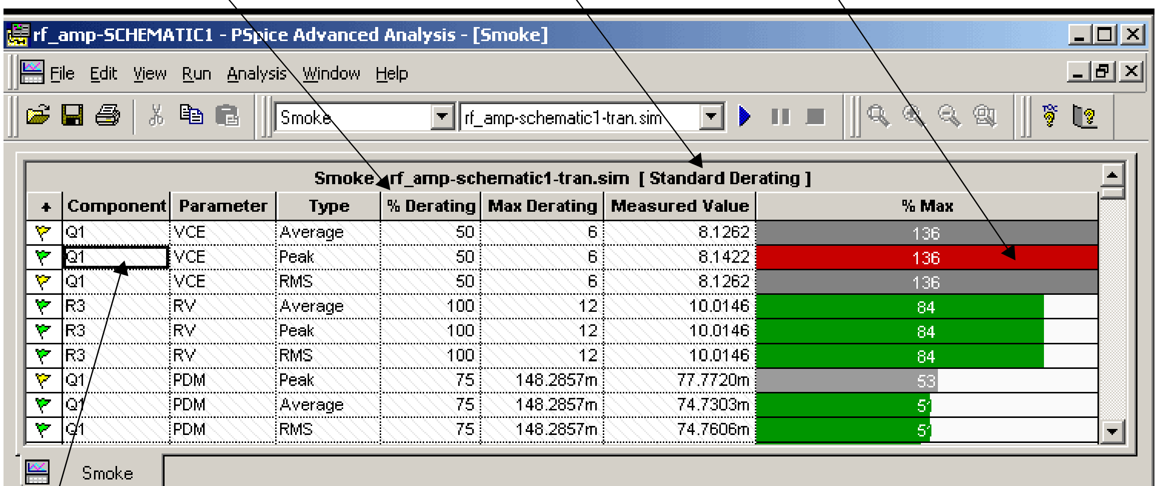
   New results appear.

The red bar indicates that Q1's VCE parameter is stressed.

Standard derating factors used in the calculations

Standard Derating appears in the title

Component Q1's VCE parameter is stressed to 136 percent of its safe operating limit



Right click on Q1 and from the pop-up menu select Find in Design. This takes you to the schematic where the component parameter can be changed.

4. Resolve the component stress:

   ❑ Right click on Q1 VCE and from the pop-up menu select **Find in Design** to go to the schematic and adjust circuit parameters.

   Or:

   ❑ Right click and from the pop-up menu select **Deratings** \ **No Derating** to change the derating option back to **No Derating**.

   Note: You might select a Q with higher VCE voltage if parameters are fine.

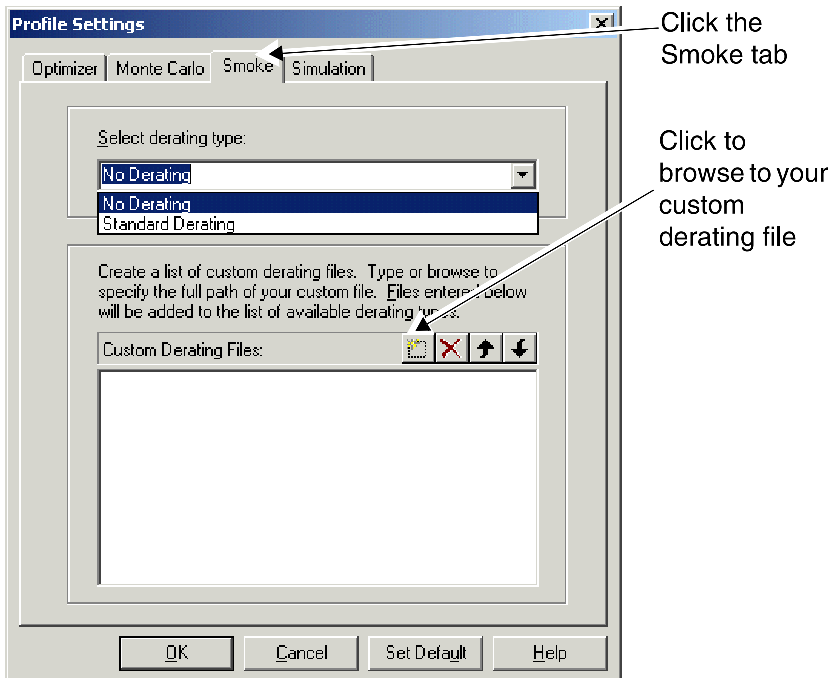
5. Click  on the top toolbar to rerun Smoke analysis after making any adjustments.

6. Check the results.

Selecting custom derating

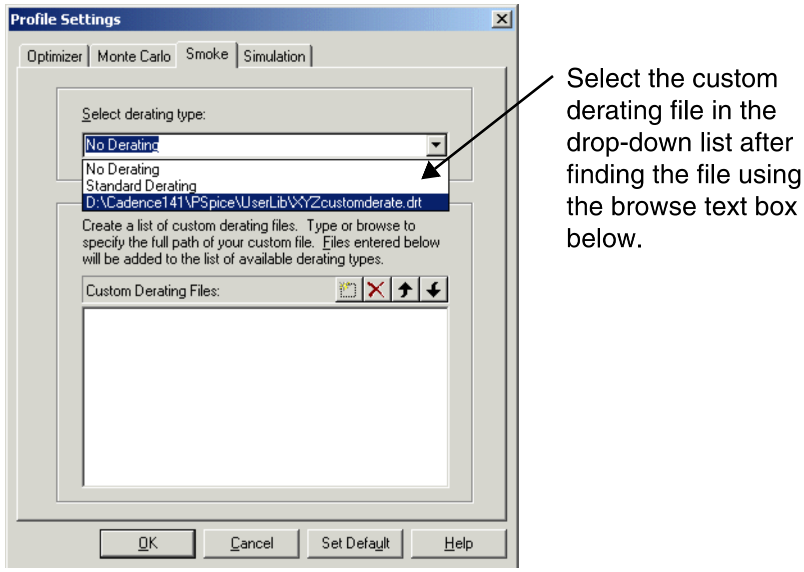
If you have your own custom derating factors, you can browse to your own file and select it for use in Smoke. For information on creating a custom derating file, see Adding Custom Derate file on page 175.

1. Once you have your custom derating file in place, right click and from the pop-up menu select **Derating**.
2. Select **Custom Derating Files** from the pull-right menu.



3. Click the browse icon.
4. Browse and select your file.

The file name is added to the list in the Custom Derating Files text box and the drop-down list.



5. Select the custom derating file from the drop-down list.
6. Click **OK**.
7. Click  on the top toolbar to run a new Smoke analysis.

   New results appear.
8. Check the results.

   To make changes, follow the steps for changing derating options or schematic component values.

   See Selecting standard derating on page 163.

For power users

Smoke parameters

The following tables summarize smoke parameter names you will see in the Smoke results. The tables are sorted by user interface parameter names and include:

- Passive component parameters

- Semiconductor component parameters
- OpAmp component parameters

For passive components, three names are used in Smoke analysis: symbol property names, symbol parameter names, and parameter names used in the Smoke user interface. This table is sorted in alphabetical order by parameter names that display in the Smoke user interface.

Smoke User Interface Parameter Name	Passive Component	Maximum Operating Condition	Symbol Property Name	Symbol Smoke Parameter Name	Variable Table Default Value
CI	Capacitor	Maximum ripple*	CURRENT	CIMAX	1 A
CV	Capacitor	Voltage rating	VOLTAGE	CMAX	50 V
CVN	Capacitor	Maximum Reverse Voltage	NEGATIVE_ VOLTAGE	CVN	10V
IV	Current Supply	Max. voltage current source can withstand	VOLTAGE	VMAX	12 V
LI	Inductor	Current rating	CURRENT	LMAX	5 A
LIDC	Inductor	DC current Value	DC_CURRE NT	DC	0.1A
LV	Inductor	Dielectric strength	DIELECTRI C	DSMAX	300 V
PDM	Resistor	Maximum power dissipation of resistor	POWER	RMAX	0.25 W
PDML**	Capacitor	Maximum power loss due to series resistance	POWER	CPMAX	0.1W

Smoke User Interface Parameter Name	Passive Component	Maximum Operating Condition	Symbol Property Name	Symbol Smoke Parameter Name	Variable Table Default Value
	Inductor	Maximum power loss due to series resistance	POWER	LPMAX	0.25W
RBA* (=1/SLOPE)	Resistor	Slope of power dissipation vs. temperature	SLOPE	RSMAX	0.005W/deg C
RV	Resistor	Voltage Rating	VOLTAGE	RVMAX	--
SLP*	Capacitor	Temperature derating slope	SLOPE of volt temperature curve	CSMAX	0.005 V/degC
TBRK*	Capacitor	Breakpoint temperature	KNEE	CBMAX	125 degC
TJL	Capacitor	Rise in temperature	RTH	THERMR	1 degC
TMAX*	Capacitor	Maximum temperature	MAX_TEMP	CTMAX	125 degC
TMAX, TB	Resistor	Maximum temperature resistor can withstand	MAX_TEMP	RTMAX	200 degC
VI	Voltage Supply	Max. current voltage source can withstand	CURRENT	IMAX	1 A

*Parameters used internally and having no impact on the result values displayed in the Smoke window

** The PDML parameter is affected by the ESR value, which is available under the property name ESR in Capacitor and DC_RESISTANCE in Inductors. The variable table default value for ESR is 0.001 Ω.

Note: Ripple current refers to the AC portion of signals due to the small variations in DC signals usually in power supply applications.

The following table lists smoke parameter names for semiconductor components. The table is sorted in alphabetical order according to parameter names that will display in the Smoke results.

Smoke Parameter Name and Symbol Property Name	Semiconductor Component	Maximum Operating Condition
IB	BJT	Maximum base current (A)
IC	BJT	Maximum collector current (A)
PDM	BJT	Maximum power dissipation (W)
RCA	BJT	Thermal resistance, Case-to-Ambient (degC/W)
RJC	BJT	Thermal resistance, Junction-to-Case (degC/W)
SBINT	BJT	Secondary breakdown intercept (A)
SBMIN	BJT	Derated percent at TJ (secondary breakdown)
SBSLP	BJT	Secondary breakdown slope
SBTSLP	BJT	Temperature derating slope (secondary breakdown)
TJ	BJT	Maximum junction temperature (degC)
VCB	BJT	Maximum collector-base voltage (V)
VCE	BJT	Maximum collector-emitter voltage (V)
VEB	BJT	Maximum emitter-base voltage (V)
F1	Bridge	Maximum forward current of Diode1 (A)
IF2	Bridge	Maximum forward current of Diode2 (A)
IF3	Bridge	Maximum forward current of Diode3 (A)
IF4	Bridge	Maximum forward current of Diode4 (A)
PDM	Bridge	Maximum power dissipation (W)
RCA	Bridge	Thermal resistance, Case-to-Ambient (degC/W)

Smoke Parameter Name and Symbol Property Name	Semiconductor Component	Maximum Operating Condition
RJC	Bridge	Thermal resistance, Junction-to-Case (degC/W)
TJ	Bridge	Maximum junction temperature (degC)
VR1	Bridge	Peak reverse voltage of Diode1 (V)
VR2	Bridge	Peak reverse voltage of Diode2 (V)
VR3	Bridge	Peak reverse voltage of Diode3 (V)
VR4	Bridge	Peak reverse voltage of Diode4 (V)
IF	Diode	Maximum forward current (A)
PDM	Diode	Maximum power dissipation (W)
RCA	Diode	Thermal resistance, Case-to-Ambient (degC/W)
RJC	Diode	Thermal resistance, Junction-to-Case (degC/W)
TJ	Diode	Maximum junction temperature (degC)
VR	Diode	Maximum reverse voltage (V)
ID	Dual MOS	Maximum drain current (A)
IG	Dual MOS	Maximum forward gate current (A)
PDM	Dual MOS	Maximum power dissipation (W)
RCA	Dual MOS	Thermal resistance, Case-to-Ambient (degC/W)
RJC	Dual MOS	Thermal resistance, Junction-to-Case (degC/W)
TJ	Dual MOS	Maximum junction temperature (degC)
VDG	Dual MOS	Maximum drain-gate voltage (V)
VDS	Dual MOS	Maximum drain-source voltage (V)
VGSF	Dual MOS	Maximum forward gate-source voltage (V)
VGSR	Dual MOS	Maximum reverse gate-source voltage (V)
IC	IGBT	Maximum collector current (A)

Smoke Parameter Name and Symbol Property Name	Semiconductor Component	Maximum Operating Condition
IG	IGBT	Maximum gate current (A)
PDM	IGBT	Maximum Power dissipation (W)
RCA	IGBT	Thermal resistance, Case-to-Ambient (degC/W)
RJC	IGBT	Thermal resistance, Junction-to-Case (degC/W)
TJ	IGBT	Maximum junction temperature (degC)
VCE	IGBT	Maximum collector-emitter (V)
VCG	IGBT	Maximum collector-gate voltage (V)
VGEF	IGBT	Maximum forward gate-emitter voltage (V)
VGER	IGBT	Maximum reverse gate-emitter (V)
ID	JFET or MESFET	Maximum drain current (A)
IG	JFET or MESFET	Maximum forward gate current (A)
PDM	JFET or MESFET	Maximum power dissipation (W)
RCA	JFET or MESFET	Thermal resistance, Case-to-Ambient (degC/W)
RJC	JFET or MESFET	Thermal resistance, Junction-to-Case (degC/W)
TJ	JFET or MESFET	Maximum junction temperature (degC)
VDG	JFET or MESFET	Maximum drain-gate voltage (V)
VDS	JFET or MESFET	Maximum drain-source voltage (V)
VGS	JFET or MESFET	Maximum gate-source voltage (V)
IFD	LED	Maximum forward current (A)
PDM	LED	Maximum power dissipation (W)
RCA	LED	Thermal resistance, Case-to-Ambient (degC/W)
RJC	LED	Thermal resistance, Junction-to-Case (degC/W)

Smoke Parameter Name and Symbol Property Name	Semiconductor Component	Maximum Operating Condition
TJ	LED	Maximum junction temperature (degC)
VD	LED	Maximum reverse voltage (V)
ID	MOSFET or Power MOSFET	Maximum drain current (A)
IG	MOSFET or Power MOSFET	Maximum forward gate current (A)
PDM	MOSFET or Power MOSFET	Maximum power dissipation (W)
RCA	MOSFET or Power MOSFET	Thermal resistance, Case-to-Ambient (degC/W)
RJC	MOSFET or Power MOSFET	Thermal resistance, Junction-to-Case (degC/W)
TJ	MOSFET or Power MOSFET	Maximum junction temperature (degC)
VDG	MOSFET or Power MOSFET	Maximum drain-gate voltage (V)
VDS	MOSFET or Power MOSFET	Maximum drain-source voltage (V)
VGSF	MOSFET or Power MOSFET	Maximum forward gate-source voltage (V)
VGSR	MOSFET or Power MOSFET	Maximum reverse gate-source voltage (V)
IC	Optocoupler	Maximum collector current (A)
IFD	Optocoupler	Maximum forward current (A)
PDM	Optocoupler	Maximum power dissipation (W)
VCEO	Optocoupler	Maximum collector-emitter voltage (V)
VD	Optocoupler	Maximum reverse voltage (V)
VECO	Optocoupler	Maximum emitter-collector voltage (V)
PDSW	Switch	Rated Switch Power (W)

Smoke Parameter Name and Symbol Property Name	Semiconductor Component	Maximum Operating Condition
SI	Switch	Rated Switch Current (A)
SV	Switch	Rated Switch Contact Voltage (V)
ITM	Varistor	Peak current (A)
IGM	Thyristor	Maximum gate current (A)
IT	Thyristor	Maximum anode current (A)
RCA	Thyristor	Thermal resistance, Case-to-Ambient (degC/W)
RJC	Thyristor	Thermal resistance, Junction-to-Case (degC/W)
TJ	Thyristor	Maximum junction temperature (degC)
VDRM	Thyristor	Maximum anode-cathode voltage (V)
VRRM	Thyristor	Maximum cathode-anode voltage (V)
Primary_Current	Transformer (Single and Double)	Primary current (A)
Isolation_Voltage	Transformer Double	Isolation Voltage between Primary and Secondary (V)
Isolation_Voltage1	Transformer Double	Isolation Voltage between Primary and Secondary (V)
Isolation_Voltage2	Transformer Double	Isolation Voltage between Primary and Secondary (V)
Secondary_one_Current	Transformer Double	First Secondary Current (A)
Secondary_two_Current	Transformer Double	Second Secondary Current (A)
Secondary_Current	Transformer Single	Secondary Current (A)
RCA	Varistor	Thermal resistance, Case-to-Ambient (degC/W)

Smoke Parameter Name and Symbol Property Name	Semiconductor Component	Maximum Operating Condition
RJC	Varistor	Thermal resistance, Junction-to-Case (degC/W)
TJ	Varistor	Maximum junction temperature (degC)
IFS	Zener Diode	Maximum forward current (A)
IRMX	Zener Diode	Maximum reverse current (A)
PDM	Zener Diode	Maximum power dissipation (W)
RCA	Zener Diode	Thermal resistance, Case-to-Ambient (degC/W)
RJC	Zener Diode	Thermal resistance, Junction-to-Case (degC/W)
TJ	Zener Diode	Maximum junction temperature (degC)

The following table lists smoke parameter names for Op Amp components. The table is sorted in alphabetical order according to parameter names that will display in the Smoke results.

Smoke Parameter Name	Op Amp Component	Maximum Operating Condition
IPLUS	OpAmp	Non-inverting input current
IMINUS	OpAmp	Inverting input current
IOUT	OpAmp	Output current
VDIFF	OpAmp	Differential input voltage
VSMAX	OpAmp	Supply voltage
VSMIN	OpAmp	Minimum supply voltage
VPMAX	OpAmp	Maximum input voltage (non-inverting)
VPMIN	OpAmp	Minimum input voltage (non-inverting)
VMMAX	OpAmp	Maximum input voltage (inverting)
VMMIN	OpAmp	Minimum input voltage (inverting)

Adding Custom Derate file

Why use derating factors?

You might need to test components for certain parameters at a lower value then specified by the manufacturer. This reduction or strict specification of value is achieved by derating factor. The ideal safe operating limits specified for parameters, such as power dissipation or temperature coefficients, need to be changed to a lower value for real applications. In addition, the parameters of a component specified in manufactured datasheet cannot always be used as is because in working environment change in one parameter affects others. For example, the manufacturer datasheet of resistors specifies the maximum temperature and the maximum power dissipation. However, when the power dissipation of a resistor is increased, the device temperature also increases. As a result, the resistors capability to handle power is reduced with increasing temperature.

Although for calculations an absolute value of the power is used, while plotting the deration curve of a resistor, the rated power is specified per unit. This means that the maximum power dissipation (PDM) is always specified as an unit.

For example, Figure 5-2 shows the curve for a resistor with power dissipation of 0.25W and maximum temperature (TMAX) of 150°C.

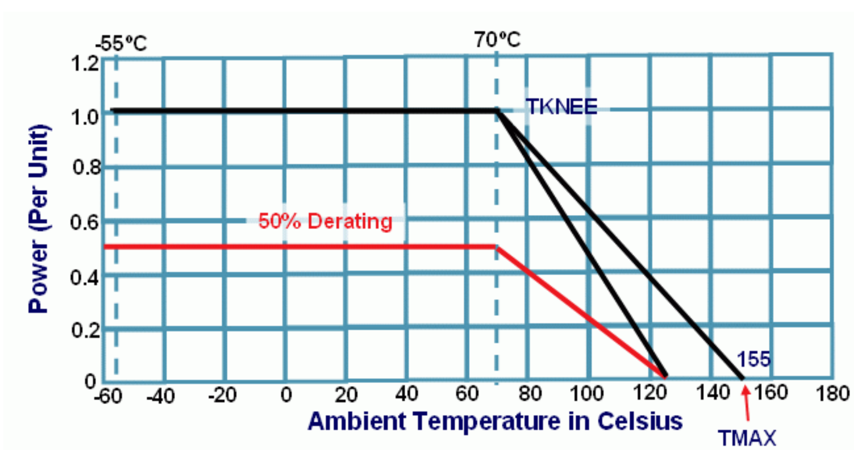


Figure 5-1 Deration Curve for a Resistor

Notice that the maximum rated power dissipation is plotted as percentage and not the absolute value.

The following two examples show the values for the different deration parameters for two different TKNEE values. The first case has a normal slope whereas the last example has a fast slope. For a slow slope the TKNEE value is much lower than the maximum value TMAX. In contrast, in Case-2, the slope angles over 10°C, where the TKNEE value is 100°C and TMAX is 110°C.

Note: The actual temperature is calculated as $T_{actual} = T_{amb} + ((RTMAX - T_{KNEE})(P_{dissipated} / (RMAX)))$, where T_{amb} is the ambient temperature and T_{max} is the maximum temperature. The TKNEE is calculated as $TKNEE = RTMAX - ((RMAX) / (RSMAX))$, where RMAX is equivalent of PDM and RSMAX is equivalent of the slope.

You can calculate TKNEE using two methods:

- By defining a parameter TKNEE on instance
- Using the existing slope of the deration curve

Note: It is recommended that the TKNEE value should be more than the simulation temperature. When TKNEE value is calculated within Smoke Analysis, a warning is generated if the TKNEE value is less than the simulation temperature. You can specify the TKNEE value on a part in the design to remove this warning.

To calculate temperature rise of a resistance use one of the following methods in the order specified:

1. If both TRISE and TKNEE are defined:

   $$T_{actual} = T_{amb} + TRISE \cdot \frac{P_{dissipated}}{RMAX}$$

   Where Tmax is the temperature rise at rated power. Users can define Tmax.

2. If TKNEE is defined but TRISE is not defined:

   $$T_{actual} = T_{amb} + ((RTMAX - T_{KNEE})(P_{dissipated} / (RMAX)))$$

   The deration slope is used to calculate the temperature rise.

3. If both TRISE and TKNEE are not defined:

   $$T_{actual} = T_{amb} + ((RTMAX - T_{KNEE})(P_{dissipated} / (RMAX)))$$

It is assumed that device has maximum rated temperature at rated power dissipation.

Case 1: KNEE value of 0°C

Figure 5-2 on page 177 shows a circuit with specified parameters RMAX, RSMAX, and RTMAX. Using the following specified smoke limits, the calculated KNEE is 0°C:

- RMAX: 1.0
- RSMAX: 0.005
- RTMAX: 200

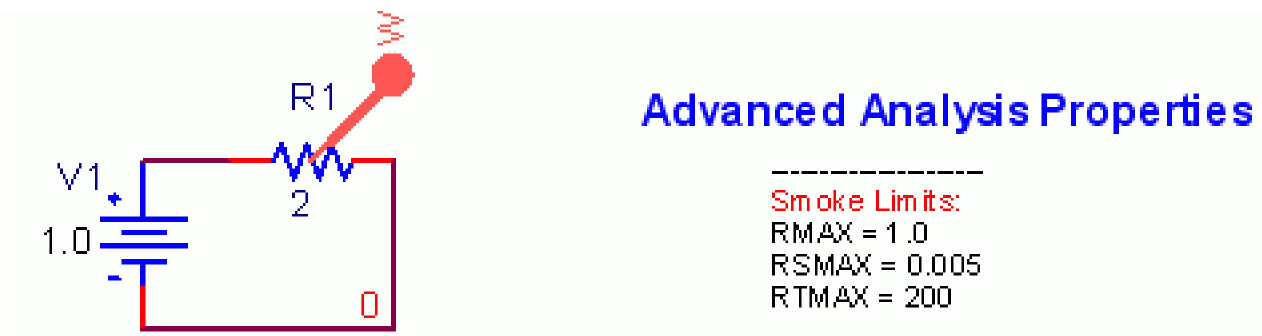


Figure 5-2 KNEE value of 0°C

The calculated values for the specified parameters are:

Parameter	Value
% Derating of Power or Derating Factor	36.5
Max Derating of Power	0.365
Max Derating of Temperature	200
Measured Power Value or Power Dissipation	500m

Measured Temperature Value or Actual Temperature	127°C
%Max of Power	137.1
%Max of Temperature	64

Case 2: KNEE value of 100°C

Using the following specified smoke limits, the calculated KNEE is 100°C:

- RMAX: 0.25
- RSMAX: 0.02
- RTMAX: 150

The calculated values for the specified parameters are:

Parameter	Value for simulation at Tamb of 27°C	Value for simulation at Tamb of 110°C
% Derating of Power or Derating Factor	100%	55%
Max Derating of Power	1	550m
Max Derating of Temperature	150	150
Measured Power Value or Power Dissipation	250m	250m
Measured Temperature Value or Actual Temperature	39.5°C	122.5°C
%Max of Power	25%	46%
%Max of Temperature	25%	82%

If you want a margin of safety in your design, apply a derating factor to your maximum operating conditions (MOCs). If a manufacturer lists 5W as the maximum operating condition for a resistor, you can insert a margin of safety in your design if you lower that value to 4.5W and run your simulation with 4.5W as the safe operating limit (SOL).

`As an equation: MOC x derating factor = SOL.`

In the example `5W x 0.9 = 4.5W`, the derating factor is 0.9. Also, 4.5W is 90% of 5W, so the derating factor is 90%. A derating factor can be expressed as a percent or a decimal fraction, depending on how it's used in calculations.

What is a custom derate file?

A custom derating file is an ASCII text file with a `.drt` extension that contains smoke parameters and derating factors specific to your project. If the "no derating" and "standard derating" factors provided with Advanced Analysis do not have the values you need for your project, you can create a custom derating file and type in the specific derating factors that meet your design specifications.

Figure 2 shows a portion of a custom derating file. The file lists resistor smoke parameters and derating factors. In your custom derating file, enter the derating factors as decimal percents in double quotes.

For the example below, if the resistor had a power dissipation (PDM) maximum operating condition of 5W, the .9 derating factor tells Advanced Analysis to use 0.9 x 5 = 4.5W as this resistor's safe operating limit.

```
("RES"
("PDM" "1")
("TMAX" "1")
("TB" "1")
)
```

Figure 5-3 Resistor smoke parameters and derating factors in a portion of a custom derating file

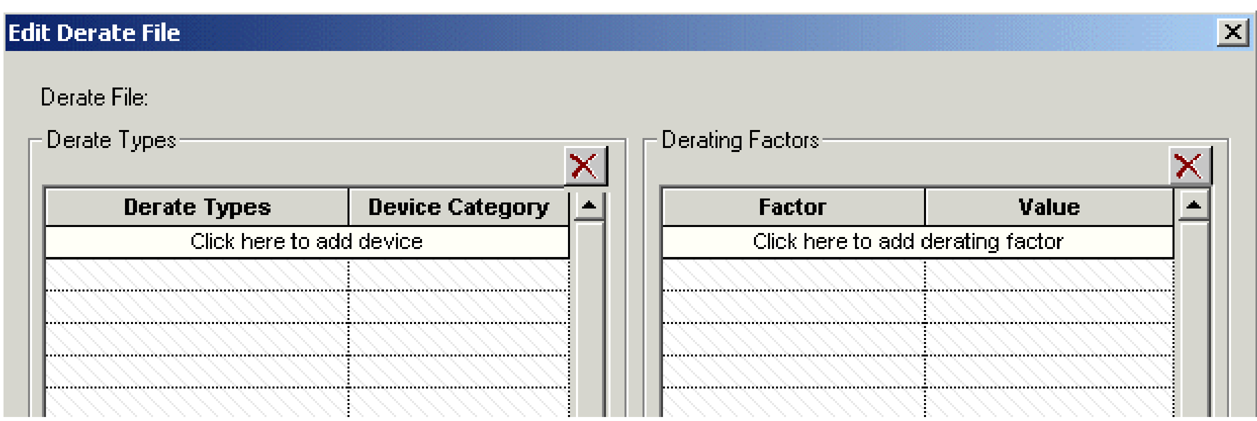
Creating a new custom derate file

Advanced Analysisprovides you the capability to create and edit derate files. You can perform this operation by using the Edit Derate File dialog box.

Note: To open the Edit Derate File dialog box, right-click in the results pane of Smoke and choose *Derating – Custom Derating Files* and then click the *Create Derate File* button.

1. Right-click in the results pane and choose *Derating – Custom Derating Files* to open the Profile Settings dialog box. Alternatively, you can choose *Edit – Profile Settings*.
2. To create a new derate file from scratch, click the *Create Derate File* button in the Profile Settings dialog box.

The Edit Derate File dialog box appears.

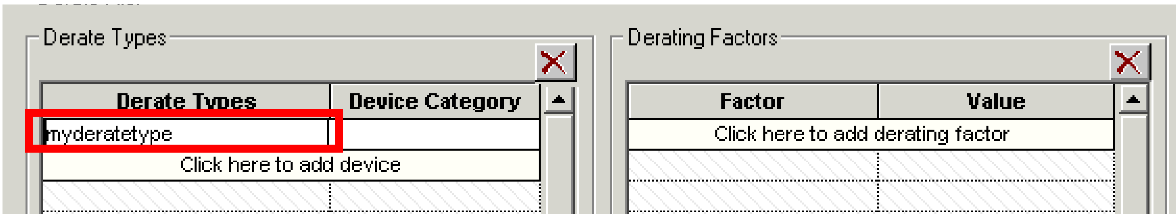


In the Edit Derate Type dialog box, type the derate type and select the a device category. The derate type can be any user defined value.

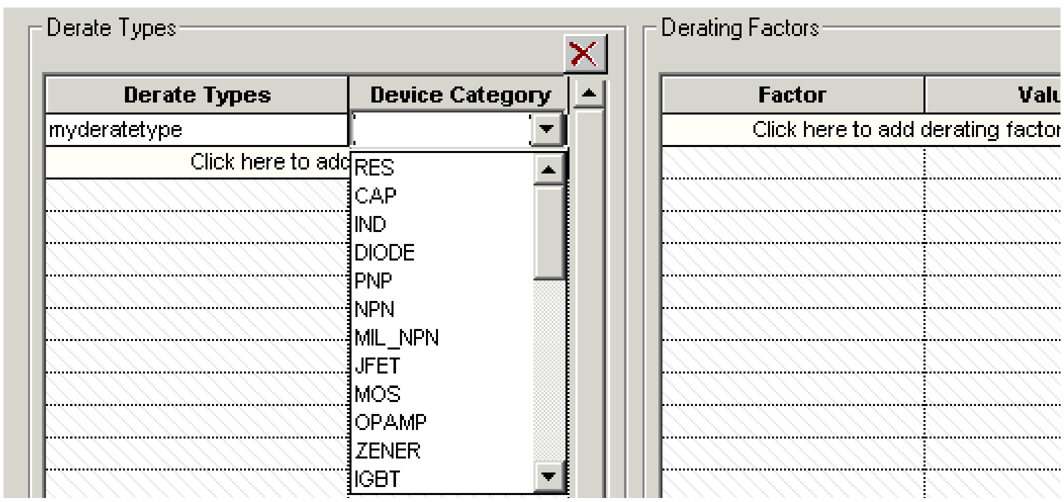
3. To add a new derate type, click the *Click here to add a device* row.

   A blank row gets added in the Derate Types pane.

4. In the Derate Types text box, enter any name, such as `myderatetype`



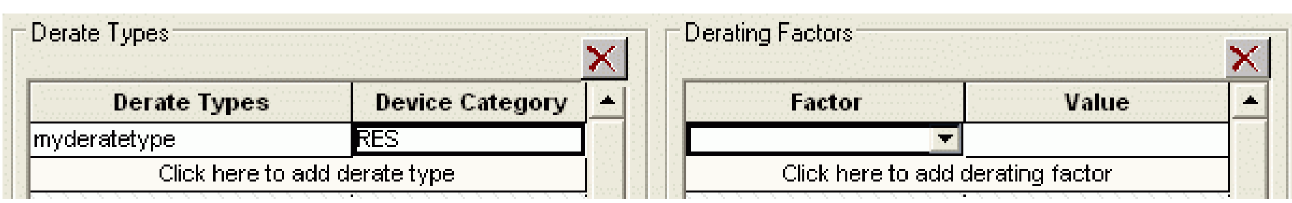
5. Click the Device Category grid.
6. From the drop-down list box select a device, such as `RES`.



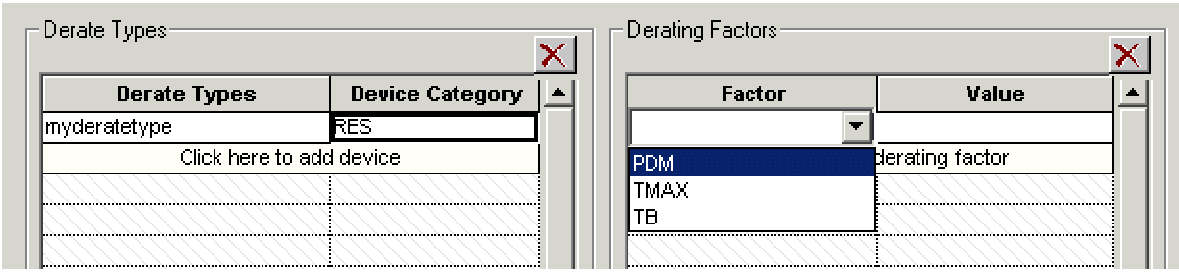
`myderatetype` is the derate type for a resistor of type `RES`.

7. To specify the derate values for various resistor parameters, click the *Click here to add derating factor* row in the Derating Factors window.

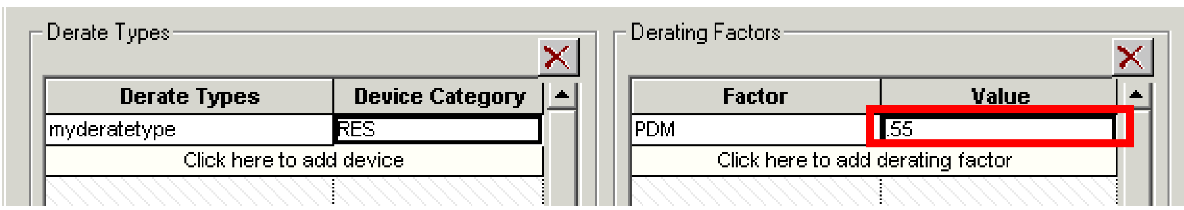
   A blank row gets added.



8. Select the derate factor from the Factor drop down list.



The corresponding default value for the derate factor is automatically filled in.



9. Modify the value of the derate factor as per the requirement.

10. Similarly, specify additional derate types and their corresponding categories, factors, and values.

Note: Derate factors are populated based on the selected device category

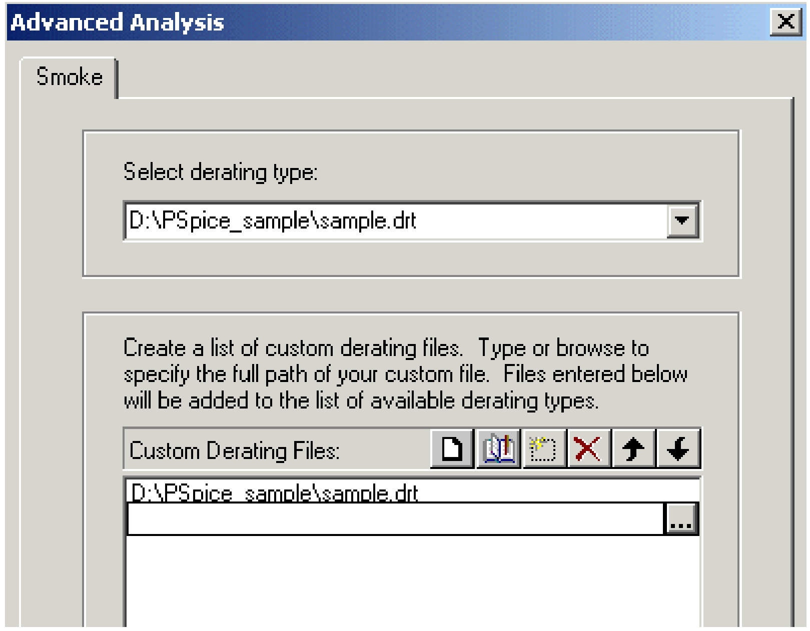
11. Save the derate file.

Note: To use the custom derate type SCHEMATIC, in the Property Editor, add a new property for the component with the name `DERATE_TYPE` and value same as the Derarte Type specified, such as *myderatetype*. Create netlist and select the corresponding derate file and run smoke.

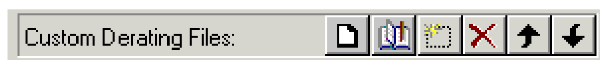
Modifying existing derate file

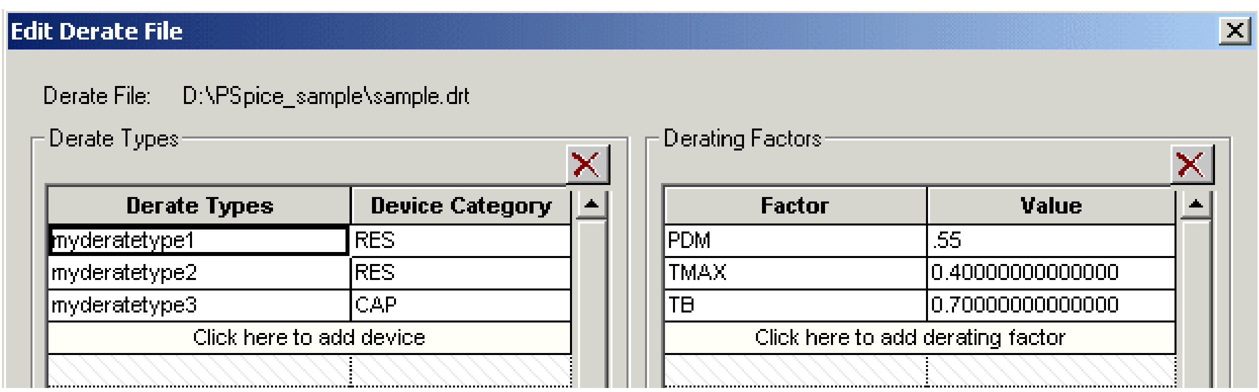
You can also use the Edit Derate File dialog box to modify the device type, device category, and the associated derating factor in an existing derate file.

1. Type the full path or browse to select an existing derate file.



2. Click the Edit Derate File button to display the Edit Derate File dialog box.





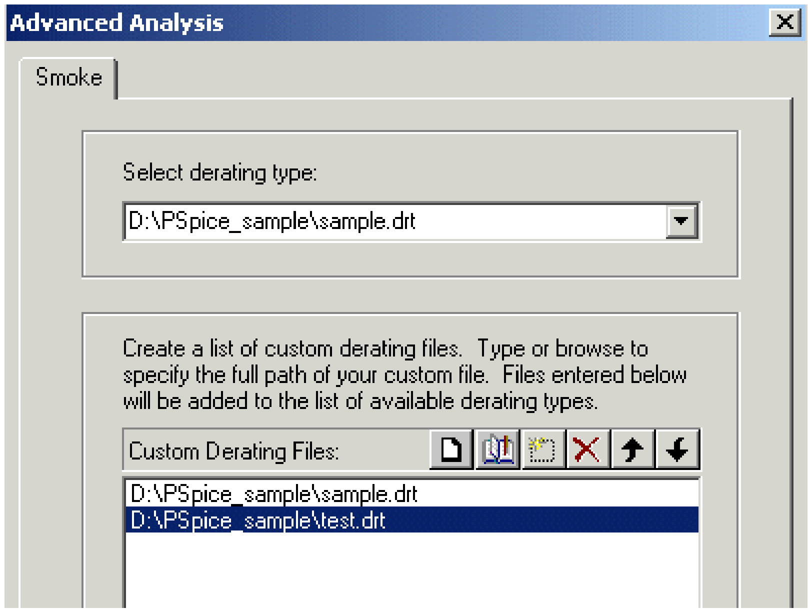
Adding the custom derating file to your design

To choose your custom derating file and apply the custom derating factors:

1. Right-click on the Smoke display.

2. From the pop-up menu, select Derating > Custom Derating Files.

   The Advanced Analysis Smoke tab dialog appears.



