Xyce™ Parallel Electronic Simulator Version 7.4 Release Notes

Sandia National Laboratories

October 21, 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

XDM

- General:
  - Substitutions done by XDM for the special variable TEMP are no longer necessary and are taken out in this release.
- HSPICE:
  - Xyce's ”-hspice-ext” option, which should be used when running HSPICE netlists translated by XDM, now by default expects the ”.” for the subcircuit hierarchy separator character. Therefore, XDM translations will leave that character unchanged rather than translating it to Xyce's native ”:” subcircuit separator character.
  - Secondary sweep of .DC analyses should now be translated correctly into Xyce.
  - Xyce can now handle resistors with solution dependent expressions for their resistance values. Therefore, XDM will no longer translate solution dependent resistors into behavioral B-element sources.

New Devices and Device Model Improvements

- The levels 1 and 2 diode now support a multiplicity factor. This is the "M" parameter on the instance line, not to be confused with the model parameter of the same name (the p-n junction grading parameter).
- The resistor model now supports solution-dependent expressions, so it is possible to have voltage-dependent resistors in Xyce netlists.

Enhanced Solver Stability, Performance and Features

- Voltage limiting in the BSIMSOI version 3 device model can now be optionally turned off, while leaving it turned on in other devices. The implementation of voltage limiting in the BSIM SOI device is unorthodox and has been observed to cause robustness problems. Turning it off can improve runtimes.
- Behavioral models have been optimized to eliminate redundant calls to the expression library. For historical reasons, the Bsrc device was calling the expression library more than it needed to. This has been corrected, and it will improve the runtime of circuits with lots of behavioral models. This same optimization has also been applied to other expression-dependent models, such as the capacitor and switch devices.
- The expression library has undergone several optimizations since the last code release. These optimizations include a reduction in the number of necessary syntax tree traversals, a reduction in the memory footprint, reduction in dynamic memory allocations and deallocations, and many evaluation costs are now amortized in the expression setup. For circuits with large numbers of expressions, these optimizations produce runtime performance improvements of netlist parsing and overall simulation speed.

Interface Improvements

- "Continuous mode" measures, that may return more than one result, are now supported for the DERIV-AT, DERIV-WHEN, FIND-AT, FIND-WHEN and WHEN measure types for .AC, .DC, .NOISE and .TRAN analyses.
- The -remeasure command line option now supports Xyce output files made with FORMAT=GNUPLOT.
- The full range of negative values (-1,-2,-3,...) are now supported for the RISE, FALL and CROSS qualifiers for the DERIV-WHEN, FIND-WHEN and WHEN measure types.
- The ERROR measure type is now supported for .AC and .NOISE analyses.
- Additional window functions (BARTLETTHANN, BLACKMAN, NUTTALL, COSINE2, COSINE4, HALFCYCLESINE, HALFCYCLESINE3 and HALFCYCLESINE6) have been added for .FFT lines. In addtional, both "symmetric" and "periodic" versions are now supported for all window functions.
- .OPTIONS FFT FFT_MODE=<0|1> now allows the Xyce user to select whether the Xyce .FFT processing and output is more compatible with HSPICE (0) or Spectre (1). The default is HSPICE.
- The BINSIZ qualifier is now implemented in Xyce for the ENOB, SFDR and SNDR measure types. It can be used to account for any "broadening" of the spectral energy in the first harmonic of the signal.
- .STEP analyses now support .FOUR analyses.
- A bug in the handling of wildcard current output was fixed and now these wildcards may be requested using the external output API irrespective of whether the netlist being run contains print lines outputting the same data.
- More complex .PRINT line wildcards, such as VR(X1*) and IR(B1?), are now supported for all voltage and current operators. So, this feature now also supports the VR, VI, VM, VP, VDB, IR, II, IM, IP and IDB operators used for .AC, .HB and .NOISE analysis. It also now supports the lead current operators IB, IC, ID, IE, IG and IS.
- All valid .PRINT line wildcards are now supported for numerical lead-current designators, such as I1 and I2. (Note: At present, those designators are only supported for the T and YGENEXT devices.)
- The use of a period as the subcircuit name separator is now supported as a user-specified option. The default separator in Xyce is still the colon. Period separators are enabled by the -hspice-ext command line option, either using all or separator as the second argument.
- Inductance values, for coupled mutual inductors are now accessible in the netlist, for outputs and also parameter-centric analyses such as .STEP and .SENS.

