Proceeding Series of the Brazilian Society of Computational and Applied Mathematics, v. 10, n. 1, 2023.

Trabalho apresentado no XLII CNMAC, Universidade Federal de Mato Grosso do Sul - Bonito - MS, 2023

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

A numerical scheme with adaptive stepsize for Stochastic Differential Equations with additive noise

Pablo Aguiar De Maio[1]
FGV, Rio de Janeiro, RJ
Hugo de La Cruz Cancino[2]
FGV, Rio de Janeiro, RJ

Abstract. This paper introduces an A-stable adaptive integrator based on the Local Linearization (LL) technique for the computer simulation of stochastic differential equations driven by additive noise. To construct the method, novel embedding stochastic LL schemes and a adaptive strategy are proposed. Simulation results are presented to illustrate the practical performance of the introduced integrator.

Key-words. stochastic differential equations, A-stability, variable stepsize methods, local linearization)

1 Introduction

Stochastic differential equations (SDEs) have become very important tools to model complex processes influenced by random noises in several areas including physics, neuroscience, molecular biology, finance, and economics, just to mention a few of them (see [8]). Since analytical solutions to SDEs are rarely available, numerical integrators are indispensable to study the behavior of the system under consideration. This motivates the search for computational methods able of solving SDEs efficiently and in a numerically stable way. In this regard, adaptive integrators, which select the stepsize variably along the integration-time, are highly desired, and in fact, the appropriate one for practical applications.

This paper introduces a new A-stable adaptive integrator, based on the Local Linearization (LL) technique, for the computer simulation of stochastic differential equations driven by additive noise. To construct the method, embedding stochastic LL schemes and a novel adaptive strategy are proposed.

The adaptive algorithm presented here features:

- a new stochastic LL-Taylor scheme (which attains stochastic order of convergence of 1.5 and deterministic strong order of 3) embedded with a known SLL-Taylor scheme in an unconventional way (Section 2)
- an adaptive strategy to choose the stepsizes (Section 3)
- advanced techniques to accept and reject steps and keep track of all the Brownian paths generated throughout the simulation process (Section 3)

In Section 5, some simulation results are presented to illustrate the practical performance of the introduced method.

[1]pablo.maio@fgv.br
[2]hugo.delacruz@fgv.br

2 The Stochastic Local Linearization Method

The Stochastic local linearization (SLL) approach consists of approximating nonlinear SDEs by locally linear SDEs. The SLL approach is particularly useful for stiff SDEs because it produces stable schemes. By approximating the nonlinear SDE with a locally linear SDE, the SLL method can provide a more efficient and accurate method for analyzing the behavior of stochastic systems, in particular, those modeled by stiff equations.

Given a discretization of a time interval, the SLL approach consists in linearize the SDE around each point of the discretization. One way to do this is by expanding the drift and diffusion terms of the SDE by a Taylor series, and retaining only the linear terms, (but there are other ways to achieve a local linearization), then the resulting linear SDE is then solved by matrix exponentiation on each point of the time discretization.

Considering SDEs as

$$dX(t) = f(t,\ X(t))dt + G(t)dW(t), \tag{1}$$

where $X : \mathbb{R}^+ \to \mathbb{R}^d$, $f : \mathbb{R}^+ \times \mathbb{R}^d \to \mathbb{R}^d$, $G : \mathbb{R}^+ \to \mathbb{R}^{d\times m}$ and, dW(t) is a Gaussian white noise, which its integral is a Brownian motion W(t). Let g^j be the column vector j of G, and g_i^j be the element at line i and column j of the matrix G. So, is intended to approximate (1) by another simpler SDE in each instant of a given time discretization, in particular, a linear SDE which is close to the original equation. For this, applying the deterministic Taylor expansion to approximate the drift function $f(t,\ X(t))$ as a function of $(t,\ X)$ at an instant $(t_n,\ X_n)$

$$f(t,\ X(t)) \approx f(t_n,\ X_n) + f_x(t_n,\ X_n)(X(t) - X_n) + f_t(t_n,\ X_n)(t - t_n), \tag{2}$$

where f_x is the jacobian matrix of f, and $f_t = \frac{\partial f}{\partial t}$. So, using (2), the following approximation of (1) for $t \in [t_n,\ t_{n+1}]$ is obtained

$$dX(t) = [f_x(t_n,\ X(t) - X_n) + (f_t(t_n,\ X_n)(t - t_n) + f(t_n,\ X_n))]\, dt + G(t)dW(t). \tag{3}$$

