Agilent Technologies

Advanced Design System 2011.01

Feburary 2011

Transient and Convolution Simulation

About Transient and Convolution Simulation

Transient Convolution Simulator solves a set of integro-differential equations that express the time dependence of the currents and voltages of the circuit. The result of such an analysis is nonlinear with respect to time and, possibly, a swept variable. In ADS, this controller is available in the Simulation-Transient palette.

Transient Convolution Simulator enables you to:

- Perform a SPICE-type transient time-domain analysis on a circuit.
- Perform nonlinear transient analysis on circuits that include the frequency-dependent loss and dispersion effects of linear models such as S-parameter models. Such analyses are known as convolution analyses.

Refer to these topics:

- *Performing a Transient or Convolution Simulation* (cktsimtrans) has the minimum setup requirements for a transient simulation.
- *Examples of Transient and Convolution Simulation* (cktsimtrans) has a detailed setup for performing a transient simulation, using a Gilbert cell mixer as the example.
- *Transient and Convolution Simulation Description* (cktsimtrans) is a brief explanation of transient and convolution simulations.
- *Transient Simulation Parameters* (cktsimtrans) provides details about the parameters available in the Transient simulation controller in ADS.
- *Troubleshooting a Transient-Convolution Simulation* (cktsimtrans) offers suggestions on how to improve a simulation.

Note
You must have licenses for **either** the W2302 Transient Convolution Simulator Element **or** both E8884 High Frequency SPICE (transient simulator) and the E8885 Convolution Simulator **or** a bundle containing these to run these types of simulations. You may build the examples without the appropriate license, but cannot run the simulations.

Performing a Transient or Convolution Simulation

Start by creating your design, then add current probes and identify the nodes from which you want to collect data.

For a successful analysis:

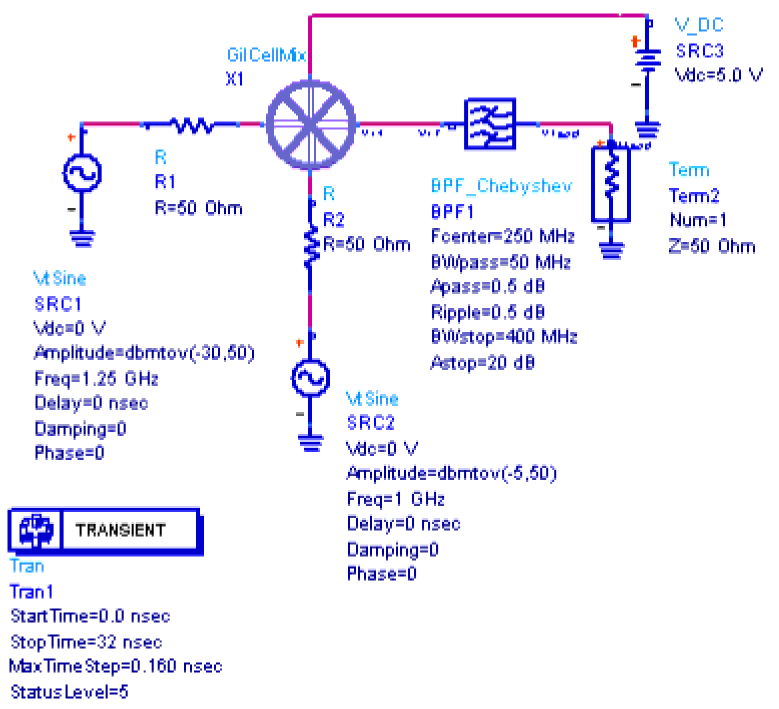
- When selecting sources, you can use either frequency-domain or time-domain sources. Transient sources are under the Sources-Time Domain palette. They are identified by the small *t* in their names (for example, VtStep:Voltage Source: Step).
- Add the Trans component to the schematic. Double-click to edit it. Fill in the fields under the *Time Setup* tab:
    - Enter the start and stop times.
    - Enter the Max time step. This is the largest time step that will be used in the simulation. It should be small enough to adequately sample the highest frequency expected in the circuit.
- To achieve the most accurate model, select the *Convolution* tab and set the Max Frequency and max impulse sample points. The other parameters here are related to generating impulse responses for convolution analysis, but in general accept the defaults. For more information, refer to the sections *Convolution Analysis* (cktsimtrans) and *Setting Max Frequency and Other Convolution Parameters* (cktsimtrans).
- If frequency is not defined by a frequency source, select the *Freq* tab and set the fundamentals and order.
- The parameters under the *Integration* tab set truncation, integration techniques, and charge accuracy. For information on integration techniques, see *Integration Methods Used in Transient-Convolution Simulation* (cktsimtrans).
- The parameters under the *Convergence* tab are used to improve convergence. For more information, see *Solving Convergence Problems* (cktsimtrans).
- You can use the steady state detector to find out the steady state conditions of a circuit. See *Using the Steady State Detector and Transient Assisted Harmonic Balance* (cktsimtrans).
- It is recommended that parameters not specifically mentioned here be left at the default values. For more information about each parameter, click *Help* in the Trans dialog box.

Examples of Transient and Convolution Simulation

This section provides an example using a Gilbert cell mixer, and lists other examples shipped with ADS that demonstrate transient simulations with other types of circuits.

The following figure illustrates the setup for a basic transient/convolution simulation.

Note
This design, *TRAN1*, is in the *Examples* directory under *Tutorial/SimModels_wrk*. The results are in *TRAN1.dds*.



Setup for Transient/Convolution simulation

To perform a basic transient/convolution simulation:

1. From the Component Palette, choose **Sources-Time Domain** or **Sources-Freq Domain** and select appropriate sources. Place the source at the input of the component or circuit under test, then define input power and edit other parameters as required. In a circuit employing a mixer, provide a source for the local oscillator (LO).

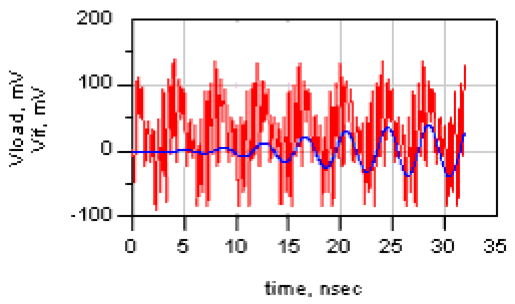
   Note
   Transient sources, available in the Time Domain Sources palette, are available for these simulations, and are identified by the small "t" in their names (for example, VtStep:Voltage Source: Step). However, standard frequency sources (in the Freq Domain Source palette) can also be used.

2. Ensure that the inputs and outputs of nodes at which you want data to be reported are appropriately labeled.

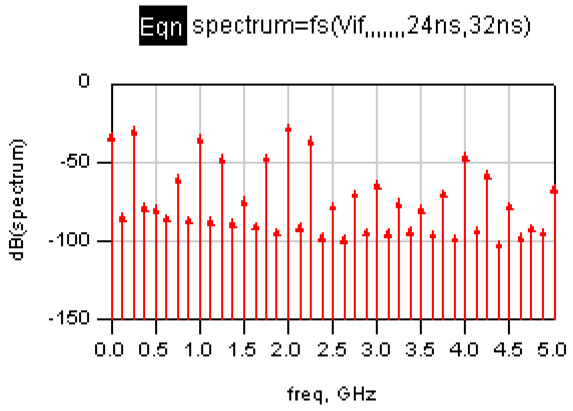
3. If they are needed, select from among various transient measurement equations in the Transient Simulation palette, place these on the schematic, and edit them as required. (See *Troubleshooting a Transient-Convolution Simulation* (cktsimtrans).) Their results can be plotted later in a Data Display window.
4. From the Component Palette, choose **Simulation-Transient**. Select and place the Tran simulation component on the schematic. You can run a basic simulation by editing the parameters *StopTime* and *MaxTimeStep* on the schematic.