Xyce/ADMS Improvements

- An error in the generation of code for derivatives of the "min" function with probe-dependent arguments has been fixed. The error impacted several internal models that had been generated using ADMS, but has not led to significant differences in simulation results. Minor differences in convergence behavior have been observed.
- Support for the atan2 function and generation of correct code for its derivatives has been added.

Important Announcements

- 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.

- The link line required to link Xyce libraries (installed by “make install”) to user code has been simplified. Now only “-lxyce” is needed, where in previous releases it was necessary to link in several related libraries.

Interface Changes in this Release

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

Change	Detail
Change in output file names produced by .FOUR analyses	Results from Fourier analyses are now output to <netlistName>.four# files, where the suffixed number (#) starts at 0 and increases for multiple iterations (.STEP iterations) of a given simulation.
Changes in the format of .FFT output sent to the <netlistName>.fft# files	The output in those files is no longer filtered based on the FREQ, FMIN and FMAX values on each .FFT line. Instead the data for every harmonic is output. The START and STOP values used for each .FFT analysis are now output.
Additional .FFT output to stdout	If .OPTION FFT FFTOUT is set to 1 then the additional metrics (ENOB, SFDR, SNR, SNDR and THD) and a sorted list of the 30 largest harmonics, for each .FFT line, are now output to stdout.
Changes to output file names produced by the -o command line option	In prior releases, the -o filename only applied to the file name produced by the primary .PRINT line. The -o filename is now used as the "basename" for all of the output files. As an example, for a .TRAN analysis, this command line will now produce the output file <results>.prn, instead of the file <results>, from the .PRINT line. Xyce -o results <netlistName> The results from any .MEASURE TRAN, .MEASURE TRAN_CONT, .FFT and/or .FOUR lines in the netlist will now be placed into the files <results>.mt0, <results>_measurename.mt0, <results>.fft0, and <results>.four0, respectively. For a .STEP analysis, the results file <results>.res will also be made.

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 301: Level 2, Nonlinear mutual inductor could cause valgrind errors	The level 2, nonliner mutual inductor was using an uninitialized data structure in the calculation of lead currents. This was causing valgrind to issue errors for tests using this device. The underlying code has been corrected.
Gitlab-ex issue 310: Modulus operator didn't correctly handle real-numbered inputs	The modulus operator in the expression library was incorrectly casting its inputs to type integer. This has been corrected and real numbered inputs are handled correctly now.
Gitlab-ex issue 279/Github issue 41: Incomplete support for ATAN2 in Xyce/ADMS	Xyce/ADMS previously had incomplete support for the standard Verilog-A function "atan2". If used in normal module scope, correct code would be emitted for the evaluation of the function itself, but place holder comments would be emitted instead of the derivatives. If used in analog function context there would be a fatal error stating that the feature was not yet implemented. Correct code for atan2 and its derivatives is now generated in both contexts.
Gitlab-ex issue 291: Incorrect behavior for DERIV measure at first simulation step for AC, DC, NOISE and TRAN measure modes	If the user requested the `DERIV-WHEN` measure value at the first simulation step then the measured value could be wrong in some cases.