This way the recursive formula of the *SLL1* scheme as being the solution of the equation (3) at t_{n+1}, starting at X_n can be defined applying this for each subinterval of a partition of $[t_0,\ T]$. Observe that considering the fact that $X(t)$ satisfies the equation (1) can make this approximation of the drift function more accurate, so taking a Taylor expansion of order 3 with the appropriated selection of terms you could obtain the following method that will be called *SLL1.5* from now on.

$$dX(t) = \left[f_x(X(t) - X_n) + \left(f_t + \frac{1}{2}\sum_{j=1}^{m} (I_{d\times d} \otimes g^{j\top}) f_{xx} g^j \right)(t - t_n) + f \right] dt + \\ + G(t)dW(t) \tag{4}$$

where $I_{d\times d}$ is the identity matrix of dimension d, and with f, f_x, f_{xx} and f_t evaluated at $(t_n,\ X_n)$, and g^j at t_n and, remark that d and m comes from the dimensions of f and G. This way, let the *SLL1.5* method as the solution of the equation (4), applied to each subinterval of a partition of $[t_0,\ T]$. The complete deduction of the *SLL1.5* method could be seen in [5].

2.1 Exponential schemes for *SLL1* and *SLL1.5*

This section will show how to build schemes for the methods above as described in [4]. It is important to note that the equations (3) of the *SLL1* and (4) of the *SLL1.5* represent SDEs with additive noise (1) and cannot be computed by themselves. Therefore, it is necessary to compute their solution to obtain practical schemes.

This way, given $h_n = t_{n+1} - t_n$, suppose that

$$\begin{aligned} (\mathit{SLL1}) \quad X_{n+1} &= X_n + \phi_1(t_n,\ X_n; h_n) + \xi_1(t_n,\ X_n; h_n), & (5)\\ (\mathit{SLL1.5}) \quad X_{n+1} &= X_n + \phi_{1.5}(t_n,\ X_n; h_n) + \xi_{1.5}(t_n,\ X_n; h_n), & (6)\end{aligned}$$

be solutions of (3) and (4) respectively . Where ϕ represents the deterministic part and ξ is the stochastic part of the solution.

This way, from [4] is known that ϕ_1 of (5) could be expressed by

$$\phi_1(t_n,\ X_n; h_n) = \begin{pmatrix} I_{d\times d} & \mathbf{0}_{d\times 2} \end{pmatrix} e^{\Lambda_1(t_n,\ X_n)h_n} \begin{pmatrix} \mathbf{0}_{d\times 1} \\ 0 \\ 1 \end{pmatrix}, \tag{7}$$

where the matrix Λ_1 is given by

$$\Lambda_1(t_n,\ X_n) = \begin{pmatrix} f_x(t_n,\ X_n) & f_t(t_n,\ X_n) & f(t_n,\ X_n) \\ \mathbf{0}_{1\times d} & 0 & 1 \\ \mathbf{0}_{1\times d} & 0 & 0 \end{pmatrix}, \tag{8}$$

where $\mathbf{0}_{a\times b}$ is the null $a \times b$ matrix. And ξ_1 is given by

$$\xi_1(t_n,\ X_n; h_n) = G(t_n)\Delta W_n, \tag{9}$$

where $\Delta W_n = W(t_{n+1}) - W(t_n) \sim \sqrt{h_n}\mathcal{N}(0,1)$ where $N(0,1)$ is an m-dimensional Standard Normal distribution. And this way the *exponential scheme of SLL1* is defined. Similarly to *SLL1*, the *exponential scheme for SLL1.5* is given by

$$\phi_{1.5}(t_n,\ X_n; h_n) = \begin{pmatrix} I_{d\times d} & \mathbf{0}_{d\times 2} \end{pmatrix} e^{\Lambda_{1.5}(t_n,\ X_n)h_n} \begin{pmatrix} \mathbf{0}_{d\times 1} \\ 0 \\ 1 \end{pmatrix}, \tag{10}$$

with $\Lambda_{1.5}$ is given by

$$\Lambda_{1.5}(t_n,\ X_n) =$$

$$= \begin{pmatrix} f_x(t_n,\ X_n) & f_t(t_n,\ X_n) + \frac{1}{2}\sum_{j=1}^{m}(I_{d\times d} \otimes g^j(t_n)^\top) f_{xx}(t_n,\ X_n) g^j(t_n) & f(t_n,\ X_n) \\ \mathbf{0}_{1\times d} & 0 & 1 \\ \mathbf{0}_{1\times d} & 0 & 0 \end{pmatrix} \tag{11}$$