3. To add one or more files to the Custom Derating Files list box, click the New(Insert) button.
4. Browse and select the custom derating file.

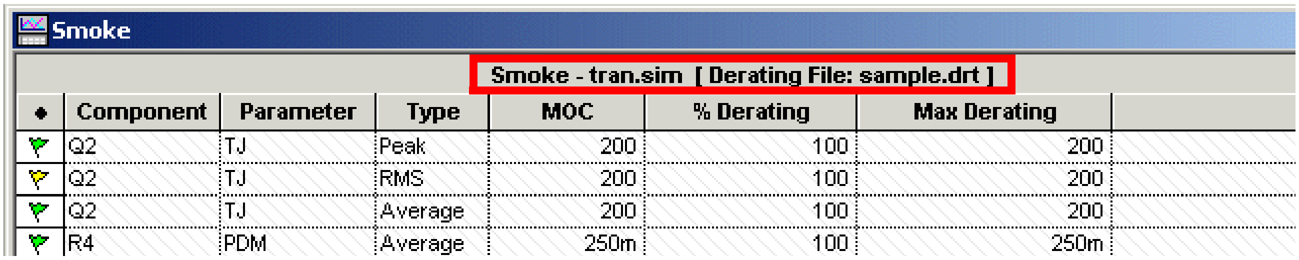
   The custom derating filename gets added in the Custom Derating Files list box.
5. In the Select derating type drop-down list, select the name of the derate file that you want to use during the smoke analysis.
6. Click OK.
7. Click the Run button (blue triangle).

   The Smoke data display title changes to "*Smoke - <profile name> [custom derate file name]*."

   Smoke results appear after the analysis in complete. The value of derate factors specified by you appear in the %Derating column.

Note: If the active derate file is different from the derate file used for the smoke results displayed, an asterix (*) symbol will be displayed along with the derate file name.

Consider an example where sample.drt was used to achieve the displayed smoke results.

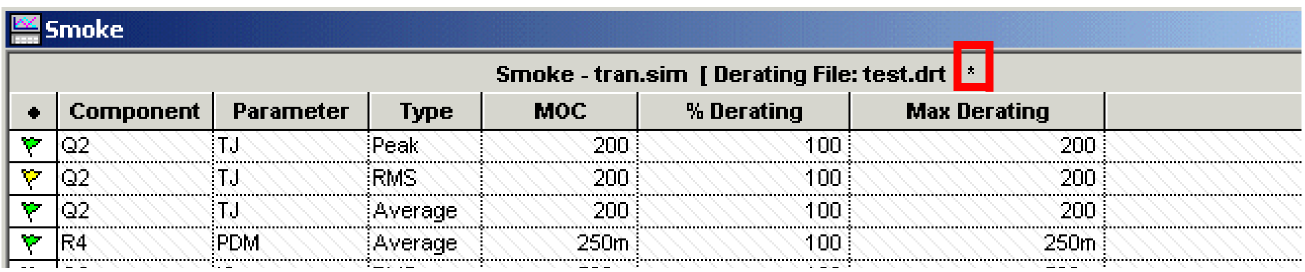


Smoke

Smoke - tran.sim [Derating File: sample.drt]

◆	Component	Parameter	Type	MOC	% Derating	Max Derating
	Q2	TJ	Peak	200	100	200
	Q2	TJ	RMS	200	100	200
	Q2	TJ	Average	200	100	200
	R4	PDM	Average	250m	100	250m

In this case, if you change the active derate file to test.drt or if you edit the existing sample.drt, an asterix (*) symbol will be displayed along with the derate file name.



Smoke

Smoke - tran.sim [Derating File: test.drt *

◆	Component	Parameter	Type	MOC	% Derating	Max Derating
	Q2	TJ	Peak	200	100	200
	Q2	TJ	RMS	200	100	200
	Q2	TJ	Average	200	100	200
	R4	PDM	Average	250m	100	250m

Caution

When you select a new derate file to be used for the smoke analysis, the contents of the %Derating column are updated with the new values only when you rerun the smoke analysis. Till you run the smoke analysis again, the values displayed in the %Derating column will be from the derate file used in the previous run.

Reading values from the derate file

To be able to use the custom derate file, add the DERATE_TYPE property on the design instance. The value assigned to the DERATE_TYPE property should match the Derate Type specified by you in the derate file.

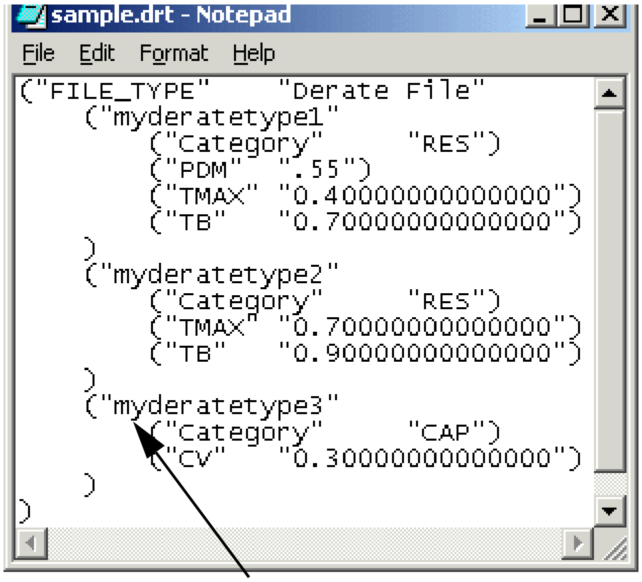
Consider a sample derate file, `sample.drt`. This derate file has two derate types for RES category, and one for capacitor. To use this derate file during the smoke analysis, load this file in Advanced

Analysis. See Adding the custom derating file to your design on page 183.

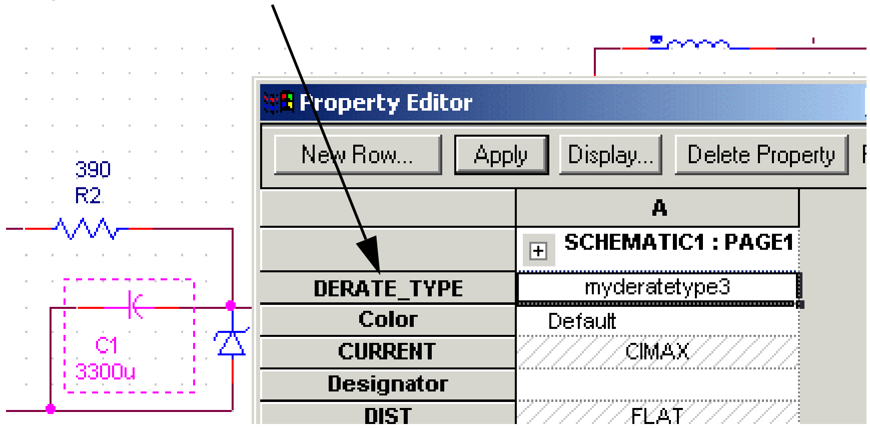
Before you can use the derate file successfully, you need to complete the following steps in Capture.

1. Select the component and right-click.
2. From the pop-up menu, select Edit properties.
3. In the Property Editor window, click the New Row button.
4. In the Add New Row dialog box, specify the name of the new property as DERATE_TYPE.

5. Specify the property value as `myderatetype3`, which is same as the derate type specified by you in the `sample.drt` file, and click OK.



Value assigned to the DERATE_TYPE is same as the derate type specified in the `.drt` file.



6. Regenerate the PSpice netlist. From the **PSpice** drop-down menu select **Create Netlist**.

7. Run the smoke analysis. From the **PSpice** drop-down menu, select **Advanced Analysis** and then choose **Smoke**.

8. In Advanced Analysis, ensure that the `sample .drt` file is loaded and active. Then run the smoke analysis.

Note: To know more about loading a customized derate file to your design, see Adding the custom derating file to your design on page 183.

Supported Device Categories

Table 5-1 Supported derate type

Device Category	Physical Device
RES	Resistor
CAP	Capacitor
IND	Inductor
DIODE	Diode
NPN	NPN Bipolar Junction Transistor
PNP	PNP Bipolar Junction Transistor
JFET	Junction FET
N-CHANNEL	N-Channel JFET
P-CHANNEL	P-Channel JFET
NMESFET	N-Channel MESFET
PMESFET	P-Channel MESFET
MOS	MOSFET
NMOS	N-Channel MOSFET
PMOS	P-Channel MOSFET
OPAMP	Operational Amplifiers
ZENER	Zener Diode
IGBT	Ins Gate Bipolar Transistor
VARISTOR	Varistor
OCNN	Octo Coupler using PNP transistor
OCNPN	Octo Coupler using NPN transistor
THYRISTOR	Thyristor
POS_REG	Positive Voltage Regulator
LED	Light Emitting Diode

Secondary Breakdown

The secondary breakdown value that Smoke uses in the safe operating area calculation for bipolar junction transistor is derived from the following:

- Maximum collector-emitter voltage (VCE)
- Secondary breakdown slope (SBSLP)
- Secondary breakdown intercept (SBINT)

The calculations are performed according to the equation

$$SBMAX = (SBINT)(V_{CE})SBSLP$$

Where,

SBSLP is equal to $(\Delta \log I_c)/(\Delta \log V_{ce})$

SBINT is equal to the collector current (Ic) at Vce

Dependence of Secondary Breakdown on Case Temperature

If the necessary parameters are specified in the device model, the secondary breakdown value calculated from the previous equation is derated according to case temperature.

Manufacturers typically account for the dependence of secondary breakdown on case temperature in one of the following ways:

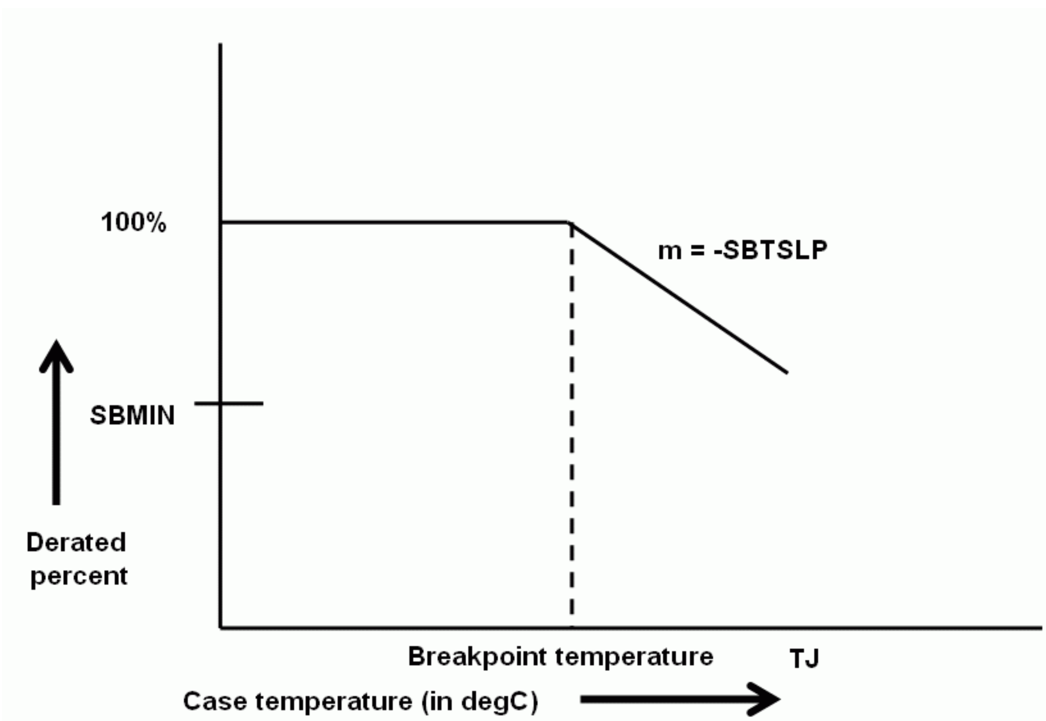
- Provide SBINT and SBSLP values that characterize secondary breakdown at optimum case temperature ($25°C$) and also provide a temperature derating curve that reflects how secondary breakdown limits decrease with increasing case temperature.
- Provide SBINT and SBSLP values that characterize secondary breakdown at a case temperature equal to the maximum junction temperature, describing the most conservative secondary breakdown effects.
- Provide SBINT and SBSLP values that characterize secondary breakdown at the optimum case temperature with no additional information about secondary breakdown at other case temperatures.

If temperature derating curves are available from manufacturers (the first alternative above), then two additional maximum operating conditions, SBTSLP and SBMIN, are included for devices in the device directory.

When SBTSLP and SBMIN are listed for a device, Smoke derives the temperature-derated secondary breakdown by first calculating a preliminary secondary breakdown using SBINT and SBSLP and then derating that value based on the actual case temperature.

The figure below shows the temperature-derating curve defined by SBTSLP, SBMIN, and TJ.

Figure 5-4 Temperature Derating Curve for SB



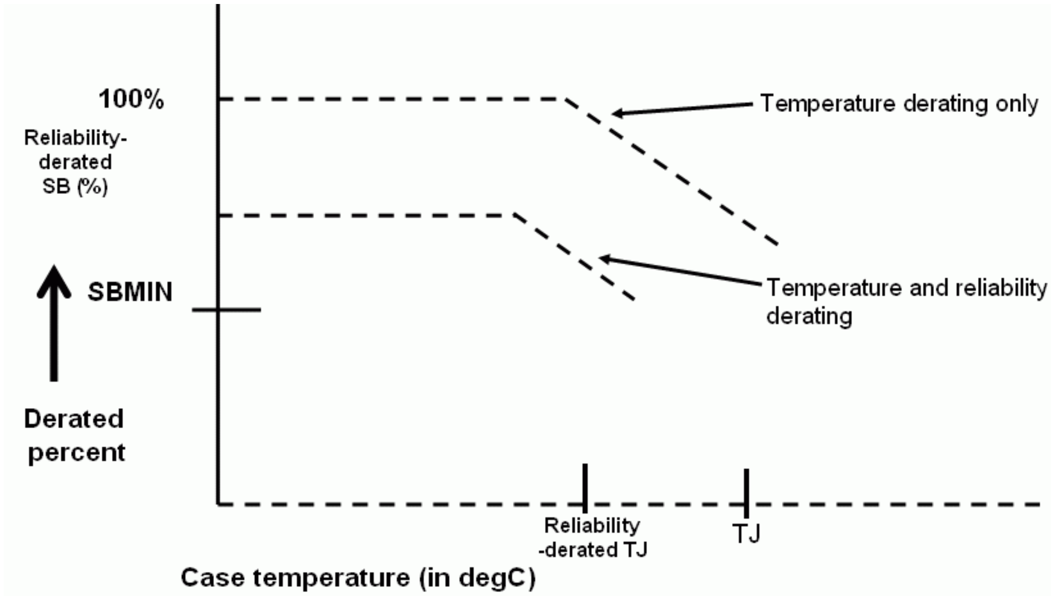
When the case temperature is equal to the maximum junction temperature (TJ), the safe operating limit for secondary breakdown is equal to SBMIN percent of the maximum secondary breakdown (SB) defined in the device model. At lower temperatures, the safe operating limit for secondary breakdown is equal to the maximum secondary breakdown value up to a breakpoint temperature and then decreases with increasing temperature until the case temperature reaches the maximum junction temperature (TJ). The slope at which the secondary breakdown limit decreases is –SBTSLP. The breakpoint temperature is defined by the values of SBMIN and SBTSLP.

When temperature derating curves are not available from manufacturers, SBTSLP and SBMIN are not included in the device library. In these cases, SBINT and SBSLP have values that characterize secondary breakdown at a case temperature equal to the maximum junction temperature, modeling the most conservative secondary breakdown limits. If you want to model the secondary breakdown limit at a temperature less than TJ, you can assign a derating factor greater than 1 to secondary breakdown (SB).

For components in the device library, you can determine whether or not secondary breakdown is derated for case temperature by checking the Diode Device Data Book. If SBTSLP, SBMIN, and TJ are listed, the temperature derating is calculated. If these values are not listed, then secondary breakdown is not derated for case temperature.

As with any other maximum operating conditions, you can add or change values for SBINT, SBSLP, SBTSLP, and SBMIN using Parameter Entry. You can assign a reliability derating to the maximum secondary breakdown value that Smoke calculates by adding a derating factor for secondary breakdown (SB) to the derating file, or to TJ by changing its value in the derating file, or both.

If you add reliability derating to SB or TJ, the temperature derating curve is adjusted as shown in the figure below.



SBMIN is always defined as a percentage of the maximum secondary breakdown (SB), not as a percentage of the reliability-derated secondary breakdown.

6

Monte Carlo

In this chapter

- Monte Carlo overview on page 195
- Monte Carlo strategy on page 196
- Monte Carlo procedure on page 199
- Example on page 209

Monte Carlo overview

Note: Monte Carlo analysis is available with the following products:

- PSpice[1] Advanced Optimizer Option
- PSpice Advanced Analysis

Monte Carlo predicts the behavior of a circuit statistically when part values are varied within their tolerance range. Monte Carlo also calculates yield, which can be used for mass manufacturing predictions.

Use Monte Carlo for:

- Calculating yield based on your specs
- Integrating measurements with graphical displays
- Displaying results in a probability distribution function (PDF) graph
- Displaying results in a cumulative distribution function (CDF) graph

1. Depending on the license available, you will access either PSpice or PSpice Simulator.

- Calculating statistical data
- Displaying measurement values for every Monte Carlo run

Monte Carlo strategy

Monte Carlo requires:

- Circuit components that are Advanced Analysis-ready

  See Chapter 2, Libraries.

- A circuit schematic and working PSpice simulation
- Measurements set up in PSpice

  See "Procedure for creating measurement expressions" on page 258.

Plan Ahead

Setting options

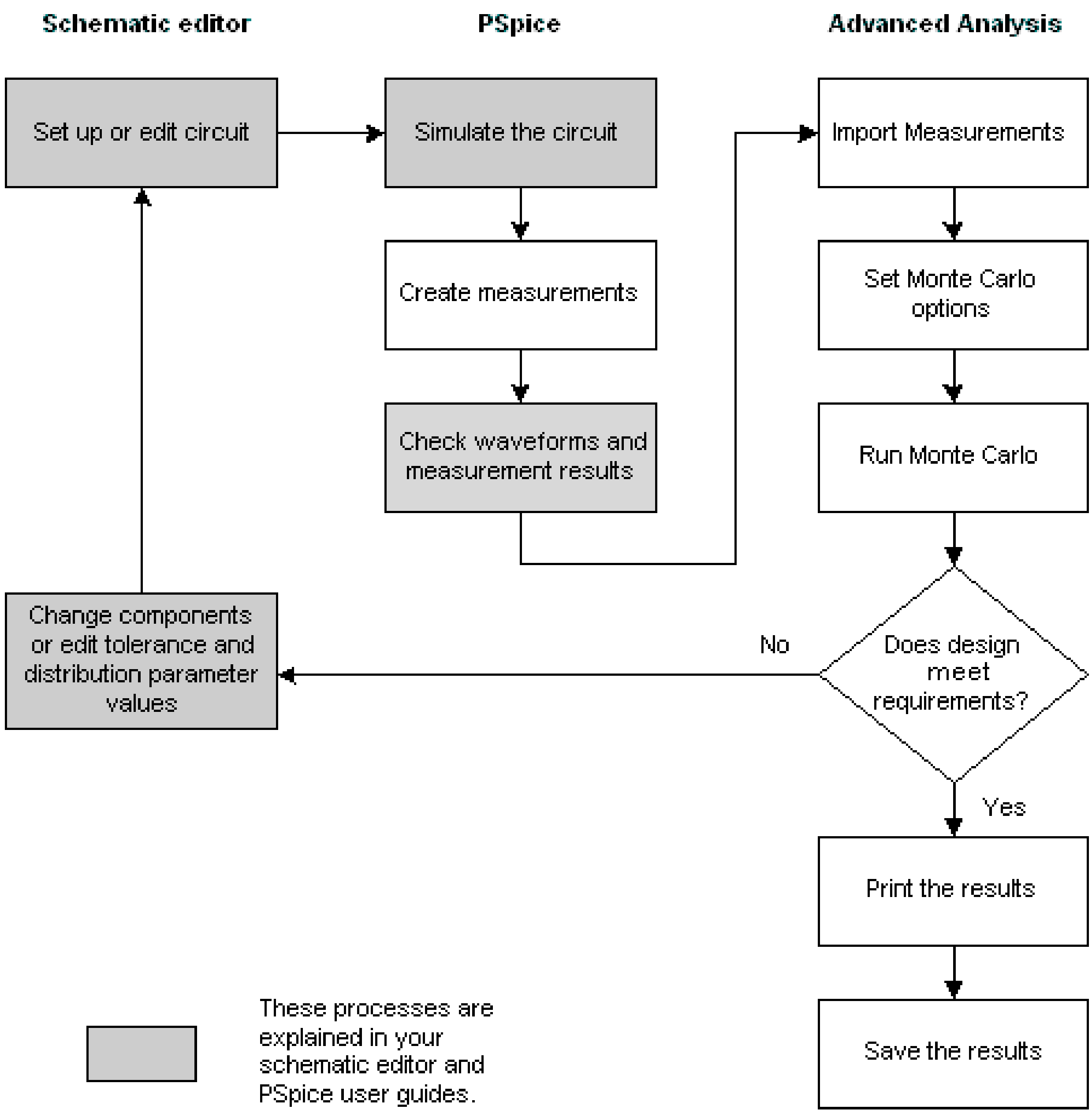
- Start with enough runs to provide statistically meaningful results.
- Specify a larger number of runs for a more accurate graph of performance distribution. This more closely simulates the effects of mass production.
- Start with a different random seed value if you want different results.
- Set the graph bin number to show the level of detail you want. Higher bin numbers show more detail, but need more runs to be useful.
- If you are planning an analysis of thousands of runs on a complex circuit, you can turn off the simulation data storage option to conserve disk space. However, at this setting, the simulation will run slower.
  To turn off data storage:
  - ❑ From the Advance Analysis menu select
    Edit / Profile Settings/ Simulation
  - ❑ From the Monte Carlo field, select **Save None**.

The simulation data will be overwritten by each new run. Only the last run's data will be saved.

Importing measurements

- Find the most sensitive measurements in Sensitivity and perform Monte Carlo analysis on those measurements only. Limiting Monte Carlo to only important measurements saves run time.

Workflow


Schematic editor
PSpice
Advanced Analysis
Set up or edit circuit
Simulate the circuit
Import Measurements
Create measurements
Set Monte Carlo options
Check waveforms and measurement results
Run Monte Carlo
Change components or edit tolerance and distribution parameter values
No
Does design meet requirements?
Yes
Print the results
Save the results
These processes are explained in your schematic editor and PSpice user guides.

Monte Carlo procedure

Setting up the circuit in the schematic editor

Starting out:

- Have a working circuit in the schematic tool.
- Circuit simulations and measurements should already be defined.

  The simulations can be Time Domain (transient), DC Sweep, and AC Sweep/Noise analyses.
- The circuit components you want to include in the data need to be Advanced Analysis-ready, with the tolerances of the circuit components specified.

  See Chapter 2, Libraries, for information about component tolerances.

To run advanced analysis monte carlo procedure:

1. From your schematic editor, open your circuit.
2. Run a PSpice simulation.

   Note: Advanced Analysis Monte Carlo does not use PSpice Monte Carlo settings.

   Note: You can run Advanced Analysis Monte Carlo on more than one simulation profile at once. However, if you have a multi-run analysis set up in PSpice (for example, a parametric sweep or a temperature sweep), Advanced Analysis Monte Carlo will reduce the simulation profile to one run before starting the Advanced Analysis Monte Carlo calculations. For temperature sweeps, the first temperature value in the list will be used for the Advanced Analysis Monte Carlo calculations.
3. Check your key waveforms in PSpice and make sure they are what you expect.
4. Test your measurements and make sure they have the results you expect.

For information on circuit layout and simulation setup, see your schematic editor and PSpice user guides.

Note: For information on setting up measurements, see “Procedure for creating measurement expressions” on page 258.

Setting up Monte Carlo in Advanced Analysis

Opening Monte Carlo

➡ From the **PSpice** menu in your schematic editor, select **Advanced Analysis / Monte Carlo**.

The Advanced Analysis Monte Carlo tool opens.

Importing measurements from PSpice

1. In the Statistical Information table, click on the row containing the text “Click here to import a measurement created within PSpice.”

   The **Import Measurement(s)** dialog box appears.

2. Select the measurements you want to include.

   For more information, see Importing measurements from PSpice on page 213 in the Example section.

Setting Monte Carlo options

From the Advanced Analysis **Edit** menu, select **Profile Settings**, click the **Monte Carlo** tab, and enter the following Monte Carlo options:

- Number of Runs

  This is the number of times the selected simulation profiles will be run. For each run, component parameters with tolerances will be randomly varied. Run number one uses nominal component parameter values. The maximum number of runs can be set between 1 to 5000.

- Starting Run Number

The default starting run number is one. This is the nominal run. If the random seed value is kept constant, then you can change the starting run number in order to duplicate a partial Monte Carlo simulation. You can use this to isolate specific random results which are of particular interest, without having to run an entire Monte Carlo simulation again.

- Random Seed Value

  The random number generator uses this value to produce a sequence of random numbers. Change the seed in order to produce a unique random sequence for each Monte Carlo simulation. If the seed and device properties are not changed, then the same sequence of random numbers will be generated each time a Monte Carlo analysis is done. You can use this procedure to reproduce a random simulation.

- Number of Bins

  This value determines the number of divisions in the histogram. A typical value is one tenth of the number of runs. The minimum value is one and the maximum value is 2000.

Running Monte Carlo

Monte Carlo calculates a nominal value for each measurement using the original parameter values.

After the nominal runs, Monte Carlo randomly calculates the value of each variable parameter based on its tolerance and a flat (uniform) distribution function. For each profile, Monte Carlo uses the calculated parameter values, evaluates the measurements, and saves the measurement values.

Monte Carlo repeats the calculations for the specified number of runs, then calculates and displays statistical data for each measurement.

For more detail on the displayed statistical data, see Example's section "Reviewing Monte Carlo data" on page 202.

Controlling MonteCarlo run

The MonteCarlo analysis can only be run if tolerances are specified for the component parameters. In case you want to prevent running these analysis on a component, you can do so by using the TOL_ON_OFF property.

In the schematic design, attach the TOL_ON_OFF property to the device instance for which you do not want to perform the Sensitivity and MonteCarlo analysis. Set the value of the TOL_ON_OFF property to OFF. When you set the property value as OFF, the tolerances attached to the component parameters will be ignored and therefore, the component parameters will not be available for analysis.

Reviewing Monte Carlo data

You can review Monte Carlo results on two graphs and two tables:

- Probability density function (PDF) graph
- Cumulative distribution function (CDF) graph
- Statistical Information table, in the **Statistics** tab
- Raw Measurements table, in the **Raw Meas** tab

Reviewing the Statistical Information table

For each run, Monte Carlo randomly varies parameter values within tolerance and calculates a single measurement value. After all the runs are done, Monte Carlo uses the run results to perform statistical analyses.

1. Click the **Statistics** tab to bring the table to the foreground.
2. Select a measurement row in the Statistical Information table.

   A black arrow appears in the left column and the row is highlighted. The data in the graph corresponds to the selected measurement only.

You can review results reported for each measurement:

Column heading...	Means...
Cursor Min	Measurement value at the cursor minimum location.
Cursor Max	Measurement value at the cursor maximum location.
Yield (in percent)	The number of runs that meet measurement specifications (represented by the cursors) versus the total number of runs in the analysis. Used to estimate mass manufacturing production efficiency.
Mean	The average measurement value based on all run values. See Raw Measurement table for run values.
Std Dev	Standard deviation. The statistically accepted meaning for standard deviation.
3 Sigma (in percent)	The number of measurement run values that fall within the range of plus or minus 3 Sigma from the mean
6 Sigma (in percent)	The number of measurement run values that fall within the range of plus or minus 6 Sigma from the mean
Median	The measurement value that occurs in the middle of the sorted list of run values. See Raw Measurement table for run values

Reviewing the PDF graph

A PDF graph is a way to display a probability distribution. It displays the range of measurement values along the x-axis and the number of runs with those measurement values along the y-axis.

1. Select a measurement row in the Statistical Information table.
2. If the PDF graph is not already displayed, right click the graph and select **PDF Graph** from the pop-up menu.

   The corresponding PDF graph will display all measurement values based on the Monte Carlo runs.
3. Right click the graph to select zoom setting, another graph type, and y-axis units.

   A pop-up menu appears.

- ❑ Select **Zoom In** to focus on a small range of values.
- ❑ Select **CDF Graph** to toggle from the default PDF graph to the CDF graph.
- ❑ Select **Percent Y-axis** to toggle from the default absolute y-axis Number of Runs to **Percent of Runs**.

4. To change the number of bins on the x-axis:

From the **Edit** menu, select **Profile Settings**, click the **Monte Carlo** tab, and typing a new number in the **Number of Bins** text box.

If you want more bars on the graph, specify more bins—up to a maximum of the total number of runs. Higher bin numbers show more detail, but require more runs to be useful.

Reviewing the CDF graph

The CDF graph is another way to display a probability distribution. In mathematical terms, the CDF is the integral of the PDF.

1. Select a measurement row in the Statistical Information table.

2. If the CDF graph is not already displayed, right click on the PDF graph and select **CDF Graph** from the pop-up menu.

The statistical display for the cumulative distribution function is shown on the CDF graph.

3. Right click the graph to select zoom setting and y-axis units.

A pop-up menu will appear.

- ❑ Select **Zoom In** to focus on a small range of values.
- ❑ Select **PDF Graph** to toggle from the current CDF graph to the default PDF graph.
- ❑ Select **Percent Y-axis** to toggle from the default absolute y-axis Number of Runs to **Percent of Runs**.

4. Change the number of bins on the x-axis by going to the **Edit** menu, selecting **Profile Settings**, clicking the **Monte Carlo** tab, and typing a new number in the **Number of Bins** text box.

If you want more bars on the graph, specify more bins, up to a maximum of the total number of runs. Higher bin numbers show more detail, but require more runs to be useful.

Working with cursors

➡ To change a cursor location on the graph, click the cursor to select it and click the mouse in a new spot on the graph. A selected cursor appears red.

The cursor’s location on the graph changes, and the measurement min or max values in the Statistical Information table are updated. A new calculated yield displays.

Restricting calculation range

To restrict the statistical calculations displayed in the Statistical Information table to the range of samples within the cursor minimum and maximum range, set the cursors in their new locations and select the restrict calculation range command from the right click pop-up menus.

1. Change cursors to new locations.

   See Working with cursors above.

2. Right click in the graph or in the Statistical Information table and select **Restrict Calculation Range** from the pop-up menu.

   The cross-hatched range of values that appears on the graph is the restricted range.

Reviewing the Raw Measurements table

The Raw Measurements table is a read-only table that has a one-to-one relationship with the Statistical Information table. For every measurement row on the Raw Measurements table, there is a corresponding measurement row on the Statistical Information table. The run values in the Raw Measurements table are used to calculate the yield and statistical values in the Statistical Information table.

1. Click the **Raw Meas** tab.

   The Raw Measurements table appears.

2. Select a row and double click the far left row header.

   The row of data is sorted in ascending or descending order.

   Note: Copy and paste the row of data to an external program if you want to further manipulate the data. Use the **Edit** menu or the right click pop-up menu copy and paste commands.

3. From the **View** menu, select **Log File** / **Monte Carlo** to view the component parameter values for each run.

Controlling Monte Carlo

If you do not achieve your goals in the first Monte Carlo analysis, there are several things you can do to fine-tune the process.

Pausing, stopping, and starting

Pausing and resuming

To review preliminary results on a large number of runs:

➡ Click on the top toolbar when the output window indicates approximately Monte Carlo run 50.

The analysis stops at the next interruptible point, available data is displayed and the last completed run number appears in the output window.

➡ Click the depressed or to resume calculations.

Stopping

➡ Click on the top toolbar.

If a Monte Carlo analysis has been stopped, you cannot resume the analysis.

Starting

➡ Click to start or restart.

Changing circuit components or parameters

If you do not get the results you want, you can return to the schematic editor and change circuit parameters.

1. Try a different component for the circuit or change the tolerance parameter on an existing component.
2. Rerun the PSpice simulation and check the results.
3. Rerun Monte Carlo using the settings saved from the prior analysis.
4. Review the results.

Controlling measurement specifications

If you do not get the results you want and your design specifications are flexible, you can add, edit, delete or disable a measurement and rerun Monte Carlo analysis.

Cells with cross-hatched backgrounds are read-only and cannot be edited.

- To exclude a measurement from the next optimization run, click the in the Statistical Information table, which removes the check mark.
- To edit a measurement, click on the measurement you want to edit, then click .
- To edit a measurement specification Min or Max, click the minimum or maximum cursor on the graph (the selected cursor turns red), then click the mouse in the spot you want.

  The new value will display in the **Cursor Min** or **Cursor Max** column in the Statistical Information table.
- To add a new measurement, click on the row that reads “Click here to import a measurement...”

  Note: For instructions on setting up new measurements, see “Procedure for creating measurement expressions” on page 258.

➡ To export a new measurement to Optimizer or Monte Carlo, select the measurement and right click on the row containing the text “Click here to import a measurement created within PSpice.”

Select **Send To** from the pop-up menu.

Printing results

➡ Click  .

Or

From the File menu, select Print.

To print information from the Raw Measurements table on the **Raw Meas** tab, copy and paste to an external program and print from that program. You can also print the Monte Carlo Log File, which contains more detail about measurement parameters. From the **View** menu select **Log File**, **Monte Carlo**.

Saving results

➡ Click  .

Or:

From the **File** menu, select **Save**.

The final results will be saved in the Advanced Analysis profile (.aap).

Example

This example uses the tutorial version of RFAmp located at:

`<target_directory>\PSpice\tutorial\capture\pspiceaa\rfamp`

`<target_directory>\PSpice\tutorial\concept\pspiceaa\rfamp`

The circuit is an RF amplifier with 50-ohm source and load impedances. It includes the circuit schematic, PSpice simulation profiles, and measurements.

For a completed example see:

`<target_directory>\PSpice\Capture_Samples\AdvAnls\RFAmp directory.`

For a completed example see:

`<target_directory>\PSpice\Concept_Samples\AdvAnls\RFAmp directory.`

Setting up the circuit in the schematic editor

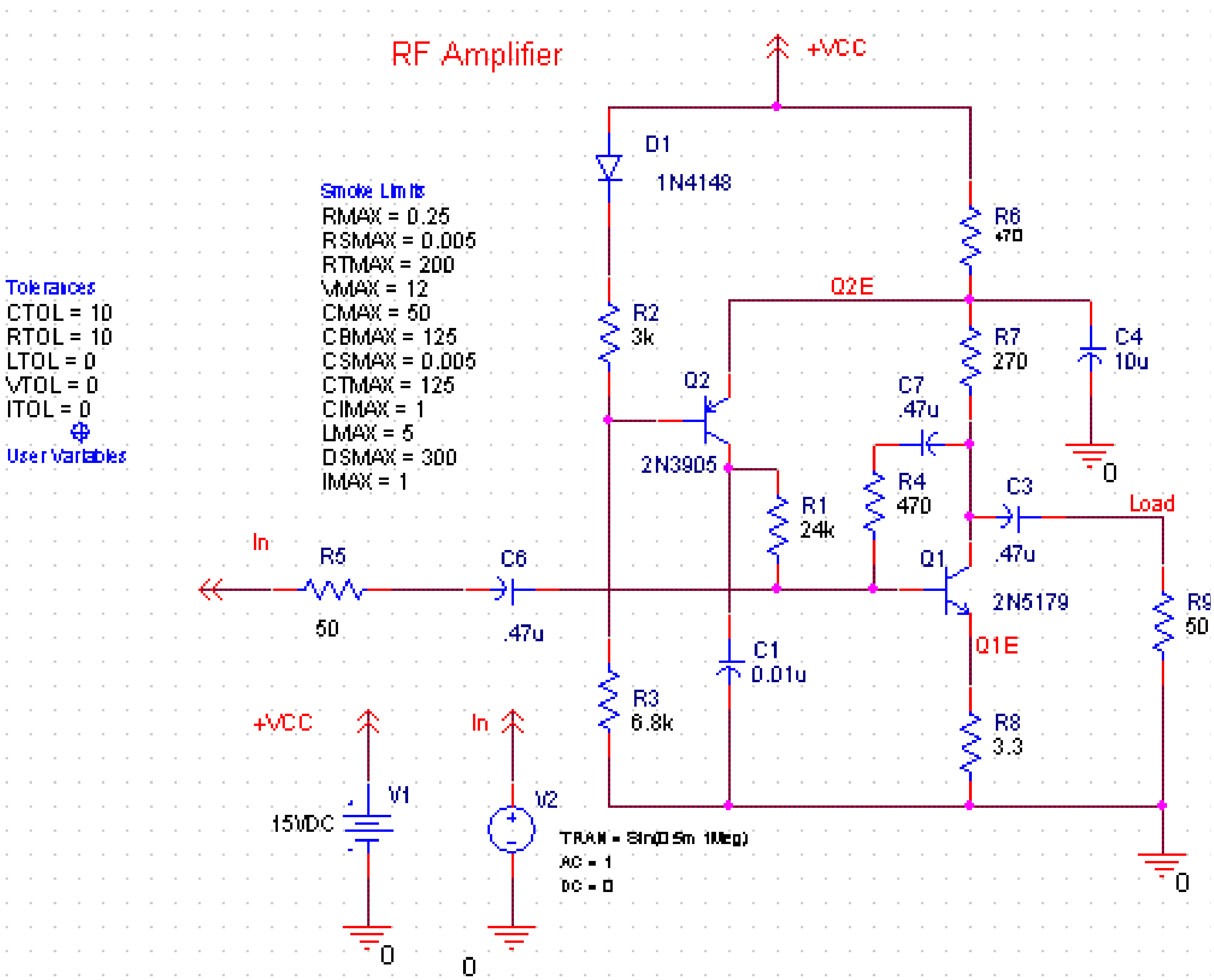
1. In your schematic editor, browse to the RFAmp tutorials directory.

