Mathematical stability problems in modern non-linear simulations programs

Ulrich L. Rohde , Rucha Lakhe

Synergy Microwave Corporation

e-mail: ulr@synergymwave.com

The use of nonlinear components such as bipolar transistors, GaAs FETs, and microwave diodes makes it necessary to predict large signal-handling performance. The traditional tools to do this were the SPICE approach and Volterra series expansion. The SPICE program is a program operating solely in the time domain. SPICE is an outstanding workhorse for dc analysis as a function of bias and temperature and transient analysis. The drawbacks of SPICE are (1) the lack of an optimizer; (2) the lack of distributed elements such as tee junctions, crosses, and others; and (3) the slow execution speed related to the time-domain approach. [1-3]

Another approach that has been tried is Volterra series expansion. This approach is a simulation where the actual computation time is somewhat independent of the values of the components used in a circuit. However, once the number of harmonics goes up, Volterra series expansion also becomes very time consuming. The Volterra series can be regarded as a nonlinear generalization of the familiar convolution integral.

The Volterra series also has the limitation that the degree of nonlinearity must be mild, as the representation otherwise requires an intractably large number of details for adequate modeling. The recently developed harmonic balance method avoids many of the time consuming mathematical approaches mentioned previously. This method is a hybrid time- and frequency-domain approach which allows all the advantages of a time-domain device model, combined with the strength of the steady-state frequency-domain technique, to be presented in the lumped and distributed circuit elements in which the device is embedded. The time-domain model can be completely general, thus bypassing complicated determination of coefficients by curve fitting over different bias levels.

How does the modern non-linear Program Work? [1]

For a fixed circuit topology (analysis case), the frequency domain is passed through only once; the admittance matrix of the linear subnetwork is computed and stored for subsequent use. In the timedomain path, the state-variable harmonics are first used to compute the corresponding time-domain waveforms. As mentioned earlier, these are fed to nonlinear device equipment to produce the time-domain device port voltages and currents. Voltage and current harmonics are then described by one- or two-dimensional fast Fourier transforms (FFTs) for the cases of single-tone and two-tone excitation, respectively. The voltage harmonics are used to generate "linear" current harmonics via the linear subnetwork admittance matrix. The two sets of current harmonics are finally compared to produce individual harmonic balance errors and a combined (global) harmonic balance error to be used in a convergence test.

In well-conditioned cases (e.g., FET circuits), a standard Newton-Raphson iteration may be used successfully as an update mechanism even though no starting-point information is available (i.e., if zero initial values are assumed for all harmonics). In such cases the harmonic balance errors are used via a simple perturbation mechanism to generate a Jacobian matrix. The latter is then inverted and applied to the error vector to generate the updated harmonic vectors. The algorithm is fast and accurate.

For circuits containing strongly nonlinear devices such as microwave diodes, a simple Newton iteration may sometimes fail to converge. To overcome this difficulty, Designer 5 incorporates a second iteration scheme based on a variable metric algorithm (quasi-Newton iteration), which is slower although considerably more robust than the regular Newton method.

In ill-conditioned cases, the quasi-Newton iteration may be used to approach the required solution. After this has been done to a satisfactory extent, automatic switchover to Newton iteration takes place, so that the approach solution can quickly be refined to any desired accuracy.

When circuit optimization is requested, the algorithm flowchart is modified. Harmonic balance errors are computed in the same way, but now the variable circuit parameters are also updated and the linear subnetwork admittance is computed at each iteration. An objective function is defined as a combination of harmonic balance error and a contribution arising from the electrical specifications. Such an objective is then minimized by the variable metric algorithm until a minimum close enough to zero is reached. Circuit parameters and state-variable harmonics are updated simultaneously, thus avoiding the nesting of nonlinear analysis and circuit optimization loops. Designer 5 is a general tool using the harmonic balance method for microwave. The harmonic balance method is a generic mathematical approach and is used for the first time in a commercial CAD software. Today the program of interest is Ansys's Designer 5. These modern programs are written mathematically so well that stability problems in oscillators do not occur in amplifiers.

Harmonic Balance Analysis (HBA)

Harmonic balance analysis is performed using a spectrum of harmonically related frequencies, similar to what you would see by measuring signals on a spectrum analyzer. The fundamental frequencies are the frequencies whose integral combinations form the spectrum of harmonic frequency components used in the analysis. On a spectrum analyzer you may see a large number of signals, even if the input to your circuit is only one or two tones. The harmonic balance analysis must truncate the number of harmonically related signals so it can be analyzed on a computer.

Analysis parameters such as **No. of Harmonics** specify the truncation and the set of fundamental frequencies used in the analysis. The fundamental frequencies are typically not the lowest frequencies (except in the single-tone case) nor must they be the frequencies of the excitation sources. They simply define the base frequencies upon which the complete analysis spectrum is built.

A project for harmonic balance analysis must contain at least the following: A top-level circuit, at least one nonlinear active device, and a frequency specification (including the number of harmonics of interest). Designer 5 has five categories of harmonic balance analysis:

- Single-tone analysis (single RF signal)
- Two-tone intermodulation analysis (two RF signals)
- Two-tone mixer analysis (one RF signal and one LO signal)
- Three-tone intermodulation analysis (three RF signals)
- Three-tone mixer analysis (two RF signals and one LO signal).

Formation of the Harmonic Balance Equations

Harmonic balance analysis involves the periodic steady-state response of a fixed circuit given a pre-determined set of fundamental tones [4, 5]. The analysis is limited to periodic responses because the basis set chosen to represent the physical signals in the circuit are sinusoids, which are periodic. The Fourier series is used to represent these signals. In the single-tone case, a signal is given by:

$$x(\mathrm{t}) = \sum_{k=-NH}^{NH} \left(\mathrm{X_k e^{jk\omega_0 t}}\right) \quad (1)$$

where $X_k = X_{-k}^*$, w_o is the fundamental frequency and NH is the number of harmonics chosen to represent the signal.

In harmonic balance, the circuit is usually divided into two subcircuits connected by wires forming multiports. One subcircuit contains the linear components of the circuit and the other contains the nonlinear device models as shown in the figure. The linear subcircuit response is calculated in the frequency domain at each harmonic component ($k*w_o$) and is represented by a multiport Y matrix. This is the function performed by linear analysis.

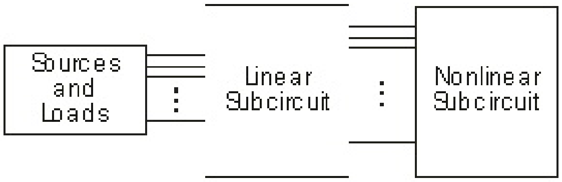


Figure 1: General block diagram showing the two subcircuits in the harmonic balance program.

