



February 2011
ADS Load Pull DesignGuide

© **Agilent Technologies, Inc. 2000-2011**

5301 Stevens Creek Blvd., Santa Clara, CA 95052 USA

No part of this documentation may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Agilent Technologies, Inc. as governed by United States and international copyright laws.

Acknowledgments

Mentor Graphics is a trademark of Mentor Graphics Corporation in the U.S. and other countries. Mentor products and processes are registered trademarks of Mentor Graphics Corporation. * Calibre is a trademark of Mentor Graphics Corporation in the US and other countries. "Microsoft®, Windows®, MS Windows®, Windows NT®, Windows 2000® and Windows Internet Explorer® are U.S. registered trademarks of Microsoft Corporation. Pentium® is a U.S. registered trademark of Intel Corporation. PostScript® and Acrobat® are trademarks of Adobe Systems Incorporated. UNIX® is a registered trademark of the Open Group. Oracle and Java and registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners. SystemC® is a registered trademark of Open SystemC Initiative, Inc. in the United States and other countries and is used with permission. MATLAB® is a U.S. registered trademark of The Math Works, Inc.. HiSIM2 source code, and all copyrights, trade secrets or other intellectual property rights in and to the source code in its entirety, is owned by Hiroshima University and STARC. FLEXIm is a trademark of Globetrotter Software, Incorporated. Layout Boolean Engine by Klaas Holwerda, v1.7 <http://www.xs4all.nl/~kholwerd/bool.html> . FreeType Project, Copyright (c) 1996-1999 by David Turner, Robert Wilhelm, and Werner Lemberg. QuestAgent search engine (c) 2000-2002, JObjects. Motif is a trademark of the Open Software Foundation. Netscape is a trademark of Netscape Communications Corporation. Netscape Portable Runtime (NSPR), Copyright (c) 1998-2003 The Mozilla Organization. A copy of the Mozilla Public License is at <http://www.mozilla.org/MPL/> . FFTW, The Fastest Fourier Transform in the West, Copyright (c) 1997-1999 Massachusetts Institute of Technology. All rights reserved.

The following third-party libraries are used by the NlogN Momentum solver:

"This program includes Metis 4.0, Copyright © 1998, Regents of the University of Minnesota", <http://www.cs.umn.edu/~metis> , METIS was written by George Karypis (karypis@cs.umn.edu).

Intel@ Math Kernel Library, <http://www.intel.com/software/products/mkl>

SuperLU_MT version 2.0 - Copyright © 2003, The Regents of the University of California, through Lawrence Berkeley National Laboratory (subject to receipt of any required approvals from U.S. Dept. of Energy). All rights reserved. SuperLU Disclaimer: THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF

SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

7-zip - 7-Zip Copyright: Copyright (C) 1999-2009 Igor Pavlov. Licenses for files are: 7z.dll: GNU LGPL + unRAR restriction, All other files: GNU LGPL. 7-zip License: This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA. unRAR copyright: The decompression engine for RAR archives was developed using source code of unRAR program. All copyrights to original unRAR code are owned by Alexander Roshal. unRAR License: The unRAR sources cannot be used to re-create the RAR compression algorithm, which is proprietary. Distribution of modified unRAR sources in separate form or as a part of other software is permitted, provided that it is clearly stated in the documentation and source comments that the code may not be used to develop a RAR (WinRAR) compatible archiver. 7-zip Availability: <http://www.7-zip.org/>

AMD Version 2.2 - AMD Notice: The AMD code was modified. Used by permission. AMD copyright: AMD Version 2.2, Copyright © 2007 by Timothy A. Davis, Patrick R. Amestoy, and Iain S. Duff. All Rights Reserved. AMD License: Your use or distribution of AMD or any modified version of AMD implies that you agree to this License. This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA Permission is hereby granted to use or copy this program under the terms of the GNU LGPL, provided that the Copyright, this License, and the Availability of the original version is retained on all copies. User documentation of any code that uses this code or any modified version of this code must cite the Copyright, this License, the Availability note, and "Used by permission." Permission to modify the code and to distribute modified code is granted, provided the Copyright, this License, and the Availability note are retained, and a notice that the code was modified is included. AMD Availability: <http://www.cise.ufl.edu/research/sparse/amd>

UMFPACK 5.0.2 - UMFPACK Notice: The UMFPACK code was modified. Used by permission. UMFPACK Copyright: UMFPACK Copyright © 1995-2006 by Timothy A. Davis. All Rights Reserved. UMFPACK License: Your use or distribution of UMFPACK or any modified version of UMFPACK implies that you agree to this License. This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License

as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA Permission is hereby granted to use or copy this program under the terms of the GNU LGPL, provided that the Copyright, this License, and the Availability of the original version is retained on all copies. User documentation of any code that uses this code or any modified version of this code must cite the Copyright, this License, the Availability note, and "Used by permission." Permission to modify the code and to distribute modified code is granted, provided the Copyright, this License, and the Availability note are retained, and a notice that the code was modified is included. UMFPACK Availability: <http://www.cise.ufl.edu/research/sparse/umfpack> UMFPACK (including versions 2.2.1 and earlier, in FORTRAN) is available at <http://www.cise.ufl.edu/research/sparse> . MA38 is available in the Harwell Subroutine Library. This version of UMFPACK includes a modified form of COLAMD Version 2.0, originally released on Jan. 31, 2000, also available at <http://www.cise.ufl.edu/research/sparse> . COLAMD V2.0 is also incorporated as a built-in function in MATLAB version 6.1, by The MathWorks, Inc. <http://www.mathworks.com> . COLAMD V1.0 appears as a column-preordering in SuperLU (SuperLU is available at <http://www.netlib.org>). UMFPACK v4.0 is a built-in routine in MATLAB 6.5. UMFPACK v4.3 is a built-in routine in MATLAB 7.1.

Qt Version 4.6.3 - Qt Notice: The Qt code was modified. Used by permission. Qt copyright: Qt Version 4.6.3, Copyright (c) 2010 by Nokia Corporation. All Rights Reserved. Qt License: Your use or distribution of Qt or any modified version of Qt implies that you agree to this License. This library is free software; you can redistribute it and/or modify it under the

terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA Permission is hereby granted to use or copy this program under the terms of the GNU LGPL, provided that the Copyright, this License, and the Availability of the original version is retained on all copies. User

documentation of any code that uses this code or any modified version of this code must cite the Copyright, this License, the Availability note, and "Used by permission." Permission to modify the code and to distribute modified code is granted, provided the Copyright, this License, and the Availability note are retained, and a notice that the code was modified is included. Qt Availability: <http://www.qtsoftware.com/downloads> Patches Applied to Qt can be found in the installation at:

\$HPEESOF_DIR/prod/licenses/thirdparty/qt/patches. You may also contact Brian Buchanan at Agilent Inc. at brian_buchanan@agilent.com for more information.

The HiSIM_HV source code, and all copyrights, trade secrets or other intellectual property rights in and to the source code, is owned by Hiroshima University and/or STARC.

Errata The ADS product may contain references to "HP" or "HPEESOF" such as in file names and directory names. The business entity formerly known as "HP EEsof" is now part of Agilent Technologies and is known as "Agilent EEsof". To avoid broken functionality and to maintain backward compatibility for our customers, we did not change all the names and labels that contain "HP" or "HPEESOF" references.

Warranty The material contained in this document is provided "as is", and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied, with regard to this documentation and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license. Portions of this product include the SystemC software licensed under Open Source terms, which are available for download at <http://systemc.org/>. This software is redistributed by Agilent. The Contributors of the SystemC software provide this software "as is" and offer no warranty of any kind, express or implied, including without limitation warranties or conditions or title and non-infringement, and implied warranties or conditions merchantability and fitness for a particular purpose. Contributors shall not be liable for any damages of any kind including without limitation direct, indirect, special, incidental and consequential damages, such as lost profits. Any provisions that differ from this disclaimer are offered by Agilent only.

Restricted Rights Legend U.S. Government Restricted Rights. Software and technical data rights granted to the federal government include only those rights customarily provided to end user customers. Agilent provides this customary commercial license in Software and technical data pursuant to FAR 12.211 (Technical Data) and 12.212 (Computer Software) and, for the Department of Defense, DFARS 252.227-7015 (Technical Data - Commercial Items) and DFARS 227.7202-3 (Rights in Commercial Computer Software or Computer Software Documentation).

Load Pull DesignGuide	7
One Tone Simulation	8
One Tone, Constant Available Source Power Load Pull	10
One Tone, Constant Available Source Power Load Pull with Swept Parameter	13
One Tone, Constant Power Delivered 2nd Harmonic Load Pull	15
One Tone, Constant Power Delivered Load Pull	17
One Tone, Constant Power Delivered Load Pull with Swept Parameter	20
One Tone, Constant Power Delivered Source Pull	24
One Tone, Swept Available Source Power Load Pull	27
Two Tone Simulation	28
Two Tone, Constant Available Source Power Load Pull	29
Two Tone, Constant Power Delivered Load Pull	31
Two Tone, Constant Power Delivered Load Pull with Swept Parameter	34
Two Tone, Swept Available Source Power Load Pull	36
WCDMA Signal Simulation	37

Load Pull DesignGuide

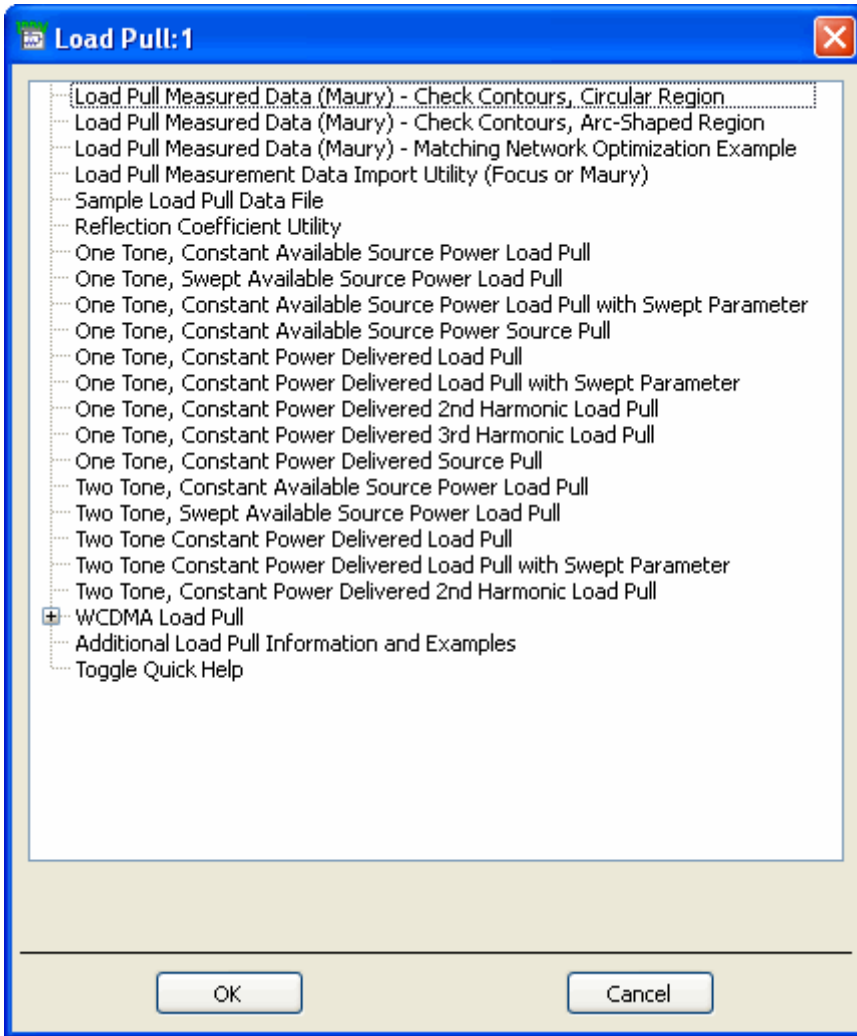
Load Pull simulation is frequently used by power amplifier designers to determine, which load impedance to present to a device or amplifier in order to achieve a particular power delivered, power-added efficiency, intermodulation distortion level, adjacent-channel leakage ratio, and other specifications.

The Load Pull DesignGuide has simulation setups for working with measured load pull data files and with nonlinear device models. The first two selections, that begin with **Load Pull Measured Data (Maury) - Check Contours...**, enable you to specify a region of the Smith Chart and verify that the measured data file will generate the contours you expect. These are useful for determining approximately the optimal load impedance. The third selection, **Load Pull Measured Data (Maury) - Matching Network Optimization...**, shows an example of optimizing an impedance matching network. The first three selections each utilize the **DataBasedLoadPull** component. This component simulates the S-parameters of the network connected to it, and uses S11 as an index into the measured load pull data file to read out (possibly with interpolation and extrapolation) the corresponding measured data.

The **Load Pull Measurement Data Import Utility (Focus or Maury)** uses the **Load Pull Utility**, which is unchanged from earlier ADS releases. This is necessary because the **DataBasedLoadPull** component does not yet work with Focus data files.

The remaining selections in this Load Pull DesignGuide are all for running various different types of load and source pulls on nonlinear device or amplifier models.

From a Schematic window, Select **DesignGuide > Load Pull** where you can see different options in the Load Pull dialog box.



When you click on one of the options in the Load Pull dialog box, for example, OneTone, Constant Available Source Power Load Pull, a simulation schematic and the corresponding data display file are copied into your working workspace directory. You modify the schematic by deleting the sample device, inserting your device, editing the parameters on the load pull instrument to set voltages as needed, and specifying the circular region of the Smith Chart for the load reflection coefficients. Then you run the simulation and view the results in the corresponding data display file. Following are the three load pull simulations:

- *One Tone Simulation* (dgldpull)
- *Two Tone Simulation* (dgldpull)
- *WCDMA Signal Simulation* (dgldpull)

The constant power delivered simulations are achieved using an optimization, and include one-tone, two-tone, and WCDMA input signals.

