Xyce™ Parallel Electronic Simulator Version 7.5 Release Notes

Sandia National Laboratories

April 28, 2022

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

- Translations of netlists using binned models will no longer have the `.OPTION PARSER MODEL_BINNING=TRUE` statement. This option is turned on by default in Xyce now and therefore no longer needs to be explicitly declared.

- Duplicate output variables will now be removed from the `.PRINT` line.

- The expression and function pre-processing capabilities have been removed from XDM.

spectre The `else` conditional block in `if-else` statments will now be commented out by XDM, since conditional statement support is not fully mature yet in Xyce.

pspice Fixed bug where instance parameters for temperature coefficients for resistors were not translated.

New Devices and Device Model Improvements

- The L-UTSOI MOSFET model (level 10240 MOSFET) is now available under an open source license and the source code is now part of the main Xyce source code repository. It had previously only been available in our "non free" binaries ("XyceNF").

- The level 1 BJT now supports a multiplicity factor `M=`.

Enhanced Solver Stability, Performance and Features

- Source stepping, which is one of the solver techniques that Xyce automatically attempts when trying to solve the DCOP, now includes current sources as well as voltage sources. As a result circuits using current sources are much more robust for the DCOP calculation.

- The new harmonics selection method based on box truncation has been added for HB analysis

- Parameter handling for `.SAMPLING` and other UQ methods is now much more efficient (see issue 306 in the fixed defects table).

Interface Improvements

- TRIG-TARG measures are now supported for .AC, .DC and .NOISE analyses. In addition, the TRIG-TARG measure is now more compatible with both ngspice and HSPICE. See gitlab-ex issues 220 and 289 in the Fixed Defects Table for more details.

- "Continuous mode" measures that may return more than one result are now supported for the TRIG-TARG measure type for .AC, .DC, .NOISE and .TRAN analyses.

- The scale parameter, which is used to automatically scale device sizing parameters is supported and can be specified using either `.option scale=number` or `.options device scale=number`. The new syntax is compatible with most other SPICE-style simulators.

- Added the mixed signal interface function `Simulator::getTimeVoltagePairsSz` supplies the maximun number of time, voltage or state values for an ADC in a subsequent call to `Simulator::getTimeVoltagePairs` or `Simulator::getTimeStatePairs`.

- Added the mixed signal interface functions `xyce_getTimeVoltagePairsADCLimitData()` and `xyce_getTimeStatePairsADCLimitData()` to allow the calling program to specify the maximum size of the allocated memory that can be used for copying ADC time-voltage and time-state history data. This allows the caller to allocate only the required amount of memory and ensure that Xyce cannot overwrite the memory area. The existing c-interface functions, `xyce_getTimeVoltagePairs()` and `xyce_getTimeStatePairs()` are now deprecated. They work in this version of Xyce as they did in the past but they will be removed from a future version of Xyce. See the Application note, Mixed Signal Simulation with Xyce 7.5 for further details.

- The Xyce Python interface file `xyceinterface.py` is now compatible with Python 3.x. The interface file is still backwards compatible with Python 2.7. The Mixed Signal Simulation with Xyce 7.5 Application Note has examples of using Xyce from Python.

- Implicit multipliers on subcircuits (`M=`) are now supported.

- When a parameter of a given name is defined multiple times in the netlist, Xyce now has several options for how this is handled. Before, Xyce would silently allow multiply-defined parameters, and use the last one encountered during parsing. There is now a command line option (`-redefined_params`), which can be used to make this a fatal error, or to use the first, rather than last multiply-defined parameter.

- Subcircuit instance parameters are now allowed to reference other subcircuit instance parameters on the same line.

- When any measure fails, Xyce now reports “FAILED” in both the console output and “.mt#” files by default. This new behavior is the same as what Xyce 7.4 would have done with the “.options MEASURE MEASFAIL=1” option specified. Previous versions of Xyce would have output “-1” for the failed measure in the “.mt#” file by default. The old behavior may be forced by specifying “.options MEASURE MEASFAIL=0 DEFAULT_VAL=-1” if desired.

Xyce/ADMS Improvements

- Several bugs were addressed and are listed in the fixed defects table.

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.

