Algebraic structures and stochastic differential equations driven by Lévy processes

PROCEEDINGS OF THE ROYAL SOCIETY A

rspa.royalsocietypublishing.org

Research

Article submitted to journal

Subject Areas:
Probability, Differential Equations, Computational Mathematics

Keywords:
Lévy processes, efficient integrators, quasi-shuffle algebra

Author for correspondence:
Charles Curry (7/12/18)
e-mail: charles.curry@math.ntnu.no

Algebraic Structures and Stochastic Differential Equations driven by Lévy processes: Supplementary material

Charles Curry[1,3], Kurusch Ebrahimi–Fard[2], Simon J.A. Malham[1] and Anke Wiese[1]

[1]Maxwell Institute for Mathematical Sciences and School of Mathematical and Computer Sciences, Heriot-Watt University, Edinburgh EH14 4AS, UK.
[2]Department of Mathematical Sciences, NTNU, 7491 Trondheim, Norway.
[3]Current address: Department of Mathematical Sciences, NTNU, 7491 Trondheim, Norway.

We present additional details concerning the results of our manuscript. In Section (1) we provide an introduction to the role of quasi-shuffle algebras and quasi-shuffle convolution algebras in the development and analysis of strong integrators for stochastic differential equations. In Section (2), we relate the stochastic Taylor expansion given to that derived in Platen & Bruti-Liberati, showing their equivalence. In Section (3), we show that the separated Taylor expansion proposed in the text takes a particularly simple form in the case of linear, constant coefficient equations. Finally, in Section (4) we present in greater detail the results of our numerical experiments, including more plots covering more cases, and with more comments concerning methodology. The MATLAB code is available as a separate file.

Royal Society Publishing
Informing the science of the future

1. SDEs and quasi-shuffle algebras

Our goal in this section is to provide an introduction to the role of quasi-shuffle algebras as well as quasi-shuffle convolution algebras in the development and analysis of strong integrators for stochastic differential equations. We will assume some basic knowledge of stochastic differential equations and their strong/pathwise solution. We consider Itô stochastic differential systems driven by scalar stochastic processes and governed by vector fields of the form:

$$Y_t = Y_0 + \sum_{i=1}^{d} \int_0^t V_i(Y_\tau)\, \mathrm{d}X_\tau^i,$$

for time interval $t \in [0, T]$ for some $T > 0$. The stochastic solution process Y is assumed to be $\mathbb{R}^N$-valued for some $N \in \mathbb{N}$. We assume the initial data $Y_0 \in \mathbb{R}^N$ is a given deterministic vector. The functions V_i, $i = 1, \ldots, d$, are assumed to be smooth governing vector fields and in general non-commuting; we explain what we mean by this further below. The X^i, $i = 1, \ldots, d$, are scalar driving stochastic processes; their precise characterization will be discussed presently. Our goals in this section are as follows, to:

(i) Illustrate the development of the stochastic Taylor series expansion for the solution Y_t to the Itô stochastic differential system above. The stochastic Taylor series expansion is the basis of the construction of many classes of strong numerical approximation schemes;
(ii) Demonstrate why a precise understanding of the relationship between the repeated integrals of the driving stochastic processes X^i is important, first, in the efficient implementation of strong numerical approximation schemes, and second, in the design of classes of numerical schemes such as those based on the exponential Lie series. A precise understanding of the algebraic structure generated by the product of such repeated integrals is crucial to both of these components in the theory and implementation of strong numerical approximation schemes; and
(iii) Show how to abstract the relationship between the repeated integrals, first to the quasi-shuffle algebra, and then second, to one further level of abstraction, the quasi-shuffle convolution algebra. We demonstrate how this abstraction and the concepts and structures provided therein, can be used to inform the construction of new numerical approximation schemes.

As we intend this to be a self-contained introduction there will inevitably be some repetition of the material in the main manuscript. We have endeavoured to achieve the three goals above with minimal repetition while at the same time trying to give a broader perspective. Our setting is a complete, filtered probability space $\left(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t \geqslant 0}, P\right)$ assumed to satisfy the usual hypothesis; see Protter [16, p. 3]. We assume the solution process $Y \in \mathbb{R}^N$ exists on some finite time interval.

Our first goal in this section is to develop the stochastic Taylor series expansion for the solution Y_t, without loss of generality, about $t = 0$. We illustrate how this is achieved in the context where the driving processes X^i, $i = 1, \ldots, d$, are independent Wiener processes. As such the quadratic covariations $[X^i, X^j] = 0$ for all $i \neq j$. The quadratic covariation of X^i with itself $[X^i, X^i]$ is known as its quadratic variation.

Remark 1.1. In fact our development of the stochastic Taylor series expansion here is sufficiently general to allow for the X^i, $i = 1, \ldots, d$, to be continuous semimartingales. These are a generalization of Wiener processes. The Lévy processes we consider in the main text, which include jumps, are a different generalization. For a collection of Wiener processes all the quadratic variations $[X^i, X^i]_t = t$ for all $i = 1, \ldots, d$. However, as we see below, our derivation of the stochastic Taylor expansion allows for more general quadratic variation and indeed is consistent with the derivation of the stochastic Taylor expansion for the case when the X^i, $i = 1, \ldots, d$, are continuous semimartingales; see Ebrahimi–Fard, Malham, Patras and Wiese [7] for more details.

The key tool for developing the stochastic Taylor series expansion for Y_t is Itô's Lemma (chain rule). For any smooth function $f\colon \mathbb{R}^N \to \mathbb{R}^N$, this states that if Y_t satisfies the stochastic differential equation above, then $f(Y_t)$ satisfies (see for example Protter [16])

$$f(Y_t) = f(Y_0) + \sum_{i=1}^{d} \int_0^t \big(V_i \cdot \partial\big) f(Y_\tau)\, \mathrm{d}X_\tau^i + \tfrac{1}{2} \sum_{i=1}^{d} \int_0^t \big(V_i \otimes V_i \colon \partial^2\big) f(Y_\tau)\, \mathrm{d}[X^i, X^i]_\tau.$$

Let us explain carefully the notation we have used here. For any $Y \in \mathbb{R}^N$ we have

$$\big(V_i \cdot \partial\big) f(Y) = \sum_{j=1}^{N} V_i^j(Y) \partial_{Y_j} f(Y).$$

In other words when we consider the evolution of any function of the solution, here $f(Y_t)$, then the corresponding first order terms involve the first order partial differential operators $V_i \cdot \partial$ acting on $f(Y_t)$. The noncommutativity of the vector fields above refers to the fact that in general we assume that any two first order partial differential operators $V_i \cdot \partial$ and $V_j \cdot \partial$ with $i \neq j$ do not commute. Hereafter we take the perspective that instead of the vector fields $V_i(Y_t)$ being functions of Y_t as they are in the original stochastic differential equation, they are now first order partial differential operators $V_i \cdot \partial$ acting on $f(Y_t)$. Naturally when $f = \mathrm{id}$ so that $f(Y_t) = Y_t$ then $\big(V_i \cdot \partial\big) f(Y) = \big(V_i \cdot \partial\big) Y = V(Y)$, and the evolution equation for $f(Y_t)$ reverts to the stochastic differential equation for Y_t. The Itô chain rule also includes second order terms as shown, where we have used the notation

$$\big(V_i \otimes V_i \colon \partial^2\big) f(Y) \coloneqq \sum_{j,k=1}^{N} V_i^j(Y) V_i^k(Y) \partial_{Y_j} \partial_{Y_k} f(Y).$$

The terms $[X^i, X^i]$ represent the quadratic variation of the X^i for $i = 1, \ldots, d$. We now utilize the following succinct notation. We set

$$D_i \coloneqq V_i \cdot \partial, \qquad D_{[i,i]} \coloneqq \tfrac{1}{2} V_i \otimes V_i \colon \partial^2 \qquad \text{and} \qquad X^{[i,i]} \coloneqq [X^i, X^i].$$

Then the Itô chain rule takes the form

$$f(Y_t) = f(Y_0) + \sum_{a \in \mathbb{A}} \int_0^t D_a f(Y_\tau)\, \mathrm{d}X_\tau^a,$$

where $\mathbb{A}$ denotes the alphabet of letters $\{1, \ldots, d, [1,1], \ldots, [d,d]\}$. We use the terminology 'alphabet' and 'letters' as opposed to 'indices' with an eye on the algebraic structures ahead. This integral equation applies for any smooth function $f\colon \mathbb{R}^N \to \mathbb{R}^N$. For example we could take f to be $D_a f$, and the relationship just above holds for the function $D_a f\colon \mathbb{R}^N \to \mathbb{R}^N$. In other words we have

$$D_a f(Y_t) = D_a f(Y_0) + \sum_{b \in \mathbb{A}} \int_0^t D_b D_a f(Y_{\tau_2})\, \mathrm{d}X_{\tau_2}^b.$$

