Appendix

RF Transient Simulator
RF Transient/Convolution Simulation

This simulator can be used to solve problems associated with circuit simulation, when the signal and waveforms involved are modulated with complex signals. Such signals represent the primary types of signals found in modern RF communication systems. The most traditional simulation solutions are based on SPICE or SPICE-like time domain algorithms. 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. These simulation techniques assume that the input stimulus can be an arbitrary baseband signal so the solution \( v(t) \) must also be assumed to be a baseband signal. This means that any high frequency carrier must be represented as baseband signals and must therefore be sampled at rates significantly higher than the most significant harmonic frequency. For example, assume a 5GHz signal with significant 3rd harmonic. To satisfy the basic Nyquist criteria, the sampling frequency must be greater than 30GHz. To obtain reasonable accuracy from the integration algorithm, a sampling frequency of 100GHz is a more realistic value. Now assume that the carrier is modulated with a symbol rate of 100 kHz, and that we want to simulate the circuit for 500 symbols. This means that we desire a total simulation time of 5 ms. However, the high carrier frequency dictates the step size that must be at least 10 ps or smaller. This means then that the circuit simulator must solve the circuit equations and outputs solutions at over 500 million time points.

**Features of Transient Analysis**

- Analyze low and high frequencies, linear and non-linear large circuits in the time domain
- Verify transient behavior like start-up time in oscillators, step-function responses in filters, pulsed RF network responses, high speed digital and switching circuits and more.
- Improved convergence for large and highly non-linear circuits.
- Time-to-frequency domain transformation allows RF designers to view the output results (such as IP3) in the frequency domain.

The main difference between the transient and convolution option lies in how each analysis characterizes the distributed and frequency-dependent elements of a circuit.
High-Frequency Spice Analysis

A High-Frequency-Spice-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.

Features of High-Frequency SPICE

- Directly uses high-frequency time-models for microstrip lines, bends, gaps, etc.
- Used to analyze the steady state response of mixers, oscillators, amplifiers and so on.
- Frequency-dependent elements are modeled with approximations that neglect some of the frequency-dependent effects, such as dispersion and high frequency loss, resulting in faster simulations – this is accurate enough for electrically small components (ICs).

Convolution Analysis

A convolution analysis, on the other hand, represents all the distributed elements in the frequency domain and hence accounts for their frequency dependent behavior. The characterization of many 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 the 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 the impulse responses.

Features of Convolution Simulation

- For more accurate analysis of the discontinuities and for more accurate frequency-domain models with dispersion effects and high frequency loss.
- Convolution simulator results in accurate high frequency results at the expense of longer simulation times.
- Handles circuits that contain distributed elements and S-Parameter data used for components.
- It can accurately analyze circuit startup and transient conditions at low- and high frequencies, where the effects of dispersion and discontinuities are significant (for electrically big components on PCB-boards).
- Extension of High-Frequency-SPICE. Basic time step and convergence algorithms and non-linear models are the same; only modeling of Frequency-dependent linear devices is enhanced.
High-Frequency Spice versus Convolution

- Always check the settings for convolution. To use only High-Frequency Spice, you must turn off (Advanced button) the *Use approximate models* setting in the transient controller.

- Almost every component in ADS has a non-convolution representation for High-Frequency Spice use. The primary exceptions are S-parameter based elements where the user either enters the S-parameter directly or reads them from a file. During High-Frequency Spice simulation, those components are simulated using just their DC response. For this reason, all S-parameter based elements should have their DC response correctly defined.

- In general, convolution simulation results in more accuracy at the expense of longer simulation time, depending upon how detailed the impulse response may be.

- Convolution is only performed on linear and time invariant elements. You don’t use nor need convolution on bias-dependent transistors and other non-linear elements.

- Convolution is a modeling issue. The real issue regarding simulation time is the ratio of convolution elements to non-linear elements. If non-linear devices greatly outnumber the convolution elements, then you’ll never notice the time needed for convolution.

