

Computer simulation of the stochastic transport equation

H. de la Cruz^{a,*}, C. Olivera^b, J. P. Zubelli^c

^a*School of Applied Mathematic, EMAP-FGV, RJ, Brasil*

^b*IMECC, State University of Campinas, RJ, Brasil*

^c*IMPA, Estrada Dona Castorina 110, RJ, Brasil*

Abstract

In this article the numerical approximation of the stochastic transport equation is considered. We propose a new computational scheme for the effective simulation of the solutions of this equation. Results on the convergence of the suggested scheme and details on its efficient implementation is presented. The performance of the introduced method is illustrated through computer simulations.

Keywords:

Partial Stochastic differential equations, random differential equations, numerical methods, stochastic transport equation, exponential methods, local linearization approach.

1. Introduction

Stochastic partial differential equations (SPDEs) are widely used as mathematical models of complex systems under random influences. Many problems in fields like physics, biology, oceanography and in areas including wave propagation, material science, evolution of biological populations, fluid dynamics among others, can be better understood using SPDEs.

In particular, the stochastic transport equation

$$\begin{cases} \frac{d}{dt}u(t, x) + b(t, x)\nabla u(t, x) + \nabla u(t, x) \frac{dB_t}{dt} = 0 \\ u(0, x) = u_0(x), \end{cases} \quad (1)$$

(where $B_t = (B_t^1, \dots, B_t^d)$ is a standard Brownian motion in \mathbb{R}^d and the stochastic integration is taken in the Stratonovich sense) arises as a prototype model in understanding a wide variety of phenomena. It appears in modeling transport in porous media, suspended sediment transport in open-channel flows, advective transport of substances, waves motions in random media and from turbulent transport theory, $u(x, t)$ be the concentration of a passive substance convected by a turbulent fluid, such as smoke in turbulent air.

The existence, uniqueness, and properties of the solutions of such equations have been well studied for the case of classical solutions in [21] and [23] via the stochastic characteristic method and for the case of L^∞ - solutions in [11] and L^p - solutions in [5]. In [1] some partial results are presented, regarding the case $L^1(\mathbb{R}^d) \cap L^\infty(\mathbb{R}^d)$. Since in general it is not possible to derive an analytical solution for (1), the accurate simulation of this equation is essential in order to get a better understanding and a valuable information of the phenomena in study.

*Corresponding author

Email addresses: hugo.delacruz@fgv.br (H. de la Cruz), colivera@ime.unicamp.br (C. Olivera), zubelli@impa.br (J. P. Zubelli)

¹partially supported by the FGV/EMAP project "Toolbox for the simulation and estimation of SDEs"

In the scenario of the numerical simulation of transport equation there are some related works but with focus on the numerical solution of the unidirectional random transport equation, which essentially is a deterministic transport equation with the velocity and/or the initial condition been stochastic input parameters (see for instance [9], [10], [26], where numerical methods are designed to compute the expectation, variance and auto-correlation function of the solution of this type of random transport equations). However, as far as we know, the accurate simulation of the strong solution of the Stratonovich transport equation (1) has not been considered so far. In the present work we attempt to go forward in this direction by proposing a new numerical method for the computational simulation of trajectories of (1).

A lot of attention has recently been paid to numerical methods for stochastic partial differential equations (SPDEs). Various numerical methods and approximation schemes for SPDEs have also been developed, analyzed, and tested, e.g., in [2], [13], [14], [16] [24] and [27] (see also the references therein). We recall that in [24] the stochastic characteristics method is used to approximate solutions of SPDEs of parabolic type.

Our aim here is to exploit the stochastic characteristics method and numerical integration of backward stochastic differential equations together with the local linearization technique to construct an integrator to (1). That is, the stochastic characteristic method provides a representation for the solution of the stochastic transport equation through the solution of a corresponding backward random equation, which can be transformed to a random integral equation. Thus, the integration of (1) is reduced to the integration of a particular integral RDE of the form

$$\mathbf{R}(t) = \mathbf{R}(t_0) - \int_{t_0}^t \mathbf{f}(s, \mathbf{R}(s)) ds + \xi_T - \xi_{T-t},$$

which in contrast with stochastic differential equations can be understood -realization by realization- as a deterministic equation. Thus, this is formulated pathwise, without the necessity of using stochastic calculus. For solving RDEs, in general, it is needed to resort to approximation methods to determine its solution.

At a first look one could think that for solving the RDE above, some of the existing computational methods for RDEs could be used (see e.g., [4], [17], [18]), but because of the particular structure of this equation, $\mathbf{R}(t)$ is obviously not differentiable in t , so it cannot be rewritten as a usual RDE. Hence, we need to construct a new numerical integrator for this kind of equations. For doing this the approach we follow here is strongly motivated by the ideas behind the so called local linearization technique, which has been successfully applied to the construction of exponential-based stable methods for a variety of differential equations including ODEs ([19], [8]), SDEs ([3], [7]) and RDEs ([4]).

We elaborate the following strategy: at each time interval of integration, the vector field \mathbf{f} is locally approximated through a first order Taylor expansion and the stochastic term is approximated by a polygonal obtained by a suitable interpolation. In this way successive Caratheodory RDEs are obtained and their solutions can be explicitly expressed in terms of a single matrix exponential times a vector. This exponential representation is a key point in the design of feasible computational algorithms implementing the method. The reason is that the involved matrix has a particular structure that allows to derive an algorithm based on the Padé method with scaling-squaring strategy in such a way that the overall computational saving achieved are significant and consequently resulting in an accurate and stable numerical schemes. It is proved that the rate of convergence of this method will be determined by the moduli of continuity of Wiener processes.

The paper is organized as follows. Section 2 reviews the stochastic characteristics method, furthermore a representation for the solution of the stochastic transport equation, through the solution of determined integral RDE, is given. Section 3 introduce a numerical method for this type of random equations and the local and global truncation error of the method are studied. Section 4 deals with the implementation of the proposed method for the stochastic transport equation. In this Section a Padé algorithm with scaling-squaring strategy is conveniently adapted for implementing the proposed integrator. Finally, Section 5 illustrates the practical performance of the numerical method through computer experiments.

2. Preliminary

In this section we present the so called stochastic characteristics method, that we will use as a key tool to design our method. Then we transform the associated backward stochastic differential equation in a convenient random integral differential equation.

2.1. Stochastic transport equation

Let us start by setting the notation used and then recalling the main results.
The equation (1) is interpreted as the following stochastic integral equation

$$u(t, x) = u_0(x) - \int_0^t b(s, x) \nabla u(s, x) ds - \sum_{i=0}^d \int_0^t u(s, x) \circ dB_s^i \quad (2)$$

Given $s \in [0, T]$ and $x \in \mathbb{R}^d$, consider the following stochastic differential equation in \mathbb{R}^d

$$X_{s,t} = x + \int_s^t b(r, X_{s,r}) dr + B_t - B_s. \quad (3)$$

We shall assume that

$$b \in L^2([0, T], C^{m,\delta}(\mathbb{R}^d)) \quad (4)$$

It is well known that under conditions (4), $X_{s,t}(x)$ is a stochastic flow of C^m -diffeomorphism (see for example [22] and [23]). Moreover, the inverse $Y_{s,t} := X_{s,t}^{-1}(x)$ satisfies the following backward stochastic differential equation

$$Y_{s,t} = x - \int_s^t b(r, Y_{r,t}) dr - (B_t - B_s). \quad (5)$$

for $0 \leq s \leq t$, see [11] or [22] pp. 234.