Substituting this for the integrand on the right-hand side in the equation for $f(Y_t)$, we obtain

$$\begin{aligned} f(Y_t) &= f(Y_0) + \sum_{a \in \mathbb{A}} \int_0^t \left(D_a f(Y_0) + \sum_{b \in \mathbb{A}} \int_0^\tau D_b D_a f(Y_{\tau_2})\, \mathrm{d}X_{\tau_2}^b \right) \mathrm{d}X_\tau^a \\ &= f(Y_0) + \sum_{a \in \mathbb{A}} \int_0^t \mathrm{d}X_{\tau_1}^a\, D_a f(Y_0) + \sum_{a,b \in \mathbb{A}} \int_0^t \int_0^{\tau_1} D_b D_a f(Y_{\tau_2})\, \mathrm{d}X_{\tau_2}^b \mathrm{d}X_{\tau_1}^a. \end{aligned}$$

Now we replace f in the equation above by $D_b D_a f$ so that we have

$$D_b D_a f(Y_t) = D_b D_a f(Y_0) + \sum_{c \in \mathbb{A}} \int_0^t D_c D_b D_a f(Y_{\tau_3})\, \mathrm{d}X_{\tau_3}^c.$$

Substituting this for the integrand in the double integral term on the first iteration above we find

$$f(Y_t) = f(Y_0) + \sum_{a\in\mathbb{A}} \int_0^t \mathrm{d}X^a_{\tau_1}\, D_a f(Y_0) + \sum_{a,b\in\mathbb{A}} \int_0^t \int_0^{\tau_1} \mathrm{d}X^b_{\tau_2} \mathrm{d}X^a_{\tau_1}\, D_b D_a f(Y_0)$$
$$+ \sum_{a,b,c\in\mathbb{A}} \int_0^t \int_0^{\tau_1} \int_0^{\tau_2} D_c D_b D_a f(Y_{\tau_3})\, \mathrm{d}X^c_{\tau_3} \mathrm{d}X^b_{\tau_2} \mathrm{d}X^a_{\tau_1}.$$

It is now clear that we can repeat this procedure ad infinitum and the precise form all the subsequent terms in this series expansion will take. Further let us introduce an even more succinct notation. Consider a word $w = a_1 a_2 \cdots a_n$ constructed from letters a_i, $i = 1, \ldots, n$, from the alphabet $\mathbb{A}$. Then let us use $D_w \coloneqq D_{a_1} \cdots D_{a_n}$ to denote the successive compositon of the partial differential operators shown. Further let us denote

$$I_w \coloneqq \int_{0\leqslant\tau_1\leqslant\cdots\leqslant\tau_n\leqslant t} \mathrm{d}X^{a_1}_{\tau_1} \cdots \mathrm{d}X^{a_n}_{\tau_n}.$$

The iteration procedure above thus produces the solution expansion

$$f(Y_t) = \sum_w I_w D_w f(Y_0).$$

Here the sum is over all words/multi-indices w that can be constructed from the alphabet $\mathbb{A}$. This is the stochastic Taylor expansion for $f(Y_t)$. The stochastic Taylor expansion for Y_t itself can be recovered by setting $f = \mathrm{id}$. The stochastic Taylor expansion is the starting point for strong stochastic differential numerical approximation schemes of higher order beyond the Euler–Maruyama approximation. In the deterministic setting, and in the stochastic setting, a very useful concept in the construction of numerical approximation schemes is the notion of the flowmap.

Definition 1.1 (Flowmap)**.** *For any smooth function $f\colon \mathbb{R}^N \to \mathbb{R}^N$, we define the flowmap φ_t associated with the stochastic differential equation above as the map prescibing the transport of the initial data $f(Y_0)$ to the solution $f(Y_t)$ at time $t \geqslant 0$, in other words $\varphi_t\colon f(Y_0) \mapsto f(Y_t)$.*

We note the following. Naturally for the case $f = \mathrm{id}$ we have $Y_t = \varphi_t(Y_0)$. By substituting this into the definition of the flowmap we deduce $\varphi_t(f) = f(\varphi_t)$. Further, from the stochastic Taylor expansion we already have an explicit representation for the flowmap, namely

$$\varphi_t = \sum_w I_w D_w.$$

Remark 1.2 (Separated stochastic Taylor expansion)**.** The stochastic Taylor expansion for $f(Y_t)$ above, as well as for the flowmap φ_t, is separated. It is a sum over terms which are the real product of the time-dependent stochastic repeated integrals I_w and the time-independent deriative terms $D_w f(Y_0)$. This separated form arose naturally and directly in our derivation of the stochastic Taylor expansion above which includes the case when the driving processes X^i, $i = 1, \ldots, d$, are independent Wiener processes, as well as more generally, continuous semimartingales. In the main text the driving stochastic processes are independent Lévy processes. The derivation of the stochastic Taylor expansion in this case is more convoluted due to the jumps of a Lévy process and the analogous derivation to that above does not produce a separated stochastic Taylor expansion directly. However, as shown in the main text, further Taylor expansion of some of the governing vector fields eventually results in a separated stochastic Taylor expansion analogous to that above.

Our second goal in this section is to elucidate the relationship between the repeated integrals I_w in the stochastic Taylor expansion. Here again for the moment, we assume that the driving stochastic processes X^i, $i = 1, \ldots, d$, are Wiener processes. However, in fact, our discussion will be sufficiently general to apply to the case when the X^i, $i = 1, \ldots, d$, are continuous semimartingales. We provide two demonstrations of why a precise understanding of these relationships is crucial to the efficient strong numerical simulation of stochastic differential equations. Such strong

numerical approximations are constructed via finite truncations of the stochastic Taylor expansion for Y_t, or equivalently, the flowmap which is applied to Y_0. Assuming that computing the terms $D_w(Y_0)$, which involves computing derivatives of the functions V_i from the original stochastic differential equation, is relatively straightforward, then the main task is to simulate the repeated integrals I_w. This is where most of the computational burden in the strong simulation of stochastic differential equations lies and any possible efficiencies should be utilized. With this in mind we notice that not all repeated integrals of a given order in the expansion are independent. For example integration parts, i.e. the product rule, reveals that

$$I_i I_j = I_{ij} + I_{ji} + \delta_{ij} I_{[i,j]},$$

for any letters $i, j \in \mathbb{A}$ and where δ_{ij} is the Kronecker delta function. Thus for instance if $i = j$ then we do not need to simulate I_{ii} as we can construct them from $I_i \equiv X^i$ and $I_{[i,i]} \equiv X^{[i,i]}$. Or for example if $i \neq j$ and we have simulated I_{ij}, then we can construct I_{ji} from I_{ij} and X^i and X^j. We start to see how a clear understanding of the algebraic structure that underpins the relationships between the repeated integrals I_w has practical impact on the numerical simualation of stochastic differential equations. This is our first example demonstrative evidence. Now for our second. Other classes of numerical methods, for example those which utilize the exponential Lie series, can be constructed by considering functions of the flowmap. For a given function $F\colon \mathrm{Diff}(\mathbb{R}^N) \to \mathrm{Diff}(\mathbb{R}^N)$, such a strong numerical simulation method would be constructed from φ_t as follows:

(i) Construct a new series $\psi_t = F(\varphi_t)$;
(ii) Truncate this series to produce the finite expansion $\hat{\psi}_t$;
(iii) Reconstruct an approximate flowmap as $\hat{\varphi}_t := F^{-1}(\hat{\psi}_t)$; and
(iv) Use $\hat{\varphi}_t$ as basis of a strong numerical approximation scheme.

If $F = \mathrm{id}$, the identity map, this approach corresponds to implementing a truncated stochastic Taylor expansion as a numerical approximation scheme. The archetypical numerical scheme constructed using the procedure just outlined is the case when $F = \log$. The series expansion $\log \varphi_t$ can be shown to be a Lie series for quite general scenarios, including the continuous semimartingale context, see Ebrahimi–Fard *et al.* [7]. As such it is known as the exponential Lie series. We call the corresponding numerical methods based on this approach Castell–Gaines methods after their implementation in Castell & Gaines [2]. To construct these methods we actually need an explicit representation for $\log \varphi_t$. This can be achieved as follows. We in fact compute $F(\varphi_t)$ for any function F with the power series representation

$$F(\phi) = \sum_{k \geq 0} c_k \phi^k,$$

with coefficients $c_k \in \mathbb{R}$ for $k \in \mathbb{N} \cup \{0\}$. By direct computation we see that

$$\begin{aligned} F(\varphi_t) &= \sum_{k \geq 0} c_k \left(\sum_w I_w D_w \right)^k \\ &= \sum_{k \geq 0} c_k \sum_{u_1, \ldots, u_k} (I_{u_1} I_{u_2} \cdots I_{u_k})(D_{u_1} D_{u_2} \cdots D_{u_k}) \\ &= \sum_w \left(\sum_{k=1}^{|w|} c_k \sum_{u_1 u_2 \cdots u_k = w} I_{u_1} I_{u_2} \cdots I_{u_k} \right) D_w. \end{aligned}$$