How the High-Frequency-Spice and Convolution Simulator Operate

1. The user specifies a time-sweep range,

   ![Time Setup](image)

   Tolerances and iteration limits:

   2. A DC analysis is performed to determine the system solution at time zero.

   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 at 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 value 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 the new time step is acceptable, the analysis is repeated from step 4. If the Integration coefficient \( \mu = 0 \), backward-Euler numerical integration is used. Otherwise, trapezoidal 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 over-est factor. 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.

**Integration Method**

Like SPICE, this simulator uses the trapezoidal integration method as the default method for calculating derivatives at each time step \( t \) in the simulation. For most circuits, this method will succeed. Trapezoidal integrates between time points by assuming line segments connect them. The local truncation error is then related to the difference between the areas determined by the present and previous time point.

\[
X'_{n+1} = \frac{2}{\Delta t} (X_{n+1} - X_n) - f_n
\]

For those few 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.
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.

For most circuits, Gear’s method is no more accurate than the trapezoidal integration technique. However, if a circuit analysis fails to converge, Gear’s method may succeed where trapezoidal integration fails. If Max Gear order is set to a number between 2 and 6, the simulator will use Gear’s method along with an adaptive step size 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.

Memory Requirements and Simulation Time

With large circuit transient simulation, memory requirement grows linearly with the number of transistors in the circuit. ADS use roughly one megabyte per 100 transistors.

Simulation time grows superlinearly with the number of transistors; N. Time of simulation grows $N^{1.2}$. Therefore, if you double the transistors in a circuit, it will take 2.3 times the original simulation time.

Example:

If we have 1000 nodes in a circuit, the full matrix size would be 1000 * 1000 = 1,000,000 entries. But this full matrix is presupposed on the fact, that every node is connected to every other node. In any practical electrical circuit, any given node is normally connected to just a handful of other nodes; there is no connection to all of the other nodes. Sparse matrix techniques allow taking advantage of this fact and we only have to keep track of those nodes that are actually connected to other nodes. This allows us to effectively store a matrix with 1000 nodes using only 10,000 entries (1% of the full size). This makes the simulator’s use of memory much more efficient. The $N^{1.2}$ factor is due to the way the sparse matrix techniques work in solving for the solution.

Parasitic extraction of interconnects resistance and capacitance from an IC layout of a typical large circuit with N non-linear devices will have 5-10 times N passive components. Therefore a rule of thumb is that extraction from the layout generates about 10 passives (RC) per transistor. This is after model reduction of the passives. Without model reduction, an extraction yields 1000 passives (RC) per transistor. Model reduction deduces this by 100 times.

When combining digital and analog circuits on one chip, parasitic and substrate coupling becomes an issue. In this case we are talking about millions of transistors on one chip. Any Spice-type time domain simulator runs out of steam past 100,000 transistor point. If we model the substrate coupling, every transistor will have about 10 passive parasitic. Therefore it is still a major and unresolved problem, on how to simulate and verify the whole chip using SPICE tool.
Convolution Tab Settings

Recent improvements to the ADS Transient simulator have greatly improved the results of Convolutions analysis. For that reason, some previous settings are now obsolete and it is recommended that the tolerance settings be left in Auto mode. Here are the impulse response truncation factors for the three settings:

Relax: ImpRelTrunc = 1e-2   ImpAbsTrunc = 1e-5
Auto:   ImpRelTrunc = 1e-4   ImpAbsTrunc = 1e-7
Strict: ImpRelTrunc = 1e-6   ImpAbsTrunc = 1e-8

Enforcing passivity - This can be turned on 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.

NOTE designs from a previous release uses PWL Continuous mode, ADS 2008 automatically sets the parameter to Discrete mode.

Approximate models – this selection 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.

Save impulse spectrum - saves the impulse response, its FFT, and the original spectrum to a dataset when discrete mode convolution is used in transient analysis.

NOTE on Noncausal Frequency Responses: In ADS 2008, the simulator produces warnings when non-causality is detected and it also attempts to eliminate the problem. In general, the supports user-defined models that can have any impedance, including 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 may produce an answer that may not be physically realistic. To eliminate this problem, change the component's definition.

Refer to the documentation (manuals) for more details Transient analysis and settings.
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 that 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 calculated. The Max Frequency parameter should never be set to a value that is greater then 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.