Xyce™ Parallel Electronic Simulator Version 7.3 Release Notes

Sandia National Laboratories

May 5, 2021

The Xyce™ Parallel Electronic Simulator has been written to support the simulation needs of Sandia National Laboratories' electrical designers. Xyce™ is a SPICE-compatible simulator with the ability to solve extremely large circuit problems on large-scale parallel computing platforms, but also includes support for most popular parallel and serial computers.

For up-to-date information not available at the time these notes were produced, please visit the Xyce™ web page at http://xyce.sandia.gov.

Contents

New Features and Enhancements 2

Interface Changes in this Release 6

Defects Fixed in this Release 7

Known Defects and Workarounds 12

Supported Platforms 19

Xyce Release 7.3 Documentation 19

External User Resources 20

Sandia National Laboratories

New Features and Enhancements

New Devices and Device Model Improvements

- The lossless transmission line history interpolation has been improved to better capture discontinuous signals such as those produced by pulsed voltage sources. Various issues related to discontinuity detection and breakpointing were also fixed in this release.
- An ideal delay device has been added. Designated the YDELAY device, it operates in a manner similar to a voltage-controlled voltage source, in that The voltage between the positive and negative control nodes is reproduced at the positive and negative output nodes delayed by a specified time.

Enhanced Solver Stability, Performance and Features

- Increased parallel stability and robustness for simulation algorithms and models on increasing numbers of processors.
- Improved handling of dynamically inserted break points from devices that use history information, such as delay device. This resolved various accuracy issues, discontinuous behavior and convergence issues.
- Xyce now supports non-intrusive Polynomial Chaos Expansion (PCE) methods as an augmentation to the `.SAMPLING` and `.EMBEDDEDSAMPLING` analyses. This includes both the regression and projection forms of non-intrusive PCE. These PCE capabilities allow uncertainty to be accurately propagated thru a Xyce calculation with a relatively small number of sample points.
- Xyce now supports fully intrusive Polynomial Chaos Expansion (PCE) methods using the new `.PCE` command. This is mostly an experimental method, and is an alternative to the non-intrusive forms of PCE that are also available in Xyce.
- The Xyce expression library now supports a fast piecewise linear table, `FASTTABLE`. This feature is the same as `TABLE`, except that it only has time integration breakpoints at the first and last time point. For very large tables this can result in a significant speedup.
- The Xyce expression library now supports a variety of higher-order interpolators including `SPLINE`, `AKIMA`, `WODICKA` and `CUBIC` splines. Another supported interpolator is `BLI` for Barycentric Lagrange Interpolation. All of these interpolators use the same syntax as `TABLE` and `TABLEFILE`.

Interface Improvements

- Support for `.FFT` and `.MEASURE FFT` has been added for `.TRAN` analyses.
- The ? wildcard character (meaning “any single character”) is now supported on .PRINT lines for all analysis modes.
- The Xyce expression library now recognizes `TEMPER` and `HERTZ` as synonyms for `TEMP` and `FREQ`, respectively.
- The Xyce expression library now recognizes the operators `nint()`, `fmod()` and `db()`.

- The default behavior of the random expression operators GAUSS, AGAUSS, RAND, UNIF and AUNIF has been changed so that they return the mean of the operator. They now only return random values if one the various uncertainty quantification (UQ) analyses options (.SAMPLING, .EMBEDDEDSAMPLING or .PCE) is specified in the netlist.
- Model binning is now turned on by default. It is not necessary to specify the netlist command .options parser model_binning=true anymore.
- Xyce's rawfile output when invoked with the "-r" option was unnecessarily changing the final "_" in any variable name to a "#". It no longer does so. This bug was never present in rawfiles produced via ".print FORMAT=RAW" lines, only the command-line version that causes all solution variables to be extracted into the rawfile.
- .param parameters are now allowed to depend on "special" variables such as TEMP, TIME, FREQ and VT. Previously, only .global_param parameters could depend on special variables.
- .param parameters are now allowed to depend on .global_param parameters.
- .param parameters in the top level scope can now be modified by commands such as .STEP, .DC, .SAMPLING, .EMBEDDEDSAMPLING, .PCE and LOCA. Previously, only .global_param parameters could be modified by these commands.
- Expressions on the right-hand-side of .param and .global_param statements no longer need to be surrounded by curly braces or single quotes.
- The search-path algorithm for files given on .INC and .LIB lines has been improved. It should be more compatible with both commercial and open-source Process Design Kits (PDKs) now.
- Objective functions for .AC sensitivities can now be specified as expressions using the acobjfunc parameter.