In this calculation the: summations $\sum_w$ are over all possible words w that can be constructed from the alphabet $\mathbb{A}$; summation $\sum_{u_1, \ldots, u_k}$ is over all possible words $u_1, \ldots, u_k$ that can be constructed from the alphabet $\mathbb{A}$; and summation $\sum_{u_1 u_2 \cdots u_k = w}$ is the sum over all possible collections of k words $u_1, \ldots, u_k$ that can be concatenated together to make the given word w—all the words $u_1, \ldots, u_k$ and w are again constructed from the alphabet $\mathbb{A}$. Finally $|w|$ denotes the length of the word w, i.e. the sum of all the individual letters from the alphabet $\mathbb{A} = \{1, \ldots, d, [1,1], \ldots, [d,d]\}$

used in the word. We observe that an integral part of the construction above are products of repeated integrals of the form $I_{u_1} I_{u_2} \cdots I_{u_k}$. This is our second example demonstrative evidence. Looking ahead to our third goal which we address presently, we can already see the need to consider combinatorial calculations, for example we need to consider all possible ways of splitting the word w into k words which concatenate together to create w. Further we need to consider the algebra of repeated integrals, i.e. we need to be able to consider linear combinations of real products of repeated integrals I_w. Before we move onto these structures, let us round off our second goal in this section with a concrete demonstration of how, for example, the Castell–Gaines numerical method can be implemented in practice. For simplicity assume there are only two driving processes so $d=2$ and they are both Wiener processes so $X^1 = W^1$ and $X^2 = W^2$. If we truncate the series $\psi_t = \log \varphi_t$, then across the computation interval $[t_m, t_{m+1}]$ we have

$$\hat{\psi}_{t_m,t_{m+1}} = \hat{I}_1(t_m) V_1 \cdot \partial + \hat{I}_2(t_m) V_2 \cdot \partial + \tfrac{1}{2}\big(\hat{I}_{12}(t_m) - \hat{I}_{21}(t_m)\big)\big((V_1 \cdot \partial)(V_2 \cdot \partial) - (V_2 \cdot \partial)(V_1 \cdot \partial)\big)$$

and then computing the inverse F^{-1} of this and applying it to the data Y_{t_m}, we have

$$Y_{t_{m+1}} \approx \exp\big(\hat{\psi}_{t_m,t_{m+1}}\big)(Y_{t_m}).$$

The truncated series $\psi_t = \log \varphi_t$ in this case is a Lie series and the terms associated with the words $w=12$ and $w=21$ can be expressed in the form shown involving the Lie bracket of the two vector fields. The hats on the increments $\hat{I}_1(t_m) = \Delta W^1(t_m)$ and $\hat{I}_2(t_m) = \Delta W^2(t_m)$ and repeated integrals $\hat{I}_{12}(t_m)$ and $\hat{I}_{21}(t_m)$ indicate realizations of these random variables—or suitable approximations thereof. The approximate flowmap is thus $\hat{\varphi}_{t_m,t_{m+1}} = \exp\big(\hat{\psi}_{t_m,t_{m+1}}\big)$ and is the basis for a strong numerical approximation scheme as outlined in (iv) above. Finally we compute $Y_{t_{m+1}}$ in practice by using a suitably high order ordinary differential numerical method to integrate $u'(\tau) = \hat{\psi}_{t_m,t_{m+1}}(u(\tau))$, across $\tau \in [0,1]$ with $u(0) = Y_{t_m}$, generating $u(1) \approx Y_{t_{m+1}}$.

Our third and final goal in this section is to concretely establish the connection between some specific Hopf combinatorial algebras, namely those involving the quasi-shuffle and concatenation products, and the analysis of functions of the stochastic Taylor expansion for the flowmap associated with a given stochastic differential equation. In fact we take this one abstraction step further and establish the connection between a quasi-shuffle convolution algebra and the analysis of the flowmap. At this stage we want to include the possibility that the driving stochastic processes are independent Lévy processes as outlined in the main text. We also want to assume a separated stochastic Taylor expansion for the flowmap as our starting point and so a few remarks are required to bridge the gap between our presentation hitherto and the full Lévy process case. To construct the separated stochastic Taylor expansion for the flowmap when the X^i, $i=1,\ldots,d$, are continuous semimartingales, we iteratively applied the Itô chain rule which involved the driving processes and the full set of quadratic variations $[X^i, X^i]$, $i=1,\ldots,d$. Once we start computing the product of repeated integrals further quadratic variations or power brackets such as $[X^i,[X^i,X^i]]$ will be generated. For the case when the driving stochastic processes X^i, $i=1,\ldots,d$, are Lévy processes, then already at the stage of deriving a separated form for the stochastic Taylor expansion for the flowmap, higher nested power brackets are generated. They are also generated once we start considering products of repeated integrals. Importantly, the covariation bracket is both commutative and associative so any nested bracket is invariant to the order of the processes and bracketing therein; see Curry, Ebrahimi–Fard, Malham and Wiese [4] for more details. As shown in the main text, provided we extend our alphabet to account for these higher nested power brackets, then by further Taylor expansions for the vector fields involving the jump processes, we can derive a separated stochastic Taylor expansion for the flowmap in the Lévy process case. We thus take as our starting point a separated stochastic Taylor expansion for the flowmap as indicated above, albeit with an extended alphabet as we have outlined.

The basis for our subsequent development hereafter is the product rule for any two repeated integrals and our computation for $F(\varphi_t)$ just above. We assume without loss of generality that $X_0^i = 0$ for all $i=1,\ldots,d$. From Protter [16, p. 58] and Curry *et al.* [4, Section 4] for words u, v and

letters a, b from the extended alphabet $\mathbb{A}$ we have the following product rule

$$I_{ua}(t)I_{vb}(t) = \int_0^t I_u(\tau_-)I_{vb}(\tau_-)\,\mathrm{d}I_a(\tau) + \int_0^t I_{ua}(\tau_-)I_v(\tau_-)\,\mathrm{d}I_b(\tau) + \int_0^t I_u(\tau_-)I_v(\tau_-)\,\mathrm{d}[I_a, I_b](\tau).$$

Let us denote by $\mathbb{A}^*$ the free monoid over $\mathbb{A}$. This is the set of all words $w = a_1a_2\cdots a_n$ that can be constructed from the letters $a_i \in \mathbb{A}$. We denote the empty word by $\mathbb{1} \in \mathbb{A}$. Let $\mathbb{RA}$ denote the $\mathbb{R}$-linear span of $\mathbb{A}$, and let $\mathbb{R}\langle\mathbb{A}\rangle$ denote the vector space of polynomials in the non-commuting variables in $\mathbb{A}^*$. We assume that $[\,\cdot\,,\,\cdot\,]\colon \mathbb{RA}\otimes\mathbb{RA}\to\mathbb{RA}$ is a commutative, associative product on $\mathbb{RA}$. The quasi-shuffle product on $\mathbb{R}\langle\mathbb{A}\rangle$, which is commutative, is generated inductively as follows: if $\mathbb{1}$ is the empty word then $u * \mathbb{1} = \mathbb{1} * u = u$ and

$$ua * vb = (u * vb)a + (ua * v)b + (u * v)[a, b],$$

for all words $u, v \in \mathbb{A}^*$ and letters $a, b \in \mathbb{A}$. Endowed with the product '$*$' just defined, $\mathbb{R}\langle\mathbb{A}\rangle$ is a commutative and associative algebra known as the quasi-shuffle algebra, which we denote by $\mathbb{R}\langle\mathbb{A}\rangle_*$. The connections between $[\,\cdot\,,\,\cdot\,]$ and the quadratic covariation, and then between quasi-shuffle product of the words ua and vb and the real product of the repeated integrals I_{ua} and I_{vb} are immediate. Indeed as outlined in the main text, we can define a word-to-integral map $\mu\colon w \mapsto I_w$ which, extended linearly to $\mathbb{R}\langle\mathbb{A}\rangle_*$, is an algebra isomorphism. Here we use the convention that repeated integrals indexed by polynomials are defined by linearity, i.e. $I_{k_uu+k_vv} = k_uI_u + k_vI_v$, for any constants $k_u, k_v \in \mathbb{R}$ and words $u, v \in \mathbb{A}^*$. Thus we have established that the algebra of repeated integrals and the quasi-shuffle algebra have precisely the same structure. Thus for example, any linear combination of products of repeated integrals has an exact corresponding representation (under μ^{-1}) and evaluation (under μ) in the quasi-shuffle algebra.

There is another natural algebra implicit in our stochastic Taylor expansion representation for the flowmap, which we assume has a separated form so that

$$\varphi_t = \sum_{w\in\mathbb{A}^*} I_w D_w.$$

The other natural algebra is the algebra associated with the composition of the partial differential operators D_w. Recall that for any word $w = a_1a_2\cdots a_n \in \mathbb{A}^*$, the partial differential operator D_w denotes $D_{a_1}D_{a_2}\cdots D_{a_n}$, where the partial differential operators are compositionally evaluated right to left on any suitable target functions. We extend this "product" convention linearly to any linear combinations of such expressions. At the abstract level on the vector space $\mathbb{R}\langle\mathbb{A}\rangle$, instead of the quasi-shuffle product, we can define another product, namely the concatenation product. The concatenation product of two words $u, v \in \mathbb{A}^*$ results in their concatenation $uv \in \mathbb{A}^*$ which is extended linearly to $\mathbb{R}\langle\mathbb{A}\rangle$. Further, we can define a word-to-operator map $\kappa\colon w \mapsto D_w$ which is an algebra homomorphism. The domain of this homomorphism is $\mathbb{R}\langle\mathbb{A}\rangle$ equipped with the concatenation product, which we denote simply by $\mathbb{R}\langle\mathbb{A}\rangle$ hereafter. We thus have two combinatorial algebras, the quasi-shuffle algebra $\mathbb{R}\langle\mathbb{A}\rangle_*$ and the concatenation algebra $\mathbb{R}\langle\mathbb{A}\rangle$. We can consider their completed tensor product $\mathbb{R}\langle\mathbb{A}\rangle_*\overline{\otimes}\,\mathbb{R}\langle\mathbb{A}\rangle$. Further we see that the flowmap is the image under $\mu\otimes\kappa$ of the element in $\mathbb{R}\langle\mathbb{A}\rangle_*\overline{\otimes}\,\mathbb{R}\langle\mathbb{A}\rangle$ given by

$$\sum_{w\in\mathbb{A}^*} w\otimes w.$$

Note the product of any two elements $u\otimes u'$ and $v\otimes v'$ in the tensor algebra $\mathbb{R}\langle\mathbb{A}\rangle_*\overline{\otimes}\,\mathbb{R}\langle\mathbb{A}\rangle$ is given by $(u\otimes u')(v\otimes v') = u * u'\otimes vv'$; see for example Reutenauer [18].