Separation of the linear and nonlinear subcircuits

The nonlinear subcircuit contains the active devices whose models compute the voltages and currents at the intrinsic ports of the device (parasitic elements are linear and absorbed by the linear subnetwork). The port voltages (*v*) and currents (*i*) are analytic or numeric functions of the device state variables (*x*). Often the state variables represent physical voltages such as diode junction voltage or FET gate voltage, but are not restricted to physical quantities. The port voltages and currents are often functions of the time derivatives of the state variables (when a nonlinear capacitor is involved) and of time-delayed state variables (such as a time-delayed current source). Generally, the nonlinear device equations are of the form:

$$v(t) = \Phi\left[\mathrm{x(t)}, \frac{\mathrm{dx}}{\mathrm{dt}}, \mathrm{K}, \frac{\mathrm{d^n x}}{\mathrm{dt^n}}, \mathrm{x(t-\tau)}\right] \quad (2)$$

$$i(t) = \Phi\left[\mathrm{x(t)}, \frac{\mathrm{dx}}{\mathrm{dt}}, \mathrm{K}, \frac{\mathrm{d^n x}}{\mathrm{dt^n}}, \mathrm{x(t-\tau)}\right] \quad (3)$$

The device state variables, port voltages, and currents are transformed to the frequency domain using the discrete Fourier transforms as **X**, $\mathbf{V}_k(\mathbf{X})$ and $\mathbf{I}_k(\mathbf{X})$, respectively. Kirchhoff's current law is applied to the interface between the subcircuits at each harmonic frequency:

$$Y_k V_k(X) + I_k(X) + J_k = 0 \quad (4)$$

where $\mathbf{J}_k$ are the Norton equivalents of the applied generators. This constitutes the harmonic balance equations at each harmonic frequency. The object of the analysis is to find the set of state variables, **X**, to satisfy equation 4.

When the analysis begins, the state variables are typically set to zero and the left side of equation 4 is non-zero. We can write an error vector:

$$E_k(X) = Y_k V_k(X) + I_k(X) + J_k \quad (5)$$

whose Euclidean norm $\mathbf{E}^t\mathbf{E} = ||\mathbf{E}||$ is called the Harmonic Balance Error (HBE). If the HBE is reduced below a tolerance, we say that equation 4 is satisfied and a solution has been obtained.

Solving Methods

The process of solving the harmonic balance equations is an iterative one. An estimate of **X** is inserted into equation (5), **E** is calculated and if it is not below the tolerance then a new value of **X** must be determined and tried. Each such loop is termed as an iteration. There have been several methods used in the past to determine new values of **X** and two that have proven to be the most general and efficient are discussed here.

The state variables, **X**, and harmonic balance residuals, **E** are complex valued. In practice these are decomposed into their real and imaginary parts so that the number of real unknowns in **X** is ND*(2*Nt+1) where ND is the total number of nonlinear device ports and Nt is the number of frequency components (=NH for single tone analysis). Now we can write **E**(**X**)=**0** as a Taylor series expansion truncated after the first derivative term:

$$E(X) = 0 \approx E\left(X^{(n)}\right) + J\left(X^{(n)}\right)\left(X - X^{(n)}\right) \tag{6}$$

where **J**, the Jacobian matrix, is the first derivative matrix of **E** with respect to **X** and superscript n indicates the current iteration. Solving for **X** and using this for the next trial:

$$X^{(n+1)} = X^{(n)} - J^{-1}\left(X^{(n)}\right)E\left(X^{(n)}\right) \tag{7}$$

This is the Newton-Raphson update method where the last right-hand term is the update. This method works in one iteration if the set of equations is linear, but will take an unknown number of iterations if nonlinear. Often the update is reduced by a factor called the Newton damping factor so the method takes smaller steps each iteration. Convergence to a solution is not guaranteed and the iterates may diverge if not controlled. Designer 5 uses enhanced versions of the Newton-Raphson method to improve convergence and speed.

Designer 5 uses an algorithm that dynamically changes the Newton damping factor during solving based on the rate of convergence. If the solver has trouble converging, the factor will be made smaller to improve convergence. If it has been reduced by more than a predetermined factor, the solver will stop and an error will be reported.

An important aspect to note is the size of the Jacobian. If **X** contains ND*(2*Nt+1) elements, then **J** contains this number squared. As a practical example, if ND=10 (5 FETs) and Nt=4, then there are 8100 entries in **J** which takes 63 kBytes. This is relatively small, but Nt becomes much larger when multi-tone excitation is considered.

Some of the controlling functions are made accessible through the **CTRL** block in the project (see the *Control Blocks* chapter). The HBE tolerance can be changed from its default by: **HBTOL** ***x*** where ***x*** is the tolerance per device port per frequency component.

The absolute harmonic balance error allowed is scaled by the number of device ports and number of frequency components so that large circuits with many frequency components meet HBE criteria similar to those of small circuits. The default for HBTOL is $1.0\text{x}10^{-6}$. For the case of 2-tone intermodulation analysis and 3-tone analysis, the allowed harmonic balance is also scaled by the relative currents of the circuit. This reduces the allowed error (effectively reducing HBTOL) to provide better accuracy of the intermodulation products.

The number of allowed iterations before the program stops can be changed from its default value of 400 by: **MAXITER** *n,* where ***n*** is an integer.

Multi-tone Analysis

The discussion above was based on single-tone analysis for conceptual simplicity. Multi-tone analysis is simply an extension of single-tone [6,7,8]. In the single-tone case, a circuit is excited with an RF source and harmonics of that source are produced by the nonlinearities of the circuit. The set of harmonics, the frequency of excitation and DC are called the *spectrum* of the analysis. The single-tone spectrum is defined as: $S_1 = k*f_0$ where k=0, 1, ... NH and f_0 is the fundamental frequency. In multi-tone analysis the spectrum is modified to include the harmonic products of each fundamental tone. The harmonic products are just integer functions of the fundamental frequencies and indicate the allowed “bins” for power conversion within a circuit. The rest of the harmonic balance analysis is exactly the same.

The conversion between time-domain waveforms and Fourier coefficients is accomplished by the discrete Fourier transform in single-tone analysis. For each additional fundamental tone, a dimension is added in the transform. This allows efficient computation between domains, but becomes CPU-intensive when more than three-dimensions are encountered.

Local Oscillator Spectrum Initialization of Mixer Circuits

For mixer analysis cases where the primary interest is the conversion gain and the RF signal powers are small compared to the LO, the circuit can be analyzed using the LO signal only and the conversion gain is determined using small-signal (linear) frequency-conversion methods. This is performed using the Small-Signal Mixer Analysis option.

For cases where the RF signal power is not insignificant compared to the LO, a full mixer spectrum must be used. Compression of the conversion gain due to high RF power can then be analyzed. Here, the mixer problem can be divided into two parts to help speed the analysis. Firstly, the LO signal is analyzed using single-tone analysis; the RF signal is turned off. Single-tone analysis is usually very fast compared with a full two-tone analysis. Once the LO signal spectrum is found, the results are used to initialize the full mixer spectrum and the RF signal is turned back on. The full spectrum is then analyzed.