XDM

- Fixed bug where relative path names to files in .lib statements were incorrect.
- XDM allows the undocumented PSpice syntax of having trailing commas in entries of "TABLE" statements.
- if/else/endif statements in HSPICE are now commented out since Xyce doesn't currently support this.
- Node names containing brackets, as seen in HSPICE, are now allowed.
- Forward slashes in HSPICE device and subckt names are now allowed.
- XDM now translates transient Spectre "sine" specifications.
- XDM comments out Spectre dc analysis statements with no parameters as it is not clear what the Xyce equivalent should be.
- XDM can now identify Spectre device instantiations of models defined in SPICE format.
- XDM removes Spectre "trise" parameter for R and C devices since this isn't available in Xyce.

- Fixed bug in Spectre translation where sometimes the model name would not appear for a instantiation of a device of that model.
- Fixed XDM crash when processing some Spectre subcircuit definitions.
- Fixed bug in translation of Spectre "include" statements that have the parameter "section". These will now be translated into ".LIB" statements in Xyce.
- Added support for translation of Spectre "port" device instantiations. Not all instance parameters are translated at this time.
- Translate the "M" unit prefix (mega) in Spectre "AC" statements to the Xyce equivalent "X".
- XDM comments out Spectre dc analysis statements with no parameters as it is not clear what the Xyce equivalent should be.
- Fixed bug where XDM aborted when translating Spectre "include" files with paths containing hyphens.

Xyce/ADMS Improvements

- A long-standing, obscure bug in ADMS's handling of "for" and "while" loops was fixed in Xyce/ADMS's version of ADMS implicit rules. This bug was only tripped when the condition for these loops was in some way operating point dependent, an unusual usage heretofore unseen in standard Verilog-A models.
- The `buildxyceplugin` script will now define a Xyce-specific symbol "__XYCE__" when it runs ADMS to generate C++ code. This symbol may be used in ifdefs in the Verilog-A source code to isolate Xyce-specific Verilog-A code.

Important Announcements

- The CMake build system is now available in addition to the Autotools build system. See the `INSTALL.md` file for the CMake build instructions.
- To enable improved performance, Xyce now requires a C++11 capable compiler.
- The model interpolation technique described in the Xyce Reference Guide in section 2.1.18 has been marked as deprecated, and will be removed in a future release of Xyce.
- Some distributions of Linux have broken builds of OpenMPI in their package repositories. Building Xyce from source code in parallel with these OpenMPI installs will result in a version of Xyce that may crash on some problems. This is not a bug in Xyce, but a packaging error of the OpenMPI package on those operating systems. Please see commentary in the "Known Defects" section of these release notes under bug number "967-SON".
- Support for the deprecated conversion of quoted-sting file names to a table of data has been removed. The supported method for reading a data file into a table is to use the `tablefile("")` keyword. Inside of expressions, `table("")` also works. For example, to set the parameter `PARAMETER` to a table of data read from a file named `"file.dat"` one can use either `PARAMETER={tablefile("file.dat")}` or `PARAMETER={table("file.dat")}`. Using `PARAMETER="file.dat"`, the old convention, will set `PARAMETER` to the string value `"file.dat"`. See the Xyce Reference Guide under `.GLOBAL_PARAM`,

`.PARAM` and Expressions for more details. Device models that can take tables of data as parameters are documented under the appropriate device.

Interface Changes in this Release

Table 1: Changes to netlist specification since the last release.

Change	Detail
Several unused nonlinear solver options removed	NORMLVL, DLSDEBUG, LINOPT, MEMORY, and constraint backtracking parameters (CONSTRAINTBT/ CONSTRAINTMAX/ CONSTRAINTMIN/ CONSTRAINTCHANGE) have been removed from the set of nonlinear solver options, since they are effectively unused.

Defects Fixed in this Release

Table 2: Fixed Defects. Note that we have multiple issue tracking systems for Sandia users. SON, which bugzilla on the open network, and SRN, which is bugzilla on the restricted network. We are also transitioning from bugzilla to gitlab issue tracking. Further, some issues are reported by open source users on GitHub and these issues may be tracked using multiple issue numbers.

