StochasticDelayDiffEq.jl - An Integrator Interface for Stochastic Delay Differential Equations in Julia

Henrik T Sykora*, Christopher V. Rackauckas †, David Widmann ‡ and Dániel Bachrathy*
**MTA-BME Lendület Machine Tool Vibration Research Group, Department of Applied Mechanics, Budapest University of Technology and Economics, Budapest, Hungary*
†Department of Mathematics, Center for Complex Biological Systems, University of California, Irvine, CA USA
‡Department of Information Technology, Centre for Interdisciplinary Mathematics, Uppsala University, Uppsala, Sweden

Summary. In this work an efficient tool is presented to numerically solve stochastic delay differential equations by transforming them to stochastic differential equations. This approach allows the use of methods originally implemented for stochastic differential equations, while algorithms specialized to delay problems can be also be included. The convergence and stability of the numerous solver methods are investigated through Monte-Carlo simulations for different case studies.

Introduction

In scientific computing to analyze the behavior of dynamical systems a widely used approach is to numerically integrate the corresponding differential equation (DE). For a large number of problem classes (e.g., ordinary DEs (ODEs), delay DEs (DDEs), stochastic ODEs (SODEs or SDEs)), there is a wide range of official integrator solutions to choose from, both open-source and commercial [8]. However, for stochastic delay DEs (SDDEs) there exists no such solver, despite that there are algorithms already introduced since the late 90's [1, 5, 3] along with proof of convergence for SDDEs with point delays.
The goal of this work is to introduce a feature rich and "easy-to-use" open source tool to solve stochastic delay differential equations. These features include but are not limited to parallelized Monte-Carlo calculations, use of arbitrary precision numbers or the automatic differentiation of the solution w.r.t. to the parameters of the SDDE. This solver is available as a Julia package under the name of StochasticDelayDiffEq.jl and is part of the DifferentialEquations.jl [7] ecosystem.

Stochastic Delay Differential Equations as SODEs

Consider the Itô stochastic delay differential equation (SDDE)

$$\mathrm{d}x(t) = f(x(t), x_t, t)\mathrm{d}t + g(x(t), x_t, t)\mathrm{d}W(t), \qquad x(\theta) = \varphi(\theta),\ -\tau \le \theta \le 0, \tag{1}$$

where $f, g : \mathbb{R}^d \times C([-\tau, 0), \mathbb{R}^d) \times \mathbb{R} \to \mathbb{R}^d$ are smooth functions, $x_t(\theta) = x(t+\theta) \in \mathbb{R}^d, \theta \in [-\tau, 0)$ defines the state at time t (segment of the solution), $W(t)$ is an $\mathcal{F}_t$-measureable standard Wiener process and the initial state $\varphi : [-\tau, 0] \to \mathbb{R}^d$ is sufficiently nice and $\mathcal{F}_0$ measureable. Note that the present state $x(t)$ is separated from the solution segment x_t and this study is restricted to systems with J number of point delays, namely $0 < \tau_1 < \tau_2 < \ldots < \tau_J := \tau$, since distributed delays can be approximated as point delays using e.g. shifted Dirac delta distributions [4] or via Clenshaw-Curtis quadrature [9].
Since all values of the interval x_t are available at time t, the SDDE (1) can be rewritten as a SODE:

$$\mathrm{d}x(t) = \hat{f}(x(t), t)\mathrm{d}t + \hat{g}(x(t), t)\mathrm{d}W(t). \tag{2}$$

The transformation is performed by dynamically embedding the initial φ and current x_t states into the functions $\hat{f}$ and $\hat{g}$ as a time dependent inhomogenity, namely

$$\hat{f}(x(t), t) = f(x(t), \phi_t, t), \qquad \hat{g}(x(t), t) = g(x(t), \phi_t, t), \tag{3}$$

where

$$\phi_t(\theta) = \begin{cases} \varphi(t+\theta) & t+\theta \le 0 \\ x(t+\theta) & t+\theta > 0 \end{cases}. \tag{4}$$

This representation is similar to the method of steps [2], however it allows uninterrupted integration (with discontinuity handling) and the utilization of the solver algorithms and features of the already existing SODE ecosystem from *StochasticDiffEq.jl* [6].

