Version 1.0.3
October 12, 2012

libSDE

libSDE User Manual

Java library for simulating Stochastic Differential Equations (SDE)

Thomas Schaffter

libSDE User Manual

Java library for simulating Stochastic Differential Equations (SDE)

Version 1.0.3
October 12, 2012

Thomas Schaffter
Ecole Polytechnique Fédérale de Lausanne (EPFL), Laboratory of Intelligent Systems, Lausanne, Switzerland

firstname.name@gmail.com

tschaffter.ch/projects/libsde

Table of Contents

1 Getting Started **1**
1.1 Introduction 1
1.2 Install libSDE 1
1.3 Include libSDE in Eclipse projects 3
1.4 Simulate stochastic differential equations using libSDE 4

1 Getting Started

1.1 Introduction

libSDE is an *open-source Java library* for simulating systems of stochastic differential equations (SDE). The main features of libSDE are:

- Itô and Stratonovich schemes
- Euler-Maruyama, Euler-Heun, derivative-free Milstein and Stochastic Runge-Kutta (SRK15) solvers

A brief introduction to the simulation of SDE systems is provided in *Numerical Integration of SDE: A Short Tutorial* (Schaffter, 2009).

Among other applications, libSDE is used to simulate *molecular noise in the dynamics of gene regulatory networks*. The present library is actually inclured in GeneNetWeaver (GNW)[1], a Java tool for *in silico* benchmark generation and performance profiling of network inference methods (Schaffter *et al.*, 2011). The model described in (Schaffter and Marbach, 2009) has been used to organize the fourth international *Dialogue for Reverse Engineering and Assessments and Methods (DREAM)* conference[2].

1.2 Install libSDE

libSDE is a Java library which can be included in any Java project. To install the library on your system, download the JAR (Java Archive) file libsde-1.0.3.jar and place it where the Java *classpath* points to. On Windows XP and 7, start by opening the *Environment Variables* dialog:

- **Windows XP** *Start* > Right-click on *Computer* > *Properties* > *Advanced system settings* > *Environment Variables...*
- **Windows 7** *Start* > Right-click on *My Computer* > *Properties* > Select tab *Advanced* > *Environment Variables*

[1] tschaffter.ch/projects/gnw
[2] wiki.c2b2.columbia.edu/dream/index.php/Challenges/

Check in *User variables* whether the variable *classpath* exists. If it doesn't, create it and set it with:

```
C:\LIBSDE_DIRECTORY\libsde-1.0.3.jar
```

where *LIBSDE_DIRECTORY* is the path to the folder containing the JAR file of libSDE. Multiple JAR files can be specified using semi-colons ';' to separate them. Figure 1.1 illustrates the procedure of setting the Java *classpath* to locate libSDE.

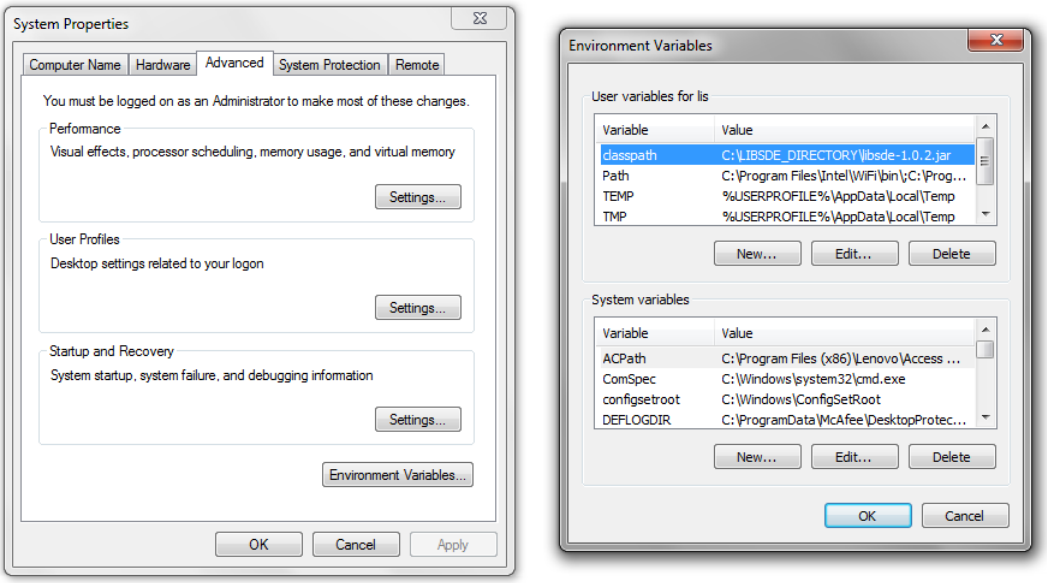

Figure 1.1: Set the Java *classpath* on Windows XP and 7 to include the libSDE library (JAR file). *LIBSDE_DIRECTORY* is the path to the folder containing the JAR file of libSDE. Multiple JAR files can be specified using semi-colons ';' to separate them.