Defect	Description
Gitlab-ex issue 8: Xyce "Override" rawfile output unconditionally replaces final "_" in a variable name with "#":	A long-standing bug in Xyce's "override" rawfile output (that produced when the command line option "-r" is used) caused all variables with underscores in their names to have the last underscore ("_") replaced with an octothorpe ("#"). This was originally done to allow Xyce's rawfiles to be readable by IC-CAP, which expects the branch current associated with a voltage source to have a "#branch" suffix, whereas Xyce's internal name for these currents have a "_branch" suffix. This bug made rawfile output names of other variables incorrect, so the code has been refined to make only appropriate changes to variable names rather than blindly replacing one character.
Gitlab-ex issue 191: Exclamation point character is not allowed in parameter names, in the new expression library	This was supported in previous versions of Xyce but was inadvertantly broken with the new expression library for Xyce 7.2. This mainly affected printing of "Y" device parameters on the `.PRINT` line, but only inside of `.PRINT` line expressions. This was due to a minor parsing mistake and has been addressed.
Gitlab-ex issue 157/Gitlab-ex issue 167/SON BUG 1190: Parameters that are defined using `.param` must be constant and non-constant parameters must be defined using `.global_param`	Xyce was originally designed with two keywords for setting user-defined parameters in the nelist; `.param` and `.global_param`. The keyword `.param` is also used in many other simulators, but the behavior in Xyce differed in that `.param` parameters were unconditionally treated as constants. This was because they were handled as string substitutions during parsing. Xyce parameters that needed to change during a calculation had to be set using the other keyword, `.global_param`. This design caused incompatibilities with other simulators, and proved confusing to users. However, it became possible to address this issue with the advent of a new expression library in Xyce 7.2 . Starting in this release, all `.param` parameters in Xyce are allowed to change during a simulation. This means they are allowed to depend on special variables such as `TIME` and are allowed to depend on `.global_param` parameters. Also, `.param` parameters that are defined in the top level circuit netlist are functionally identical to `.global_param` parameters, and can be swept using analysis statements such as `.DC`, `.STEP` and `.SAMPLING`.

Table 2: Fixed Defects. Note that we have two multiple issue tracking systems for Sandia Users. SON and SRN refer to our legacy open- and restricted-network Bugzilla system, and Gitlab refers to issues in our gitlab repositories.

Defect	Description
Gitlab-ex issue 150/Github issue 19: Error Manager does not play well with repeatedly declared simulator API objects	If a user code links to the Xyce libraries and allocates objects of one of the Xyce simulator classes (either the base class Xyce::Circuit::Simulator class or any of its derived classes) more than once, the error manager's use of static data to keep count of errors caused any failing simulation to cause all subsequent simulations to fail. Now, each simulator object clears the error manager's error counts upon construction. This fix does not eliminate problems related to the use of static data in the error handling package. Specifically, if multiple simulator objects were run simultaneously in different threads then any fatal error in any object's simulation would cause all to fail. At this time only sequential allocation and operation of simulator objects in user code is supported.
Gitlab-ex issue 153/Github issue 21: External Output Interface does not prepend TIME	The documentation of the external output interface in an application note claimed that if a user created an external output object that did not include output of "TIME" for a transient run or "FREQ" for a frequency-domain run, that the outputter would automatically prepend that variable to the list of outputs actually delivered. That was not happening due to a simple coding error. Now the interface performs that operation as intended.
Gitlab-ex issue 143: Fix transmission line breakpoint/discontinuity detection	When the transmission line was modified in Xyce 7.2 to stop forcing a maximum timestep equal to the transmission line delay, bugs in the handling of discontinuity detection and dynamically inserted breakpoints were exposed. As a result, transmission lines with very short time delays could cause Xyce to detect a discontinuity at one end of the line, but fail to set a breakpoint to catch the discontinuity at the other end before the time integration had already advanced too far. The result of this combination of bugs could lead to strange interpolation artifacts near the times when breakpoints should have occurred. The bugs in the time integration package's timestep selection and in the nonlinear solver have been addressed, and circuits involving transmission lines with very short delays will no longer miss breakpoints in this manner.
Gitlab-ex issue 82: Make AC sensitivities work in parallel	The initial implementation of AC parameter sensitivities did not work in parallel, but this now works correctly.
Gitlab-ex issue 115: Fix AC sensitivities to be correct for nonlinear device model parameters	The original implementation of AC sensitivities had a math mistake that caused the calculation to be incorrect when applied to parameters from nonlinear devices. The math error has been fixed and now nonlinear AC sensitivies work correctly.
Gitlab-ex issue 119: using `.sens` with `.four` can result in an indexing mistake	It was observed that when using .sens with `.four`, it was possible under some circumstances that the order of outputs got mixed up. In other words, some outputs in the `*.four` file would be given the wrong label. This was due to an indexing mistake and has been fixed.

Table 2: Fixed Defects. Note that we have two multiple issue tracking systems for Sandia Users. SON and SRN refer to our legacy open- and restricted-network Bugzilla system, and Gitlab refers to issues in our gitlab repositories.