One Tone Simulation

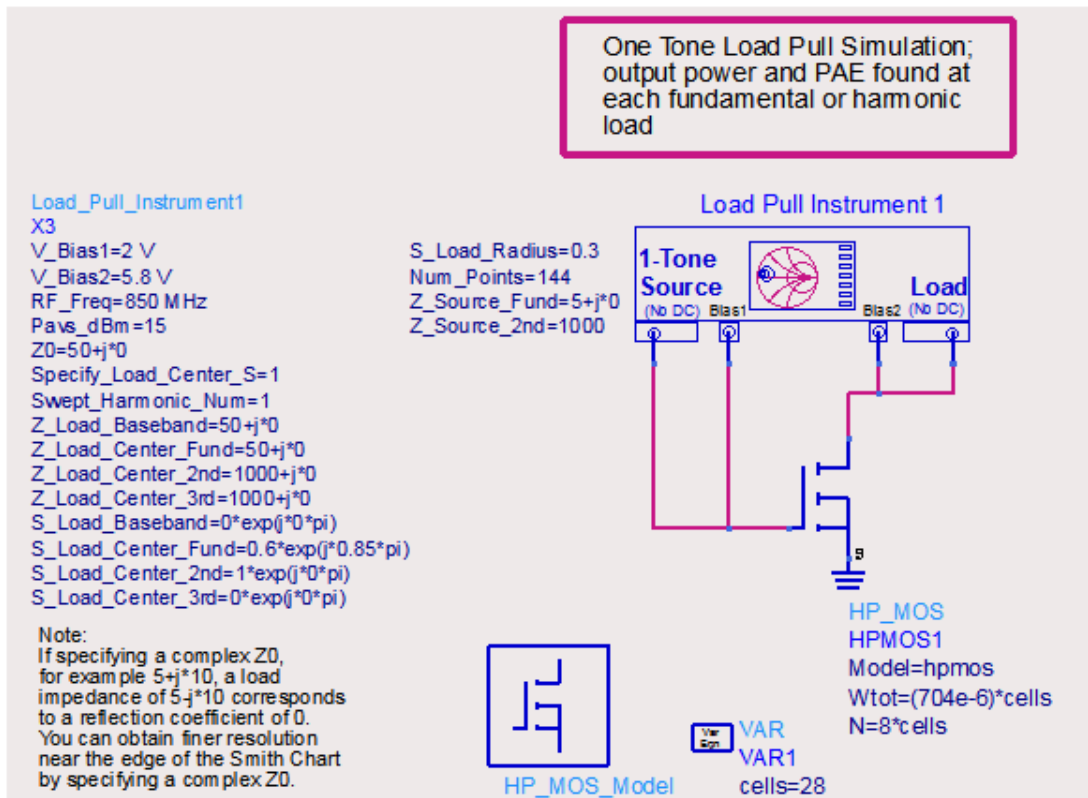
Contents

- *One Tone, Constant Available Source Power Load Pull* (dgldpull)
- *One Tone, Swept Available Source Power Load Pull* (dgldpull)

- *One Tone, Constant Available Source Power Load Pull with Swept Parameter (dgldpull)*
- *One Tone, Constant Available Source Power Source Pull (dgldpull)*
- *One Tone, Constant Power Delivered Load Pull (dgldpull)*
- *One Tone, Constant Power Delivered Load Pull with Swept Parameter (dgldpull)*
- *One Tone, Constant Power Delivered 2nd Harmonic Load Pull (dgldpull)*
- *One Tone, Constant Power Delivered Source Pull (dgldpull)*

One Tone, Constant Available Source Power Load Pull

This setup simulates the device or amplifier with a constant available source power while the load reflection coefficient is swept in a circular region of the Smith Chart. This simulation is the fastest, simplest one. It just runs a load pull simulation at a single available source power level. This is useful for seeing the power, gain, and PAE contours of a particular device at a single input power level. It does not provide information about how far into compression the device is. However, you can quickly change the settings in the schematic and see if you are getting the performance you are expecting.



You need to specify multiple things while using this schematic, such as:

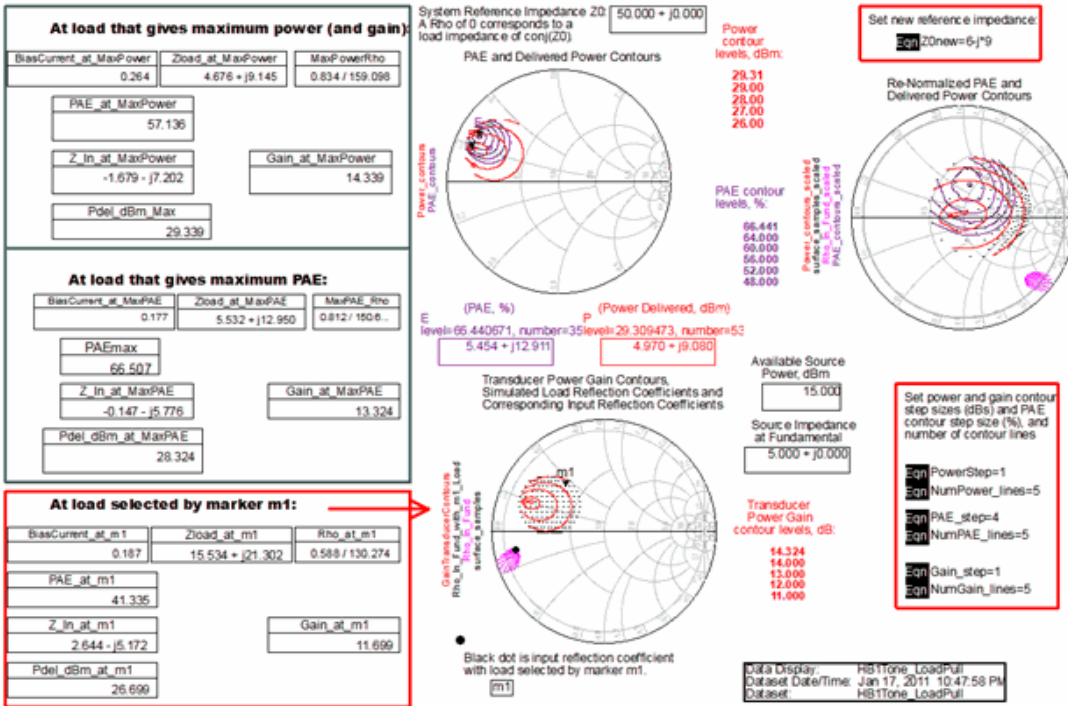
1. Replace the device with your device or amplifier.
2. Set the bias voltages V_{Bias1} and V_{Bias2} or modify the bias network in the Load_Pull_Instrument1 subcircuit, as needed. However, the DC power consumption is computed in an equation within the subcircuit, assuming current probe Is_1 is connected to supply voltage node Vs_1 and current probe Is_2 is connected to supply voltage node Vs_2 . If you delete any of these or re-name them, you must update the Pdc equation so the DC power consumption is computed correctly.
3. Specify the available source power $Pavs_dBm$ and input frequency RF_Freq , source impedances at the fundamental $Z_Source_Fund_$, and 2nd harmonic Z_Source_2nd frequencies.
4. There are three main ways to specify the region of the Smith Chart that will be sampled by the load pull:

1. Set the reference impedance Z_0 to 50, and set `Specify_Load_Center_S=1`. In this case, the center of the circle of simulated reflection coefficients (and harmonics) will be set by the `S_Load_Center_*` parameters, which are reflection coefficients.
2. Set the reference impedance Z_0 to 50, and set `Specify_Load_Center_S=0`. In this case, the center of the circle of simulated reflection coefficients (and harmonics) will be set by the `Z_Load_Center_*` parameters, which are impedances.
3. Set the reference impedance Z_0 to the complex conjugate of the impedance at the center of the circle of reflection coefficients you want to simulate. In this case, you could set `Specify_Load_Center_S=1` and use the `S_Load_Center_*` parameters. Remember that in this case, a reflection coefficient of 0 corresponds to setting the load impedance to the complex conjugate of the reference impedance.
5. Specify the radius of the circle of the reflection coefficients `S_Load_Radius` and the number of points `Num_Points`.

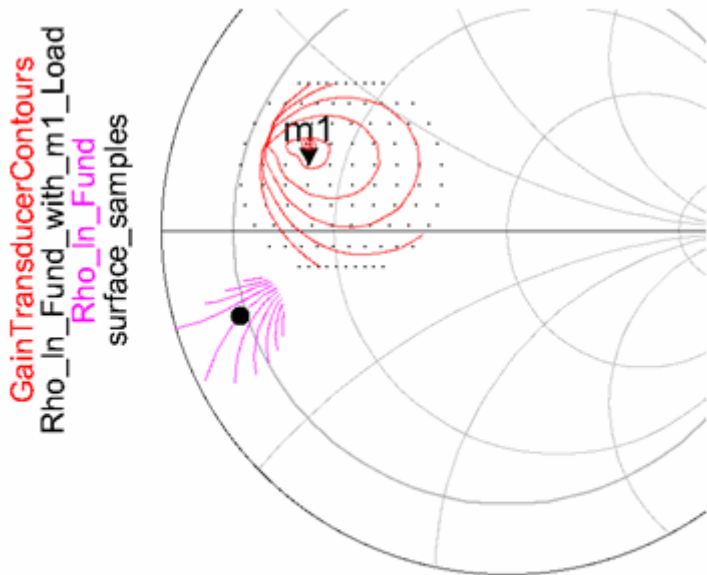
If the device or amplifier is potentially unstable and the circle of reflection coefficients that you specify includes the unstable region, the simulation may run into convergence problems. This would be due to the device wanting to oscillate. A solution to this problem is to add stabilizing components at the input, output, or in parallel with the device. You may want to use a simulation setup for this purpose, **DesignGuide > Amplifier > S-Parameter Simulations > Feedback Network Optimization** to Attain Stability. Another solution is to specify the circle of reflection coefficients such that the unstable region is avoided.

You can select **DesignGuide > Load Pull > Reflection Coefficient Utility** to see a data display with a graphical tool to help you see the circle that corresponds to particular values of the `s11_rho` and `s11_center` variables.

Run the simulation just as you would any other. When it finishes, open the `HB1Tone_LoadPull` data display.



This shows the contours of power delivered, PAE, and gain. You may want to change the step sizes between and the number of contour lines. The boxes on the left show that you get slightly different performance results depending on whether you choose the load to maximize power delivered or PAE. The lower Smith Chart shows how the input reflection coefficients vary as a function of the load reflection coefficients.

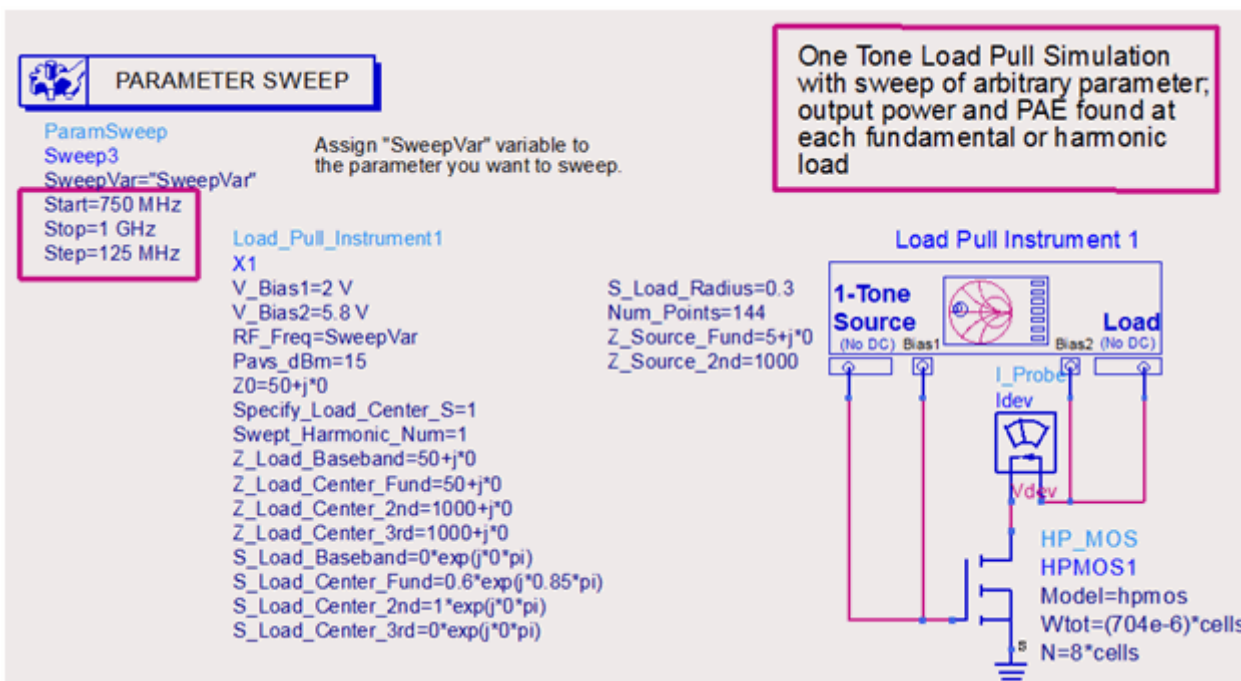


- Black dot is input reflection coefficient with load selected by marker m1.

One Tone, Constant Available Source Power Load Pull with Swept Parameter

The **One Tone, Constant Available Source Power Loadpull with Swept Parameter** simulation allows you to see how the contours and performances vary with some arbitrary swept parameter (such as a bias voltage and the input frequency). However, the available source power is held constant.

This setup adds the sweep of an arbitrary parameter to the simplest constant available source power load pull simulation. You have to assign the parameter *SweepVar* to some component or variable on the schematic. In this case it is assigned to the input signal frequency *RF_Freq*.



The above figure specifies the limits of the swept variable, *SweepVar*, and assigning it to *RF_Freq*.

One Tone, Constant Available Source Power Source Pull

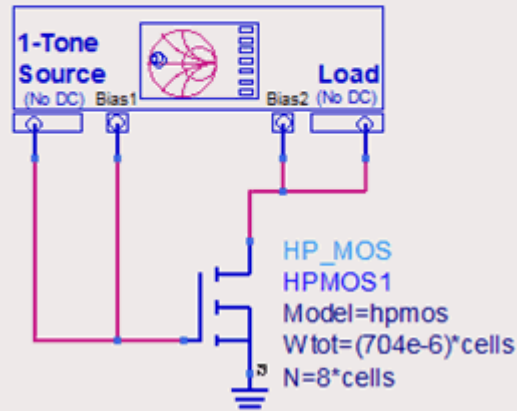
This setup simulates the device or amplifier with a constant available source power while the load reflection coefficient is swept in a circular region of the Smith Chart. This simulation is the fastest, simplest one. It just runs a load pull simulation at a single available source power level. This is useful for seeing the power, gain, and PAE contours of a particular device at a single input power level. It does not provide information about how far into compression the device is. However, you can quickly change the settings in the schematic and see if you are getting the performance you are expecting.