This method is most useful for three-tone mixer problems, due to the large number of spectral components used in the analysis. The primary use of the three-tone mixer analysis is to determine the intermodulation products of the IF products. This precludes the use of small-signal mixer analysis (since the intermodulation products cannot be determined using linear frequency conversion methods), but the RF signals are generally small compared to the LO. By solving the LO problem first, which is the primary nonlinear problem, and then introducing the RF signals, the analysis time can be reduced by a factor of about three. The actual time reduction depends on the circuit, the RF power levels, and the conversion gain.

Using the LO harmonic spectrum to initialize the full mixer spectrum is the default for three-tone analysis. The option is not the default for two-tone analysis, because significant time improvements have not been observed.

Number of Spectral Components and Reduced Spectrum Option

The number of spectral components considered in each type of analysis is related to the number of fundamental tones and the nonlinearity specified. The tables below list the number of spectral components for several nonlinearities considered in two-tone and three-tone analyses. The reduced spectrum option removes selected spectral components where significant harmonic power is not expected. The results of the analysis will not degrade at low power levels, but may yield different results for high power levels, depending on the circuit. Usually, the difference in results is negligible for practical cases.

The reduced spectrum option is especially useful for three-tone mixer analysis where the primary objective is to obtain the intermodulation intercept point with the IF. Low RF signal power levels are used and the analysis results are unaffected by the reduced spectrum option (the number of LO harmonics is not affected).

Number of Spectral Components (excluding DC) for Two-Tone and Three-Tone Intermodulation Analysis				
nonlinearity INTM m	**two-tone Full (default)**	**two-tone Reduced**	**three-tone Full (default)**	**three-tone Reduced**
3	12	8	31	21
4	20	12	64	31
5	30	22	115	79

6	42	30	188	115
7	56	44		209
8	72	56		
9	90	74		
10	110	90		

Number of Spectral Components (excluding DC) for Two-Tone Mixer analysis						
	#SB (M2) = 1		**#SB (M2) = 2**		**#SB (M2) = 3**	
#LO (M1)	**Full (default)**	**Reduced**	**Full (default)**	**Reduced**	**Full (default)**	**Reduced**
2	7	7	12	12	17	17
4	13	13	22	18	31	23
6	19	19	32	24	45	29
10	31	31	52	36	73	41
15	46	46	77	51	108	56
20	61	61	102	66	143	71
25	76	76	127	81	178	86
30	91	91	152	96	213	101

Note: #LO is the number of local oscillator harmonics; #SB is the number of RF sidebands

Number of Spectral Components (excluding DC) for Three-Tone Mixer analysis				
	INTM (M2) = 3		**INTM (M2) = 5**	
# LO (M1)	**Full**	**Reduced (default)**	**Full**	**Reduced (default)**
2	62	42	152	112
4	112	52	274	122
6	162	62		132
10	262	82		152
15		107		177
20		132		202
25		157		227
30		182		252

Notes: Entries that have been filled-in can be simulated

#LO is the number of local oscillator harmonics; INTM is the intermodulation order

The total number of spectral components grows very quickly with the level of nonlinearity and number and fundamental tones.

- Two-tone or three-tone intermodulation spectrum: The highest order group of spectral components, except those in the fundamental group (the intermodulation products), are ignored. In this case, the *n* in REDUCED*n* is ignored.

- Two-tone mixer analysis: All sidebands except the first sideband above the *n*th local oscillator harmonic will be ignored.

- Three-tone mixer analysis: All sideband groups at or below the *n*th local oscillator harmonic will be the same as the reduced two-tone intermodulation spectrum; all the sidebands above the *n*'th local oscillator harmonic will contain the two fundamental frequencies only.

Understanding the reduced spectrum is a little complicated. If the analysis is run with several reduced spectrum values and the spectrums are compared, then a better understanding of the spectral selections will be attained. Many studies were conducted and showed that the (default) reduced spectrum option for three-tone mixer intermodulation analysis affected analysis accuracy only slightly.

Sparse Jacobian Techniques

The Jacobian matrix, when properly arranged, can be treated as a sparse matrix by pre-setting some entries to zero [6]. The physical reason for doing this is that most of the power transfer takes place between the harmonic frequencies of the fundamentals and much less takes place between the other frequencies in the spectrum. We can therefore set these derivatives to zero within the Jacobian. When this criterion is not met, the band of non-zero entries is widened to include cross-harmonic terms.

Because the Jacobian structure is properly arranged, sparse matrix techniques are efficiently employed. General purpose sparse matrix solvers that analyze the sparsity structure are avoided and specialized solvers can be used that are much more efficient. Designer 5 automatically sets the bandwidth of the sparse tridiagonal matrix and dynamically alters it if the nonlinearity of the circuit is too great for the sparse assumptions. In this way the simulator achieves convergence using the minimal amount of computation time and memory that is possible for a given problem. For circuits with many devices under multi-tone operation, the CPU time may be decreased by a factor of 40.

A control parameter is made available to override the initial default sparsity parameter that controls the Jacobian bandwidth. The initial setting is 0 and can be changed by:

DIAG *n*

where ***n*** will be the initial sparsity parameter. Typical values range between 0 and 6. The sparsity parameter will still be dynamically altered during execution if needed. If ***n*** is greater than Nt/3 (Nt is the number of frequencies), then the program will use the full Jacobian. If only the full Jacobian is desired, then set n to a large number.

Using a sparse Jacobian does not affect the final values or accuracy of the results. It will only affect the convergence properties of the particular problem.

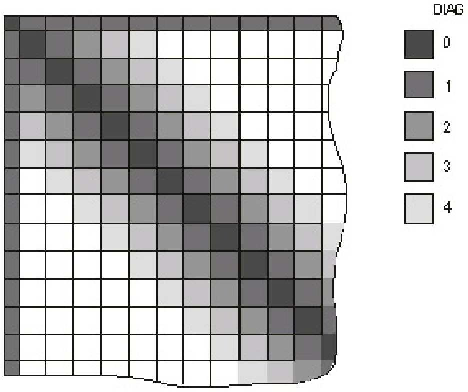


Figure 2: Sparse Jacobian block structure

Iterative Newton Method

One of the short comings of harmonic-balance methods is the large memory requirements when a circuit has many nonlinear devices and/or multi-tone analysis is needed. The Jacobian system matrix grows large and must be stored and factored. Sparse methods may not be enough to keep the problem within the memory bounds and acceptable computational resources of desktop computers. Designer 5 uses a technique that efficiently solves large systems of equations without direct factorization of the system matrix. In this way, there is no simplification or approximation made to the problem and the full accuracy of the conventional harmonic-balance method is completely maintained. The convergence and power-handling capabilities of conventional harmonic-balance analysis are also fully maintained. The method is completely automatic and does not require any user intervention. An internal software switch detects when the new method should be used and automatically invokes it.