Defect	Description
Gitlab-ex issue 123: POLY in the new expression library has problems at higer orders (>2)	The new expression library had mistakes in the POLY operator for higher-order polynomials, which have been corrected.
Gitlab-ex issue 99: Fix build flags associated with Trilinos	Xyce is able to leverage some features of Trilinos in a beta mode. Some of these were not enabled in the CMake build system, or were not functioning correctly. This issue was focused on making those work properly. Specifically, it was targeted at the inclusion of Amesos2, Stokhos, ShyLU, the Amesos2 Basker (a templated linear solver), and the ShyLU Basker (a multi-threaded linear solver). It also made Belos a required Trilinos package.
Gitlab-ex issue 108: Fix various problems in the limit operator in the new expression library	The "limit" function in the new expression library had mistakes in both the derivative calculation and also in the breakpointing function. Both issues have been fixed.
Gitlab-ex issue 113: Make flex/bison a required capability	Prior to the rewrite of the expression library, flex and bison were optional; however, now they are required. The autotools and CMake build systems have now been modified to reflect that. The ability to turn off the reaction parser capability with a flag has been retained, however (with the default being "on").
Gitlab-ex issue 128: Fix Xyce/ADMS handling of some while and for loops	Xyce/ADMS was emitting improper code for "for" or "while" loops that contained what ADMS flagged as "nonlinear" conditions. This turned out to be due to an error in the implicit rules of ADMS that has been present for at least 10 years, in which this unusual type of condition resulted in an inappropriate double call to an early dependency-tracking template. The result was that contents of the code block controlled by the loop construct was in backwards order, and all function precomputation was happening twice (due to the presence of duplicate entries in the "function" data structure). Xyce/ADMS uses a modified version of ADMS's implicit rules file instead of the one that ships with ADMS, and this error has now been fixed in Xyce's modified version. No standard Verilog-A models make use of "for" or "while" loops that would have tripped this bug, but such a usage was found in a Verilog-A model inside a proprietary vendor PDK.
Gitlab-ex issue 103: Fix glitches in transmission line at breakpoints	The lossless transmission line is an ideal behavioral device that operates by keeping a record of the histories of the two ends of the line, and using interpolation on that history to look up prior behavior at a time in the past in order to determine the current behavior. This interpolation is generally done using three-point quadratic Lagrange interpolation. It has been determined that Xyce has historically been producing small "glitches" at the corners of pulsed signals because quadratic interpolation is not really appropriate with piecewise linear signals at the corners where the slope changes signficantly. Xyce now detects when such discontinuities in derivative occur and performs linear interpolation instead. This prevents the small overshoots that have been observed in these cases in all prior versions of Xyce.

Table 2: Fixed Defects. Note that we have two multiple issue tracking systems for Sandia Users. SON and SRN refer to our legacy open- and restricted-network Bugzilla system, and Gitlab refers to issues in our gitlab repositories.

Defect	Description
Gitlab-ex issue 137: Address robustness issues in parallel restarting (BUG_456)	Parallel regression testing illustrated failures in restarting. The parallel hang can be tracked down to the restoration of restart data and bad logic for distributing the RestartNode objects.
Gitlab-ex issue 149: The BUG_507_SON test is failing on 4-processors in parallel	This weekly regression test failure on 4-processors was due to recent changes in the parallel assignment of shared solution nodes. A balanced assignment strategy was implemented to address the issue.
Gitlab-ex Issue 101, SON BUG 274: Remove deprecated parsing of quoted filename to table	Double quoted strings were once parsed as filenames into a lookup table. That syntax has been deprecated for some time and has now been removed. Use the `tablefile(''filename'')` syntax to read a file into a table. A double quoted string is now only interpreted as a string. Additionally, the syntax `string('''')` has also been removed as it was used to signify a string.
Gitlab-ex Issue 27: Clear out C++11 conditionals	To enable improved performance, Xyce now requires a C++11 capable compiler. Therefore, all the code conditionals related to C++11 features have been removed from the code.
1085-SON: Expression library mishandled `.FUNC` definitions of functions that began with "I" and were two characters long	Xyce's expression library previously assumed that all terms of the form "`Ix(<arguments>)`" are lead current expressions, where "x" is either a lead designator such as "D", "G" , or "S" for a MOSFET or "C", "B", "E" for a BJT, or a digit indicating the pin number of the device associated with the lead. This assumption made it impossible for users to define a function with a two-character name starts with "I".

1203-SON: Issues with temperature-dependent device instance-parameters that were expression-based	The values for temperature-dependent device instance parameters could be incorrect if their expression used a global parameter. This could occur if the default temperature of 27C was used. It could also occur if the temperature was set with a `.OPTIONS DEVICE` line or via `.STEP`. A simple example is:
	
	`.GLOBAL_PARAM GTEMP=37`
	`R1 1 0 1  TC=0.1 temp={GTEMP}`
	
	If the temperature was set with a `.OPTIONS DEVICE` line then the use of a "normal" parameter in their expression could also give an incorrect answer. An example is:
	
	`.PARAM ptemp=10`
	`R2 2 0 1  TC=0.1 temp={temp+ptemp}`

1239-SON: Address failures when parallel regression testing uses up to 4 processors	Parallel regression testing illustrated failures when used on more than 2 processors . These failures were caused by test and code issues and have been addressed for testing up to 4 processors.

