This BETA implementation of the slow-scale SSA and documentation was intended for release with StochKit2.0.6 (July 2012). The original code and documentation was written by Kevin Sanft.

Table of Contents

Introduction

The slow-scale stochastic simulation algorithm (ssSSA) is an approximate variant of Gillespie's stochastic simulation algorithm (SSA) [1,2]. The implementation of the ssSSA contained in this distribution is a preliminary BETA version. Please follow the installation instructions carefully and review the list of known bugs and planned future updates.

A publication describing the specific algorithms used in this implementation is in preparation. In short, this implementation attempts to dynamically identify the fast process, writes custom equilibrium function code to file and compiles the functions during the simulation. To estimate the equilibrium of the fast process, an estimate of the ODE equilibrium is used. The compilation process introduces several seconds of overhead on most machines. The code monitors for changes in the fast process and will pause, recompile, and resume the simulation when new fast processes are identified. A short exact SSA presimulation is performed to determine the initial partition for the first ssSSA realization. Note that the ssSSA is not beneficial for many models; only models where the fastest reactions reach partial equilibrium on a fast timescale may see significant performance benefits from the ssSSA.

Thank you for beta testing this ssSSA implementation. If you discover any bugs that are not in the list of known bugs and planned future updates below, please contact the StochKit team by emailing stochkit@cs.ucsb.edu. Please provide a description of the error (including any error messages), your operating system, your model file, and any other information that might help us to detect the source of the problem (e.g. the command options you specified).

Installation and dependencies

This beta version of the ssSSA runs only on Mac OS X and most Linux/Unix. NOTE: This version currently does not run on Windows.

Dependencies: This ssSSA implementation requires that the following libraries be installed prior to installation:

1. ginac (GiNaC; see http://www.ginac.de/. NOTE: GiNaC requires the CLN library, consult the GiNaC documentation.)
2. gsl and gslcblas (GNU Scientific Library; see http://www.gnu.org/software/gsl/. NOTE: gslcblas may be replaced by an optimized cblas library, e.g. from ATLAS but the user must change the Makefile in the src/solvers/ssSSA_dynamic directory appropriately.)
3. boost (Boost C++ Libraries; see http://www.boost.org/. NOTE: a version of the boost library is distributed with StochKit2. To allow this ssSSA code to use that version, you must add the following to your LD_LIBRARY_PATH (Linux) or DYLD_LIBRARY_PATH (Mac OS X) environment variable: <path to StochKit2.0.*>/libs/boost_1_42_0/stage/lib/)

If the above dependencies are installed in non-standard locations, you may need to modify the Makefile in the src/solvers/ssSSA_dynamic directory manually.

In the instructions below, we assume that the StochKit release containing this ssSSA implementation has been installed. If not, follow the StochKit installation instructions before proceeding.

After the above dependencies have been installed, navigate to the StochKit2.0.*/src/solvers/ssSSA_dynamic directory and type: make

A few *warnings* about unused variables and defined but unused functions are expected. But the code should compile without *errors*.

An executable named ssssa should be created in the main StochKit directory.

Use

The interface and options for “ssssa” is the same as “ssa” (except as noted in known bugs and planned future updates). Consult the main StochKit2 manual for more information on the ssa simulation options.

Typical output is shown below.

```
$ ssssa -m models/examples/heat_shock_mass_action.xml -t 500 -r 1000
-i 10 -f
StochKit MESSAGE (ssssa_automatic): starting slow-scale SSA
simulation (beta version)...
StochKit MESSAGE (ssssa_automatic): running presimulation...
StochKit MESSAGE (ssssa_automatic): presimulation finished (took
3.46784 seconds to simulate 5000000 steps).
```

```
StochKit MESSAGE (ssssa_automatic): presimulation reached max steps
allowed at t=54.2655.
Based on presimulation, estimated simulation times are:
        exact SSA simulation (direct method): 31952.5 seconds.
        slow-scale SSA: 1038.32 seconds (NOT INCLUDING VFP COMPILE
TIME AND REPARTITION OVERHEAD).
StochKit MESSAGE (ssssa_automatic): generating code...
StochKit MESSAGE: compiling generated code...this will take a few
moments...
StochKit MESSAGE (SlowScaleSSA::resume_simulation): resuming
simulation...
...simulation finished.
StochKit MESSAGE (ssssa_automatic): total elapsed time was
approximately 1238.98 seconds.
StochKit MESSAGE (ssssa_automatic): time spent compiling was
approximately 19.9085 seconds.
done!
```

After calling the ssssa executable, several status messages are printed. An SSA presimulation is run, crude simulation time estimates are provided, then equilibrium code is generated and compiled. Then the simulation begins/resumes. Additional messages may be displayed before the "...simulation finished" message. Finally, some timing data is displayed.