Lemma 1. *Let $u_0 \in C^{m,\delta}(\mathbb{R}^d)$ and assume (4) for $m \geq 3$. Then the Cauchy problem (2) has a unique solution $u(t, \cdot)$ for $0 \leq t \leq T$ such that it is a C^m -semimartingale which can be represented as*

$$u(t, x) = u_0(X_t^{-1}(x)) \quad (6)$$

Proof. See for instance [22] and [23]. □

2.2. A RDE for the stochastic flow.

With the purpose to treat numerically the transport equation (2) we rewrite the backward stochastic differential equation (5) conveniently. First fix t and we make the change of variable $s = t - u$, thus we have that

$$Y(t - u) = x - \int_{t-u}^t b(r, Y_r) dr - (B_t - B_{t-u}).$$

Now, we denote $R(\tau) = Y(t - \tau)$ and we have that $R(\tau)$ satisfies the following equation

$$R(\tau) = x - \int_{t-\tau}^t b(r, R_{t-\tau}) dr - (B_t - B_{t-\tau}).$$

Making the change of variable $t - r = s$ we get

$$R(\tau) = x - \int_0^\tau b(t - \tau, R_s) ds - (B_t - B_{t-\tau}).$$

Finally it holds that

$$R(\tau) = x - \int_0^\tau \mathbf{f}(s, R_s) ds - (B_t - B_{t-\tau}).$$

where $\mathbf{f}(s, \cdot) = b(t - s, \cdot)$.

3. Proposed method for the random integral equation

Let (Ω, \mathcal{F}, P) be a complete probability space, and $(\mathcal{F}_t)_{t \geq 0}$ be an increasing right continuous family of complete sub σ -algebras of \mathcal{F} . Consider the d -dimensional random integral equation

$$\mathbf{R}(t) = \mathbf{R}(t_0) - \int_{t_0}^t \mathbf{f}(s, \mathbf{R}(s)) ds + \xi_T - \xi_{T-t}, \quad t \in [t_0, T], \quad (7)$$

where ξ_t is a \mathcal{F}_t -adapted finite continuous processes. It is assumed that $\mathbf{f} \in \mathcal{C}^1(\mathbb{R} \times \mathbb{R}^d, \mathbb{R}^d)$.

3.1. Formulation of the method

Let $(t)_h = \{t_n : n = 0, 1, \dots, N\}$ be a partition of the time interval $[t_0, T]$, with equidistant stepsize $h < 1$, i.e., defined as a sequence of times $t_0 < t_1 < \dots < t_N = T$ such that $t_n = t_0 + nh$, for $n = 0, 1, \dots, N$. Note that from (7), for $t \in [t_n, t_{n+1}]$

$$\mathbf{R}(t) = \mathbf{R}(t_n) - \int_{t_n}^t \mathbf{f}(s, \mathbf{R}(s)) ds + \xi_{T-t_n} - \xi_{T-t}$$

Define $\mu(t) = \xi_{T-t}$, starting from the initial value $\mathbf{R}_0 = \mathbf{R}(t_0)$, the approximations $\{\mathbf{R}_i\}$ to $\{\mathbf{R}(t_i)\}$, ($i = 1, 2, \dots, N$) are obtained recursively as follows.

For each time interval $\mathbf{\Lambda}_n = [t_n, t_{n+1}]$ we consider the random local problem

$$\mathbf{R}(t) = \mathbf{R}_n - \int_{t_n}^t \mathbf{f}(s, \mathbf{R}(s)) ds + \mu(t_n) - \mu(t) \quad (8)$$

Then, the idea is to get an approximation of $\mathbf{R}(t_{n+1})$, through the solution of the auxiliary equation resulting from approximating \mathbf{f} and the stochastic increment in (8).

For this, let's consider $\bar{h} = h^\gamma$ with $\gamma \geq 4$ and such that $h^{1-\gamma} \in \mathbb{N}$ (we need to take \bar{h} in this way in order to guaranty the convergence of the method we are constructing here. It will be clarified in the Section 3) and let $(t_n)_{\bar{h}} = \{t_n^{(i)} : t_n^{(i)} = t_n + i\bar{h}, i = 0, 1, \dots, [h^{1-\gamma}] + 1\}$ a partition of $\mathbf{\Lambda}_n$.

For $t \in \mathbf{\Lambda}_n$, define $n_t = \max\{k : t_n^{(k)} \leq t < t_n^{(k+1)}\}$ then

$$\xi_{T-t_n} - \xi_{T-t} = (\xi_{T-t_n} - \xi_{T-t_n^{(n_t)}}) + (\xi_{T-t_n^{(n_t)}} - \xi_{T-t}),$$

thus, by a linear interpolation to $\mu(t) = \xi_{T-t}$ in $[t_n^{(n_t)}, t_n^{(n_t+1)}]$

$$\mu(t) \approx \bar{\mu}(t) = \mu(t_n^{(n_t)}) + \frac{\mu(t_n^{(n_t+1)}) - \mu(t_n^{(n_t)})}{\bar{h}}(t - t_n^{(n_t)}), \quad (9)$$

so,

$$\begin{aligned} \xi_{T-t_n} - \xi_{T-t} &= (\xi_{T-t_n} - \xi_{T-t_n^{(n_t)}}) + (\xi_{T-t_n^{(n_t)}} - \xi_{T-t}) \\ &= (\xi_{T-t_n} - \xi_{T-t_n^{(n_t)}}) + \mu(t_n^{(n_t)}) - \mu(t) \\ &\approx (\xi_{T-t_n} - \xi_{T-t_n^{(n_t)}}) + \frac{\mu(t_n^{(n_t)}) - \mu(t_n^{(n_t+1)})}{\bar{h}}(t - t_n^{(n_t)}) \\ &= (\xi_{T-t_n} - \xi_{T-t_n^{(n_t)}}) + \frac{\xi_{T-t_n^{(n_t)}} - \xi_{T-t_n^{(n_t+1)}}}{\bar{h}}(t - t_n^{(n_t)}) \\ &= (\xi_{T-t_n} - \xi_{T-t_n^{(n_t)}}) + \frac{\xi_{T-t_n^{(n_t)}} - \xi_{T-t_n^{(n_t+1)}}}{\bar{h}}(t - t_n^{(n_t)}) \end{aligned}$$

On the other hands from a first order Taylor expansion of \mathbf{f} at (t_n, \mathbf{R}_n) :

$$\mathbf{f}(s, \mathbf{R}(s)) \approx \mathbf{f}(t_n, \mathbf{R}_n) + \mathbf{f}_{\mathbf{x}}(t_n, \mathbf{R}_n)(\mathbf{R}(s) - \mathbf{R}_n) + \mathbf{f}_t(t_n, \mathbf{R}_n)(s - t_n),$$