Table 2: Fixed Defects. Note that we have two multiple issue tracking systems for Sandia Users. SON and SRN refer to our legacy open- and restricted-network Bugzilla system, and Gitlab refers to issues in our gitlab repositories.

Defect	Description
1241-SON: Expression library parsing bottleneck on large expressions	It was previously determined that the expression library in Xyce could be the source of a severe parsing bottleneck when expressions are large and complex. Expressions of this sort show up most often when parsing large PDKs with complex use of the `.FUNC` feature, and when using the "tablefile" feature to import a large file of time/voltage pairs for use in a `B` source. These issues were addressed by the new expression library.
1246-SON: Supernoding seg faults / fails to remove all nodes in parallel	Parallel regression testing illustrated a failure to perform supernoding when run on more than 2 processors. This failure was caused by a code issue and has been addressed.
1263-SON: Oscillator HB seg faults in linear solver when run in parallel using 4 processors	Parallel regression testing illustrated a failure in oscillator HB algorithm due to the linear solver and builder, when run on 4 or more processors. This failure was caused by a code issue and has been addressed.
1311-SON: PowerGridTransformer tests sometimes hang in parallel on more than 2 processors	Parallel regression testing illustrated failures of circuits that have PowerGridTransformer models when run on more than 2 processors. These failures were caused by a code issue and has been addressed.
Gitlab-ex Issue 130: The FFTW interface does not work properly for even number of samples	The results from .HB and .FFT analysis were wrong for the N/2 harmonic when the number of samples N is even. The real part of the result was incorrectly copied to the imaginary part for the N/2 harmonic. This led to the wrong results. This is fixed now.
Gitlab-ex Issue 159: The max time step is ignored for the first step out of a break point	The max time step can be specified by a user or come from some special devices. Some of them are not applied to the first step out of a break point. This is fixed now.
Gitlab-ex Issue 122: The dynamically inserted break points from delay device were not handled correctly	The devices that use history information can dynamacally insert break points during the transient simulation. This was not handled correctly and led to accuracy issues, discontinuous behavior, wrong results and convergence issues. This is fixed now.

Known Defects and Workarounds

Table 3: Known Defects and Workarounds.

Defect	Description

Gitlab issue 85 Complex-valued parameters are not handed correctly	The Xyce expression library was rewritten for the 7.2 release, and has added support for complex numbers in expressions. However, the use of complex-valued parameters and global parameters is not correct yet. This is because parameters and global parameters are still assumed to always be real numbers. An example is:
	
	`.PARAM P1={log10(-2)}`
	`V1  1 0 1`
	`R1  1 0 1`
	`.OP`
	`.PRINT DC`
	`+ {Re(P1)} {Img(P1)}`
	`+ {Re(log10(-2))} {Img(log10(-2))}`
	`.END`
	
	The output will have `RE(P2)` equal 3.01e-01 and `IMG(P2)` equal 0, which is incorrect. However, the non-parameter fields will be output as `Re(log10(-2))` equal to 3.01e-01 and `Img(log10(-2))` equal to -1.36e+00, which is correct. The code assumes that the parameter `P1` is unconditionally real.

Gitlab issue 60 Xyce/ADMS omits derivative code for output arguments of analog functions if return value derivatives are not needed	Verilog-A permits analog functions (user defined functions) to have arguments that can be used to return values in addition to the return value of the function. These output arguments have their values calculated as a "side effect" of the function call. Due to a difficulty with bookkeeping, if the return value of the function is neither used in sources nor ddx() calls, Xyce/ADMS will not emit any code that calls the function in such a way that the derivatives of the output arguments would be computed. This can lead to incorrect results if the output arguments are later used in any way where their derivatives are required (e.g. on the right-hand side of non-noise contributions, or as the argument to be differentiated in a ddx() call). ***Workaround***: Either do not write analog functions with output arguments (thereby never having side-effects, a best practice), or make sure that the return value of the function is always used in a manner such that its derivatives will be required (use in non-noise contributions or as the argument to be differentiated by ddx().
Gitlab issue 138 Oscillator HB tests fail in parallel on 8 processors	This is due to bad memory access in the solution vector by the expression library. ***Workaround***: There is none.

Table 3: Known Defects and Workarounds.

Defect	Description
Gitlab issue 28 Limitations on allowed parameter names is not fully documented	The exact limitations on allowed parameter names is not clear in the documentation, nor is any exhaustive list available. Single-character non-alphabetic names are mostly illegal for either `.param` or `.global_param` names, but there may be other undocumented limitations. These invalid parameter names will generally cause Xyce to exit with an appropriate error message.