One Tone Source Pull Simulation;
output power and PAE found at
each fundamental or harmonic
source impedance

Source Pull Instrument 1

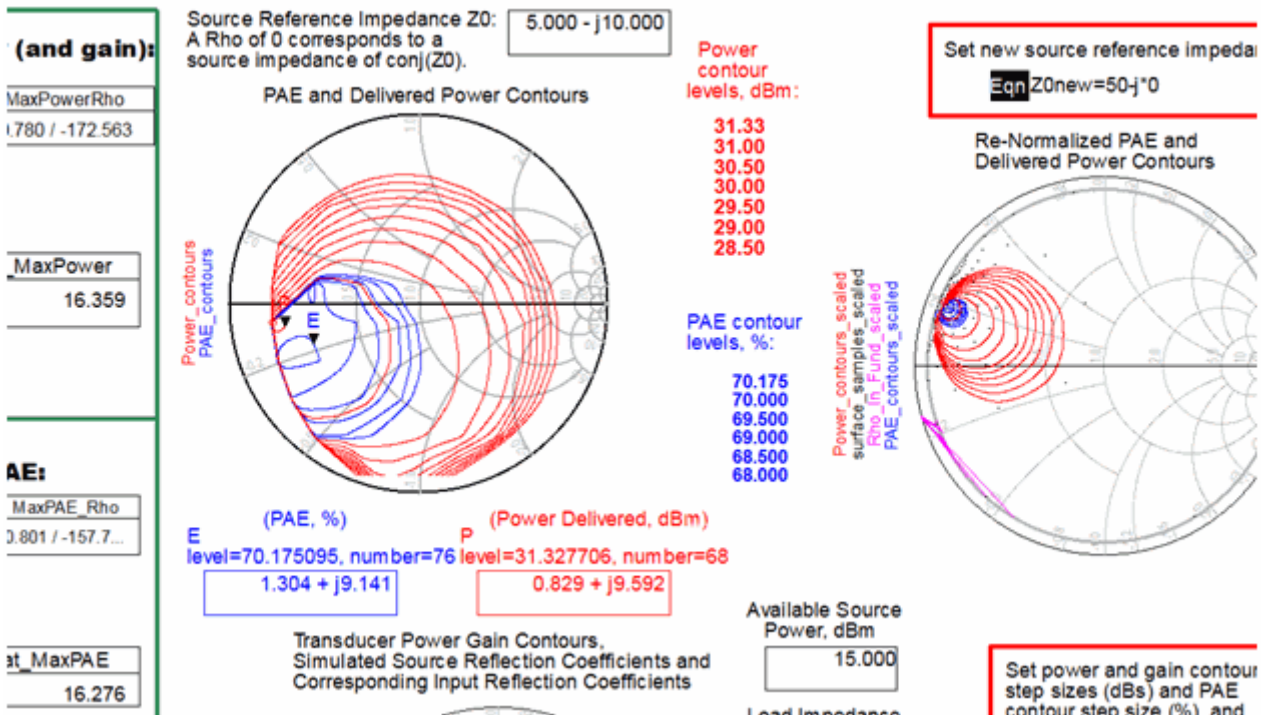
X1
 V_Bias1=2 V
 V_Bias2=5.8 V
 RF_Freq=850 MHz
 Pavs_dBm=15
 Z0_Src=5-j*10
 Specify_Src_Center_S=0
 Swept_Harmonic_Num=1
 Z_Src_Baseband=50+j*0
 Z_Src_Center_Fund=5+j*10
 Z_Src_Center_2nd=50+j*0
 Z_Src_Center_3rd=50+j*0
 S_Src_Baseband=0*exp(j*0*pi)
 S_Src_Center_Fund=0*exp(j*0*pi)
 S_Src_Center_2nd=0*exp(j*0*pi)
 S_Src_Center_3rd=0*exp(j*0*pi)
 S_Src_Radius=1
 Num_Points=100
 Z0_Load=50+j*0
 Specify_Load_S=0
 Z_Load_Baseband=50+j*0
 Z_Load_Fund=4.7+j*9.1
 Z_Load_2nd=50+j*0
 Z_Load_3rd=50+j*0
 S_Load_Baseband=0*exp(j*0*pi)
 S_Load_Fund=0*exp(j*0*pi)
 S_Load_2nd=0*exp(j*0*pi)
 S_Load_3rd=0*exp(j*0*pi)

Source Pull Instrument 1



Because the optimal source impedance is usually close to the edge of the Smith Chart, in this case we have set the reference impedance for the source $Z0_Src$ to a value that is near the complex conjugate of what we expect will be the optimum.

The left Smith Chart below shows much better resolution of the contour lines because its center corresponds to $5+j*10$ Ohms.



One Tone, Constant Power Delivered 2nd Harmonic Load Pull

The **One Tone, Constant Power Delivered 2nd Harmonic Loadpull (HB1Tone_2ndHarmLoadPull_ConstPdel)** simulation runs a load pull in which the load at the fundamental frequency is held constant and the load at the 2nd harmonic is allowed to vary. As above, the available source power is optimized until the desired power is delivered to the load. The second harmonic may only have a significant effect on the performances of devices being driven well into compression. This simulation is identical to other load pull simulations, except that the load impedance at the second harmonic frequency is swept.

You must start this simulation using Simulate > Optimize, or click the Optimize icon.

One Tone Load Pull Simulation; the available source power is optimized such that the power delivered to the load is within the limits you specify.

```

Load_Pull_Inst_Const_Pdel1
X1
V_Bias1=1.5 V
V_Bias2=5.8 V
RF_Freq=850 MHz
Pavs_dBm_min=-10
Pavs_dBm_max=20
Pdel_dBm_goal_min=25.0
Pdel_dBm_goal_max=25.1
Z0=50+j*0
Specify_Load_Center_S=1
Swept_Harmonic_Num=2
S_Load_Baseband=0*exp(j*0*pi)
S_Load_Center_Fund=0.9*exp(j*146*pi/180)
S_Load_Center_2nd=0*exp(j*0*pi)
S_Load_Center_3rd=1*exp(j*0*pi)
S_Load_Radius=1
          
```

Load Pull Instrument,
Constant Power Delivered

Note: This is exactly the same as the HB1Tone_LoadPull_ConstPdel schematic except for different parameter values on the Load_Pull_Inst_Const_Pdel1 instrument. Therefore, if you started with the HB1Tone_LoadPull_ConstPdel schematic, you could continue using that schematic and just change the parameters on the instrument to sweep the 2nd or 3rd harmonic loads.

This schematic is identical to the corresponding one for a load pull at the fundamental except that now the `Swept_Harmonic_Num` variable is set to 2 instead of 1. (In fact, you could just use the schematic set up for the fundamental sweep, instead of this one.) Since we are sweeping the impedance at the second harmonic frequency, the load at the fundamental frequency will be fixed at `S_Load_Center_Fund` or `Z_Load_Center_Fund`. The further into compression the device is being driven or the more non-linearly it is operating, the larger the effect of the second harmonic impedance.

The variation in performance due to the 2nd harmonic impedance may be small.

At load that gives minimum bias current:

MinBiasCurrent	Zload_at_MinCurrent	MinCurrentRho
0.080	-8.276E-15 + j144.2	1.000 / 38.247
PAE_at_MinCurrent	GainComp_at_MinCurrent	
70.014	-5.833	
Z_in_at_MinCurrent	Gain_at_MinCurrent	
-0.498 - j7.097	8.277	
Pdel_dBm_at_MinCurrent		
25.002		

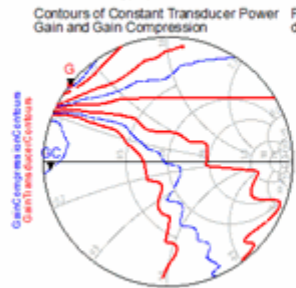
At load that gives maximum PAE:

BiasCurrent_at_MaxPAE	Zload_at_MaxPAE	MaxPAE_Rho
0.081	0.000 + j73.831	1.000 / 86.213
PAE_max	GainComp_at_MaxPAE	
70.391	-6.091	
Z_in_at_MaxPAE	Gain_at_MaxPAE	
-0.666 - j7.050	8.534	
_el_dBm_at_MaxPAE		
25.054		

At load that gives minimum gain compression:

BiasCurrent_at_MinGainComp	Zload_at_MinGainComp	MinGainCompRho
0.098	-8.950E-16 + j17.337	1.000 / 141.753
PAE_at_MinGainComp	GainCompMin	
61.431	-8.307	
Z_in_at_MinGainComp	Gain_at_MinGainComp	
-2.864 - j7.100	10.751	
Pdel_dBm_at_MinGainComp		
25.067		

System Reference Impedance Z0: 50.000 - j0.000
 A Rho of 0 corresponds to a load impedance of cony(Z0)



For all of these contours, the power delivered to the load was approximately:
 Pdelivered_dBm_mean: 25.009

Maximum Transducer Power Gain and contour levels, dB:
 10.740
 10.000
 9.000
 8.000
 7.000
 6.000

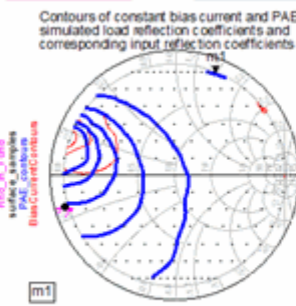
Minimum Gain Compression and contour levels, dB:
 -8.000
 -6.000
 -4.000
 -2.000

Set bias current contour step size and PAE contour step size (%), and number of contour lines

Eqn CurrentStep=20m
 Eqn NumCurrent_lines=5
 Eqn PAE_step=5
 Eqn NumPAE_lines=5
 Eqn Gain_step=1
 Eqn NumGain_lines=5
 Eqn GainCompStep=2
 Eqn NumGainComp_lines=5

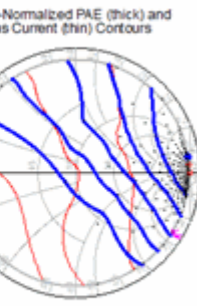
G (Gain) level: 10.740024, number=9
 0.003 + j17.301

GC (Gain Compression) level: -2.000000, number=1
 1.329 - j1.644



Maximum PAE and contour levels, %:
 70.321
 65.000
 60.000
 55.000
 50.000
 45.000

Set new reference impedance:
 Eqn Z0new=5j*10



Rho_in_Tund with m1 Load surface is on PAE contours
 BiasCurrentContours
 m1

Minimum Bias Current and contour levels, Amps:
 79.62 m
 100.0 m
 120.0 m
 140.0 m

surface samples scaled
 Rho_in_Tund scaled
 BiasCurrentContours_scaled

Black dot is input reflection coefficient with load selected by marker m1.

One Tone, Constant Power Delivered Load Pull

The **One Tone, Constant Power Delivered Loadpull (HB1Tone_LoadPull_ConstPdel)** simulation optimizes the available source power level until the desired power is delivered to each load reflection coefficient. You would use this if your device or amplifier needs to deliver a particular power level and you want to choose the optimum load considering other performances (such as gain, gain compression, PAE, and bias current.)

This setup sweeps the load reflection coefficient in a circular region of the Smith Chart and optimizes the source power level for each load reflection coefficient until the desired power is delivered to the load. The data display shows contours of constant PAE, bias current, gain, and gain compression. The input reflection coefficient is also shown for a particular load that you specify. These data allow you to pick the optimal load that produces the best PAE, gain, gain compression, or bias current, or make trade-offs amongst these specifications.

You must start this simulation using Simulate > Optimize, or click the Optimize icon.

One Tone Load Pull Simulation; the available source power is optimized such that the power delivered to each load is within the limits you specify.


```

Load_Pull_Inst_Const_Pdel1
X1
V_Bias1=2V
V_Bias2=5.8V
RF_Freq=850 MHz
Pavs_dBm_min=-10
Pavs_dBm_max=20
Pdel_dBm_goal_min=25.0
Pdel_dBm_goal_max=25.1
Z0=50+j*0
Specify_Load_Center_S=1
Swept_Harmonic_Num=1
S_Load_Baseband=0*exp(j*0*pi)
S_Load_Center_Fund=0.55*exp(j*0.85*pi)
S_Load_Center_2nd=1*exp(j*0*pi)
S_Load_Center_3rd=1*exp(j*0*pi)
S_Load_Radius=0.4
          
```

Num_Points=169
Z_Source_Fund=5+j*0
Z_Source_2nd=1000

Load Pull Instrument,
Constant Power Delivered

1-Tone Source (No DC) Bias1 Load Bias2 (No DC)

I_Probe
Idev

Vdev

HP_MOS
HPMOS1
Model=hpmos
Wtot=(704e-6)*cells
N=8*cells


You have to make the same types of edits to this schematic as with the others described above. Also, you have to specify the minimum and maximum allowed values of the available source power, Pavs_dBm_min and Pavs_dBm_max, respectively.

During the optimization, the available source power is adjusted within these limits until the power that you want is delivered to the load. Depending on how high a power you want delivered to the load and the gain of the device, you may have to adjust the Pavs_dBm_max limit.

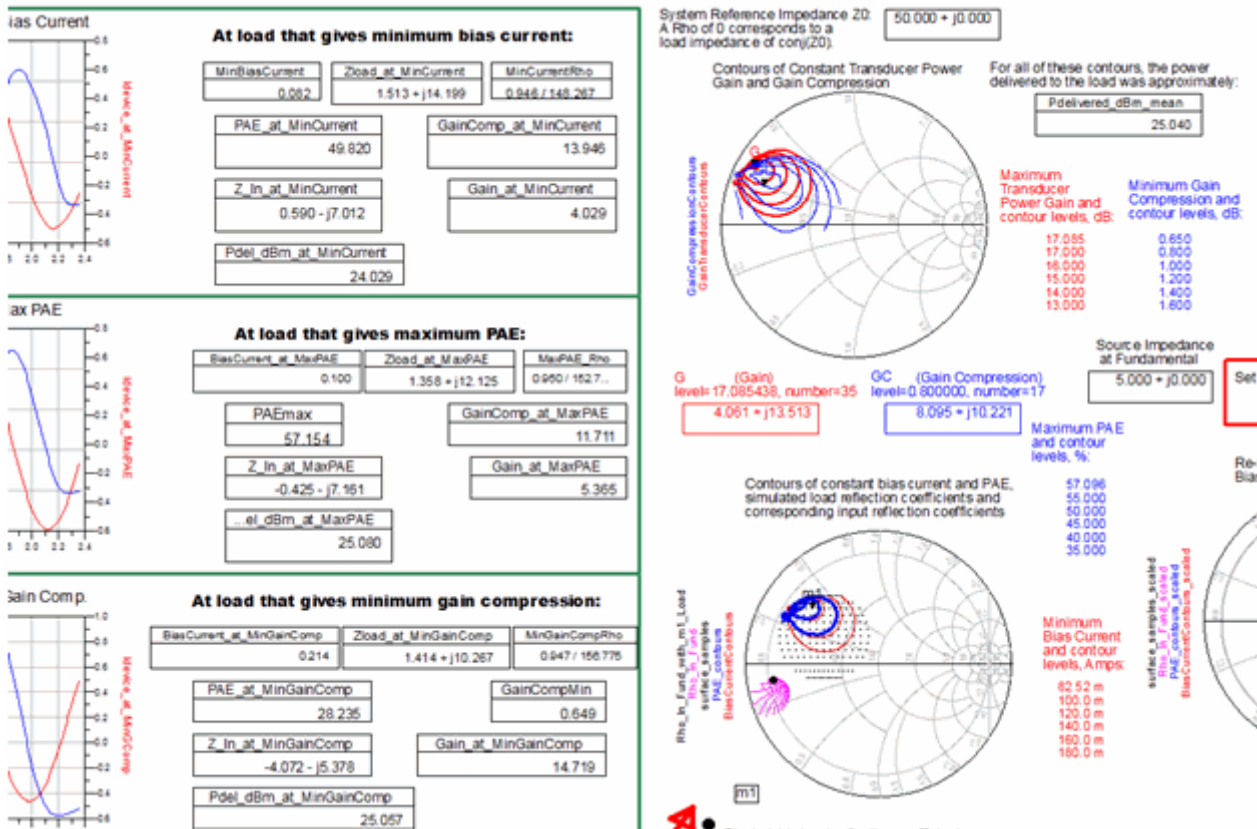
In this example, the power delivered (based on Pdel_dBm_goal_min and

Pdel_dBm_goal_max) is to be between 25 and 25.1 dBm. With this value and a Pavs_dBm_max value of 20 dBm, it is specified that the lowest transducer power gain accepted is 5 dB.