- Xyce binary installers now contain all the files that would be installed by “make install” instead of just the Xyce binary. This includes Xyce headers and library files that would be used to link external codes to Xyce. Use of the Xyce executable itself is unchanged by this packaging update. Use of these headers and libraries in user code requires that the user have the same compilers that the Xyce team used to build the binary, which may not be the case for all users.

Interface Changes in this Release

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

Change	Detail
Continued support for FRAC_MAX qualifier on TRIG-TARG measure lines	For backwards compatibility with previous Xyce versions for internal users, `.OPTIONS MEASURE USE_LTTM=<value>` has been added. This option defaults to 0, which uses the new version of the TRIG-TARG measure; while setting it to 1 will use the old version of the TRIG-TARG measure for all TRIG-TARG measures in the netlist. If the FRAC_MAX qualifier is used on a TRIG-TARG line then Xyce will automatically default to `USE_LTTM=1` for that particular measure line.

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 220: .MEASURE trig/targ doesn't match ngspice when there is negative setup time.	Previous Xyce versions would not return a negative value from a `TRIG-TARG` measure. The `TARG` clause would only be evaluated if the `TRIG` clause was satisfied. A measure line such as this will now properly return M4 = -0.5 with targ = 0.25 and trig = 0.75, instead of failing to find the targ time.
	`VPWL2 2 0 pwl(0 0 0.5 1 1 0)`
	`.MEASURE TRAN M4 TRIG V(2)=0.5 CROSS=2`
	`+ TARG V(2)=0.5 CROSS=1`

Gitlab-ex issue 270: Inconsistency in column headers for .PRINT SENS for TRAN/DC and AC	The format of the column headers in the sensitivity output files is different between AC and DC/TRAN analysis. An example of a sensitivity data column name in a DC analysis output file might look like this:
	`d{V(B)}/d(R1:R)_Dir`
	while in an AC analysis output file the title of the sensitivity data might be
	`d_VR(B)/d_R1:R_dir`
	Additionally, the complex and polar notation in the AC output implies that the objective is a voltage (using VR, VI, VM, VP). This change tries to make all the sensitivity column headers follow the same convention:
	`d_{expression}/d_parameter_dir`
	(the suffix may be either _dir or _adj). The component notaion on the AC output has been updated to use Re(exp), Im(exp), Mag(exp) and Ph(exp). In working this issue, the processing logic for AC .SENS lines was also refactored. The objvars and acobjfunc components of a .SENS line are now handled by the same processing functions. With this change, the restriction against using both objective function specifiers on the same .SENS line was removed.

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 289 Improvements to TRIG-TARG measure	Separate `TD` (time delay) and/or `AT` qualifiers are now allowed for both the `TRIG` and `TARG` clauses. Expression support has also been improved. Previously, expressions did not work correctly in the `TARG` clause. Examples are as follows:
	`.TRAN 0 1`
	`.MEASURE TRAN M1 TRIG V(1)=0.5 CROSS=1 TD=0.3`
	`+ TARG V(1)=0.5 CROSS=1 TD=0.8`
	`.MEASURE TRAN M2 TRIG AT=0.05`
	`+ TARG V(1)=0.5 CROSS=1`
	`.MEASURE TRAN M3 TRIG V(1)=0.5`
	`+ CROSS=1 TARG AT=0.8`
	Xyce will now properly report M1 = 0.5 with targ = 0.875 and trig = 0.375, M2 = 0.075 with targ = 0.125 and trig = 0.05, and M3 = 0.675 with targ = 0.8 and trig = 0.125.