A brief summary of the method and its advantages is given: Conventional harmonic-balance computes and stores the Jacobian matrix. The iterative solution of the harmonic-balance equations requires factorization of the Jacobian to obtain updates of the circuit voltages. As the number of nonlinear devices in the circuit increases and the number of spectral components used to analyze the circuit increases, the Jacobian matrix can become very large, requiring tens or hundreds of megabytes of storage and several minutes of CPU time to factor it. The calculation and factorization of the Jacobian typically occurs several times during a single harmonic-balance solution. The new method, based on an iterative approach known as *Krylov Subspace Methods*, avoids direct storage and factorization of the Jacobian. Rather, a series of matrix-vector operations replaces the full storage and factorization steps while retaining full numerical accuracy.

Observed speed-up factors depend on the number of nonlinear devices in the circuit and the number of spectral components used in the analysis as well as the convergence properties of the harmonic-balance algorithm. Speed improvements over conventional harmonic balance analysis from 2x to 10x for circuits consisting of a few transistors under two and three-tone excitation have routinely been observed. A circuit containing 20 FETs under three-tone analysis exhibited a speed improvement factor of 30x. Memory requirements have also been tremendously reduced. The 20 FET circuit originally required >200MB and now will analyze with 64MB. As the circuit becomes more “complex” the new methods provides better speed and memory improvements.

Designer 5 Outputs

During analysis, Designer 5 generates a number of output files that are used to store textual, graphical and initialization information. The files generated are: **myfile.aud**

The audit file contains textual information about the analysis. In its basic form it contains the final results of the network functions. Additional information can be requested by setting the verbosity flag in the control block as:

VERBOSE *n*

where 0 ≤ ***n*** ≤ 4. The higher the verbosity number is, the more output that is generated about the final and initial points at each sweep step.

Sweeping Frequency, Power and Voltage Sources

Each source in the design can be swept in amplitude. Also, the tones defined for the analysis can be swept. When more than one source or frequencies are swept, an ambiguity arises as to the order of precedence. The following rules apply in the cases of multiple sweeps:

1) When more than one source is swept and no frequencies are swept, then the sources sweep in unison. That is, each source is stepped at the same time. This is a one-dimensional sweep.

2) When more than one frequency is swept and no sources are swept, then the frequencies sweep in unison. This is also a one-dimensional sweep.

3) When frequencies and sources are both swept, the program performs a two-dimensional sweep where the sources are swept in the innermost loop. A matrix results where the source sweep is the most rapidly changing index. An exception to this case is during noise analysis, where the swept frequency deviation will be the innermost loop.

Generating Large-Signal S-Parameters

Since large-signal S-parameters are poorly defined, but widely used, we will show two methods of generating them. If your definition of large-signal S-parameters is different, you can redefine the example to suit your own needs. Here lies the ambiguity as to what one means by large-signal S-parameters. It will depend on the specific application and must be tailored in each case. Presented below is one interpretation:

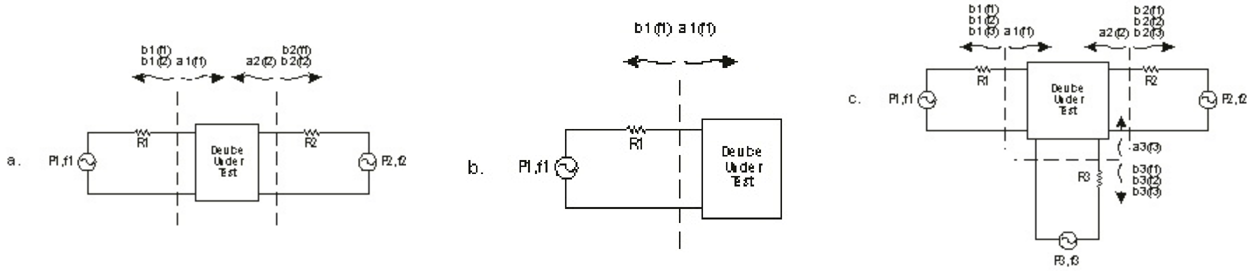


Figure 8: Schematic diagram: Generating large-signal *S*-parameters

The calculation of *S*-parameters in the large-signal regime is not as straightforward as it is in the linear, small-signal regime. The large-signal *S*-parameters are dependent on the power of the excitation sources at each external circuit port as well as the circuit bias and terminations. Guidelines will be given here on using Designer 5 to generate large-signal *S*-parameters, but the proper use of these S-parameters in circuit design is up to you.

Consider a two-port circuit whose large-signal *S*-parameters are desired. If we apply a source at port 1 with port 2 terminated, we could measure the reflected and transmitted waves, and conversely for a source applied to port 2 [10, 11]. However, this assumes that when the device under test is actually used, it will be terminated in the same impedance as it was tested. This is rarely the case. Typically the device is embedded in some matching network which presents a complex impedance to the device. Therefore, the operating regime of the device will change and its large-signal *S*-parameters will be altered.

We could then hypothesize that a source can be placed at each port and the traveling waves could be measured at each port. The problem here is that it is not possible to distinguish between the reflected wave at a port and the transmitted wave due to the source at the other port because the sources are the same frequency. If we perturb the frequency of one of the sources, then the reflected and transmitted waves due to each source can be resolved. This, however, requires a two-tone analysis. The situation is illustrated in the diagram for a two-port device under test.

The difference in frequency between the two sources can be made small, on the order of 0.001%. This is recommended for circuits of large *Q*. Typically the difference used is about 0.1% because the S-parameters of the device under test do not change rapidly with frequency.

This example shows some of the inconsistencies associated with large-signal *S*-parameters. For example, what happens to the power that is converted to other harmonic products? These will depend on the bias point, harmonic terminations, etc. In practice, the powers measured include all harmonic powers incident on the detector, whereas in the calculations we can pick out the precise fundamental powers. Also, we chose incident power levels as 10dBm and 8dBm, but how do we know if these are correct until *after* the design is done? There are several approximations like these that are assumed to be small when using large-signal *S*-parameters in active circuit design. Nonetheless, these parameters persist in design and can be computed using Designer 5.

In some cases where it can be approximated that one or more ports will be conjugately matched so a source doesn't need to be present there, higher port parameters can also be computed using repetitive analyses.

Other so-called conversion parameters can be computed. For example, if a mixer conversion matrix is desired between the RF and IF frequencies, the corresponding transmission parameters can be computed using TG between the proper harmonic numbers. The reflection coefficients can be found by using RL at the source ports and source harmonics.

Algorithm for Single-Tone Y-parameters Evaluation of Nonlinear Systems [12]:

For single-tone bandpass analysis, it is assumed that a nonlinear element's measurements are obtained when both the source and load have a 50Ω termination and the input is a bandpass single-tone signal. The measurements obtained are directly related to the large signal S-parameters of the two-port nonlinear element and the power available at the input and output ports.

For a given single-tone input, we can refer to large signal S-parameters as the operating point of the nonlinear two-port element when operating independently and terminated in 50Ω. One would certainly expect this operating point to change when this nonlinear element is embedded in a nonlinear topology system (as in Figure 3) composed of linear and nonlinear components.