You may also specify different load impedances or reflection coefficients at the harmonic frequencies and (for the source) at the fundamental and harmonic frequencies.

To launch the simulation, click the *Optimize* icon . If the simulation is started by hitting the F7 key or by selecting **Simulate > Simulate**, then an optimization is not executed and the simulation results are not displayed correctly in the data display.

After running the optimization, the **HB1Tone_LoadPull_ConstPdel** data display shows the results.



To see the contours effectively, you may need to change the CurrentStep, PAE_step, Gain_step, and GainCompStep variables. These set the step sizes between the contours.

The upper Smith Chart shows contours of constant gain and gain compression. The lower left Smith Chart shows contours of constant bias current and power-added efficiency (PAE), as well as the simulated load reflection coefficients and the corresponding input reflection coefficients.

In the green boxes on the left side are data that correspond to a particular optimal condition such as minimum bias current, maximum PAE, or minimum gain compression. However, you have to make sure that the desired power delivered was actually achieved. For some load impedances close to the edge of the Smith Chart this may be difficult.

You also have the option of selecting any of the simulated load reflection coefficients with marker m1. The corresponding data appears in a separate box.

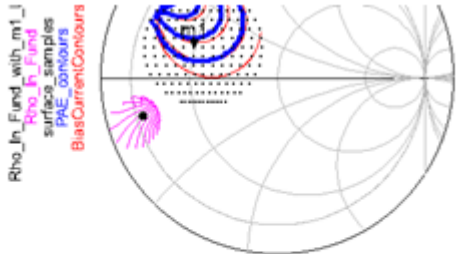
At load that gives minimum gain compression:

BiasCurrent_at_MinGainComp	Zload_at_MinGainComp	MinGainCompRho
0.142	6.630 + j13.024	0.779 / 150.333

PAE_at_MinGainComp	GainCompMin
39.043	0.752

Z_In_at_MinGainComp	Gain_at_MinGainComp
-0.440 - j5.152	16.663

Pdel_dBm_at_MinGainComp
25.048



Rho_In_Fund_with_m1
rho_Fund
surface_samples
PAE_contours
BiasCurrentContours

● Black dot is input reflection coefficient with load selected by marker m1.

m1
impedance = 12.806 + j8.018

M
B
r
1

At load selected by marker m1:

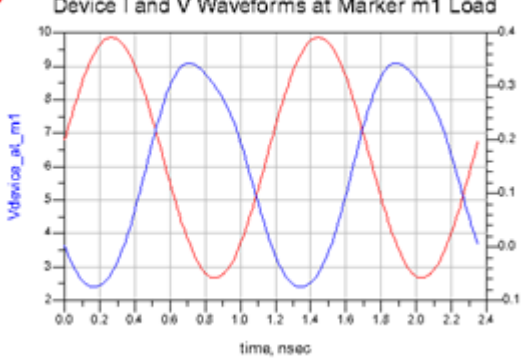
BiasCurrent_at_m1	Zload_at_m1	Rho_at_m1
0.166	12.806 + j8.018	0.601 / 160.560

PAE_at_m1	GainComp_at_m1
33.147	0.912

Z_In_at_m1	Gain_at_m1
1.398 - j7.182	14.410

Pdel_dBm_at_m1
25.093

Device I and V Waveforms at Marker m1 Load



Idevice_at_m1

Vdevice_at_m1

Data within the red box corresponds to the reflection coefficient selected by marker M1.

One Tone, Constant Power Delivered Load Pull with Swept Parameter

The **One Tone, Constant Power Delivered Loadpull with Swept Parameter (HB1Tone_LoadPull_ConstPdel_Sweep)** simulation adds the sweep of an arbitrary parameter. This sort of simulation is very useful if you want to see how the optimal load impedance, load pull contours, and the device performances vary versus some arbitrary parameter. For example, how does the optimal load vary with frequency and how do the power added efficiency and other parameters change with the bias voltage. If you have included some stabilization network around the device, how does the performance vary as you change one of the parameters.

This simulation setup and data display are nearly identical to the version without the parameter sweep.

PARAMETER SWEEP

ParamSweep
Sweep3
SweepVar=" SweepVar"
Start=1.5
Stop=2.25
Step=0.25

Assign "SweepVar" variable to the parameter you want to sweep.

One Tone Load Pull Simulation; the available source power is optimized such that the power delivered to each reflection coefficient is within the limits you specify.

Load Pull Instrument, Constant Power Delivered

1-Tone Source (No DC) Bias1
Load (No DC) Bias2

Num_Points=169
Z_Source_Fund=5+j0
Z_Source_2nd=1000

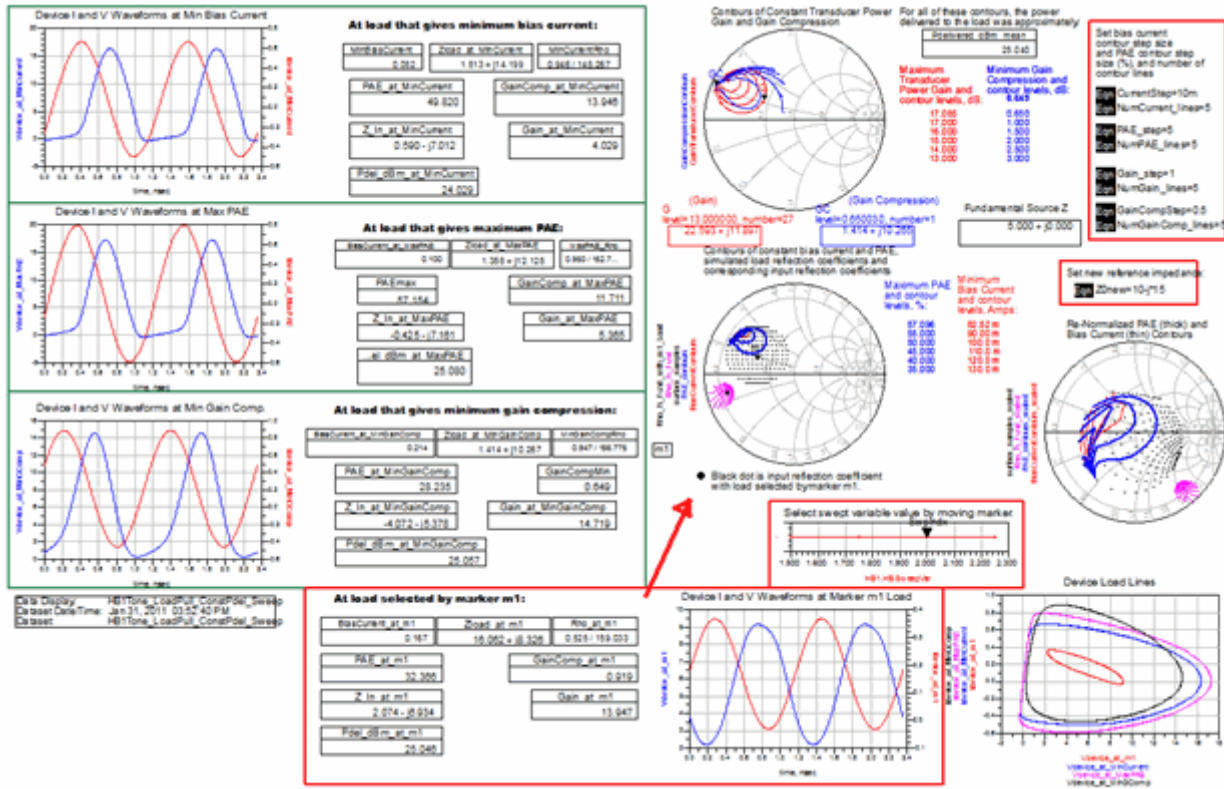
You must start this simulation using Simulate > Optimize, or click the Optimize icon.

HP_MOS Model
hpmos
VAR
VAR1
cells=28

HP_MOS
HPMOS1
Model=hpmos
Wtot=(704e-6)*cells
N=8*cells

Load_Pull_Inst_Const_Pdel1
X1
V_Bias1=SweepVar
V_Bias2=5.8 V
RF_Freq=850 MHz
Pavs_dBm_min=-10
Pavs_dBm_max=20
Pdel_dBm_goal_min=25.0
Pdel_dBm_goal_max=25.1
Z0=50+j0
Specify_Load_Center_S=1
Swept_Harmonic_Num=1
S_Load_Baseband=0*exp(j*0*pi)
S_Load_Center_Fund=0.55*exp(j*0.85*pi)
S_Load_Center_2nd=1*exp(j*0*pi)
S_Load_Center_3rd=1*exp(j*0*pi)
S_Load_Radius=0.4

You have to assign the swept variable *SweepVar* to some parameter on the schematic as well as set its sweep limits. In this case, the gate bias voltage V_{Bias1} will be swept from 1.5 to 2.25 Volts. After running the optimization, open the **HB1Tone_LoadPull_ConstPdel_Sweep** data display and make sure the default dataset name is set to the name of the dataset the simulation just created.



All the data is now indexed to the swept parameter value, which you select by moving the *SweepIndx* marker. There is a change in the power-added efficiency as you change the bias voltage.

At load that gives minimum bias current:

MinBiasCurrent	Zload_at_MinCurrent	MinCurrentRho
0.102	1.513 + j14.199	0.946 / 148.267
PAE_at_MinCurrent	GainComp_at_MinCurrent	
41.647	14.232	
Z_In_at_MinCurrent	Gain_at_MinCurrent	
0.744 - j6.631	4.227	
Pdel_dBm_at_MinCurrent		
24.227		

At load that gives maximum PAE:

BiasCurrent_at_MaxPAE	Zload_at_MaxPAE	MaxPAE_Rho
0.118	1.358 + j12.125	0.950 / 152.7...
PAEmax	GainComp_at_MaxPAE	
47.688	10.059	
Z_In_at_MaxPAE	Gain_at_MaxPAE	
-0.523 - j5.283	8.856	
...el_dBm_at_MaxPAE		
25.009		

The above figure displays the data with swept variable (gate bias) set to 2.25 Volts.

At load that gives minimum bias current:

MinBiasCurrent	Zload_at_MinCurrent	MinCurrentRho
0.061	2.008 + j16.682	0.930 / 143.049
PAE_at_MinCurrent	GainComp_at_MinCurrent	
51.810	0.627	
Z_In_at_MinCurrent	Gain_at_MinCurrent	
1.132 - j7.720	3.178	
Pdel_dBm_at_MinCurrent		
23.178		

At load that gives maximum PAE:

BiasCurrent_at_MaxPAE	Zload_at_MaxPAE	MaxPAE_Rho
0.081	4.200 + j15.860	0.858 / 144.5...
PAEmax	GainComp_at_MaxPAE	
69.881	-6.744	
Z_In_at_MaxPAE	Gain_at_MaxPAE	
-0.331 - j7.110	9.868	
...el_dBm_at_MaxPAE		
25.098		

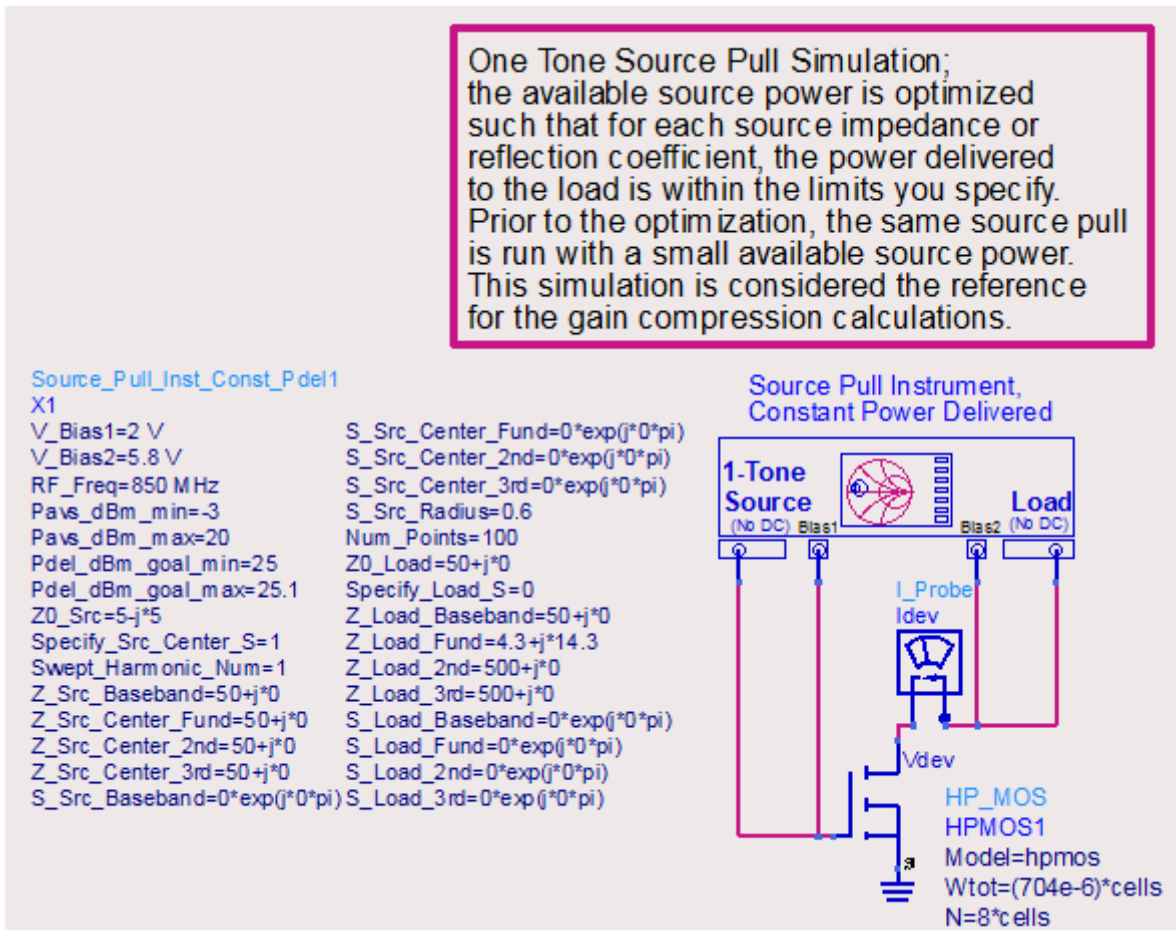
The above figure displays the data with swept variable (gate bias) set to 1.5 Volts.