Now, for $\xi_{1.5}$ of (6) the Itô-Taylor expansion of order 1.5 is considered only for the stochastic part of (4), this way, discarding the rest r_3

$$\begin{aligned} \xi_{1.5}(t_n,\ X_n; h_n) &= G(t_n) \underbrace{\int_{t_n}^{t_{n+1}} dW(s)}_{=\Delta W_n} + f_x(t_n,\ X_n)G(t_n) \underbrace{\int_{t_n}^{t_{n+1}} \int_{t_n}^{s} dW(u)ds}_{=\Delta Z_n} + \\ &\quad + G_t(t_n)h_n \underbrace{\int_{t_n}^{t_{n+1}} dW(s)}_{=\Delta W_n} - G_t(t_n) \underbrace{\int_{t_n}^{t_{n+1}} \int_{t_n}^{s} dW(u)ds}_{=\Delta Z_n}, \end{aligned} \tag{12}$$

where

$$\Delta W_n = \int_{t_n}^{t_{n+1}} dW(s) \quad \text{and} \quad \Delta Z_n = \int_{t_n}^{t_{n+1}} \int_{t_n}^{s} dW(u)ds, \tag{13}$$

then

$$\begin{aligned} \xi_{1.5}(t_n,\ X_n; h_n) &= G(t_n)\Delta W_n + f_x(t_n,\ X_n)G(t_n)\Delta Z_n + \\ &\quad + G_t(t_n)(\Delta W_n h_n - \Delta Z_n), \end{aligned} \tag{14}$$

this way, the *exponential scheme of SLL1.5* is defined from (6), (10) and (14). From [7] is known that *SLL1* and *SLL1.5* attain stochastic strong order of convergence of 1 and 1.5 respectively, as their names suggest, and a deterministic strong order of convergence of 2. In [5], this author used the two methods above to build an adaptive LL scheme for SDEs, and it gave good results in terms of efficiency, and accuracy, but it fails when simulating SDEs where the stochastic part became too small because when $g = 0$, both schemes (*SLL1*, and *SLL1.5*) became the same, and this will make the adaptive algorithm fail, this would be clear when the adaptive algorithm be presented, but that is the motivation behind the SLL scheme presented below.

2.2 Strong local Linearization scheme of order (3.0, 1.5)

In order to improve the *SLL1.5* method described above, the same idea presented in [3] for the LL scheme for ODEs was used here, which involves adding part of the deterministic rest to the LL approximation of the SDE. This way we introduce the following new *SLL* scheme for SDEs

$$\textit{SLL (3.0, 1.5)} \quad X_{n+1} = X_n + \phi_{1.5}(t_n,\ X_n; h_n) + \xi_{1.5}(t_n,\ X_n; h_n), \tag{15}$$

$$\phi_{(3,\ 1.5)}(t_n,\ X_n; h_n) = \begin{pmatrix} I_{d\times d} & \mathbf{0}_{d\times 3} \end{pmatrix} e^{\Lambda_{(3,\ 1.5)}(t_n,\ X_n)h_n} \begin{pmatrix} \mathbf{0}_{d\times 1} & 0 & 0 & 1 \end{pmatrix}^\top, \tag{16}$$

where

$$\Lambda_{(3,\ 1.5)}(t_n,\ X_n) = \begin{pmatrix} f_x(t_n,\ X_n) & a_2 & a_1 & f(t_n,\ X_n) \\ \mathbf{0}_{1\times d} & 1 & 0 & 0 \\ \mathbf{0}_{1\times d} & 0 & 1 & 0 \\ \mathbf{0}_{1\times d} & 0 & 0 & 0 \end{pmatrix}, \tag{17}$$

with

$$a_1 = f_t(t_n,\ X_n) + \frac{1}{2}\sum_{j=1}^{m}(I_{d\times d} \otimes g^j(t_n)^\top) f_{xx}(t_n,\ X_n) g^j(t_n) \tag{18}$$

$$\begin{aligned} a_2 &= i_{d\times d} \otimes f^\top(t_n,\ X_n) f_{xx}(t_n,\ X_n) f(t_n,\ X_n) + f_{tt}(t_n,\ X_n) + \\ &\quad + 2 f_{xt}(t_n,\ X_n) f(t_n,\ X_n). \end{aligned} \tag{19}$$

From theorem 4.2 in [4], can be proved that it attains a deterministic order of convergence of 3, and strong order of convergence of 1.5 in the stochastic sense.