The new operating point (i.e., large-signal S-parameters) is determined for each nonlinear element using an iterative algorithm where the levels of the incident powers at both ports are interpolated iteratively until the algorithm converges to the actual operating point. This nonlinear frequency domain iterative algorithm accounts for all nonlinearities and inter-stage mismatches in the system.

For the multi-channel nonlinear topology in Figure 3, it is always assumed that parallel nonlinear channels connected to the same linear electrical subsystem are not coupled (i.e., non-interacting). This, in simple terms, implies that signals traveling in one nonlinear path do not spill over into the other nonlinear path (by virtue of the S-parameters describing the linear electrical subsystem).

This effectively implies that the signals in nonlinear channels are not coupled, and as a result, the impedances Zs_n, Zin_n, $Zout_n$, and Zl_n for the n^{th} nonlinear element (1≤n≤N) in Figure 3 are well defined. With that important assumption, the iterative algorithm used for evaluating the operating point for each nonlinear element proceeds as follows (refer to Figure 3):

1. Assume an initial guess of $P1_n(0) = P2_n(0) = 0$for the first iteration (k=0) for the n^{th} nonlinear element (1≤n≤N).

2. Calculate the power-dependent S-parameters for the n^{th} nonlinear element (1≤n≤N) at the k^{th} iteration (k≥0). For the first iteration (k=0), this would basically yield the n^{th} nonlinear element small signal S-parameters since the initial estimate for the incident powers is zero.

3. Calculate the entire system's *Y* matrix, the impedances shown in the above figure $Zs_n(k)$, $Zin_n(k)$, $Zout_n(k)$, and $Zl_n(k)$, and the nodal voltages at the input and output ports of the n^{th} nonlinear element at the k^{th} iteration.

4. Recalculate the incident powers at each port with (1≤n≤N) and (k ≥ 0) according to

$$P1_n(k+1) = \frac{\left|\frac{v1_n(k)[zin_n(k) + Zs_n(k)]}{Zin_n(k)}\right|^2}{4Re\{Zs_n(k)\}}$$

$P1_n(k + 1)$ = The power available at the input port of the n^{th} nonlinear element at the k^{th} iteration when the output port is terminated in $Zl_n(k)$.

and

$$P2_n(k+1) = \frac{\left|\frac{v2_n(k)[Zout_n(k) + Zl_n(k)]}{Zout_n(k)}\right|^2}{4Re\{Zl_n(k)\}}$$

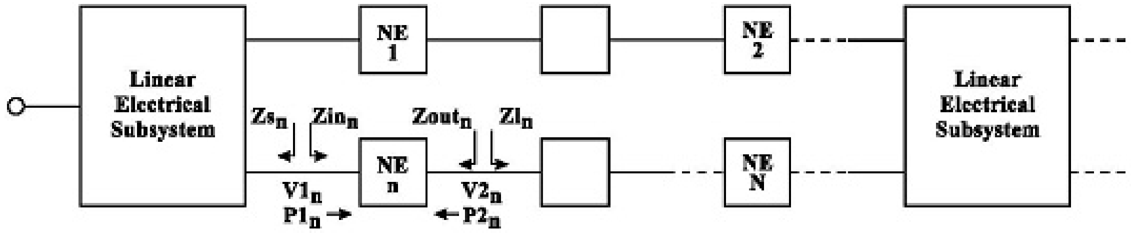


Figure 3: General non-linear electrical topology containing N non-linear elements (NE= nonlinear elements)

$P2_n(k + 1)$ = The power available at the output port of the n^{th} nonlinear element at the k^{th} iteration when the input port is terminated in $Zs_n(k)$.

5. Form the error function

$$Error(k) = \frac{\sum_{n=1}^{N}\sqrt{[P1_n(k+1) - P1_n(k)]^2 + [P2_n(k+1) - P2_n(k)]^2}}{\sum_{n=1}^{N}\sqrt{[P1_n(k)]^2 + [P2_n(k)]^2}}$$

The algorithm is assumed convergent if the condition$Error(k) \leq N \times 10^{-6}$ is met. If this convergence condition is not satisfied, steps 2-5 above are repeated until the algorithm converges. If the algorithm fails to converge after 30 iterations, an error in analysis message will be displayed.

Linear Electrical Discrete Time Simulation

During signal analysis, all electrical components and sub-designs are converted to time domain behavioral models (i.e., unidirectional models). This time domain model is extracted from the corresponding frequency response evaluated during the course of the simulation.

The extracted time domain behavioral models will typically contain the transient, steady state, and noise responses of the electrical sub-design (all in accordance with the frequency-domain signal and noise response of the electrical sub-design as well as impedance mismatches within the sub-design). Therefore, the processing of modulated signals through electrical components/sub-designs will include the transient, steady state, group delay, and noise effects.

There are two techniques available for discrete time simulation of linear electrical components and sub-designs: Convolution and Impulse Invariance.

Setting Discrete Time Simulation Control Parameters

For the discrete time simulation of a mixed-mode communications system, the user must carefully select the following control parameters:

1. $t_s = 1/f_s$= Simulation time step (which may assume different values at different points in the system)
2. *MIN_BW* = Minimum bandwidth of an electrical component or sub-design in the complete mixed mode system.
3. *MAX_RATIO* = Ratio of a local maximum to the global maximum in the impulse response.

Step 1: Choosing the Simulation Time Step

Choosing the simulation time step t_s is not a direct process in discrete time signal analysis. For a typical wireless communications system, the user always begins by setting the bit rate for binary data sources (or sampling rate if a source happens to be a waveform source).

Binary source components in Designer 5 (e.g., BSRC) have a parameter that determines their output bit rate. In a typical baseband modulation process, binary bits (at a user-defined bit rate) are mapped unto information symbols (to yield a given symbol rate). Each symbol is then represented by a user-selected number of samples (typically by upsampling or repeating each symbol) to finally yield a desired sampling rate f_s and the corresponding simulation time step of $t_s = 1/f_s$. These samples are then filtered to yield the discrete baseband modulation waveform $S(nt_s)$ described above.

When choosing the simulation time step $t_s$caution should be exercised to preserve the Nyquist criterion for the signal $S(t)$. In other words, the user must ensure that the discrete signal $S(nt_s)$ (at any point in the system) has at least a Nyquist sampling rate or higher, where the Nyquist sampling rate is equal to twice the bandwidth of the continuous signal $S(t)$.

At the same time, the simulation time step must be chosen in accordance with the Nyquist criterion for system bandwidth. In other words, if the bandwidth of a filter or electrical sub-design is *BW*, then, $f_s \geq 2BW$. Good results may be obtained for values of$f_s \geq 5BW$.

In conclusion, the simulation time step t_s must be chosen in accordance with the expected (baseband or bandpass) signal $S(t)$ and (baseband or bandpass) system bandwidths. For most practical applications, the signal and system bandwidths are of the same order, but in general, t_s at any point in the system must be chosen in accordance with the larger of the signal bandwidth and system bandwidth at that point.