We now turn our attention to the computation, evaluation and representation of functions $F(\varphi_t)$ of the flowmap φ_t. Assuming that F has a power series representation as indicated above, recall our previous result from computing $F(\varphi_t)$, namely:

$$F(\varphi_t) = \sum_{w\in\mathbb{A}^*}\left(\sum_{k=1}^{|w|} c_k \sum_{u_1u_2\cdots u_k = w} I_{u_1}I_{u_2}\cdots I_{u_k}\right) D_w.$$

We see that this is the image under $\mu \otimes \kappa$ of the element in $\mathbb{R}\langle\mathbb{A}\rangle_* \overline{\otimes} \mathbb{R}\langle\mathbb{A}\rangle$ given by

$$F\left(\sum_{w\in\mathbb{A}^*} w \otimes w\right) = \sum_{w\in\mathbb{A}^*}\left(\sum_{k=1}^{|w|} c_k \sum_{u_1u_2\cdots u_k=w} u_1 * u_2 * \cdots * u_k\right) \otimes w.$$

We have essentially completed our first level of abstraction, the flowmap or any function of it can be represented in the tensor algebra $\mathbb{R}\langle\mathbb{A}\rangle_* \overline{\otimes} \mathbb{R}\langle\mathbb{A}\rangle$. Note that when computing the function F on the flowmap, we made a choice to rearrange the sum by collecting all words w on the right of the tensor '$\otimes$'. There is dual procedure which involves collecting all the words on the left instead. The interested reader should consult Reutenauer [18, Chapter 3], our main inspiration for the abstractions and computations in this section, for this dual point of view. Back to our convention above of collecting all the words on the right and combinatorial operations thereof on the left. We see that we can write

$$\sum_{w\in\mathbb{A}^*} w \otimes w = \sum_{w\in\mathbb{A}^*} \mathcal{I}(w) \otimes w \qquad \text{and} \qquad F\left(\sum_{w\in\mathbb{A}^*} w \otimes w\right) = \sum_{w\in\mathbb{A}^*} \mathcal{F}(w) \otimes w,$$

where $\mathcal{I}$ is the identity endomorphism on $\mathbb{R}\langle\mathbb{A}\rangle_*$, while also on $\mathbb{R}\langle\mathbb{A}\rangle_*$, $\mathcal{F}$ is the endomorphism

$$\mathcal{F}(w) = \sum_{k=1}^{|w|} c_k \sum_{u_1u_2\cdots u_k=w} u_1 * u_2 * \cdots * u_k.$$

Remark 1.3 (Quasi-shuffle Hopf algebras)**.** We remark that: (i) Classical introductions to Hopf algebras can be found in Radford [17] and Sweedler [19]; (ii) It is well-known in some physics communities that Hopf algebras provide efficient computational methods for working with groups, see for example Cartier [1] and Manchon [13]; and (iii) As we remark in the main manuscript, Hudson and collaborators studied quasi-shuffle Hopf algebras in the guise of sticky shuffle product Hopf algebras in the context of quantum stochastic calculus, see Hudson [9].

We now begin our second level of abstraction, and as such, we have come to the point where we need to endow $\mathbb{R}\langle\mathbb{A}\rangle_*$ and $\mathbb{R}\langle\mathbb{A}\rangle$ with a bialgebra structure. A bialgebra is a vector space equipped not only with a product, but also a coproduct, with the product and coproduct obeying certain compatability relations. The next section contains a brief overview of the algebraic concepts; see Reutenauer [18] for more details. We adhere to a minimalist briefing on this here for simplicity. Let $\langle\,\cdot\,,\,\cdot\,\rangle\colon \mathbb{R}\langle\mathbb{A}\rangle \otimes \mathbb{R}\langle\mathbb{A}\rangle \to \mathbb{R}$ denote the bilinear form where, for any words $u, v \in \mathbb{A}^*$, we set $\langle u, v\rangle$ to be 1 if $u = v$ and 0 if $u \neq v$. For this scalar product, the free monoid $\mathbb{A}^*$ forms an orthonormal basis. With this in hand, we can further endow $\mathbb{R}\langle\mathbb{A}\rangle_*$ and $\mathbb{R}\langle\mathbb{A}\rangle$ with the following respective coproducts.

Definition 1.2 (Deconcatenation and de-quasi-shuffle coproducts)**.** *We define the deconcatenation coproduct* $\Delta\colon \mathbb{R}\langle\mathbb{A}\rangle \to \mathbb{R}\langle\mathbb{A}\rangle \otimes \mathbb{R}\langle\mathbb{A}\rangle$ *for any word* $w \in \mathbb{R}\langle\mathbb{A}\rangle$ *by*

$$\Delta(w) \coloneqq \sum_{u,v} \langle uv, w\rangle\, u \otimes v.$$

We also define the de-quasi-shuffle coproduct $\Delta'\colon \mathbb{R}\langle\mathbb{A}\rangle \to \mathbb{R}\langle\mathbb{A}\rangle \otimes \mathbb{R}\langle\mathbb{A}\rangle$ *for any word* $w \in \mathbb{R}\langle\mathbb{A}\rangle$ *by*

$$\Delta'(w) \coloneqq \sum_{u,v} \langle u * v, w\rangle\, u \otimes v.$$

Note for example, the action of the deconcatenation coproduct on a word $w \in \mathbb{A}^*$, is to generate a sum of all of its possible two-partitions $u \otimes v$ including $\mathbb{1} \otimes w$ and $w \otimes \mathbb{1}$. Endowed with quasi-shuffle product and deconcatenation coproduct $\mathbb{R}\langle\mathbb{A}\rangle_*$ is a bialgebra. Further, endowed with the concatenation product and de-quasi-shuffle coproduct $\mathbb{R}\langle\mathbb{A}\rangle$ is also a bialgebra. An endomorphism known as the antipode can be defined for both these bialgebras, and is given in Hoffman [8]. Equipped with antipodes, both these bialgebras become Hopf algebras. We can now define the convolution of any two endomorphisms on the quasi-shuffle Hopf algebra $\mathbb{R}\langle\mathbb{A}\rangle_*$.

Definition 1.3 (Convolution product). *Suppose $\mathcal{G}_1$ and $\mathcal{G}_2$ are two linear endomorphisms on the Hopf quasi-shuffle algebra $\mathbb{R}\langle\mathbb{A}\rangle_*$. We define their quasi-shuffle convolution product $\mathcal{G}_1 \star \mathcal{G}_2$ by the formula*

$$\mathcal{G}_1 \star \mathcal{G}_2 := * \circ (\mathcal{G}_1 \otimes \mathcal{G}_2) \circ \Delta.$$

In other words, since the deconcatenation product Δ splits any word $w \in \mathbb{A}^*$ in the sum of all its two-partitions $u \otimes v$, including when u or v are the empty word $\mathbb{1}$, we have

$$(\mathcal{G}_1 \star \mathcal{G}_2)(w) := \sum_{uv=w} \mathcal{G}_1(u) * \mathcal{G}_2(v).$$

Let us denote by $\mathrm{End}(\mathbb{R}\langle\mathbb{A}\rangle_*)$ the $\mathbb{R}$-module of linear endomorphisms of $\mathbb{R}\langle\mathbb{A}\rangle_*$. The quasi-shuffle convolution product on $\mathrm{End}(\mathbb{R}\langle\mathbb{A}\rangle_*)$ naturally extends as follows

$$(\mathcal{G}_1 \star \mathcal{G}_2 \star \cdots \star \mathcal{G}_k)(w) := \sum_{u_1 u_2 \cdots u_k = w} \mathcal{G}_1(u_1) * \mathcal{G}_2(u_2) * \cdots * \mathcal{G}_k(u_k),$$

for any $\mathcal{G}_i \in \mathrm{End}(\mathbb{R}\langle\mathbb{A}\rangle_*)$, $i = 1, \ldots, k$. By convention if $k > |w|$ then we set the convolution product to zero. Now recall our endomorphism $\mathcal{F} \in \mathrm{End}(\mathbb{R}\langle\mathbb{A}\rangle_*)$ generated when we computed the function F of the flowmap. We can now express $\mathcal{F}$ as follows,

$$\begin{aligned}\mathcal{F}(w) &= \sum_{k=1}^{|w|} c_k \sum_{u_1 u_2 \cdots u_k = w} u_1 * u_2 * \cdots * u_k \\ &= \sum_{k \geqslant 1} c_k \mathcal{I}^{\star k}(w).\end{aligned}$$

In other words we can identify the flow map with the identity endomorphism $\mathcal{I}$ as we have already indicated, and we can identify the function F of the flowmap with the endomorphism

$$\mathcal{F}^\star(\mathcal{I}) = \sum_{k \geqslant 1} c_k \mathcal{I}^{\star k}.$$

We have now essentially completed our second level of abstraction. The idea now is as follows. We assume a given fixed stochastic differential system, i.e. one for which the driving stochastic processes and governing vector fields are fixed. The extended alphabet and words are thus fixed. Then different numerical integration schemes based on the map-truncate-invert approach, are distinguished by their actions on those words represented by the corresponding endomorphisms acting on them. On further piece in the puzzle is required, a measure to compare the different remainders generated by different endomorhpisms $\mathcal{F}^\star(\mathcal{I})$. This is provided by constructing an inner product on $\mathrm{End}(\mathbb{R}\langle\mathbb{A}\rangle_*)$ via a corresponding expectation map which associates the correct expectation with any word w, i.e. repeated integral I_w, constructed from the extended alphabet.