Gitlab-ex issue 304: Change Xyce behavior for failed measures	Xyce was using “-1” as the output value when any measure (from a `.MEASURE` statement) failed. This was inappropriate, because -1 is a legal value for many measures, making it difficult to tell whether the measure worked or not. In Xyce 7.4 an option `.options measure MEASFAIL=1` that allowed a user to request that Xyce output “FAILED” for failed measures instead. In Xyce 7.5 this style of output has been made the default. The old behavior can be forced by using `.options measure MEASFAIL=0 DEFAULT_VAL=-1` if desired.
Gitlab-ex issue 306: Using expression random operators with sampling is too slow for large numbers of parameters	For complicated netlists and PDKs, Xyce was inefficient in processing large numbers of uncertain parameters, when specified using expression-based random operators such as `AGAUSS`. There were several compounding issues that caused this to happen, and they have been corrected.
Gitlab-ex issue 318: Fix .PREPROCESS REPLACEGROUND to avoid unnecessary replacement of ground synonyms	When using “.PREPROCESS REPLACEGROUND TRUE” to replace ground synonyms, character strings were being replaced on lines where it was not necessary, like subcircuit definition lines. This has been corrected.
Gitlab-ex issue 319: Improve error checking for invalid .MEASURE lines	The error checking for invalid `DERIV-AT`, `DERIV-WHEN`, `FIND-AT`, `FIND-WHEN` and `WHEN` measures has been improved. Some invalid measure lines that would previously be reported as “FAILED” are now correctly reported as parsing errors.
Gitlab-ex issue 321: Fix Inf/NaN trap in DampedNewton nonlinear solver	The convergence check in the DampedNewton nonlinear solver was not correctly trapping NaNs in the residual vector. Thus, the nonlinear solver failure logic would not be correct, allowing the nonlinear solver to continue when it should stop. This has been corrected.

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 328: Support C for temperature units	When specifying temperature, Xyce would inappropriately exit with error if the units of Celsius were explicitly used in the netlist. This has been corrected, and now a line such as `.options device temp=25C` will work correctly. The units for temperature in Xyce have always been Celsius, so this was only a parsing issue.
Gitlab-ex issue 338: Added the function `Simulator::getTimeVoltagePairsSz`	The function `Simulator::getTimeVoltagePairsSz` supplies the maximum number of time, voltage or state values for an ADC in a subsequent call to `Simulator::getTimeVoltagePairs` or `Simulator::getTimeStatePairs`. Also, calling `Simulator::getTimeStatePairs` no longer clears the voltage history of an ADC, so a calling program that needs both state and voltage history can first call `Simulator::getTimeStatePairs` and then call `Simulator::getTimeVoltagePairs` to get both. Calling `Simulator::getTimeVoltagePairs` still clears the ADC history. See the Application note, Mixed Signal Simulation with Xyce 7.5 for further details.
Gitlab-ex issue 340: Bug in breakpointing, when an expression combines a ternary and a table	Under rare circumstances, if an expression combined a ternary operator and a time-dependent table, an error in breakpointing could cause the time integrator to get stuck in a near-infinite loop. This has been fixed.
Gitlab-ex issue 348: Added the function `xyce_getTimeVoltagePairs-ADCLimitData()`	A new function has been added to the `XyceCInterface` called `xyce_getTimeVoltagePairsADCLimitData()` which limits the data copied to the caller allocated space to whatever maximum allocation length is provided. This avoids potential memory overwriting that could occur with the general access function `xyce_getTimeVoltagePairsADC()`. See the Application note, Mixed Signal Simulation with Xyce 7.5 for further details.
Gitlab-ex issue 351: Support Python 3 in Xyce	The Python interface to Xyce in `xyce_interface.py` has been updated to support Python 3.x. See the Application note, Mixed Signal Simulation with Xyce 7.5 for further details.
Gitlab-ex issue 355: Make it possible to turn off initial junction voltages globally in device package w/o turning off voltlim	Most semiconductor device models apply a initial junction voltages during the initial iterations of a DCOP solve. However, when applying source stepping to obtain the solution, it doesn't make sense to apply these initial junction voltages, as the intent of source stepping is to initially start with all sources set to zero. This has been fixed, and is automatically applied when Xyce performs source stepping. It is also possible to manually disable initial junction voltages from then netlist via `.options device all_off=true`.
Gitlab-ex issue 358/GitHub issue 49: Ternary operator broken in analog function context	There were some use cases in which Xyce/ADMS would emit bad code for ternary operator expressions in analog function context. Correct code is now emitted in all cases.

