SIAM J. NUMER. ANAL.
Vol. 46, No. 4, pp. 1732–1768
© 2008 Society for Industrial and Applied Mathematics

ADAPTIVE WEAK APPROXIMATION OF DIFFUSIONS WITH JUMPS*

E. MORDECKI†, A. SZEPESSY‡, R. TEMPONE§, AND G. E. ZOURARIS¶

Abstract. This work develops adaptive time stepping algorithms for the approximation of a functional of a diffusion with jumps based on a jump augmented Monte Carlo Euler–Maruyama method, which achieve a prescribed precision. The main result is the derivation of new expansions for the time discretization error, with computable leading order term in a posteriori form, which are based on stochastic flows and discrete dual backward functions. Combined with proper estimation of the statistical error, they lead to efficient and accurate computation of global error estimates, extending the results by A. Szepessy, R. Tempone, and G. E. Zouraris [*Comm. Pure Appl. Math.*, 54 (2001), pp. 1169–1214]. Adaptive algorithms for either deterministic or trajectory-dependent time stepping are proposed. Numerical examples show the performance of the proposed error approximations and the adaptive schemes.

Key words. diffusions with jumps, weak approximation, error control, Euler–Maruyama method, a posteriori error estimates, backward dual functions

AMS subject classifications. 65C30, 65Y20, 65L50, 65H35, 60H35, 60H10

DOI. 10.1137/060669632

1. Introduction.

Since the introduction by Kiyoshi Itô of the theory of stochastic integration [26], diffusion processes have become a benchmark model in the mathematical sciences. Nevertheless, the study of more complex phenomena, including abrupt changes, pathwise features different from the ones of diffusions, and heavy tails, leads to the introduction of models with jumps.

In our view, the introduction of jumps in stochastic modeling is realized at least in the following three major directions. First, diffusions with jumps, the case we consider in this work, are obtained by adding jumps to a diffusion-type dynamics. Jumps have exponentially distributed interarrival times and amplitudes generated by time-dependent mark distributions. This jump structure ensures the unpredictability of the epoch and the magnitude of the jumps, a fact that is observed in financial markets, where it leads to the incompleteness of the market. This first approach was initiated with the work by Merton [37]. For other models we refer to Cont and Tankov [11] and the references therein. In a second approach, the jump intensity depends on the state of the process, implying a certain possibility to predict the jump

*Received by the editors September 12, 2006; accepted for publication (in revised form) January 11, 2008; published electronically April 11, 2008. This work was supported by Swedish Research Council for Engineering Science (TFR) grant 222-148, the Swedish National Network in Applied Mathematics (NTM), start up funds from the School of Computational Science, Florida State University, the Dahlquist research fellowship at KTH, Project 10.101 FCE - DINACYT, Uruguay, CSIC - Udelar, Human Resources Program, the Facultad de Ciencias, Udelar, Montevideo, Uruguay, and the Institute of Applied and Computational Mathematics, FO.R.T.H., Greece.

http://www.siam.org/journals/sinum/46-4/66963.html

†Centro de Matemática, Facultad de Ciencias, Universidad de la República, Iguá 4225 C.P. 11400, Montevideo, Uruguay (mordecki@cmat.edu.uy).

‡Department of Mathematics, KTH, S-100 44 Stockholm, Sweden (szepessy@nada.kth.se).

§Department of Mathematics and School of Computational Science, Florida State University, 400 Dirac Science Library, Tallahassee, FL 32306-4120, and School of Computer Sciences and Communication, KTH, S-100 44 Stockholm, Sweden (rtempone@scs.fsu.edu).

¶Department of Mathematics, University of Crete, GR-714 09 Heraklion, Crete, Greece, and Institute of Applied and Computational Mathematics, FO.R.T.H., GR-711 10 Heraklion, Crete, Greece (zouraris@math.uoc.gr).

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

epoch. State-dependent jump diffusion processes are intensively used in default risk theory, for instance, in the form of affine jump diffusion processes [13]. The third approach assumes that not only the large movements but also the small movements observed in the dynamics are generated by jumps. In some works even the Brownian driving process is replaced by a stable-type process (i.e., a Lévy process with jump measure with infinite mass around zero). This gives a process with an infinite number of jumps on every bounded time interval, a situation known as *infinite activity*. Initial references to infinite activity are, for instance, [34] and [17]. More recent works include [10].