> **Note**
> The *Max time step* value should be small enough to adequately sample the highest frequency expected in the circuit. The simulator may use a smaller timestep if needed but will never use a larger value.

5. To edit the frequency of the fundamental and any other frequencies to be considered, select the *Freq* tab. This frequency information is required only if the frequency is not set explicitly in the frequency domain sources.
6. The parameters that make it possible to obtain the most accurate model are *Max Frequency* and *Max impulse sample points*, under the *Convolution* tab of the simulation component. It is recommended that you leave the other parameters under this tab at their default settings and edit them only in special cases.
7. For details regarding all Transient parameters, double-click the simulation component to edit it, select the tab of interest, and click the **Help** button. Many parameters in this simulator apply only to special cases.
8. Launch the simulation. The data resulting from this simulation will be identified by TRAN1. The large trace is a plot of Vif versus time, the output of the filter showing all harmonic components. The small sinusoid is a plot of Vload following the filter. The filter output has been given enough time to approach a steady-state amplitude.

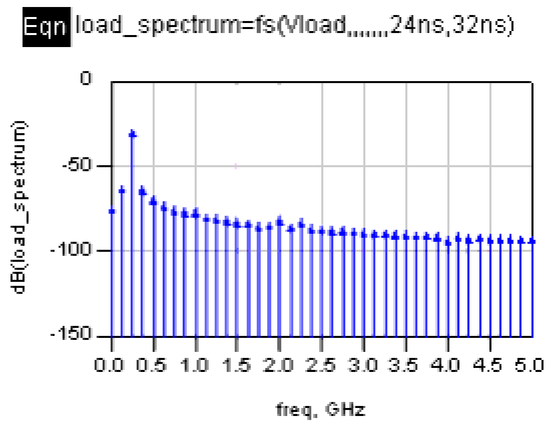


9. A plot of fs(Vif,,,,,,,4ns,16ns) shows a time-to-frequency transform from 4 to 16 ns for the mixer output before filtering. (The seven commas represent *fs* parameters not used here.) Run the simulation longer to observe the noise floor drop.



10. Finally, an *fs* plot of the filter output (Vload) between 24 and 32 ns shows the

response after steady-state amplitude has been approximated. Essentially only the bandpass frequency remains.



Additional ADS Examples

For a list of additional examples demonstrating transient simulation, see the following table. The table gives you the locations of their descriptions in the *Examples* documentation, and the location of the example workspaces available in the *$HPEESOF_DIR/examples* directory.

Example	**Documentation Location**	**Example Workspace Location**
Cosimulation of Baseband Sine Wave and Amplifier Circuit	Examples > Communication Systems > Baseband Sine Wave and Amplifier Cosimulation	/examples/Com_Sys/Co_Sim_wrk/Co_sim_1
Frequency and Time Domain Simulation of Multi-Coupled Microstrip Lines	Examples > RF Board > Multi-Coupled Microstrip Lines	/examples/RF_Board/MultilayerMeas_wrk/
Basic PLL Simulation using Circuit Envelope	Examples > RF Board > PLL Examples > PLL Simulations using Circuit Envelope	/examples/RF_Board/PLL_Examples/PLL_VideoExamples_wrk
TDR and S-parameter Simulations of Microstrip Step Discontinuities	Examples > RF Board > Microstrip Step Discontinuities	/examples/RF_Board/TDRmeas_vs_model_wrk
Various Simulations of A Power Amplifier	Examples > RFIC > Power Amplifier	/examples/RFIC/amplifier_wrk
RFIC Oscillator Simulations	Examples > RFIC > RFIC Oscillator	/examples/RFIC/RFICoscillator_wrk
Oscillator Simulations using Transient, Harmonic Balance and Envelope Simulators	Examples > Tutorial > Oscillator Simulations	/examples/Tutorial/Osc_Tran_HB_Env_wrk

Transient and Convolution Simulation Description

The transient and convolution simulators are SPICE-like in their operation. They solve a set of integro-differential equations that express the time dependence of the currents and voltages of the circuit under analysis. The result is a nonlinear analysis with respect to time and, possibly, a swept variable.

The main difference between the transient and convolution options lies in how each analysis characterizes the distributed and frequency-dependent elements of a circuit, as discussed below.

Transient Analysis

A transient analysis is performed entirely in the time-domain, and so is unable to account for the frequency-dependent behavior of distributed elements such as microstrip elements, S-parameter elements, and so on. Therefore, in a transient analysis, such elements must be represented by simplified, frequency-independent models such as lumped equivalents, transmission lines with constant loss and no dispersion, short circuits, open circuits, and the like. These assumptions and simplifications are usually very reasonable at low frequencies.

Convolution Analysis

A convolution analysis represents all the distributed elements in the frequency domain and hence accounts for their frequency-dependent behavior. The characterization of many RF and microwave distributed elements is best accomplished in the frequency domain, because the exact time-domain equivalents for these elements cannot always be obtained.

Convolution converts the frequency-domain information from all the distributed elements to the time domain, effectively resulting in the impulse response of those elements. The time-domain input signals at an element's terminals are convolved with the impulse-response of the element to yield the output signals. Elements that have exact lumped equivalent models-including nonlinear elements-are characterized entirely in the time domain without using impulse responses.

Note
In a convolution analysis, all elements are characterized by means of the full frequency-domain model, through the use of either an exact time-domain model or convolution. However, there may be minor differences between the results of a convolution simulation and the results of a transient simulation of the same circuit.

A convolution analysis requires both a convolution license and a transient license, and is performed whenever a convolution license is available. If the simplified approximate models are preferred, in this situation (for speed), set the *Use Approximate Models When Available* option to yes.

Transient/Convolution Simulation Process

The following steps describe how both the transient and convolution simulators operate:

1. The user specifies a time-sweep range, tolerances, and iteration limits.
2. A DC analysis is conducted to determine the system solution at zero time.
3. Inside the simulator, a breakpoint table is constructed to deal with frequency-domain-devices and data. Independent source waveforms frequently have sharp transitions that may not normally coincide with the time step calculated by the program. Such is the case with the piecewise linear sources. The breakpoint table contains a sorted list of all the transition points of the independent sources. During the simulation, whenever the next time point is sufficiently close to one of the breakpoints, the time step is adjusted to land exactly on the breakpoint. This prevents unnecessary time-step reductions in the vicinity of the transitions.
4. An internal control variable updates the current time, and the values of the independent sources are calculated at that time.
5. An attempt is made to solve the system of equations through numerical integration and a finite number of Newton-Raphson iterations. If the number of iterations exceeds *Max iterations per time point*, then the time step is reduced by a factor of *Integration coefficient mu* divided by 8. If this new time step is acceptable, the analysis is repeated from step 4. If *Integration coefficient mu* = 0, backward-Euler numerical integration is used. Otherwise, trapezoidal numerical integration is used.
6. Following convergence, the local truncation error is calculated. The default *Trapezoidal* integration method is used to estimate the error, unless *Gear's* method is selected.
7. The time step interval is calculated. By default, the time step is computed for transient analysis by means of the truncation error estimate method.
8. The error tolerance is compared with the value in the *Local truncation error over-est factor* field (under the *Integration* tab). If the error is within acceptable limits, the results are stored and analysis continues at the next time point. Otherwise, the analysis is repeated at a smaller time step.
9. Steps 3 through 9 are repeated until the user-specified time-sweep range has been analyzed.

Time Step Control Characteristics

These are the specific characteristics for time step control.
Local Truncation Error:

- Estimates the LTE made on every capacitor and inductor.
- Determines the time step size to ensure the largest LTE remains within the accepted tolerance.