2.3 A-Stability of SLL schemes

Theorem 2.1. *The Local Linearization schemes (in particular SLL1, SLL1.5, and SLL(3, 1.5)) are A-stable.*

Proof. Given a test equation

$$dX(t) = \lambda X(t)dt + dW(t), \tag{20}$$

the ODE of its deterministic part is given by

$$dX(t) = \lambda X(t)dt, \tag{21}$$

this way, both equations (3) and (4) became

$$dX(t) = \lambda X(t)dt + G(t)dW(t), \tag{22}$$

So, all the LL discretizations of (20) could be represented by

$$X_{n+1} = X_n + \phi(t_n, X_n) + \xi_n(t_n) \tag{23}$$

where $X_n + \phi(t_n,\ X_n)$ is a step of a LL discretization for the ODE (21), and $\xi_n(t_n)$ is a step from some discretization of the stochastic problem (22). From [2] we know that the LL schemes are A-stable for ODEs and, how ξ_n doesn't depend on X_n for all n, this implies that the SLL schemes are A-stables. $\square$

3 Adaptive strategy

New step selection (Adapted from [1])

Given the approximations y_n^1 and $y_n^{1.5}$ characterized by (5) and (15) respectively, at time t_n and a timestep h

1. **Compute the approximations** y_{n+1}^1 and $y_{n+1}^{1.5}$ at $t_{n+1} = t_n + h$.

2. **Evaluate the Error**

$$error = \sqrt{\frac{1}{d}\sum_{i=1}^{d}\left(\frac{y_n^{1.5} - y_n^1}{tol_i}\right)^2}. \tag{24}$$

3. **Compute the new timestep.**

$$h_{new} \quad = \quad h\sqrt{\left(\frac{1}{error}\right)}, \tag{25}$$

   where h is the size of the last step given.

4. **Validation of $y_n^{1.5}$.** If $error \geqslant 1$ and $h - h_{\min} > 0$, $y_n^{1.5}$ is rejected, and return to Step 1, and set $h = h_{new}$

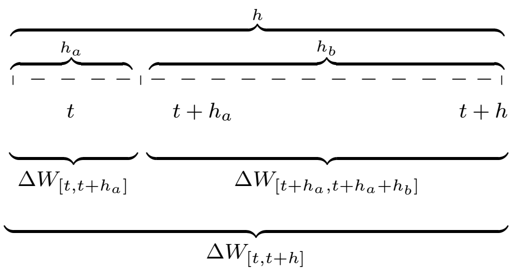
   Else, the step $y_n^{1.5}$ is accepted at the instant t_{n+1}, and **return** to step 1, with $n = n + 1$ and $h = h_{new}$.

5. **Final timestep control.** If $t_{n+1} = T$, the algorithm stops. If $t_{n+1} + h_{new} > T$, recompute $h_{new} = T - t_{n+1}$.

Note that the error estimation (24) will become zero if the approximations y_{n+1}^1 and $y_{n+1}^{1.5}$ where the same, if this happens for every n this will lead to the acceptance of every step tried.

3.1 Reconstructing the Brownian Path

When a step is rejected in the previous algorithm the Brownian path is updated, and this generated random number cannot be ignored, but the next step to try will fall before the last value of the Brownian path, as can be seen in the following diagram



To solve this problem, the solution presented in [1] was used with a small improvement, here, the Brownian path is updated for every tried step, when the original only updates for the accepted steps, and for the first step rejected.

4 Embedding Stochastic LL schemes

In order to create useful adaptive schemes, they need to be computationally efficient and, when LL schemes are used this becomes quite complicated because on each attempt to take a step, is needed to compute two exponential of matrices, which is computationally costly. In [5] the author presented a way to embed the schemes *SLL1* and *SLL1.5*, which later evolved to a published paper [6], where presented a more general approach to embed two exponential schemes. This approach consists in alter the Padé algorithm[3] to compute the exponential of a matrix to compute both exponentials in only one.