In some applications, the user may be interested in generating direct waveforms without having to convert binary information to symbols and symbols to samples. The Designer 5 system has a good number of waveform sources that can generate a variety of periodic and transient waveforms. In addition, arbitrary waveforms may be imported for the discrete time system analysis from MATLAB, WinIQSim and other system simulators by means of an external waveform file.

All waveform source components in the Designer 5 system have a parameter that determines the desired sampling rate f_S, which in turn will set the desired simulation time step.

As an example, consider the second order (type 1) **PLL** project shown below in Figure 4. Note that the sample rate parameter for the complex constant source (CCONST) is set to 50 kHz (well beyond the Nyquist rate of the electrical sub-design or loop filter of the Phase Locked Loop). This sampling rate implies a time step of 20µs. After analyzing the project, the PLL time domain response shown below in Figure 5 will be displayed.

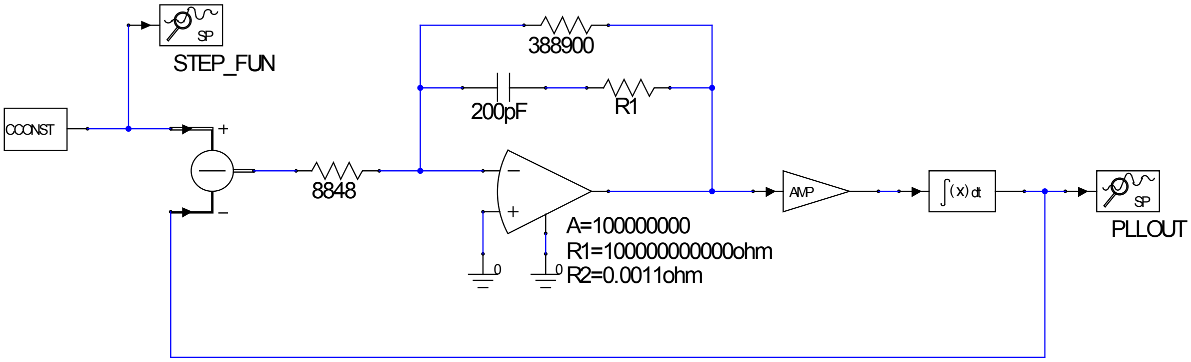


Figure 4: Example of PLL project to show the time domain response

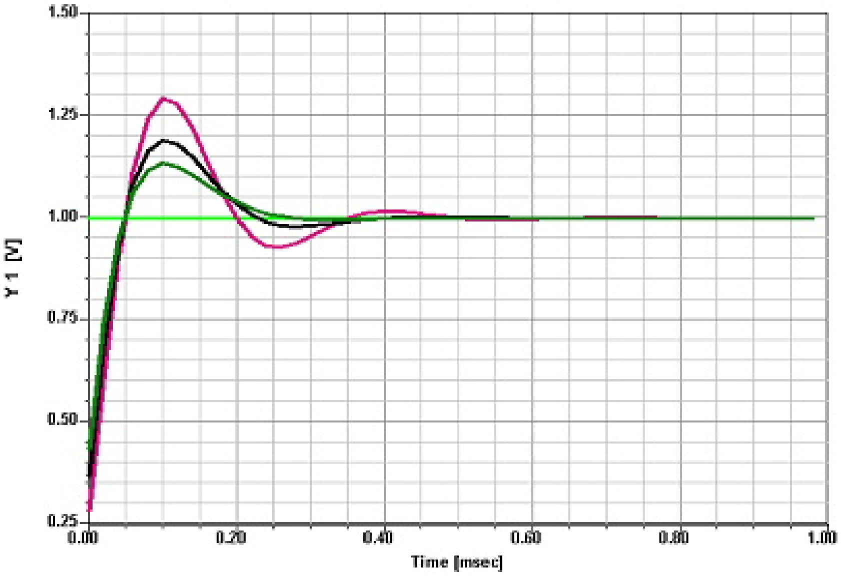


Figure 5: PLL time domain response for 3 different R1 values in the loop filter. Simulation sampling rate $f_S = 50KHz$ (values of R1 used are R1=74800, R1=174800, R1=274800)

If the sample rate of the CCONST source is readjusted to 20 KHz and the project is reanalyzed, the resulting change in time domain response for the PLL can be seen below in Figure 6 in terms of the "smoothness" of the output curves. There is a warning message displayed for this step.

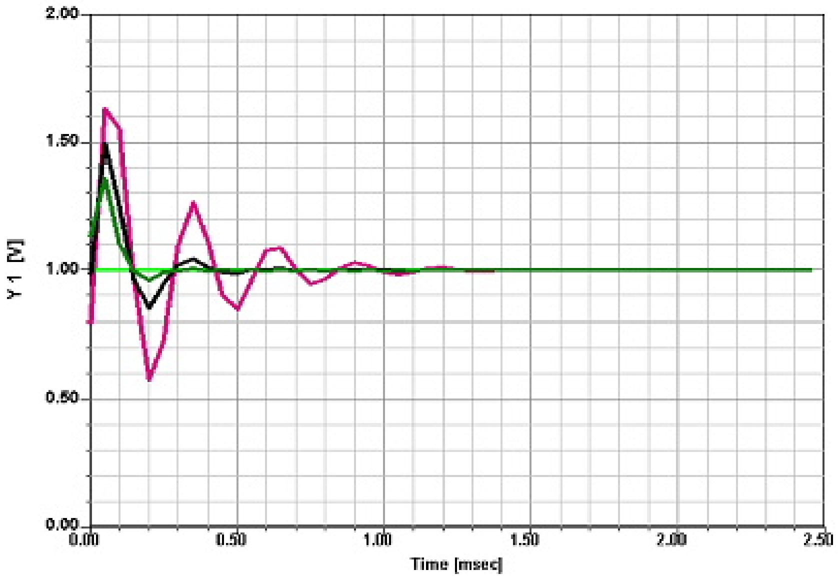


Figure 6: PLL time domain response for 3 different R1 values in the loop filter. Simulation sampling rate $f_s = 20KHz$ (values of R1 used are R1=74800, R1=174800, R1=274800)

As can be seen from the two figures (figure 5 and figure 6), choosing a higher sampling rate results in more accurate simulation results.

Step 2: Choosing the Minimum Bandwidth Control Variable

Another important control parameter for analyzing mixed systems is *MIN_BW* which is the minimum bandwidth of an electrical component or sub-design. It is up to the user to define this minimum bandwidth as the point where the frequency domain response is down by 3dB, 10dB or more. The simulation speed and accuracy of mixed mode systems largely depends on selecting this control parameter. The parameter *MIN_BW* may be specified through the Discrete Time Domain System solution setup. To specify a reasonable value for *MIN_BW*, it may be worthwhile to evaluate the frequency response (i.e., $S_{21}(f)$) of electrical sub-designs using the single tone frequency domain sweep analysis discussed in the Frequency Domain Analysis topic. This will help identify the *MIN_BW* in the system more appropriately.