2. A brief overview of the algebraic concepts

In this section we briefly collect the formal definitions that are at play in our work. For details the reader is invited to consult [13,17–19]. An algebra over the field $\mathbb{K}$ (we took $\mathbb{K} = \mathbb{R}$ above) is denoted by (A, m_A, η_A) where A is a $\mathbb{K}$-vector space equipped with an associative product $m_A : A \otimes A \to A$, that is $m_A \circ (m_A \otimes \mathrm{id}_A) = m_A \circ (\mathrm{id}_A \otimes m_A) : A \otimes A \otimes A \to A$ and a unit map $\eta_A : \mathbb{K} \to A$. A $\mathbb{K}$-coalgebra $(C, \Delta_C, \epsilon_C)$ consists of a $\mathbb{K}$-vector space C carrying a coassociative coproduct map $\Delta_C : C \to C \otimes C$, i.e. $(\Delta_C \otimes \mathrm{id}_C) \circ \Delta_C = (\mathrm{id}_C \otimes \Delta_C) \circ \Delta_C : C \to C \otimes C \otimes C$. The counit map $\epsilon_C : C \to \mathbb{K}$ satisfies $(\epsilon_C \otimes \mathrm{id}_C) \circ \Delta_C = \mathrm{id}_C = (\mathrm{id}_C \otimes \epsilon_C) \circ \Delta_C$. Its kernel $\ker(\epsilon_C) \subset C$ is called the augmentation ideal. A coalgebra is called cocommutative if $\tau \circ \Delta_C = \Delta_C$, where $\tau : C \otimes C \to C \otimes C$ is the flip map $\tau(x \otimes y) := y \otimes x$. In the following we use Sweedler's notation for the coproduct of an element $x \in C$, $\Delta_C(x) = \sum x^{(1)} \otimes x^{(2)} \in C \otimes C$. The notion of bialgebra combines both algebra and coalgebra in a compatible way. It consists of a $\mathbb{K}$-algebra (B, m_B, η_B) which is also a $\mathbb{K}$-coalgebra $(B, \Delta_B, \epsilon_B)$, such that the multiplication m_B

and the unit η_B are morphisms of $\mathbb{K}$-coalgebras with the natural coalgebra structure on the space $B \otimes B$. The commutativity of the following diagrams encodes these compatibilities

$$\begin{array}{ccc} B\otimes B & \xrightarrow{m_B} & B \\ {\scriptstyle \tau_2(\Delta_B\otimes\Delta_B)}\downarrow & & \downarrow{\scriptstyle \Delta_B} \\ B\otimes B\otimes B\otimes B & \xrightarrow{m_B\otimes m_B} & B\otimes B \end{array} \qquad \begin{array}{ccc} B\otimes B & \xrightarrow{\epsilon_B\otimes\epsilon_B} & \mathbb{K}\otimes\mathbb{K} \\ {\scriptstyle m_B}\downarrow & & \downarrow{\scriptstyle m_{\mathbb{K}}} \\ B & \xrightarrow{\epsilon_B} & \mathbb{K} \end{array}$$

$$\begin{array}{ccc} \mathbb{K} & \xrightarrow{\eta_B} & B \\ {\scriptstyle \Delta_{\mathbb{K}}}\downarrow & & \downarrow{\scriptstyle \Delta_B} \\ \mathbb{K}\otimes\mathbb{K} & \xrightarrow{\eta_B\otimes\eta_B} & B\otimes B \end{array} \qquad \begin{array}{ccc} \mathbb{K} & \xrightarrow{\eta_B} & B \\ & {\scriptstyle \mathrm{id}_{\mathbb{K}}}\searrow \quad \swarrow{\scriptstyle \epsilon_B} & \\ & \mathbb{K} & \end{array},$$

where $\tau_2 := (\mathrm{id}_B \otimes \tau \otimes \mathrm{id}_B)$. Equivalently, Δ_B and ϵ_B are morphisms of $\mathbb{K}$-algebras with the natural algebra structure on the space $B \otimes B$. By a slight abuse of notation one writes $\Delta_B(m_B(b \otimes b')) = \Delta_B(b)\Delta_B(b')$ for $b, b' \in B$, saying that the *coproduct of the product is the product of the coproducts*. The identity element in B will be denoted by $\mathbf{1}_B$ and we assume that all algebra morphisms are unital. A graded bialgebra B consists of $\mathbb{K}$-vector spaces B_n, $n \geq 0$, such that

(i) $B = \bigoplus_{n\geq 0} B_n$,
(ii) $m_B(B_n \otimes B_m) \subseteq B_{n+m}$,
(iii) $\Delta_B(B_n) \subseteq \bigoplus_{p+q=n} B_p \otimes B_q$.

An element $x \in B_n$ is given the degree $\deg(x) = n$. For a connected graded bialgebra B, the degree zero part is $B_0 = \mathbb{K}\mathbf{1}_B$. A graded bialgebra $B = \bigoplus_{n\geq 0} B_n$ is of finite type if its homogeneous components B_n are finite dimensional $\mathbb{K}$-vector spaces. One can show [13] that the coproduct of any element $x \in B$ is given by

$$\Delta_B(x) = \sum x^{(1)} \otimes x^{(2)} = x \otimes \mathbf{1}_B + \mathbf{1}_B \otimes x + \sum{}' x' \otimes x'',$$

where

$$\Delta'(x) := \sum{}' x' \otimes x'' \in \bigoplus_{\substack{p+q=n \\ p>0,\ q>0}} B_p \otimes B_q$$

is the so-called reduced coproduct, which is coassociative on the augmentation ideal $\ker(\epsilon_B) := B^+ := \bigoplus_{n>0} B_n$. Here a variant of Sweedler's notation is used for the reduced coproduct $\Delta'(x) = \sum{}' x' \otimes x''$. Elements in the kernel of Δ'_B are called primitive elements of B. Note that $\mathbf{1}_B$ is a so-called group-like element, i.e. $\Delta_B(\mathbf{1}_B) = \mathbf{1}_B \otimes \mathbf{1}_B$.

Let A be a $\mathbb{K}$-algebra and C a $\mathbb{K}$-coalgebra C. The convolution product of two linear maps $f, g \in \mathrm{Hom}_{\mathbb{K}}(C, A)$ gives the linear map

$$(f \star g)(a) := m_A \circ (f \otimes g) \circ \Delta_C(a) = \sum f(a^{(1)})\, g(a^{(2)}),$$

for any $a \in C$. In other words the convolution is given by the sequence

$$C \xrightarrow{\Delta_C} C \otimes C \xrightarrow{f\otimes g} A \otimes A \xrightarrow{m_A} A.$$

From associativity of A and coassociativity of C follows that $\mathrm{Hom}_{\mathbb{K}}(C, A)$ with the convolution product $f \star g$ for $f, g \in \mathrm{Hom}_{\mathbb{K}}(C, A)$ is an unital associative $\mathbb{K}$-algebra with unit $\eta := \eta_A \circ \epsilon_C$ [13]. For a bialgebra B one can define the convolution algebra structure on $\mathrm{Hom}_{\mathbb{K}}(B, B)$ with unit $\eta := \eta_B \circ \epsilon_B$. For maps $f_i \in \mathrm{Hom}_{\mathbb{K}}(C, A)$, $i = 1, \ldots, n$, $n > 1$, we define the product

$$f_1 \star f_2 \star \cdots \star f_n := m_A \circ (f_1 \otimes f_2 \otimes \cdots \otimes f_n) \circ \Delta_C^{(n-1)},$$

where $\Delta_C^{(0)} := \mathrm{id}_C$, and for $n > 0$, $\Delta_C^{(n)} := (\Delta_C^{(n-1)} \otimes \mathrm{id}_C) \circ \Delta_C$.