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 364: AC sensitivities for nonlinear parameters are very slow	The AC sensitivity calculation had an inefficiency in it which made it really slow for large numbers of parameters, particularly for parameters from nonlinear devices, which had to rely on a finite differenced matrix derivative. Most of the work required could be performed a single time, rather than every time through the main loop.
Gitlab-ex issue 365: IE() does not work for BSIM-SOI 4.x	The BSIM-SOI 3 (level 10) and BSIM-SOI 4.x (levels 70 and 70450) devices all support a fourth node named "E", but until now the "IE()" print operator only worked for printing lead currents through this node for the BSIM-SOI 3. Prior versions of Xyce required use of "I4()" to output this lead current. Now, BSIM-SOI 4 and all BSIM-CMG models support "IE()" lead current operators. Wildcards for IE will now print lead currents for all devices that have an "E" node.
Gitlab-ex issue 366: IB() does not work for MOSFETs derived from Verilog-A	Most MOSFET devices have "B" nodes (either bulk or body, depending on which model is under consideration). None of the devices generated from Verilog-A supported printing of the lead current associated with this node via the "IB()" print accessor, and required instead that one use "In" (where "n" is the position of the node on the instance line). Now they are correctly accessible using "IB()".
Gitlab-ex issue 369: Out-of-bounds data storage in dx2 capability of expression library	The expression library, when used to support a behavioral model, computes an array of one or more derivatives when it is evaluated. The expression library was incorrectly written to assume that the size of this array would never change. However, this was incorrect for the use case of `.func`, which might be called multiple times, with arguments containing differing numbers differentiable variables. This was a use case that didn't come up very often, but when it did caused a memory error.
Gitlab-ex issue 371: Missing parameter definitions on .SENS line can cause Xyce to crash	When processing a .SENS line in Xyce, some error cases were not being checked. As a result, certain incomplete .SENS lines would either cause Xyce to crash or silently fail to generate sensitivity output (with no error message generated). Additional error checks have been put into place to ensure a .SENS line is valid (or report an error if not).
Gitlab-ex issue 372: AC sensitivities are incorrect when using .param for sensitivity parameters	When performing AC sensitivities with respect to .param parameters, there was an arithmetic mistake in the setup of the right-hand-side vector. This has been corrected.
Gitlab-ex issue 377: The limit random operator (2-argument version) is not correct in the expression library	Xyce has supported two versions of limit, one with two arguments and one with three. The version with two arguments was not correct, in multiple ways. When used without UQ (or sampling) it was supposed to return the mean, but instead returned the sum of the two arguments. When used with sampling, it incorrectly behaved the same as `rand()`. Both problems have been corrected.

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 390: Level 1 BJT lacks a multiplicity factor	Multiplicity is not supported in Xyce except at a low level inside each device model. The level 1 BJT did not have the code needed to support this common option.
Gitlab-ex issue 396: Incorrect code generated for noise contributions that reference analog functions	Xyce/ADMS was generating incorrect C++ code when a noise contribution referenced analog functions on its RHS.
Gitlab-ex issue 398: Incorrect code generated for single-line case items	Xyce/ADMS was generating incorrect C++ code when a case item consisted of a single probe-dependent assignment, and in analog functions when the case item was a single line assignment that depends on the function arguments. There was no error if the item was enclosed in a begin/end pair.
Gitlab-ex issue 405: Xyce does not resolve subcircuit instance parameters referencing other subcircuit instance parameters on the same line	Subcircuit instance parameters were handled in the same manner as device instance parameters, and thus forbidden to directly reference each other. However, this was not the behavior of most simulators so this has been corrected.

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 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.

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.

1037-SON: The use of non-constant values in .PARAM statements may lead to unexpected results	This netlist line (`.PARAM PA = {TEMP}`) is forbidden in Xyce since the special variable `TEMP` is not constant. However, that netlist line will not produce a Xyce parsing error, and the value of `PA` in the simulation may then be set to zero in some contexts. ***Workaround***: Non-constant values should only be used in `.GLOBAL PARAM` statements in Xyce. This restriction may be different than in other Spice-like simulators.

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-tutorials/running-the-xyce-regression-suite/`. 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 7.5 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 7.5 Documentation

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

- Xyce Version 7.5 Release Notes (this document)
- Xyce Users' Guide, Version 7.5
- Xyce Reference Guide, Version 7.5
- 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.
SAND2022-5611 O