When the *GainComp* values are negative, it means gain expansion is occurring. This could be because the device is biased off or at a low bias current and that this bias point shifts when a large enough signal is applied.


One Tone, Constant Power Delivered Source Pull

The **One Tone, Constant Power Delivered Source Pull (HB1Tone_SourcePull_ConstPdel)** simulation is most useful for determining the optimal source impedance to present to a device. The source impedance should mostly affect the gain and gain compression. This setup sweeps the source reflection coefficient in a circular region of the Smith Chart and optimizes the source power level for each source reflection coefficient until the desired power is delivered to the load. The data display shows contours of constant PAE, gain, and gain compression. This allows you to pick the optimal source that produces the best PAE, gain, or gain compression, or make trade-offs amongst these specifications.

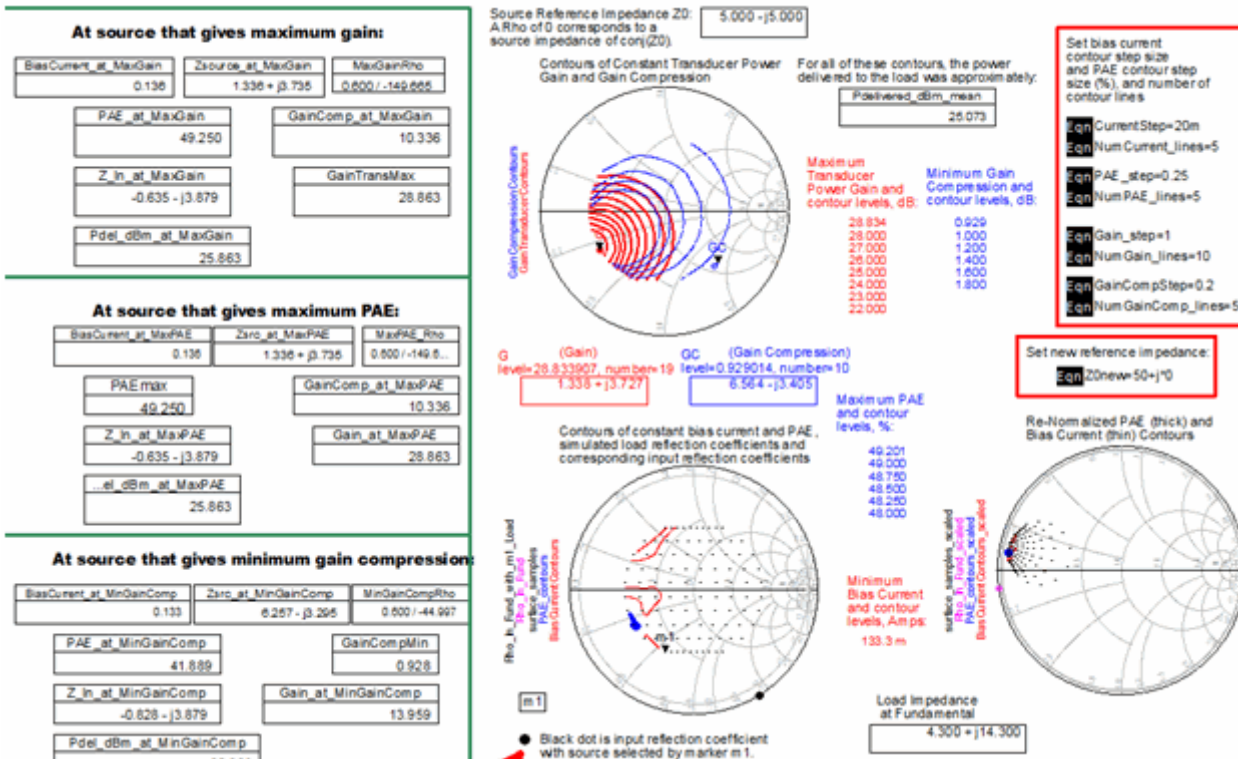
The lower left *SmallSignal* and *Sweep3* parameter sweeps are only used to obtain output powers when the device is being driven with a small signal. These output powers are used as references in the gain compression computations.



Relative to the other constant power delivered simulation schematics above, the only difference is that you have to specify the constant load impedance at the fundamental frequency, instead of the source impedance. You may also specify different load and source impedances at the harmonic frequencies. Typically, you would first run a load pull simulation to determine the fundamental load impedance to use here.

To launch the simulation, click the *Optimize*  icon. If the simulation is started by hitting the F7 key or by selecting **Simulate > Simulate**, then an optimization is not executed and the simulation results are not displayed correctly in the data display.

After running the optimization, the **HB1Tone_SourcePull_ConstPdel** data display shows the results.



To see the contours effectively, you may need to change the **PAE_step**, **Gain_step**, and **GainCompStep** variables. These set the step sizes between the contours. As stated above, if you have modified the bias network, you will have to edit the *Pdc* equation on the **Equations** page.

Also, the bias supply current calculations only include the current in the probe *Is_high*. If you change the name of the current probe, you will need to edit the *BiasCurrent* equation on the **Equations** page.

The upper Smith Chart shows contours of constant gain and gain compression. The lower left Smith Chart shows contours of constant power-added efficiency (PAE) which may not vary much, as well as the simulated source reflection coefficients and the corresponding input reflection coefficients, which may be just a single point since they should not depend on the source impedance. The lower right Smith Chart shows the same data on a Smith Chart with a different reference impedance.

In the boxes on the left side are data that correspond to a particular optimal condition such as maximum gain, maximum PAE, or minimum gain compression. However, you have to make sure that the desired power delivered was actually achieved.

The source that corresponds to the maximum gain is very nearly satisfying the conditions for oscillation at the input ($Z_{In_at_MaxGain} + Z_{source_at_MaxGain} = 0$, approximately), so you would want to avoid setting the source impedance to this value or you would want to add some sort of stabilization network around the device.

You also have the option of selecting any of the simulated source reflection coefficients with marker m1. The corresponding data appear in a separate box.

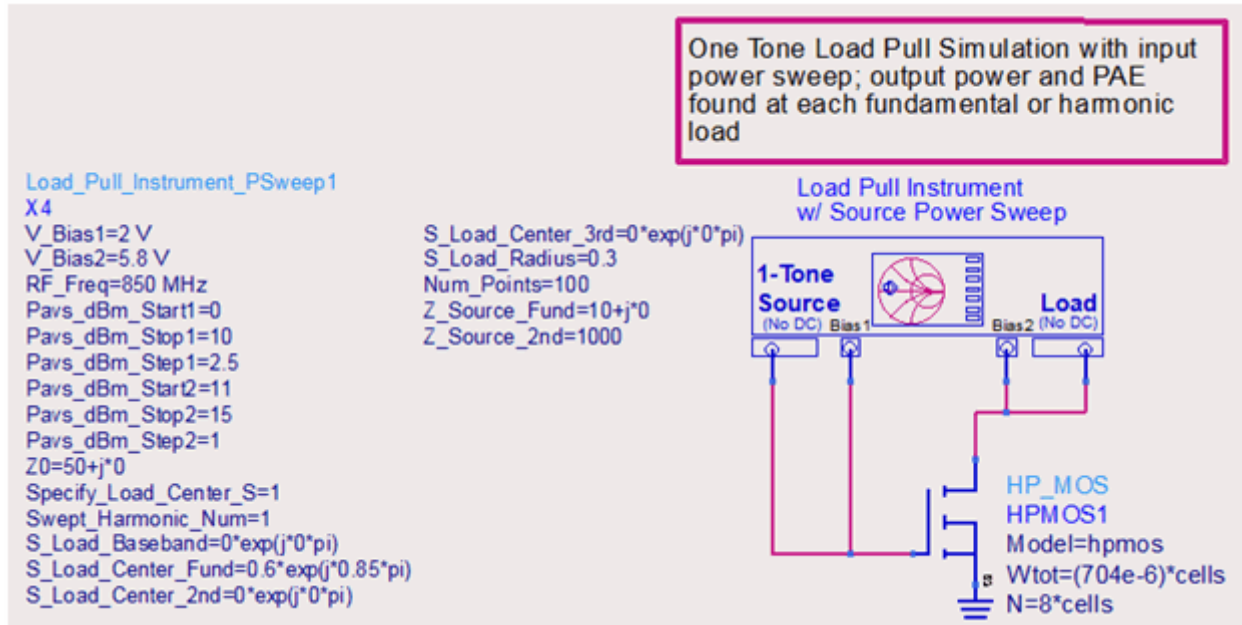
At source selected by marker m1:

BiasCurrent_at_m1	Zsource_at_m1	Rho_at_m1
0.134	2.835 + j7.012	0.895 / 163.983
PAE_at_m1	GainComp_at_m1	
41.951	1.469	
Z_In_at_m1	Gain_at_m1	
-0.848 - j3.883	18.160	
Pdel_dBm_at_m1		
25.088		

The above figure shows the performance data corresponding to the source impedance selected by marker 1. This enables you to see potential trade-offs. As you move away from the maximum gain source impedance, the gain drops rapidly.

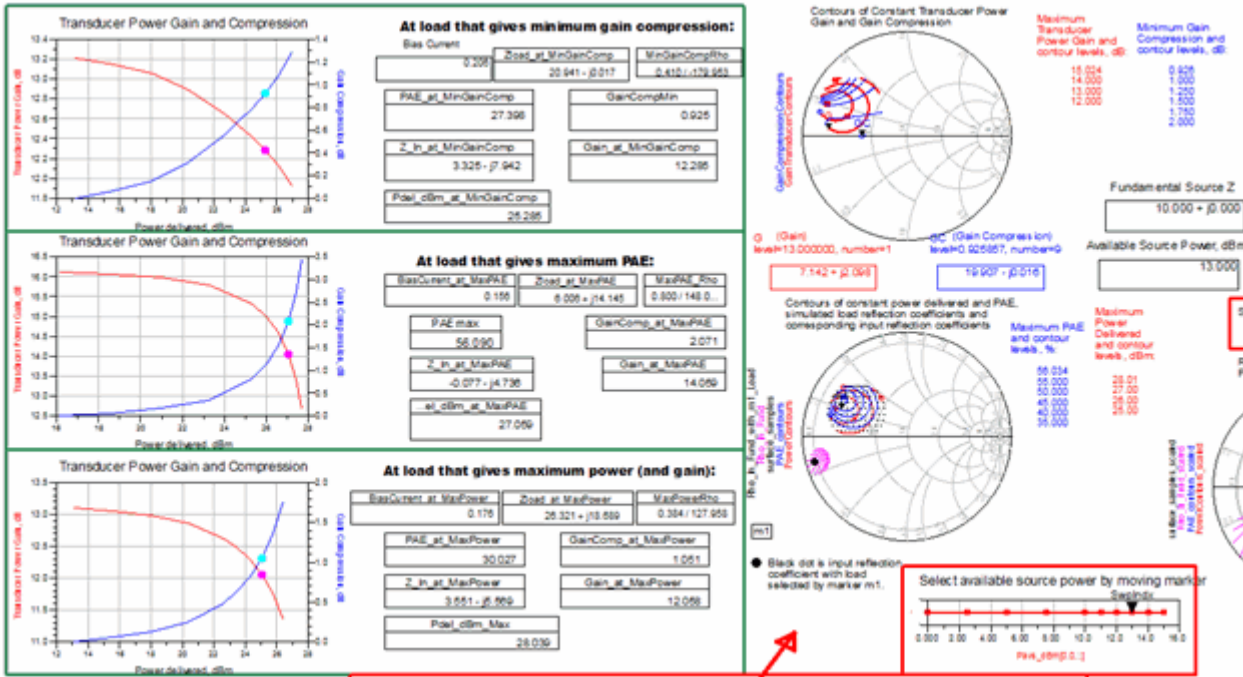
One Tone, Swept Available Source Power Load Pull

The **One Tone, Swept Available Source Power Load Pull** setup adds a sweep of the available source power for each value of the load reflection coefficient. It enables you to see how the contours change as the device is driven from small signal to large signal into compression. The schematic has an additional parameter sweep for the available source power variable *Pavs_dBm*.



In addition to the various settings done in the **HB1Tone_LoadPull** schematic, you also have to specify the sweep limits for the Pavs (Power available from the source) variable. Here we have two ranges, one with a coarser step (assumed to be the linear region) and one with a finer step (in the compressed region.)

After running the simulation, open the **HB1Tone_LoadPull_PSwEEP** data display and make sure the default dataset name is set to the name of the dataset generated by the simulation.



The data display shows the transducer power gain, gain compression, power delivered, and PAE contours for a particular value of the available source power that you select using *SwpIndx* marker. The gain and gain compression curves that correspond to several optimal load points (for minimum gain compression, maximum PAE, and maximum power delivered) are also shown.

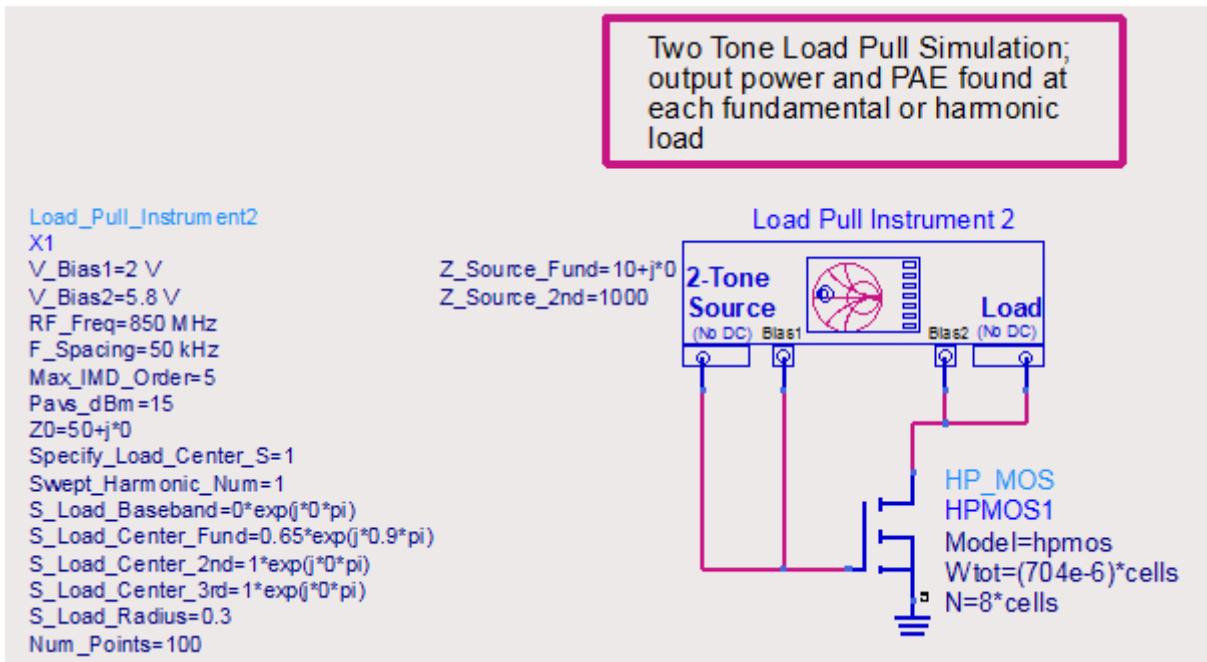
Two Tone Simulation

Contents

- *Two Tone, Constant Available Source Power Load Pull (dglpull)*
- *Two Tone, Swept Available Source Power Load Pull (dglpull)*
- *Two Tone, Constant Power Delivered Load Pull (dglpull)*
- *Two Tone, Constant Power Delivered Load Pull with Swept Parameter (dglpull)*

Two Tone, Constant Available Source Power Load Pull

This simulation setup and data display is identical to the "One Tone" version, except that now two tones are supplied instead of one. A two tone test signal stresses the device more because of its much higher peak-to-average ratio. The data display from this simulation shows the same information as shown in the one-tone version and also includes intermodulation distortion.