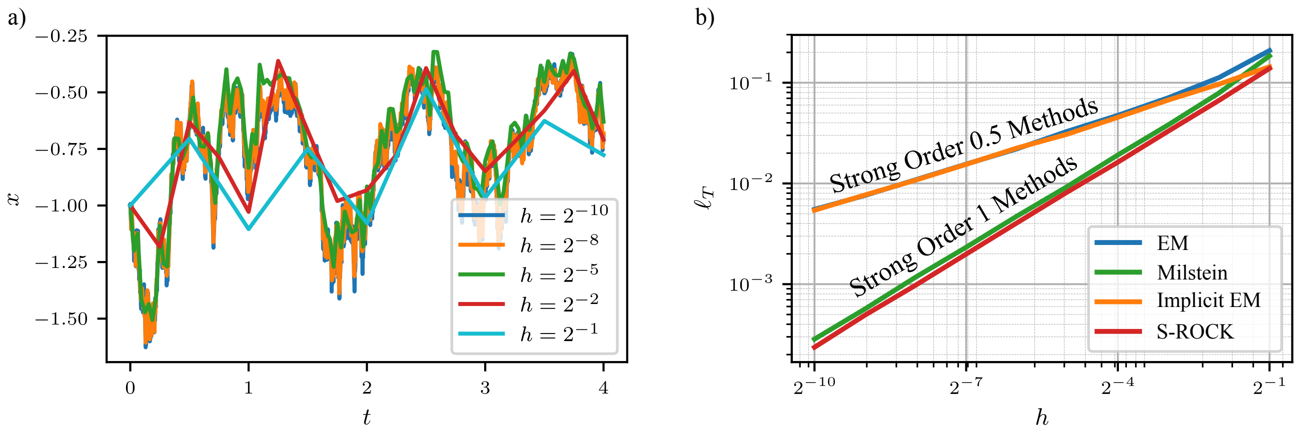

Figure 1: Demonstration of the convergence of multiple methods for $f(x(t), x(t-1), t) = x(t) - \sin(x(t-1))$, $g(x(t), x(t-1), t) = \sin(x(t))$ and $\varphi(t) = t - 1$. The sample trajectories in a) are calculated with the Euler-Maruyama method using multiple time resolutions h, and with delay $\tau = 1$, while the trajectories are integrated up to time $T = 4$. For the convergence plots in b) the ℓ_T measure is used, defined in Eq. (7).

Convergence tests

As an example, if the general SDDE (1) has a single point delay it can be rewritten in the form

$$\mathrm{d}x(t) = f(x(t), x(t-\tau), t)\mathrm{d}t + g(x(t), x(t-\tau), t)\mathrm{d}W(t). \tag{5}$$

If the Euler-Maruyama method is chosen from the package as the integrator algorithm with time step $h = \tau/N_h$, where $N_h \in \mathbb{N}^+$, the approximation of the value $x_{n+1} \approx x(t_{n+1} = (n+1)\,h)$ inherently reduces to:

$$x_{n+1} = x_n + f(x_n, x_{n-N_h}, t_n)\,h + g(x_n, x_{n-N_h}, t_n)\,\xi_n\,\sqrt{h}, \qquad \xi_n \sim \mathcal{N}(0,1), \tag{6}$$

which is shown to be convergent to the solution of (5) in [1]. The presented solver package can help to show the strong and weak convergence of other methods from the *StochasticDiffEq.jl* as well as newly created methods for SDDEs. Some examples are shown in Fig. 1, where numerical Monte-Carlo experiments were used to approximate the convergence of the solution at the final time point T by studying

$$\ell_T(h) = \mathbb{E}\left(\left\|x^h(T) - x^{h_{\text{ref}}}(T)\right\|_2\right), \tag{7}$$

where x^h is the solution of the SDDE (1) approximated with time resolution h and $h_{\text{ref}} = 2^{-15}$ is the reference time resolution.

Acknowledgement

This work was supported by the Hungarian Scientific Research Fund (OTKA FK-124462), by the Hungarian Ministry of Human Capacities (NTP-NFTÖ-19-B-0127) and by the Higher Education Excellence Program of the Ministry of Human Capacities in the frame of Artificial intelligence research area of Budapest University of Technology and Economics (BME FIKP-MI).

References

[1] Baker C.T.H., Buckwar E. (1999), Introduction to the Numerical Analysis of Stochastic Delay Differential Equations, MCCM, Numerical Analysis Technical Report, Manchester University, ISSSN 1360–1725

[2] Bellman R. (1961), On the computational solution of differential-difference equations, J. Math. Anal. Appl. 2, 108-110.

[3] Cao W., Zhang Z., Karniadakis G.E. (2015), Numerical Methods for Stochastic Delay Differential Equations Via the Wong-Zakai Approximation, SIAM Journal on Scientific Computing, 37(1), pp. A295–A318

[4] Insperger T., Stépán G. (2011) Semi-Discretization for Time-Delay Systems. Springer, NY.

[5] Kloeden P.E., Shardlow T. (2012), The Milstein Scheme for Stochastic Delay Differential Equations Without Using Anticipative Calculus, Stochastic Analysis and Applications, 30(2), pp. 181–202

[6] Rackauckas C., Nie Q., (2016). Adaptive Methods for Stochastic Differential Equations via Natural Embeddings and Rejection Sampling with Memory. Discrete and Continuous Dynamical Systems — Series B, 22(7), pp. 2731–2761.

[7] Rackauckas, C., Nie, Q., (2017). DifferentialEquations.jl — A Performant and Feature-Rich Ecosystem for Solving Differential Equations in Julia. Journal of Open Research Software. 5(1), p.15.

[8] Rackauckas C.V. (2018), A Comparison Between Differential Equation Solver Suites In MATLAB, R, Julia, Python, C, Mathematica, Maple, and Fortran. The Winnower 6:e153459.98975.

[9] Lloyd N. T. (2000) Spectral Methods in MATLAB. SIAM.