- The estimated LTE is inversely proportional to TruncTol.
- The accepted tolerance is proportional to I_RelTol x TruncTol and V_RelTol x TruncTol.

Iteration-Count:

- Determines the time step size based on the number of Newton iterations required for previous time point.
- No direct relationship between iterations and LTE.
- Effectively controlled by Max time step (for linear circuits).

Fixed:

- The time step is fixed and equal to Max time step.

Break Points:

- Generated by built-in independent sources whenever an abrupt change in slope occurs.
- Ensure that corners in waveforms are not missed.
- ADS always places time points on a break point (except fixed time step).
- Backward Euler is used on time points that are the first time step after break points.
- The step size is reduced when time point is close to a break point.

Integration Methods Used in Transient-Convolution Simulation

Like SPICE, this simulator uses the trapezoidal integration method described by the following equation as the default method for calculating derivatives at each time step t in the simulation.

$$x'_{n+1} = \frac{2}{\Delta t}(x_{n+1} - x_n) \; - \; x'_n$$

For most circuits, this method will succeed. For those that do not, the simulator also supports Gear's backward difference method:

$$x'_{n+1} = \alpha_0 x_{n+1} + \alpha_1 x_n + \alpha_2 x_{n-1} + \ldots + \alpha_k x_{n+1-k}$$

In this equation, the index k is called the order of the integration.

For most circuits, Gear's method is no more accurate than the default trapezoidal integration technique. However, if a circuit analysis fails to converge, Gear's method may succeed where trapezoidal integration fails. In particular, oscillator circuits and any circuit that is characterized by stiff state equations may benefit from Gear's method.

Note
For a discussion of Gear's method and stiff state equations, refer to Chua and Lin, *Computer-Aided Analysis of Electronic Circuits: Algorithms and Computation Techniques*, Prentice-Hall, 1975.

If *Time Step Control* is set to *TruncError* and *Max Gear order* (under the *Integration* tab) is set to a number between two and six, the simulator will use Gear's method along with an adaptive stepsize algorithm that picks the largest possible step size at each point in the simulation. For each time step, the order of Gear's method will be chosen (up to the value of *Max Gear order*) to maintain accuracy with the largest possible time step. This potentially speeds up simulations with no loss in accuracy. If Gear Integration is selected with fixed timestep, then the integration will always be done at the fixed order given by *Max Gear order*.

The integration order at each time step is output to the dataset as the variable *tranorder*. This data is used by the fs() function, in data display, to do accurate interpolation of the data when an FFT is required. For the default trapezoidal integration, this will normally have a value of two, except at source-induced breakpoints where it will be one.

Using the Steady State Detector and Transient Assisted Harmonic Balance

You can perform a transient simulation with the steady state detector to find out the steady state conditions of a circuit. This includes whether or not steady state was reached, the time at which steady state was reached, and the frequency of oscillation in the event of having an oscillator circuit. To get this information, enable the *Detect Steady State* parameter and enter the frequency of the source that is driving the circuit for Freq[1] (or the potential oscillation frequency for an autonomous circuit). The resulting steady state values will appear in the ADS status window.

The Transient simulator may also be used to generate an initial guess for a harmonic balance simulation. For circuits that are highly nonlinear and contain sharp-edged waveforms (such as dividers), a transient simulation often provides a good initial guess for the starting point of harmonic balance.

Transient assisted harmonic balance is automated and can be set to Auto, On, or Off mode from the TAHB tab on the Harmonic Balance simulation controller. However, if you prefer to perform a manual TAHB simulation, there are two ways to do it.

On the *Freq* tab, fill in the frequency fields as you would for a harmonic balance simulation. The Frequency values can still be used, independent of this solution mode, to define the fundamental frequencies and *_freqN* variables used in sources. The *Order* and *Maximum order* information is used to determine the number of frequencies for which to compute a harmonic balance solution.

The first way is to enable the steady state detector and allow the transient simulator to capture the steady state portion of the solution waveforms. When taking this approach, be sure to give at least one the frequency and order parameter (Freq[1], Order[1]), and select the box labeled *Write initial guess for HB*. The transient simulator will report whether or not steady state was reached, and if so, the time at which it was reached and frequency of oscillation (when simulating an oscillator). The simulator will stop once steady state has been reached and transform just the last period of the solution.

The second way is to manually adjust the *StartTime* and *StopTime* to capture the steady state portion of the solution. As with the first method, at least one frequency and order parameter (Freq[1], Order[1]) must be given and *Write initial guess for HB* should be selected. The time to frequency domain transform does not start until after *StartTime* has been reached, so set *StartTime* appropriately so that the non-steady state portion is not transformed. *StopTime* should be set so at least one full period of the steady state solution occurs after *StartTime*. Set *MaxTimeStep* small enough to accommodate the largest signal frequency. With this approach, it is recommended to plot the transient results in the data display and verify that the waveforms are very near steady state. For best results, especially in multi-tone applications, enable *Apply Window*. This applies a window to the time domain data. This window helps to minimize the spectral leakage when multiple frequency tones are present.

In both cases, the name for the initial guess file should also be entered for the *File* parameter.

For circuits with multiple sources, it is strongly recommended to do a single tone transient simulation when generating the initial guess for harmonic balance. In other words, use only Freq[1] and Order[1] when setting up the fundamental frequency for a Transient simulation. This should be the most nonlinear tone. This is typically the tone with the largest power that would drive the circuit into compression.

Monte-Carlo Noise in the Time Domain

Noise in transient simulations can be added as pseudo-random voltages and currents at each timepoint. The noise has a Gaussian amplitude distribution. The noise can be frequency- or bias-dependent, as appropriate for each component. All components that generate noise in a linear or harmonic balance simulation generate the appropriate noise, with the following exceptions:

- S-parameter data from files that include a noise parameter block with NFmin, Rn and Sopt
- Behavioral components for amplifiers and mixers
- Noisy2Port component
- NoiseCorr noise correlation component
- SDD
- Mextram504 and Hicum transistor models

The noise is added to the signal, so there is no easy way to separate the signal from the noise. The full, nonlinear circuit equations are applied to this composite signal of random voltages and currents, so no small-signal assumptions about the relative size of the noise are required.

The *Noise bandwidth* parameter (under the Transient Noise tab) enables the generation of noise in transient analysis. Because transient analysis uses a variable timestep method, the noise must be limited to a bandwidth of less than 1/(2*MaxTimeStep). If this parameter is left blank or specified as zero, no noise is generated.

The *Noise scale* parameter is used to multiplicatively scale all of the noise sources in the circuit. This can be useful when the noise levels are very low, as this allows them to be increased so they are visible above the numerical noise in the simulation. However, if the noise is increased too much, it can change the nonlinear operation of the circuit.

The noise is generated by a random number generator. It will produce a different sequence of random numbers each time the simulation is run. If a repeatable sequence is required, it can be obtained by setting the simulator variable *__randseed* to an integer value with a schematic equation. For example,

`__randseed=12345` (two underscores precede `randseed`)

Transient Simulation Parameters

ADS provides access to Transient simulation parameters enabling you to define aspects of the simulation listed in the following table:

Tab Name	**Description**	**For details, see...**
Time Setup	Sets parameters related to time and frequency.	Defining the Time Setup
Integration	Selects an integration mode and sweep offset, turns on source and resistor noise, and sets device-fitting parameters. Options for oscillator analysis are also available.	Setting the Integration Method
Convolution	Sets parameters related to convolution analysis setup. Includes access to advanced convolution options.	Setting Up Convolution Analysis
Convergence	Sets parameters related to achieving convergence.	Setting Up the Convergence
Options	Sets parameters related to simulation reporting levels and saving operating point level data.	Setting Up Optional Parameters
Noise	Sets noise bandwidth and scale.	Defining the Noise Parameters
Freq	Sets parameters for the fundamental frequencies which are used for computing an initial guess for a harmonic balance simulation. This approach is called Transient-Assisted Harmonic Balance (TAHB).	Setting the Fundamental Frequencies
Output	Selectively save simulation data to a dataset.	For details, see *Selectively Saving and Controlling Simulation Data* (cktsim).
Display	Control the visibility of simulation parameters on the schematic.	For details, see *Displaying Simulation Parameters on the Schematic* (cktsim).