Gitlab-ex issue 283: FIND-AT measure gives incorrect value if the AT value is the first time, frequency or DC sweep value in the simulation	As an example, this set of `.AC` and `.MEASURE AC` lines would return a measure value of -nan.
	`.ac dec 5 100 1e6`
	`.MEASURE AC vibat100 FIND vi(b) at=100`

Gitlab-ex issue 290: Indexing mistake in the expression library's `DDX` operator	The `DDX` operator had a mistake in the routine which found the dependent variable in the expression syntax tree. This mistake only applied to derivatives that were taken with respect to voltage nodes, when there were multiple nodes present in the expression. The consequence would be that the derivative was taken with respect to the wrong variable. This mistake has been corrected.
Gitlab-ex issue 263: Nonlinear mutual inductor parameter handling is incompatible with `STEP`	The nonlinear mutual inductor was performing unit conversion of model parameters within the model constructor. While this works in most cases, it would fail in a `STEP` analysis where a model parameter is changing. The model parameteres, gap, path and area are now handled correctly and are compatible with `STEP`.

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 274: Xyce/ADMS generages incorrect code for derivatives of "min" function	A logic error in the Xyce/ADMS code generation back-end was producing incorrect code for derivatives of expressions involving the "min" function whose arguments were dependent on probes. This had minor impact on the convergence behavior some internal models in Xyce due to errors in the Jacobians of those models that used this function.
Gitlab-ex issue 264: Add multiplicity to levels 1 and 2 diodes	An instance parameter "M" has been added to the level 1 and 2 diodes to implement multiplicity (number of devices in parallel). This instance parameter must not be confused with the model parameter of the same name, which implements the P-N junction grading parameter.
Gitlab-ex issue 257: Improve error handling for invalid FREQ values on .FOUR lines and invalid AT values on FOUR measure lines	If a Xyce user entered an invalid `FREQ` value on a `.FOUR` line or an invalid `AT` value on a `FOUR` measure line then that would cause the simulation to exit before all of the post-processing output files had been made at the end of the transient simulation.
Gitlab-ex issue 243/Github issue 29: I(*) not expanded in external output	The design of the code that handled output wildcards involving currents (including I(*), P(*), and W(*)) was such that wildcards requested by external output interfaces through the library API would not have currents provided unless a `.PRINT` line in the netlist also requested that wildcard. The handling of this feature has been reworked so that wildcards in external output requests function correctly irrespective of any `.PRINT` lines in the netlist, and even if there are no `.PRINT` lines at all.
Gitlab-ex issue 204/Github issue 24: Interlibrary dependency problems	For user-compiled and installed Xyce builds beginning with release 6.1 and up to release 7.3, Xyce created and installed multiple, interdependent libraries such as "libxyce", "libADMS" and "libNeuronModels", all of which had to be linked into any user code that used Xyce. As of release 7.4, only libxyce is installed, and it contains all functions previously placed into the other libraries. If you are building codes to be linked against Xyce libraries, you must simplify your link line to link in just "-lxyce".
Gitlab-ex issue 221: `.OPTIONS MEASURE MEASFAIL=1` does not work properly for successful `TRIG-TARG` measures	If `.OPTIONS MEASURE MEASFAIL=1` was used in a netlist then the results for a successful `TRIG-TARG` measure would be correct in the descriptive information sent to stdout. However, the measure's value would be shown as FAILED in the `<netlistName>.mt0` file.
Gitlab-ex issue 229: Parsing error when V() expression contains extraneous trailing spaces	The new expression library did not handle whitespace correctly, in certain contexts. This has been fixed in the expression lexer.
Gitlab-ex issue 235: Extend the R=0 capability to allow use of resistor .MODELs	Resistors that have a resistance value of zero are treated different inside of Xyce. This different handling did not allow for such resistors to refer to `.MODEL` statements. That has been fixed and now this use case works.

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 239: `.options parser scale` doesn't play nice with dependent parameters	If Xyce was used with a PDK that used `.option scale` (or Xyce's version `.options parser scale`), Xyce would apply this scale factor in the Instance::processParams function of the relevant MOSFET devices. However, if the device instance contained any dependent parameters, then every time those parameters were updated, this function was called again, and thus the scale factor was applied each time. This resulted in the wrong L and W values. This has been corrected and scale factors are now applied no more than once.
Gitlab-ex issue 245: An endless loop in parsing can be caused if a `.SUBCKT` in an include file is not terminated properly	If a subcircuit was defined in an include file that was not terminated properly, the Xyce parser would continue parsing without acknowledging that the subcircuit had ended. This would result in an error in the parsing later or an endless loop. This has been corrected.
Gitlab-ex issue 262: Bug in `.PREPROCESS ADDRESISTORS` logic that generates file output	If `.PREPROCESS ADDRESISTORS` is used in a file to add resistors to one-terminal connections or to make DC paths to ground, then all `.PREPROCESS` calls are commented out in the file outputted that includes those resistors. This included `.PREPROCESS` calls that are executed to replace ground synonyms or remove unused devices, which need to be called each time the circuit is simulated. This no longer happens.