1309-SON: Incorrect results for AVG, INTEG, RMS measures when FROM and/or TO values are not equal to a time-step or sweep value	The `AVG`, `INTEG` and `RMS` measures can return an incorrect value if the `FROM` or `TO` qualifiers are given on the measure line and those values are not equal to an accepted time-step value, or one of the specified AC, DC or NOISE sweep values. A simple example for AC measures is:
	
	`.AC DEC 5 100Hz 1e6`
	`.MEASURE AC avg1  AVG VR(B) FROM=70e3`
	
	The answer will be correct if `FROM`=100e3, which is a requested AC sweep value. It will be incorrect for `FROM`=70e3.
	Workaround: A workaround is less obvious for TRAN measures. However, this `.OPTIONS` line can be used to force Xyce to take a time-step at the requested `FROM` and/or `TO` values:
	
	`.OPTIONS TIMEINT BREAKPOINTS=<fromValue>, <toValue>`

1262-SON: Duplicate L device definitions are not a parsing error when one of the duplicate L devices is part of a K device	As an example, this netlist will not produce a parsing error. Instead, the first L1 definition will be used in the K1 device definition.
	
	`* parsing fails to detect duplicate L1 devices`
	`V1 1 0 SIN(0 1 1KHz)`
	`L1 1 2 1e-3`
	`R1 2 0 1`
	`C1 2 0 1e-9`
	`* mutual inductor definition, with duplicate L1 device`
	`L1 4 0 1e-6`
	`L2 3 0 1mH`
	`K1 L1 L2 0.75`
	`.TRAN 0 1ms`
	`.PRINT TRAN V(1) v(2)`
	`.END`
	
	Workaround: There is none.

1031-SON: .OP output is incomplete in parallel	When Xyce is run in parallel, the `.OP` output may be incomplete. ***Workaround***: One workaround is to run the netlist in serial. Another one is to use these command line options: `-per-processor -l output`. In that case, the per-processor log files will have the `.OP` information for the devices that were instantiated on each processor.

Table 3: Known Defects and Workarounds.

Defect	Description
1009-SON: Transient adjoint sensitivities don't work with `.STEP`	Transient adjoint sensitivities require backward integrations to be performed after the primary transient forward integration. Doing this properly requires information to be stored during the forward solve, and for certain bookkeeping to be performed. Currently, these extra operations to support transient adjoints are not properly set up for `.STEP` analysis. ***Workaround***: None
1006-SON: SDT (expression library time integration) derivatives are not supported, so SDT can't be used for sensitivity analysis objective functions	SDT is a function supported by the Xyce expression library to compute numerical time integration. When this function is used, the expression library does not produce correct derivatives. This impacts Jacobian matrix entries, when SDT is used with a Bsrc, and it also impacts sensitivity analysis, when SDT is used in an objective function. For the former case, this can result in a lack of robustness for circuits that contain SDT-Bsrc devices. For the latter case, the objective function will simply be incorrect. ***Workaround***: None

1004-SON: Ill-defined .STEP behavior for "default parameters" for transient sources (SIN, EXP, PWL, PULSE and SFFM)	If, for example, these netlist lines are used in a transient (`.TRAN`) simulation:
	`V1 1 0 SIN(0 1 1)`
	`.STEP V1 1 2 1`
	then Xyce will run the simulation without warnings or errors, but no instance parameter of source `V1` will be stepped.
	Workaround: Explicitly use the desired stepped parameter (e.g., `V0`) on the `.STEP` line. For example, `.STEP V1:V0 1 2 1` would work correctly.

991-SON: Non-physical BH Loops in non-linear mutual inductor	Nonlinear mutual inductors that have high coupling coefficients (i.e. model parameter `ALPHA` over 1.0e-4) and low loss characteristics (i.e. zero `GAP`) can produce B-H loops with nonphysical hysteresis. ***Workaround***: Lower `ALPHA` values or larger `GAP` values can ameliorate this issue, but the root cause is still under investigation.
800-SON: Use of global parameters in expressions on .MEASURE lines will yield incorrect results	The use of global parameters in expressions on .MEASURE lines is not allowed, as documented in the Xyce Reference Guide. However, instead of producing a parsing error the measure statement will be evaluated with the specified qualifier value (e.g., `FROM`) being left at its default value. ***Workaround***: None, other than not doing this.

Table 3: Known Defects and Workarounds.

Defect	Description
970-SON: Some devices do not work in frequency-domain analysis	Devices that may be expected to work in AC or HB analysis do not at this time. For AC this includes, but is not limited to, the lossy transmission line (LTRA) and lossless transmission line (TRA). For HB, the transmission lines do work but the nonlinear dependent sources (B source and nonlinear E, F, G, or H source) do not work when the expression is explicitly time-dependent. ***Workaround***: The LTRA and TRA models will need to be replaced with lumped transmission line models (YTRANSLINE) for AC analysis. There is not yet a workaround for the time-dependent B source in harmonic balance.