   `<target directory>\PSpice\tutorial\Capture\pspiceaa\rfamp`

   `<target directory>\PSpice\tutorial\Concept\pspiceaa\rfamp`

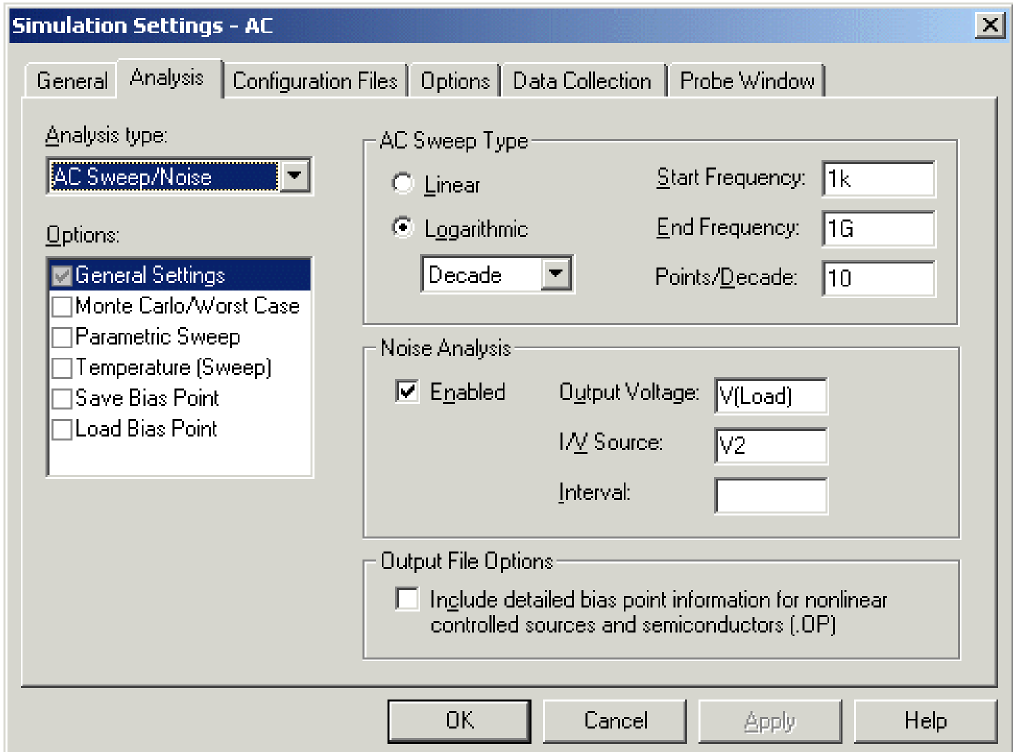
2. Open the RFAmp project.

The RF amplifier circuit example



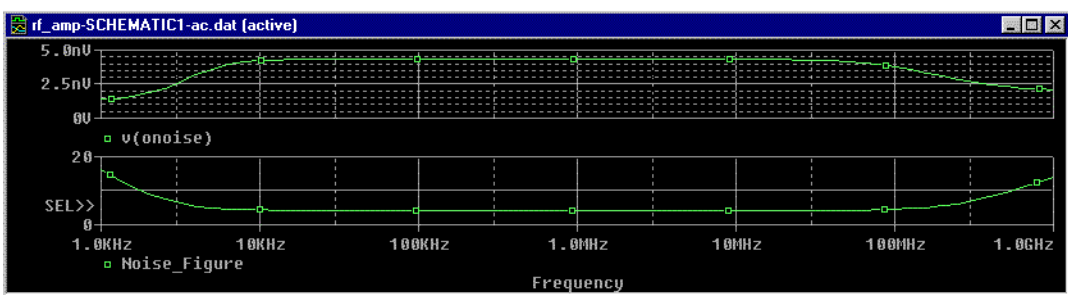
3. Select the SCHEMATIC1-AC simulation profile.

The AC simulation included in the RF amp example



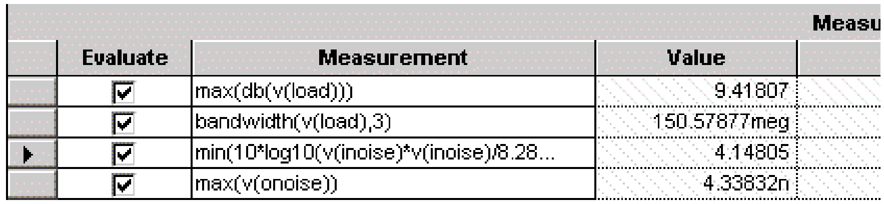
1. Click  to run a PSpice simulation.
2. Review the results.

   The waveforms in PSpice are what we expected.



The measurements in PSpice give the results we expected.

In PSpice, View Measurement Results



	Evaluate	Measurement	Value
	☑	max(db(v(load)))	9.41807
	☑	bandwidth(v(load),3)	150.57877meg
▶	☑	min(10*log10(v(inoise)*v(inoise)/8.28...	4.14805
	☑	max(v(onoise))	4.33832n

Setting up Monte Carlo in Advanced Analysis

Opening Monte Carlo

➡ From the schematic editor **PSpice** menu, select **Advanced Analysis** / **Monte Carlo**.

The Advanced Analysis Monte Carlo tool opens.

PDF/CDF graph

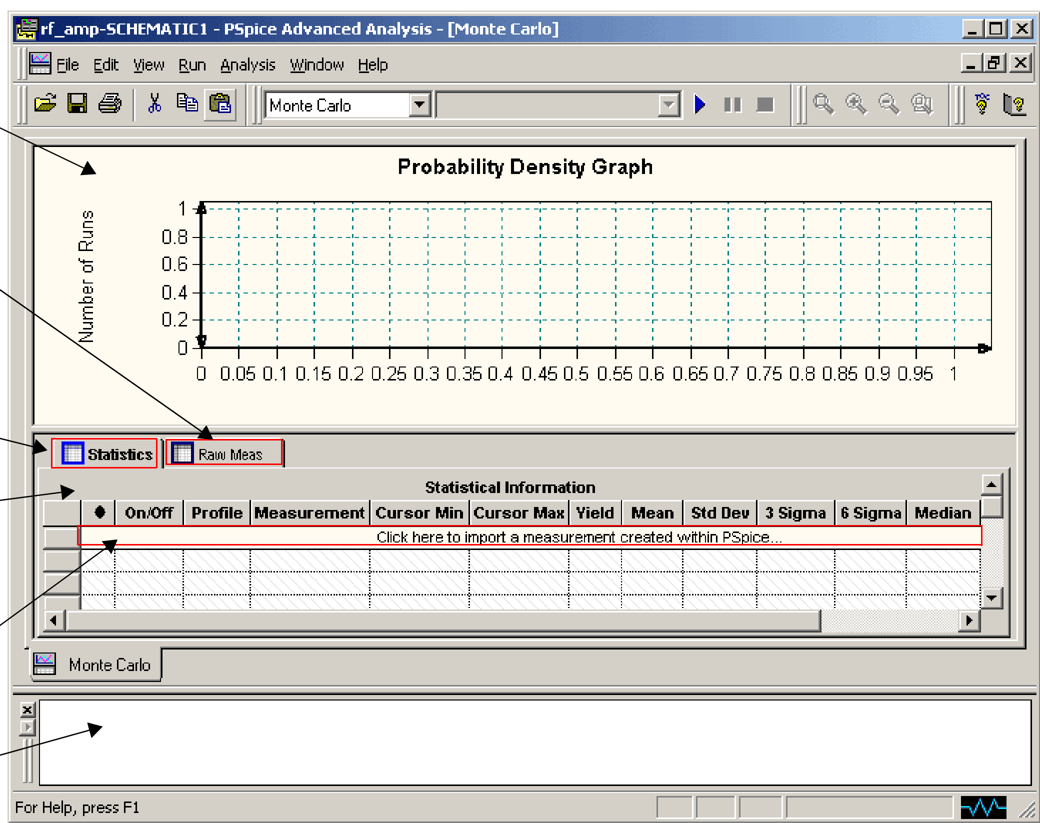
Raw Measurements tab

Statistics tab

Statistical Information table

Click to import more measurements

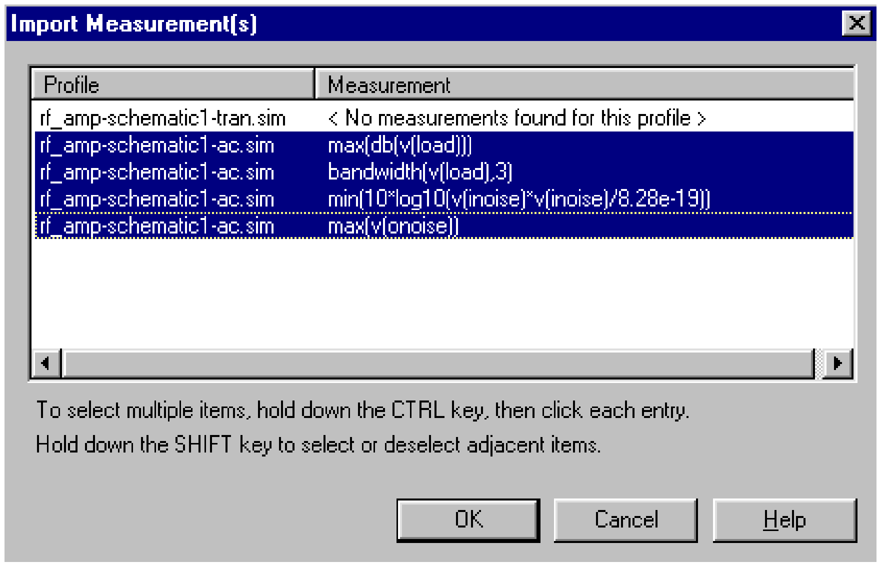
Output window



Importing measurements from PSpice

1. In the Statistical Information table, click on the row containing the text "Click here to import a measurement created within PSpice."

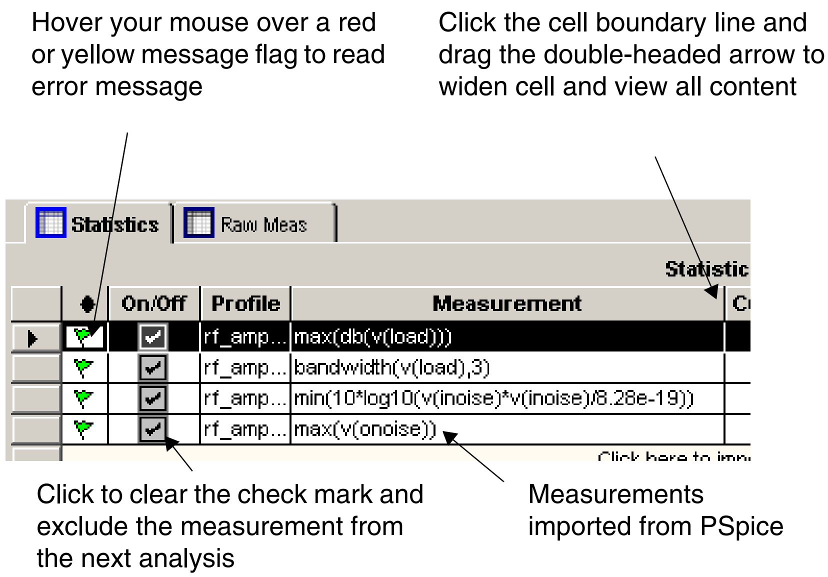
   The **Import Measurement(s)** dialog box appears.



2. Select the four measurements:

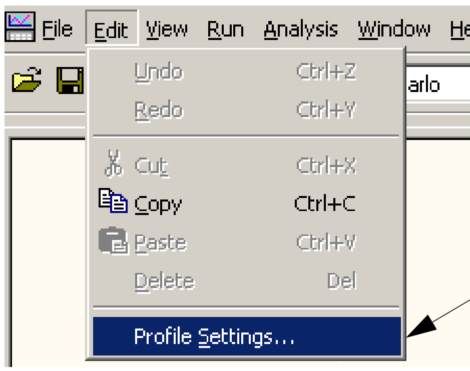
   - Max(DB(V(Load)))
   - Bandwidth(V(Load),3)
   - Min(10*Log10(V(inoise)*V(inoise)/8.28e-19))
   - Max(V(onoise))

3. Click **OK**.


Hover your mouse over a red or yellow message flag to read error message
Click the cell boundary line and drag the double-headed arrow to widen cell and view all content
Statistics
Raw Meas
Statistic
On/Off
Profile
Measurement
rf_amp...
max(db(v(load)))
bandwidth(v(load),3)
min(10*log10(v(inoise)*v(inoise)/8.28e-19))
max(v(onoise))
Click to clear the check mark and exclude the measurement from the next analysis
Measurements imported from PSpice

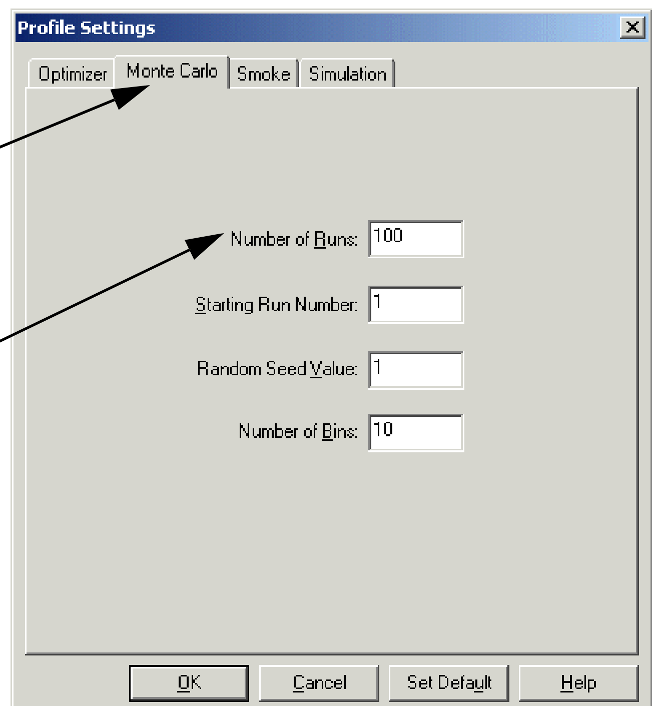
Setting Monte Carlo options

1. From the Advanced Analysis **Edit** menu, select **Profile Settings**, click the **Monte Carlo** tab, and enter the values shown in the dialog box.



Select Profile Settings from the Edit menu to bring up Monte Carlo options

Click to select Monte Carlo settings

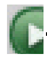


Set Number of Runs to 100

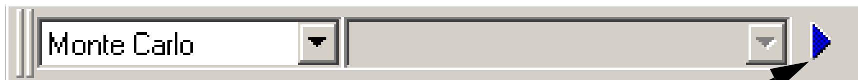
2. Click **OK**.

Running Monte Carlo

Starting the analysis

1. Click .

Select Monte Carlo from the drop-down list



Click to start a Monte Carlo analysis

The Monte Carlo analysis begins. The messages in the output window give you the status.

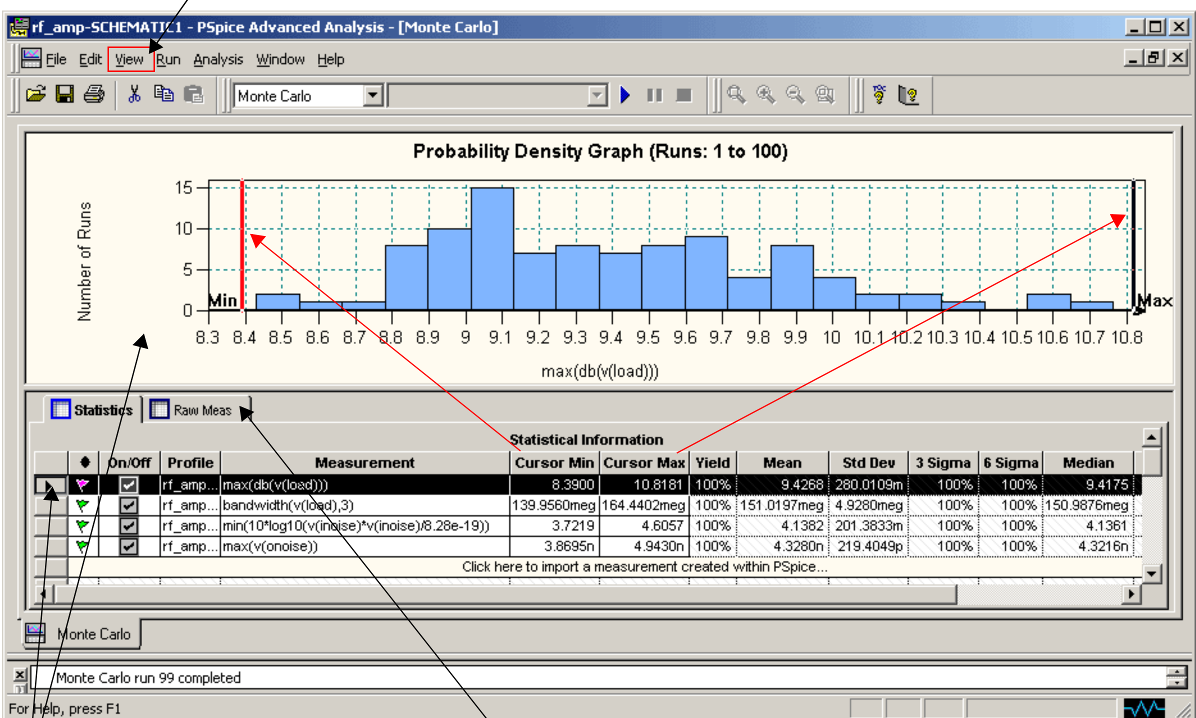
Monte Carlo calculates a nominal value for each measurement using the original parameter values.

After the nominal runs, Monte Carlo randomly calculates the value of each variable parameter based on its tolerance and a flat (uniform) distribution function. For each profile, Monte Carlo uses the calculated parameter values, evaluates the measurements, and saves the measurement values.

Monte Carlo repeats the above calculations for the specified number of runs, then calculates and displays statistical data for each measurement.

Ten bins of measurement data are displayed on the graph.

From the View menu, select Log File / Monte Carlo
to see parameter values and other details



The selected measurement's min, max, and other run results are plotted on the PDF graph

Click Raw Meas tab for 100 run results

Reviewing Monte Carlo data

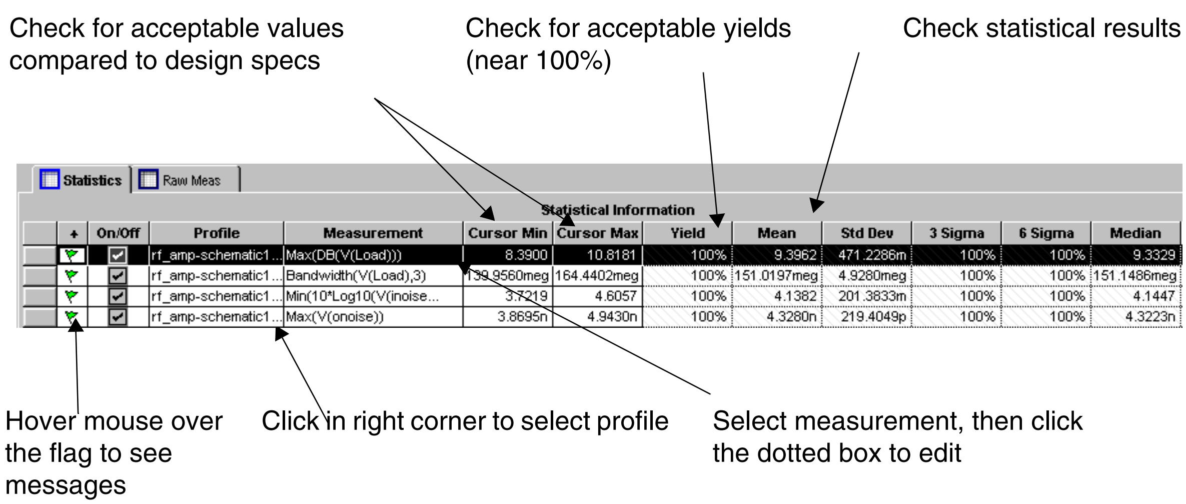
The Statistics tab is already in the foreground and the Statistical Information table contains results for the four imported measurements.

➡ Select the **Max(DB(V(load)))** measurement row.

A black arrow appears in the left column and the row is highlighted. The values in the PDF graph correspond to this measurement.

For each Monte Carlo run, Monte Carlo randomly varies parameter values within tolerance and calculates a single measurement value. After all the runs are done, Monte Carlo uses the run results to perform statistical analyses. The following statistical results are reported for our example: Mean, Std Dev, 3 Sigma, 6 Sigma, and Median.

In addition a yield is calculated and reported.

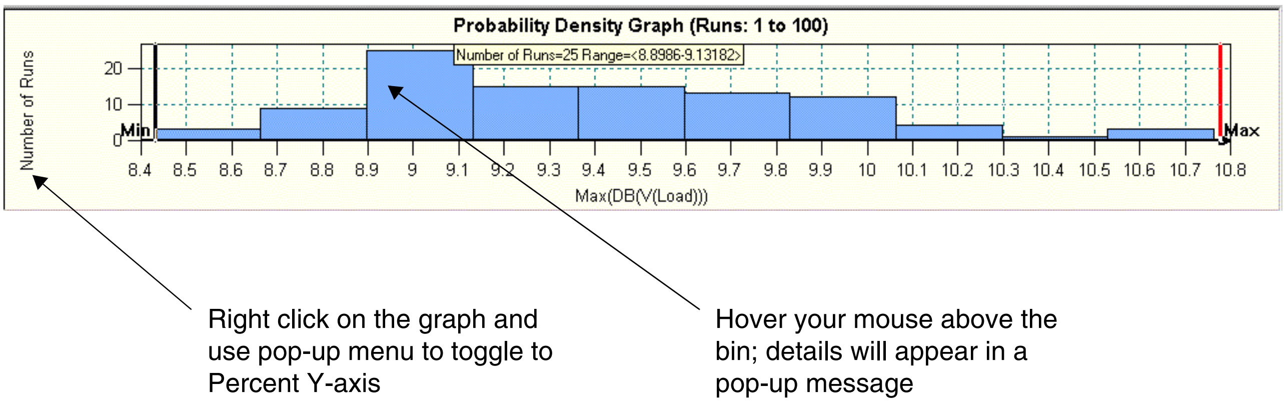


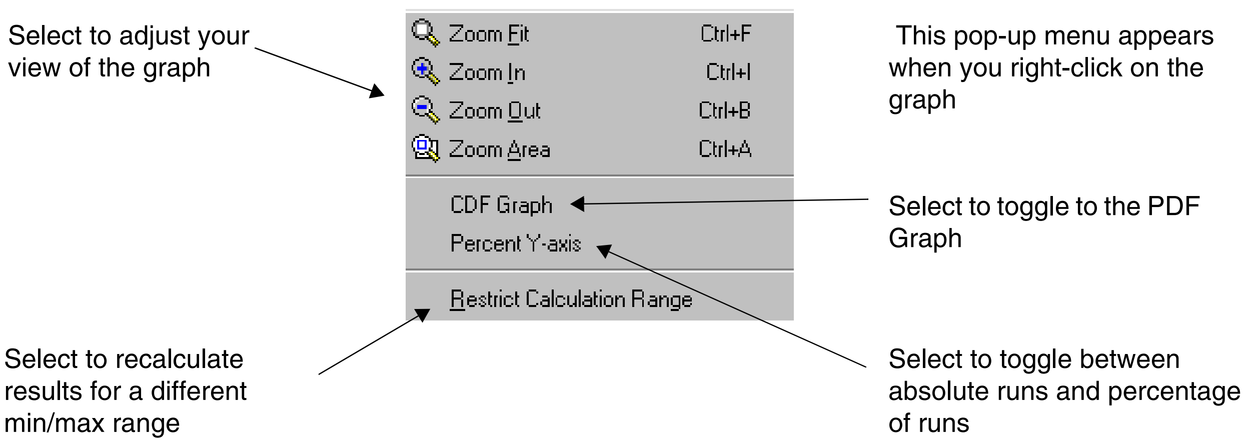
+	On/Off	Profile	Measurement	Cursor Min	Cursor Max	Yield	Mean	Std Dev	3 Sigma	6 Sigma	Median
		rf_amp-schematic1...	Max(DB(V(Load)))	8.3900	10.8181	100%	9.3962	471.2286m	100%	100%	9.3329
		rf_amp-schematic1...	Bandwidth(V(Load),3)	139.9560meg	164.4402meg	100%	151.0197meg	4.9280meg	100%	100%	151.1486meg
		rf_amp-schematic1...	Min(10*Log10(V(inoise...	3.7219	4.6057	100%	4.1382	201.3833m	100%	100%	4.1447
		rf_amp-schematic1...	Max(V(onoise))	3.8695n	4.9430n	100%	4.3280n	219.4049p	100%	100%	4.3223n

Reviewing the PDF graph

The PDF graph is a bar chart. The x-axis shows the measurement values calculated for all the Monte Carlo runs.

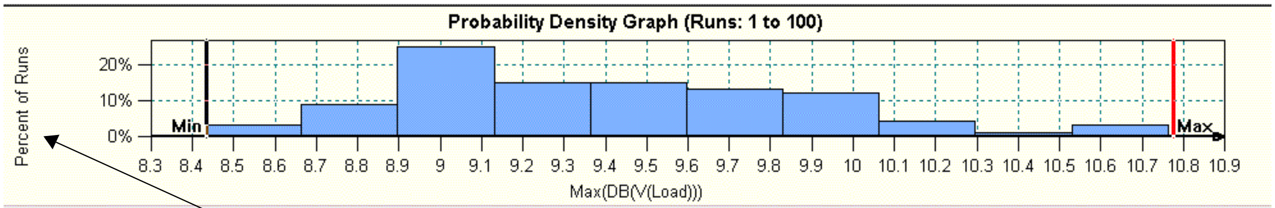
The y-axis shows the number of runs with measurement results between the x-axis bin ranges. The statistical display for this measurement's probability density function is shown on the PDF graph.





1. Right click on the graph and select **Percent Y-axis** from the pop-up menu.

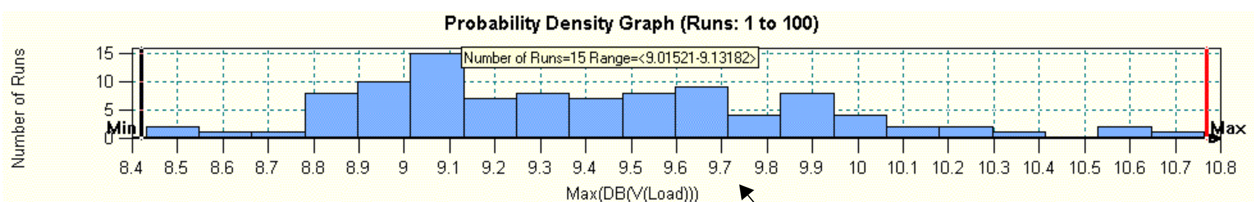
The Y-axis units changes from **Number of Runs** to **Percent of Runs**.



Y-axis changed to Percent of Runs

2. From the **Edit** menu, select **Profile Settings**, click the **Monte Carlo** tab, select the **Number of Bins** text box and type the number 20 in place of 10.

   Notice the higher level of detail on the PDF graph.



Same statistical results, but 20 bins are specified: twice as many as first PDF graph

3. Right click on the graph and from the pop-up menu select **Zoom In** to view a specific range.
4. Select **Zoom Fit** to show the entire graph with cursors.
5. Click the **Max** cursor to select it (it turns red when selected), then click the mouse in a new location on the x-axis.

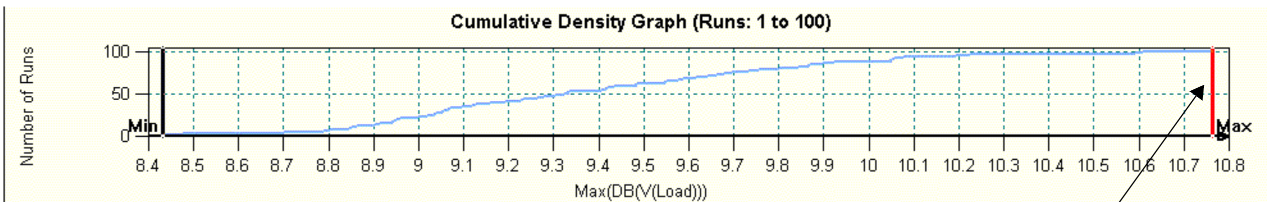
   The cursor's location changes and the max value and yield numbers are updated in the Statistical Information table.

   Note: Moving the cursor does not update the rest of the statistical results for this new min / max range. Use **Restrict Calculation Range** to recalculate the rest of the statistical results for this min / max range.

Reviewing the CDF graph

The CDF graph is a cumulative stair-step plot.

1. Select the **Max(DB(V(Load)))** measurement in the Statistical Information table.
2. Right click on the PDF graph and select **CDF Graph** from the pop-up menu.



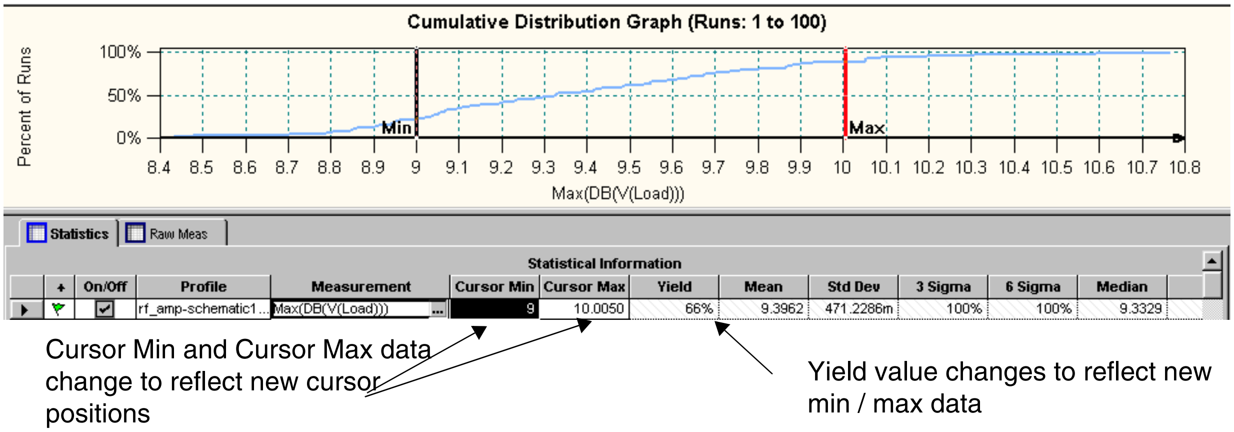
CDF graph with max cursor selected; before cursor is moved for restricted range calculation

3. Right click on the graph and select **Zoom In** to view a specific range.
4. Click the **Max** cursor to select the cursor.

   The Max cursor turns red.
5. Click the mouse at 10 on the x-axis.

   The cursor moves to the new position on the x-axis.
6. Click the **Min** cursor and click the mouse at 9 on the x-axis.

When you change the cursor location the min, max, and yield values are updated on the Statistical Information table.



Restricting the calculation range

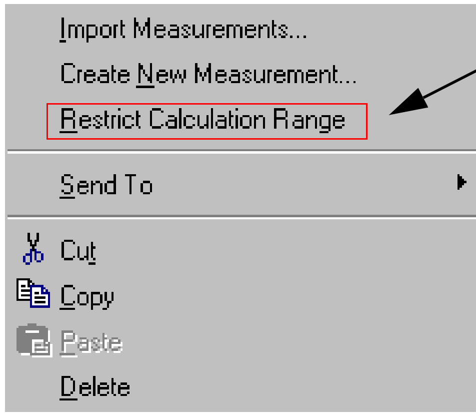
To quickly view statistical results for a different min / max range, use the **Restrict Calculation Range** command.

1. Set the graph cursors at **Min** = 9 and **Max** = 10.

   Or:

   Edit the min or max values in the Statistical Information table.

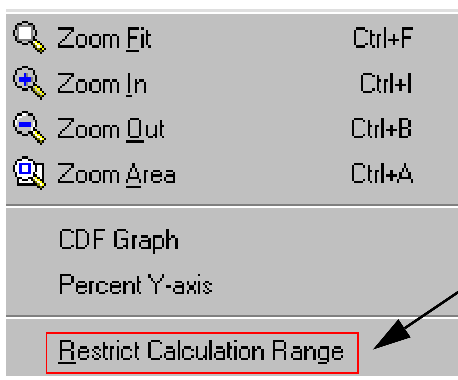
2. Right click in the table or on the graph and select **Restrict Calculation Range** from the pop-up menu.



Right click in the table for this pop-up menu

Or

Select **Restrict Calculation Range** from either menu



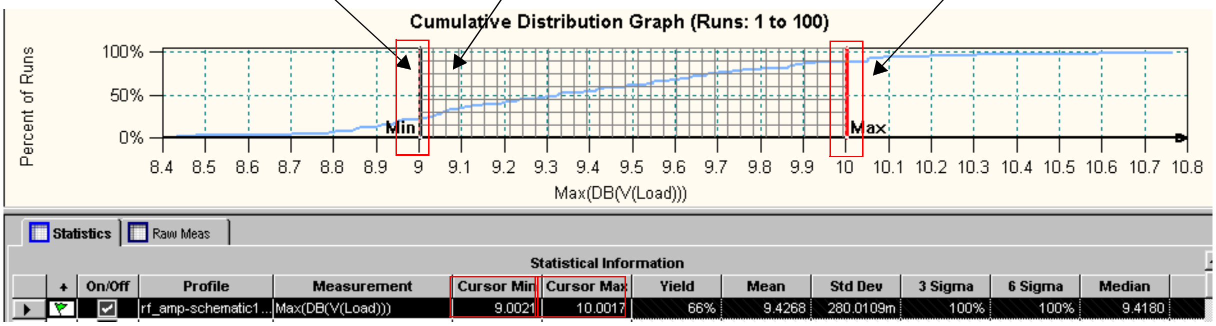
Right click in the graph for this pop-up menu

Monte Carlo recalculates the statistics and only includes the restricted range of values.

Min cursor changed to 9

Restricted range is cross-hatched

Max cursor changed to 10



Editing the Cursor Min and Max cells also changes the range

Note the new results for statistics based on the restricted range

Raw Measurements Table

This read-only table has a one-to-one relationship with the Statistical Information Table. For every row on this table, there is a corresponding row on the other table where the statistics are displayed.

1. Click the **Raw Meas** tab.

   The Raw Measurements table appears.

2. Select the **Max(DB(V(load)))** measurement row and double click the far left row header.

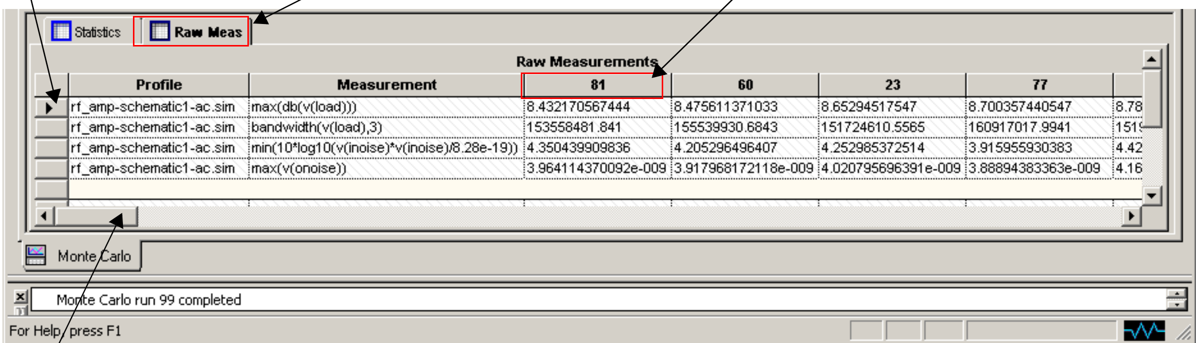
   The row run data is sorted in ascending order.

   Note: If you want to use the data in an external program, you can copy and paste a row of data.

Double click on a row header to sort run data

Click on the Raw Meas tab to select

Run 81 has the lowest measurement value



Scroll to see all run values

Data rows can be copied and pasted to external programs

Controlling Monte Carlo

Pausing, stopping, and starting

Click to start

Click to stop

Click to pause

Pausing and resuming

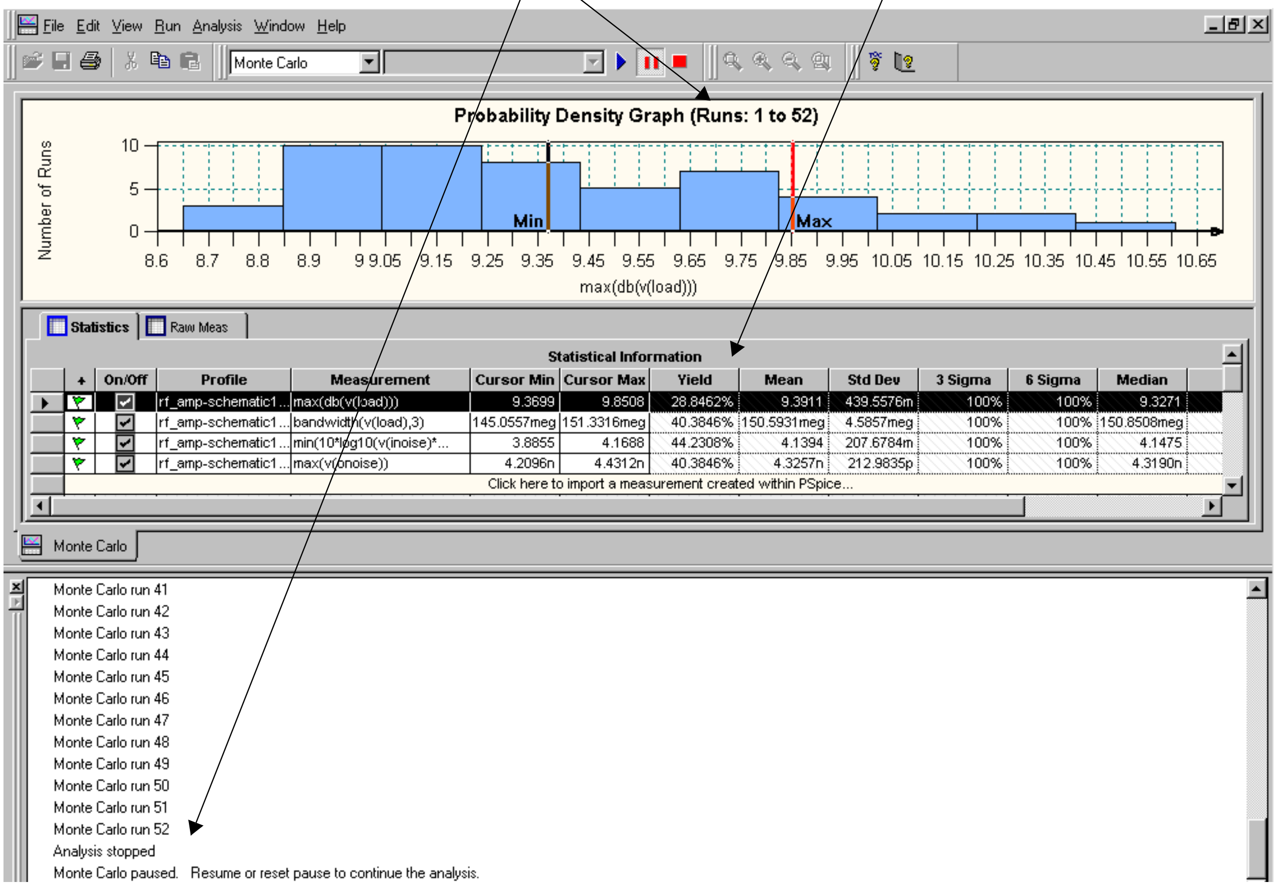
1. Click ❚❚ on the top toolbar.

   The analysis stops, available data is displayed, and the last completed run number appears in the output window.

2. Click or to resume calculations.

The title and the messages in the output window show how the number of runs made before pausing

Partial results: compare these with final 100-run results



Stopping

➡ Click on the top toolbar.

Note: Mont Carlo does not save data from a stopped analysis. After stopping, you cannot resume the same analysis.

Starting

➡ Click to start or restart.

Changing components or parameters

When running other examples, if you do not get the results you want, go to the schematic editor and change circuit information.

1. Try a different component for the circuit

   Or:

   Change the tolerance of a parameter on an existing component.