5 Simulations

Example 5.1. *Given the following SDE*

$$d\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = \left[A_2 \cdot \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + \begin{pmatrix} 2sen(t) \\ 998(\cos(t) - sen(t)) \end{pmatrix}\right] dt + e^{A_2 t}\begin{pmatrix} 1 \\ 1 \end{pmatrix} dW(t) \tag{26}$$

where $A_2 = \begin{pmatrix} -2 & 1 \\ 998 & -999 \end{pmatrix}$, *with initial condition* $\begin{pmatrix} y_1(0) \\ y_2(0) \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$.

And solution

$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = 2e^{-t}.\begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} sen(t) \\ \cos(t) \end{pmatrix} + e^{-t}.\begin{pmatrix} 1 \\ 1 \end{pmatrix}.W(t). \tag{27}$$

Experiment: First will be computed the solution with the adaptive scheme presented here with tolerance $tol = 10^{-4}$, them, a solution with fixed stepsizes will be computed with a number of steps equal to the sum of accepted and rejected steps that came out of the computation of the solution with the adaptive scheme (to make a fair competition). And both will run over the same Brownian path. This was repeated 2000 times to take the averages of error and accepted and

[3]The Padé algorithm is one of the standard algorithms to compute the exponential of matrices numerically.

rejected steps, where the error is given by $error = \max\left\{\frac{\sum_{i=0}^{n}\sqrt{X_i - \hat{X}_i}}{n}\right\}$. The results are in Table 1.

Table 1: Experiment Results.

	Error ± sd	Number of Steps (Accept./Rejec.)	Comp. time
Adap. SLL	$2.3809 \times 10^{-6} \pm 8.0 \times 10^{-7}$	$974.8 \pm 30.9 \mid 159.2 \pm 11.9$	0.30 ± 0.0353
SLL (3, 1.5)	$6.3604 \times 10^{-4} \pm 3.1 \times 10^{-4}$	$1134.1 \pm 40.2 \mid -$	0.47 ± 0.0741

6 Summary and conclusions

This scheme uses a state-of-the-art Local Linearization approach to construct stable schemes that are embeddable in a non-conventional way using what was presented in [6], and has a well-proved variable-stepsize strategy while keep track of the all Brownian path generated during the simulation. The results shown that the adaptive SLL scheme presented has great performance for the example 5.1, achieving an error three orders of magnitude smaller than the same method with fixed stepsizes as can be seen in Table 1.

References

[1] Pamela Marion Burrage and Kevin Burrage. "A Variable Stepsize Implementation for Stochastic Differential Equations". In: **SIAM Journal on Scientific Computing** 24.3 (2002), pp. 848–864.

[2] Hugo de la Cruz et al. "Local Linearization Runge–Kutta methods: A class of A-stable explicit integrators for dynamical systems". In: **Mathematical and Computer Modelling** 57.3 (2013), pp. 720–740.

[3] Hugo De la Cruz et al. "A higher order local linearization method for solving ordinary differential equations". In: **Applied mathematics and computation** 185.1 (2007), pp. 197–212.

[4] Hugo De la Cruz Cancino et al. "High order local linearization methods: An approach for constructing A-stable explicit schemes for stochastic differential equations with additive noise". In: **BIT Numerical Mathematics** 50 (2010), pp. 509–539.

[5] Pablo Aguiar De Maio. "Um método de linearização local com passo adaptativo para solução numérica de equações diferenciais estocásticas com ruído aditivo". Master dissertation. Escola de Matemática Aplicada da Fundação Getúlio Vargas, 2015.

[6] Juan Carlos Jimenez, Hugo de la Cruz, and Pablo Aguiar De Maio. "Efficient computation of phi-functions in exponential integrators". In: **Journal of Computational and Applied Mathematics** 374 (2020), p. 112758.

[7] Juan Carlos Jimenez and Hugo de la Cruz Cancino. "Convergence rate of strong Local Linearization schemes for stochastic differential equations with additive noise". In: **BIT Numerical Mathematics** 52.2 (2012), pp. 357–382.

[8] Peter E Kloeden and Eckhard Platen. **Numerical Solution of Stochastic Differential Equations**. Springer, 1999.