Table 3: Known Defects and Workarounds.

Defect	Description
967-SON: Zoltan segmentation fault with OpenMPI 2.1.x and 3.0.0 on some systems	It has been observed that when Xyce and Trilinos are built with OpenMPI 2.1.x or 3.0.0 on certain unsupported operating systems, a small number of test cases in the regression suite crash with a segmentation fault inside the Zoltan library. The Xyce team has determined that this is not a bug in either Xyce or Zoltan, but is instead due to some pre-packaged OpenMPI binaries on some operating systems having been built with an inappropriate option. This option, "–enable-heterogeneous" is explicitly documented in OpenMPI documentation as broken and unusable since 2013, but some package managers have OpenMPI binaries built with this option explicitly enabled. Turning on this option causes the resulting OpenMPI build to perform certain communication operations in a way that does not adhere to the MPI standard. There is nothing that can be done in Xyce or Zoltan to fix this issue — it is entirely a bug in the OpenMPI library as built on that system. A new test case has been added to the Xyce test suite in order to detect this problem. The test is "MPI_Test/bug_967", and it will be run whenever the test suite is invoked with the "+parallel" tag as described in the documentation for the test suite at `https://xyce.sandia.gov/documentation/RunningTheTests.html`. If this test fails, your system has a broken OpenMPI build that cannot be used with Xyce. At the time of this writing, this issue is present in Ubuntu Linux versions 17.10 up to (but not including) 20.04 LTS, and there is an open bug report for it at `https://bugs.launchpad.net/ubuntu/+source/openmpi/+bug/1731938`. The issue may be present in other distros of Linux that are derived from Debian (as is Ubuntu), but we cannot confirm this. ***Workaround***: The only workaround for this problem is to build OpenMPI from source yourself, and not to include "–enable-heterogeneous" in its configure options. You should also post a bug report in your operating system's issue tracker requesting that they rebuild their OpenMPI binaries without the "–enable-heterogeneous" option. If you are using Ubuntu, you should register with that issue tracking system and add yourself to the list of people it affects in the existing bug report (doing so increases the "heat" of the bug, which may increase the likelihood of it being fixed).
964-SON: Compatibility of .PRINT TRANADJOINT with .STEP	The use of `.PRINT TRANADJOINT` is not compatible with `.STEP`. The resultant Xyce output will not be correct. ***Workaround***: There is none.

Table 3: Known Defects and Workarounds.

Defect	Description
932-SON: Analysis lines do not support expressions for their operating parameters	The Xyce parser and analysis handlers do not yet support the use of expressions on netlist analysis lines such as `.TRAN`. The parameters of these analysis lines (such as stop time for `.TRAN` or fundamental frequency for `.HB`) may only be expressed as literal numbers. ***Workaround***: There is no workaround internal to Xyce. Use of an external netlist preprocessor would be required.
883-SON .PREPROCESS REPLACEGROUND does not work on nodes referenced in expressions	The `.PREPROCESS REPLACEGROUND` feature does not replace ground synonyms if they appear in B source expressions. ***Workaround***: Do not use ground synonyms (`GND`, `GROUND`, etc.) in expressions. Use a literal "0" when referring to the ground node in expressions.
783-SON: Use of ddt in a B-Source definition may produce incorrect results	The `DDT()` function from the Xyce expression package, which implements a time derivative, may not function correctly in a B-Source definition. ***Workaround***: None.
727-SON: Xyce parallel builds hang randomly on OS X	During Sandia's internal nightly testing of the OSX parallel builds, we see that Xyce "hangs on exit" with an estimated frequency of less than 1-in-5000 simulation runs. We have not seen this issue with parallel builds for either RHEL6 or BSD. The hang is on exit, whether on a successful exit or on an error exit. The hang occurs after all of the Xyce output has occurred though. So, the user will get their sim results, but might have trouble if the individual Xyce runs are part of a larger script. ***Workaround***: None.
661-SON Lead currents and power accessors (I(), P() and W()) do not work properly in .RESULT Statements	There are two issues. First, `.RESULT` statements will fail netlist parsing if the requested lead current is omitted from the `.PRINT TRAN` line. As an example, this statement (`.RESULT I(R1)`) requires either `I(R1)`, `P(R1)` or `W(R1)` to be on the `.PRINT TRAN` line. Second, the output value, in the `.res` file, for the lead current or power calculation will always be zero.
583-SON: Switch with RON=0 leads to convergence failure.	The switch device does not prevent a user from specifying `RON=0` in its model, but then takes the inverse of this value to get the "on" conductance. The resulting invalid division will either lead to a division by zero error on platforms that throw such errors, or produce a conductance with "Not A Number" or "Infinity" as value. This will lead to a convergence failure. ***Workaround***: Do not specify an identically zero resistance for the switch's "on" value. A small value of resistance such as 1e-15 or smaller will generally work well as a substitute.
469-SON: Belos memory consumption on FreeBSD and excessive CPU on other platforms	Memory or thread bloat can result when using multithreaded dense linear algebra libraries, which are employed by Belos. If this situation is observed, either build Xyce with a serial dense linear algebra library or use environment variables to control the number of spawned threads in a multithreaded library.
468-SON: It should be legal to have two model cards with the same model name, but different model types.	SPICE3F5 and ngspice only require that model cards of the same type have unique model names. They accept model cards of different types with the same name. Xyce requires that all model card names be unique.