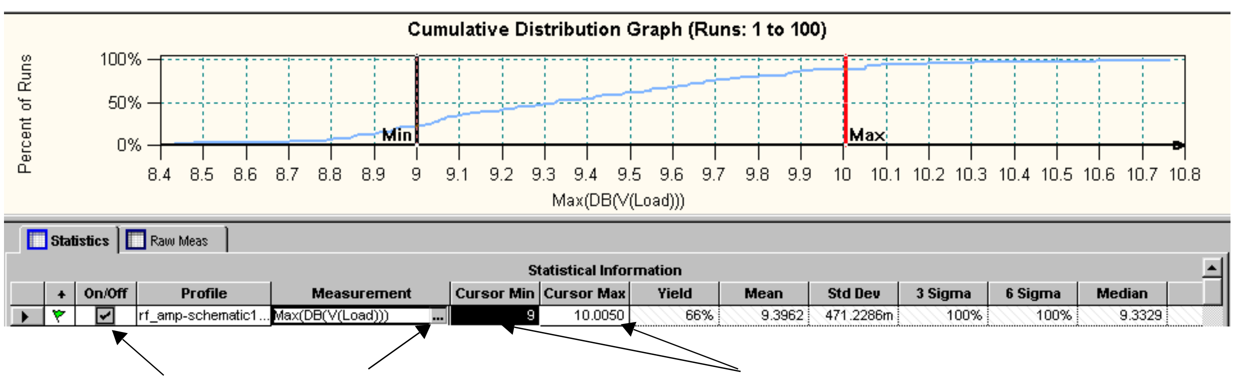
2. Rerun the PSpice simulation and verify that the results are what you expect.

3. Rerun Monte Carlo using the settings saved from the prior analysis.

4. Review the results.

Controlling measurement specifications

If you do not get the results you want and your design specifications are flexible, you can change a specification or delete a measurement and rerun Monte Carlo analysis.



Click here to remove the check mark and exclude the measurement from further analysis

Click on the dotted box and edit the measurement expression

Edit Cursor Min and Cursor Max values on the table; rerun Monte Carlo; observe new results.

Storing simulation data

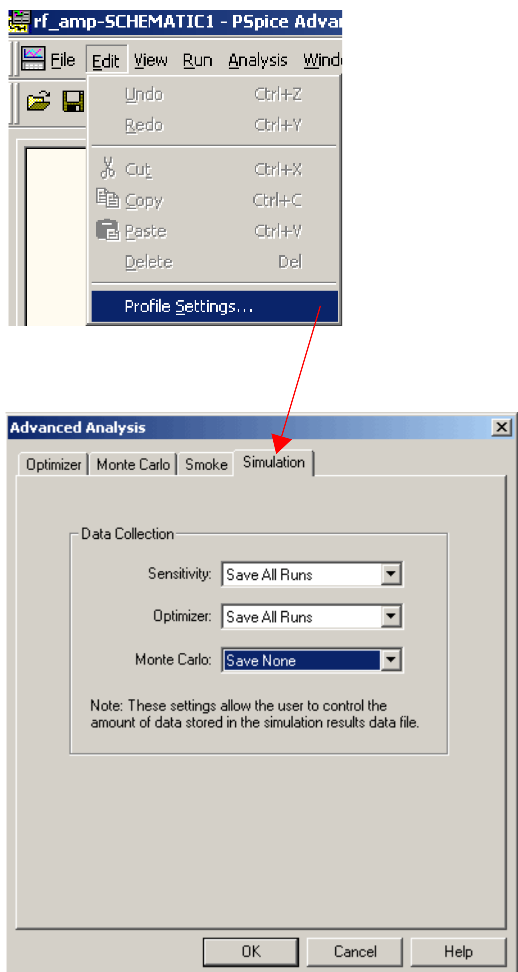
If you are planning an analysis of thousands of runs on a complex circuit, you can turn off the simulation data storage option to conserve disk space.

To turn off data storage:

1. From the Advance Analysis menu select *Edit / Profile Settings/ Simulation*.
2. From the Monte Carlo field, select **Save None**.

   The simulation data will be overwritten by each new run. Only the last run's data will be saved.

3. From the Advance Analysis menu select *Edit / Profile Settings/ Simulation*.



4. From the Monte Carlo field, select **Save None**.

The simulation data will be overwritten by each new run. Only the last run's data will be saved.

Changing components or parameters

When running other examples, if you do not get the results you want, go to the schematic editor and change circuit information.

1. Try a different component for the circuit

   Or:

   Change the tolerance of a parameter on an existing component.

2. Rerun the PSpice simulation and verify that the results are what you expect.
3. Rerun Monte Carlo using the settings saved from the prior analysis.
4. Review the results.

Printing results

➡ Click  .

Or

From the File menu, select Print.

To print information from the Raw Measurements table on the **Raw Meas** tab, copy and paste to an external program and print from that program. You can also print the Monte Carlo Log File, which contains more detail about measurement parameters.

Saving results

➡ Click .

Or:

From the **File** menu, select **Save**.

The final results will be saved in the Advanced Analysis profile (.aap).

7

Parametric Plotter

In this chapter

- Overview on page 231
- Launching Parametric Plotter on page 232
- Sweep Types on page 233
- Specifying measurements on page 238
- Running Parametric Plotter on page 240
- Viewing results on page 240
- Example on page 244

Overview

Note: Parametric Plotter is available only if you have PSpice[1] Advanced Analysis license.

The Parametric Plotter added to Advanced Analysis provides you with the functionality of sweeping multiple parameters. Once you have created and simulated a circuit, you can use the Parametric Plotter to perform this analysis.

The Parametric Plotter gives users the flexibility of sweeping multiple parameters. It also provides a nice and an efficient way to analyze sweep results. Using Parametric Plotter, you can sweep any number of design and model parameters (in any combinations) and view results in PPlot/Probe in tabular or plot form.

Using the Parametric Plotter, you can:

1. Depending on the license available, you will access either PSpice or PSpice Simulator.

- Sweep multiple parameters.
- Allow device/model parameters to be swept.
- Display sweep results in spreadsheet format.
- Plot measurement results in Probe UI.
- Post analysis measurement evaluation

Launching Parametric Plotter

From design entry tool[1]

➡ From the PSpice menu in design entry tool, select Advanced Analysis > Parametric Plot.

The Parametric Plotter window appears.

Stand Alone

1. From the Start menu, choose *Programs – Cadence Release 17.2-2016 – OrCAD Products – PSpice Advanced Analysis*.
2. Open the `.aap` file.
3. From the Analysis drop-down list, select Parametric Plotter.

   The Parametric Plotter window appears.

You can now use the Parametric Plotter to analyze your circuit. Using Parametric Plotter is a two steps process.

1. In the first step, you select the parameters to be swept and also specify the sweep type. See “Sweep Types” on page 233.
2. In the second step, you specify the measurements to evaluated at each sweep. See “Specifying measurements” on page 238.

After you have identified the sweep parameters and specified measurements, run the sweep analysis and view the results in the Results tab or the Plot Information tab of the Measurements window.

1. In this guide, design entry tool is used for both OrCAD Capture and Design Entry HDL. Any differences between the two tools is mentioned, if necessary.

Sweep Types

Advanced Analysis Parametric Plotter is used to perform the sweep analysis. When you run a sweep analysis, you evaluate the results of sweeping one or more parameter values, on the circuit output.

During the sweep analysis, the parameters values are varied as per the user specifications. There are four possible ways in which you can vary the parameter values. These are:

- Discrete Sweep
- Linear Sweep
- Logarithmic octave sweep
- Logarithmic decade sweep

Discrete Sweep

For discrete sweep, you need to specify the actual parameter values to be used during the simulation runs. The parameter values are used in the order they are specified.

Example

You can specify the values of variable parameters as 10, 100, 340, and so on.

Linear Sweep

For Linear sweep, specify the Start, End, and Step values. For each run of the parametric plotter, the parameter value is increased by the step value. In other words, the parameter values used during the simulation runs is calculated as `Start Value + Step Value`. This cycle continues till the parameter value is either greater than or equal to the `End Value`.

Example

If for a parameter you specify the start value as 1, End value as 2.5, and the step value as 0.5, the parameter values used by the Parametric Plotter are 1, 1.5, 2, and 2.5.

Logarithmic octave sweep

In the logarithmic octave sweep, the parameters are varied as a function of `ln(2)`.

For Logarithmic Octave sweep, you need to specify the Start Value, End Value, and number of points per Octave.

Number of points per Octave is number of points between the start value and two times start value. For example, if the start value is 10, number of points per Octave is 5, this implies that for sweep analysis, the Parametric Plotter will pick up 5 value between 10 and 20, with 20 being the fifth value.

During the analysis the parameter value in increased by a factor that is calculated using the following equation:

```
factor = exp[(ln(2)/N]
```

Where

N Number of points per octave

Example

Consider that the sweep type for a parameter is LogarithmicOct. The start value, end value and the number of points per Octave are specified as 10, 30, and 2, respectively.

The values used by the Parametric Plotter for LogarithmicOct sweep type will be `10`, `14.142`, `20`, `28.284`, and `40`.

In this example, the difference between start and end values is more than an octave, therefore, the actual number of values used by the Parametric Plotter is more than 2.

Logarithmic decade sweep

If the sweep type is LogarithmicDec, the parameter values are varied as a function of `ln(10)`. For Logarithmic decimal sweep, you need to specify the Start Value, End Value, and number of points per decade.

Number of points per decade is number of points between the start value and 10 times start value. For example, if the start value is 10, number of points per decade is 5, this implies that for sweep analysis, the Parametric Plotter will pick up 5 value between 10 and 100, with 100 being the fifth value.

During the analysis the parameter value in increased by a factor, which is calculated using the following equation:

```
factor = exp[(ln(10)/N]
```

Where

N Number of points per decade

Example

If you specify the start value as 10, end value as 100, and number of points per decade as 5, the parameter values used for sweep analysis will be 10, 15.8489, 25.1189, 39.8107, 63.0957, and 100.

Adding sweep parameters

In the Sweep Parameters window, add the parameters values that you want to vary during the sweep analysis.

1. In the Sweep Parameters window, click the *Click here to import a parameter from the design property map* row.

   The Parameter Selection dialog box appears with a list of components and the parameters for which you can sweep the parameter values.

   Only the component parameters that have been defined in the schematic, appear in the Parameter Selection dialog box.

2. For the parameter that you want to vary, specify the Sweep Type.

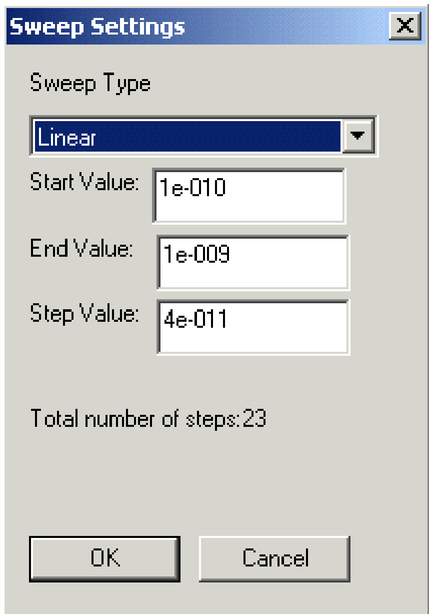
a. In the Parameter Selection dialog box, click the *Sweep Type* grid.

b. From the drop-down list, select the sweep type as `Discrete`, `Linear`, `LogarithmicDec`, or `LogarithmicOct`.

Note: Sweep type defines the method used by the Parametric Plotter to calculate variable parameter values. To know more about the sweep types, see "Sweep Types" on page 233.

3. To specify the sweep values for the selected parameter, click the Sweep Values grid.

The Sweep Settings dialog box appears.



4. In the Sweep Settings dialog box, the sweep type you selected in the previous step appears in the *Sweep Type* drop-down list box. Specify the parameter values that would be used for each parameter during sweep analysis.

To know more about the sweep types and sweep values to be specified, see Sweep Types on page 233

5. Click OK to save your specifications.

The selected parameters get added in the sweep parameter window. When you add the parameters, a Sweep Variable is automatically assigned to each of the parameters.

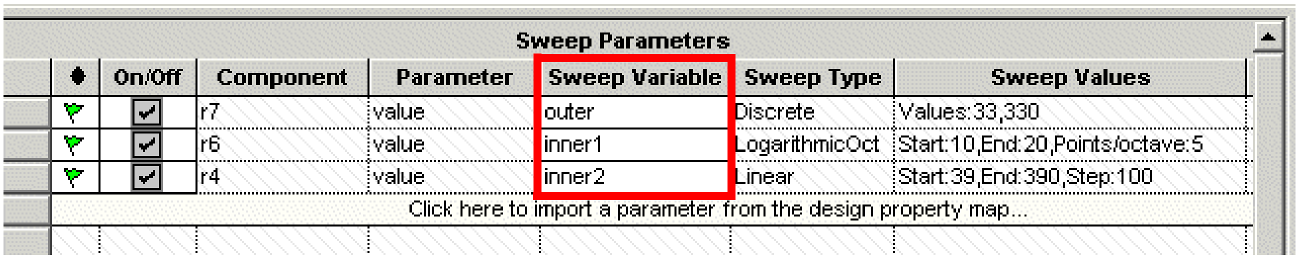


Figure 7-1 Setting sweep parameters

The value of the sweep variable is an indication of how parameters will be varied during sweep analysis. Sweep Variables values are assigned in the order in which sweep parameters are defined. If required, you can change these values. While modifying the values of Sweep Variable, ensure that each parameter has a unique value of sweep variable attached to it. Also the values should follow the sequence. For example, if you select three parameters to be varied during the sweep analysis, the sweep variables should have values as `outer`, `inner1`, and `inner2`. You cannot have random values such as `inner1`, `inner2`, and `inner4`.

For the sweep analysis, the values of parameters is varied in nested loops. For example, if you select two variables, the outer variable is fixed for the analysis, while the inner variable goes through all of its possible values. The outer variable is then incremented to its next value, and the inner variable again cycles through all of its possible values. This process is continued for all possible values of the outer variable.

The result for each run of the analyzer appears in the Results pane. By default, the results are displayed in the order described above.

Tip

Similar process is followed in case multiple (more than two) parameter values need to be varied.

For example, in Figure 7-1 on page 237, for constant values of r7 and r6, the value of r4 will be varied. The values of r7 and r6 will not change till r4 has been assigned all possible values within the range specified by the user. After r4 completes a cycle, the value of r6 will be increased, and r4 will again be varied for all possible values.

Specifying measurements

Parametric Plotter is used for evaluating the influence of changing parameter values on an expression and on a trace. A measurement can be defined as an expression that evaluates to a single value, where a trace is an expression that evaluates to a curve.

Adding measurement expressions

You can either add a measurement expression that was created in PSpice A/D or can even create a new measurement in PSpice Advanced Analysis.

Adding measurements created in PSpice

1. In the Measurements tab, click the *Click here to import a measurement created in PSpice* row.

   The Import Measurements dialog box appears. This dialog box lists only the measurements that you created in PSpice A/D.

2. Select the measurement that you want to be evaluated and click OK.

   Selected measurement gets added in the Measurements tab.

Tip

Only the measurements that are listed in the Measurements Results window of PSpice A/D are available in the Import Measurements dialog box.

Adding new measurements

1. In the Measurements tab, right-click and select Create New Measurements.

The New Measurement dialog box appears.

Note: For a two-pin device, the New Measurement dialog box in Advanced Analysis will show the current through the device and not through the pins. For example, for a diode D, it will only show I(D) and not I(D:1) or I(D:2).

2. From the Profile drop-down list, select the simulation profile for which you want to create the measurement.
3. From the Measurements drop-down list, select the Measurement that you want to evaluate.
4. From the Simulation Output Variables list specify the variable on which the measurement is to be performed and click OK

   The new measurement gets added to the Measurements tab.

Important

Using the New Measurements dialog box, you can only add the already defined measurements to the Parametric Plotter window. To define new measurements in PSpice use the *Trace > Measurements* command in PSpice A/D.

Adding a trace

Using the Parametric Plotter, you can evaluate the influence of changing parameter values on a trace. To be able to do this, you need to add a trace in the Measurements tab.

1. From the Analysis drop-down menu, select *Parametric Plotter > Create New Trace*.

   Alternatively, right-click on the Measurements tab and select *Create New Trace*.

   The New Trace Expression dialog box appears.
2. Create an expression to define the new trace and click OK.

   The trace expression gets added in the Measurement window, with type as Trace.

Running Parametric Plotter

After you have specified the measurements and the list of variable parameters, run the Parametric Plotter.

➡ From the Run drop-down menu choose Start Parametric Plotter.

Note: Alternatively, click the Run button on the toolbar or press `<CTRL>+<R>` keys.

For optimized performance of Parametric Plotter, maximum number of parametric sweeps supported in one session is 1000. If for your selection of parameters and measurements, the total number of sweeps required is greater than 1000, an error message is displayed in the Output Window, and analysis stops. As the simulation progresses, the Output Window also shows the profile selected and the number of sweep run being executed.

Important

The Number of parametric sweeps required, which is displayed in the Output window, should be interpretted as the number of sweeps required per profile. The total number of sweeps required is calculated separately for each profile.

Viewing results

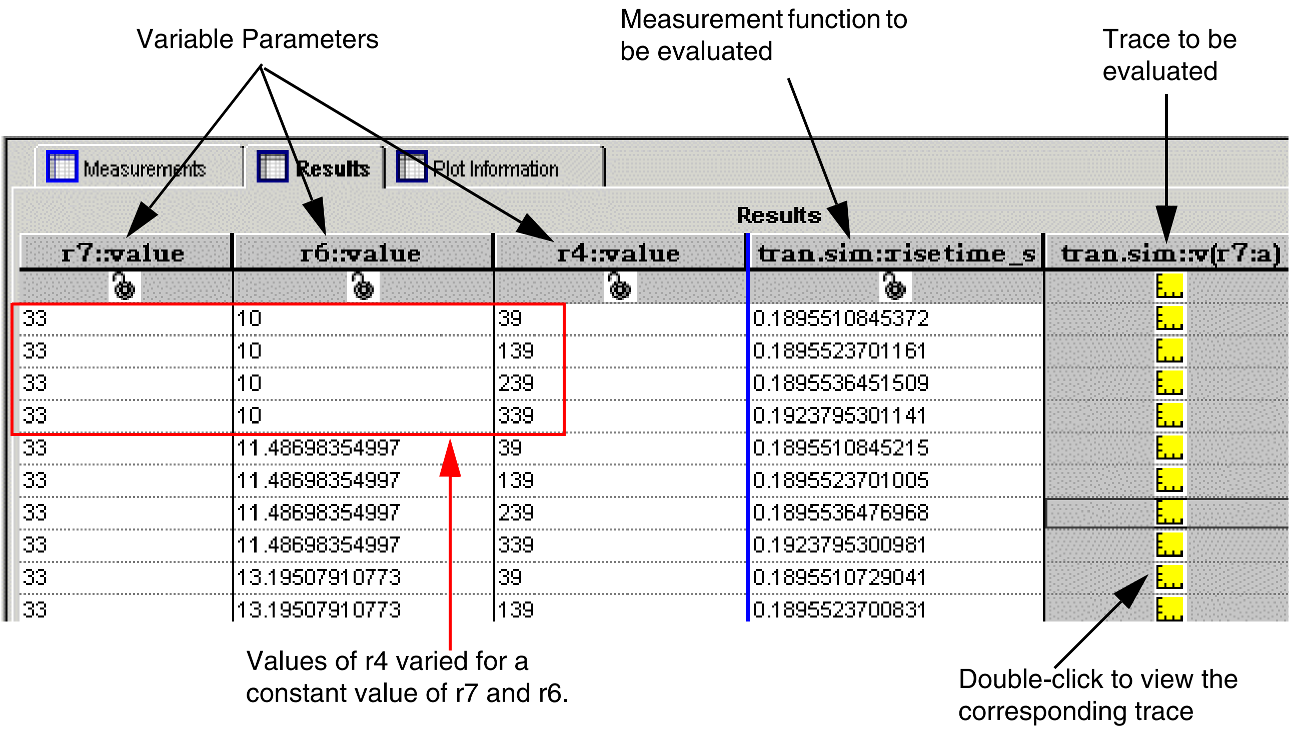
The results of the parametric sweep analysis are displayed in form of a spread sheet in the Results tab of the Measurement window. For the same results, you can define plot information using the Plot Information tab. The plot information is displayed in the PSpice Probe window.

Results tab

The results tab displays the simulation result for each run of the Parametric Plotter. Each run of the parametric plotter is indicated by a row in the Results tab. Therefore, if for the complete analysis Parametric plotter completes 100 runs, there will be 100 rows in the results tab.

The number of columns in the results tab is equal to the number of variable parameters and the number of measurements or the traces to be evaluated. There is one column each for a variable parameter and measurement expression to be evaluated.

In case of traces, instead of the measurement value, a trace is generated for each run of Parametric Plotter. As traces cannot displayed on the Results tab, therefore, instead of each trace a yellow colored bitmap is visible. To view the complete trace, double-click the yellow colored bitmap in the Results pane. The trace gets displayed in the PSpice Probe window.



Analyzing Results

You can set up the Parametric Plotter to display data in a number of ways.

Sorting values

You can sort the results of the sweep analysis according to the values in any column.

For example, if you want to view the result of keep r4 to a constant value of 39, sort the values in the third column and view the results.

To sort the values displayed in a column, double-click on the column name. Once the contents of the column are sorted, subsequent click on the column name with toggle the order of sorting.

For example, after the Results pane is populated, double-clicking the column name arranges the values in ascending order. Now if you again double-click on the column name, the column contents willl get arranged in descending order.

Locking Values

While analyzing the simulation results, you can lock the values displayed in one column. Once you have locked the values of a column, the order in which the values are displayed in that column do not change. You can then sort the values in other columns.

For example, you can sort the values of r7 and lock the column. If you now sort the values of r6, the values will be sorted for fixed value of r7.

To lock the values displayed in a column, click the lock icon at the top of the column.

Plot Information tab

The Plot Information tab can be used to specify a plot that you want to view in the Probe window. Using the Plot Information tab, you can view multiple traces in one window. This is useful when you want to view the result of varying a parameter on the output.

At any given point of time, you can add a maximum of four plots.

Adding plot

1. From the *Analysis* menu select *Parametric Plotter > Add New Plot*.

   The Plot Wizard appears.

   Note: Alternatively, right-click on the Plot Information tab and select Add Plot.

2. In the Select Profile page of the Plot Wizard, specify the simulation profile for which you want the profile to be created and click Next.
3. In the select X-Axis Variable page of the wizard, specify the variable parameter that you want to plot on the X-axis of the plot.

   From the variables drop-down list you can select any of the sweep parameter or the measurements that you specified in the Measurements tab.

   Besides the variable parameter and the measurements, the drop-down list has an extra entry, which is time or frequency.

   When you select a transient profile, you can select Time as the X-Axis variable and plot out results against time. When you select a AC profile, you can select Frequency as the X-Axis variable.
4. Click Next.
5. In the Select Y-Axis Variable page, select the variable to be plotted in the Y-axis and click Next.

   Depending on your selection in the previous page of the Plot wizard, either the measurement expressions or traces appears in the Variables drop-down list.

   When you select time or frequency as X-Axis Variable, all the traces added by you in the Measurements tab appear in the drop-down list. For all other selections of X-Axis Variables, the measurements added by you in the Measurements tab, are listed in the drop-down list.
6. In the Select Parameter page of the Plot Wizard, specify the parameter that will be varied for each trace to be plotted and click Next.
7. In cases where there are more than two variable parameters, you need to specify a constant value for the variable parameters that are not covered in step 3 or step 6.

   Right-click on the parameter value and choose Lock.
8. Click Finish.

The complete plot information gets added in the Plot Information tab.

Viewing the plot

1. Select the plot to be displayed in the PSpice probe window.
2. From the Analysis drop-down menu, choose Parametric Plotter > Display Plot.

   Alternatively, right-click on the selected row and choose Display Plot.

   The PSpice probe window appears with multiple traces.

Measurements Tab

Example

In this section, you will use Parametric Plotter to evaluate a simple test circuit for inductive switching. This circuit is created using a power mosfet from the `PWRMFET.OLB`.

The design example is available at
`..\tools\pspice\tutorial\capture\pspiceaa\snubber`
or
`..\tools\pspice\tutorial\concept\pspiceaa\snubber`
for Capture and Design Entry HDL respectively.

Add two voltage markers added to the circuit as shown in Figure 7-3 on page 246, are used to plot the input and the output voltages.

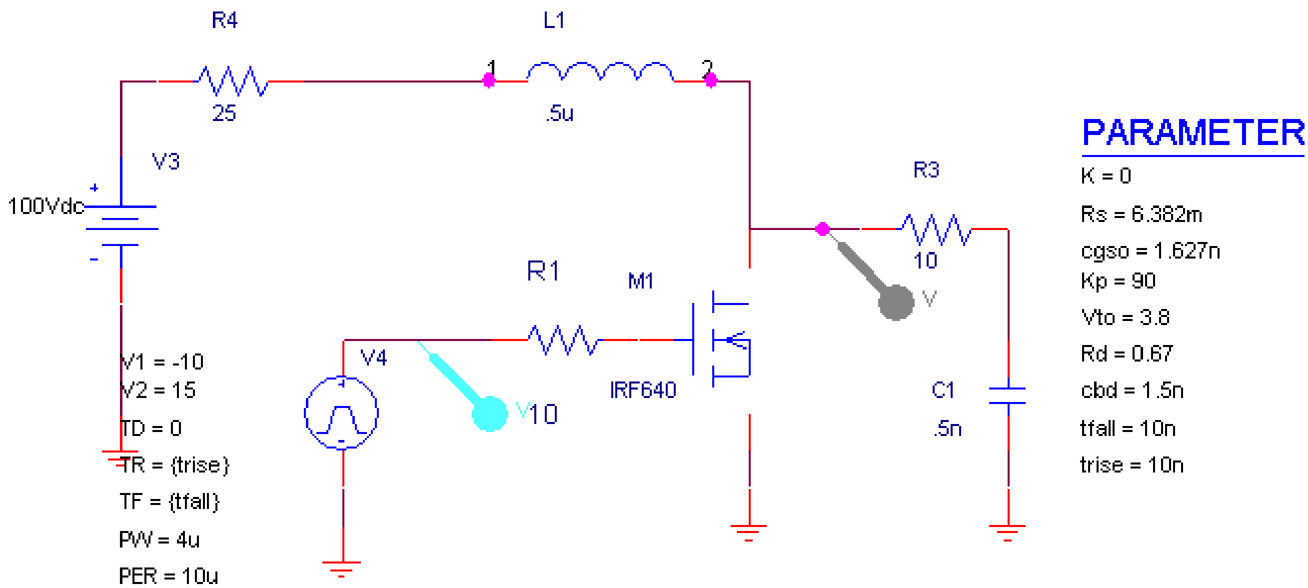


Figure 7-2 Inductive switching circuit

To view the input and the output voltages, you first need to simulate the circuit.

Simulating the circuit

➡ From the PSpice menu in the design entry tool, select Run.

The input and the output waveforms are displayed in Figure 7-3. The output waveform displays a spike at every falling edge of the input waveform.

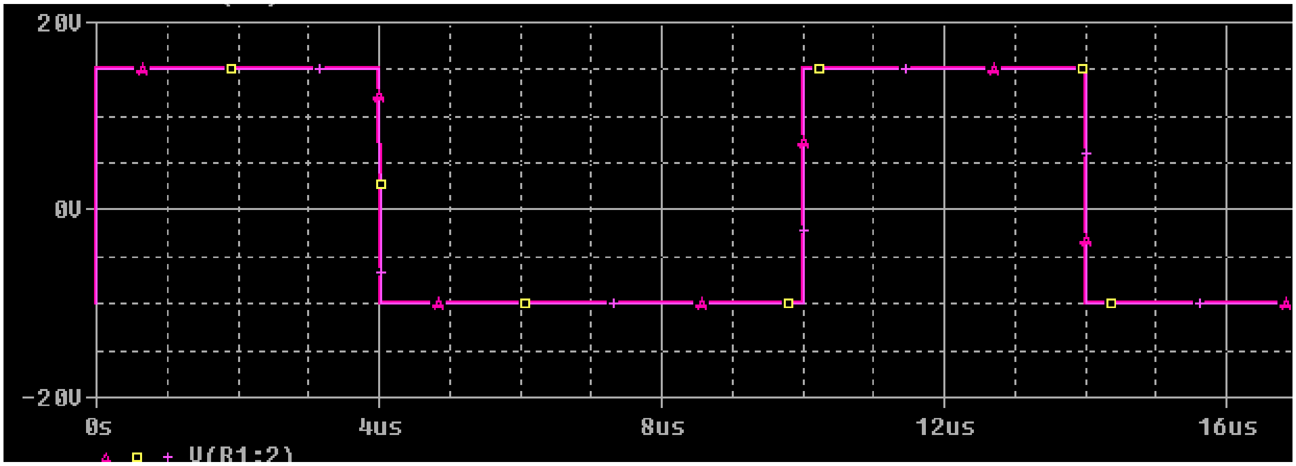


Figure 7-3 Input waveform

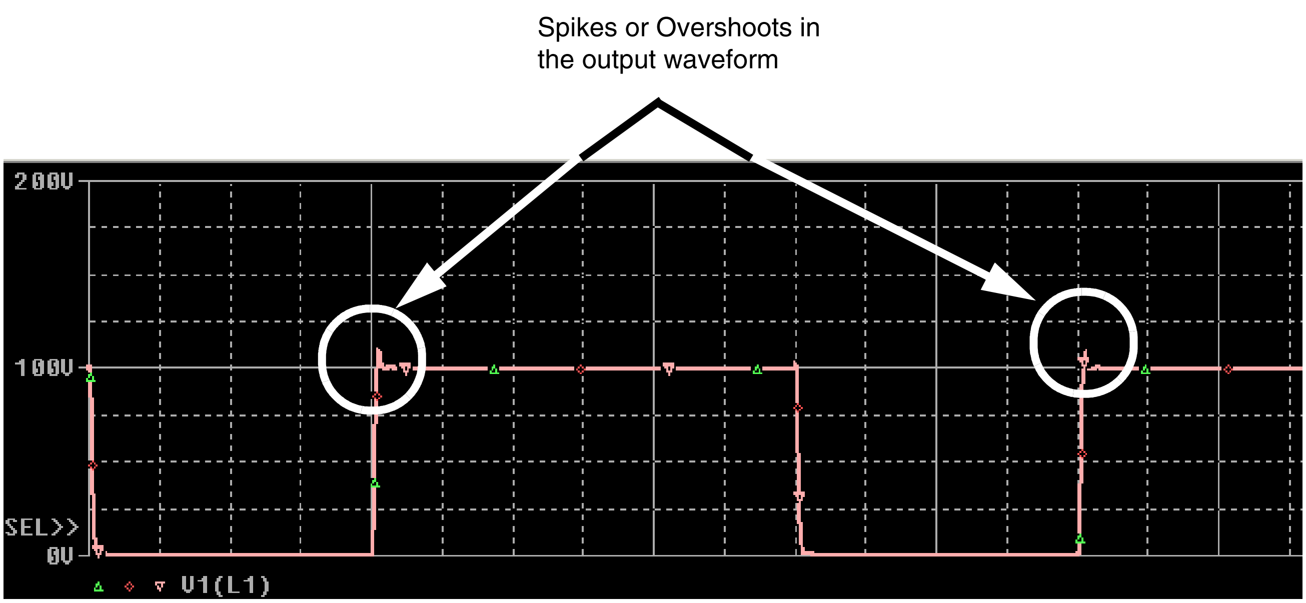


Figure 7-4 Output Waveform

Before users can use the output waveform, they need to adjust the circuit components so as to reduce the overshoot within the limit acceptable to the user. This can easily be done by increasing the values of resistor R3 and capacitor C1. But this results in increasing power dissipation across resistor R3.

Therefore, the design challange here is to balance the power dissipation and the voltage overshooot.

To find an acceptable soultion to the problem, we will vary the values of resistance R3, capacitor C1, and rise time of the input pulse and monitor the effect of varying the parameter values on the overshoot and the power dissipation across resistor R3.

To achieve this, use Parametric Plotter to run the sweep analysis. Before you can run the sweep analysis, complete the following sequence of steps.

1. Launch Parametric Plotter
2. Add sweep parameters
3. Add measurements
4. Run sweep analysis

Launch Parametric Plotter

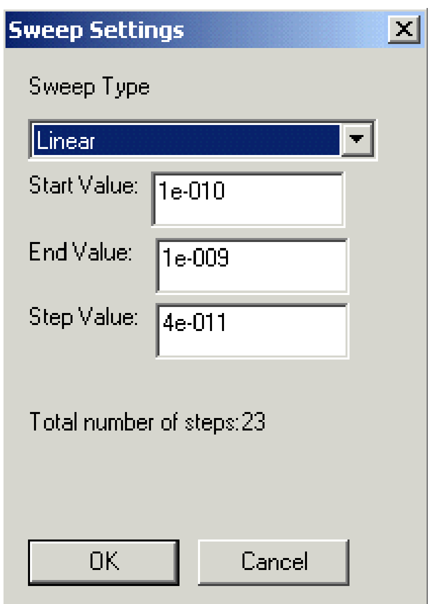
om the PSpice menu in the design entry tool, select Advanced Analysis > Parametric Plot.

Add sweep parameters

For the switching circuit design, we will vary `trise` linearly, specify discrete values for `R3`, and vary `C1` logarithmically.

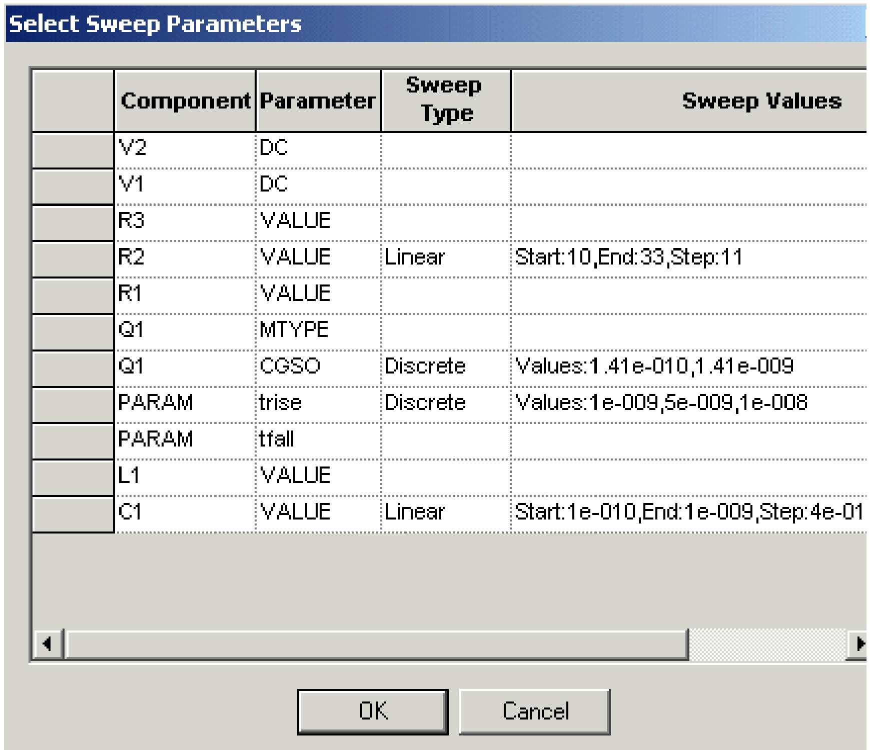
1. In the Sweep Parameters window, click the *Click here to import a parameter from the design property map* row.
2. In the Sweep Parameters window, select the parameter named `trise` and click inside the corresponding Sweep Type grid.
3. From the drop-down list, select Linear.
4. To specify the range within which the parameter values should be varied, click corresponding Sweep Values grid.
5. In the Sweep Settings dialog box, specify start value as `5n`, stop value as `12n` and the step value as `1n`.
   This implies that the rise time of the pulse will ve varied from 5 nano seconds to 12 nano seconds.

6. To add resistor R3 as the next sweep parameter, click the sweep type grid corresponding to the component named R3.
7. From the drop-down list, select Discrete.
8. To specify the values of resitor R3, click corresponding Sweep Values grid.
9. To specify a discrete value for resistor R3, click the New button and enter 5.
10. Similarly, specify other values as 15 and 20.
11. Click OK to close the Sweep Settings dialog box.
12. Finally, to add capacitor C1 as a sweep parameter and vary the capacitance value, click the sweep type grid corresponding to capacitor C1 and select Linear from the drop-down list.
13. Click the Sweep Values grid.
14. In the Sweep Settings dialog box, specify theStart Value as .1n, End value as 1n, and number of points as 10, and click OK.

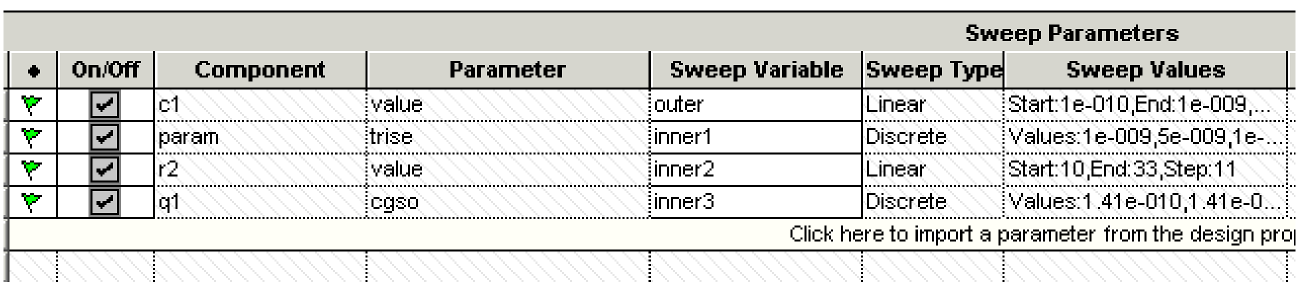


This implies that the sweep analysis will be performed for 10 values of capacitance between .1 nano farads to 1 nano farads.

15. In the Select Sweep Parameters dialog box, click OK to save your changes.



The changes are reflected in the Sweep Parameters window.



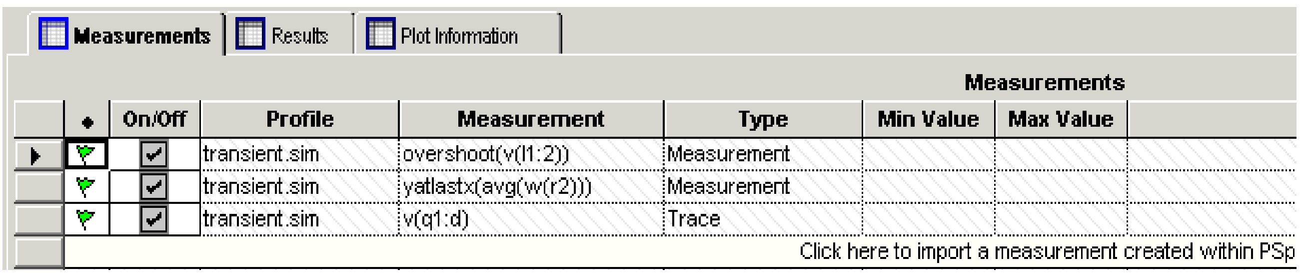
Besides the values entered by you in the Select Sweep Parameters dialog box, the Sweep Variable column also gets populated. Parametric Plotter assigns variables to the parameters depending on the order in which they are added. If required you can change this order.

Add measurements

To evalute the influence of varying parameter values on the overshoot and power disspipation across resistor R3, and to include a trace, add these three as the measurement expressions to be evaluated.