A Hopf algebra is a $\mathbb{K}$-bialgebra H together with a $\mathbb{K}$-linear map $S : H \to H$ called the antipode; see [17,19] for details. The antipode S has the property of being an antihomomorphism, i.e. $S(m_H(x \otimes y)) = m_H(S(y) \otimes S(x))$ and $\Delta_H \circ S = (S \otimes S) \circ \tau \circ \Delta_H$. The necessarily unique antipode S is the inverse of the identity map $\mathrm{id}_H : H \to H$ with respect to the convolution product $S \star \mathrm{id}_H = \eta_H \circ \epsilon_H = \mathrm{id}_H \star S$. If the Hopf algebra H is commutative or cocommutative, then $S \circ S = \mathrm{id}_H$. It is a well-known result [13] that any connected graded bialgebra $H = \bigoplus_{n \geq 0} H_n$ is a connected graded Hopf algebra. The antipode is defined as the convolution inverse of the identity map, $S := \mathrm{id}_H^{\star(-1)}$. A character $\phi \in \mathrm{Hom}_{\mathbb{K}}(H, A)$ is defined to be a unital algebra morphism taking values in a commutative $\mathbb{K}$-algebra A. An infinitesimal character consists of a linear map ξ such that $\xi(m_H(x \otimes y)) = 0$ for any $x, y \in \ker(\epsilon_H)$. This implies that $\xi(\mathbf{1}_H) = 0$, where $\mathbf{1}_H$ is the algebra unit in H.

3. Platen and Bruti–Liberati form

We show that the stochastic Taylor expansion derived in Platen & Bruti-Liberati [15] is an equivalent though different representation of the stochastic Taylor expansion we give in Theorem 2.1 of the manuscript. The equivalence of the expansions is seen from the identity

$$\int_0^t \int_{\mathbb{R}^{\ell-d}} (\tilde{V}_{-1} \circ f)(y_{s_-}, v)\, \overline{Q}(\mathrm{d}v, \mathrm{d}s) = \sum_{i=d+1}^{\ell} \int_0^t \int_{\mathbb{R}} (\tilde{V}_i \circ f)(y_{s_-}, v)\, \overline{Q}^i(\mathrm{d}v, \mathrm{d}s).$$

The expression on the left is the encoding employed by Platen and Bruti-Liberati of the jump terms in the stochastic differential equation while our encoding is that on the right. The equivalence is explained as follows. On the left above $\overline{Q}$ is a compensated Poisson random measure on $\mathbb{R}^\ell \times \mathbb{R}_+$. The Lévy-driven equation may be written in this form, where $V_{-1}(x, v) = \sum_{i=1}^{\ell-d} v_i V_{i+d}(x)$ and $\overline{Q}$ is the compensation of the Poisson random measure Q defined such that $Q(B, (a, b]) = \#\{\Delta J_s^1 \in B_1, \ldots, \Delta J_s^\ell \in B_\ell : \ s \in (a, b]\}$ for a Borel set $B = B_1 \times \cdots \times B_\ell \subset \mathbb{R}^\ell$ bounded away from the origin, i.e. $0 \notin \bar{B}$. As independent Lévy processes almost surely never jump simultaneously (see Cont & Tankov [3, Theorem 5.3]), the measure Q is concentrated on sets of the form $0 \times \cdots \times 0 \times B_i \times 0 \times \cdots \times 0$ with intensity measure $\rho(0 \times \cdots \times 0 \times B_i \times 0 \times \cdots \times 0)\mathrm{d}t = \rho^i(B_i)\mathrm{d}t$. The identity above thus follows. The stochastic Taylor expansion given in Platen & Bruti-Liberati [15] is of the same form as the expansion we have given, but where the alphabet is instead $\{-1, 0, \ldots, d\}$, and the operator associated to the letter -1 is the multi-dimensional shift $\tilde{V}_{-1} : f(y) \mapsto f\big(y + V_{-1}(y, v)\big) - f(y)$.

4. Linear vector fields and linear diffeomorphisms

The separated stochastic Taylor expansion has a simple form when the governing vector fields are linear with constant coefficients and we consider the action of the flowmap on homogeneous linear diffeomorphisms, say $f(y) = Fy$, where $F = [f_{ij}]$ is an $N \times N$ matrix. The identity map is a special case of such a linear diffeomorphism which generates the solution y_t directly. In this case, the operators in the expansion are given by matrix multiplication. We write $V_i(y) = A^i y$, where $A^i = [a^i_{jk}]$ are constant $N \times N$ matrices and consider the action of the flowmap on linear functions. First, for $i = 1, \ldots, d$ by direct computation we find $\tilde{V}_i \circ f(y) = FA^i y$. Moreover, for $i = d+1, \ldots, \ell$ the higher order differential operators $\tilde{V}_{i(m)}$ with $m \geq 2$ vanish due to the linearity, and we obtain $(\tilde{V}_i \circ f)(y, v) = v(V_i \cdot \nabla) f(y) = vFA^i y$. The above relation shows in addition that the term in $\tilde{V}_0 \circ f$ involving an integral over the jump sizes vanishes. As the functions we act on are linear, the second order terms of $\tilde{V}_0$ also vanish. We therefore have $\tilde{V}_0 \circ f(y) = f(V_0(y)) = FA^0 y$. The operators $\tilde{V}_i$ act by matrix multiplication. It follows that the operators $\tilde{V}_w$ act on linear functions by matrix multiplication in the reverse order, $\tilde{V}_{a_1 \ldots a_k} \circ f(y) = FA^{a_k} \cdots A^{a_1} y$.

5. Numerical experiments

Our numerical investigations principally concern the equation $\mathrm{d}y_t = A_0 y_t \mathrm{d}t + A_1 y_t \mathrm{d}W_t^1 + V_2(y_t)\mathrm{d}W_t^2 + A_3 y_t \mathrm{d}\tilde{N}_t$, where W_t^1, W_t^2 are Wiener processes and $\tilde{N}_t$ is a standard Poisson process with intensity λ. Here V_2 is the nonlinear vector field

$$V_2(x_1, x_2, x_3, x_4) = \big(\sin(x_1), \cos(x_2), x_4, -\sin(x_3)\big)^T,$$

whilst the constant coefficient linear vector fields are defined by the following matrices

$$A_0 = \begin{pmatrix} 0.314724 & 0.132359 & 0.457507 & 0.457167 \\ 0.405792 & -0.402460 & 0.464889 & -0.014624 \\ -0.373013 & -0.221502 & -0.342387 & 0.300280 \\ 0.413376 & 0.046882 & 0.470593 & -0.358114 \end{pmatrix}$$

$$A_1 = \begin{pmatrix} -0.078239 & 0.155741 & 0.178735 & 0.155478 \\ 0.415736 & -0.464288 & 0.257740 & -0.328813 \\ 0.292207 & 0.349129 & 0.243132 & 0.206046 \\ 0.459492 & 0.433993 & -0.107773 & -0.468167 \end{pmatrix}$$

$$A_3 = \begin{pmatrix} -0.223077 & 0.194829 & -0.061256 & -0.313127 \\ -0.453829 & -0.182901 & -0.118442 & -0.010236 \\ -0.402868 & 0.450222 & 0.265517 & -0.054414 \\ 0.323458 & -0.465554 & 0.295200 & 0.146313 \end{pmatrix}.$$

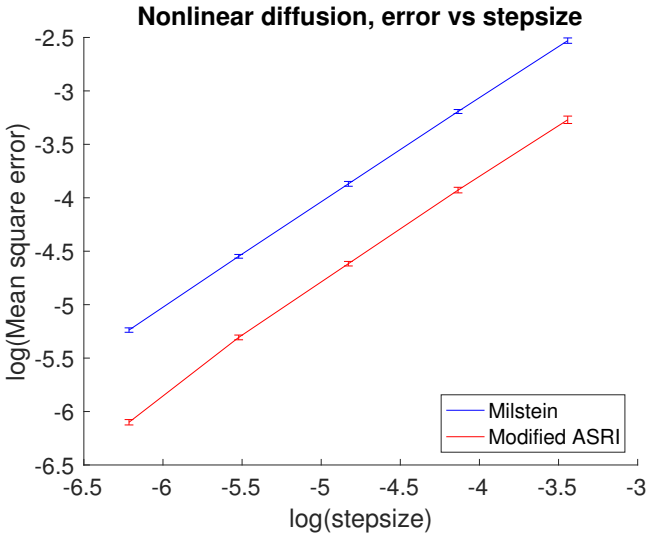
We also considered the special cases where: (i) V_2 was set to zero, so we have only linear coefficients ('Linear jump diffusion'), and (ii) $\tilde{N}_t$ was set to zero, so there are only continuous driving processes ('Nonlinear diffusion'). We compare the global mean square error $E(\sup_{0\le t\le T}|y_t - \hat{y}_t|^2)^{1/2}$, estimated by sampling 1000 paths, for two approximations of $\hat{y}_t$: the Milstein scheme and the order 1 modified antisymmetric sign reverse integrator (mASRI). The intensity of $\tilde{N}_t$ was taken to be $\lambda = 50$ and the initial condition was $y_0 = (1, 0.8, 0.6, 0.4)^T$.