Table 3: Known Defects and Workarounds.

Defect	Description
250-SON: NODESET in Xyce is not equivalent to NODESET in SPICE	As currently implemented, `.NODESET` applies the initial conditions given throughout a full nonlinear solve for the operating point, then uses the result as an initial guess for a second nonlinear solve with no constraints. This is not the same as SPICE, which merely applies the given initial conditions to a single nonlinear solve for the first two iterations, then lets the problem converge with no further constraints. This can lead to a Xyce `.NODESET` failing where the same netlist in SPICE might not, if the initial conditions are such that a full nonlinear solve cannot converge with those constraints in place. There is no workaround.
247-SON: Expressions don't work on .options lines	Expressions enclosed in braces ({ }) are handled specially throughout Xyce, and may only be used in certain contexts such as in device model or instance parameters or on `.PRINT` lines.
49-SON Xyce BSIM models recognize the model TNOM, but not the instance TNOM	Some simulators allow the model parameter TNOM of BSIM devices to be specified on the instance line, overriding the model parameter TNOM. Xyce does not support this.
27-SON: Fix handling of .options parameters	When specifying .options for a particular package, what gets applied as the non-specified default options might change.

2119-SRN: Voltages from interface nodes for subcircuits do not work in expressions used in device instance parameters	This bug can be illustrated with this netlist fragment:
	
	`X1 1 2 MySub`
	`.SUBCKT MYSUB a c`
	`R1  a b 0.5`
	`R2  b c 0.5`
	`.ENDS`
	`B1 3 0 V={V(X1:a)}`
	
	This fragment will produce the netlist parsing error `Directory node not found: X1:A`. The workaround is to use `V={V(1)}` in the B-source expression instead. This bug also affects the solution-dependent capacitor.

1923-SRN: LC lines run out of memory, even if equivalent (larger) RLC lines do not.	In some cases, circuits that run fine using an RLC approximation for a transmission line, exit with an out-of-memory error if the (supposedly smaller) LC approximation is used.
1595-SRN: Xyce won't allow access to inductors within subcircuits for mutual inductors external to subcircuits	It is not possible to have a mutual inductor outside of a subcircuit couple to inductors in a subcircuit. ***Workaround***: Put all inductors and mutual inductance lines that couple to them together at the same level of circuit hierarchy.

Supported Platforms

Certified Support

The following platforms have been subject to certification testing for the Xyce version 7.3 release.

- Red Hat Enterprise Linux 7, x86-64 (serial and parallel)
- Microsoft Windows 10, x86-64 (serial)
- Apple macOS 10.14 and 10.15, x86-64 (serial and parallel)

Build Support

Though not certified platforms, Xyce has been known to run on the following systems.

- FreeBSD 11.x on Intel x86-64 and AMD64 architectures (serial and parallel)
- Distributions of Linux other than Red Hat Enterprise Linux 6
- Microsoft Windows under Cygwin and MinGW.

Xyce Release 7.3 Documentation

The following Xyce documentation is available on the Xyce website in pdf form.

- Xyce Version 7.3 Release Notes (this document)
- Xyce Users' Guide, Version 7.3
- Xyce Reference Guide, Version 7.3
- XDM Users Guide
- Xyce Mathematical Formulation
- Power Grid Modeling with Xyce
- Application Note: Coupled Simulation with the Xyce General External Interface
- Application Note: Mixed Signal Simulation with Xyce 7.2

Also included at the Xyce website as web pages are the following.

- Frequently Asked Questions
- Autotools Building Guide (instructions for building Xyce from the source code)
- Running the Xyce Regression Test Suite
- Xyce/ADMS Users' Guide
- Tutorial: Adding a new compact model to Xyce

External User Resources

- Website: http://xyce.sandia.gov
- Google Groups discussion forum: https://groups.google.com/forum/#!forum/xyce-users
- Email support: xyce@sandia.gov
- Address:

Electrical Models and Simulation Dept.
Sandia National Laboratories
P.O. Box 5800, M.S. 1177
Albuquerque, NM 87185-1177

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.
SAND2021-5533 O