To install libSDE on Linux or Mac OS X, open a console and type:

```
$ export CLASSPATH=CLASSPATH:/LIBSDE_DIRECTORY/libsde-1.0.3.jar
```

where *LIBSDE_DIRECTORY* is the path to the folder containing the JAR file of libSDE. Note that the symbol $ is used to indicate a command line to execute as user, so $ must not be typed in the console.

It is important to note that setting the classpath as shown above will be valid only for the current session, so we recommend doing the following manipulation to make the change permanent. Start by displaying and copying the current value of *CLASSPATH* (if not empty):

```
$ echo $CLASSPATH
```

To set automatically *CLASSPATH* at startup, append the following line to the file */home/USER/.bash_profile* (Linux) or */Users/USER/.bash_profile* (Mac OS X) where *USER* is your computer username. If the file doesn't exist, create it. By default, this is the case on Mac OS X. Note the use of colon ':' to separate multiple directories or JAR files in *CLASSPATH*.

```
export CLASSPATH=CLASSPATH:/LIBSDE_DIRECTORY/libsde-1.0.3.jar
```

1.3 Include libSDE in Eclipse projects

1. Create a new folder named *lib* in the project folder
2. Download libsde-1.0.3.jar and libsde-1.0.3-javadoc.zip to *lib*
3. Left pane of Eclipse > Right-click on the project icon > *Refresh*
4. Right-click on the project icon > *Properties* > *Java Build Path* > Tab *Libraries*
5. Click on *Add External JARs...* > Select the file libsde-1.0.3.jar located in the folder *lib* > Click on *Ok*
6. Click on *Ok*

The libSDE library is now accessible in the section *Referenced Libraries* from the project hierarchy in the left pane of Eclipse. We also recommend you to install the library documentation (Javadoc). While coding, place the mouse cursor over the name of a method available in libSDE to display a tooltip describing the usage of the method. To install the Javadoc:

7. Left pane of Eclipse > *Referenced Libraries* > Right-click on libsde-1.0.3.jar > *Properties* > *Javadoc Location*
8. Check *Javadoc in archive*
9. Check *External file*
10. Click on *Browse* > Select libsde-1.0.3-javadoc.zip > Click on *Ok*
11. Click on *Ok*

1.4 Simulate stochastic differential equations using libSDE

The package ch.epfl.lis.sde.examples contains a simple example to help you simulating your own SDE system using one of the solvers available in libSDE. First, extend the class Sde to describe your system of stochastic equations as it is done in ExampleSde. Finally, have a look at the implementation of TimeSeriesExperiment where a system including $N = 50$ equations (defined in ExampleSde) is solved using the Stochastic Runge-Kutta solver (SRK15).

If you have imported the libSDE project in Eclipse, you can run TimeSeriesExperiment which will save to file the time series of the fifty equations. A Matlab script is available in the folder *m-files* to plot the fifty trajectories generated (see Fig. 1.2).

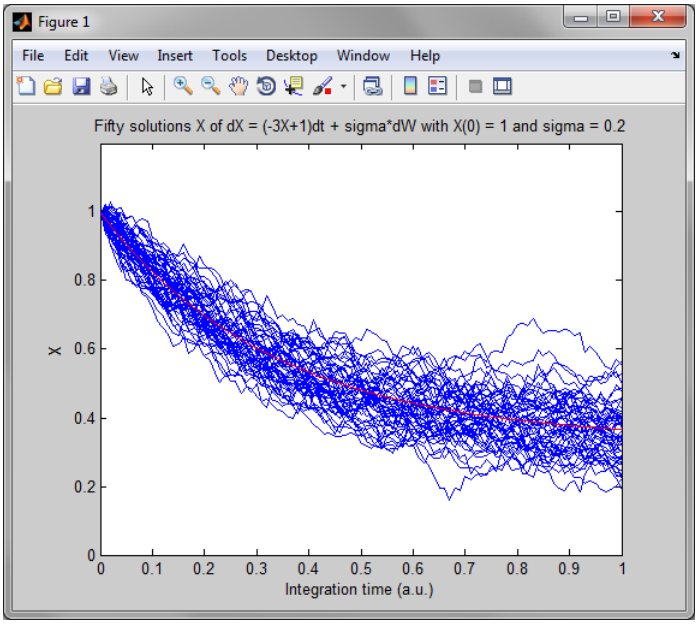

Figure 1.2: Fifty stochastic simulations of $dX = (-3X+1)dt+\sigma dW$ with $X(0) = 1$ and $\sigma = 0.2$. The expected solution is $E(X) = \frac{2}{3}exp(-3t) + \frac{1}{3}$ (in red). Here Itô and Stratonovich schemes are equivalent because the drift coefficient is a constant term. dX is integrated using the SRK15 solver implemented in libSDE.

Bibliography

Schaffter, T. (2009). Numerical Integration of SDE: A Short Tutorial. Technical report, Swiss Federal Institute of Technology in Lausanne (EPFL).

Schaffter, T. and Marbach, D. (2009). Stochastic Simulations for DREAM4. Technical report, Swiss Federal Institute of Technology in Lausanne (EPFL).

Schaffter, T., Marbach, D., and Floreano, D. (2011). GeneNetWeaver: in silico benchmark generation and performance profiling of network inference methods. *Bioinformatics*, **27**(16), 2263–2270.