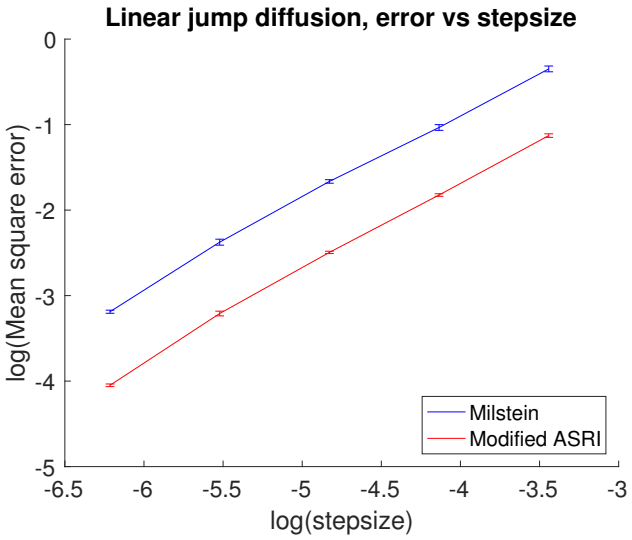
We briefly describe the methods used to generate the iterated integrals. In the absence of jumps, the integrals $\int_t^{t+h} W_s^1 dW_s^2$ and $\int_t^{t+h} W_s^2 dW_s^1$ were simulated using a truncated Fourier transform, as detailed in Kloeden and Platen [10]. Where one or more jumps occur in a timestep, the situation changes. To simulate the integrals $\int_t^{t+h} W_s^i dN_s$ and $\int_t^{t+h} N_s dW_s^i$, we simulate the W_t^i at each jump time, which allows for exact computation of the aforementioned integrals. The integrals $\int_t^{t+h} W_s^1 dW_s^2$ and $\int_t^{t+h} W_s^2 dW_s^1$ are not independent of the values of W_t^i at the jump times, so the truncated Fourier method becomes impractical. We therefore simulate the W_t^i on a finer grid, incorporating the jump times, and use the trapezoidal approximation of the integrals, see Milstein and Tretyakov [14]. There are two relevant parameters pertaining to the simulation of the iterated Wiener integrals: the number of terms p retained in the Fourier series, and the number of points M used in the trapezoidal approximation. To obtain comparable accuracy for the two methods we take $M = 5(p+1)$. For integrators of order 1, p should scale like $1/h$, where h is the timestep, see Milstein and Tretyakov [14].

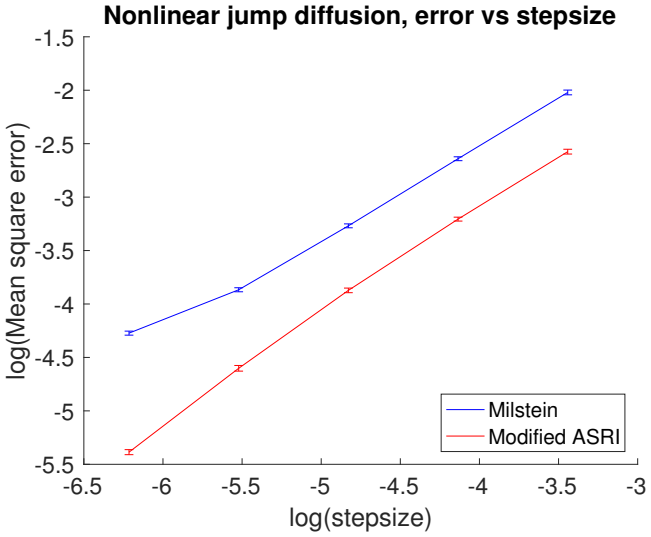
As the equation has no explicit solution, we estimate the mean square error by comparing the sample paths simulated with those obtained by a numerical scheme for the same simulated driving processes and iterated integrals, but employing a smaller time step. In practice, we simulate only at the finest scale, and then extrapolate the values to the coarser scale. To obtain estimates of the computational time, we run and time the code to generate the iterated integrals at the coarser scales remembering to scale p with $1/h$. It is important to note that the mean square estimates obtained by this method does not include the simulation error for the iterated integrals. We therefore experimented with different values of p on the coarsest time step (in the figures below the values of p quoted correspond to the value for p used at the most coarse scale; it is increased in proportion with $1/h$ for smaller stepsizes). The relative size of the simulation time and the evaluation time is key to the effectiveness of the (m)ASRI scheme. As the simulation time

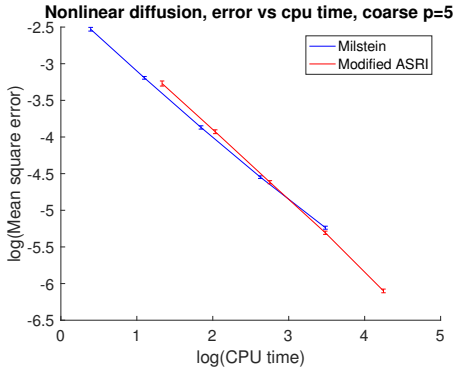
increases with p, and p scales like $1/h$, it can be shown that below a critical step size, the simulation time dominates the evaluation time, see Lord, Malham and Wiese [11]. This step size depends on the value of p at the coarse scale, and is lower for the (m)ASRI scheme than the Milstein scheme, due to the increased evaluation cost of the former.

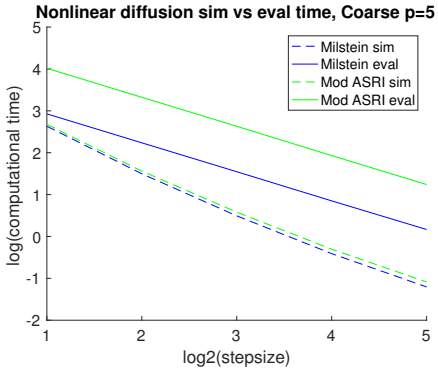
We show the results of our simulations. In accordance with the theory, in all cases the mASRI scheme has lower mean square error for a given timestep than the Milstein scheme. We then plot the mean square error against the CPU time taken in each case, together with the plot of simulation time versus evaluation time. The time steps employed are typically such that these values are comparable, although in practice smaller time steps would frequently be used. In all cases, the evaluation time of the mASRI scheme was greater than the simulation time, whilst the simulation time often overtook the evaluation time of the Milstein scheme as the timestep became smaller. Below its critical step size, the mASRI scheme will always be more efficient than the Milstein scheme. Generically, it tends to outperform the Milstein scheme also for step sizes below the critical step size of the Milstein scheme, but above the mASRI critical size. If less accuracy is required, the Milstein scheme may be preferable, although this is not always the case.

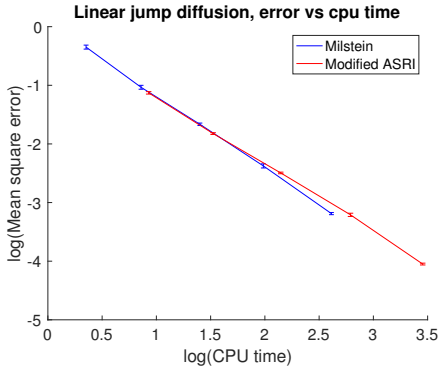


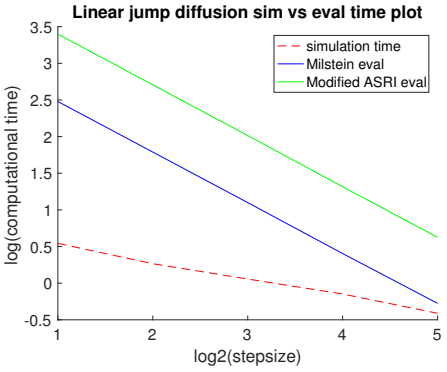

Linear jump diffusion, error vs stepsize
log(Mean square error)
log(stepsize)
Milstein
Modified ASRI

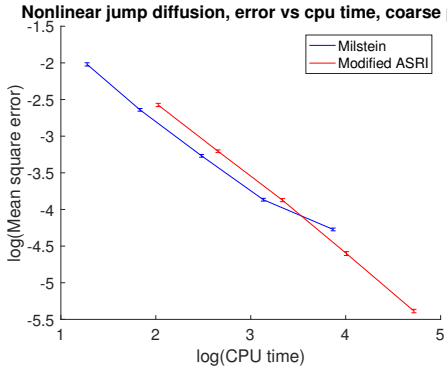

Nonlinear jump diffusion, error vs stepsize
log(Mean square error)
log(stepsize)
Milstein
Modified ASRI

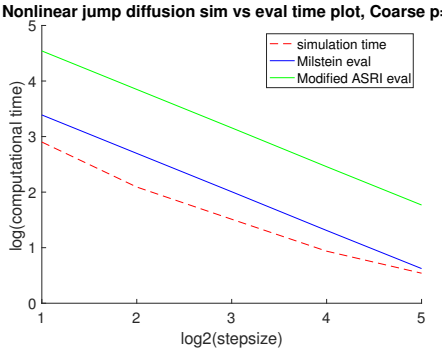

Nonlinear diffusion, error vs cpu time, coarse p=5
log(Mean square error)
log(CPU time)
Milstein
Modified ASRI

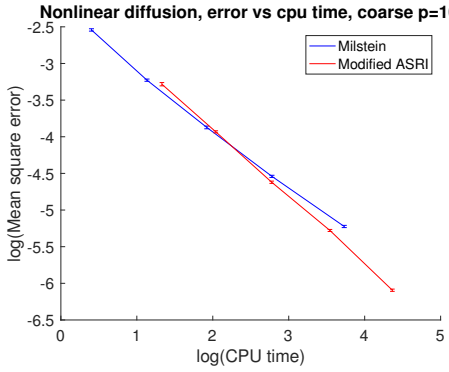

Nonlinear diffusion sim vs eval time, Coarse p=5
log(computational time)
log2(stepsize)
Milstein sim
Milstein eval
Mod ASRI sim
Mod ASRI eval