Here also you have to replace the sample device with yours and adjust the bias voltages as needed. In addition to all the other variables that you must specify as before, you need to specify the following two variables:

1. Frequency spacing between the two tones `F_Spacing`.
2. Maximum order of intermodulation distortion tones to be included in the simulation `Max_IMD_Order`.

The simulation results include similar information as shown above, with the addition of intermodulation distortion.

At load that gives maximum power (and gain):

BiasCurrent_at_MaxPower	Zload_at_MaxPower	MaxPowerRho
0.270	5.128 + j7.143	0.817 / 163.572

PAE_at_MaxPower
49.011

Z_In_at_MaxPower	Gain_at_MaxPower
-1.309 - j8.481	13.787

Pdel_dBm_Max	IMD_3rd_dBc_at_MaxPower
28.787	-22.482

At load that gives maximum PAE:

BiasCurrent_at_MaxPAE	Zload_at_MaxPAE	MaxPAE_Rho
0.196	6.264 + j11.632	0.788 / 153.4...

PAE_max
55.721

Z_In_at_MaxPAE	Gain_at_MaxPAE
-0.167 - j6.591	12.770

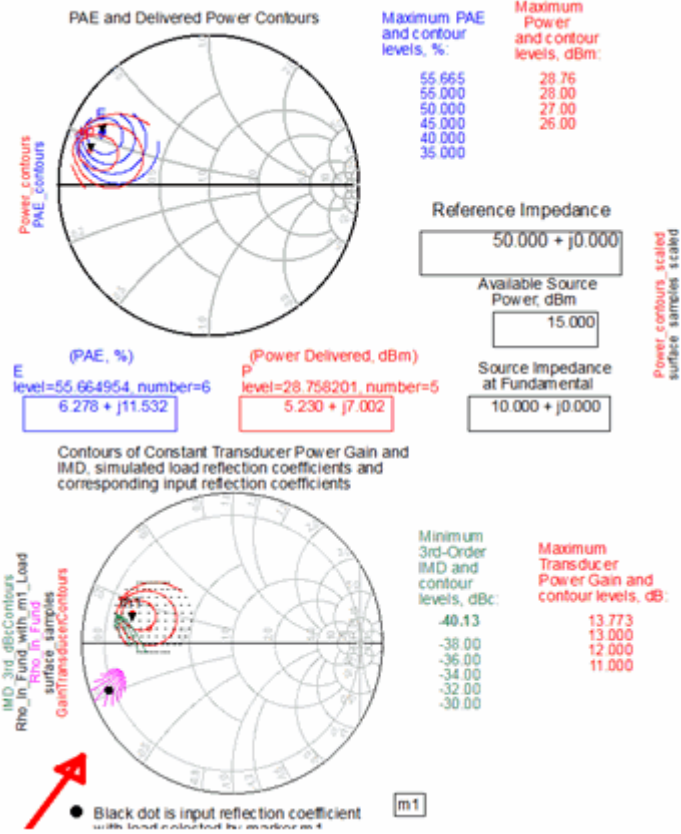
...el_dBm_at_MaxPAE	IMD_3rd_dBc_at_MaxPAE
27.770	-16.310

At load that gives minimum 3rd-order IMD:

BiasCurrent_at_MinIMD3	Zload_at_MinIMD3	MinIMD3_Rho
0.352	1.764 + j4.606	0.932 / 169.460

PAE_at_MinIMD3
21.175

Z_In_at_MinIMD3	Gain_at_MinIMD3



Two Tone, Constant Power Delivered Load Pull

This simulation setup and data display is identical to the "One Tone" version, except that now two tones are supplied instead of one. A two tone test signal stresses the device more because of its much higher peak-to-average ratio. The data display from this simulation shows the same information as shown in the one-tone version and also includes intermodulation distortion.

 A different device is used here.

You must start this simulation using Simulate > Optimize, or click the Optimize icon.

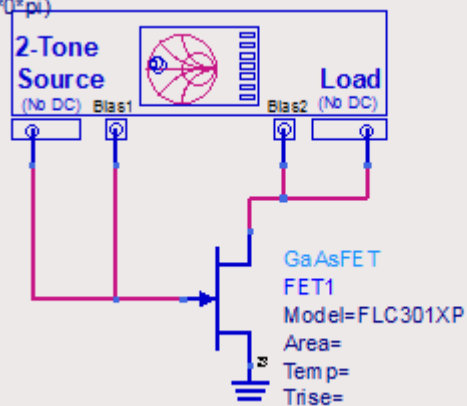
Two Tone Load Pull Simulation; the available source power is optimized such that the power delivered to the load is between $P_{del_dBm_goal_min}$ and $P_{del_dBm_goal_max}$. Prior to the optimization, a load pull is run with a small available source power. This simulation is considered the reference for the gain compression calculations.

Load_Pull_Inst_Const_Pdel2
X1

V_Bias1=-2 V
V_Bias2=5 V
RF_Freq=850 MHz
F_Spacing=100 kHz
Max_IMD_Order=4
Pavs_dBm_min=-3
Pavs_dBm_max=20
Pdel_dBm_goal_min=25
Pdel_dBm_goal_max=25.1
Z0=50+j*0
Specify_Load_Center_S=1
Swept_Harmonic_Num=1
S_Load_Baseband=0*exp(j*0*pi)
S_Load_Center_Fund=0.4*exp(j*1*pi)
S_Load_Center_2nd=1*exp(j*0*pi)

S_Load_Center_3rd=1*exp(j*0*pi)
S_Load_Radius=0.5
Num_Points=144
Z_Source_Fund=10+j*0
Z_Source_2nd=1000

Load Pull Instrument,
Constant Power Delivered



Here also you have to replace the sample device with yours and adjust the bias voltages as needed. In addition to all the other variables that you must specify as before, you need to specify the following two variables:

1. Frequency spacing between the two tones $F_Spacing$.
2. Maximum order of intermodulation distortion tones to be included in the simulation Max_IMD_Order .

The simulation results include similar information as shown above, with the addition of intermodulation distortion.

At load that gives maximum PAE:

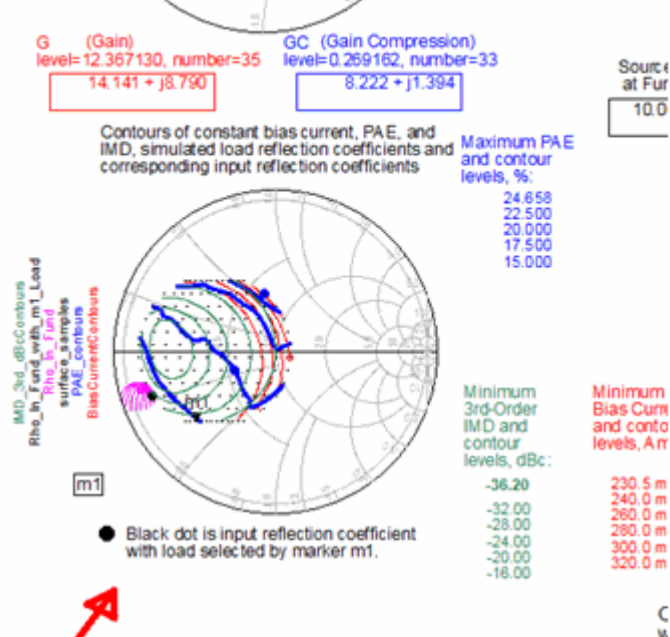
BiasCurrent_at_MaxPAE	Zload_at_MaxPAE	MaxPAE_Rho
0.230	32.969 + j29.219	0.384 / 100.8...
PAEmax	GainComp_at_MaxPAE	
24.683	7.363	
Z_In_at_MaxPAE	Gain_at_MaxPAE	
1.575 - j7.355	5.034	
...el_dBm_at_MaxPAE	IMD_3rd_dBc_at_MaxPAE	
25.031	-13.356	

At load that gives minimum gain compression:

BiasCurrent_at_MinGainComp	Zload_at_MinGainComp	MinGainCompRho
0.377	8.204 + j1.422	0.718 / 176.651
PAE_at_MinGainComp	GainCompMin	
16.926	0.269	
Z_In_at_MinGainComp	Gain_at_MinGainComp	
0.848 - j8.282	11.087	
Pdel_dBm_at_MinGainComp	IMD_3rd_dBc_at_MinGainComp	
25.100	-36.203	

At load that gives minimum 3rd-order IMD:

BiasCurrent_at_MinIMD3	Zload_at_MinIMD3	MinIMD3_Rho
0.377	8.204 + j1.422	0.718 / 176.651
PAE_at_MinIMD3	GainComp_at_MinIMD3	
16.926	0.269	
Z_In_at_MinIMD3	Gain_at_MinIMD3	
0.848 - j8.282	11.087	
Pdel_dBm_at_MinIMD3	IMD_3rd_dBcMin	
25.100	-36.203	



At load selected by marker m1:

BiasCurrent_at_m1	Zload_at_m1	Rho_at_m1
0.368	10.158 - j18.238	0.697 / -138.538
PAE_at_m1	GainComp_at_m1	
14.751	2.912	
Z_In_at_m1	Gain_at_m1	
3.831 - j8.594	5.597	
Pdel_dBm_at_m1	IMD_3rd_dBc_at_m1	
25.093	-16.797	

There is a clear trade-off between PAE and distortion. For this bias point, if you want maximum PAE, you suffer a lot of gain compression and intermodulation distortion.

At load that gives maximum PAE:

BiasCurrent_at_MaxPAE	Zload_at_MaxPAE	MaxPAE_Rho
0.230	32.969 + j29.219	0.384 / 100.8...
PAEmax	GainComp_at_MaxPAE	
24.683	7.363	
Z_In_at_MaxPAE	Gain_at_MaxPAE	
1.575 - j7.355	5.034	
...el_dBm_at_MaxPAE	IMD_3rd_dBc_at_MaxPAE	
25.031	-13.356	


Tolerating a lower PAE allows much lower gain compression and intermodulation distortion.

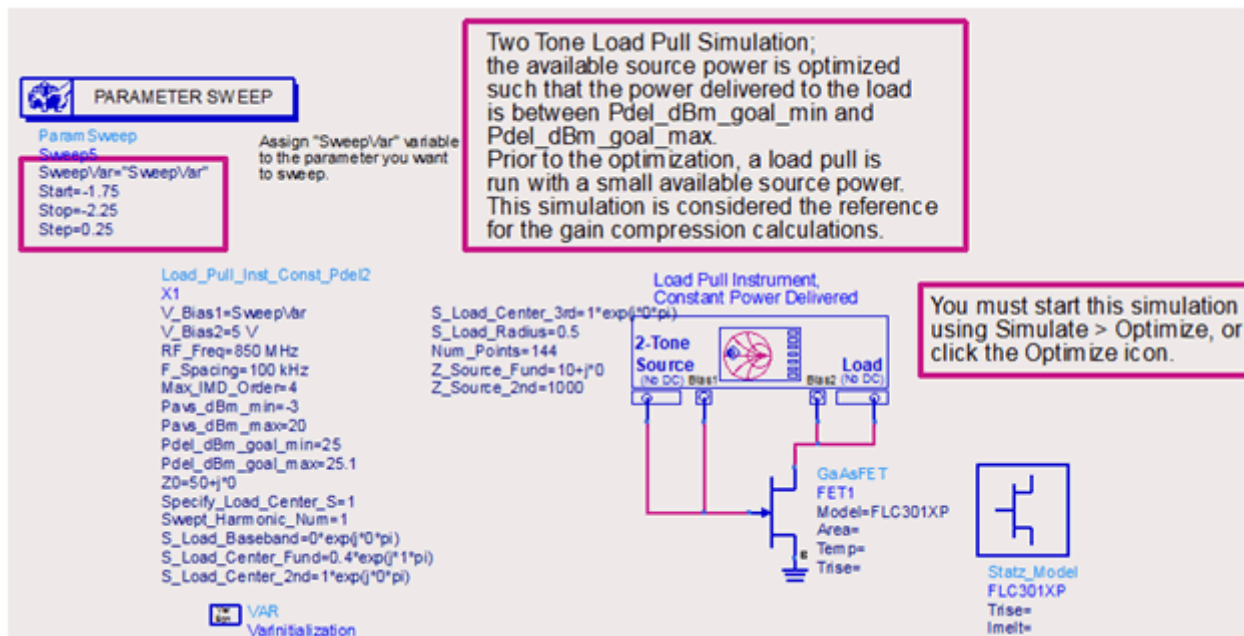
At load that gives minimum 3rd-order IMD:

BiasCurrent_at_MinIMD3	Zload_at_MinIMD3	MinIMD3_Rho
0.377	8.204 + j1.422	0.718 / 176.651
PAE_at_MinIMD3		GainComp_at_MinIMD3
16.926		0.269
Z_in_at_MinIMD3		Gain_at_MinIMD3
0.848 - j8.282		11.087
Pdel_dBm_at_MinIMD3		IMD_3rd_dBcMin
25.100		-36.203

Two Tone, Constant Power Delivered Load Pull with Swept Parameter

This simulation setup and data display are identical to the "One Tone" version, except that now two tones are supplied instead of one. A two tone test signal stresses the device more because of its much higher peak-to-average ratio. The data display from this simulation shows the same information as shown in the one-tone version and also includes intermodulation distortion.

 A different device is used here.