Using Transient Parameters in ADS

When using this controller, here are tips about preparing your design for simulation:

- While ten Transient parameters affect convolution, some components allow eight of these parameters to be set individually for the component.
- Component values override those on the controller.
- Whenever possible, set values on the component.
- Don't restrict adaptive impulse calculation except where needed.
- Don't limit Max Frequency (ImpMaxFreq) for all components if only one component requires a limit.

Setting Initial Conditions with InitCond Components

There are two elements for setting initial conditions in a transient simulation InitCond and InitCondbyName. InitCond and InitCondByName are used to provide an initial DC value for transient analysis only. These elements attach the specified voltage source with a series resistor to the specified node(s) to force a value. The DC solution for the entire circuit is then calculated. This DC solution is then used as the starting state for the transient analysis.

Defining the Time Setup

Following is information on the parameters related to time and frequency. The following table describes the parameter details. Names listed in the *Parameter Name* column are used in netlists and on schematics.

Transient Simulation Time Setup Parameters

Setup Dialog Name	Parameter Name	Description
Output Times		
Start time	StartTime	The time at which the simulator begins outputting time-point results. This enables control over large amounts of output data.
Stop time	StopTime	The time at which the simulator stops outputting time-point results. Must be long enough if steady state is needed. You must specify this parameter.
Max time step	MaxTimeStep	The largest time step to be taken in the simulation. You must specify this parameter.
Min time step	MinTimeStep	The smallest time step to be taken in the simulation. Generally the default value is satisfactory.
Limit timestep for Transmission Line	LimitStepForTL	Where transmission lines are involved, setting this option further limits the time step to half of the shortest transmission line's delay time.

Setting the Integration Method

Following is information on setting up the Integration portion of the simulation. The following table describes the parameter details. Names listed in the *Parameter Name* column are used in netlists and on schematics.

Transient Simulation Integration Parameters

Setup Dialog Name	Parameter Name	Description
Time step control method	TimeStepControl	
Fixed	Fixed	Selects a fixed time-step method. The simulation is performed with a uniform, constant time step that is specified by Max time step (under the Time Setup tab). It is quicker than the other methods. However, it is not as robust, because it cannot select a smaller time step when convergence problems are encountered.
Iteration Count	Iteration Count	Uses the number of Newton-Raphson iterations that were needed to converge at a time point as a measure of the rate of change of the circuit. If the number of iterations is less than an internal threshold, the time step is doubled; if the number is greater than Max iterations per time step, the time step is scaled by a factor of Integration coefficient mu divided by 8 (see below). This method has a minimal computational overhead, but does not take into account the true rate of change of circuit variables. Use this if no energy storage component is present and the Local truncation error is not checked.
Trunc Error	Trunc Error	Default. Uses the current estimate of local truncation error to determine an appropriate time step. Although it takes longer than Iteration count, it sets a meaningful error bound on computed output values.
Local truncation error over-est factor	TruncTol	A value against which the simulation's error tolerance is compared. In transient analysis each time step is computed by means of the truncation-error estimate method. If the error is within acceptable limits, the results are stored and analysis continues at the next time point. Otherwise, the analysis is repeated at a smaller time step. Increase this value to relax local truncation error convergence tolerance without relaxing the Newton iteration convergence tolerance.
Charge accuracy	ChargeTol	The minimum charge value used to determine the charge tolerance when computing the local truncation error. Default = 1e-14.
Integration	IntegMethod	
Trapezoidal	Trapezoidal	Default. Integrates between time points by assuming they are connected by line segments. The local truncation error is then related to the difference between the areas determined by the present and previous time points.
Gear's	Gear's	Integrates by assuming that the time points are connected by a polynomial curve. The order of the polynomial is controlled by the Max Gear order parameter. Lower-order polynomials tend to create greater truncation error, while higher-order polynomials can become unstable.
Max Gear order	MaxGearOrder	Determines the maximum order of the polynomial when Gear's method is used. The default is 2. This is available only when Gear's is selected.
Integration coefficient mu	Mu	A coefficient that determines the degree of mixing of the trapezoidal (mu = 0.5) and backward-Euler (mu = 0.0) methods when the trapezoidal method is used. This is available only when Trapezoidal is selected. The valid range for mu is: 0.0 <= mu <= 0.5. Caution: Do not set mu to 1.0; this results in a divide-by-zero condition. The integration order at each timestep is output to the dataset as the variable tranorder. This data is used by the fs() function in the data display server to do accurate interpolation of the data when an FFT is required. For the default trapezoidal integration, this will normally have a value of 2, except at source-induced breakpoints where it will be 1.

Setting Up Convolution Analysis

Following is information on setting up the Convolution portion of the simulation. The following table describes the parameter details. Names listed in the *Parameter Name* column are used in netlists and on schematics.

Note
It is recommended that the Convolution parameters are left at their default settings.

Transient Simulation Convolution Parameters

Setup Dialog Name	Parameter Name	Description
Tolerance - Sets the tolerance for relative and absolute truncation factors for the impulse response: ImpRelTrunc and ImpAbsTrunc. Values are set depending on the option selected for Tolerance. Auto is the default. ImpRelTrunc default value is 1e-4. ImpAbsTrunc default value is 1e-7, and it controls how small the impulse must be before it is considered zero.		

Relax		When selected, impulse response truncation factors are set to the following:
		ImpRelTrunc = 1e-2
		ImpAbsTrunc = 1e-5

Auto		Default setting. Default values are used for ImpRelTrunc=1e-4, and ImpAbsTrunc=1e-7.

Strict		When selected, impulse response truncation factors are set to the following:
		ImpRelTrunc = 1e-6
		ImpAbsTrunc = 1e-8

Enforce Passivity	ImpEnforcePassivity	Default setting is on (selected). When selected, this option enforces passivity for linear frequency domain components which are simulated using discrete mode convolution. Similarly, if EnforcePassivity=yes in a SnP component, passivity will be enforced in that particular device. The EnforcePassivity setting of SnP component overwrites the ImpEnforcePassivity setting of the transient controller in an individual device. Therefore, if a SnP component is active by design, EnforcePassivity should be disabled by setting EnforcePassivity=No in that paticular SnP component.
Advanced		Click Advanced on the Convolution tab to set these parameters. For parameter descriptions, see the following section *Defining Advanced Convolution Parameters*.

Defining Advanced Convolution Parameters

Use the following parameters for additional control of convolution simulations.