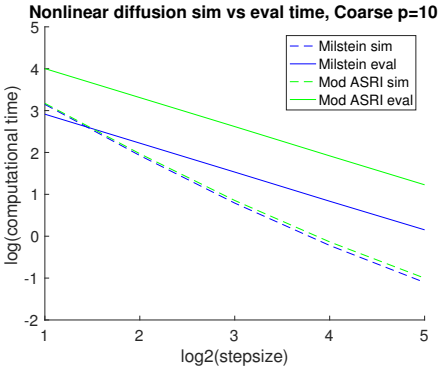

Linear jump diffusion, error vs cpu time
Milstein
Modified ASRI
log(Mean square error)
log(CPU time)

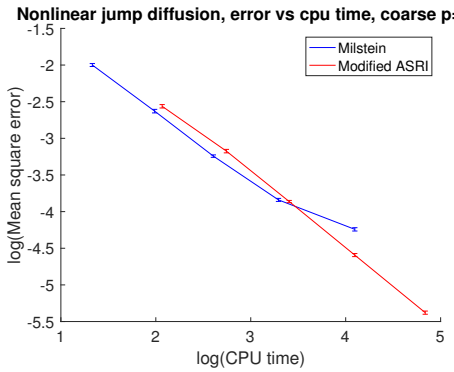

Linear jump diffusion sim vs eval time plot
simulation time
Milstein eval
Modified ASRI eval
log(computational time)
log2(stepsize)

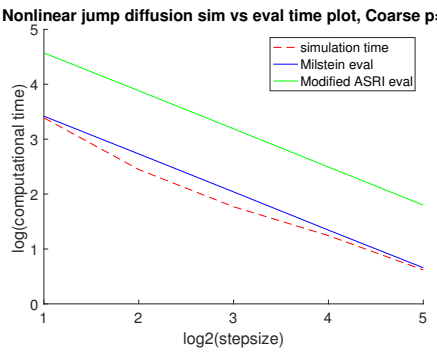

Nonlinear jump diffusion, error vs cpu time, coarse p=5
Milstein
Modified ASRI
log(Mean square error)
log(CPU time)

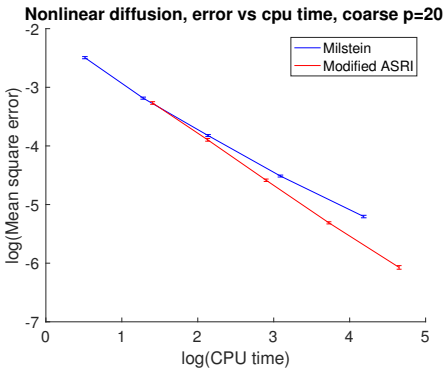

Nonlinear jump diffusion sim vs eval time plot, Coarse p=5
simulation time
Milstein eval
Modified ASRI eval
log(computational time)
log2(stepsize)

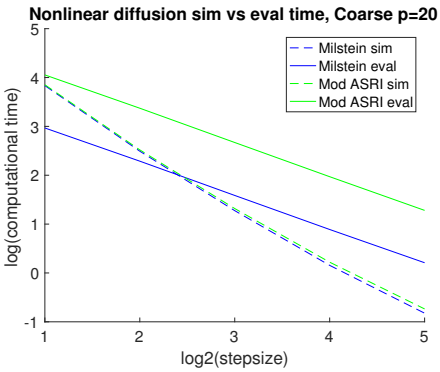

Nonlinear diffusion, error vs cpu time, coarse p=10
Milstein
Modified ASRI
log(Mean square error)
log(CPU time)

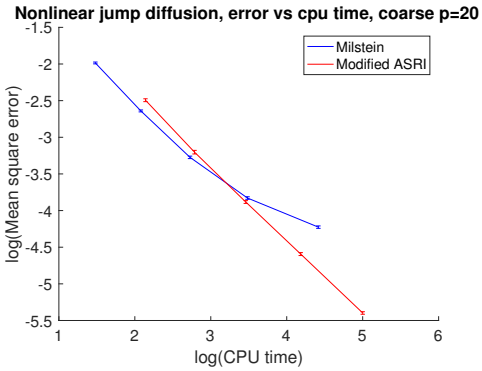

Nonlinear diffusion sim vs eval time, Coarse p=10
Milstein sim
Milstein eval
Mod ASRI sim
Mod ASRI eval
log(computational time)
log2(stepsize)

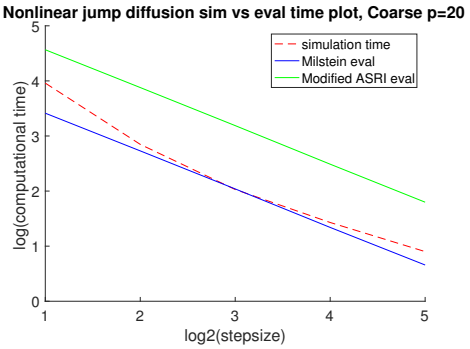

Nonlinear jump diffusion, error vs cpu time, coarse p=10
Milstein
Modified ASRI
log(Mean square error)
log(CPU time)


Nonlinear jump diffusion sim vs eval time plot, Coarse p=10
simulation time
Milstein eval
Modified ASRI eval
log(computational time)
log2(stepsize)


Nonlinear diffusion, error vs cpu time, coarse p=20
Milstein
Modified ASRI
log(Mean square error)
log(CPU time)


Nonlinear diffusion sim vs eval time, Coarse p=20
Milstein sim
Milstein eval
Mod ASRI sim
Mod ASRI eval
log(computational time)
log2(stepsize)


Nonlinear jump diffusion, error vs cpu time, coarse p=20
Milstein
Modified ASRI
log(Mean square error)
log(CPU time)


Nonlinear jump diffusion sim vs eval time plot, Coarse p=20
simulation time
Milstein eval
Modified ASRI eval
log(computational time)
log2(stepsize)

References

1. Cartier P. 2007 *A primer of Hopf algebras*, in Frontiers in Number Theory, Physics, and Geometry II, Springer, Berlin Heidelberg, 537–615.
2. Castell F, Gaines J. 1996 The ordinary differential equation approach to asymptotically efficient schemes for solutions of stochastic differential equations, *Ann. Inst. H. Poincaré Probab. Statist.* **32**, 231–250.
3. Cont R, Tankov P. 2004 *Financial Modelling with Jump Processes*, Chapman & Hall/CRC.
4. Curry C, Ebrahimi–Fard K, Malham SJA, Wiese A. 2014 Lévy processes and quasi-shuffle algebras. *Stochastics* **86**(4), pp. 632–642.
5. Ebrahimi–Fard K, Lundervold A, Malham SJA, Munthe–Kaas H, Wiese A. 2012 Algebraic structure of stochastic expansions and efficient simulation. *Proc. R. Soc. A* doi:10.1098/rspa.2012.0024.
6. Ebrahimi–Fard K, Malham SJA, Patras F, Wiese A. 2015 Flows and stochastic Taylor series in Itô calculus, *J. Phys. A.* **48**, 495202 (17pp). (doi:10.1088/1751-8113/48/49/495202)
7. Ebrahimi–Fard K, Malham SJA, Patras F, Wiese A. 2015 The exponential Lie series for continuous semimartingales, *Proc. R. Soc. A* **471**. (doi:10.1098/rspa.2015.0429)
8. Hoffman ME. 2000 Quasi-shuffle products. *Journal of Algebraic Combinatorics* **11**, pp. 49–68.
9. Hudson RL. 2012 Sticky shuffle product Hopf algebras and their stochastic representations, in "New trends in stochastic analysis and related topics", *Interdiscip. Math. Sci.* **12**, World Sci. Publ., Hackensack, NJ, 165–181.
10. Kloeden PE, Platen E. 1999 *Numerical Solution of Stochastic Differential Equations* (3rd Printing), Berlin etc: Springer.
11. Lord G, Malham SJA, Wiese A. 2008 Efficient strong integrators for linear stochastic systems, *SIAM J. Numer. Anal.* **46**(6), 2892–2919.
12. Malham SJA, Wiese A. 2009 Stochastic expansions and Hopf algebras, *Proc. R. Soc. A* **465**, 3729–3749. (doi:10.1098/rspa.2009.0203)
13. Manchon D. 2008 *Hopf algebras and renormalisation*, in Handbook of Algebra, Vol. 5, M. Hazewinkel, Ed., Elsevier, Amsterdam, 365–427.
14. Milstein GN, Tretyakov MV. 2003 *Stochastic Numerics for Mathematical Physics*, Berlin etc: Springer.
15. Platen E, Bruti–Liberati N. 2010 *Numerical Solution of Stochastic Differential Equations with Jumps in Finance.* Berlin etc: Springer.
16. Protter PE. 2005 *Stochastic Integration and Differential Equations*. Second Edition, Springer.
17. Radford DE. 2012 *Hopf Algebras*, World Scientific Press, Singapore, 2012.
18. Reutenauer C. 1993 *Free Lie algebras*. London Mathematical Society Monographs New Series 7, Oxford Science Publications.
19. Sweedler ME. 1969 *Hopf Algebras*, W. A. Benjamin, Inc., New York.