PARAMETER SWEEP

Param Sweep
Sweep5
SweepVar="SweepVar"
Start=-1.75
Stop=-2.25
Step=0.25

Assign "SweepVar" variable to the parameter you want to sweep.

Two Tone Load Pull Simulation; the available source power is optimized such that the power delivered to the load is between Pdel_dBm_goal_min and Pdel_dBm_goal_max. Prior to the optimization, a load pull is run with a small available source power. This simulation is considered the reference for the gain compression calculations.

You must start this simulation using Simulate > Optimize, or click the Optimize icon.

Load Pull Instrument, Constant Power Delivered

2-Tone Source (No DC Bias)
Load (No DC Bias)

GaAsFET
FET1
Model=FLC301XP
Area=
Temp=
Trise=

Statz_Model
FLC301XP
Trise=
Imeit=

Load Pull Parameters:

- Load_Pull_Inst_Const_Pdel2
- X1
- V_Bias1=SweepVar
- V_Bias2=5 V
- RF_Freq=850 MHz
- F_Spacing=100 kHz
- Max_IMD_Order=4
- Pavs_dBm_min=-3
- Pavs_dBm_max=20
- Pdel_dBm_goal_min=25
- Pdel_dBm_goal_max=25.1
- Z0=50+j0
- Specify_Load_Center_S=1
- Sweep_Harmonic_Num=1
- S_Load_Baseband=0*exp(j*0*pi)
- S_Load_Center_Fund=0.4*exp(j*1*pi)
- S_Load_Center_2nd=1*exp(j*0*pi)

Source Parameters:

- S_Load_Center_3rd=1*exp(j*0*pi)
- S_Load_Radius=0.5
- Num_Points=144
- Z_Source_Fund=10+j0
- Z_Source_2nd=10j0

VAR Initialization

Here also you have to replace the sample device with yours and adjust the bias voltages as needed. You have to assign the swept variable SweepVar to some parameter on the schematic. In addition to all the other variables that you must specify as before, you need to specify the following two variables:

1. Frequency spacing between the two tones $F_Spacing$.
2. Maximum order of intermodulation distortion tones to be included in the simulation Max_IMD_Order .

The simulation results include similar information as shown above, with the addition of intermodulation distortion.

At load that gives minimum bias current:

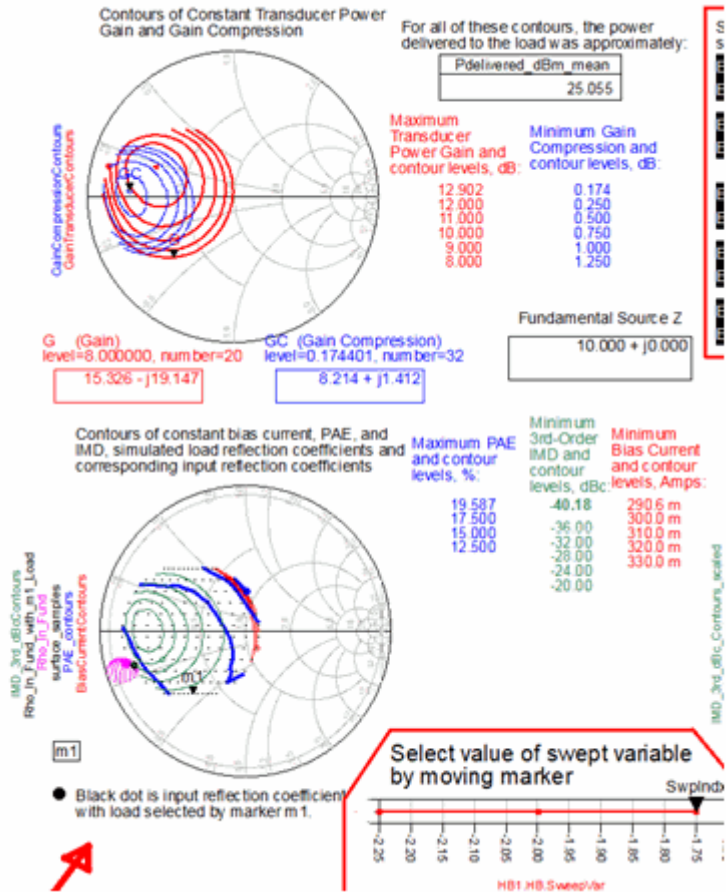
MinBiasCurrent	Zload_at_MinCurrent	MinCurrentRho
0.290	42.420 + j27.271	0.294 / 89.094
PAE_at_MinCurrent	GainComp_at_MinCurrent	
19.597	7.292	
Z_in_at_MinCurrent	Gain_at_MinCurrent	
1.735 - j7.296	5.082	
Pdel_dBm_at_MinCurrent	IMD_3rd_dBc_at_MinCurrent	
25.073	-13.668	

At load that gives maximum PAE:

BiasCurrent_at_MaxPAE	Zload_at_MaxPAE	MaxPAE_Rho
0.300	32.969 + j29.219	0.384 / 100.8...
PAE_max	GainComp_at_MaxPAE	
19.606	6.774	
Z_in_at_MaxPAE	Gain_at_MaxPAE	
1.534 - j6.942	5.913	
el_dBm_at_MaxPAE	IMD_3rd_dBc_at_MaxPAE	
25.092	-14.392	

At load that gives minimum gain compression:

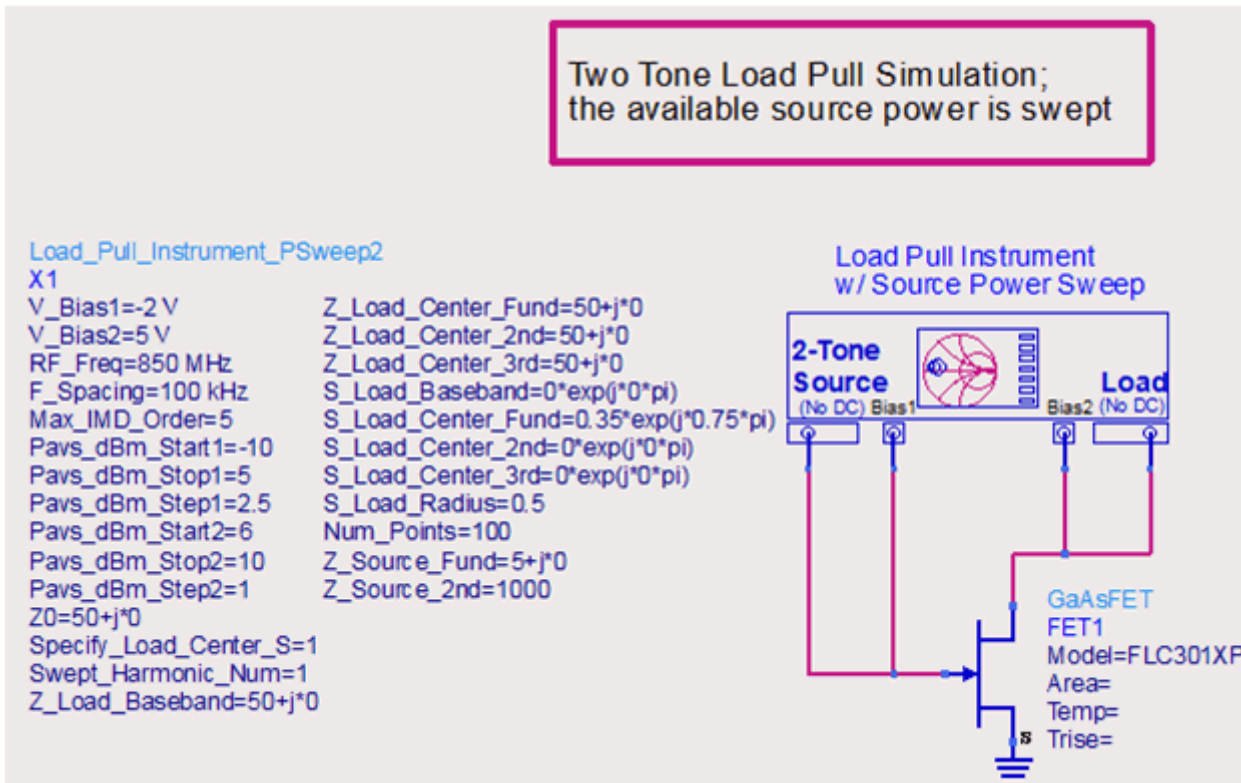
asCurrent_at_MinGainComp	Zload_at_MinGainComp	MinGainCompRho
0.458	8.204 + j1.422	0.718 / 176.651
PAE_at_MinGainComp	GainCompMin	
13.675	0.174	
Z_in_at_MinGainComp	Gain_at_MinGainComp	
0.897 - j8.023	11.840	
Pdel_dBm_at_MinGainComp	IMD_3rd_dBc_at_MinGainComp	
25.010	-40.183	



There is a clear trade-off between PAE and distortion. For this bias point, if you want maximum PAE, you suffer a lot of gain compression and intermodulation distortion.

Two Tone, Swept Available Source Power Load Pull

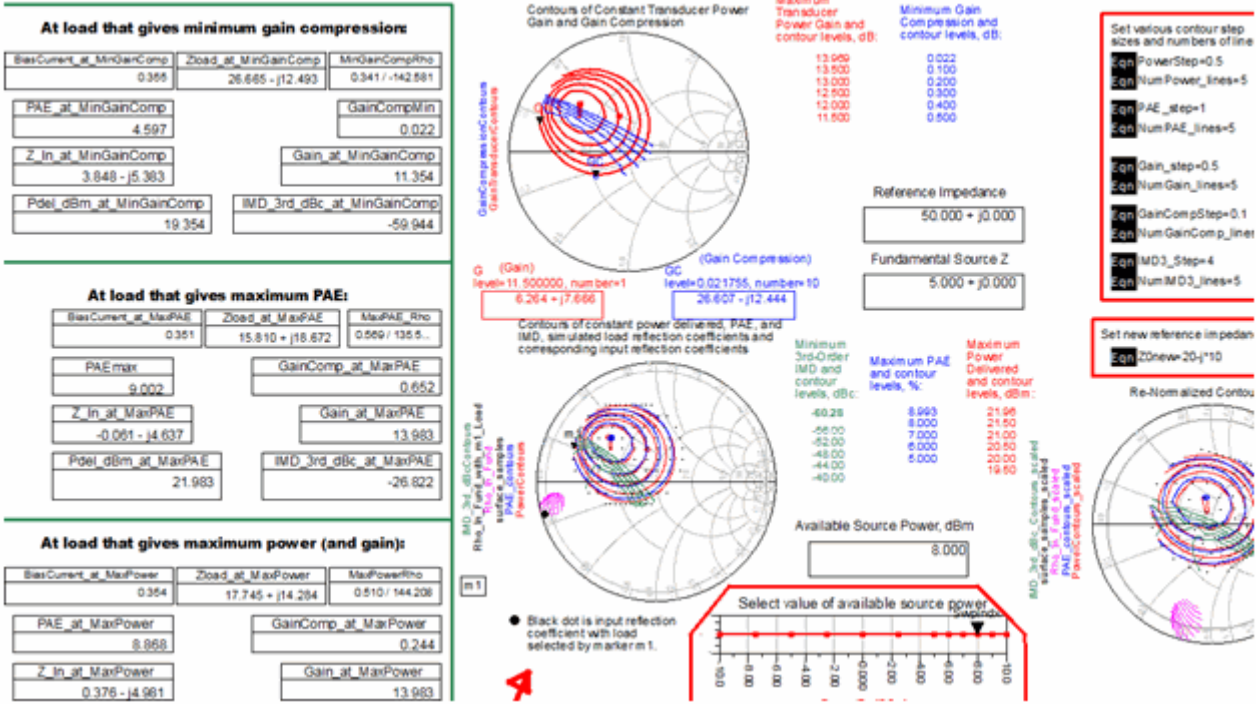
This simulation setup and data display is identical to the "One Tone" version, except that now two tones are supplied instead of one. A two tone test signal stresses the device more because of its much higher peak-to-average ratio. The data display from this simulation shows the same information as shown in the one-tone version and also includes intermodulation distortion.



Here also you have to replace the sample device with yours and adjust the bias voltages as needed. In addition to all the other variables that you must specify as before, you need to specify the following two variables:

1. Frequency spacing between the two tones $F_Spacing$.
2. Maximum order of intermodulation distortion tones to be included in the simulation Max_IMD_Order .

The simulation results include similar information as shown above, with the addition of intermodulation distortion.



WCDMA Signal Simulation

WCDMA uses a source with WCDMA modulation instead of one or two sinusoids. The adjacent channel power ratios are computed in addition to other performances such as gain, gain compression, and PAE. These simulations take longer to run, so you might want to run at least one of the one- or two-tone simulations above first.

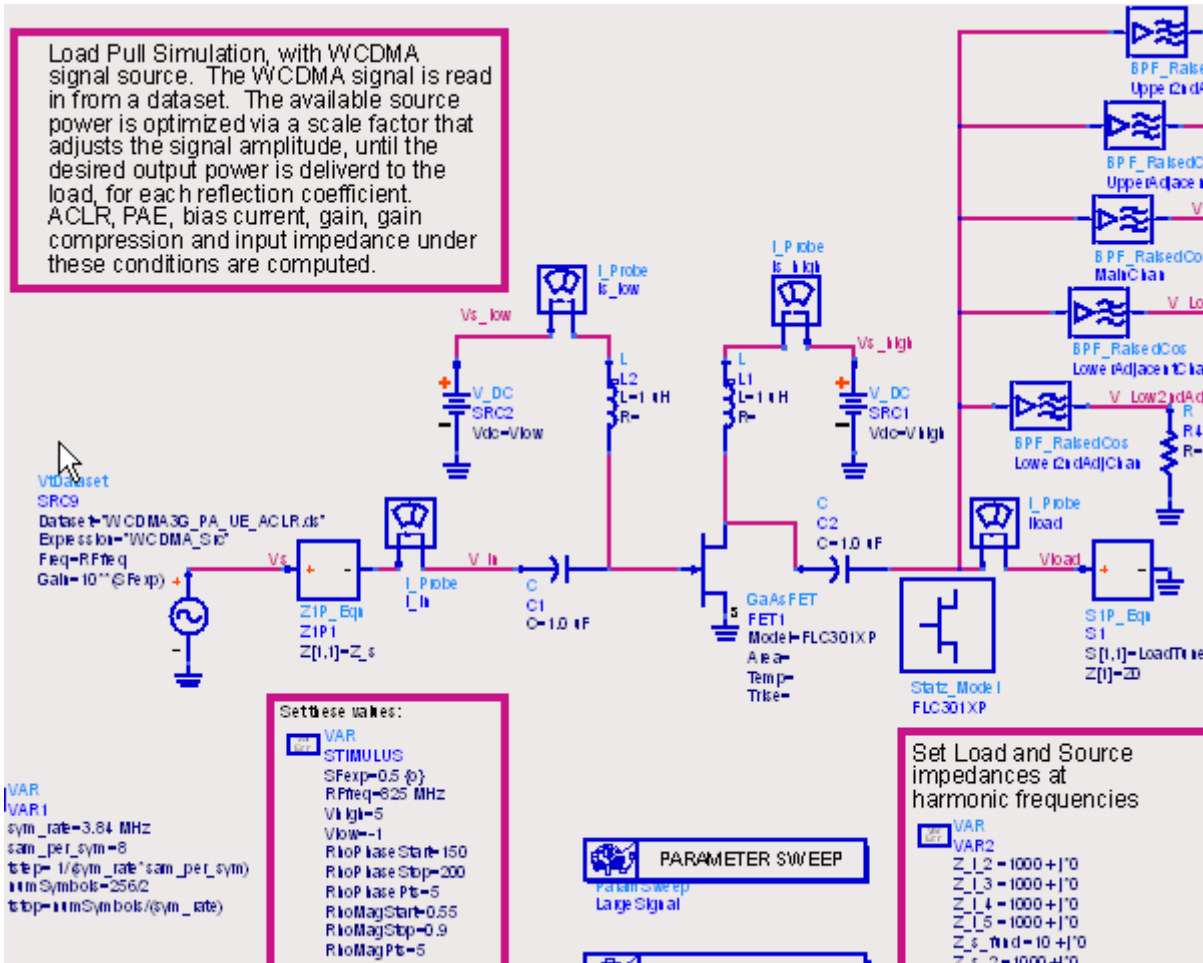
Select **WCDMA Load Pull > Constant Power Delivered, Mag/Phase Load Pull** to copy WCDMA_LoadPullMagPh_ConstPdel schematic and corresponding data display into your workspace.