Then it follows, by using the approximations above, that the solution of (8) in Λ_n can be approximated by the solution of

$$\begin{aligned} \mathbf{R}(t) &= \mathbf{R}_n - \int_{t_n}^t (\mathbf{f}(t_n, \mathbf{R}_n) + \mathbf{f}_{\mathbf{x}}(t_n, \mathbf{R}_n)(\mathbf{R}(s) - \mathbf{R}_n) + \mathbf{f}_t(t_n, \mathbf{R}_n)(s - t_n)) ds \\ &\quad + (\xi_{T-t_n} - \xi_{T-t_n^{(n_t)}}) + \frac{\xi_{T-t_n^{(n_t)}} - \xi_{T-t_n^{(n_t+1)}}}{\bar{h}}(t - t_n^{(n_t)}) \end{aligned}$$

or, equivalently,

$$\begin{aligned} \mathbf{R}(t) &= \mathbf{R}_n - \int_{t_n}^t (\mathbf{f}(t_n, \mathbf{R}_n) + \mathbf{f}_{\mathbf{x}}(t_n, \mathbf{R}_n)(\mathbf{R}(s) - \mathbf{R}_n) + \mathbf{f}_t(t_n, \mathbf{R}_n)(s - t_n)) ds \\ &\quad + \sum_{k=0}^{[h^{1-\gamma}]} \left((\xi_{T-t_n} - \xi_{T-t_n^{(k)}}) + \frac{\xi_{T-t_n^{(k)}} - \xi_{T-t_n^{(k+1)}}}{\bar{h}}(t - t_n^{(k)}) \right) \mathbf{1}_{[t_n^{(k)} \ t_n^{(k+1)}]}(t) \end{aligned} \quad (10)$$

Now, by differentiation, we obtain the (pathwise) Caratheodory differential equation

$$\mathbf{R}'(t) = -\mathbf{f}_{\mathbf{x}}(t_n, \mathbf{R}_n)(\mathbf{R}(t) - \mathbf{R}_n) - \mathbf{f}_t(t_n, \mathbf{R}_n)(t - t_n) + \left(\mathbf{f}(t_n, \mathbf{R}_n) + \sum_{k=0}^{[h^{1-\gamma}]} \frac{\xi_{T-t_n^{(k)}} - \xi_{T-t_n^{(k+1)}}}{\bar{h}} \mathbf{1}_{[t_n^{(k)} \ t_n^{(k+1)}]}(t) \right) \quad (11)$$

$$\mathbf{R}(t_n) = \mathbf{R}_n,$$

$$\forall t \in \{\Lambda_n \setminus (t_n)_{\bar{h}}\}.$$

Let's denote

$$\mathbf{A}_n = -\mathbf{f}_{\mathbf{x}}(t_n, \mathbf{R}_n)$$

$$\mathbf{b}_n = -\mathbf{f}_t(t_n, \mathbf{R}_n)$$

$$\mathbf{c}_n^k = \mathbf{f}(t_n, \mathbf{R}_n) - \mathbf{A}_n \mathbf{R}_n - \mathbf{b}_n t_n + \sum_{k=0}^{[h^{1-\gamma}]} \frac{\xi_{T-t_n^{(k)}} - \xi_{T-t_n^{(k+1)}}}{\bar{h}} \mathbf{1}_{[t_n^{(k)} \ t_n^{(k+1)}]}(t)$$

Since the inhomogeneity in (11) is discontinuous in $\{t_n^k\}$, the solution of (11) at $t = t_{n+1}$ is obtained as follows. Starting at $\mathbf{R}(t_n^0) = \mathbf{R}(t_n) = \mathbf{R}_n$, $\mathbf{R}(t_n^{k+1})$ is computed recursively from $\mathbf{R}(t_n^k)$ by solving in $[t_n^k, t_{n+1}^{k+1}]$ the linear differential equation

$$\mathbf{R}'(t) = \mathbf{A}_n \mathbf{R}(t) + \mathbf{b}_n t + \mathbf{c}_n^k \quad (12)$$

with initial condition $\mathbf{R}(t_n^k)$ in $t = t_n^k$.

That is,

$$\begin{aligned} \mathbf{R}(t_n^{k+1}) &= e^{\mathbf{A}_n(t - t_n^k)} \mathbf{R}(t_n^k) + \int_{t_n^k}^{t_n^{k+1}} e^{\mathbf{A}_n(t-s)} (\mathbf{b}_n s + \mathbf{c}_n^k) ds \\ &= \mathbf{R}(t_n^k) + \int_0^{\bar{h}} e^{\mathbf{A}_n(\bar{h}-s)} (\mathbf{b}_n s + \mathbf{d}_n^k) ds \\ &:= \varphi(\mathbf{R}(t_n^k)) \end{aligned} \quad (13)$$

where

$$\mathbf{d}_n^k = \mathbf{c}_n^k + \mathbf{A}_n \mathbf{R}(t_n^k) + \mathbf{b}_n t_n^k$$

As we want to approximate $\mathbf{R}(t_{n+1})$, we conclude that

$$\begin{aligned} \mathbf{R}_{n+1} &= \varphi^{([h^{1-\gamma}] + 1)}(\mathbf{R}_n) \\ &= \underbrace{\varphi \circ \varphi \circ \dots \circ \varphi}_{([h^{1-\gamma}] + 1) \text{ times}}(\mathbf{R}_n) \end{aligned}$$

Summarizing, the numerical integrator for (7) is given by the recursive equation

$$\mathbf{R}_{n+1} = \varphi^{([h^{1-\gamma}] + 1)}(\mathbf{R}_n) \quad (14)$$

for $n = 0, 1, \dots, N - 1$ with $\mathbf{R}(t_0) = \mathbf{R}_0$ and $\gamma \geq 2$.

An important problem in the evaluation of (14) is the efficient computation of φ . A naive way to do this is thought the explicit computation of the integral in (13). However this procedure might eventually fail, since it is not computationally feasible in case of singular or poor-condicioned matrix \mathbf{A}_n . In section 4 we will propose an efficient algorithm for computing \mathbf{R}_{n+1} .

3.2. Convergence analysis

In this section the convergence and velocity of convergence of the method (14) is studied. The following lemma will be useful for deriving the main results of this section.

Lemma 2. *Let*

$$\varpi_\mu(\bar{h}) = \sup_{|t-s| \leq \bar{h}} \|\mu(t) - \mu(s)\|$$

be the modeli of continuity of μ and let

$$\bar{\mu}(t_n^k + u) = \mu(t_n^k) + \frac{\mu(t_n^{k+1}) - \mu(t_n^k)}{\bar{h}} u$$

be the linear interpolation to μ for all $u \in [t_n^k, t_n^{k+1}]$. Then

$$\sup_{0 \leq u \leq \bar{h}} \|\mu(t_n^k + u) - \bar{\mu}(t_n^k + u)\| \leq 2\varpi_\mu(\bar{h})$$

Proof. We have that

$$\begin{aligned} \sup_{0 \leq u \leq \bar{h}} \|\mu(t_n^k + u) - \bar{\mu}(t_n^k + u)\| &= \sup_{0 \leq u \leq \bar{h}} \left\| \mu(t_n^k + u) - \mu(t_n^k) - \frac{\mu(t_n^{k+1}) - \mu(t_n^k)}{\bar{h}} u \right\| \\ &\leq \sup_{0 \leq u \leq \bar{h}} \|\mu(t_n^k + u) - \mu(t_n^k)\| + \sup_{0 \leq u \leq \bar{h}} \left\| \frac{\mu(t_n^{k+1}) - \mu(t_n^k)}{\bar{h}} u \right\| \\ &\leq 2\varpi_\mu(\bar{h}) \end{aligned}$$

□

Next theorem provides a local error of convergence

Theorem 3. *Let $L_{n+1} = \|\mathbf{R}(t_{n+1}) - \varphi^{([h^{1-\gamma}] + 1)}(\mathbf{R}(t_n))\|$ be the local error of the approximation to the exact solution $\mathbf{R}(t)$ at time t_{n+1} and let $\varpi_\mu(\bar{h}) = O(\bar{h}^\beta)$. Then, there exist a constant K , not depending on h such that*

$$L_{n+1} \leq K h^p,$$

with $p = \min(\gamma\beta, 3)$.