The number of frequency points used to evaluate the frequency response of an electrical component/sub-design (during discrete time analysis) is given by the next power of 2 greater than or equal to *K*, where

K = {Minimum power of 2≥ $\frac{10f_s}{MIN\_BW}$}

and the frequency step is given by

$df = \frac{f_s}{K} \leq \frac{MIN\_BW}{10}$ = Sweep resolution used for warning message in the discrete time domain analysis dialog, where f_s is the sampling rate of the input signal(s) to the electrical component/sub-design**.**

Step 3: Choosing the Maximum Ratio Control Variable

Once the frequency response of a linear electrical RF sub-design is obtained, the impulse response is extracted from the frequency response by means of the inverse FFT operation and the pre-defined variable MAX_RATIO (0 < MAX_RATIO ≤ 1).

MAX_RATIO is a critical parameter used to determine the actual length of the impulse response used for discrete time simulation. Since the inverse FFT of the frequency response yields an impulse response of length *K/*2 samples, not all of these samples will be used in the actual discrete time domain simulation. Starting at the last impulse response sample at time $Kt_S/2$ and moving towards *t* = 0 (see Figure 7), the pre-defined variable MAX_RATIO is used to truncate the impulse response at the point where the ratio of the impulse response local maximum to the global maximum is ≥ MAX_RATIO. Thus, specifying a smaller MAX_RATIO value will typically result in longer impulse responses (and longer simulation time) but more accuracy.

The default value for MAX_RATIO is $1e^{-10}$, which should produce the most accurate results. At the expense of accuracy, the user may specify larger values for MAX_RATIO if a shorter simulation time is desired.

A graphical illustration of the effects MAX_RATIO has on determining the length of the impulse response is shown below.

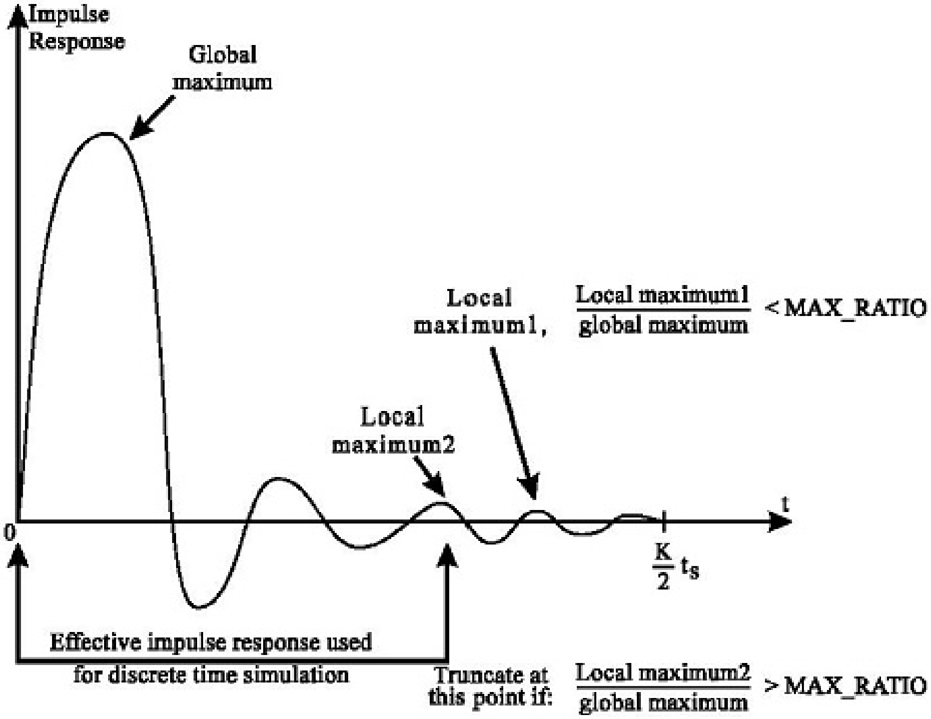

Impulse Response
Global maximum
Local maximum1,
$\frac{\text{Local maximum1}}{\text{global maximum}}$ < MAX_RATIO
Local maximum2
t
0
$\frac{K}{2} t_s$
Effective impulse response used for discrete time simulation
Truncate at this point if:
$\frac{\text{Local maximum2}}{\text{global maximum}}$ > MAX_RATIO

Figure 7: Graphical illustration of the significance of the maximum ratio control parameter.

Consider the PLL example discussed above. The responses shown in Figures 8 and 9 below reflect two different values for the MAX_RATIO control parameter, namely 0.001 in Figure 8 and 0.1 in Figure 9, with a fixed sampling rate of 50 kHz. Clearly, the impact of choosing a larger MAX_RATIO (Figure 9) can be seen. This is due to the fact that a larger MAX_RATIO resulted in a premature truncation of the impulse response of the PLL electrical loop filter, and consequently, the simulation results are inaccurate. In general, a longer discrete time impulse response (smaller MAX_RATIO) requires a longer simulation time, but will tend to yield more accurate results.

Note If starting at $Kt_S/2$ and moving towards *t* = 0, the search process fails to detect a local maximum/global maximum ratio that is less than MAX_RATIO, a warning message at the end of the simulation will be displayed.

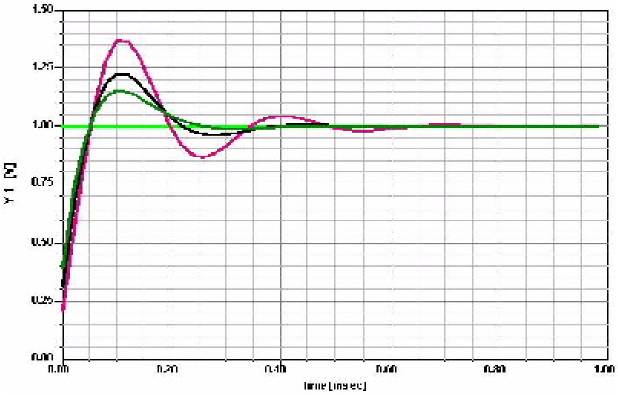


Figure 8: PLL response from MAX_RATIO = 0.001

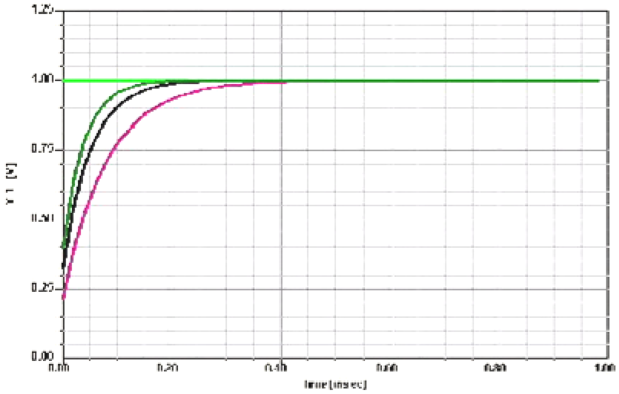
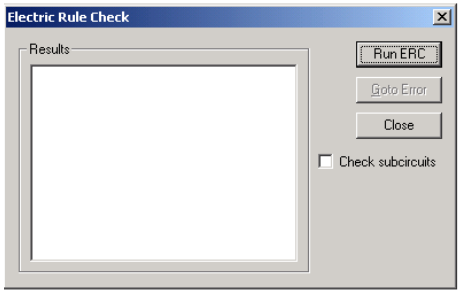