1. In the Measurements tab, select Click here to add a measurement created in PSpice row.
2. In the Import Measurement(s) dialog box, select `Overshoot(V(l1:2)),yatlastX(AVG(W(R2))), and v(q1:d)` from the `transient.sim` profile.
3. Click OK.

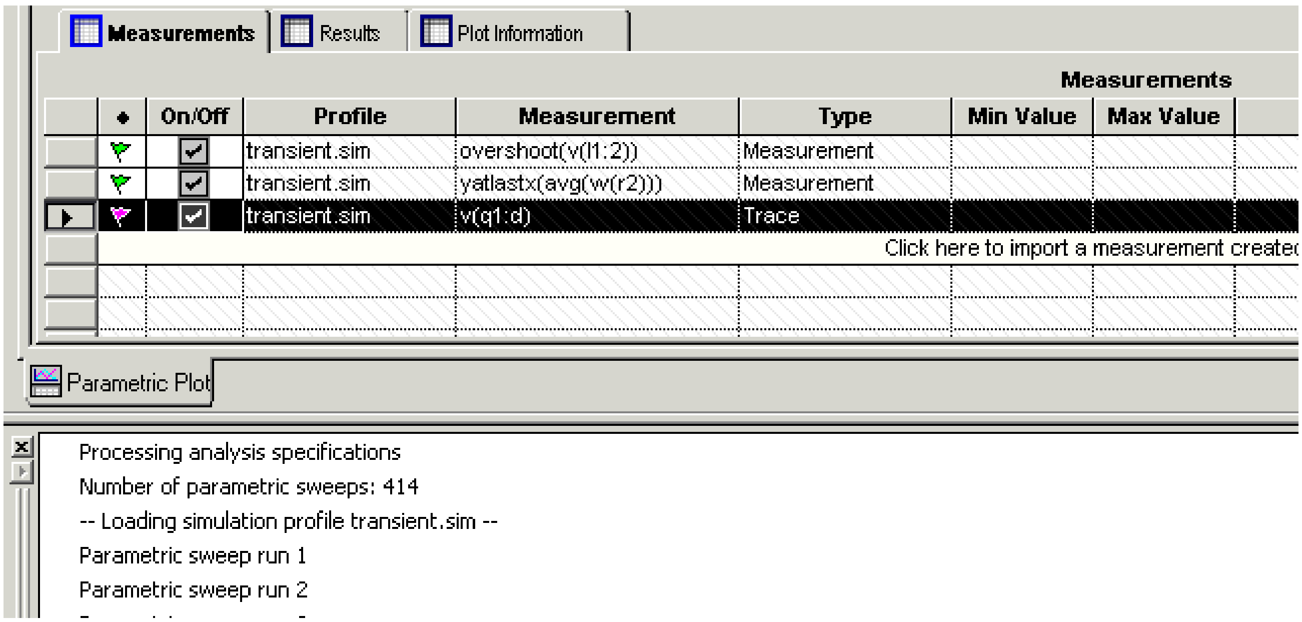
   The measurements get added to the Measurements tab.



Run sweep analysis

➡ To run the sweep analysis, click the Start button on the toolbar.

As Parametric Plotter starts running the Output window is populated with the total number of sweeps required to complete the analysis.



Once the analysis is over, the Min value and the Max Value columns are populated for each measurement specified in the Measurements tab. Besides this, results of each run of Parametric Plotter are displayed in the Results tab.

Measurements | **Results** | Plot Information

				Results		
c1::value	param::trise	r2::value	q1::cgso	transient.sim::	transient.sim::yat	transient.si
1e-010	1e-009	10	1.41e-010	119.9166302241	0.0321775477176	
1e-010	1e-009	10	1.41e-009	119.9821925127	0.0324279773713	
1e-010	1e-009	21	1.41e-010	119.914137655	0.0603237567281	
1e-010	1e-009	21	1.41e-009	119.9107487393	0.06554780049446	
1e-010	1e-009	32	1.41e-010	119.9309683295	0.08955525820374	
1e-010	1e-009	32	1.41e-009	119.8280060467	0.08928913342263	
1e-010	5e-009	10	1.41e-010	119.943226676	0.0308821876144	
1e-010	5e-009	10	1.41e-009	119.9433181952	0.03261846159308	
1e-010	5e-009	21	1.41e-010	119.9281708798	0.05919543114364	
1e-010	5e-009	21	1.41e-009	119.8828182479	0.06712229020074	
1e-010	5e-009	32	1.41e-010	119.9165215125	0.08948388197998	
1e-010	5e-009	32	1.41e-009	119.9152673946	0.08914627690525	
1e-010	1e-008	10	1.41e-010	119.9326834952	0.03060373697535	

Figure 7-5 Results tab in Parametric Plotter

In the Results tab, you can sort and lock the results displayed in various columns. For example, consider that in case of the inductive switching circuit, your primary goal is to restrict the power loss, which is measured by `yatlastx(avg(w(r2))`, to less than `0.006`, and then minimize the overshoot.

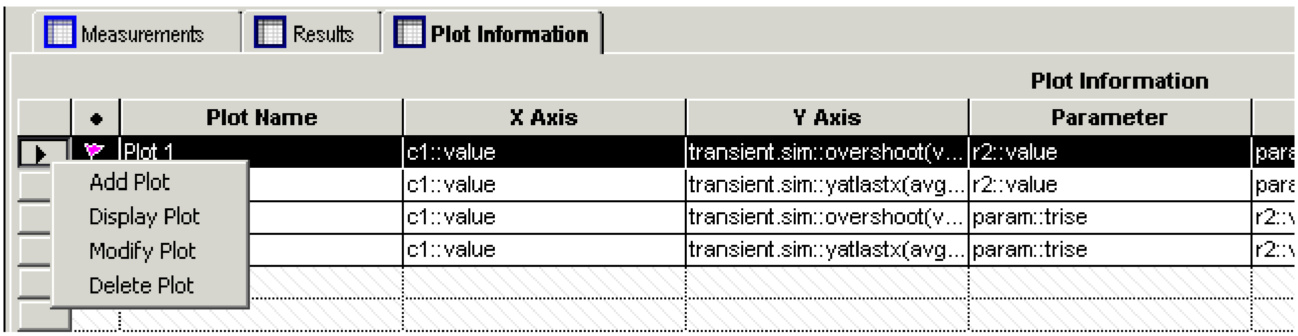
To achieve your goal, first sort the values displayed in the sixth column of Figure 7-5 on page 252. To sort the values, double-click on the column heading. The values get assorted in the ascending order. Next you lock the sorted values. To lock the values, click the lock icon on the top of the column.

After sorting the power loss values, sort the values displayed in the fifth column of Figure 7-5 on page 252. As a result of this sorting the values in the last column do not get disturbed. As a result, for all values of `atlastx(avg(w(r2))`, to less than `0.006`, the overshoot values get sorted. Thus you can view the combination(s) of the parameter values for which both the outputs are in the desired range.

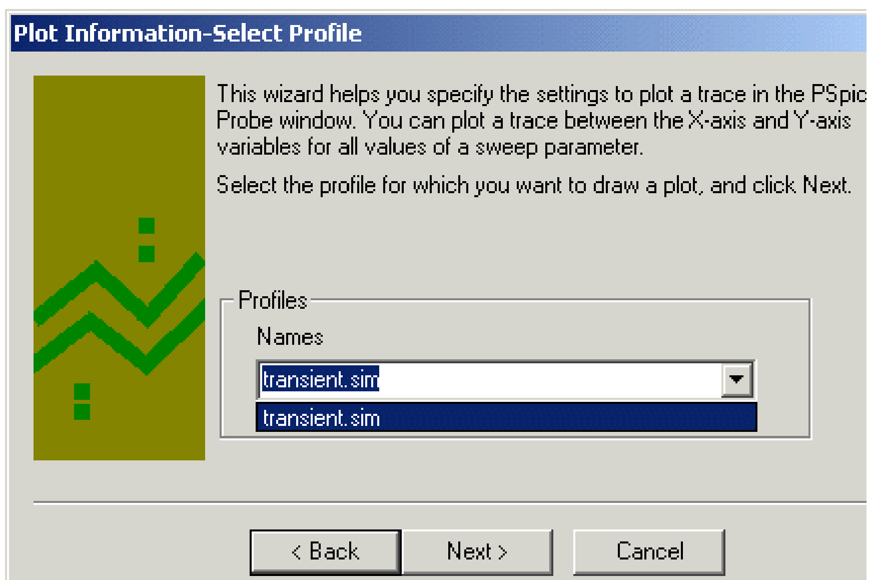
Add Plot

You can plot a trace between the X-axis and Y-axis variables for all values of a sweep parameter by using the Plot wizard. This wizard helps you specify the settings to plot a trace in the PSpice Probe window.

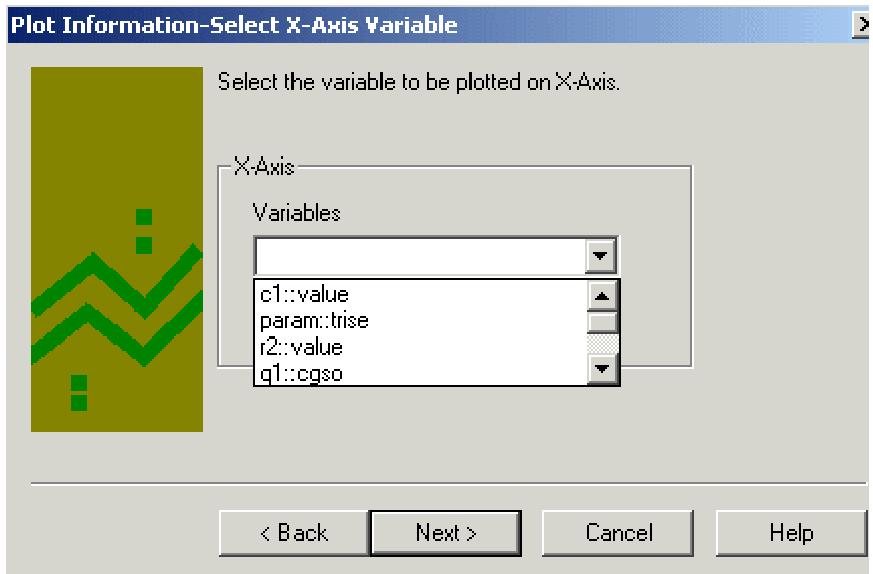
1. In the Plot Information tab, right-click in the plot information row and then click Add Plot. This displays the Plot wizard.



2. Select the `transient.sim` profile, and click Next.

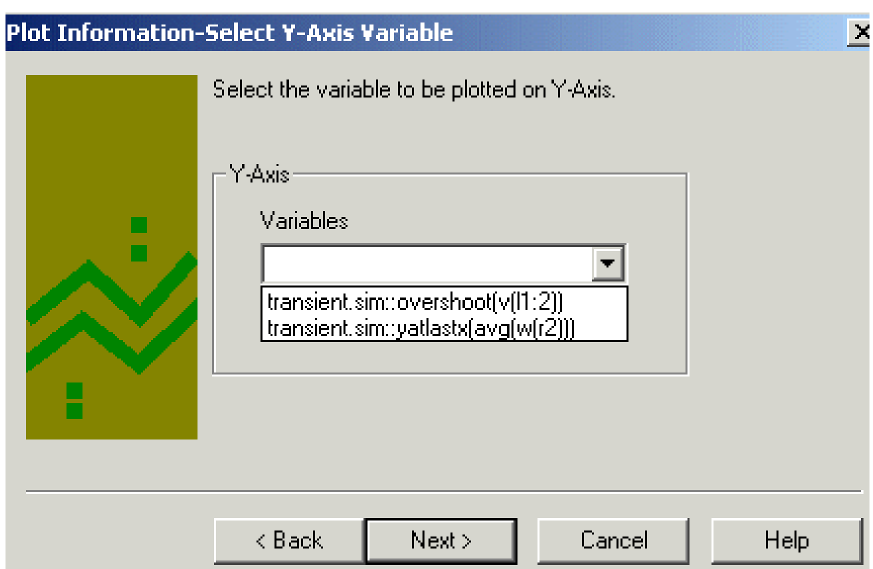


3. Select `r2::value` as the variable to be plotted on the X-axis, and click Next.

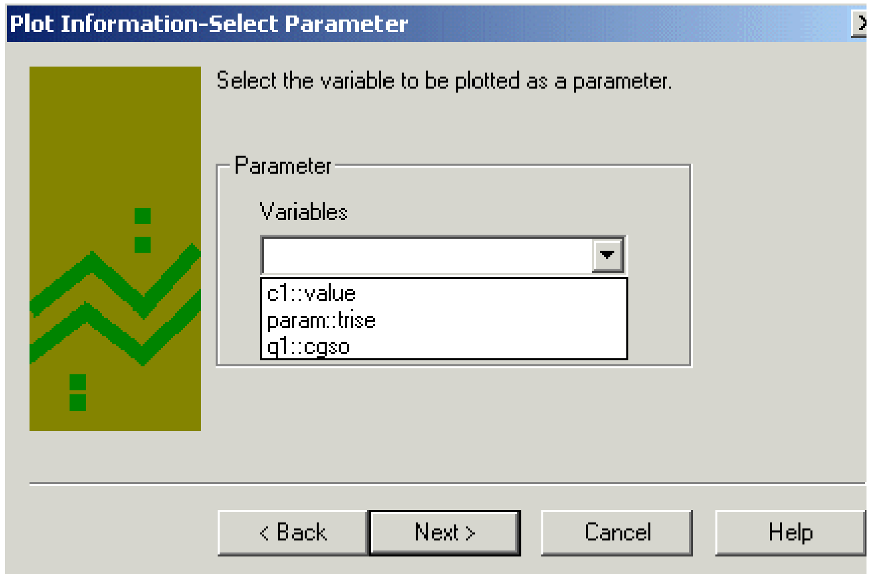


Note: If you select a Parameter or Measurement variable to be plotted on the X-axis, you will only be allowed to select a "Measurement" variable to be plotted on the Y-axis. If you select Time/Frequency variable, the wizard will only display a list of available traces that can be plotted on the Y-axis.

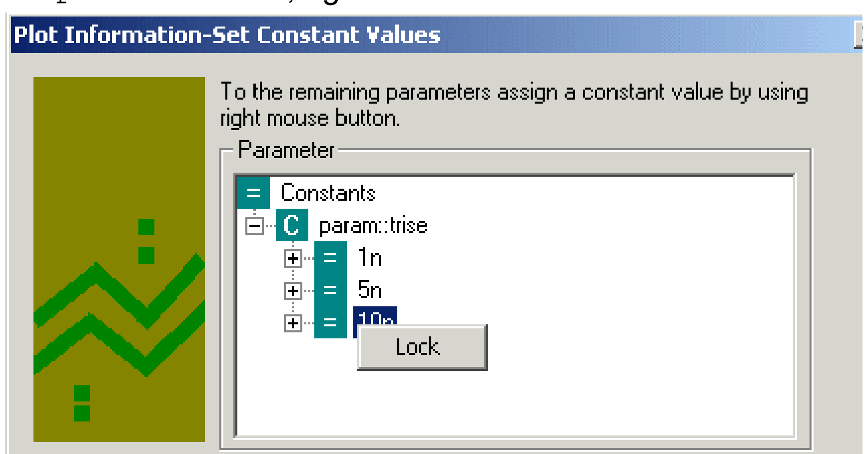
4. Select `transient.sim::overshoot(v[l1:2])` as the variable to be plotted on the Y-axis, and click Next.



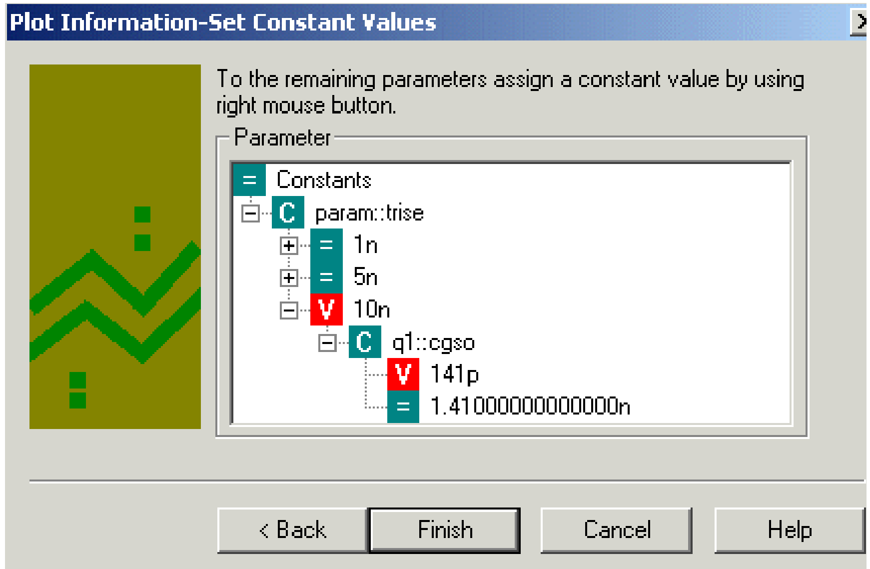
5. Select `c1::value` as the parameter to be varied, such that for each possible value of this parameter, you have a unique x-y trace, and click Next.



6. The remaining sweep parameters and their possible values are listed. For each parameter, select a constant value to be used for drawing the trace(s). To assign a constant valueto `param::trise`, right-click on `10n` and lock it.

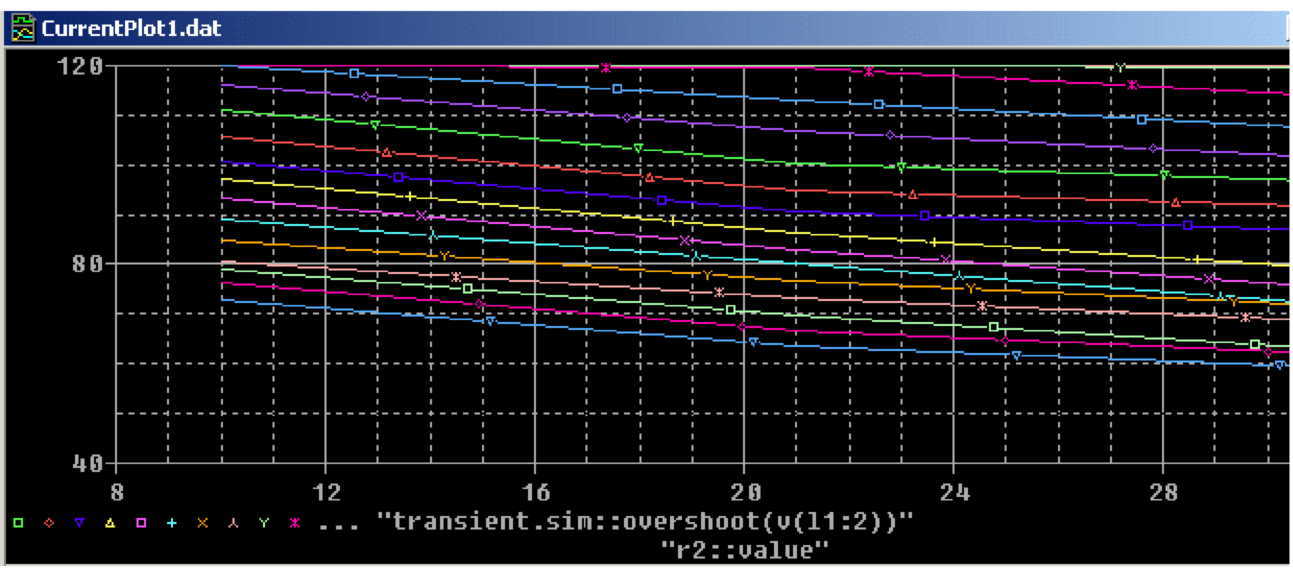


7. Similarly assign a constant value to q1::cgso. Click Finish.



8. In the Plot Information tab, right-click in the plot information row and then click Display Plot. This displays the trace that you plotted.

.



8

Measurement Expressions

In this chapter

- Measurements overview on page 257
- Measurement strategy on page 258
- Procedure for creating measurement expressions on page 258
- Example on page 260
- For power users on page 269

Measurements overview

Measurement expressions evaluate the characteristics of a waveform. A measurement expression is made by choosing the waveform and the waveform calculation you want to evaluate.

The waveform calculation is defined by a measurement definition such as rise time, bandpass bandwidth, minimum value, and maximum value.

For example, if you want to measure the risetime of your circuit output voltage, use the following expression:

Risetime(v(out))

For a list of the PSpice[1] measurement definitions, see Measurement definitions included in PSpice on page 265.

You can also create your own custom measurement definitions. See Creating custom measurement definitions in the Power user section of this chapter.

1. Depending on the license available, you will access either PSpice or PSpice Simulator.

Measurement strategy

- Start with a circuit created in a design entry tool[1] and a working PSpice simulation.
- Decide what you want to measure.
- Select the measurement definition that matches the waveform characteristics you want to measure.
- Insert the output variable (whose waveform you want to measure) into the measurement definition, to form a measurement expression.
- Test the measurement expression.

Procedure for creating measurement expressions

Setup

Before you create a measurement expression to use in Advanced Analysis:

1. Design a circuit in a design entry tool.
2. Set up a PSpice simulation.

   The Advanced Analysis tools use these simulations:

   - Time Domain (transient)
   - DC Sweep
   - AC Sweep/Noise
3. Run the circuit in PSpice.

   Make sure the circuit is valid and you have the results you expect.

1. In this guide, design entry tool is used for both OrCAD Capture and Design Entry HDL. Any differences between the two tools is mentioned, if necessary.

Composing a measurement expression

These steps show you how to create a measurement expression in PSpice. Measurement expressions created in PSpice can be imported into Sensitivity, Optimizer, and Monte Carlo.

You can also create measurements while in Sensitivity, Optimizer, and Monte Carlo, but those measurements cannot be imported into PSpice for testing.

First select a measurement definition, and then select output variables to measure. The two combined become a measurement expression.

Work in the Simulation Results view in PSpice. In the side toolbar, click on  .

1. From the **Trace** menu in PSpice, select **Measurements**.

   The **Measurements** dialog box appears.

2. Select the measurement definition you want to evaluate.

3. Click **Eval** (evaluate).

   The **Arguments for Measurement Evaluation** dialog box appears.

4. Click the **Name of trace to search** button.

   The **Traces for Measurement Arguments** dialog box appears.

Note: You will only be using the Simulation Output Variables list on the left side. Ignore the Functions or Macros list.

5. Uncheck the output types you don’t need (if you want to simplify the list).

6. Click on the output variable you want to evaluate.

   The output variable appears in the **Trace Expression** field.

7. Click **OK**.

   The **Arguments for Measurement Evaluation** dialog box reappears with the output variable you chose in the **Name of trace to search** field.

8. Click **OK**.

   Your new measurement expression is evaluated and displayed in the PSpice window.

9. Click **OK** in the **Display Measurement Evaluation** pop-up box to continue working in PSpice.

   Your new measurement expression is saved, but it no longer displays in the window. The only way to get another graphical display is to redo these steps.

   You can see a numerical evaluation by following the next steps.

Viewing the results of measurement evaluations

1. From the **View** menu in PSpice, select **Measurement Results**.

   The **Measurement Results** table displays below the plot window.

2. Click the box in the **Evaluate** column.

   The PSpice calculation for your measurement expression appears in the **Value** column.

Example

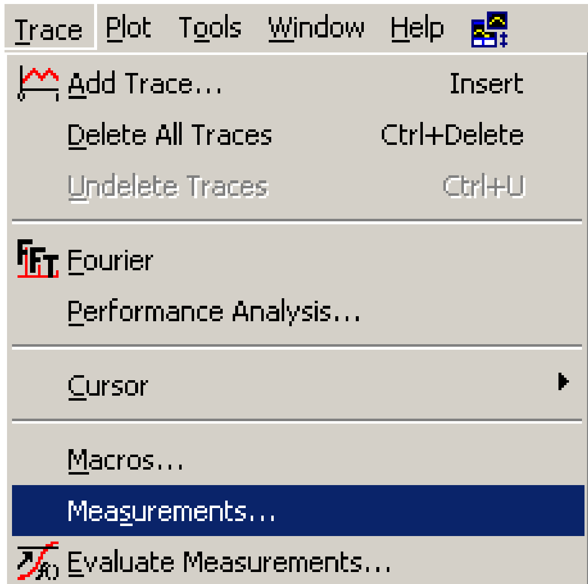
First you select a measurement definition, and then you select an output variable to measure. The two combined become a measurement expression.

Note: For the current design example, work in the Simulation Results view in PSpice.

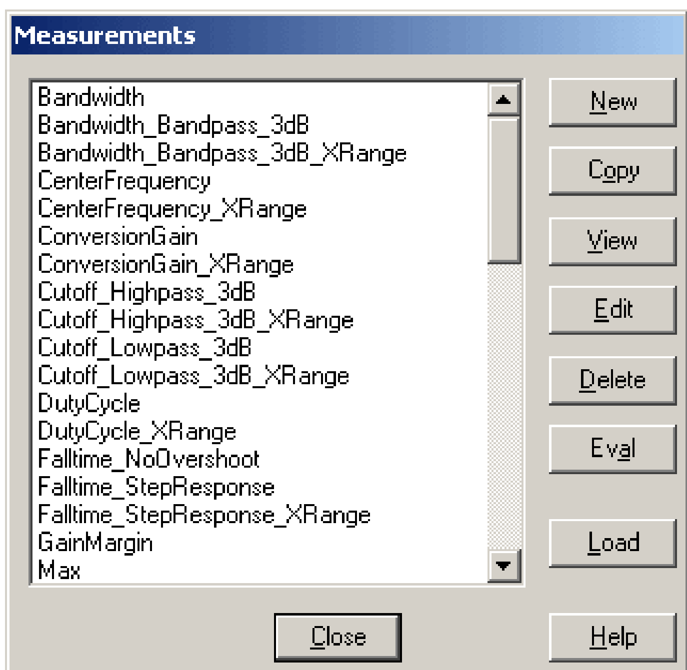
1. In the side toolbar, click on  .

2. From the **Trace** menu in PSpice, select **Measurements**.

The **Measurements** dialog box appears.



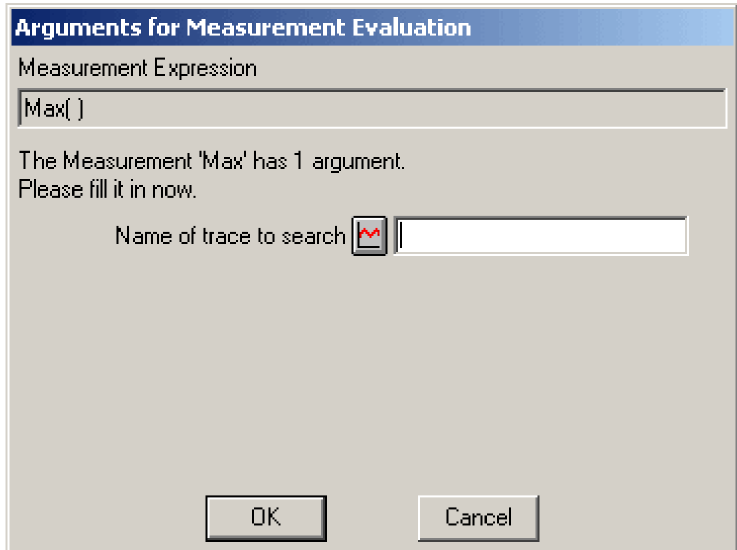
3. Select the measurement definition you want to evaluate.
4. Click **Eval** (evaluate).



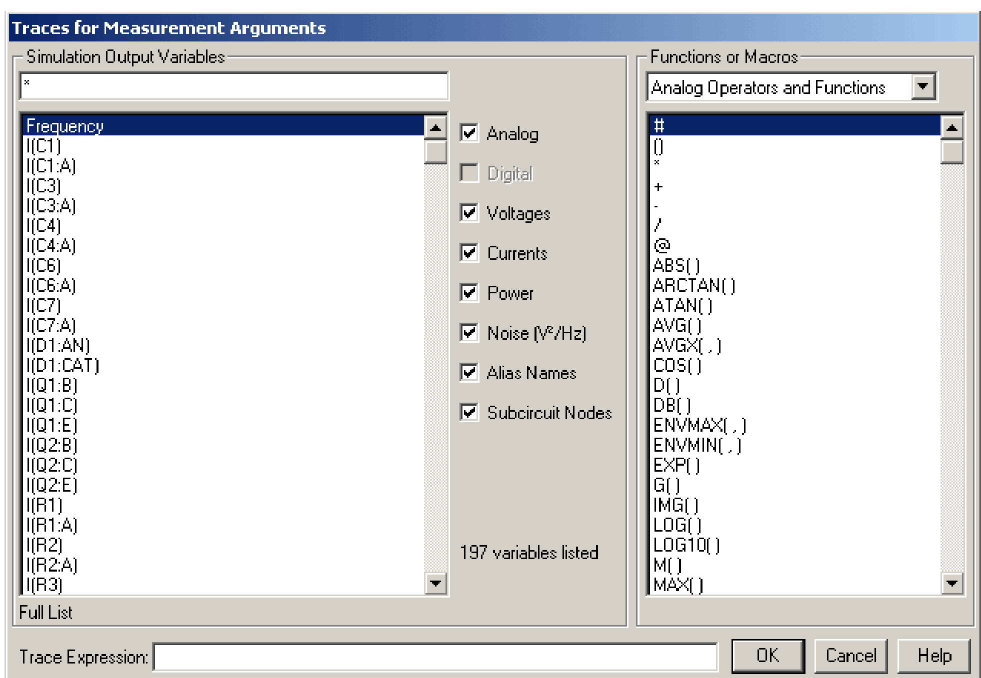
The **Arguments for Measurement Evaluation** dialog box appears.

5. Click the **Name of trace to search** button.

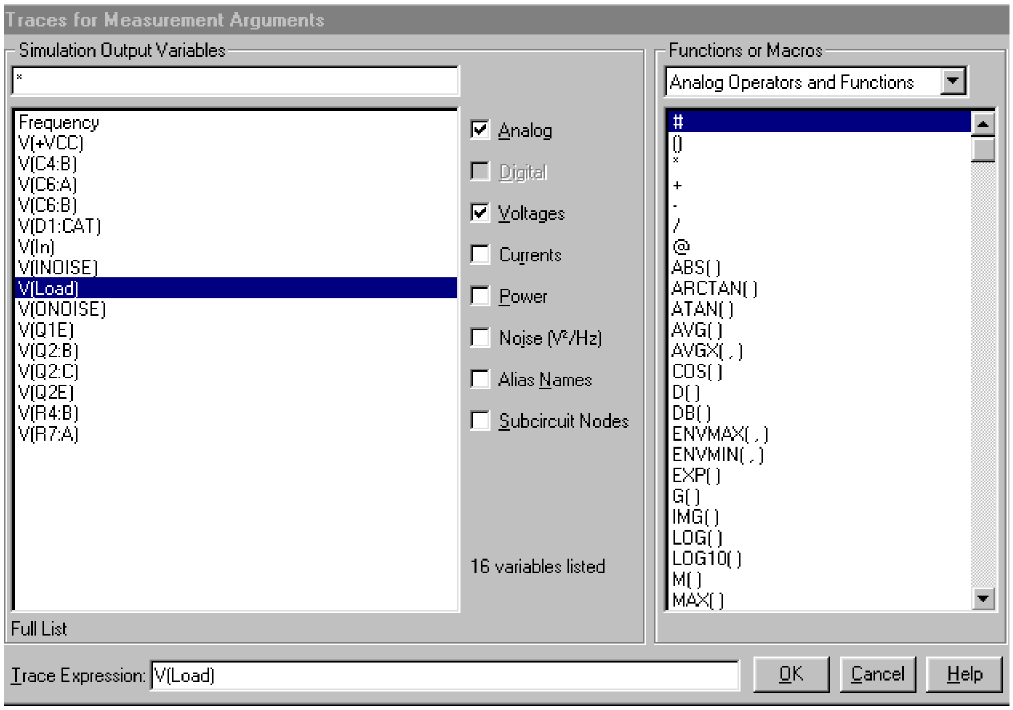
   The **Traces for Measurement Arguments** dialog box appears.



Note: You will only be using the Simulation Output Variables list on the left side. Ignore the Functions or Macros list.



6. Uncheck the output types you don't need (if you want to simplify the list).

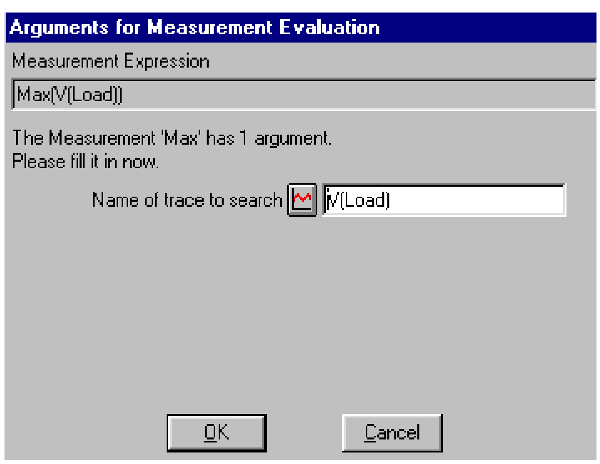


7. Click on the output variable you want to evaluate.

   The output variable appears in the **Trace Expression** field.

8. Click **OK**.

   The **Arguments for Measurement Evaluation** dialog box reappears with the output variable you chose in the **Name of trace to search** field.

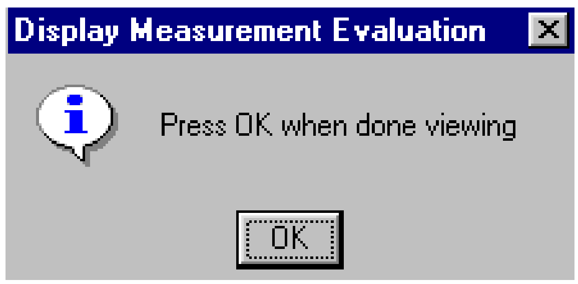


9. Click **OK**.

   Your new measurement expression is evaluated and displayed in the PSpice window.

10. Click **OK** in the **Display Measurement Evaluation** pop-up box to continue working in PSpice.

    Your new measurement expression is saved, but does not display in the window. The only way to get another graphical display is to redo these steps. You can see a numerical evaluation by following the next steps.

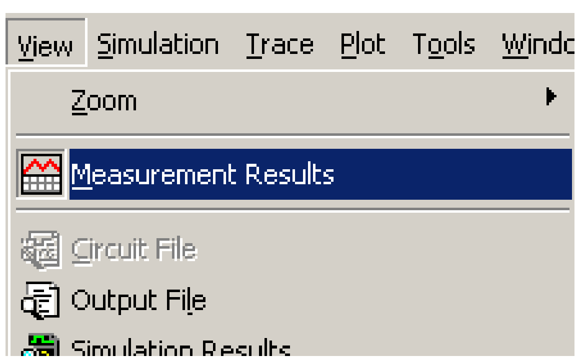


11. Click **Close**.

Viewing the results of measurement evaluations.

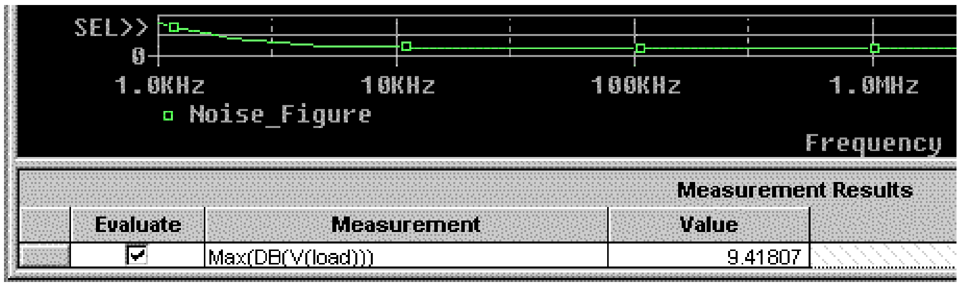
1. From the **View** menu, select **Measurement Results**.

   The **Measurement Results** table displays below the plot window.



2. Click the box in the **Evaluate** column.

A checkmark appears in the **Evaluate** column checkbox and the PSpice calculation for your measurement expression appears in the **Value** column.



Measurement definitions included in PSpice

Definition	Finds the. . .
Bandwidth	Bandwidth of a waveform (you choose dB level)
Bandwidth_Bandpass_3dB	Bandwidth (3dB level) of a waveform
Bandwidth_Bandpass_3dB_XRange	Bandwidth (3dB level) of a waveform over a specified X-range
CenterFrequency	Center frequency (dB level) of a waveform
CenterFrequency_XRange	Center frequency (dB level) of a waveform over a specified X-range

Definition	Finds the. . .
ConversionGain	Ratio of the maximum value of the first waveform to the maximum value of the second waveform
ConversionGain_XRange	Ratio of the maximum value of the first waveform to the maximum value of the second waveform over a specified X-range
Cutoff_Highpass_3dB	High pass bandwidth (for the given dB level)
Cutoff_Highpass_3dB_XRange	High pass bandwidth (for the given dB level)
Cutoff_Lowpass_3dB	Low pass bandwidth (for the given dB level)
Cutoff_Lowpass_3dB_XRange	Low pass bandwidth (for the given dB level) over a specified range
DutyCycle	Duty cycle of the first pulse/period
DutyCycle_XRange	Duty cycle of the first pulse/period over a range
Falltime_NoOvershoot	Falltime with no overshoot.
Falltime_StepResponse	Falltime of a negative-going step response curve
Falltime_StepResponse_XRange	Falltime of a negative-going step response curve over a specified range
GainMargin	Gain (dB level) at the first 180-degree out-of-phase mark
Max	Maximum value of the waveform
Max_XRange	Maximum value of the waveform within the specified range of X
Min	Minimum value of the waveform
Min_XRange	Minimum value of the waveform within the specified range of X
NthPeak	Value of a waveform at its nth peak
Overshoot	Overshoot of a step response curve

Definition	Finds the. . .
Overshoot_XRange	Overshoot of a step response curve over a specified range
Peak	Value of a waveform at its nth peak
Period	Period of a time domain signal
Period_XRange	Period of a time domain signal over a specified range
PhaseMargin	Phase margin
PowerDissipation_mW	Total power dissipation in milli-watts during the final period of time (can be used to calculate total power dissipation, if the first waveform is the integral of V(load)
Pulsewidth	Width of the first pulse
Pulsewidth_XRange	Width of the first pulse at a specified range
Q_Bandpass	Calculates Q (center frequency / bandwidth) of a bandpass response at the specified dB point
Q_Bandpass_XRange	Calculates Q (center frequency / bandwidth) of a bandpass response at the specified dB point and the specified range
Risetime_NoOvershoot	Risetime of a step response curve with no overshoot
Risetime_StepResponse	Risetime of a step response curve
Risetime_StepResponse_XRange	Risetime of a step response curve at a specified range
SettlingTime	Time from <begin_x> to the time it takes a step response to settle within a specified band
SettlingTime_XRange	Time from <begin_x> to the time it takes a step response to settle within a specified band and within a specified range
SlewRate_Fall	Slew rate of a negative-going step response curve

Definition	Finds the. . .
SlewRate_Fall_XRange	Slew rate of a negative-going step response curve over an X-range
SlewRate_Rise	Slew rate of a positive-going step response curve
SlewRate_Rise_XRange	Slew rate of a positive-going step response curve over an X-range
Swing_XRange	Difference between the maximum and minimum values of the waveform within the specified range
XatNthY	Value of X corresponding to the nth occurrence of the given Y_value, for the specified waveform
XatNthY_NegativeSlope	Value of X corresponding to the nth negative slope crossing of the given Y_value, for the specified waveform
XatNthY_PercentYRange	Value of X corresponding to the nth occurrence of the waveform crossing the given percentage of its full Y-axis range; specifically, nth occurrence of Y=Ymin+(Ymax-Ymin)*Y_pct/100
XatNthY_Positive Slope	Value of X corresponding to the nth positive slope crossing of the given Y_value, for the specified waveform
YatFirstX	Value of the waveform at the beginning of the X_value range
YatLastX	Value of the waveform at the end of the X_value range
YatX	Value of the waveform at the given X_value
YatX_PercentXRange	Value of the waveform at the given percentage of the X-axis range
ZeroCross	X-value where the Y-value first crosses zero

Definition	Finds the. . .
ZeroCross_XRange	X-value where the Y-value first crosses zero at the specified range

For power users

Creating custom measurement definitions

Measurement definitions establish rules to locate interesting points and compute values for a waveform. In order to do this, a measurement definition needs:

- A measurement definition name

  So it will come when it's called.