Proof. Let us consider the following equations in $[t_n, t_{n+1}]$

$$\mathbf{x}(t) = \mathbf{R}_n - \int_{t_n}^t \mathbf{f}(s, \mathbf{x}(s)) ds - (\mu(t) - \mu(t_n)) \quad (15)$$

$$\mathbf{u}(t) = \mathbf{R}_n - \int_{t_n}^t \mathbf{f}(s, \mathbf{u}(s)) ds - (\bar{\mu}(t) - \mu(t_n)) \quad (16)$$

$$\mathbf{y}(t) = \mathbf{R}_n - \int_{t_n}^t (\mathbf{f}(t_n, \mathbf{R}_n) + \mathbf{f}_{\mathbf{x}}(t_n, \mathbf{R}_n)(\mathbf{R}(s) - \mathbf{R}_n) + \mathbf{f}_t(t_n, \mathbf{R}_n)(s - t_n)) ds - (\bar{\mu}(t) - \mu(t_n)) \quad (17)$$

with $\bar{\mu}(t)$ as defined in (9).

From (15) and (16) and assuming \mathbf{f} is Liptshitz continuous, it is obtained that

$$\begin{aligned} \|\mathbf{x}(t) - \mathbf{u}(t)\| &\leq \int_{t_n}^t \|\mathbf{f}(s, \mathbf{x}(s)) - \mathbf{f}(s, \mathbf{u}(s))\| ds + \|\mu(t) - \bar{\mu}(t)\| \\ &\leq L \int_{t_n}^t \|\mathbf{x}(t) - \mathbf{u}(t)\| ds + C\bar{h}^\beta, \end{aligned}$$

since $L > 0$ and $C\bar{h}^\beta$ is a constant, from Gronwall inequality we conclude that

$$\|\mathbf{x}(t) - \mathbf{u}(t)\| \leq C\bar{h}^\beta e^{K(t-t_n)} \leq C\bar{h}^\beta e^{Kh} = O(\bar{h}^\beta) = O(h^{\gamma\beta}). \quad (18)$$

On the other hand, since $\forall t \in \{\Lambda_n \setminus (t_n)_{\bar{h}}\}$ the function $\bar{\mu}(t)$ is differentiable, then $\mathbf{u}(t)$ and $\mathbf{y}(t)$ satisfy

$$\begin{aligned} \mathbf{u}'(t) &= \mathbf{f}(s, \mathbf{u}(t)) + \frac{\mu(t_n^{(n_t+1)}) - \mu(t_n^{(n_t)})}{\bar{h}} \\ \mathbf{u}(t_n) &= \mathbf{R}_n \end{aligned}$$

and

$$\begin{aligned} \mathbf{y}'(t) &= \mathbf{f}(s, \mathbf{y}(t)) + \frac{\mu(t_n^{(n_t+1)}) - \mu(t_n^{(n_t)})}{\bar{h}} \\ \mathbf{y}(t_n) &= \mathbf{R}_n. \end{aligned}$$

From this equations, using the rules of the matrix differential calculus it can be shown that

$$\begin{aligned} \mathbf{u}'(t_n) &= \mathbf{y}'(t_n) \\ \mathbf{u}''(t_n) &= \mathbf{y}''(t_n), \end{aligned}$$

thus, applying the Taylor formula to $\mathbf{u}(t) - \mathbf{y}(t)$ it is obtained that

$$\|\mathbf{u}(t) - \mathbf{y}(t)\| \leq \left\| \mathbf{x}(t_n) - \mathbf{u}(t_n) + \left(\mathbf{u}'(t_n) - \mathbf{y}'(t_n) \right) (t - t_n) + \left(\frac{\mathbf{u}''(t_n) - \mathbf{y}''(t_n)}{2} \right) (t - t_n)^2 \right\| \quad (19)$$

$$\begin{aligned} &+ \frac{1}{6} \sup_{\theta \in [t_n, t_{n+1}]} \left\| \mathbf{u}'''(\theta) - \mathbf{y}'''(\theta) \right\| \left| (t - t_n)^3 \right| \\ &= O(h^3) \end{aligned} \quad (20)$$

Finally, from (18) and (19) it is obtained that

$$\begin{aligned} \|\mathbf{x}(t) - \mathbf{y}(t)\| &\leq \|\mathbf{x}(t) - \mathbf{u}(t)\| + \|\mathbf{u}(t) - \mathbf{y}(t)\| \\ &\leq O(h^{\gamma\beta}) + O(h^3) = O^{\min(\gamma\beta, 3)} \end{aligned}$$

hence, $L_{n+1} \leq Kh^p$, with $p = \min(\gamma\beta, 3)$

□

The next theorem deals with the convergence and velocity of convergence of the proposed methods

Theorem 4. *Let $\gamma\beta \geq 2$, then the numerical integrator (14) is globally convergent and we have that*

$$\sup_n \|\mathbf{R}(t_n) - \mathbf{R}_n\| = O(h^{p-1}),$$

where $p = \min(\gamma\beta, 3)$.

Proof. Let us define $\phi \equiv \varphi([h^{1-\gamma}] + 1)$ and $E_{n+1} = \|\mathbf{R}(t_{n+1}) - \mathbf{R}_{n+1}\|$, then

$$\begin{aligned} E_{n+1} &= \|\mathbf{R}(t_{n+1}) - \mathbf{R}_n - \phi(t_n, \mathbf{R}_n, h)\| \\ &\leq \|\mathbf{R}(t_{n+1}) - \mathbf{R}(t_n) - \phi(t_n, \mathbf{R}(t_n), h)\| + \|\mathbf{R}(t_n) - \mathbf{R}_n\| + \|\phi(t_n, \mathbf{R}(t_n), h) - \phi(t_n, \mathbf{R}_n, h)\|, \end{aligned}$$

From lemma 1 p. 1289 in [20] we have that $\|\phi(t_n, \mathbf{R}(t_n), h) - \phi(t_n, \mathbf{R}_n, h)\| \leq hK \|\mathbf{R}(t_n) - \mathbf{R}_n\|$ were K is a constant. Inserting this in the expression above and by the theorem of the local error we have that

$$\begin{aligned} E_{n+1} &\leq E_n + L_{n+1} + hKE_n \\ &\leq (1 + hK) E_n + Ch^p, \end{aligned}$$

Thus, we have that

$$\begin{aligned} E_{n+1} &\leq \left(1 + (1 + Kh) + \dots + (1 + Kh)^{n-1}\right) Ch^p \\ &\leq \left(\frac{(1 + Kh)^n - 1}{Kh}\right) Ch^p, \end{aligned}$$

since $1 + Kh < e^{Kh}$ implies that $(1 + Kh)^n < e^{Knh} = e^{TK}$ we conclude that

$$E_{n+1} \leq \left(\frac{e^{TK} - 1}{K}\right) Ch^{p-1}.$$

Since $p - 1 = \min(\gamma\beta - 1, 2)$ and $\gamma\beta \geq 2$, we have convergence of the numerical method, with rate of convergence $p - 1$. \square

4. Implementation details

In this section an efficient computational algorithm to implement (14) is provided. At first we will show that $\mathbf{R}(t_n^{k+1})$ in (13) can be represented in terms of a single appropriated exponential of a matrix. Then we will discuss in detail an efficient and accurate alternative to evaluate this matrix exponential defining $\mathbf{R}(t_n^{k+1})$.