Figure 9: PLL response from MAX_RATIO = 0.1

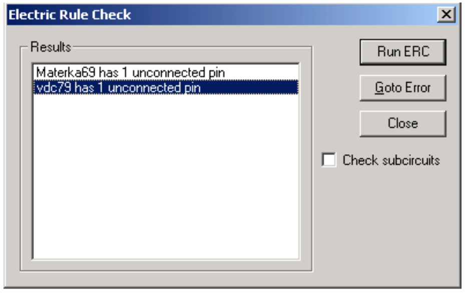
Checking Connectivity

The Electric Rule Check (ERC) feature checks the circuit for valid connectivity. ERC automatically conducts rule checking for ports, connections, and components of the active schematic.

1. To test for connectivity, select Schematic > Electric Rule Check, which opens the following dialog:



2. Select Check subcircuits to run the electric rule check on subcircuits of the active schematic display.
3. Click Run ERC to begin the error check.
4. If an error is displayed in the Results window double-click the error message or select the message and click Goto Error to go directly to the object in the Schematic Editor that caused the error.



Materka69 has 1 unconnected pin
vdc79 has 1 unconnected pin

Possible causes of Electric Rule Check error messages include:

• Unconnected Pins — A component, or port, with a pin that is not connected to anything else.

• Overlapping Components — A component that completely overlaps another such that the two components appear as one component. (Often caused by accidentally clicking twice when placing a component.)

• Nets with Multiple Output Pins — A net which has more than one output pin connected to it. (This is rare since most component pins are labeled as input and output.)

If You Encounter Convergence Difficulties

The nonlinear solver have been greatly improved, but you may still have convergence problems with some circuits, particularly, highly nonlinear circuits with bipolar transistors or circuits with high drive levels may pose a problem. For such circuits, the following hints are suggested to enable finding a solution:

1) Check the circuit connections. Improper node connections and/or missing units on parameters are the most common causes for convergence problems and messages that indicate “Singular Jacobian.” This commonly happens when the active devices are not biased properly or the signal path is not connected. Use the **Show Bias Point** option on the **Analysis** dialog to check for proper bias.

2) Check that the bias sources are properly connected. If constant current sources are used, make sure the current flow is in the desired direction.

3) Add losses in the circuit. When initial designs are simulated it is common to use ideal elements that don't have losses (e.g., transmission lines using characteristic impedance and electrical length only). This may pose problems in the analysis of the linear subcircuit at DC or when computing the Jacobian for nonlinear analysis.

4) Approach the solution point incrementally. By sweeping the source voltage or power toward the desired level, the circuit is driven gradually into the region where convergence is difficult to obtain. During a source sweep, the results of the previous step are used for the initial iterate of the subsequent step, the starting point is closer to the vicinity of the desired solution than a “cold” start from zero initial values. Designer 5 also employs automatic step reduction on power sweeps, whereby the step size is halved if convergence was not obtained on the previous step.

5) A similar solution to the above is to start the analysis from the previous solution. The *.VAR file should be backed up, the solution options should start from a previous solution, and the DC initialization should be disabled. This method can also be useful when manually tuning the circuit to achieve a desired response (if it is a single-point analysis).

6) If insufficient sampling points are used to represent the time-domain waveforms, there will be significant aliasing errors in the FFT.

References

[1]. G. Vendelin, A. Pavio, U.L. Rohde, *Microwave circuit design using the Linear and Nonlinear Techniques*, Wiley, New York, 2005

[2]. U.L. Rohde and D. Newkirk, *RF/Microwave circuit design for Wireless Applications*, Wiley, New York, 2000.

[3]. U.L. Rohde, A.K. Poddar, and G. Boeck, *The Design of Modern Microwave Oscillators for Wireless Applications: Theory and Optimization*, Wiley, New York, 2005.

[4]. V. Rizzoli, A. Lipparini, and Ernesto Marazzi, “A General-Purpose Program for Nonlinear Microwave Circuit Design,” *IEEE Trans. Microwave Theory Tech.*, vol. 31, no. 9, pp. 762-770, September 1983.

[5]. V. Rizzoli, A. Lipparini, A. Costanzo, F. Mastri, C. Cecchetti, A. Neri, and D. Masotti, “State-of-the-Art Harmonic-Balance Simulation of Forced Nonlinear Microwave Circuits by the Piecewise Technique,” *IEEE Trans. Microwave Theory Tech.*, vol. 40, no. 1, pp. 12-28, January 1992.

[6]. V. Rizzoli, C. Cecchetti, A. Lipparini, “ A General-Purpose Program for the Analysis of Nonlinear Microwave Circuits Under Multitone Excitation by Multidimensional Fourier Transform,” *17th European Microwave Conf.*, pp. 635-640, September 1987.

[7]. V. Rizzoli and A. Neri, “State-of-the-Art and Present Trends in Nonlinear Microwave CAD Techniques,” IEEE Trans. Microwave Theory Tech., vol. 36, no. 2, pp. 343-365, February 1988.

[8]. V. Rizzoli, C. Cecchetti, A. Lipparini, and F. Mastri, “General-Purpose Harmonic-Balance Analysis of Nonlinear Microwave Circuits Under Multitone Excitation,” *IEEE Trans. Microwave Theory Tech.*, vol. 36, no. 12, pp. 1650-1660, December 1988.

[9]. V. Rizzoli, F. Mastri, F. Sgallari, V. Frontini, “The Exploitation of Sparse-Matrix Techniques in Conjunction with the Piecewise Harmonic-Balance Method for Nonlinear Microwave Circuit Analysis,” *1990 MTT-S Int. Microwave Symp. Digest*, pp. 1295-1298, June 1990.

[10]. V. Rizzoli, A. Lipparini, and F. Mastri, “Computation of Large-Signal S-Parameters by Harmonic-Balance Techniques,” *Electron. Lett.*, vol. 24, pp. 329-330, Mar. 1988.

[11]. V. Rizzoli, F. Mastri, and F. Sgallari, and G. Spaletta, “Harmonic-Balance Simulation of Strongly Nonlinear Very Large-Size Microwave Circuits by Inexact Newton Methods,” *IEEE MTT-S*, pp. 1357-1360, 1996.

[12]. Ansys Designer 5 manual.

[13]. Andrei Vladimirescu, *The Spice Book,* Wiley NY, 1994