Advanced Convolution Options

Setup Dialog Name	Parameter Name	Description
Use approximate models when available	ImpApprox	Causes the simulator to bypass impulse-based convolution (when that option is available to you). Instead, it uses models that, although somewhat less accurate, can provide faster simulations. These approximations neglect effects such as frequency-dependent loss and dispersion, but include the basic delay and impedance. These models are the default, if no convolution license is available. Default setting is unselected.
Approximate short transmission lines	ShortTL_Delay	Specifies a limit on the time delay below which transmission lines will be approximated instead of modeled as a delay line. This enables you to analyze very short transmission lines with a Laplace transform approximation. Also, it does not require the simulator to take the very small time steps normally associated with short transmission lines. Only single, two terminal, transmission lines (like MLIN and TLIN, but not MCLIN and TLIN4) can be approximated this way.
Max Frequency	ImpMaxFreq	The maximum frequency to which a frequency-domain device is evaluated to obtain its impulse response. By default, the program chooses this value according to signal source bandwidth. You do not normally need to set this, unless it is necessary to increase it to model the high frequency components due to circuit nonlinearity.
Delta Frequency	ImpDeltaFreq	The frequency interval between samples in the evaluation of frequency-domain-defined devices.
Save Impulse Spectrum	ImpSaveSpectrum	Saves the impulse response, its FFT, and the original spectrum to a dataset when discrete mode convolution is used in transient analysis. The default value is no (unselected). The information added to the dataset uses names similar to those shown here where CMP1 is the component name: CMP1_FFT_IMP: FFT of final impulse response used in convolution. CMP1_IMP: Final impulse response used in convolution. CMP1_OR: Original spectrum. CMP1_S0: Exists only if passivity is enforced. Spectrum after causality but before passivity correction.
Enforce Strict Passivity	ImpEnforceStrictPassivity	Default is off (unselected). When selected, this option enforces strict passivity for linear frequency domain components in convolution. This option is suggested only when "Enforce passivity" is selected and convolution still fails to converge. Although this situation is rare, it may happen when a linear frequency domain component has a large number of ports. When selected, impulse response calculation is slower. This option takes effect only when regular "Enforce passivity" is selected.

Backward Compatibility for Convolution

Due to the improved convolution simulation algorithm in ADS 2008 the parameters listed in the following table that were available in previous releases are obsolete. Parameters for *Convolution Mode* and *Tolerance* will be handled in ADS 2008 as described here:

- ADS 2008 always uses *Discrete* mode for *Convolution Mode* (*ImpMode*) and the *PWL Continuous* mode is disabled. If a design from a previous release uses *PWL Continuous* mode, ADS 2008 automatically sets the parameter to *Discrete* mode.
- In designs created in a previous release, the values of parameters supported by ADS

2008 will be honored except *ImpRelTrunc* and *ImpAbsTrunc*. Regardless of their values in older designs, ADS 2008 will use the *Auto* mode for *Tolerance*. The user can change the *Tolerance* mode by selecting *Relax* or *Strict*.

Convolution Parameters Obsoleted in ADS 2008

Setup Dialog Name	Parameter Name
Max impulse sample points	ImpMaxPts
Convolution interpolation order	ImpInerpOrder
Convolution mode	ImpMode
Discrete	Discrete
PWL Continuous	PWL Continuous
Smoothing window type	ImpWindow
Rectangle	Rectangle
Hanning	Hanning
Non-causal fcn imp response length	ImpNoncausalLength

Setting Up the Convergence

Following is information on setting up the Tran Convergence portion of the simulation. The following table describes the parameter details. Names listed in the *Parameter Name* column are used in netlists and on schematics.

Transient Simulation Convergence Parameters

Setup Dialog Name	Parameter Name	Description
Use user-specified initial conditions	UseInitCond	Instructs the simulator to use user-specified initial conditions to compute the initial charges and fluxes in the circuit. The simulator will skip time=0 dc analysis. When this option is enabled, the simulator uses the currents specified in the InitCond field of the inductors and the branch voltages specified in the InitCond field of the capacitors, as well as the node voltages specified in both the Simulation-Transient InitCond and Simulation-Transient InitCondByName components to compute the initial charges and fluxes in the circuit. The voltage given in the capacitor's InitCond field will take precedence over the voltage given in the Simulation-Transient InitCond and Simulation-Transient InitCondByName components if there is a conflict. When the option is disabled and the Simulation-Transient InitCond and/or Simulation-Transient InitCondByName components are placed on the schematic, the simulator will start the transient analysis with a DC analysis at time=0 and try to force those node voltages identified in the Simulation-Transient InitCond and/or Simulation-Transient InitCondByName components to be close to the voltages specified. The values, if any, given in the InitCond field of the capacitors and inductors, however, will not be used.
Connect all nodes to GND via GMIN during initial DC analysis	LoadGminDC	Useful in conditions where open circuits would result in a singular matrix error (where the diagonal term is missing). When selected, this option instructs the simulator to insert a resistor valued at 1e12 ohms between any node and ground, whether at the circuit or the device level. ("GMIN" stands for minimum conductance.) This allows you to still obtain useful results.
Perform KCL check for convergence	CheckKCL	Checks to verify how closely Kirchoff's Current Law is satisfied at each node. This also depends on how the current tolerances are set (under the Convergence tab of the Options component). This option is selected by default, but may help convergence if not performed.
Check only delta voltage for convergence	CheckOnlyDeltaV	Looks for only the voltage difference between two consecutive iterations. This less-stringent check saves both time and memory. This option is selected by default.
Check for strange behavior at every timestep	OverloadAlert	Looks for inordinate device currents or voltages and returns a warning message when it finds any for devices or models which support the Overload Alert and have their limit values specified. Although this can be slower, it is useful where out-of-bounds conditions are suspected.
Skip device evaluation if volt chg are small between iters	DeviceBypass	Instructs the simulator to bypass the full evaluation of a device if it finds small voltage changes between iterations. This can improve simulation time in digital circuits containing many devices. "Small" voltage change is defined by voltage tolerances set under the Convergence tab of the Options component. This may increase analysis speed for digital circuits.
Max iterations per time step	MaxIters	The maximum number of iterations allowed for each time step.
Max iterations @ initial DC	MaxItersDC	The maximum number of iterations allowed during DC analysis, before the stepping of the source begins.
IV_RelTol	IV_RelTol	This is the transient relative voltage and current tolerance. The default is 1e-3. When simulation options are included in the simulation (using the Options controller), and there are other simulation controllers besides transient in an ADS design, use this parameter to set specific relative tolerances to be used for transient only. The value of *IV_RelTol* will be used for both current and voltage relative tolerance for transient. The *IV_RelTol* parameter makes it possible to have relative tolerances

		specifically for transient.

Setting Up Optional Parameters

Following is information on setting up the Tran Options portion of the simulation. The following table describes the parameter details. Names listed in the *Parameter Name* column are used in netlists and on schematics.

Transient Simulation Options

Setup Dialog Name	**Parameter Name**	**Description**
Levels		Select the degree of simulation information to be reported.
Status level	StatusLevel	Prints information about the simulation in the Status/Summary part of the Message Window. - 0 reports little or no information, depending on the simulation engine. - 1 and 2 yield more detail. - Use 3 and 4 sparingly since they increase process size and simulation times considerably. The type of information printed may include the sum of the current errors at each circuit node, whether convergence is achieved, resource usage, and where the dataset is saved. The amount and type of information depends on the status level value and the type of simulation.
Device operating point level	DevOpPtLevel	Enables you to save all the device operating-point information to the dataset. If this simulation performs more than one Transient analysis (from multiple Transient controllers), the device operating point data for all Transient analyses will be saved, not just the last one. Default setting is None.
None	None	No information is saved.
Brief	Brief	Saves device currents, power, and some linearized device parameters.
Detailed	Detailed	Saves the operating point values which include the device's currents, power, voltages, and linearized device parameters.
Output solutions		
Output all internal time points	OutputAllPoints	Causes the simulator to save simulation results at all internal timepoints; this option is on by default. Deselecting this option causes results to be saved at least as often as the Max timestep option but some of the intermediate points will be suppressed. For simulations that take many small timesteps due to automatic timestep control, but whose output is still well-sampled at Max timestep, this can make the resulting datasets smaller and make the post-processing of the data faster.

Defining the Noise Parameters

Following is information on setting up the Tran Noise portion of the simulation. The following table describes the parameter details. Names listed in the *Parameter Name* column are used in netlists and on schematics.

Transient Simulation Noise Parameters

Setup Dialog Name	**Parameter Name**	**Description**
Noise bandwidth	NoiseBandwidth	Enables the generation of pseudorandom noise at each timestep. NoiseBandwidth controls the bandwidth of the generated noise and must be less than or equal to 1/(2*MaxTimeStep). If this parameter is not specified or is zero, no noise is generated.
Noise scale	NoiseScale	A multiplicative scaling applied to all generated noise.