- A marked point expression

These are the calculations that compute the final point on the waveform.

- One or more search commands

  These commands specify how to search for the interesting points.

Strategy

1. Decide what you want to measure.
2. Examine the waveforms you have and choose which points on the waveform are needed to calculate the measured value.
3. Compose the search commands to find and mark the desired points.
4. Use the marked points in the Marked Point Expressions to calculate the final value for the waveform.
5. Test the search commands and measurements.

Note: An easy way to create a new definition:

From the PSpice **Trace** menu, select **Measurements** to open the **Measurements** dialog box, then:

- ❑ Select the definition most similar to your needs
- ❑ Click **Copy** and follow the prompts to rename and edit.

Writing a new measurement definition

1. From the PSpice **Trace** menu, choose **Measurements**.

   The **Measurements** dialog box appears.

2. Click **New**.

   The **New Measurement** dialog box appears.

3. Type a name for the new measurement in the **New Measurement name** field.

   Make sure **local file** is selected.

This stores the new measurement in a .prb file local to the design.

4. Click **OK**.

   The **Edit New Measurement** dialog box appears.

5. Type in the marked expression.

6. Type in any comments you want.

7. Type in the search function.

Note: For syntax information, see Measurement definition syntax on page 273.

Your new measurement definition is now listed in the **Measurements** dialog box.

Using the new measurement definition

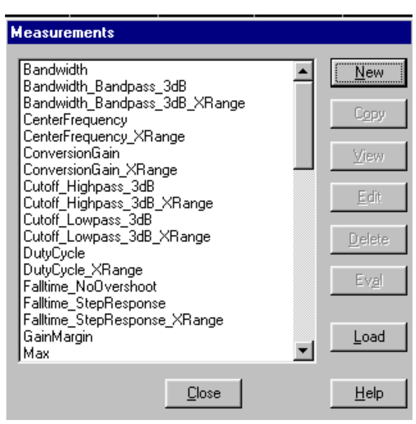
Your new measurement definition is now listed in the **Measurements** dialog box.

Note: For steps on using a definition in a measurement expression to evaluate a trace, see Composing a measurement expression on page 259.

Definition example

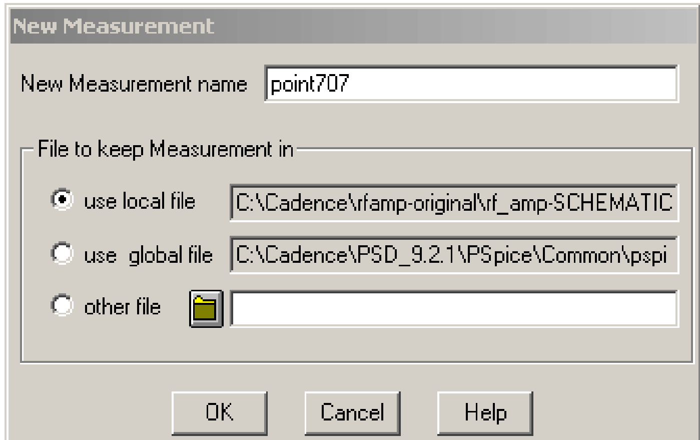
1. From the PSpice**Trace** menu, choose **Measurements**.

   The **Measurements** dialog box appears.



2. Click **New**.

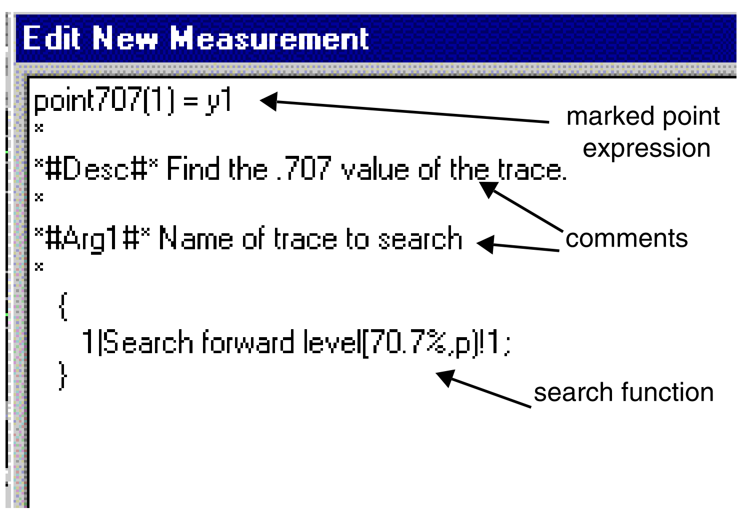
   The **New Measurement** dialog box appears.



3. Type in a name in the **New Measurement name** field.
4. Make sure **use local file** is selected.

   This stores the new measurement in a .prb file local to the design.
5. Click **OK**.

   The **Edit New Measurement** dialog box appears.



6. Type in the marked expression:

   point707(1) = y1

7. Type in the search function.

```
{
    1|Search forward level(70.7%, p) !1;
  }
```

Note: The search function is enclosed within curly braces.

Always place a semi-colon at the end of the last search function.

8. Type in any explanatory comments you want:

```
*
*#Desc#* Find the .707 value of the trace.
*
*#Arg1#* Name of trace to search
*
```

Note: For syntax information, see Measurement definition syntax on page 273.

Using the new measurement definition

Your new measurement definition is now listed in the **Measurements** dialog box.

For an example of using a definition in a measurement expression to evaluate a trace, see Example on page 260.

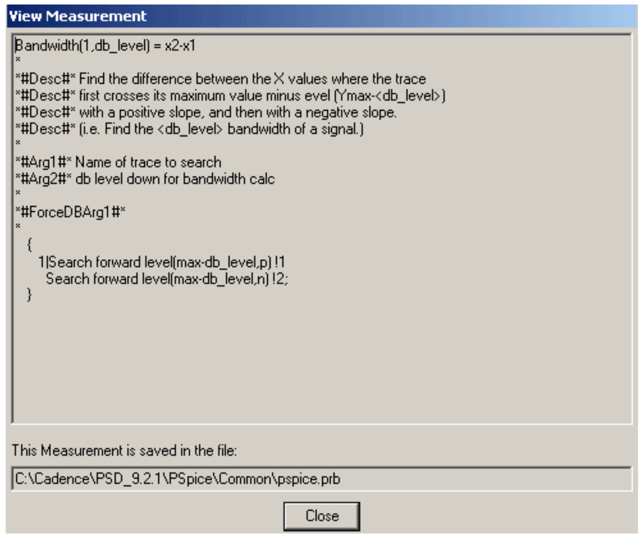
Measurement definition syntax

Check out the existing measurement definitions in PSpice for syntax examples.

1. From the **Trace** menu, choose **Measurements**.

   The **Measurement** dialog box appears.

2. Highlight any example, and select **View** to examine the syntax.



Measurement definition: fill in the place holders

measurement_name (1, [2, …, n][, subarg1, subarg2, …, subargm]) = marked_point_expression

{

1| search_commands_and_marked_points_for_expression_1;

2| search_commands_and_marked_points_for_expression_2;

n| search_commands_and_marked_points_for_expression_n;

Measurement name syntax

Can contain any alphanumeric character (A-Z, 0-9) or underscore _ , up to 50 characters in length. The first character should be an upper or lower case letter.

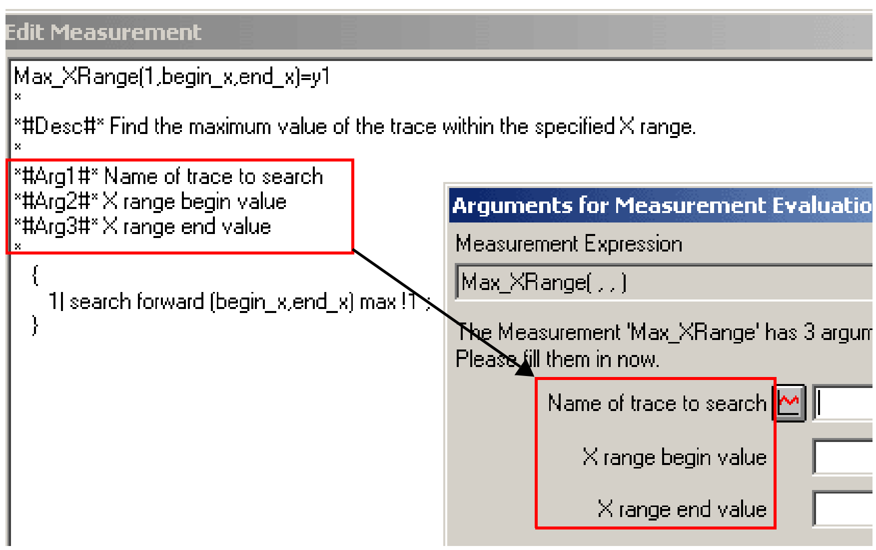
Examples of valid function names: Bandwidth, CenterFreq, delay_time, DBlevel1.

Comments syntax

A comment line always starts with an asterisk. Special comment lines include the following examples:

#Desc#	The measurement description
#Arg1#	Description of an argument used in the measurement definition.

These comment lines will be used in dialog boxes, such as the **Arguments for Measurement Evaluation** box.



Marked Point Expressions syntax

A marked point expression calculates a single value, which is the value of the measurement, based on the X and Y coordinates of one or more marked points on a curve. The marked points are found by the search command.

All the arithmetic operators (+, -, *, /, ( )) and all the functions that apply to a single point (for example, ABS(), SGN(), SIN(), SQRT()) can be used in marked point expressions.

The result of the expression is one number (a real value).

Marked point expressions differ from a regular expression in the following ways:

- Marked point coordinate values (for example, x1, y3), are used instead of simulation output variables (v(4), ic(Q1)).
- Multiple-point functions such as d(), s(), AVG(), RMS(), MIN(), and MAX() cannot be used.
- Complex functions such as M(), P(), R(), IMG(), and G() cannot be used.
- One additional function called MPAVG can also be used. It is used to find the average Y value between 2 marked points. The format is:

  MPAVG(p1, p2,[<.fraction>])

  where p1 and p2 are marked X points and fraction (expressed in decimal form) specifies the range. The range specified by [<.fraction>] is centered on the midpoint of the total range. The default value is 1.

Example:

The marked point expression

MPAVG (x1, x5, .2)

will find the halfway point between x1 and x5 and will calculate the average Y value based on the 20 percent of the range that is centered on the halfway point.

Search command syntax

search [direction] [/start_point/] [#consecutive_points#] [(range_x [,range_y])]

[for]

[repeat:] <condition>

Brackets indicate optional arguments.

You can use uppercase or lowercase characters, because searches are case independent.

[direction]

forward or backward

The direction of the search. Search commands can specify either a forward or reverse direction. The search begins at the origin of the curve.

[Forward] searches in the normal X expression direction, which may appear as backwards on the plot if the X axis has been reversed with a user-defined range.

Forward is the default direction.

[/start_point/]

The starting point to begin a search. The current point is the default.

Use this…	To start the search at this…
^	the first point in the search range
Begin	the first point in the search range
$	the last point in the search range
End	the last point in the search range
xn	a marked point number or an expression of marked points, for example, x1 (x1 - (x2 - x1) / 2)

[#consecutive points#]

Defines the number of consecutive points required for a condition to be met. Usage varies for individual conditions; the default is 1.

A peak is a data point with one neighboring data point on both sides that has a lower Y value than the data point.

If *[#consecutive_points#]* is 2 and *<condition>* is PEak, then the peak searched for is a data point with two neighboring data points on both sides with lower Y values than the marked data point.

[(range_x[,range_y])]

Specifies the range of values to confine the search.

The range can be specified as floating-point values, as a percent of the full range, as marked points, or as an expression of marked points. The default range is all points available.

Examples

This range…	Means this…
(1n,200n)	X range limited from 1e-9 to 200e-9, Y range defaults to full range
(1.5,20e-9,0,1m)	both X and Y ranges are limited
(5m,1,10%,90%)	both X and Y ranges are limited
(0%,100%,1,3)	full X range, limited Y range
(,,1,3)	full X range, limited Y range
(,30n)	X range limited only on upper end

[for] [repeat:]

Specifies which occurrence of *<condition>* to find.

If repeat is greater than the number of found instances of *<condition>*, then the last *<condition>* found is used.

Example

The argument 2:LEvel would find the second level crossing.

<condition>

Must be exactly one of the following:

- ❑ LEvel(value[,posneg])

- SLope[(posneg)]
- PEak
- TRough
- MAx
- MIn
- POint
- XValue(value)

Each *<condition>* requires just the first 2 characters of the word. For example, you can shorten LEvel to LE.

If a *<condition>* is not found, then either the cursor is not moved or the measurement expression is not evaluated.

LEvel(vahlue[,posneg])

[,posneg] Finds the next Y value crossing at the specified level. This can be between real data points, in which case an interpolated artificial point is created.

At least [#consecutive_points#]-1 points following the level crossing point must be on the same side of the level crossing for the first point to count as the level crossing.

[,posneg] can be Positive (P), Negative (P), or Both (B). The default is Both.

(value) can take any of the following forms:

Value form	Example
a floating number	1e5 100n 1
a percentage of full range	50%

Value form	Example
a marked point	x1 y1
or an expression of marked points	(x1-x2)/2
a value relative to startvalue	.-3 ⇒ startvalue -3 .+3 ⇒ startvalue +3
a db value relative to startvalue	.-3db ⇒ 3db below startvalue .+3db ⇒ 3db above startvalue
a value relative to max or min	max-3 ⇒ maxrng -3 min+3 ⇒ minrng +3
a db value relative to max or min	max-3db ⇒ 3db below maxrng min+3db ⇒ 3db above minrng

decimal point (.)

A decimal point (.) represents the Y value of the last point found using a search on the current trace expression of the measurement expression. If this is the first search command, then it represents the Y value of the startpoint of the search.

SLope[(posneg)]

Finds the next maximum slope (positive or negative as specified) in the specified direction.

[(posneg)] refers to the slope going Positive (P), Negative (N), or Both (B). If more than the next [#consecutive_points#] points have zero or opposite slope, the Slope function does not look any further for the maximum slope.

Positive slope means increasing Y value for increasing indices of the X expression.

The point found is an artificial point halfway between the two data points defining the maximum slope.

The default [(posneg)] is Positive.

PEak

Finds the nearest peak. At least [#consecutive_points#] points on each side of the peak must have Y values less than the peak Y value.

TRough

Finds nearest negative peak. At least [#consecutive_points#] points on each side of the trough must have Y values greater than the trough Y value.

MAx

Finds the greatest Y value for all points in the specified X range. If more than one maximum exists (same Y values), then the nearest one is found.

MAx is not affected by [direction], [#consecutive_points#], or [repeat:].

MIn

Finds the minimum Y value for all points in the specified X range.

MIn is not affected by [direction], [#consecutive_points#], or [repeat:].

POint

Finds the next data point in the given direction.

XValue(value)

Finds the first point on the curve that has the specified X axis value.

The (value) is a floating-point value or percent of full range.

XValue is not affected by [direction], [#consecutive_points#], [(range_x [,range_y])], or [repeat:].

(value) can take any of the following forms:

Value form	Example
a floating number	1e5
	100n
	1
a percentage of full range	50%
a marked point	x1
	y1
or an expression of marked points	(x1+x2)/2
a value relative to startvalue	.-3 ⇒ startvalue -3
	.+3 ⇒ startvalue +3
a db value relative to startvalue	.-3db ⇒ 3db below startvalue
	.+3db ⇒ 3db above startvalue
a value relative to max or min	max-3 ⇒ maxrng -3
	min+3 ⇒ minrng +3

Syntax example

The measurement definition is made up of:

- A measurement name
- A marked point expression
- One or more search commands enclosed within curly braces

This example also includes comments about:

- The measurement definition
- What arguments it expects when used
- A sample command line for its usage

Any line beginning with an asterisk is considered a comment line.

Risetime definition

```
Risetime(1) = x2-x1
    *
    *#Desc#* Find the difference between the X values
    *#Desc#* where the trace first crosses 10% and then
    *#Desc#* 90% of its maximum value with a positive
    *#Desc#* slope.
    *#Desc#* (i.e. Find the risetime of a step response
    *#Desc#* curve with no overshoot. If the signal has
    *#Desc#* overshoot, use GenRise().)
    *
    *#Arg1#* Name of trace to search
    *
    * Usage:
    *Risetime(<trace name>)
    *
       {
          1|Search forward level(10%, p) !1
            Search forward level(90%, p) !2;
       }
```

The name of the measurement is Risetime. Risetime will take 1 argument, a trace name (as seen from the comments).

The first search function searches forward (positive x direction) for the point on the trace where the waveform crosses the 10% point in a positive direction. That point’s X and Y coordinates will be marked and saved as point 1.

The second search function searches forward in the positive direction for the point on the trace where the waveform crosses the 90% mark. That point’s X and Y coordinates will be marked and saved as point 2.

The marked point expression is x2-x1. This means the measurement calculates the X value of point 2 minus the X value of point 1 and returns that number.

9

Optimization Engines

In this chapter

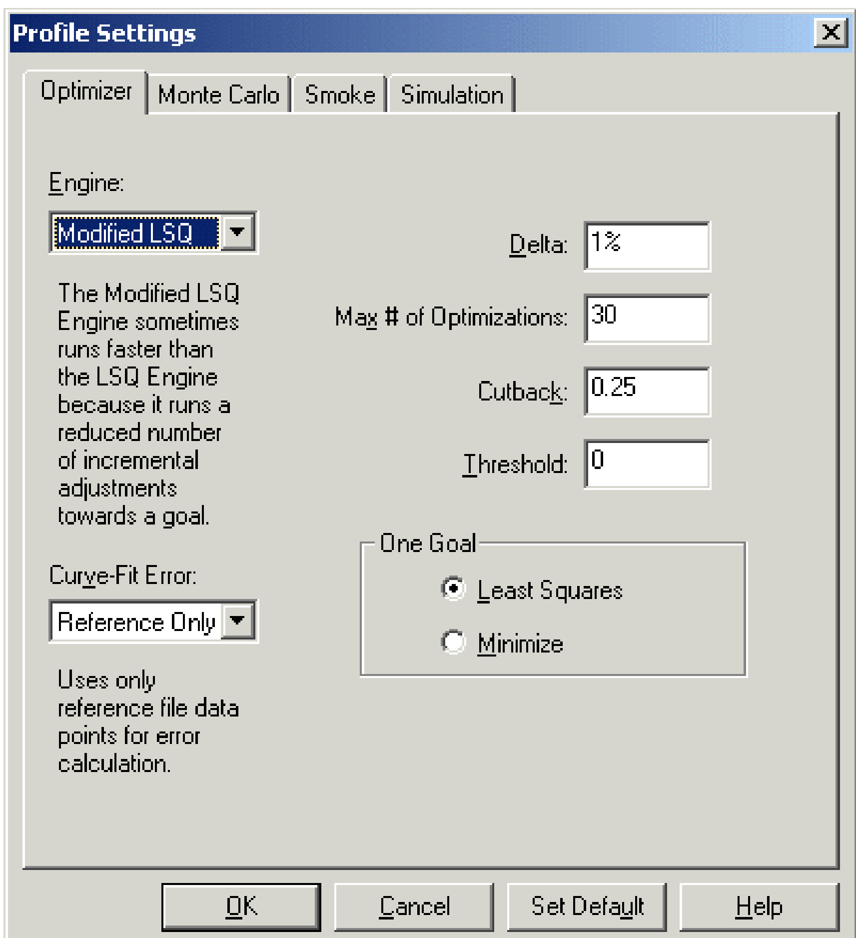
- Modified LSQ engine on page 286
- Random engine on page 291
- Discrete engine on page 294

Modified LSQ engine

The Modified LSQ engine uses both constrained and unconstrained minimization algorithms, which allow it to optimize goals subject to nonlinear constraints.

Configuring the Modified LSQ engine

1. From the Advanced Analysis **Edit** menu, select **Profile Settings**.
2. Click the **Optimizer** tab.
3. From the **Engine** drop-down list, select **Modified LSQ**.



4. Edit default values in the text boxes.

See detailed explanations provided on the next few pages.

5. Select the **One Goal** option that you prefer: **Least Squares** or **Minimize**.

   See Single goal optimization settings on page 290 for details.

6. Click **OK**.

Modified LSQ Engine Options	Function	Default Value
Delta	The relative amount (as a percentage of current parameter value) the engine moves each parameter from the proceeding value when calculating the derivatives.	1%
Max # of Optimizations	The most attempts the Modified LSQ Engine should make before *giving up* on the solution (even if making progress).	20
Cutback	The minimum fraction by which an internal step is reduced while the Modified LSQ Engine searches for a reduction in the goal's target value. If the data is noisy, consider increasing the Cutback value from its default of 0.25.	0.25
Threshold	The minimum step size the Modified LSQ engine uses to adjust the optimization parameters.	0

Delta calculations

The optimizer uses gradient-based optimization algorithms that use a finite difference method to approximate the gradients (gradients are not known analytically). To implement finite differencing, the Modified LSQ engine:

1. Moves each parameter from its current value by an amount Delta.
2. Evaluates the function at the new value.
3. Subtracts the old function value from the new.
4. Divides the result by Delta.

Note: There is a trade-off. If Delta is too small, the difference in function values is unreliable due to numerical inaccuracies. However if Delta is too large, the result is a poor approximation to the true gradient.

Editing Delta

Enter a value in the Delta text box that defines a fraction of the parameter's total range.

> Example: If a parameter has a current value of 10^{-8}, and Delta is set to 1% (the default), then the Modified LSQ Engine moves the parameter by 10^{-10}.

The 1% default accuracy works well in most simulations.

If the accuracy of your simulation is very different from typical (perhaps because of the use of a non-default value for either RELTOL or the time step ceiling for a Transient analysis), then change the value of Delta as follows:

- If simulation accuracy is better, smaller adjustments are needed; decrease Delta by an appropriate amount.
- If simulation accuracy is worse, larger adjustments are needed; increase Delta by an appropriate amount.

Note: The optimum value of Delta varies as the square root of the relative accuracy of the simulation. For example, if your simulation is 100 times more accurate than typical, you should reduce Delta by a factor of 10.

Threshold calculations

The Threshold option defines the minimum step size the Modified LSQ Engine uses to adjust the optimization parameters.

The optimizer assumes that the values measured for the specifications change continuously as the parameters are varied. In practice, this assumption is not justified. For some analyses, especially transient analyses, the measurement expression values show discontinuous behavior for small parameter changes. This can be caused by accumulation of errors in iterative simulation algorithms.

The hypothetical data glitch figure demonstrates a typical case. The effect of the glitch is serious—the optimizer can get stuck in the spurious local minimum represented by the glitch. The optimizer's threshold mechanism limits the effect of unreliable data.

Between iterations

Enter a value that defines a fraction of the current parameter value.

Example: A Threshold value of 0.01 means that the Modified LSQ Engine will change a parameter value by 1% of its current value when the engine makes a change.

By default, Threshold is set to 0 so that small changes in parameter values are not arbitrarily rejected. To obtain good results, however, you may need to adjust the Threshold value. When making adjustments, consider the following:

- If data quality is good, and Threshold is greater than zero, reduce the Threshold value to find more accurate parameter values.
- If data quality is suspect (has potential for spurious peaks or glitches), increase the Threshold value to ensure that the optimizer will not get stuck during the run.

Least squares / minimization

The Modified LSQ Engine implements two general classes of algorithm to measure design performance: least squares and minimization. These algorithms are applicable to both unconstrained and constrained problems.

Least squares

When optimizing for more than one goal, the Modified LSQ Engine always uses the least-squares algorithm. A reliable measure of performance for a design with multiple targets is to take the deviation of each output from its target, square all deviations (so each term is positive) and sum all of the squares. The Modified LSQ Engine then tries to reduce this sum to zero.

This technique is known as least squares. Note that the sum of the squares of the deviations becomes zero only if all of the goals are met.

Minimization

Another measure of design performance considers a single output and reduces it to the smallest value possible.

Example: Power or propagation delay, each of which is a positive number with ideal performance corresponding to zero.

Single goal optimization settings

When optimizing for more than one goal, the Modified LSQ Engine always uses the least-squares algorithm. For a single goal, however, you must specify the algorithm for the optimizer.

1. Do one of the following:

   ❑ Select the **Least Squares** option button to minimize the square of the deviation between the measured and target value.

   Or:

   ❑ Select the **Minimize** option button to reduce a value to the smallest possible value.

If your optimization problem is to maximize a single goal, then set up the specification to minimize the negative of the value.

For example: To maximize gain, set up the problem to minimize –*gain*.

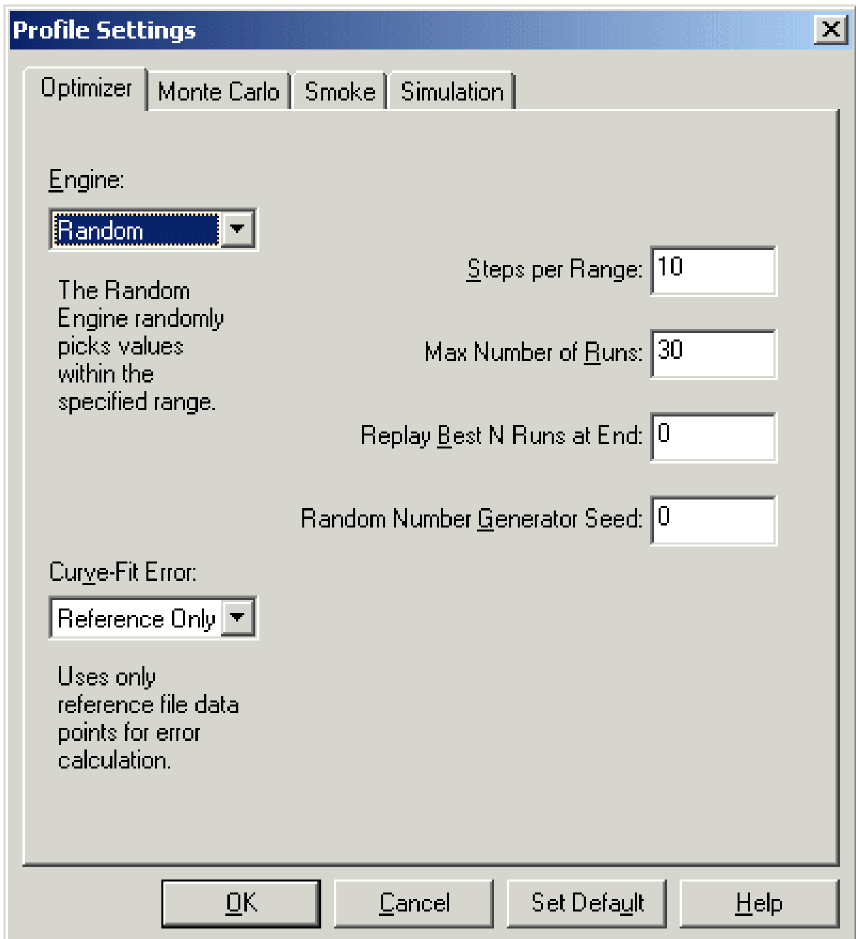
Random engine

When you use the Modified LSQ engines, it is sometimes difficult to determine where your starting points for optimization should be. The Random engine provides a good way to find these points.

The Random engine applies a grid to the design space and randomly runs analysis at the grid points. It keeps track of the grid points already run so that it never runs a duplicate set of parameter values. Once it finishes its initial analysis, it reruns the best points so you can easily use them for Modified LSQ.

Configuring the Random Engine

The Random Engine defaults are listed in a dialog box available from the **Optimizer** tab's, **Engine**, **Random** options.



To view and change the default options:

1. From the Advanced Analysis **Edit** menu, select **Profile Settings**
2. Click the **Optimizer** tab and select **Random** from the **Engine** drop-down list.
3. Edit the default value in the text box.

4. Click **OK**.

Random Engine Options	Default Value
Steps per Range	10
Max Number of Runs	10
Replay Best N Runs at End	0
Random Number Generator Seed	0

Steps per Range

Specifies the number of steps into which each parameter's range of values should be divided.

For example, if this option is set to 7 and you have the following parameters

Parameter	Min	Max
A	1	4
B	10	16

The possible parameter values would be

Parameter A = 1, 1.5, 2, 2.5, 3, 3.5, 4

Parameter B = 10, 11, 12, 13, 14, 15, 16

Max Number of Runs

Specifies the maximum number of random trial runs that the engine will run. The engine will run either the total number of all grid points or the number specified in this option, whichever is less.

Note: With 10 parameters, the number of grid points in the design exploration ($NumSteps^{\#params}$) would be 8^{10} = 1,073,741,824.

For example, if Max Number of Runs is 100, Steps per Range is 8, and you have one parameter being optimized, there will be 8 trial runs. However, if you have 10 parameters being optimized, then there will be 100 runs.

Replay Best N Runs at End

Specifies the number of “best” runs the engine should rerun and display at the end of the analysis.

Note: The Replay runs are done after the trial runs. If Max Number of Runs is 100 and Replay is 10, there may be up to 110 runs total.

Random Number Generator Seed

Specifies the seed for the random number generator. Unlike the Monte Carlo tool, the seed in this engine does not automatically change between runs. Therefore, if you rerun the Random engine without changing any values, you will get the same results.

Discrete engine

The Discrete engine finds the nearest commercially available value for a component. The other engines calculate component values, but those values might not be commercially available.

The discrete engine is a conceptual engine, rather than a true engine in that it does not actually perform an optimization, it finds available values from lists.

An example is a resistor that is assigned an optimal value of 1.37654328K ohms, which is not a standard value. Depending on the parameter tolerance and the manufacturer’s part number, the only values available might be 1.2K and 1.5K ohms. The Discrete engine selects parameter values based on discrete value tables for these parameters.

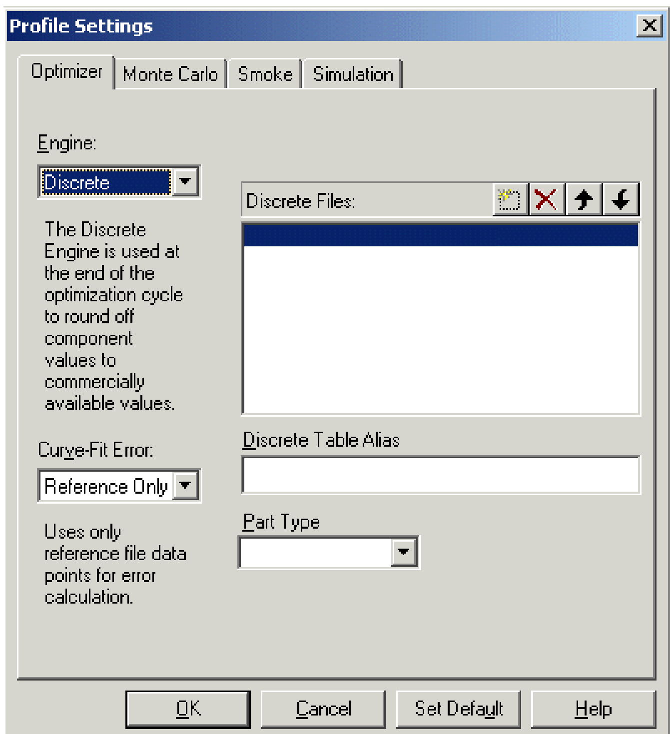
Once a value is selected, the engine makes a final run that lets you review the results in both the Optimizer and the output tools. If the results of the discrete analysis are not acceptable, the design can be optimized again to find another global minimum that might be less sensitive.

Commercially available values

Advanced Analysis includes discrete tables of commercially available values for resistors, capacitors, and inductors. These tables are text files with a .table file extension.

See "Assigning available values with the Discrete engine" on page 117 for instructions on selecting the discrete tables provided with Advanced Analysis Optimizer.

In addition, you can add your own discrete values tables to an Advanced Analysis project using the dialog box shown below. To know more about the adding user-defined discrete value tables, see Adding User-Defined Discrete Table on page 144.



After you have found commercial values for your design, you should run Monte Carlo and Sensitivity to ensure that the design is producible. Occasionally, the optimization process can find extremely good results, but it can be sensitive to even minor changes in parameter values.

10

Troubleshooting

In this chapter

- Troubleshooting feature overview on page 297
  - Procedure on page 298
  - Example on page 300
- Common problems and solutions on page 311

Troubleshooting feature overview

The Advanced Analysis troubleshooting feature returns you to PSpice[1] to analyze any measurement specification that is causing a problem during optimization.

Strategy

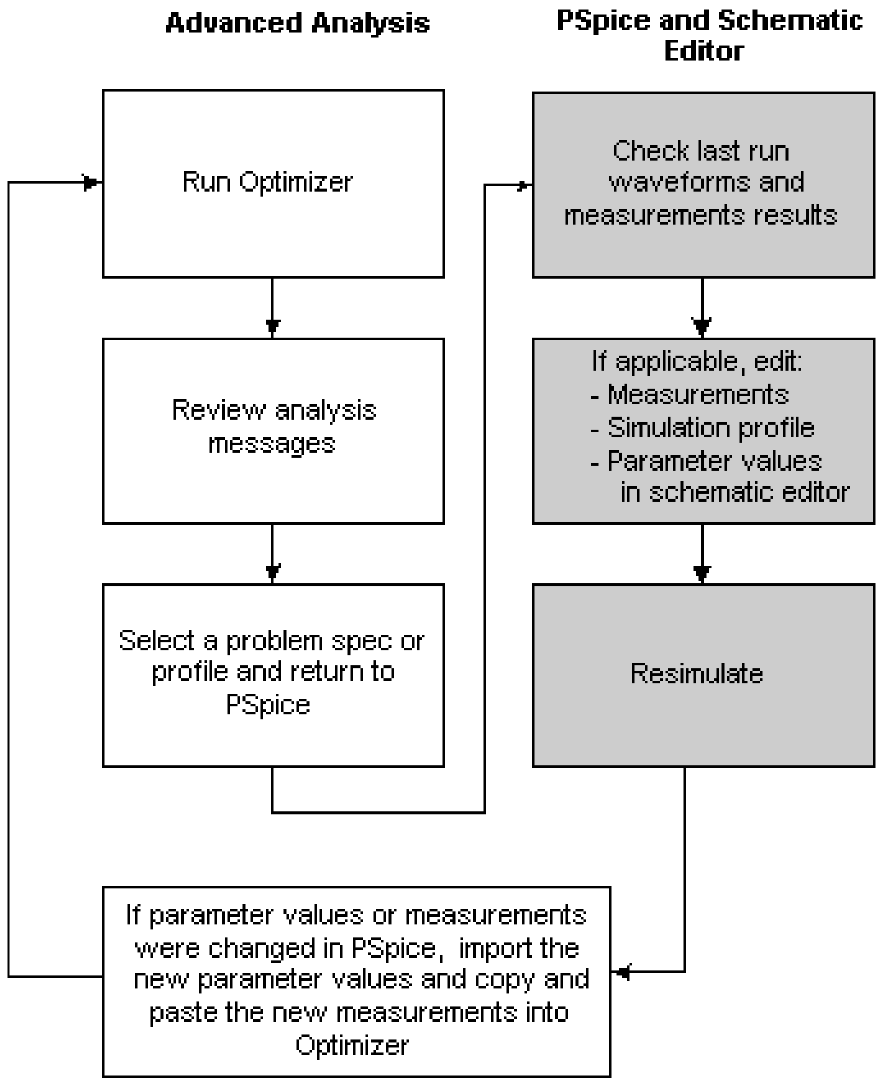
When an Optimizer analysis fails, the error message displayed in the output window or a yellow or red flag in the Specifications table shows you which measurement and simulation profile is associated with the failure.

If the failure is a simulation failure (convergence error) or a measurement evaluation error, the troubleshooting feature can help track down the problem.

From the Optimizer tool in Advanced Analysis, you can right click on a measurement specification and select **Troubleshoot in PSpice**. PSpice will display two curves, one with the data from the original schematic values and one with the data of the last analysis run.

1. Depending on the license available, you will access either PSpice or PSpice Simulator.

Workflow



Procedure

When an optimization analysis fails, you can use the troubleshooting feature to troubleshoot a problem specification.

Read the error message in the output window to locate the specification to troubleshoot, or look for a yellow or red flag in the first cell of a specification row.

1. Right click anywhere in the specification row you want to troubleshoot.

   A pop-up menu appears.

2. Select **Troubleshoot in PSpice**.

   PSpice opens and the measurement specification data is displayed in the window.

   The first trace shows the data from the run with the original schematic values.

   The second trace shows the data from the last run.

3. Right click on a trace, and from the pop-up menu select **Information**.

   A message appears about the trace data.

4. Make any needed edits:

   - In the PSpice window, check the measurement plot or click on  to view the simulation output file.
   - In the PSpice Measurements Results table, check the measurement syntax and the variables used.
   - In PSpice, click  to edit the simulation profile.
   - In the schematic editor, make changes to parameter values.

5. Rerun the simulation in the schematic editor.

6. Return to Advanced Analysis.

7. If you made changes:

   - To a measurement in PSpice, copy the edited measurement from PSpice to the Advanced Analysis Specifications table (Use Windows copy and paste)
   - To parameter values in your schematic editor, import the new parameter data by clicking on the Optimizer Parameters table row titled “Click here to import a parameter...”

8. Right click in the Error Graph and from the pop-up menu select **Clear History**.

9. Rerun Optimizer.

Example

To show how to use the troubleshooting feature, we need an optimization project that fails to find a solution. We'll use the example in the Troubleshoot folder from the Tutorial directory. This example results in an unresolved optimization.