This setup sweeps the load reflection coefficient in a fan-shaped region of the Smith Chart and optimizes the source power level for each load reflection coefficient until the desired power is delivered to the load. The source is a WCDMA signal read in from a dataset, and its amplitude (and thus the available source power) is set by a variable, *SFexp*, and the gain applied to this signal is $10^{**}(SFexp)$. The WCDMA signal was generated by connecting a Timed Sink to the output of the signal source in the ADS example `examples/WCDMA3G/WCDMA3G_PA_Test_wrk/WCDMA3G_PA_UE_ACLR` schematic.

The load pull is performed twice.

1. In first case, *SFexp* is set to 0.01. This is assumed to make the input signal small enough that the amplifier is operating linearly. The gain under this condition for each load reflection coefficient is the reference used to compute the gain compression.
2. In second case, *SFexp* is optimized until the desired power is delivered to the load.

The data display shows contours of constant PAE, ACLR, bias current, gain, and gain compression. The input reflection coefficient is also shown for a particular load that you specify. This allows you to pick the optimal load that produces the best PAE, ACLR, gain compression, or bias current, or make trade-offs among these specifications.



When using this schematic, there are a number of different things you need to specify, and these are listed in a paragraph on the schematic. First, you would replace the device with your device or amplifier. You have to set the bias voltages or modify the bias network, as needed. However, the data display calculates the DC power consumption assuming current probe Is_low is connected to supply voltage node Vs_low and current probe Is_high is connected to supply voltage node Vs_high . If you delete any of these or re-name them, you will have to modify the equations like **$Is_highDC = \text{mean}(Is_high.i[0])$** and the Pdc equation on the schematic.

You have to specify the range of phases and magnitudes of the reflection coefficients. The total simulation time will increase linearly with the product of the numbers of phases and magnitudes simulated. The tradeoff is that you should get better contour lines with more points simulated.

```

RhoPhaseStart=150
RhoPhaseStop=200
RhoPhasePts=6
RhoMagStart=0.55
RhoMagStop=0.9
RhoMagPts=5
    
```

If the device or amplifier is potentially unstable and the region of reflection coefficients that you specify includes the unstable region, the simulation may run into convergence problems. This would be due to the device wanting to oscillate. A solution to this problem is to add stabilizing components at the input, output, or in parallel with the device. You may want to use a simulation setup for this purpose, **DesignGuide > Amplifier > S-Parameter Simulations > Feedback Network Optimization to Attain Stability**. Another solution is to specify the range of reflection coefficients such that the unstable region is avoided.

You also have to specify the reference impedance, Z_0 , and the source center frequency, $RFfreq$.

You have to specify the nominal and allowed range of the signal source gain scale factor exponent, $SFexp$. It is necessary to adjust this gain to set the available source power, because we are using a voltage source to generate the signal.

VtDataset

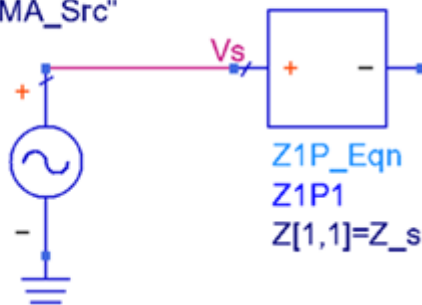
SRC9

Dataset="WCDMA3G_PA_UE_ACLR.ds"

Expression="WCDMA_Src"

Freq=RFfreq

Gain=10**(SFexp)



It is not obvious what the relationship is between this exponent and the available source power. However, the *WCDMA_SrcTest* schematic in the same workspace allows you to sweep this scale factor and calculate the corresponding available source power. In this case, stepping $SFexp$ from -1 to 1 increases the available source power from about -20 dBm to +20 dBm. This does vary with the source impedance you specify.

When setting the range of $SFexp$ values, you want the highest $SFexp$ value to correspond to the maximum available source power you will accept. For example, if you want to deliver 27 dBm to the load and you want the device to provide at least 7 dB of transducer power gain, you would set the maximum value of $SFexp$ to correspond to an available source power of approximately 20 dBm. The optimization will run fastest if $SFexp$ is allowed to vary over a relatively large range, but with the nominal value close to the value needed to give you the desired output power. You may get extremely high gain compression values for some reflection coefficients near the edge of the Smith Chart.

Figure: Prior to starting on-screen editing

Set these values:

```

Var Eqn  VAR
          STIMULUS
          SFexp=0.5 {o}
          RFfreq=825 MHz
          Vhigh=5
          Vlow=-1

```

Figure: While performing on-screen editing

Set these values:

```

Var Eqn  VAR
          STIMULUS
          SFexp=0.5 opt{ 0 to 1 }
          RFfreq=825 MHz
          Vhigh=5
          Vlow=-1

```

During the optimization, this variable is adjusted within the limits until the power that you want is delivered to the load.

You specify the desired power to be delivered to the load in *OptimGoal1*.

GOAL

```

Goal
OptimGoal1
Expr="mag(Pdel_dBm-27)"
SimInstanceName="Env1"
Weight=1.0
LimitType[1]="LessThan"
LimitMin[1]=
LimitMax[1]=0.25

```

In this case, we want the power delivered to be within 0.25 dB of 27 dBm. You may also specify different load and source impedances at the harmonic frequencies and (for the source) at the fundamental frequency.

Set Load and Source impedances at harmonic frequencies

```

Var Eqn VAR
VAR2
Z_l_2 =1000 + j*0
Z_l_3 =1000 + j*0
Z_l_4 =1000 + j*0
Z_l_5 =1000 + j*0
Z_s_fund =10 + j*0
Z_s_2 =1000 + j*0
Z_s_3 =1000 + j*0
Z_s_4 =1000 + j*0
Z_s_5 =1000 + j*0

```

There is a bank of raised cosine filters connected to the output node. These are used to compute the upper and lower adjacent and second adjacent channel leakage ratios. If you modify this setup to simulate a signal corresponding to a different standard (non-WCDMA, for example) then you most likely will need to modify the filter parameters and the channel frequency limits, which are specified as an offset from the carrier center frequency, *RFfreq*.

```

Meas Eqn MeasEqn
meas1
Up2ndAdjLimits=mainlimits+(10 MHz)
UpAdjLimits=mainlimits+(5 MHz)
mainlimits={{(-3.84 MHz)/2,(3.84 MHz)/2}
LowAdjLimits=mainlimits-(5 MHz)
Low2ndAdjLimits=mainlimits-(10 MHz)

```

As mentioned above the simulation time is directly proportional to the total number of different load reflection coefficients. It also depends directly on the number of symbols simulated at each load. When initially exploring the Smith Chart to find an approximate optimal load, it might be useful to run the simulation with a relatively small number of symbols. Later, after determining a smaller, optimal region of the Smith Chart, you might want to increase the number of symbols to get more accurate results.

```

Var Eqn VAR
VAR1
sym_rate=3.84 MHz
sam_per_sym=8
tstep= 1/(sym_rate*sam_per_sym)
numSymbols=256/2
tstop=numSymbols/(sym_rate)

```

If using a data file as the source, setting the simulation time step *tstep* equal to the time step in the data file is good, to minimize effects that may arise due to interpolation.



Envelope

Env1

Freq[1]=RFfreq

Order[1]=5

SweepOffset=12/sym_rate

Stop=tstop+12/sym_rate

Step=tstep

The Envelope analysis includes a *SweepOffset*. This simulates but does not keep the first 12 symbols in the simulation during which the input signal amplitude is ramping on. If you want to include this turn-on ramp data in the post-processing computations, just set **SweepOffset=0**.

Because the simulation includes an optimization, click *Optimize* icon to launch the simulation. If instead you just launch the simulation by hitting the F7 key or selecting **Simulate > Simulate**, an optimization will not be run and the data display will not display the simulation results.

After running the optimization, this *WCDMA_LoadPullMagPh_ConstPdel* data display shows the results.

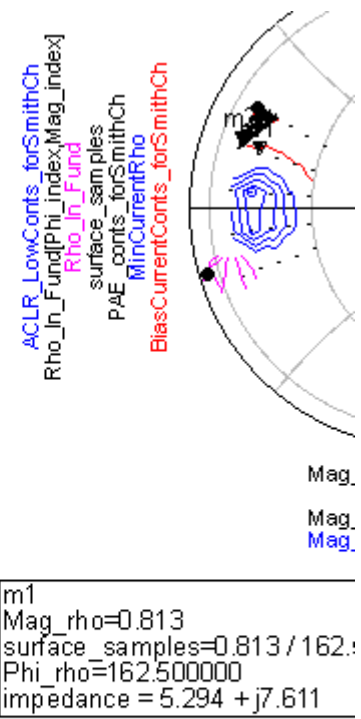
Lower and Upper ACLRs	
MaxPAE	-24.725 -24.257
-j9.027	
Lower and Upper 2nd ACLRs	
MaxPAE	-37.333 -36.540
27.172	

it gives minimum gain compression:

MinGainComp	Zload_at_MinGainComp	MinGainCompRho
0.774	5.182 + j2.160	0.813 / 175.000
GainComp	GainCompMin	Gain_at_MinGainComp
12.930	0.139	12.692
Lower and Upper ACLRs		
GainComp		
78 - j8.563		-46.303 -47.661
Lower and Upper 2nd ACLRs		
at_MinGainComp		
27.001		-61.513 -61.123

it gives minimum ACLR:

at_MinACLR	Zload_at_MinACLR	MinACLR_Rho
0.774	5.182 + j2.160	0.813 / 175.000
MinACLR	GainComp_at_MinACLR	Gain_at_MinACLR
12.930	0.139	12.692



At load selected

BiasCurrent_at_m1	
0.753	
PAE_at_m1	
14.133	

To see the contours effectively, you may need to change the *CurrentStep*, *PAE_step*, *Gain_step*, *GainCompStep*, and *ACLR_step* variables. These set the step sizes between the contours. The bias supply current calculations only include the current in the probe *Is_high*. If you change the name of the current probe, you will need to edit the BiasCurrent equation on the Equations page.

The upper Smith Chart shows contours of constant gain and gain compression. The lower left Smith Chart shows contours of constant bias current, power-added efficiency (PAE), and lower adjacent channel ACLR, as well as the simulated load reflection coefficients and the corresponding input reflection coefficients. The lower right Smith Chart shows the same data on a Smith Chart with a different reference impedance.

In the red boxes on the left side are data that correspond to a particular optimal condition such as minimum bias current, maximum PAE, minimum gain compression, or minimum ACLR. However, you have to make sure that the desired power delivered was actually achieved. In some cases with load reflection coefficients very close to the edge of the Smith Chart, the desired power will not be achieved.

The results show there is a trade-off between power-added efficiency and distortion. You can get slightly better PAE if you are willing to tolerate higher ACLR levels.

At load that gives maximum PAE:		
BiasCurrent_at_MaxPAE	Zload_at_MaxPAE	MaxPAE_Rho
0.493	$2.820 + j13.358$	0.900 / 150.000
PAEmax	GainComp_at_MaxPAE	Gain_at_MaxPAE
16.976	6.475	6.226
Z_In_at_MaxPAE	Lower and Upper ACLRs	
$-1.164 - j7.026$	-32.391	-29.710
Pdel_dBm_at_MaxPAE	Lower and Upper 2nd ACLRs	
25.935	-48.213	-48.853

Note that the power delivered specification is not satisfied. You could re-run the simulation, allowing SFexp to vary over a larger range, or you could move marker m1 to a load near this one and see if the power delivered specification is satisfied. Moving marker m1 allows you to select any of the simulated load reflection coefficients. The corresponding data appears in a separate box, which allows you to see potential tradeoffs as you move around the Smith Chart.

At load selected by marker m1:

BiasCurrent_at_m1	Zload_at_m1	Rho_at_m1
0.627	5.540 + j13.244	0.813 / 150.000
PAE_at_m1	GainComp_at_m1	Gain_at_m1
16.671	4.003	10.244
Z_In_at_m1	Lower and Upper ACLRs	
-1.109 - j6.373	-29.667	-31.243
Pdel_dBm_at_m1	Lower and Upper 2nd ACLRs	
27.066	-50.757	-50.776

The gain compression is still excessive, however.

At load that gives minimum ACLR:

BiasCurrent_at_MinACLR	Zload_at_MinACLR	MinACLR_Rho
0.777	5.172 + j3.029E-15	0.813 / 180.000
PAE_at_MinACLR	GainComp_at_MinACLR	Gain_at_MinACLR
12.911	0.139	12.212
Z_In_at_MinACLR	Lower and Upper ACLRs	
1.065 - j8.926	-47.249	-48.655
Pdel_dBm_at_MinACLR	Lower and Upper 2nd ACLRs	
27.057	-62.383	-62.053

The ACLR is somewhat sensitive to the load impedance.

The **WCDMA Load Pull > Constant Power Delivered, Circular Region Load Pull** menu pick copies into your workspace a schematic and data display nearly identical to the **WCDMA_LoadPullMagPh_ConstPdel** ones above, except that it sweeps a circular region of the Smith Chart instead of a fan-shaped region.

Time Taken by Simulation

For 25 different load reflection coefficients, *numSymbols*=128, and the optimization type set to Gradient with *MaxIters*=5, this simulation required about 4 minutes. For 49 different load reflection coefficients, about 6.75 minutes were required.