The equation (12) can be rewritten

$$\begin{aligned} \mathbf{R}'(t) &= \mathbf{A}_n(\mathbf{R}(t) - \mathbf{R}_n^k) + \mathbf{b}_n(t - t_n^k) + \mathbf{d}_n^k \\ \mathbf{R}(t_n^k) &= \mathbf{R}_n^k \end{aligned}$$

where \mathbf{R}_n^k is the approximation to $\mathbf{R}(t_n^k)$.

Let $\mathbf{Z}(t) = (\mathbf{R}(t) - \mathbf{R}_n^k, t - t_n^k, \mathbf{d}_n^k, 1)^\top \in \mathbb{R}^{2d+2}$ then

$$\begin{aligned} \mathbf{Z}'(t) &= \mathbf{M} \mathbf{Z}(t) \\ \mathbf{Z}(t_n^k) &= [\mathbf{0}_{1 \times d} \ 0 \ \mathbf{d}_n^k \ 1]^\top, \end{aligned}$$

where

$$\mathbf{M} = \begin{pmatrix} \mathbf{A}_n & \mathbf{b}_n & \mathbf{I}_{d \times d} & \mathbf{0}_{d \times 1} \\ \mathbf{0}_{1 \times d} & 0 & \mathbf{0}_{1 \times d} & 1 \\ \mathbf{0}_{d \times d} & \mathbf{0}_{d \times 1} & \mathbf{0}_{d \times d} & \mathbf{0}_{d \times 1} \\ \mathbf{0}_{1 \times d} & 0 & \mathbf{0}_{1 \times d} & 0 \end{pmatrix}.$$

The solution of this equation is

$$\mathbf{Z}(t) = e^{\mathbf{M}(t-t_n^k)} \mathbf{Z}(t_n^k).$$

Therefore, looking at the first component of $\mathbf{Z}(t)$, it follows that the solution, $\mathbf{R}(t)$, of (12) in $[t_n^k, t_n^{k+1}]$ can be computed by $\mathbf{R}(t) = \mathbf{R}_n^k + [\mathbf{I}_{d \times d} \quad \mathbf{0}_{d \times (d+2)}] e^{\mathbf{M}(t-t_n^k)} [\mathbf{0}_{d \times d} \quad \mathbf{0} \quad \mathbf{d}_n^k \quad 1]^\top$.

Thus, in particular

$$\mathbf{R}(t_n^{k+1}) = \mathbf{R}_n^k + [\mathbf{I}_{d \times d} \quad \mathbf{0}_{d \times (d+2)}] e^{\mathbf{M} \bar{h}} [\mathbf{0}_{d \times d} \quad \mathbf{0} \quad \mathbf{d}_n^k \quad 1]^\top. \quad (21)$$

Thus, the numerical implementation of $\mathbf{R}(t_n^{k+1})$ is reduced to the use of a algorithm to compute exponential of matrices (See [25] for a review). In particular, those algorithms based on the rational (p, q) -Padé approximation ($p \leq q \leq p+2$) combined with the “scaling and squaring” strategy provide stable approximations to the matrix exponential. However, because of the size of the matrix \mathbf{M} , this turns out that a straight forward implementation of the Padé method could be prohibitively expensive.

In the rest of this section we will derive an algorithm that alleviates significantly the computational burden. Our key idea is to exploit the special structure of the matrix \mathbf{M} and to adapt conveniently the “scaling and squaring” strategy in such a way that the computational saving achieved are significant.

Before this, we first summarize the existing Padé algorithm with “scaling and squaring” strategy on which we will work.

4.1. The Padé algorithm for computing the matrix exponential

The (p, q) rational Padé approximation to $e^{\mathbf{C}}$ is defined by

$$\mathbf{P}_{p,q}(\mathbf{C}) = [\mathbf{D}_{p,q}(\mathbf{C})]^{-1} \mathbf{N}_{p,q}(\mathbf{C}),$$

where

$$\mathbf{N}_{p,q}(\mathbf{C}) = \sum_{j=0}^p \frac{(p+q-j)!p!}{(p+q)!j!(p-j)!} \mathbf{C}^j$$

and

$$\mathbf{D}_{p,q}(\mathbf{C}) = \sum_{j=0}^q \frac{(p+q-j)!q!}{(p+q)!j!(q-j)!} (-\mathbf{C})^j.$$

Diagonal approximation (that is, $p = q$) are preferred, since $\mathbf{P}_{p,q}$ with $p > q$ ($p < q$) is less accurate than $\mathbf{P}_{p,p}$ ($\mathbf{P}_{q,q}$), and $\mathbf{P}_{p,p}$ ($\mathbf{P}_{q,q}$) can be evaluated at the same cost. From now on, we denote $\mathbf{D}_{q,q}(\mathbf{C})$, $\mathbf{N}_{q,q}(\mathbf{C})$, $\mathbf{P}_{q,q}(\mathbf{C})$ by $\mathbf{D}_q(\mathbf{C})$, $\mathbf{N}_q(\mathbf{C})$, $\mathbf{P}_q(\mathbf{C})$ respectively.

$e^{\mathbf{C}}$ can be well approximated by Padé approximant only near the origin, that is, for small $\|\mathbf{C}\|$. For this reason $e^{\mathbf{C}}$ is approximated by $(\mathbf{P}_q(\frac{\mathbf{C}}{m}))^m$ where m is the minimum integer such that $\|\frac{\mathbf{C}}{m}\| < \frac{1}{2}$. In order to reduce the number of matrix multiplications, the idea is to choose m to be a power of two. Then $(\mathbf{P}_q(\frac{\mathbf{C}}{m}))^m$ can be efficiently computed by repeated squaring.

The Padé algorithm with scaling-squaring strategy for computing $e^{\mathbf{C}}$ can be described as follows.