Strategy

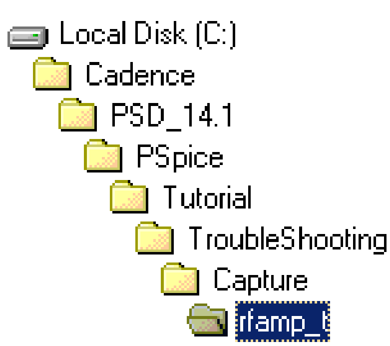
In this example we'll:

- Open the RF amp circuit in the Troubleshooting directory
- Run the AC simulation and open Optimizer
- Use the troubleshoot function to view waveforms of the problem measurement

Setting up the example

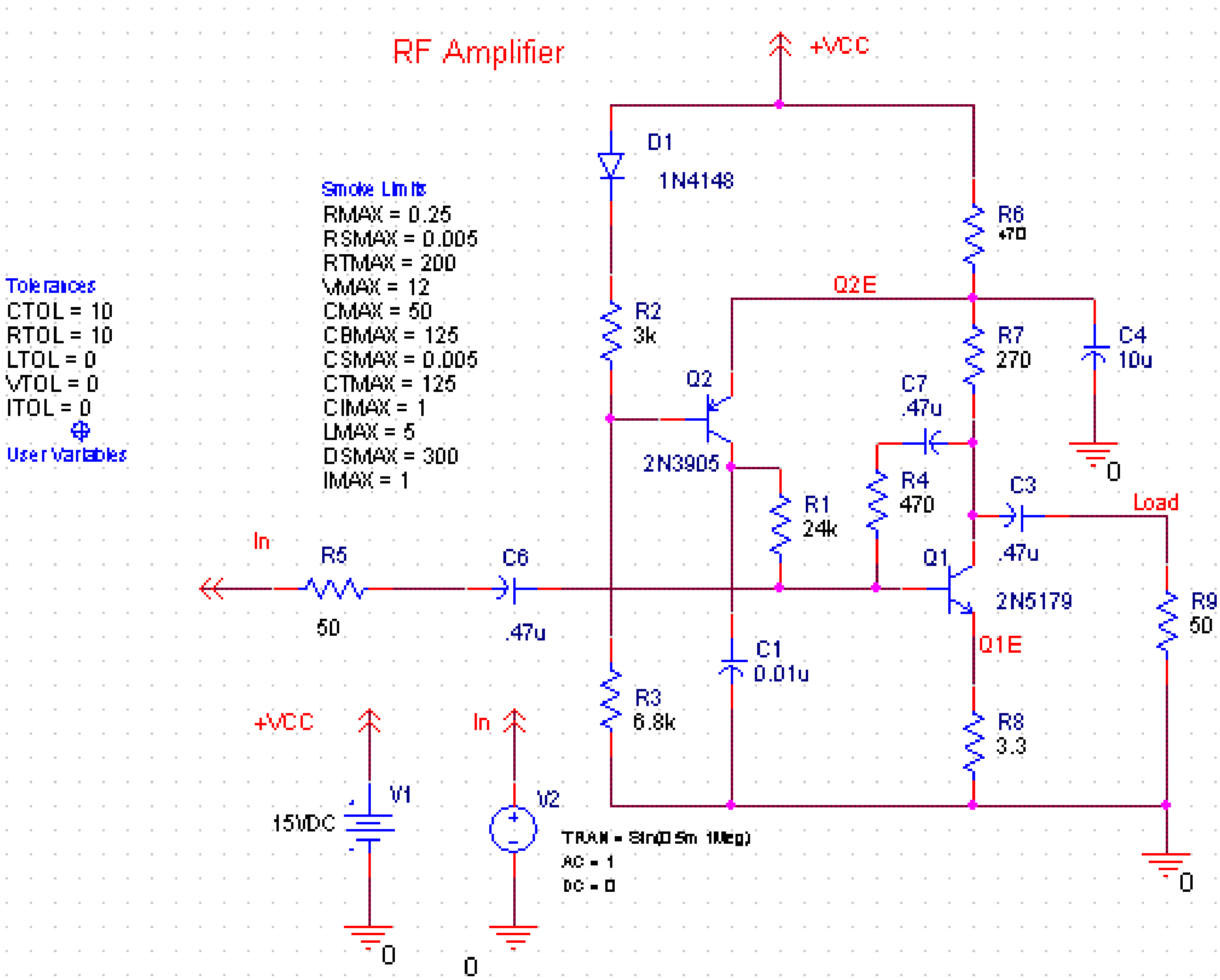
1. In your schematic editor, browse to the TroubleShoot directory:

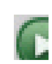
   <target directory> \ PSpice \ Tutorial \



2. From your schematic editor, open the rfampt project from the rfampt folder.

3. Open the schematic page.



4. With the SCHEMATIC1-AC simulation profile selected, click  to run the simulation.

5. From **PSpice** menu in a Cadence design entry tool[1], select **Advanced Analysis** / **Optimizer**.

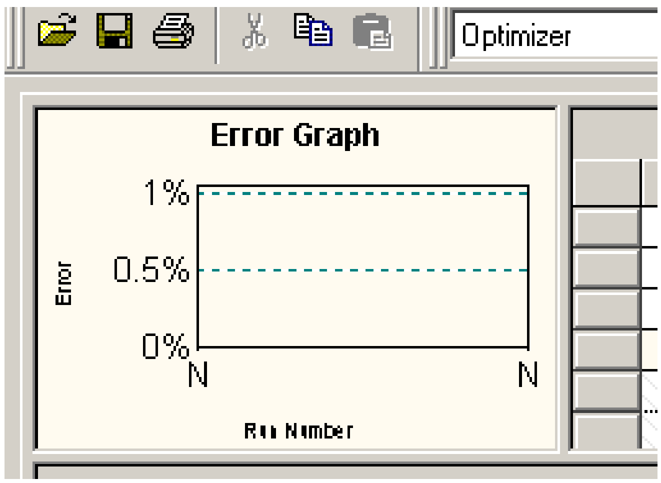
Advanced Analysis opens to the Optimizer view.

1. In this guide, design entry tool is used for both OrCAD Capture and Design Entry HDL. Any differences between the two tools is mentioned, if necessary.

There are four measurement goals included in this example.

				Specifications [Next Run]				
	+	On/Off	Profile	Measurement	Min	Max	Type	Weight
		✔	rf_amp-schematic1...	Max(DB(V(Load)))	5	5.5000	Constraint	20
▸		✔	rf_amp-schematic1...	Bandwidth(V(Load),3)	200meg		Goal	1
		✔	rf_amp-schematic1...	Min(10*Log10(V(inoise...		5	Constraint	1
		✔	rf_amp-schematic1...	Max(V(onoise))		3n	Constraint	20

6. If there is any history in the Error Graph, right click in the error graph window and select **Clear History** from the pop-up menu.

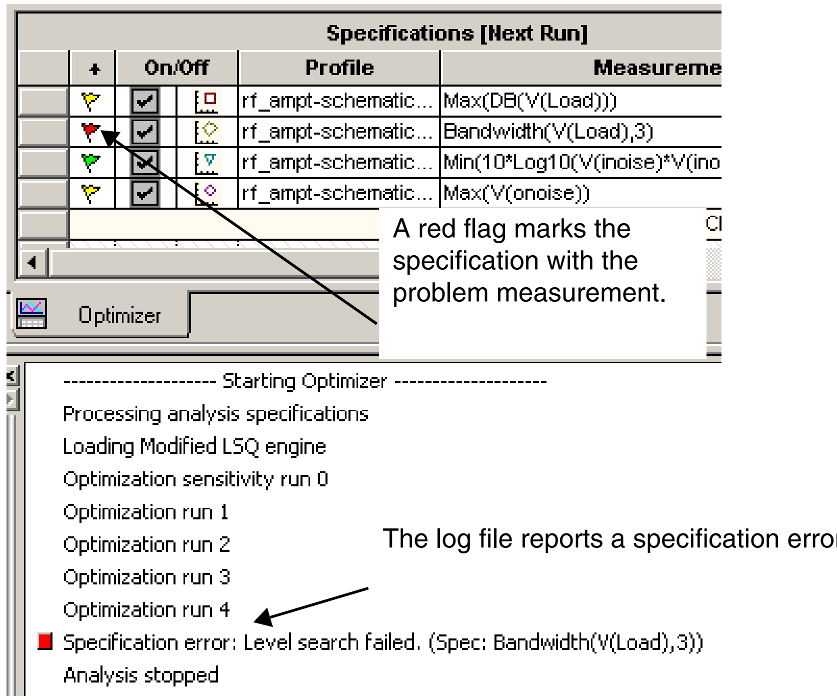


Error Graph with history cleared

7. Make sure the **Modified LSQ** engine is selected and click on the top toolbar to start the optimization.

Click to start optimization

The optimization starts and makes four run attempts.

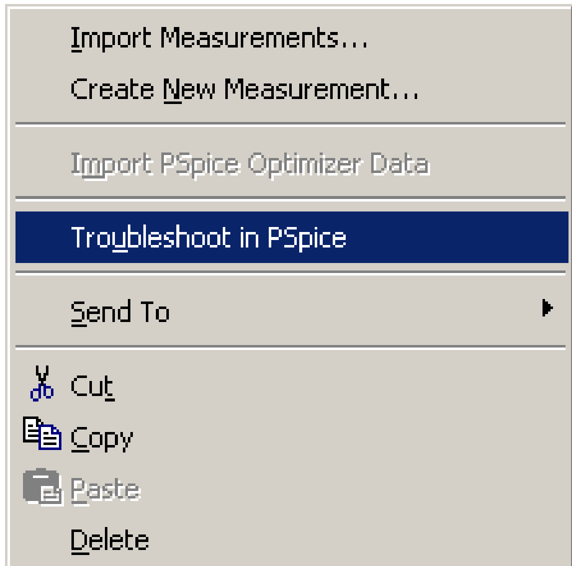


The Optimizer failed to find a solution. Let's troubleshoot the problem measurement in PSpice.

Using the troubleshooting function

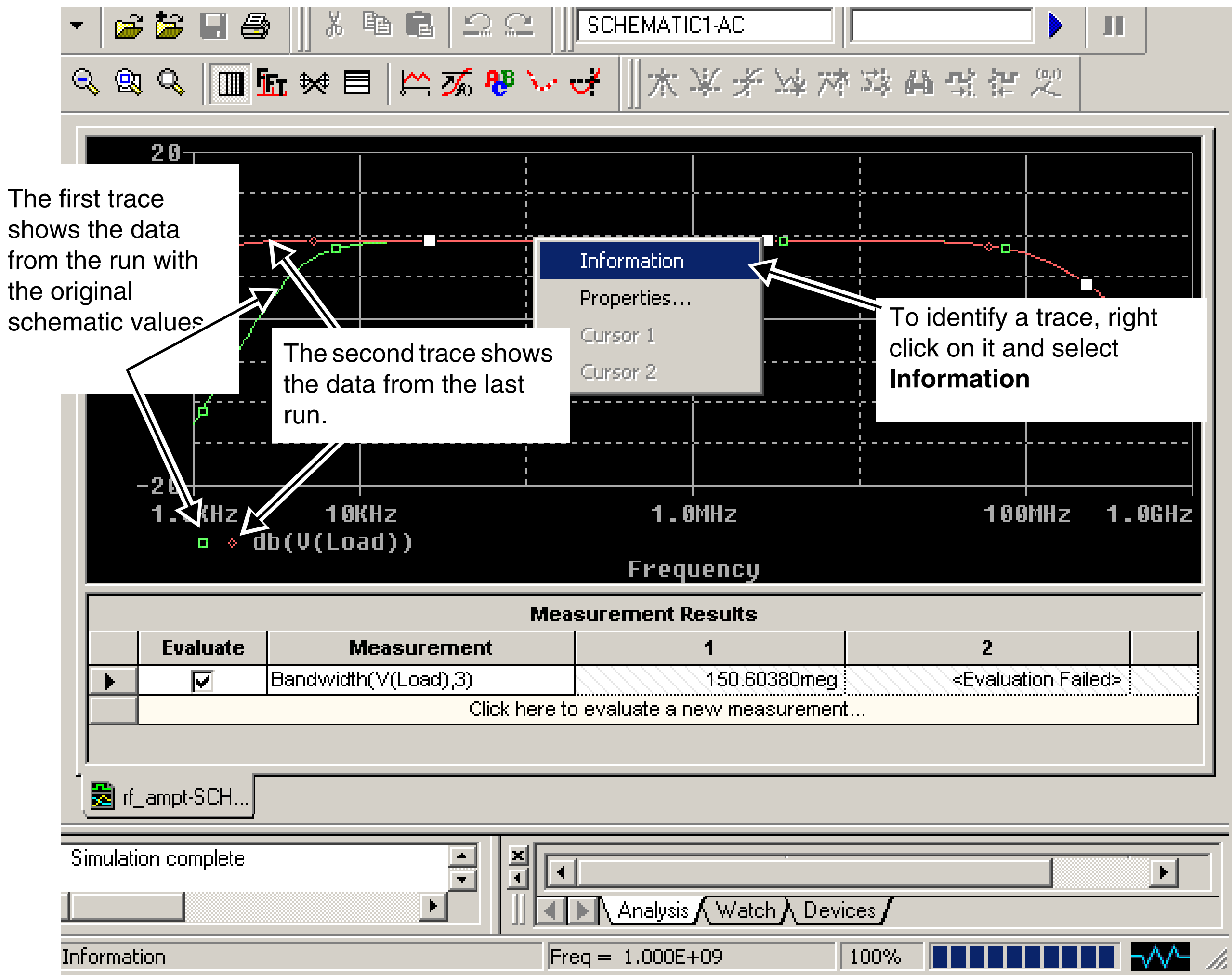
1. Right click in the specification row marked by the red flag (second row, Bandwidth(V(Load),3)).

A pop-up menu appears.

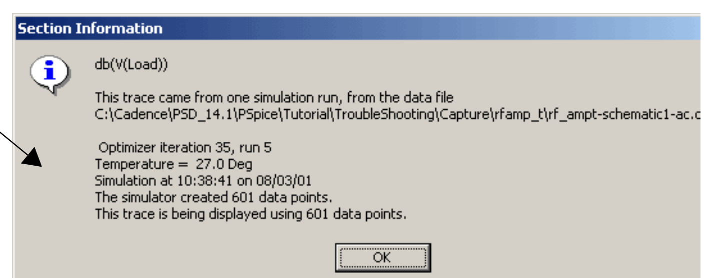


2. From the pop-up menu, select **Troubleshoot in PSpice**.

PSpice opens and the measurement specification data displays in the window.



A message appears about the trace data

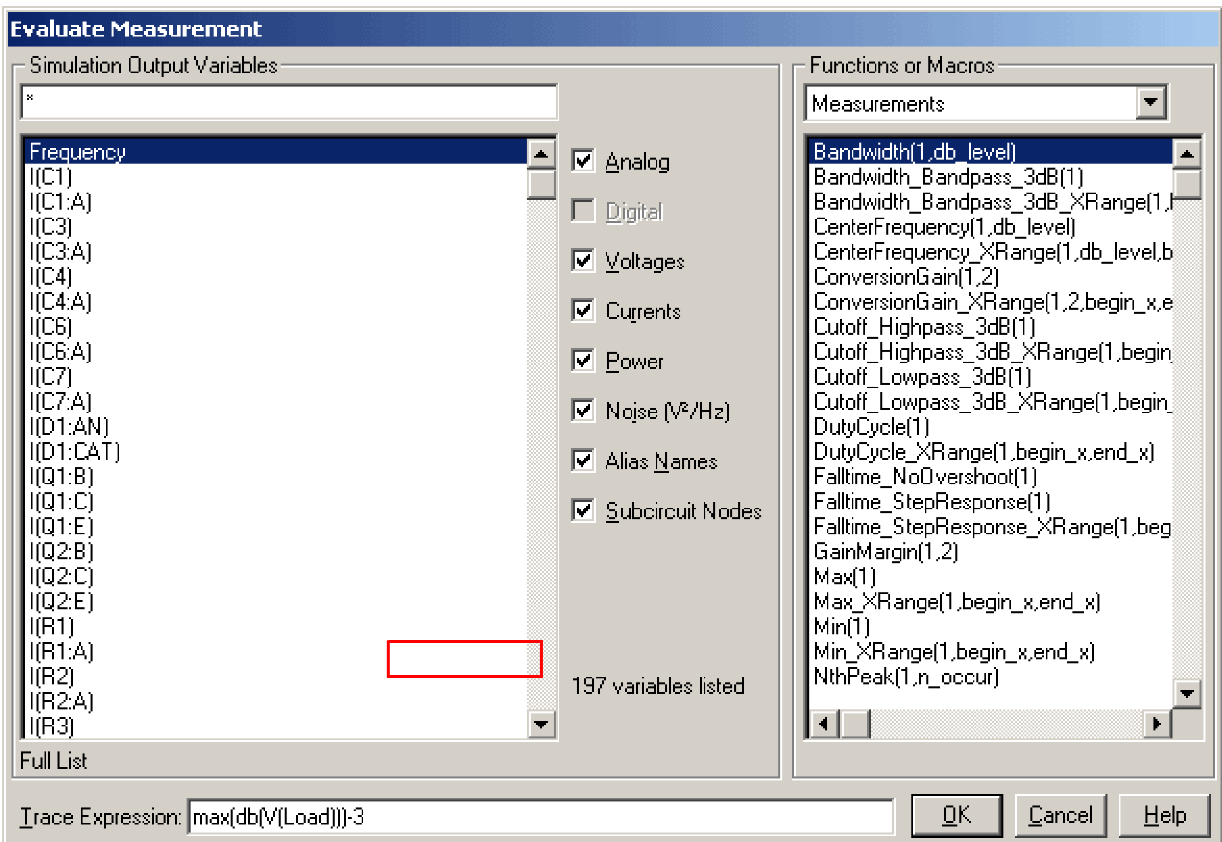


Analyzing the trace data

We know the bandwidth constraint failed. We'll add a measurement in PSpice to find the -3dB point of the trace.

1. Click at the bottom of the Measurements Results table.

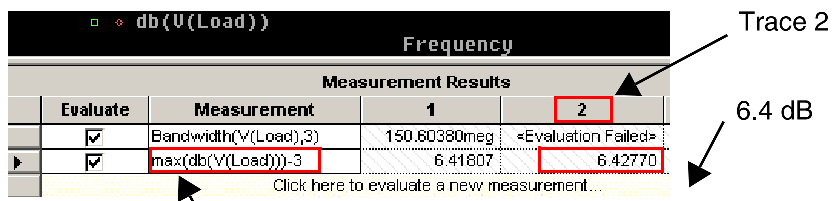
   The Evaluate Measurement dialog box appears.



2. In the Trace Expression field at the bottom, type in:

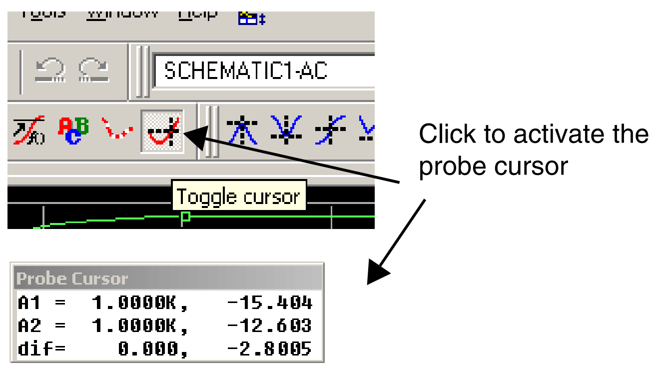
   max(db(v(load)))-3

A measurement that calculates the -3dB point appears in the Measurement Results table.

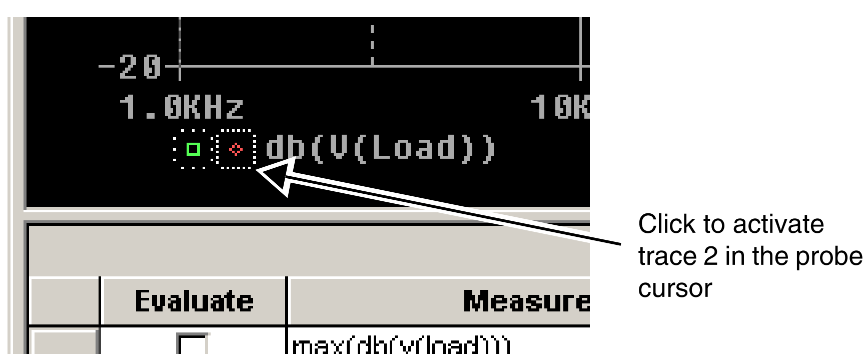


The new measurement shows that the -3dB point of trace 2 is at 6.4 dB

3. Click  to enable the Probe cursor.



4. Activate trace 2 in the probe cursor.



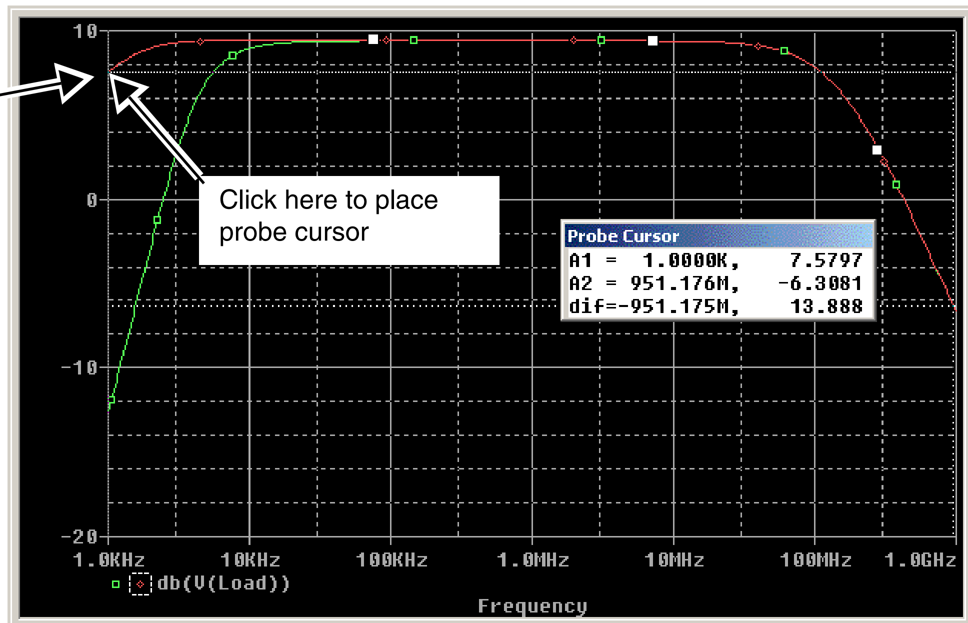
5. Click at the left end of trace 2.

   The probe cursor shows that trace 2's -3dB point (6.4dB) occurs before 1kHz.

The Optimizer is increasing the bandwidth as we asked it to in the measurement specification, but not exactly in the way we wanted.

While this results show a slightly higher bandwidth, we are more interested in increasing the cut-off frequency.

At 1kHz, the gain is 7.58 dB



Resolving the optimization

One solution may be to introduce a specification that keeps the low frequency cutoff above 1kHz, but this would complicate the optimization and take longer to complete.

Another solution may be to simplify things. It could be that we have given the optimizer too many degrees of freedom (parameters), some of which may not be necessary for meeting our goals.

Let's check out the bandwidth measurement in Sensitivity to see which components are the most sensitive.

Sensitivity check

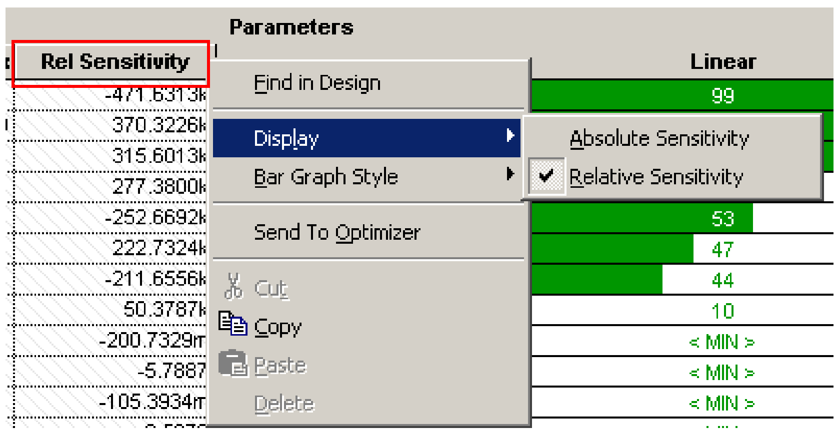
1. Return to Advanced Analysis and from the View menu, select **Sensitivity**.

The Sensitivity tool opens.

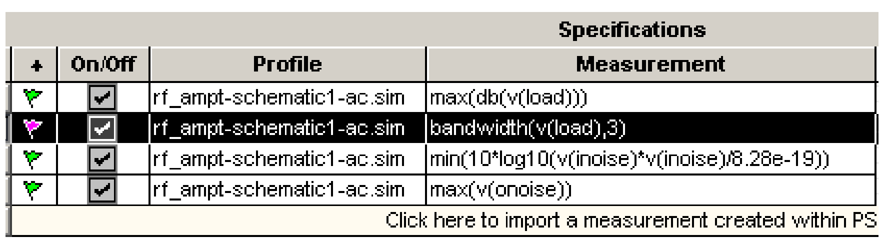
2. Make sure **Rel Sensitivity** is displayed in the Parameters table.

   If you need to change the display from absolute to relative sensitivity:

   ❑ Right click and from the pop-up menu choose Display / Relative Sensitivity.



3. In the Specifications table, select the bandwidth measurement (second row).



4. Click  on the top toolbar to start the sensitivity analysis.

   Sensitivity runs.

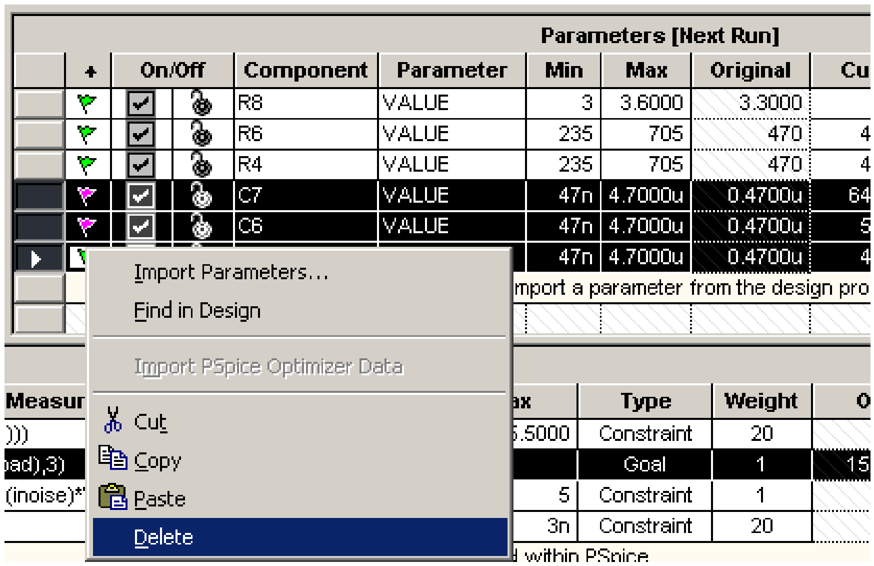
We can see that in the relative sensitivity analysis, Capacitors 3, 6, and 7 are not critical to the bandwidth response.

We'll return to Optimizer and remove the capacitors from the optimization analysis. Reducing variables may help Optimizer reach a solution.

Optimizer rerun

1. Return to the Optimizer tool and in the Parameters table, hold down your shift key and select the capacitor rows.

2. Right click and select **Delete** from the pop-up menu.



3. If there is any history in the Error Graph, right click in the Error Graph window and select **Clear History** from the pop-up menu.

4. Select the Modified LSQ engine and click  on the top toolbar to start the optimization.

   The optimization starts and finds a solution.

Common problems and solutions

This section suggests solutions to problems you may encounter in any of the Advanced Analysis tools.

Check the following tables for answers these problems:

- Analysis fails
- Results are not what you expected
- Can't make user interface do what you want
- Not enough disk space or memory

Analysis fails

Problem: Analysis fails	Possible cause	Solution
Smoke analysis won't run.	May not have a transient profile in the design. If a transient profile is included in the design, Smoke automatically picks the first transient profile for the analysis.	Smoke analysis only works if you have one or more transient profiles. Smoke does not work on AC or DC sweeps.
Smoke analysis won't run: message says "cannot find .dat file."	Transient analysis simulation may not be done.	Simulate the transient analysis in PSpice, review the waveform and measurement results, then run Smoke.

Problem: Analysis fails	Possible cause	Solution
Smoke analysis fails: Output window displays the following error for smoke parameters: “Data not found for Smoke test. Please verify Save Data and Data Collection options in the simulation profile”	Data save start time is not zero or data collection options for voltages, currents and power is not set to **All**.	From the **Simulation** menu in PSpice, choose **Edit Profile** to open the **Simulation Settings** dialog box. Ensure that the data save start time in the **Analysis** tab is 0. Smoke analysis works only if data save start time is zero seconds. Or From the **Simulation** menu in PSpice, choose **Edit Profile** to open the **Simulation Settings** dialog box. Ensure that the data collection options in the **Data Collection** tab is set to **All** for voltages, currents and power.
Monte Carlo analysis takes too long.	The number of runs may be too large.	Decrease the number of runs in the Monte Carlo settings tab (from the **Edit** menu, select **Profile Settings** and click the **Monte Carlo** tab).
I get an evaluation error message.	You might be using the wrong profile for the type of measurement you’re evaluating.	Check the selected profile and change it to the profile that applies to your measurement. For example, change to an AC profile to evaluate bandwidth.

Problem: Analysis fails	Possible cause	Solution
Optimization didn't converge.	The engine may have found a local minimum, which may not be the best solution.	Use the Random engine to search for alternate starting points. Go to the Error Graph history and copy the best Random engine result to the Nth run (the end). Then switch to the Modified LSQ engine to pinpoint the final answer.
Optimization didn't converge after running through several iterations.	The parameters have changed the circuit's behavior, so the simulation results may not provide the information needed to meet the measurement goal.	Use the **Troubleshoot in PSpice** feature to check the shapes of the traces and make sure they are appropriate for the desired measurement (right click on a measurement row and select the Troubleshoot command from the pop-up menu). For example, do the traces show that the filter still looks like a bandpass? Try changing the simulation settings to increase the range of frequencies. Or Restrict the parameter ranges in the Optimizer **Parameters** table to prevent the problem.
Optimization didn't converge, but it looked like it was improving.	Too few iterations.	Increase number of iterations in the Optimizer engine settings tab (from the **Edit** menu, select **Profile Settings** and click the **Optimizer** tab.)

Problem: Analysis fails	Possible cause	Solution
Optimization didn't converge. Parameters didn't change much from their original values.	Selected parameters may not be sensitive to the chosen measurement.	Choose different parameters more sensitive to the chosen measurement.
Optimization didn't converge. It was improving for a few iterations, then the Error Graph traces flattened out.	One or more parameters may have reached its limit.	If appropriate, change the range of any parameter that is near its limit, to allow the parameter to exceed the limit. If the limit cannot be changed, you may want to disable that parameter because it is not useful for optimization and will make the analysis take longer.

Results are not what you expected

Return to top of table.

Problem: Results are not what you expected	Possible cause	Solution
I set up my circuit and ran Smoke, but I’m not getting the results I expected.	Your components may not have smoke parameters specified.	Replace your existing components with those containing smoke parameters. or For R, L, and C components, add the design variables table (default variables) to your schematic. This table contains default smoke parameters and values. See the Libraries chapter of this manual for instructions on how to add this table to your schematic. or Add smoke parameters to your component models.
Smoke analysis peak results don’t look right: measured values are too small.	Transient analysis may not be long enough to include the expected peaks or may not have sufficient resolution to detect sharp spikes.	Check the transient analysis results in PSpice. Make sure the analysis includes any expected peaks. If necessary, edit the simulation profile to change the length of the simulation or to take smaller steps for better resolution.

Problem: Results are not what you expected	Possible cause	Solution
Smoke analysis average or RMS measured results are not what I expected.	Transient analysis may not be set up correctly.	Check the transient analysis results in PSpice. Make sure the average of voltages and currents over the entire range is the average value you're looking for. If you want the measurement average to be based on steady-state operation, make sure the analysis runs long enough and that you only save data for the period over which you want to average.
I selected a custom derating or standard derating file in Smoke, but my %Derating and %Max values didn't change.	Need to click the Run button to recalculate the Smoke results with the new derating factors.	In Smoke, click on the top toolbar and wait for the new values to appear.
My Smoke result has a yellow flag and a cell is grey.	The limit (average, RMS, or peak) is not typically defined for this parameter. Grey results show the calculated simulation values; however, grey results also indicate that comparison with the limit may not be valid.	The information is provided this way for user convenience, to show all calculated simulation values (average, RMS, and peak), but comparison to limits requires user interpretation. The color coding is intended to help.
The derating factor for the PDM smoke parameter isn't 100% even though I'm using No Derating.	This is OK. Smoke applies a thermal correction to the calculation.	None needed. This is normal behavior.
My Optimizer results don't look right. The current results are missing.	Your cursor might be set on a prior run in the Error Graph. The results you see are history.	In the Error Graph, click on the Nth (end) run's vertical line. Current results will appear in the **Parameters** table.

Problem: Results are not what you expected	Possible cause	Solution
In Optimizer, I finally get a good parameter value, but as I continue optimizing other things, the good parameter value keeps changing.	The good parameter value needs to be locked in so it won't change for the next runs.	In the Optimizer **Parameters** table, click the icon for the applicable parameter. This will close the lock and the parameter value will not change for subsequent runs.
In Optimizer, there aren't any discrete values listed for my component.	Discrete values tables are provided for RLCs. If your component is not an RLC, you'll have to create a discrete values table.	Create a discrete values table for your non-RLC component using instructions provided in "Adding User-Defined Discrete Table" on page 144.
Can't see the Optimizer discrete tables column.	Optimizer engine is not set to **Discrete**.	Change the Optimizer engine to **Discrete** in the drop-down list.
I can't find my individual Monte Carlo run results.	Raw measurement tab is not selected.	Click on the tab labeled **Raw Meas** to bring individual run results to the foreground on your screen.
I want more detail on my Monte Carlo graph.	Bin size is too small for desired detail.	Increase bin size in the Monte Carlo setting tab (from the **Edit** menu, select **Profile Settings** and click the **Monte Carlo** tab).
The Monte Carlo PDF / CDF graph doesn't look right for my measurement.	The applicable measurement row may not be highlighted.	Click on the measurement row. The resulting graph corresponds to that measurement.
I can't see the CDF graph.	Graph defaults to PDF view.	Right click the graph and select **CDF graph** from the pop-up menu.
I can't find the parameter values for my Monte Carlo runs.	Monte Carlo parameter values are only available in the log file.	From the **View** menu, select **Log File** / **Monte Carlo** and scroll through the file to the applicable run.

Can't make user interface do what you want

Return to top of table.

Problem: Can't make user interface do what you want	Possible cause	Solution
I can't get all my red bar graphs to appear at the top of my Smoke or Sensitivity tables.	Data isn't sorted.	Click twice on the bar graph column header. The first click puts all the red bars at the bottom. The second click puts them at the top.
I don't want to see the grey bars in Smoke.	Average, RMS, or peak limits that don't apply to your parameter may be selected for viewing.	Double click the message flag column header. This will sort the grey bars so they appear at the bottom of the data display. or Right click and uncheck the average, RMS, or peak values on the right click pop-up menu.
Why can't I use my Monte Carlo settings and results from PSpice A/D?	The programs are separate and use different input.	Advanced Analysis Monte Carlo provides more information and can be run on more than one specification simultaneously. This is the trade-off.
Monte Carlo cursor won't drag to a new location.	The cursor can be moved, but it doesn't use the drag and drop method.	Click once on the cursor. Click in your desired location. The cursor moves to the location of the second click.

Not enough disk space or memory

Return to top of table.

Problem: Not enough disk space or memory	Possible cause	Solution
I get a disk space error message or an out of memory message and I’m running a Monte Carlo analysis.	Too much data is being saved for the Monte Carlo runs. For example, in a 10,000-run Monte Carlo analysis where all data is collected and saved, the data file and memory usage may become very large.	Turn off the option to save all simulation waveform data in Advanced Analysis. By doing this, saved data will be limited to just the current run. However, at this setting, the simulation will run slower. To turn off the data storage: **1.** From the Advance Analysis menu select: Edit / Profile Settings/ Simulation tab **2.** From the Monte Carlo field, select **Save None** from the drop-down list Advanced Analysis will overwrite the data file for each run.

Problem: Not enough disk space or memory	Possible cause	Solution
I get a disk space error message or an out of memory message and I'm running a Monte Carlo analysis (continued).	Too much data is being collected for each simulation run. For instance, collecting voltages, currents, power, digital data, noise data, and all of these for internal subcircuit components results in a large data file and large memory use. Note: You can also place markers on nets, pins, and devices on the schematic and collect data at these marker locations. In PSpice, set the data collection option to **At Markers Only** for all the data types you want. See the schematic editor help for more information on how to use markers on the schematic.	Limit data collection to only the information that is needed to perform Advanced Analysis. You can do this in conjunction with the data file solution mentioned on the previous page or do just this and save data for all Monte Carlo runs. To change data collection options for each simulation, do the following for each simulation profile used in Advanced Analysis: **1.** From the PSpice Simulation menu, select Edit Profile. **2.** In the Simulation Settings dialog box, select the Data Collection tab. **3.** Set the data collection option to **None** for all the data types that are not required. Use the drop-down list to select the option. **4.** Set the data collection option to **All but Internal Subcircuits** for data required for Advanced Analysis. Use the drop-down list to select the option.

A

Property Files

PSpice[1] A/D has an additional feature called Advanced Analysis. Using Advanced Analysis, you can run the following analyses:

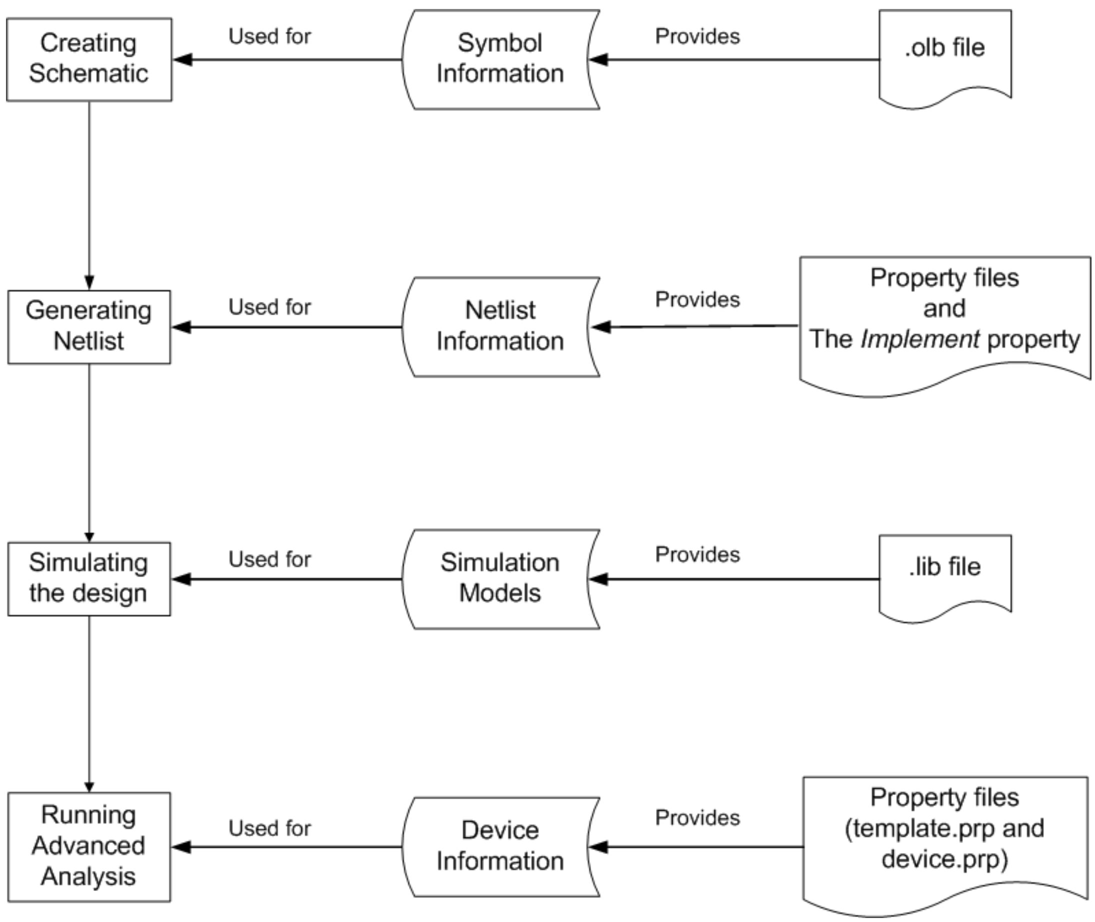
- Sensitivity
- Monte Carlo
- Optimizer
- Smoke
- Parametric Plotter

For Advanced Analysis runs along with the simulation data, Advanced Analysis needs other device-specific data as well. Device-specific data, such as device parameter tolerance and maximum operating conditions, is available in property files. These property files are shipped along with PSpice libraries.