Setting the Fundamental Frequencies

Following is information on setting up the frequency portion of the simulation. The following table describes the parameter details. Names listed in the *Parameter Name* column are used in netlists and on schematics.

Transient Simulation Frequency Parameters

Setup Dialog Name	Parameter Name	Description
Fundamental Frequencies		
Edit		Edit the *Frequency* and *Order* fields, then click *Add* to add the frequency to the list in the Select area.
Frequency	Freq[n]	The frequency of the fundamental(s). Change value by typing over the entry in the field. Select the units (None, Hz, kHz, MHz, GHz) from the drop-down list.
Order	Order[n]	The maximum order (harmonic number) of the fundamental(s) that will be considered. Change value by typing over the entry in the field.
Select		Contains the list of fundamental frequencies and their orders. Use the Edit area to add fundamental frequencies to this window. - Add - Adds a frequency to the list. - Cut - Removes selected frequency from the list. - Paste - Enables you to move an item cut from the list to a new position.
Maximum mixing order	MaxOrder	Determines how many mixing products are to be transformed for the multi-tone harmonic balance simulation. A mixing term, or mixing product, is a combination of two or more fundamentals or their successive harmonics. Mixing products will occur when there are multiple frequencies in a circuit. For example, consider having a simulation with Freq[1]=f1, Order[1]=4, Freq[2]=f2, Order[2]=5, and MaxOrder=3. The mixing products that are to be transformed are f1-f2, 2f1-f2, f1-2f2, f1+f2, 2f1+f2, and f1+2f2. If Maximum order is 0 or 1, no mixing products are simulated. If the *MaxOrder* is not given, then it will be set to the largest order of the fundamental.
Compute HB Solution		
Write Initial Guess for HB	HB_Sol	Check this box to generate a transient initial guess for a harmonic balance simulation.
File	HB_OutFile	Enter the name of the file for the transient initial guess. Be sure to use the file name when running the harmonic balance simulation. If no file name is entered (and *Write Initial Guess for HB* has been selected), then the name of the design will be used as the default name.
Apply Window	HB_Window	Applies a window to the time domain data. This window helps to minimize the spectral leakage when multiple frequency tones are present. The window type is a Blackman window. For multi-tone applications, it is recommended to enable the *Apply Window* option.
Detect steady state	SteadyState	When enabled, causes the transient simulator to determine if the circuit reaches steady state. If steady state is reached, then the time value and frequency of oscillation (if simulating an oscillator) will be reported. At least one frequency and order pair (Freq[1] and Order[1]) must be specified when selecting this parameter. For a transient assisted harmonic balance simulation, enable SteadyState for the simulator to generate a transient initial guess (which captures the steady state portion of the waveform) for later use in the harmonic balance simulation. The simulator will stop when a steady state has been reached and transform just the last period of the solution. Thus, the transient simulation may end earlier than the *StopTime* when steady state is reached.

Troubleshooting a Transient-Convolution Simulation

This section presents suggestions for using this simulation tool and improving the accuracy of results.

Avoiding Simulation Errors in Transient Analysis

This section lists a variety of steps that can be taken to avoid errors in simulation.

1. Check the circuit's schematic diagram carefully, and turn on the topology checker if it has been turned off. Consider using DC_Block and DC_Feed components where applicable.
2. Check the parameter *Min time step*. This parameter sets the smallest time step that the simulator is allowed to take, and should be smaller than the fastest rise time in the circuit. The default value of *Min time step* is *Max time step*/10^{12}. (*Stop time* is the last time point in the simulation.) The default value should be satisfactory.
3. Check to verify that the absolute and relative current and voltage tolerances (in the Options component) are not too small. For initial Transient analysis, try to use I_RelTol = V_RelTol = $1e^{-3}$, and tighten these values only when higher accuracy is needed. The simulation will run much faster with these settings compared to $1e^{-6}$, and will relax Newton convergence tolerance as well as LTE tolerance. Try increasing I_AbsTol to $1e^{-10}$ instead of using the default $1e^{-12}$.
4. The models for some frequency-dependent devices have a high-frequency limit, beyond which they are not valid. Unless the very-high-frequency response of the device is important, the simulation results will still be valid.
5. If a lossy inductor model is included in the circuit, and the inductance has been set to zero, you may need to replace the lossy inductor with a resistor.
6. A lossy inductor model cannot be used with an initial condition. To solve this problem, replace the lossy inductor model with a lossless inductor in series with a resistor.
7. The simulator supports user-defined models that can have any impedance. However, it is easy for users to define nonphysical or noncausal components for which there is no correct answer. If a component has a constant reactance that does not vary with frequency (or has a nonzero reactance at DC), then the component is mathematically nonphysical. In these cases, the simulator will produce an answer that may not be physically realistic. To eliminate this problem, change the component's definition.
8. Sometimes, in the case of user-defined devices, the simulator cannot handle certain types of time dependencies with guaranteed accuracy. These devices often work correctly, but the simulation results should be checked carefully.
9. The simulator cannot support S-parameter ports with zero impedances. To use a source with a zero impedance, use a simple voltage source instead.
10. Transient analysis convergence problems are often caused by jumps in the solution. This most often occurs in circuits with overly simplified models that exhibit positive

feedback, or when the circuit contains nodes that do not have a capacitive path to ground. Add a small capacitor from the troublesome node to ground and give a complete capacitance model when specifying the nonlinear device model parameters.

11. Generally analog circuits are sensitive to truncation error due to their relative long time constants. Use LTE time step control to ensure the accuracy of the results. Also try relaxing TruncTol by increasing this value to 10 times or more to relax LTE tolerance.
12. Try different integration methods. Backward Euler (Gear1 or Mu=0 in Trapezoidal) and Gear2 are stable for all stable and some unstable differential equations. However, trapezoidal rule are stable only on stable differential equations. Switch to Gear1 or Gear2 when trapezoidal rule fails on unstable differential equations.
13. Add break points. Use piecewise linear source to add break points to the region where the waveform changes abruptly.

Solving Convergence Problems

Nonconvergence is a numerical problem encountered by the simulator when it cannot reach a solution, within a given tolerance, after a given number of iterations.

There is no single solution for solving convergence problems in transient and convolution analysis. Several ways to approach those problems are listed below.

- Make sure that you have specified an appropriate value for *Max time step*. Reduce this value if necessary to ensure enough time points for sharp edges.
- Vary *Integration coefficient mu* in the case of high-Q circuits.
- Increase the time-point iteration limit, *Max iterations per time step*. Increase this value to 50 or more to increase the possible number of Newton iterations on each time step.
- If the circuit includes a GaAsFET , reduce the value of the transit time *TAU* in the device model itself.
- Adjust the current and voltage relative and absolute tolerances in the Options component. Note the following guidelines:
  The relative tolerance parameters have a greater effect on simulation speed and accuracy than the absolute tolerance parameters.
  Each order-of-magnitude change in accuracy (for example, from 10^{-3} to 10^{-4}) will result in approximately three times as many time points in the simulation.

Typical Convergence Problems

If you can attribute nonconvergence to any of the following areas, try these tips:

Capacitor model problems:

- Use simplified device models that do not include capacitance model or incomplete capacitance model give a complete capacitance model when specifying nonlinear

device model parameters, in junction capacitance, include both depletion (at least) and diffusion capacitances.
- Discontinuous jumps in waveforms when circuit contains nodes have no capacitive path to ground add small capacitor to ground or specify Cmin.
- Capacitance model does not conserve charge GaAsFET Statz's, MOSFET Meyer's capacitance models switch to charge based model.
- Large floating capacitors that are similar to the small-floating resistor problem in DC (finite precision problem) check capacitance unit, use smaller capacitance.
- Discontinuous capacitance models in user defined model, SDD device fix the model.