1. Determine the minimum integer k such that $\|\frac{\mathbf{C}}{2^k}\| < \frac{1}{2}$
2. Compute $\mathbf{N}_q(\frac{\mathbf{C}}{2^k})$ and $\mathbf{P}_q(\frac{\mathbf{C}}{2^k})$
3. Compute $\mathbf{P}_q(\frac{\mathbf{C}}{2^k}) = [\mathbf{D}_q(\frac{\mathbf{C}}{2^k})]^{-1} \mathbf{N}_q(\frac{\mathbf{C}}{2^k})$, by solving the system $\mathbf{D}_q(\frac{\mathbf{C}}{2^k}) \mathbf{P}_q(\frac{\mathbf{C}}{2^k}) = \mathbf{N}_q(\frac{\mathbf{C}}{2^k})$ (using, for instance, a suitable Gaussian elimination)
4. Compute $[\mathbf{P}_q(\frac{\mathbf{C}}{2^k})]^{2^k}$ by squaring $\mathbf{P}_q(\frac{\mathbf{C}}{2^k})$ k times

4.2. The Adapted Padé algorithm for computing $e^{\mathbf{M}\bar{h}}$

Now we apply the (q, q) Padé method above to compute the exponential of the matrix we are interested, namely $\mathbf{C} = \mathbf{M}\bar{h}$. Let us denote $\mathbf{A} = \mathbf{A}_n\bar{h}$ and $\mathbf{b} = \mathbf{b}_n\bar{h}$, then

$$\mathbf{C} = \begin{pmatrix} \mathbf{A} & \mathbf{b} & \mathbf{I}_{d \times d} & \mathbf{0}_{d \times 1} \\ \mathbf{0}_{1 \times d} & 0 & \mathbf{0}_{1 \times d} & 1 \\ \mathbf{0}_{d \times d} & \mathbf{0}_{d \times 1} & \mathbf{0}_{d \times d} & \mathbf{0}_{d \times 1} \\ \mathbf{0}_{1 \times d} & 0 & \mathbf{0}_{1 \times d} & 0 \end{pmatrix}.$$

Let k the minimum integer such that $\|\frac{\mathbf{C}}{2^k}\| < \frac{1}{2}$ and the coefficients $\alpha_j = \frac{(2q-j)!q!}{(2q)!j!(p-j)!}$ ($j = 0, \dots, q$), then

$$\left(\frac{\mathbf{C}}{2^k}\right) = \begin{pmatrix} \frac{\mathbf{A}}{2^k} & \frac{\mathbf{b}}{2^k} & \frac{\mathbf{I}_{d \times d}}{2^k} & \mathbf{0}_{d \times 1} \\ \mathbf{0}_{1 \times d} & 0 & \mathbf{0}_{1 \times d} & \frac{1}{2^k} \\ \mathbf{0}_{d \times d} & \mathbf{0}_{d \times 1} & \mathbf{0}_{d \times d} & \mathbf{0}_{d \times 1} \\ \mathbf{0}_{1 \times d} & 0 & \mathbf{0}_{1 \times d} & 0 \end{pmatrix}, \quad \left(\frac{\mathbf{C}}{2^k}\right)^2 = \begin{pmatrix} \frac{\mathbf{A}^2}{(2^k)^2} & \frac{\mathbf{A}\mathbf{b}}{(2^k)^2} & \frac{\mathbf{A}}{(2^k)^2} & \frac{\mathbf{b}}{(2^k)^2} \\ \mathbf{0}_{1 \times d} & 0 & \mathbf{0}_{1 \times d} & 0 \\ \mathbf{0}_{d \times d} & \mathbf{0}_{d \times 1} & \mathbf{0}_{d \times d} & \mathbf{0}_{d \times 1} \\ \mathbf{0}_{1 \times d} & 0 & \mathbf{0}_{1 \times d} & 0 \end{pmatrix}, \dots,$$

and in general, by an induction argument, we have that for any $j \geq 2$, $j \in \mathbb{N}$

$$\left(\frac{\mathbf{C}}{2^k}\right)^j = \begin{pmatrix} \frac{\mathbf{A}^j}{(2^k)^j} & \frac{\mathbf{A}^{j-1}\mathbf{b}}{(2^k)^j} & \frac{\mathbf{A}^{j-1}}{(2^k)^j} & \frac{\mathbf{A}^{j-2}\mathbf{b}}{(2^k)^j} \\ \mathbf{0}_{1 \times d} & 0 & \mathbf{0}_{1 \times d} & 0 \\ \mathbf{0}_{d \times d} & \mathbf{0}_{d \times 1} & \mathbf{0}_{d \times d} & \mathbf{0}_{d \times 1} \\ \mathbf{0}_{1 \times d} & 0 & \mathbf{0}_{1 \times d} & 0 \end{pmatrix}.$$

Hence,

$$\begin{aligned} \mathbf{N}_q\left(\frac{\mathbf{C}}{2^k}\right) &= \sum_{j=0}^q \alpha_j \left(\frac{\mathbf{C}}{2^k}\right)^j \\ &= \begin{pmatrix} \mathbf{I} + \mathbf{A} \left(\alpha_1 \left(\frac{\mathbf{I}}{2^k}\right) + \mathbf{A}\mathbf{S}\right) & \left(\frac{\alpha_1}{2^k}\mathbf{I} + \mathbf{A}\mathbf{S}\right)\mathbf{b} & \left(\frac{\alpha_1}{2^k}\mathbf{I} + \mathbf{A}\mathbf{S}\right) & \mathbf{S}\mathbf{b} \\ \mathbf{0}_{1 \times d} & 1 & \mathbf{0}_{1 \times d} & \frac{\alpha_1}{2^k} \\ \mathbf{0}_{d \times d} & \mathbf{0}_{d \times 1} & \mathbf{I}_{d \times d} & \mathbf{0}_{d \times 1} \\ \mathbf{0}_{1 \times d} & 0 & \mathbf{0}_{1 \times d} & 1 \end{pmatrix}, \end{aligned}$$

where,

$$\mathbf{S} = \left(\alpha_2 \frac{\mathbf{I}}{(2^k)^2} + \alpha_3 \frac{\mathbf{A}}{(2^k)^3} + \dots + \alpha_q \frac{(\mathbf{A})^{q-2}}{(2^k)^q} \right).$$

Similarly,

$$\begin{aligned} \mathbf{D}_q\left(\frac{\mathbf{C}}{2^k}\right) &= \sum_{j=0}^q \alpha_j \left(-\frac{\mathbf{C}}{2^k}\right)^j \\ &= \begin{pmatrix} \mathbf{I} + \mathbf{A} \left(-\frac{\alpha_1}{2^k}\mathbf{I} + \mathbf{A}\bar{\mathbf{S}}\right) & \left(-\frac{\alpha_1}{2^k}\mathbf{I} + \mathbf{A}\bar{\mathbf{S}}\right)\mathbf{b} & \left(-\frac{\alpha_1}{2^k}\mathbf{I} + \mathbf{A}\bar{\mathbf{S}}\right) & \bar{\mathbf{S}}\mathbf{b} \\ \mathbf{0}_{1 \times d} & 1 & \mathbf{0}_{1 \times d} & -\frac{\alpha_1}{2^k} \\ \mathbf{0}_{d \times d} & \mathbf{0}_{d \times 1} & \mathbf{I}_{d \times d} & \mathbf{0}_{d \times 1} \\ \mathbf{0}_{1 \times d} & 0 & \mathbf{0}_{1 \times d} & 1 \end{pmatrix}, \end{aligned}$$

where,

$$\bar{\mathbf{S}} = \left(\alpha_2 \frac{\mathbf{I}}{(2^k)^2} - \alpha_3 \frac{\mathbf{A}}{(2^k)^3} + \dots + (-1)^{q-2} \alpha_q \frac{(\mathbf{A})^{q-2}}{(2^k)^q} \right).$$

Now, we will compute $\mathbf{P}_q\left(\frac{\mathbf{C}}{2^k}\right) = [\mathbf{D}_q\left(\frac{\mathbf{C}}{2^k}\right)]^{-1} \mathbf{N}_q\left(\frac{\mathbf{C}}{2^k}\right)$ by solving the matrix linear system $\mathbf{D}_q\left(\frac{\mathbf{C}}{2^k}\right) \mathbf{P}_q\left(\frac{\mathbf{C}}{2^k}\right) = \mathbf{N}_q\left(\frac{\mathbf{C}}{2^k}\right)$.