Property files are organized as the template property file and the device property file. The template property file contains generic information for a particular class of devices. The device property file contains information specific to a device.

1. Depending on the license available, you will access either PSpice or PSpice Simulator.

The diagram shown below depicts the Capture-PSpice flow and the files used in the flow.



Template property file

The template property file (TEMPLATES.PRP) contains information for all device types supported by PSpiceAMS. Only the information that is common across a set of devices is available in the template property file. Model information contained in this file includes simulation information and smoke information.

The template property file contains definitions of simulation parameters. It also lists the default values and the units for each of the simulation parameters.

For smoke, it lists parameter definitions, node to port mapping information, and the list of the smoke tests to be performed for a particular device or a family of devices.

A template property file has the following sections:

- The model_info section
- The model_params section
- The smoke section
    - max_ops_desc
    - pre_smoke
    - max_ops
    - smoke_tests

The template for the TEMPLATES.PRP file is shown below:

```
("0"
    (Creator "Template property file created by 
analog_uprev on Wed Jan  3 09:57:42 IST 2001")
    ("model_info"
        ( ... )
    )
    (SMOKE
        ( "pre_smoke"
            (...)
        )
        ( "max_ops"
            (...)
        )
        ( "smoke_tests"
            (...)
        )
    )
)
(...)

("4"
    (Creator "....")
    ("model_info"
        (...)
    )
    ("model_params"
    ("level_0"
        ( "IS"
            ( ... )
        )
        ...
        )
    (SMOKE
        ...
        ...
    )
```

Table A-1 lists the sections of property files and the analysis in which these sections are used.

Table A-1 *Usage of different sections of a property file*

Statements/Sections in the property file...	Used in...
model_params	Optimization
	Monte Carlo analysis
	Sensitivity analysis
POSTOL and NEGTOL	Monte Carlo analysis
	Sensitivity analysis
DERATE_TYPE	Smoke analysis
smoke section	Smoke analysis
max_ops	Smoke analysis

The model_info section

A part of the TEMPLATES.PRP file containing the model_info section for an OPAMP model is shown below:

```
("739"
("model_info"
        ( SYMBOL_TYPE  "38" )
        ( DEFAULT_SYMBOL  "5_Pin_Opamp" )
        ( NAME  "FET Input Opamp" )
        ( "spice_dsg" "X" )
        ( "model_type" "M" )
    )
...
```

The first line in a template property file specifies the model template number. The model template number is used as a reference in the device property file to locate the generic model definition in the template property file.

The model_info section contains information such as symbol type, default symbol, symbol name, spice designator, and model type. Spice designator indicates the type of PSpice device. For example, the spice designator for an template-based diode model is X and the spice designator for the diode model based on device characteristic curves is D. Similarly, the model type can be either M for macro models or P for primitive models.

The model_params section

The model_params section lists all simulation parameters, along with the parameter types and the default values of the parameters, tolerances, and distributions.

All the parameters listed in this section are used for Sensitivity, Monte Carlo, and Optimizer runs. All of these properties can be made available to the Optimizer, provided they are added as properties on the part symbol in the schematic editor. These properties can also be used for Monte Carlo analysis if they have a POSTOL and NEGTOL place holders.

The model_params section starts with a level entry, which indicates the level of simulation parameters supported. For some of the models, there can be more that one level present in the property file. In case of multiple level models, as the parameter level goes higher, the number of simulation parameters included in the model increases. The highest level has all the simulation parameters of lower levels and some more simulation parameters.

For most of the models, the level is level_0 indicating that the model is a single-level model, and therefore, all the simulation parameters listed under level_0 are used while simulating the models.

If the level values are level_1, level_2, and level_3, the model is a multi-level model. For multi-level models, you can specify the simulation parameters to be used while simulating the device, using the LEVEL property on the device symbol. For example, if you specify the value of the LEVEL property as 2, only the simulation properties listed under level_1 and level_2 are used while simulating the device.

> **Note:** For some of the models, the simulation parameters are divided into different levels. The level of parameters determines the complexity of the model. Higher the level more complex is

the model. Level 1 indicates the lowest level of complexity. While simulating a device, you can specify the level of the simulation parameters to be used by adding the LEVEL property on the symbol in the schematic editor. Use Level 1 simulation parameters when you want to fast but not so accurate simulation results. Using Level 3 parameters increases the accuracy of simulation results but also increases the simulation time.

Template-based OPAMP models are an example of multi-level models supported by PSpiceAMS.

A part of the TEMPLATES.PRP file containing the model_params section for an OPAMP model is shown below.

```
...
("model_params"
    ("level_1"
    ( "VOS"
        ( "description" "Offset voltage" )
        ( "units" "V" )
        ( "val" "1e-6" )
        ...
("level_2"
 ( "CMRR"
    ( "description" "Common-mode reject." )
    ( "units" "V/V" )
    ( "val" "100000" )
    ...
    ...
```

Within the LEVEL section, various simulation parameters are defined. A parameter definition includes parameter description, measurement unit, and the default parameter value.

The information listed under the model_params section is used by the Model Editor also. The Model Editor reads this information and displays it in the Parameter Entry form.

The smoke section

This section of the template property file is used during the smoke analysis. The main objective of a smoke analysis is to calculate the safe operating limit of all the parts used in a circuit, given the

Maximum Operating Conditions (MOCs) for each device in the circuit. These MOCs are defined in the smoke section of the property file.

The smoke section of the template property file contains smoke parameter definitions and how to measure them for a particular device or family of devices. Smoke parameters are used for defining maximum conditions that can be applied to a device.

The max_ops_desc section

The max_ops_desc section contains the description of the smoke parameters along with the unit of measurement for the parameter. All the entries in this section are displayed in the smoke parameters window in Model Editor.

For example:

```
( "IPLUS"
    ("description"  "Max input current(+)" )
    ("unit"  "A" )
)
```

Where:

`IPLUS`	smoke parameter
`("description"  "Max input current(+)"`	description of IPLUS; maximum input current at the positive terminal.
`("unit"  "A" )`	unit of measurement is Ampere

The pre_smoke section

The pre_smoke section lists default mapping between the node names and the corresponding port names in the part symbol. This information is visible to you in the Test Node Mapping frame in the Model Editor. For template-based models, this information is not editable, but for non-parameterized models, you can edit this information. A sample of the pre_smoke section is shown below:

```
( "pre_smoke"
```

```
( NODE_POS "PIN" )
( NODE_NEG "NIN" )
( NODE_VCC "PVss" )
( NODE_VEE "NVss" )
( NODE_GND "0" )
( TERM_POS "-1" )
( TERM_NEG "-2" )
( TERM_OUT "-3" )
( DERATE_TYPE "OPAMP" )
```

The pre_smoke section also lists the derate type. The DERATE_TYPE line specifies the derate type to be used for the model. The derate types are defined in the STANDARD.DRT file.

Note: To find out more about derate types and derating files, see the chapter on Smoke.

Table A-2 lists the supported DERATE_TYPEs.

Table A-2 Supported derate type

DERATE_TYPE	Part
RES	Resistor
CAP	Capacitor
IND	Inductor
DIODE	Diode
NPN	NPN Bipolar Junction Transistor
PNP	PNP Bipolar Junction Transistor
JFET	Junction FET
N-CHANNEL	N-Channel JFET
P-CHANNEL	P-Channel JFET
NMESFET	N-Channel MESFET
PMESFET	P-Channel MESFET
MOS	MOSFET
NMOS	N-Channel MOSFET

Table A-2 Supported derate type

DERATE_TYPE	Part
PMOS	P-Channel MOSFET
OPAMP	Operational Amplifiers
ZENER	Zener Diode
IGBT	Ins Gate Bipolar Transistor
VARISTOR	Varistor
DIODE_BRIDE	Half Wave and Full Wave Rectifier
OCNN	
OCNPN	Octo Coupler using NPN transistor
THYRISTOR	Thyristor
SCR	Silicon Controlled Rectifier
VSRC	Voltage Source
C_REG_DIODE	Current Regulator Diode
POS_REG	Positive Voltage Regulator
LED	Light Emitting Diode
LASER	Laser
DUALNPN	Dual NPN Transistor
DUALPNP	Dual PNP Transistor
DUALNMOS	Dual NMOS
DUALPMOS	Dual PMOS
NPN_PNP	NPN and PNP transistors fabricated together
NMOS_PMOS	NMOS and PMOS fabricated together

Using DERATE_TYPE, the derating factor is read from the STANDARD.DRT file. This file lists the default derating factor for all the smoke parameters for a particular device.

Derating factor is the safety factor that you can add to a manufacturer's maximum operating condition (MOC). It is usually a percentage of the manufacturer's MOC for a specific component. MOCs, the derating factor, and Safe Operating Limits (SOL) are connected by the following equation.

$$\mathrm{MOC} \times \mathrm{derating\ factor} = \mathrm{SOL}$$

You can also create you own derate file. You can use the CUSTOM_DERATING_TEMPLATE.DRT file as the template for creating new derate files.

Tip

To find out how to create custom derating files see, Adding Custom Derate file.

The max_ops section in the template property file lists the default values of MOCs. This information can be overridden by the information contained in the device property file.

Finally, the smoke test section of a template property file defines the test performed and the nodes for which the test holds.

Example:

A section of the template property file defining the IMINUS smoke parameter is listed below:

```
( "IMINUS"
        ("test" "current_test" )
        ("term" TERM_NEG)
```

To test for the maximum input current at the negative terminal of OPAMP, Advanced Analysis runs the current_test.

Note: The actual value of the terminal is obtained from the device_pre_smoke or PORT_ORDER section of the corresponding DEVICE.PRP file.

A list of valid test types and their descriptions are listed in the table below:

Test Name	Descriptions
current_test	Finds current in the specified terminal
power_test	Finds power dissipation of the device
temp_null_test	
voltage_test	Finds voltage between two nodes
abs_voltage_test	Finds absolute voltage between two nodes
neg_current_test	Finds negative current in the specified terminal
breakdown test	Finds breakdown voltage between two nodes
abs_current_test	Finds absolute current at the terminal

The device property file

A device property file lists all the models associated with a device. A device property file lists the port order and maximum operating values or smoke parameter values entered by a user for a model. Information in the DEVICE.PRP file is divided into the device_info section and the device_max_ops section. Usually, the name of a device property file indicates the device type as well. For example, IGBT.PRP is the device property file for IGBT models based on device characteristic curves, and AA_IGBT.PRP is the device property file for IGBT models based on PSpice provided templates.

A sample device property file for a parameterized or a template-based model is shown below:

```
("awbad201a"
    (Creator "Device property file created by
analog_uprev on Thu Mar  1 18:48:14 IST 2001")
    ("device_info"
        ( MODEL_TYPE 739 )
        ( SYMBOL_NAME "7_Pin_Opamp" )
        ( PORT_ORDER
            ("PIN")
            ("NIN")
            ("OUT")
            ("PVSS")
            ("NVSS")
            ("CMP1")
            ("CMP2") )
    )
    ("model_params"
    ("level_1"
        ( "VOS"
            ( "val" "0.7m" )
            ( "postol" "1.3m" )
            ...
    ("level_2"
        ( "CMRR"
            ( "val" "6.3E4" )
        ...
    )
    ("level_3"
```

```
            ( "CINDM"
                ( "val" "1p" )
            ...
            ...
        )
        ("device_max_ops"
            ( VDIFF "30" )
            ( VSMAX "40" )
            ( VSMIN "0" )
        )
    )
```

The first line in a DEVICE.PRP file is the file header or indicates the name of the model. For example, in the section shown above, *awbad201a* is the model name. The prefix `awb` in the model name indicates that it is an parameterized model shipped with PSpiceAMS. Parts created using the Model Editor do not have the *awb* prefix.

Within a model definition, you have the following sections:

- device_info
- device_max_ops
- model_params

The device_info section

This section lists the MODEL_TYPE, SYMBOL_NAME, and PORT_ORDER. The first line in the device_info section specifies MODEL_TYPE. The syntax is

`( MODEL_TYPE` *`Numeric_value`* `)`

For example:

`( MODEL_TYPE` *`706`*`)`

MODEL_TYPE refers to the model template number in the template property file.

The line (SYMBOL_NAME "7_Pin_Opamp”) refers to the name of the schematic symbol. The line is used by the Model Editor during part creation. In the above example, the schematic symbol created by the Model Editor will have *7_Pin_Opamp* as the symbol name.

Finally, PORT_ORDER lists the pin names in the order of the interface nodes on the .SUBCKT statement in the PSpice model. The PORT_ORDER information is available only for template-based PSpice models and is used during netlist creation.

The model_params section of a device property file lists the default value of the simulation parameter, the default positive and negative tolerance values, and the default distribution type. By default, the distribution type is flat for all parameters. The distribution type is used during the Monte Carlo analysis.

Finally, the device_max_ops section displays the maximum operating values for each of the smoke parameters. If a smoke parameter for a model does not appear in this list, the default value as listed in the template property file is used.

The device_pre_smoke section

The device_pre_smoke section is present in the device property files of all the non-parameterized PSpice model libraries provided by OrCAD and the libraries that have been created or edited using the Model Editor.

The device_pre_smoke section lists the default mapping between the node names and the corresponding port names in the part symbol. This section is copied from the pre_smoke section of the template property file. The entries in the device_pre_smoke section have higher precedence than the default values specified in the pre_smoke section.

For the non-parameterized models, the port names entered by users in the Test Node Mapping section, are written in the device_pre_smoke section. Users can get the port names of a part by opening the symbol in a schematic editor. A part of the BIPOLAR.PRP file with the device_pre_smoke section is shown below.

```
("device_pre_smoke"
    (TERM_IC "C")
    (TERM_IB "B")
    (NODE_VC "C")
    (NODE_VB "B")
    (NODE_VE "E")
    (DERATE_TYPE "PNP")
```

Tip

To get the port names by opening the symbol in Capture:
1) Select the part in Capture.
2) From the Edit menu, choose Part. The symbol view of the part displays.
4) Double-click the pin. The Name field in the Pin Properties dialog box displays the port name.

Optional sections in a device property file

Some simulation models have more than one physical device attached to them. In such cases, though the simulation model for physical devices is the same, the device-specific information stored in the device property file is different. For example, each of the physical device can have different smoke data.

The device property files of the models that have more than one physical devices attached to them have an index section. The index section has an Implementation statement that lists all the physical devices associated with a model.

A section of the OPAMP.PRP file, with the Implementation statement is shown below:

Model name as appears in the .lib file.

```
("MAX403"
      (Creator  "Device property file created by
prp_generator on Sun May 12 19:51:54 IST 2002")
      ("Hierarchical""yes")
      ("Implementation"
           ("MAX403ESA")
           ("MAX403EPA")
           ("MAX403CSA")
           ("MAX403CPA")
      )
)
```

Device name as appears in the .olb file.

Each of the device listed below the Implementation statement has all the entries in the device property file as any other device. The Model Editor uses the Implementation statement to access the device-specific information of the associated parts for the same model.

B

The Special Library

This chapter describes the various models in the special.olb library and their parameters, such as vector symbols, printing and plotting symbols, the watch symbol, and nodeset symbols.

VECTOR1

Properties	Description
File	If vector source is attached and no file name is assigned to any of the vector, the results are placed in a file with the name representing the simulation type and extension .vec, such as tran.vec for a transient analysis. The file is saved in the <Profile-name> folder.
Radix	The radix represents the number of bits in a Bus. For example, for a 16 bit Bus a source vector Vector16 is attached. The valid values for Radix are BINARY, HEX, and OCT. For VECTOR1, by default, the radix is BINARY. For other vector symbols the default is HEX. An output file example for BINARY will be: A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15 0s 0000000000000000 56s 0000000000000001 In HEX representation, four bits represent the values from 0 to E.An output file example for HEX will be: HEX(A0 A1 A2 A3) HEX(A4 A5 A6A7)HEX(A8 A9 A10 A11) HEX(A12 A13 A14 A15) 0s 0000 56s 0001 In the OCT representation, three bits represent values from 0 to 7. An output file example for OCT will be: OCT(A0 A1 A2) OCT(A3 A4 A5) OCT(A6 A7 A8) OCT(A9A10 A11) 0s 000 56s 001

Properties	Description
Signames	The output file is assigned the name of the net if a net name is attached. If not attached to a net name, the output file contains the NODE number. For example, if no net name is attached, the file will be: N35080 0s 0 56s 1 If the net name assigned is SS, SS 0s 0 56s 1
POS	An optional parameter that specifies the column position in the file and can have any value from 1 to 255. By default, the column position is determined by the order in which the .VECTOR command appears in the circuit file. However, users can use POS to position preferred vector source results to a specified column location in output vector file. For example, to to place the vector result in the second column, specify POS=2.

Properties	Description
BIT	Defines the bit position within a single hexadecimal or octal digit in the .vec file For example, if you specify BIT = 3, the following output is obtained: OCT(A fill0_1 fill0_0) 0s 0 100ns 4 200ns 0 300ns 4 400ns 0 500ns 4 600ns 0 700ns 4 Vector1 : Wire Vector2 : Bus with width 2 Vector4 : Bus with width 4

VPRINT1

Properties	Function
AC	Plots the magnitude of the voltage against the frequency of the node to which the symbol is attached if value is set to AC. VM (net name) is obtained in the .out file.
DB	Plots in decibel the magnitude of the voltage to which the symbol is attached if value is set to DB for AC analysis. VDB (net name) is obtained in the .out file.
DC	Plots the DC transfer curves if value is set to DC.
IMAG	Plots the imaginary part of the voltage of the node to which the symbol is attached if value is set to IMAG for AC analysis. VI (net name) is obtained in the .out file.
REAL	Plots the real value if value is set to REAL for AC analysis. VR (net name) is obtained in the .out file.
TRAN	Plots the voltage and time in a tabular format in the .out file if value is set to TRANS, for TRANSIENT type.
MAG	Plots the magnitude of the voltage of the node to which the symbol is attached if value is set to MAG for AC analysis. VM (net name) is obtained in the .out file.
PHASE	Plots the phase of the voltages if the value is set to PHASE for AC analysis. VP (Net name) is in the .out file.

VPRINT2

Properties	Function
AC	Plots the magnitude of the differential voltages between nodes to which the symbol is attached if value is set to AC. VM (net name) is obtained in the .out file.
DB	Plots in decibel the magnitude of the voltage to which the symbol is attached if value is set to DB for AC analysis. VDB (net name) is obtained in the .out file.
DC	Plots the DC transfer curves if value is set to DC.
IMAG	Plots the imaginary part of the differential voltages between the nodes to which the symbol is attached if value is set to IMAG for AC analysis. VI (net name) is obtained in the .out file.
REAL	Plots the real value or magnitude of the differential voltage between the nodes to which the symbol is attached if value is set to REAL for AC analysis. VR (net name) is obtained in the .out file.
TRAN	Plots the differential voltage and time in a tabular format in the .out file if value is set to TRANS, for TRANSIENT type.
MAG	Plots the magnitude of the differential voltage between the nodes to which the symbol is attached if value is set to MAG for AC analysis. VM (net name) is obtained in the .out file.
PHASE	Plots the phase of the differential voltages between the nodes to which the symbol is attached if the value is set to PHASE for AC analysis. VP (Net name) is in the .out file.

IPRINT1

Properties	Function
AC	Plots the magnitude of the current in the branch for AC analysis. IM (net name) is obtained in the .out file.
DB	Plots the magnitude of current in decibels in the branch to which the symbol is attached. IDB (net name) is obtained in the .out file.
DC	Plots the current and voltage in tabular format in the .out file.
IMAG	Plots the imaginary part of the current in the branch. II (net name) is obtained in the .out file.
REAL	Plots the real part of the current in the branch. IR (net name) is obtained in the .out file.
TRAN	Plots the current and time in the .out file.
MAG	Plots the magnitude of current in the branch. IM (net name) is obtained in the .out file.
PHASE	Plots the phase of current in the branch.

PRINT1

PRINT1 has the same properties as mentioned for VPRINT1 and an additional Analysis property.

The values for the Analysis property can be AC or DC, or you can leave it blank. The following table describes the result of specifying these values.

Values	Description
AC	Plots the magnitude of the voltage versus Frequency of the node to which the symbol is attached.
DC	Plots DC transfer curves.

Values	Description
Blank	Plots the voltage versus time at that nodes to which the symbol is attached. By default it will plot the values for transient type in the out file.

WATCH

Pauses the simulation for values between the higher and lower limits and plots the values for voltages only. Following are the properties:

Properties	Description
ANALYSIS	Set value to either AC or DC.
HI	Specify the higher limit. Generates error if value is not assigned.
LO	Specify the lower limit.

IC1

You can use the property Value to specify the required initial condition for calculating bias point for that node only. PSpice attaches a voltage source with a 0.0002 Ω series resistance to each net to which an IC symbol is connected.The voltages are clamped this way for the entire bias point calculation. The simulation starts from the set value of IC.

IC1 does not work with AC or DC Analysis.

IC2

You can use the property Value to specify the required initial condition for calculating bias point between two nets. The bias point is set with respect to the positive sign in the symbol and netlisted with the .IC extension.

IC2 does not work with AC or DC analysis.

NODESET1

You can use the Value property to specify the required initial condition to calculate the bias point. The model calculates the bias point by providing an initial guess for the node to which it is attached. It is effective for bias point and for the first step of the DC sweep. However, it has no effect during the rest of the DC sweep.

NODESET2

You can use the Value property to specify the required initial condition to calculate the bias point. The model calculates the bias point by providing an initial guess between the nodes to which it is attached.

Additional Symbols

The following table describes some of the useful symbols.

Symbol	Description
CD4000_PWR	CMOS power supply for CD 4000 series components.
DIGIFPWR	TTL/CMOS power supply for CMOS and TTL series.
ECL_10K_PWR	Digital ECL interface power supply. Can be used in place of separate voltage sources for ECL simulation.
ECL_100K_PWR	Digital ECL interface power supply. Can be used in place of separate voltage sources for ECL simulation.

Glossary

A

absolute sensitivity

The change in a measurement caused by a unit change in parameter value (for example, 0.1V: 1Ohm).

The formula for absolute sensitivity is:

[(Ms - Mn) / (Pn * 0.4 * Tol)]

Where:

Mn = the measurement from the nominal run

Ms = the measurement from the sensitivity run for that parameter

Tol = relative tolerance of the parameter

B

bimodal distribution function

Related to Monte Carlo. This is a type of distribution function that favors the extreme ends of the values range. With this distribution function, there is a higher probability that Monte Carlo will choose values from the far ends of the tolerance range when picking parameter values for analysis.

C

component

A circuit device, also referred to as a part.

component parameter

A physical characteristic of a component. For example, a breakdown temperature is a parameter for a resistor. A parameter value can be a number or a named value, like a programming variable that represents a numeric value. When the parameter value is a name, its numerical solution can be varied within a mathematical expression and used in optimization.

constraint

Related to Modified LSQ optimization engine. An achievable numerical value in circuit optimization. A constraint is specified by the user according to the user's design specifications. The Modified LSQ engine works to meet the goals, subject to the specified constraints.

cumulative distribution function (CDF)

A way of displaying Monte Carlo results that shows the cumulative probability that a measurement will fall within a specified range of values. The CDF graph is a stair-step chart that displays the full range of calculated measurement values on the x-axis. The y-axis displays the cumulative number of runs that were below those values.

D

derating factor

A safety factor that you can add to a manufacturer's maximum operating condition (MOC). It is usually a percentage of the manufacturer's MOC for a specific component. "No derating" is a case where the derating factor is 100 percent. "Standard derating" is a case where derating factors of various percents are applied to different components in the circuit.

device

See component

distribution function

Related to Monte Carlo. When Monte Carlo randomly varies parameter values within tolerance, it uses that parameter's distribution function to make a decision about which value to select. See also: Flat (Uniform), Gaussian (Normal), Bimodal, and Skewed distribution functions. See also cumulative distribution function.

Discrete engine

Related to the Optimizer. The Discrete engine is a calculation method that selects commercially available values for components and uses these values in a final optimization run. The engine uses default tables of information provided with Advanced Analysis or tables of values specified by the user.

discrete values table

For a single component (such as a resistor), a discrete values table is a list of commercially available numerical values for that component. Discrete values tables are available from manufacturers, and several tables are provided with Advanced Analysis.

E

error graph

A graph of the error between a measurement's goal or constraint and the calculated value for the measurement. Sometimes expressed in percent.

Error =
(Calculated meas. value - Goal value) / Goal value

Error =
(Calculated meas. value - Constraint) / Constraint

F

flat distribution function

Also known as Uniform distribution function. Related to Monte Carlo. This is the default distribution function used by Advanced Analysis Monte Carlo. For a Flat (Uniform) distribution function, the program has an equal probability of picking any value within the allowed range of tolerance values.

G

Gaussian distribution function

Also known as Normal distribution function. Related to Monte Carlo. For a Gaussian (Normal) distribution function, the program has a higher probability of choosing from a narrower range within the allowed tolerance values near the mean.

global minimum

Related to the Optimizer. The global minimum is the optimum solution, which ideally has zero error. But factors such as cost and manufacturability might make the optimum solution another local minimum with an acceptable total error.

goal

A desirable numerical value in circuit optimization. A goal may not be physically achievable, but the optimization engine tries to find answers that are as close as possible to the goal. A goal is specified by the user according to the user's design specifications.

H

I

J

K

L

local minimum

Related to the Optimizer. Local minimum is the bottom of any valley in the error in the design space.

M

Maximum Operating Conditions (MOCs)

Maximum safe operating values for component parameters in a working circuit. MOCs are defined by the component manufacturer.

Modified Least Squares Quadratic (LSQ) engine

A circuit optimization engine that results in fewer runs to reach results, and allows goal- and constraint-based optimization.

measurement expression

An expression that evaluates a characteristic of one or more waveforms. A measurement expression contains a measurement definition and an output variable. For example, Max(DB(V(load))). Users can create their own measurement expressions.

model

A mathematical characterization that emulates the behavior of a component. A model may contain parameters so the component's behavior can be adjusted during optimization or other advanced analyses.

Monte Carlo analysis

Calculations that estimate statistical circuit behavior and yield. Uses parameter tolerance data. Also referred to as yield analysis.

N

nominal value

For a component parameter, the nominal value is the original numerical value entered on the schematic.

For a measurement, the nominal value is the value calculated using original component parameter values.

normal distribution function

See Gaussian distribution function

O

optimization

An iterative process used to get as close as possible to a desired goal.

original value

See nominal value

P

parameter

See component parameter

parameterized library

A library that contains components whose behaviors can be adjusted with parameters. The Advanced Analysis libraries include components with tolerance parameters, smoke parameters, and optimizable parameters in their models.

part

See component

probability distribution function (PDF) graph

A way of displaying Monte Carlo results that shows the probability that a measurement will fall within a specified range of values. The PDF graph is a bar chart that displays the full range of calculated measurement values on the x-axis. The y-axis displays the number of runs that met those values. For example, a tall bar (bin) on the graph indicates there is a higher probability that a circuit or component will meet the x-axis values (within the range of the bar) if the circuit or component is manufactured and tested.

Q

R

Random engine

Related to Optimizer. The Random engine uses a random number generator to try different parameter value combinations then chooses the best set of parameter values in a series of runs.

relative sensitivity

Relative sensitivity is the percent change in measurement value based on a one percent positive change in parameter value for the part.

The formula for relative sensitivity is:

[(Ms - Mn) / (0.4*Tol)]

Where:

Mn = the measurement from the nominal run

Ms = the measurement from the sensitivity run for that parameter

Tol = relative tolerance of the parameter

S

Safe Operating Limits (SOLs)

Maximum safe operating values for component parameters in a working circuit with safety factors (derating factors) applied. Safety factors can be less than or greater than 100 percent of the maximum operating condition depending on the component.

sensitivity

The change in a simulation measurement produced by a standardized change in a parameter value:

$$S(\text{measurement}) = \frac{\Delta_{\text{measurement}}}{\Delta_{\text{parameter}}}$$

See also relative and absolute sensitivity.

skewed distribution function

Related to Monte Carlo. This is a type of distribution function that favors one end of the values range. With this distribution function, there is a higher probability that Monte Carlo will choose values from the skewed end of the tolerance range when picking parameter values for analysis.

Smoke analysis

A set of safe operating limit calculations. Uses component parameter maximum operating conditions (MOCs) and safety factors (derating factors) to calculate if each component parameter is operating within safe operating limits. Also referred to as stress analysis.

specification

A goal for circuit design. In Advanced Analysis, a specification refers to a measurement expression and the numerical min or max value specified or calculated for that expression.

T

U

uniform distribution function

See flat distribution function

V

W

weight

Related to Optimizer. In Optimizer, we are trying to minimize the error between the calculated measurement value and our goal. If one of our goals is more important than another, we can emphasize that importance, by artificially making that goal's error more noticeable on our error plot. If the error is artificially large, we'll be focusing on reducing that error and therefore focusing on that goal. We make the error stand out by applying a weight to the important goal. The weight is a positive integer (say, 10) that is multiplied by the goal's error, which results in a "magnified" error plot for that goal.

worst-case maximum

Related to Sensitivity. This is a maximum calculated value for a measurement based on all parameters set to their tolerance limits in the direction that will increase the measurement value.

worst-case minimum

Related to Sensitivity. This is a minimum calculated value for a measurement based on all parameters set to their tolerance limits in the direction that will decrease the measurement value.

X

Y

yield

Related to Monte Carlo. Yield is used to estimate the number of usable components or circuits produced during mass manufacturing. Yield is a percent calculation based on the number of run results that meet design specifications versus the total number of runs. For example, a yield of 99 percent indicates that of all the Monte Carlo runs, 99 percent of the measurement results fell within design specifications.

Z

Index

Symbols

101, 102
@Max 75
@Min 75
<MIN> 64

A

absolute sensitivity 349
accuracy
  and RELTOL 288
  and Threshold value 289
accuracy of simulation
  adjusting Delta value for 288
  optimum Delta value variation 288
add plot 242
adding
  measurement expressions in parametric Plotter 238
  plots in parametric Plotter 242
  sweep parameters 235
  traces in parametric Plotter 239
advanced analysis
  files 20

B

bar graph style
  linear view 76
  log view 76
bimodal distribution function 349

C

CDF graph 204
circuit preparation
  adding additional parameters 37
  creating new designs 36
  modifying existing designs 47
  selecting parameterized components 36
  setting parameter values 37
  using the design variables table 45
clear history 113
component 349
component parameter 349
configuring
  the Monte Carlo tool 200
  the Optimizer tool 97
  the Sensitivity tool 59
  the Smoke tool 163
constraint 88, 89, 350
  See Also specification
cross-hatched background 114
cumulative distribution function (CDF) 350
cursors 205
custom derating
  selecting the option 165

D

data
  sorting 61
  viewing 62
Delta option 287
derate type 329
derating factor 331, 350
derivatives 93
  calculating 287
  finite differencing 287
design variables table 45
device 350
device property files 20
dialog box
  Arguments for Measurement Evaluation 261
  Display Measurement Evaluation 264
  Measurements 261
  Traces for Measurement Arguments 262
Discrete engine 294, 351
discrete sweep 233
discrete value tables 20
discrete values table 351
DIST 27
distribution function 350
  flat 351

Gaussian 352
normal 352
skewed 355
uniform 351
distribution parameter
DIST 27

E

engine
Discrete 117, 147, 294
Modified LSQ 147, 286
Random 291
error graph 351
evaluation 91
See Also goal function, Probe
See Also PSpice Optimizer expression
See Also trace function, Probe
exponential numbers
numerical conventions 22
expression 92

F

file extensions
.aap 20
.drt 20
.prp 20
.sim 20
Find in Design 78, 118
flat distribution function 351

G

Gaussian distribution function 352
global minimum 352
goal 88, 89, 352
See Also specification
goal function, Probe
discontinuities 92
goal functions 259
goals
defining for optimization 95
graphs
cumulative distribution function 204
cursors 205
monte carlo CDF graph 221
monte carlo PDF graph 203, 218, 354
optimizer Error Graph 112, 115, 130
probability distribution function 203
sensitivity bar graph 64, 74
smoke bar graph 160, 162, 164

I

implementation statement 337
iterations, limiting in Enhanced LSQ optimization 287

K

keywords
semiconductors 169

L

least squares
constrained 87
unconstrained 87
least squares algorithm 290
Least Squares option 290
libraries
installation location 29
selecting parameterized components 36
tool tip 35
libraries used in examples
ANALOG 39
PSPICE_ELEM 48
linear bar graph style 64
linear sweep 233
local minimum 352
log bar graph style 64
Logarithmic Decade sweep 235
Logarithmic Octave sweep 234
logarithmic sweep
Decade 235
Octave 234
LTOL% 37

M

Max. Iterations option 287
maximum operating conditions (MOCs) 353

measurement
- disable 115
- editing 115, 135
- exclude from analysis 115
- expressions 257
- hiding trace on graph 115
- importing from PSpice 115
- strategy 258

measurement definition
- selecting and evaluating 259
- syntax 273
- writing a new definition 271

measurement definitions
- creating custom definitions 269

measurement expression 353
- measurement definition 259
- output variable 259
- output variables 259
- value in PSpice 260
- viewing in PSpice 260

measurement expressions
- composing 259
- creating 259
- parametric plotter 238
- PSpice Simulation Results view 259
- setup 258
- Simulation Results view 259

measurement expressions included in PSpice (list) 265

measurement results
- PSpice view menu 260

measurements
- overview 257

minimization 290
- constrained 87

minimization algorithm 290

Minimize option 290

model 353

Modified LSQ engine 353

Modified LSQ engine options 287

monte carlo
- adding a measurement 207
- allowable PSpice simulations 19
- analysis runs 201
- CDF graph 204
- controlling measurement specifications 207
- cursors 205
- distribution parameters 27
- editing a measurement 207
- editing a measurement spec min or max value 207
- example 210
- excluding a measurement from analysis 207
- overview 195
- pausing analysis 206
- pdf graph 203
- printing raw measurement data 208
- procedure 199
- raw measurements table 205
- restricting calculation range 205
- resuming analysis 206
- statistical information table 202
- stopping analysis 206
- strategy 196
- workflow 198

monte carlo results
- 3 sigma 203
- 6 sigma 203
- cursor max 203
- cursor min 203
- mean 203
- median 203
- standard deviation 203
- yield 203

monte carlo setup options
- number of bins 201
- Number of runs 203
- number of runs 200
- random seed value 201
- starting run number 200

N

negative sensitivity 63

NEGTOL 37

nominal value 353

normal distribution function 352

numerical conventions 22
- mega 23
- milli 22

O

optimization 87
- choosing least squares or minimization 289
- constrained least squares 87
- constrained minimization 87

controlling parameter
perturbation 287
for one goal 289
goals 95
limiting iterations 287
procedure overview 95
unconstrained least squares 87
optimizations
Advanced Analysis 85
PSpice 54
Optimizer 92
optimizer
adding a new measurement 115
allowable PSpice simulations 19
analysis runs 112
clearing the Error Graph history 113
constraints 101
controlling component parameters 113
controlling optimization 113
displaying run data 112
editing a measurement 115
excluding a measurement from
analysis 115
goals 101
hiding a measurement trace 115
importing measurements 101
overview 85
pausing a run 113
procedure 96
setting up component parameters 98
setting up in Advanced Analysis 98
setting up measurement
specifications 100, 101, 102
setting up specifications 100
setting up the circuit 96
starting a run 112, 113
stopping a run 113
strategy 97
weighting the goals or constraints 101
workflow 94
Optimizer expression 92
options
Delta 287
Least Squares 290
Max. Iterations 287
Minimize 290
original value 353
output variables
selecting 259

P

parameter 25, 88
parameterized components 26
parameterized library 354
Parameterized Part icon 35
parameters
controlling perturbation 287
distribution 26, 27
optimizable 26, 28
overriding global values 53
sending to Optimizer from
Sensitivity 79
setting up 95
setting values 37
smoke 26, 28, 166
tolerance 26, 27
using the schematic editor 37
Parametric Plotter
add plot 242
adding expressions 238
adding traces 239
run 240
view plot 244
viewing results 240
part 354
PDF graph 203
performance 90
positive sensitivity 63
POSTOL 37
probability distribution function (PDF)
graph 354
Probe
trace function 91
problems, common solutions to 311
project setup
validating the initial project 19
property
TOL_ON_OFF 59
property file
device 321, 333
Template 321, 323

R

Random engine 291, 354
NumRuns option 294
NumSteps option 293
options 293 to 294

Raw Measurements table 205
read-only data 114, 154, 224
Red 160
references
  auto-help 14
  related documentation 13
relative sensitivity 75, 355
RELTOL option 288
requirements, see specifications 90
restricting calculation range 205

S

safe operating limits (SOLs) 355
see also property 25
see measurements 259
Send to Optimizer 80
senitivity
  positive 63
sensitivity 355
  absolute 81
  absolute sensitivity 75
  allowable PSpice simulations 19
  analysis runs 83
  example 68
  import measurements 60
  interpreting MIN results 64
  negative 63
  overview 55
  procedure 58
  relative 82
  relative sensitivity 75
  results 61
  setting up in Advanced Analysis 59
  setting up the circuit 58
  strategy 56
  workflow 58
  worst-case maximum measurements 83
  worst-case minimum measurements 83
  zero results 64
setting up
  the Monte Carlo tool 200
  the Optimizer tool 97
  the Sensitivity tool 59
single-point analyses 91
skewed distribution function 355
smoke
  allowable PSpice simulations 19
  analysis runs 149
  changing derating options 163
  deratings 154
  example 157
  looking up parameter names 166
  overview 149
  procedure 151
  starting a run 158
  strategy 150
  viewing results 159
  workflow 151
smoke parameters 166
  op amps 174
  passive components 167
  RLCs 167
  semiconductors 169
smoke results display options
  temperature parameters only 161
  values 159
smoke setup options
  custom derating 165
  no derating 163
  standard derating 163
specification 88
  conflicting 90
  See Also constraint
  See Also goal
Standard Derating
  selecting the option 163
stress analysis
  see Smoke
sweep parameters
  add 235
sweep type 233
  discrete 233
  linear 233
  logarithmicDec 235
  logarithmicOct 234
syntax
  measurement definition comments 275
  measurement definition example 276, 282
  measurement definition marked point expressions 275
  measurement definition names 274
  measurement definition search command 276
  measurement definitions 273

T

technical note
    creating a custom derating file 155
Temperature Parameters Only 161
test node mapping 328, 335
TOL_ON_OFF 59
TOLERANCE 47
tolerance
    as percent or absolute values 27
    NEGTOL 27
    POSTOL 27
tolerance parameters
    TOLERANCE 47
trace
    parametric plotter 239
trace function, Probe 91
troubleshooting
    table of common problems 311
    using the troubleshooting tool 297

U

uniform distribution function 351
units 22

V

validating the initial project 19
VALUE 38
variables component 45
view plot 244

W

weight 356
workflow
    monte carlo 198
    optimizer 94
    overall 20
    sensitivity 58
    smoke 151
worst-case maximum 357
worst-case minimum 357

Y

yield 357
yield analysis
    see Monte Carlo