Known bugs and planned future updates

- Does not run on Windows
- Does not run in parallel (i.e. ssssa runs on only one processor and the -p option has no effect)
- --species and --label simulation options might not function properly
- Running two instances of the ssssa on the same model at the same time will lead to errors
- Only works for models where all reactions are Type=mass-action of order<=2 (ie bimolecular or less, including 0th order synthesis)
- None of the ssSSA specific parameters are currently accessible to or modifiable by the user
- The fast process is not *sampled* at output times. Instead, the *effective populations* (as calculated by the equilibrium function) of the fast species are recorded. This leads to non-integer valued output and usually reduces the variance for fast species.
- Information (not simulation output data) from previous simulations is discarded (including presimulation data, initial partition data, fast process equilibrium functions, etc.). This means that a new presimulation will be ran and the discovered fast process have to be recompiled.
- No analysis/checking is done to see if the selected fast process has a unique, stable equilibrium. In the worst case, the fast process could be bistable and the equilibrium approximation will be inaccurate.
- The VFP (virtual fast process) is restricted to (any number of) reversible pairs.

- The relaxation time of an IVFP (independent virtual fast process) is estimated based on the longest relaxation time of all the reversible pairs within that IVFP. In a model such as S1<=>S2<=>S3<=>S4->S5 with S4->S5 slow and relaxation time of all the fast reversible pairs roughly equivalent, this can result in an underestimate for the relaxation time.
- Models with dimerization reactions such as A+A➔[products] can produce larger than expected error when A is a fast species with a small population. When the A+A reaction is fast, the error occurs due to the use of $\frac{c}{2}A^2$ as the propensity when determining the equilibrium. When the A+A reaction is a slow reaction, the errors occur for the same reason, as the propensity function is adjusted to avoid negative values that can occur when using the usual $\frac{c}{2}\bar{A}(\bar{A}-1)$ effective propensity. In both cases, propensity estimates involving A will incorrectly be nonzero when the total available population of A is 1; this can lead to a warning stating that the system was "unable to apply an alternate stoichiometry."
- In rare cases when the fast process changes and a fast species changes to a slow species, a bias may occur. As an illustrative example, consider the following model:
  - R1,2: A+E<=>S (with rate constant C_f=fast)
  - R3,4: B+E<=>S (with rate constant C_f=fast)
  - R5: S->D (with rate constant C_s=slow)
  - A(0)=X1, B(0)=X2, S(0)=D(0), E(0)=X1.

  When slow reaction R5 fires, one S is removed from the fast process. Equivalently, the system may remove one A and one E (or one B and one E) from the fast process. If, for example, the system removes A and E many times for many firings of R5 and eventually depletes E down to 0. Then, if R1,2,3,4 become slow reactions, the populations of A and B will be biased (there will be less A than B, whereas the populations should be equal in expectation). This is only a problem if the E is completely depleted; if E is not completely depleted, the fast species are sampled before changing the fast process and A and B will be equal in expectation.
- In some instances, conservation may be violated. Efforts are made to inform the user when this may have occurred.
- The system uses a crude ODE simulation to try to predict the frequency of *future* reactions. Ideally, the frequency and length of these forward frequency estimates would be determined dynamically and adjust themselves based on performance; however, a simpler strategy based on a fixed number of slow reaction steps and the total simulation time is currently used.
- The estimate of the ssSSA running time is based on performance data from Kevin Sanft's Macbook laptop (2 GHz Intel Core 2 Duo, 1 GB 667 MHz DDR2 SDRAM), rather than being determined on a per machine basis at installation. Your machine may give better or worse relative performance. Also, the presimulation may not be representative of the entire realization or ensemble, which can affect the time estimates.

Explanations of StochKit MESSAGE/WARNING/ERROR messages

Below is a list of messages that may be produced by the ssSSA simulation. This list is not exhaustive.

- StochKit MESSAGE (MasterVirtualFastProcess::setFastReactionIndexes): detected an IVFP that has not yet been compiled.
    - This message appears when a new fast process has been detected ("IVFP" stands for Independent Virtual Fast Process). When this occurs, the simulation is paused while equilibrium functions for the new IVFP are compiled.
- StochKit MESSAGE (SlowScaleSSA::resume_simulation): changed initial (t=0) partition in attempt to improve performance.
    - This message appears when the solver believes that using a different initial fast process (from the one selected based on the presimulation) will lead to better performance.
- StochKit WARNING (IndependentVirtualFastProcess::fireSlowReaction): unable to apply an exact alternate stoichiometry, will attempt to recover (conservation may be violated). (This message will only be displayed once per thread.)
    - The exact fast stoichiometry could not be applied directly and the system could not find an alternate equivalent stoichiometry. So, the system attempts to fire the reaction and produce a valid population. This process can lead to conservation violations.
- StochKit WARNING (IndependentVirtualFastProcess::equilibrium): negative equilibrium population detected, setting to 0.
    - The fast species equilibrium approximation function returned a negative value. This can be caused by conservation violations or rounding errors.
- StochKit WARNING (ssssa_automatic): presimulation may not have simulated long enough to provide a good sample, so including transient data to determine initial partition.
    - In this beta implementation, the presimulation will only simulate up to 5000000 steps. This message indicates that the presimulation did not cover much of the total realization simulation time and may not be a representative sample.

References

1. Y. Cao, D.T. Gillespie, L.R. Petzold. The slow-scale stochastic simulation algorithm. J Chem Phys., 122, 014116 (2005).
2. D.T. Gillespie. Exact stochastic simulation of coupled chemical reactions. J Phys Chem., 1977, 81 (25), pp 2340-2361.