It is not hard to see that $\mathbf{P}_q(\frac{\mathbf{C}}{2^k})$ has the form

$$\mathbf{P}_q(\frac{\mathbf{C}}{2^k}) = \begin{pmatrix} \mathbf{U1} & \mathbf{U2} & \mathbf{U3} & \mathbf{U4} \\ \mathbf{0}_{1 \times d} & 1 & \mathbf{0}_{1 \times d} & \frac{\alpha_1}{2^{k-1}} \\ \mathbf{0}_{d \times d} & \mathbf{0}_{d \times 1} & \mathbf{I}_{d \times d} & \mathbf{0}_{d \times 1} \\ \mathbf{0}_{1 \times d} & 0 & \mathbf{0}_{1 \times d} & 1 \end{pmatrix}$$

where the matrices $\mathbf{U1}, \mathbf{U3}$ and the vectors $\mathbf{U2}, \mathbf{U4}$ satisfy, the systems of linear equations

$$\begin{aligned} \left[\mathbf{I} + \mathbf{A} \left(-\frac{\alpha_1}{2^k} \mathbf{I} + \mathbf{A} \bar{\mathbf{S}} \right) \right] \mathbf{U1} &= \left[\mathbf{I} + \mathbf{A} \left(\frac{\alpha_1}{2^k} \mathbf{I} + \mathbf{A} \mathbf{S} \right) \right] \\ \left[\mathbf{I} + \mathbf{A} \left(-\frac{\alpha_1}{2^k} \mathbf{I} + \mathbf{A} \bar{\mathbf{S}} \right) \right] \mathbf{U2} &= \left[\left(\frac{\alpha_1}{2^k} \mathbf{I} + \mathbf{A} \mathbf{S} \right) \mathbf{b} \right] - \left[\left(-\frac{\alpha_1}{2^k} \mathbf{I} + \mathbf{A} \bar{\mathbf{S}} \right) \mathbf{b} \right] \\ \left[\mathbf{I} + \mathbf{A} \left(-\frac{\alpha_1}{2^k} \mathbf{I} + \mathbf{A} \bar{\mathbf{S}} \right) \right] \mathbf{U3} &= \left(\frac{\alpha_1}{2^k} \mathbf{I} + \mathbf{A} \mathbf{S} \right) - \left(-\frac{\alpha_1}{2^k} \mathbf{I} + \mathbf{A} \bar{\mathbf{S}} \right) \\ \left[\mathbf{I} + \mathbf{A} \left(-\frac{\alpha_1}{2^k} \mathbf{I} + \mathbf{A} \bar{\mathbf{S}} \right) \right] \mathbf{U4} &= \mathbf{S} \mathbf{b} - \left[\left(-\frac{\alpha_1}{2^k} \mathbf{I} + \mathbf{A} \bar{\mathbf{S}} \right) \mathbf{b} \frac{\alpha_1}{2^{k-1}} \right] - \bar{\mathbf{S}} \mathbf{b}, \end{aligned}$$

Since the fundamental matrix of each system above is the same one, we can exploit this, and that $\mathbf{U2} = (\mathbf{U3}) \mathbf{b}$, to yield significant improvements in the computational cost when solving these set of simultaneous equations. That is, we can the LU descomposition to $\left[\mathbf{I} + \mathbf{A} \left(-\frac{\alpha_1}{2^k} \mathbf{I} + \mathbf{A} \bar{\mathbf{S}} \right) \right]$ and then use the standard procedure for obtaining the definitive solution (see for instance [15]).

One the $\mathbf{U1}, \mathbf{U2}, \mathbf{U3}, \mathbf{U4}$ are obtained, it remains to compute $(\mathbf{P}_q(\frac{\mathbf{C}}{2^k}))^m$, where $m = 2^k$, to conclude the scalling-squaring Padé method. By an induction argument, it is not hard to prove that

$$\begin{aligned} \left(\mathbf{P}_q(\frac{\mathbf{C}}{2^k}) \right)^m &= \begin{pmatrix} \mathbf{U1} & \mathbf{U2} & \mathbf{U3} & \mathbf{U4} \\ \mathbf{0}_{1 \times d} & 1 & \mathbf{0}_{1 \times d} & \frac{\alpha_1}{2^{k-1}} \\ \mathbf{0}_{d \times d} & \mathbf{0}_{d \times 1} & \mathbf{I}_{d \times d} & \mathbf{0}_{d \times 1} \\ \mathbf{0}_{1 \times d} & 0 & \mathbf{0}_{1 \times d} & 1 \end{pmatrix}^m \\ &= \begin{pmatrix} (\mathbf{U1})^m & \left(\sum_{i=0}^{m-1} (\mathbf{U1})^i \right) \mathbf{U2} & \left(\sum_{i=0}^{m-1} (\mathbf{U1})^i \right) \mathbf{U3} & \left(\sum_{i=0}^{m-1} (\mathbf{U1})^i \right) \mathbf{U4} + \frac{\alpha_1}{2^{k-1}} \left(\sum_{i=0}^{m-2} (m-1-i) (\mathbf{U1})^i \right) \mathbf{U2} \\ 0_{1 \times d} & 1 & 0_{1 \times d} & \frac{\alpha_1}{2^{k-1}} \\ 0_{d \times d} & 0_{d \times 1} & \mathbf{I}_{d \times d} & 0_{d \times 1} \\ 0_{1 \times d} & 0 & 0_{1 \times d} & 1 \end{pmatrix}. \end{aligned}$$

Then, by substituting the above Padé approximation to $e^{\mathbf{M} \bar{h}}$ in (21), the implementation of the approximation \mathbf{R}_n^{k+1} to $\mathbf{R}(t_n^{k+1})$ is

$$\begin{aligned} \mathbf{R}_n^{k+1} &= \mathbf{R}_n^k + \left[\mathbf{I}_{d \times d} \quad \mathbf{0}_{d \times (d+2)} \right] \left(\mathbf{P}_q(\frac{\mathbf{C}}{2^k}) \right)^m \left[\mathbf{0}_{d \times d} \quad 0 \quad \mathbf{d}_n^k \quad 1 \right]^\top \\ &= \mathbf{L} \mathbf{R}_n^k + \mathbf{Q}, \end{aligned} \tag{22}$$

where

$$\mathbf{L} = \left(\sum_{i=0}^{m-1} (\mathbf{U1})^i \right) (\mathbf{U3}),$$

and

$$\mathbf{Q} = \left(\sum_{i=0}^{m-1} (\mathbf{U1})^i \right) \mathbf{U4} + \frac{\alpha_1}{2^{k-1}} \left(\sum_{i=0}^{m-2} (m-1-i) (\mathbf{U1})^i \right) (\mathbf{U3}) \mathbf{b}.$$

Note that \mathbf{L} and \mathbf{Q} remain the same in each subinterval $[t_n^k, t_{n+1}^{k+1}]$, so these values are computed only once in each interval $[t_n, t_{n+1}]$.