Gitlab-ex issue 277: Measure names that start with an "operator letter" do not work in EQN measures	A measure name that started with a letter that was valid as the first character in an operator name ('V', 'I', 'P', 'W', 'D', 'N' or 'S') would work correctly in (for example) a `MAX` measure. However, that `MAX` measure could not subsequently be used in an `EQN` measure. A simple example is as follows:
	`.MEASURE TRAN V1MAX MAX V(1)`
	`.MEASURE TRAN EQN1 EQN V1MAX`

Gitlab-ex issue 296: Error in Parallel::Directory when dependent variables for devices included the ground node	A runtime failure was observed from the Parallel::Directory when the ground node was a dependent variable for a device parameter. The Parallel::Directory is a class that the Topology package uses to determine ownership of shared voltage nodes in parallel, which cannot include the ground node. The dependent variable resolution has changed to address this issue.

1322-SON: Preserve ordering of .PRINT line wildcards	In previous Xyce versions, these two `.PRINT` lines would both output the `V(*)` wildcard results before the `I(*)` wildcard results.
	`.PRINT TRAN V(*) I(*)`
	`.PRINT TRAN I(*) V(*)`

1145-SON: `.GLOBAL_PARAM` does not work in `.IC` statements	This has been resolved and global parameters can be used to set the value of an `.IC` statement.

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
1134-SON: Mismatched (or missing) expression delimiters are not flagged as an error on .MEASURE lines	In previous versions of Xyce, incorrect use of expression delimiters might not be flagged as an error during netlist parsing of `.MEASURE` lines. A simple example was: `.measure tran noEndCurly eqn {V(1)`
1029-SON: Improve checking of `.PRINT` type versus analysis type	The checking of the `.PRINT` type versus the analysis type had some deficiencies. It would falsely report an error if a `.PRINT TRANADJOINT` line preceeded the `.PRINT TRAN` line. It could falsely not report an error if the `.OP` statement preceeded the primary analysis statement (e.g., `.TRAN`), or if a valid `.PRINT` line was the first `.PRINT` line in the netlist but a subsequent `.PRINT` line was not allowed for the requested primary analysis type.

Known Defects and Workarounds

Table 3: Known Defects and Workarounds.

Defect	Description
Gitlab-ex issue 316: Model binning and UQ analyses on sizing parameters such as L and W don't play well together	Currently, Xyce performs model binning at the parser level, and is unable to subsequently update model bins if the values of L and W are changed after parsing is over. This means that analyses that modify sizing parameters (such as `.STEP` or `.SAMPLING`) may produce incorrect results.

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().

Table 3: Known Defects and Workarounds.

Defect	Description
Gitlab issue 41 Xyce/ADMS does not handle modulus operator according to LRM specifications	The Verilog-A language reference manual specifies how the modulus operator (%) should behave, and it is valid for all argument types and values. Xyce/ADMS does not follow this spec and simply emits the equivalent C++ expression using the same operator. As a result, expressions using the modulus operator are only correct if the arguments are both integer expressions. In most cases where this condition is not met, the code generated by Xyce/ADMS will not even compile.
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.

Table 3: Known Defects and Workarounds.

Defect	Description
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.
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 and later, 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 12.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 7.4 Documentation

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

- Xyce Version 7.4 Release Notes (this document)
- Xyce Users' Guide, Version 7.4
- Xyce Reference Guide, Version 7.4
- 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
- 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.
SAND2021-13318 O