Slow Transient analysis:

- Make sure I_RelTol and V_RelTol are set to 1e^{-3} or not set at all.
- Decrease these values when higher accuracy is needed.

Oscillator circuit does not oscillate:

- Apply a short pulse at the beginning of the simulation.
- Avoid using Gear2 or backward Euler.

Circuit exhibits ringing or divergence:

- Reduce Mu value from 0.5 toward 0 if trapezoidal rule is used.
- Use Gear1 or Gear2.

Circuit does not converge at first time point:

- Reduce Min time step.

Avoiding Simulation Errors in Convolution Analysis

This section lists a variety of tips that can be used to avoid errors in simulation.

Using Convolution:

- Don't set any convolution parameters (let the adaptive algorithm figure it out).
- Set ImpMaxFreq first (larger than signal bandwidth).
- Set convolution parameters on component, not controller, when possible.
- Don't allowed measured data to be extrapolated (either set ImpMaxFreq or provide more data).

Convolution Modeling for Time-Domain Simulation:

- In time-domain simulation, simulate devices that can only be defined in the frequency domain:
    - Transmission lines with dispersion
    - Devices with frequency-dependent loss

  - Measured frequency-domain data
- Convolution is the key
  - Inverse Fourier transform of frequency-domain data produces the impulse response h(t)
  - The impulse response is convolved with time-domain signal

Time and Frequency Range:

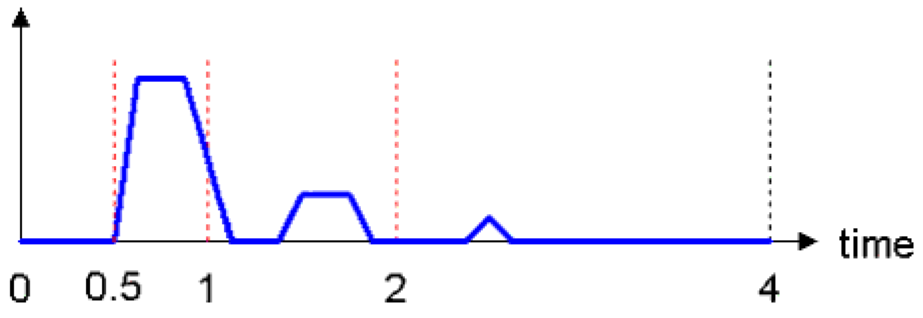
- Impulse response is computed from the inverse Fourier transform of frequency-domain response. Frequency is uniformly sampled from 0 to some upper value.
- Upper frequency sets the time-domain spacing of the impulse response
- Frequency spacing sets the length of the impulse response

Adaptive Impulse Response Calculation:

- Estimate of system bandwidth is made from source frequencies and rise times - initial guess at fmax.
- Build a trial impulse response with 32 timepoints - very coarse frequency spacing.
- Build a second impulse response with 64 timepoints - less coarse frequency spacing.
- Keep doubling the number of timepoints until a good impulse response is obtained - increase fmax, decrease Df.
- y11 and y12 may be sampled with different fmax and Df.
- Adaptive calculation is only done if ImpDeltaFreq is not specified - don't set ImpDeltaFreq if you don't have to.

Good Impulse Responses:

- Compare impulse responses with N and 2N points. The second impulse response is twice as long in time domain and has half the frequency spacing.
- An impulse is considered "good" when no appreciable energy is present in the second half of the impulse response.
- If energy is present in the second half, implies either that the impulse is not long enough or it is noncausal.
- If not good, Controller keeps doubling the length.
- Controller also tries doubling the maximum frequency, giving smaller impulse timesteps.



Interpolation:

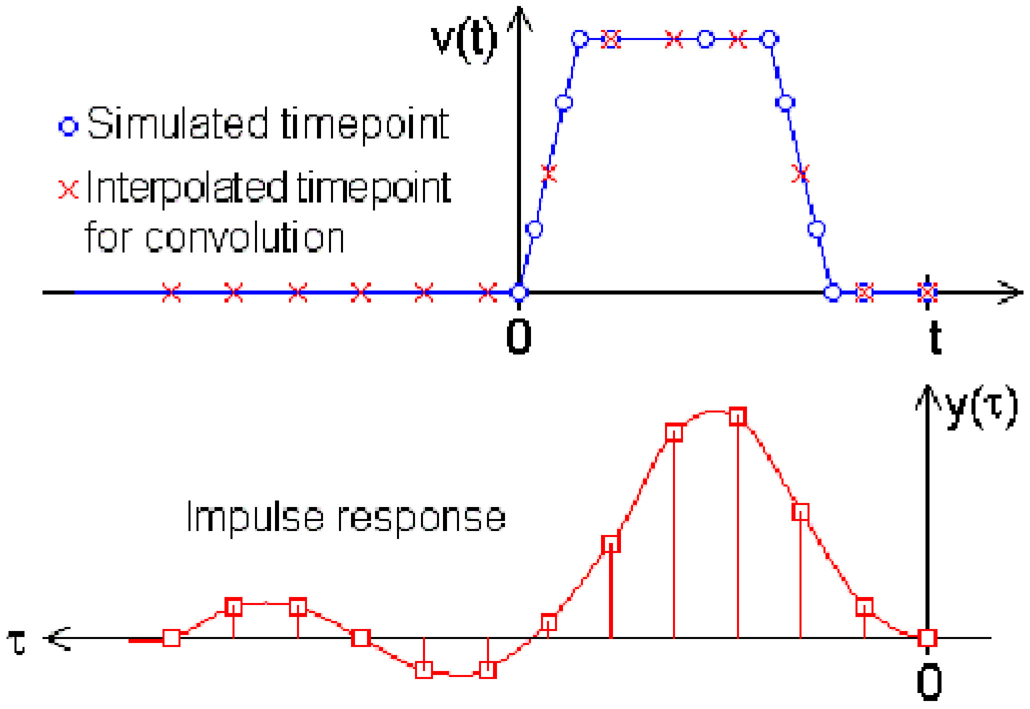
- The impulse response is sampled with a uniform timestep, but is not guaranteed to match the simulation timestep. The simulation may even be using a variable timestep.

- Interpolate the signal v(t) to match the timepoints in the impulse response.
- Don't interpolate the impulse response because the Fourier transform of the interpolated impulse response would no longer match the original frequency response.

Impulse Evaluation:

- Signal response at time zero extends back to minus infinity.
- Evaluate the integral as a sum.

$$i(t) = \sum_{0}^{n} v(t - \tau_n)\, y(\tau_n)$$



Solving an Invalid Impulse Response

This is the most commonly encountered problem during convolution. It does not necessarily imply noncausality but means that significant energy is present in the second half of the impulse response. In addition, simulation results may or may not be valid.

- Set ImpMaxFreq or ImpDeltaFreq. Set ImpMaxFreq first, typically only for measured data.
- For every component that generates this message, fix each component one at a time to simplify the design.

Viewing an Impulse Response

- In an S-parameter simulation, analyze over the given frequency spacing and maximum frequency - inverse Fourier transform the response by plotting ts(x) .
- In the time domain, apply an impulse and simulate plot the transient result - the pulse rise time is used to set fmax and thus can influence the impulse response.

Setting ImpMaxFreq and ImpDeltaFreq

Generally a good impulse response can be found without manually setting ImpMaxFreq and ImpDeltaFreq.