4.3. Implementation of the method

Now that, from (22), we have a computational algorithm to implement \mathbf{R}_n^{k+1} from \mathbf{R}_n^k , the integrator (14) can be efficiently implemented by the recursive equation

$$\begin{aligned} \mathbf{R}_{n+1} = & (\mathbf{I} + \mathbf{L}\mathbf{A}_n)^{h^{1-\gamma}} \mathbf{R}_n + \sum_{i=0}^{h^{1-\gamma}} (\mathbf{I} + \mathbf{L}\mathbf{A}_n)^{i-1} (\mathbf{Q} + \mathbf{L}(\mathbf{f}(t_n, \mathbf{R}_n) - \mathbf{A}_n \mathbf{R}_n)) \\ & + h^{-\gamma} \left(\sum_{i=0}^{h^{1-\gamma}} (\mathbf{I} + \mathbf{L}\mathbf{A}_n)^{i-1} \Delta \mu^{(k-i)} \right) \mathbf{L} \end{aligned}$$

5. Concluding Remarks

In this work we introduce a reliable integrator for the computer simulation of a Stratonovich stochastic transport equation. For this we develop a computational method which is obtained via the solution of an associated integral RDE and the related stochastic flow of characteristics. For solving this integral RDE pathwise in a stable way, the key idea was to use a local linearization technique and a representation of the solution of the resulting equation via a suitable matrix exponential that remarkably remains invariant independently of each given realization of the randomness. It is important to note that thanks to the particular structure of the involved matrix, the computational burden was reduced (via a adapted Padé method with scaling-squaring strategy) to the same computational load of solving simultaneous linear systems through a LU decomposition of a unique matrix. It is worth to note that the proposed method is well suited for parallel computing, and since the computational cost scales with the dimension of the underline equation, the approach has great potential even for very large simulation models.

Finally we mention the interesting possibility of applying the computational strategy we follow in this paper for the simulation of the transport equation driven by a rough path. That is the SPDE

$$\begin{cases} \frac{d}{dt} u(t, x) + b(t, x) \nabla u(t, x) + \nabla u(t, x) \frac{dV_t}{dt} = 0, \\ u(0, x) = u_0(x), \end{cases} \quad (23)$$

where V_t is a rough path and the integration is taken in the sense of the rough path theory (see [12] for a general reference). Rough path theory is suitable to study differential equations driven by processes with low regularity as the fractional Brownian motion. It is important remark that in this case there exist a representation of the solution of (23) via characteristics, so the application of the method could be also possible in this case.

Acknowledgement 5. *C.Olivera would like to thank the hospitality of the Instituto Nacional de Matemática Pura e Aplicada (IMPA), where this work was initiated.*

- [1] C. Le Bris, P.-L. Lions, Renormalized solutions of some transport equations with partially $W^{1,1}$ velocities and applications, *Annali di Matematica*, 183 (1): (2003) 97-130.
- [2] Z. Brzezniaka, F. Flandoli Almost sure approximation of Wong-Zakai type for stochastic partial differential equations Stochastic Processes and their Applications, Volume 55, Issue 2, February 1995, 329–358
- [3] F. Carbonell, J.C. Jimenez, Weak local linear discretizations for stochastic differential equations: convergence and numerical schemes, *J. Comput. Appl. Math.*, 197 (2006) 578-596.
- [4] F. Carbonell, J.C. Jimenez, R.J. Biscay, H. de la Cruz, Rate of convergence of local linearization schemes for initial-value problems, *Appl. Math. Comput.*, 171 (2005) 1282-1295.
- [5] P. Catuogno, C. Olivera, L^p -solutions of the stochastic transport equation, to appears in *Random Operator and Stochastic Equations*, arXiv:1102.3076.
- [6] M. Caruana and N. Victoir, Partial differential equations driven by rough paths , *Journal of Differential Equations*, 247, 1, 140-173, 2009
- [7] H. de la Cruz, R.J. Biscay, J.C. Jimenez, F. Carbonell, T. Ozaki, High Order Local Linearization methods: an approach for constructing A-stable high order explicit schemes for stochastic differential equations with additive noise, *BIT Num. Math.* 50 (2010) 509-539.
- [8] H. de la Cruz, R.J. Biscay, J.C. Jimenez, F. Carbonell, Local Linearization - Runge Kutta Methods: a class of A-stable explicit integrators for dynamical systems, *Math. Comput. Modell.*, 57 (2013) 720-740.

- [9] F. Dorini, M. C. Cunha, A finite volumen method for the mean of the solution of the random transport equation, *Appl. Math. Comput* 187 (2) (2007), 912-921.
- [10] F. Dorini, M. C. Cunha, Statistical moments of the random linear transport equation, *J. Comput. Phys* 227 (2008), 8541-8550
- [11] F. Flandoli, M. Gubinelli, E. Priola, 2010. Well-posedness of the transport equation by stochastic perturbation, *Invent. Math.*, 180(1): 1-53
- [12] P. Friz Multidimensional dimensional processes seen as rough paths, Cambridge University Press, 2010.
- [13] I. Gyongy Approximations of stochastic partial differential equations, In: *Stochastic partial differential equations and applications (Trento, 2002)*, Lecture Notes in Pure and Appl. Math., Vol. 227, Dekker, New York, 2002, 287–307.
- [14] I. Gyongy, N. Krylov On the splitting-up method and stochastic partial differential equations, *Ann. Appl. Prob.* 31 (2003), 564–591.
- [15] Golub, G. H. and Van Loan, C. F. *Matrix Computations*. The Johns Hopkins University Press, 3th Edition, 1996.
- [16] A. Jentzen and P. Kloeden, *Taylor Approximations for Stochastic Partial Differential Equations* , Cambridge University Press, 2011.
- [17] A. Jentzen and P. E Kloeden, Pathwise convergent higher order numerical schemes for random ordinary differential equations. *Proc. R. Soc. A* 463 no. 2087,(2007), 2929-2944.
- [18] A. Jentzen and P. E Kloeden, Pathwise Taylor schemes for random ordinary differential equations. *BIT* 49 no. 1 (2009),113-140.
- [19] J.C. Jimenez, C. Mora, R.J. Biscay, Dynamic properties of the local linearization method for initial-value problems, *Appl. Math. Comput.*, 126 (2002) 63-80.
- [20] J.C. Jimenez, F. Carbonell, Rate of convergence of Local Linearization schemes for initial-value problems, *Applied Mathematics and Computation*, 171 (2005) 1282-1295.
- [21] H. Kunita, *First Order Stochastic Partial Differential Equations* , Proceedings of the Taniguchi International Symposium on Stochastic Analysis, 1984.
- [22] H. Kunita, Stochastic differential equations and stochastic flows of diffeomorphisms, *Lectures Notes in Mathematics*, Springer-Verlag, Berlin, 1097 (1982) 143-303.
- [23] H. Kunita, *Stochastic flows and stochastic differential equations*. Cambridge University Press. 1990
- [24] G. N. Milstein, M. V. Tretyakov, Solving parabolic stochastic partial differential equations via averaging over characteristics, *Math. Comp.* 78, no. 268, (2009) 2075–2106
- [25] C. Moler, C.F. Van Loan, Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later, *SIAM Review*. 45 (2003) 3-49.
- [26] H. Osnes, H.P. Langtangen, A study of some finite difference schemes for a unidirectional stochastic transport equation, *SIAM J. Sci. Comput.* 19, no. 3, (1998) 799-812.
- [27] H. Yoo. Semi-discretization of stochastic partial differential equations on \mathbb{R}^1 by a finite difference method, *Math. Comp.* 69 (2000), 653–666.