- If ImpMaxFreq is set, the adaptive algorithm tries different lengths but doesn't modify fmax.
- If ImpDeltaFreq is set, the adaptive algorithm is disabled and the impulse is computed from ImpDeltaFreq and ImpMaxFreq.
- Set ImpMaxFreq on the component, then set ImpDeltaFreq on component if necessary, and finally, set ImpMaxFreq on the transient controller if necessary.
- For transmission lines, set ImpMaxFreq to at least n/td, where td is the delay time and n is a small integer (2-3).
- For lowpass and bandpass filters, set ImpMaxFreq to at least twice the upper passband edge.

Measured Data with S2P Component

- The algorithm that computes the impulse response has no special knowledge of the component it's working on and assumes data is available at any desired frequency. It has no knowledge of:
  - f_{low} and f_{high} of measured data
  - frequency spacing of measured data
- S2P interpolates and extrapolates data as needed.
- Be sure to supply good data to prevent dangerous extrapolation extends down to DC and up to f_{max}.
- Set ImpMaxFreq on S2P component to match frequency limits in datafile (avoid extrapolation).
- Typically there is not enough frequency-domain data in the S2P file for use in the simulation.
  - Given a pulse with a rise time of tr, the equivalent bandwidth in Hertz is 0.35/tr (0.1 ns rise time represents a 3.5 GHz bandwidth).
  - Package models typically must be measured up to 10x higher than the signal frequency to represent transmission line effects well.

Solving a Noncausal Impulse Response

This is the second most commonly encountered problem during convolution. The Time-domain simulation starts at time zero and moves forward in time, computing the value of next timepoint from all previous timepoints. And the Controller deals with this by introducing a delay to force causality. Length of delay set to ImpNoncausalLength (default=32) with timestep set by default ImpMaxFreq.

Simulation results will not be accurate because of the added delay, especially if the delay is added in a critical timing or phase path.

All physically realizable devices are causal (the output is dependent only on past states and not any future states) while noncausal devices are nonphysical. Some ADS components, user-defined data or equations may be noncausal.

- Frequency-dependent real part with constant imaginary part, for example resistance as a function of frequency without any reactance.
- Constant real and constant non-zero imaginary part.
- Negative time delays.
- INDQ, CAPQ, PLCQ, SLCQ have problems in some modes.

Comparing Time-Domain and Steady-State Results

Certain circuit elements such as microstrip, discontinuities, and so on, are represented by simplified models in a transient analysis. A few planar discontinuities are also treated as ideal short or open circuits. Therefore, results from transient or convolution simulation may differ from those for steady-state simulation on the same circuit, depending upon the types of elements used in the circuit. (A convolution simulation should yield results closer to those of a steady-state simulation.)

The frequency of operation also plays a major role. At low frequencies, simplified models or short-circuits may be valid approximations for certain dispersive models.

Setting Max Frequency and Other Convolution Parameters

In general, the synthesized impulse response accurately represents the frequency domain function over the frequency range specified by the convolution control parameter *Max Frequency*. The simulation techniques require negligible spectral energy outside this frequency range. An accurate solution is not guaranteed if this principle is not obeyed. Therefore, setting *Max Frequency* correctly is the most important aspect of any transient simulation using convolution-based devices.

Max Frequency is similar to choosing the number of harmonics in a harmonic balance simulation or a time step in SPICE. In these cases you must estimate the value of the parameter prior to the simulation. Examination of the results reveals whether the parameter was chosen correctly. The built-in estimation of *Max Frequency* is always a good starting point.

The remaining convolution-control parameters described below are best used with their default values for any causal device, such as a microstrip transmission line.

The *Discrete* convolution mode option, by causing a periodic extension of the frequency response, generally leads to the most accurate and efficient description of the device from DC to *Max Frequency*. Provided *Max Frequency* is set correctly, the frequency response beyond *Max Frequency* is irrelevant, because no significant spectral energy exists beyond this value. The *PWL Continuous* option always leads to a low-pass response (which may be desirable in cases where a low-pass response is being modeled). The discrete mode is many times faster than the continuous mode.

Delta frequency defines the frequency spacing with which the convolution-based devices are sampled in the frequency domain. If *Delta frequency* is not specified, the simulator adaptively samples the frequency function until an appropriate value is determined. If you are unsure about the correct value of *Delta frequency*, simply allow the simulator to decide.

Non-causal fcn imp response length adjusts the length of the impulse response associated with the treatment of noncausal frequency responses (see discussion below).

Smoothing window type specifies the smoothing window to be applied to the time-domain impulse responses that are derived from noncausal frequency functions (such as Hilbert transforms). The window reduces ripple in the approximation caused by discontinuities in the frequency function. (Refer to *Dealing with Noncausal Frequency Responses".* (cktsimtrans))

Max impulse sample points places an upper limit on the allowed impulse-response length. It is mainly used when *Max Frequency* is specified but *Delta Frequency* is not. It is necessary to increase this value (default = 4096) if you specify a frequency response that requires a long impulse response.

Relative impulse truncation factor and **Absolute impulse truncation factor** are used to remove redundancy from the impulse responses. Setting them to a small value leads to longer impulse responses and a more accurate description of the frequency response of a convolution-based device. The simulator uses them to make decisions about the relative sizes of individual members of an impulse response.

Dealing with Noncausal Frequency Responses

A causal system has the property that if the input is zero for $t < t_1$, then the output is also zero for $t < t_1$. The value of t_1 is usually defined to be 0. Therefore, the impulse response of a causal system is zero for negative time. This may be stated in the frequency domain by saying that the real and imaginary parts of a causal frequency response are related by the Hilbert transform: the Kramers-Kronig relation. Only causal frequency-domain transfer functions can be handled directly by means of transient simulation methods.

However, there are many ideal frequency functions that are noncausal but are extremely valuable in processing signals (such as Hilbert transforms). The transient simulator provides a way to use these functions in defining the frequency response of a nonlinear device.

A frequency response is considered noncausal if it has one of the following forms:

R + jI
R(w) + j0
R + jI(w)

When you select *Non-causal fcn imp response length*, non-causal frequency functions are approximated by a digital filter that introduces a time delay in the impulse response. This time delay is sufficient to make the apparent impulse response causal. Non-causal fcn imp response length defines the length of this filter. The *Smoothing window type* can be selected to reduce Gibb's phenomenon, which may be present if the non-causal frequency response contains discontinuities. Increasing non-causal leads to a more accurate description of the frequency function over the chosen band set by Max Frequency. However, increasing non-causal causes a longer delay through the filter and a longer simulation time. Only even values of non-causal are allowed. If an odd value is specified, the simulator adds 1 and issues a warning. The delay associated with the causal approximation is given by the equation: (noncausal/2-1)/(2*max Frequency).

If a non-causal frequency response is specified which does not fit into one of the forms described above, the simulator will assume it is a causal function. This will almost certainly cause unexpected and erroneous results. Noncausal frequency responses should be avoided if possible.

The concept of causality is not an issue when using the harmonic balance simulator. All excitations and responses are sums of sinusoids and exist for all time. For this reason, any bounded frequency response can be simulated. When moving from harmonic balance to transient simulation, it must be remembered that non-causal frequency functions cannot be simulated directly. Only a band-limited approximation to these functions can be simulated. This is particularly important when using SDD weighting functions.

Using Measured and Simulated S-Parameter Data

The ability to handle convolution-based devices allows the user of measured or simulated S-parameter data to describe a wide variety of devices and circuits. A dataset or file containing the S-parameter values can be used to integrate the frequency response into a time-domain simulation. This adds a tremendous amount of flexibility to the number and types of devices and circuits which can be used in a simulation.

When S-parameter data is used, it is important that the frequency response be adequately sampled over the entire bandwidth to ensure negligible interpolation errors when the impulse response is being calculated. The Max Frequency parameter should never be set to a value which is greater that the maximum S-parameter data frequency. doing so will lead to erroneous results as the available data would have to be extrapolated. S-parameter data must also extend all the way down to DC.