The computation of functionals of the above-mentioned stochastic processes, when closed solutions are not available, led to the development of time discretization schemes, following the approach by Maruyama [35]. In this wide field of applied mathematics, standard references are the books by Kloeden and Platen [31] and by Milstein and Tretyakov [39]. The main emphasis in these books is on the development of different numerical schemes and the determination of their *order* of convergence. Consider a given functional θ (for instance, an expectation, related to an option price in finance) of a stochastic process defined on a time interval $[0, T]$. Assuming the necessary regularity, we say that a scheme has (weak) order r if it produces a sample average approximate value, $\widehat{\theta}$, such that $|\widehat{\theta}-\theta| = \mathcal{O}\big((\Delta t_{max})^r + M^{-1/2}\big)$, using a time discretization $\{0 = t_0 < t_1 < \cdots < t_N = T\}$ and M independent samples.

Motivated by practical reasons, we formulate in this work a weak approximation problem in the following way: given the functional θ, the numerical scheme for time discretization, and a certain desired accuracy $\mathrm{TOL} > 0$, produce a *computable* sample average estimator $\widehat{\theta}$ such that $|\widehat{\theta} - \theta| \leq \mathrm{TOL}$ with a probability close to one. We also seek to construct $\widehat{\theta}$ using minimal computational effort.

To solve this problem we use the so-called jump augmented Euler–Maruyama scheme (also known as jump adapted Euler–Maruyama scheme; cf. [47]) and develop novel time discretization error estimation formulas. Based on these computable error estimates we propose *a posteriori adaptive* methods that compute the approximation $\widehat{\theta}$ and estimate its numerical error due to time discretization and statistical sampling. If the prescribed accuracy TOL is not yet achieved, the adaptive algorithms decide whether to subdivide a set of time intervals with largest error contribution or to increase the number of realizations, up to the point of achieving the accuracy goal. To the best of our knowledge, there are no references that follow this approach for any class of jump processes.

Regarding the jump intensity λ, we consider a moderate number of jumps, motivated by reported results from the calibration of jump diffusion models in finance. For instance, the value $\lambda = 0.089$ was reported for the calibration of the Merton model with option prices on the S&P 500 index, and $\lambda \approx 1$ for the Kou model applied to the German DAX index; see [1] and [12]. These empirical facts allow us to rely on the augmentation of the grid with the jump times, reported to be more accurate for strong approximation of diffusion with jumps [6]. In Remark 4.1 we discuss the corresponding additional work due to the inclusion of the jump points in the time discretization.

The main contribution in this work is the generalization of the approach proposed by Szepessy, Tempone, and Zouraris for diffusions in [51] to the context of diffusion with large jumps. This generalization is based on the probabilistic properties of diffusions with jumps, performing two steps of simulation. The first step relates to the jump structure, and the second relates to the diffusion part. We provide

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

corresponding analysis for the approximation errors in each step. In section 2 we present an example with a singularity and many jumps. There, our new algorithms save a factor of one hundred in the number of time steps required to produce the same accuracy as the uniform time stepping algorithm. Furthermore, for one of those algorithms the additional computational work to generate the adaptive time steps is negligible.

The other two cases not considered here, namely, diffusions with state-dependent intensity jumps and high intensity of small jumps, seem to require other techniques and would therefore be interesting for future work; see the end of section 1.2.1.

Efficient adaptive time stepping methods, with theoretical basis, use a posteriori error information, since in general a priori knowledge is not as precise as the a posteriori one. Adaptive time stepping methods can be based on Theorem 2.2 or Theorem 3.1, where error estimates with leading order terms in computable a posteriori form are obtained. These estimations require the use of stochastic flows and dual functions in the presence of jumps. Dual functions are standard in optimal control theory, in particular for adaptive mesh control for ordinary and partial differential equations (see [4], [5], [16], [28], [29], [41], [43], and [44]) and were successfully applied in the probabilistic context in [15], [40], [45], and [51].

Alternative deterministic methods involve the numerical approximation of the Kolmogorov backward equation (2.5), which is a nonlocal partial integrodifferential equation. Direct discretization of such time-dependent integrodifferential equations requires the approximation of integrals at each time step and may suffer from the curse of dimensionality. On the other hand, the Euler–Maruyama method avoids these expensive computations. Examples of this deterministic approach include the work by Andersen and Andreasen [1], which uses operator splitting to approximate the integral term explicitly in Fourier space while approximating the other terms in the equation implicitly, and the work by Matache, von Petersdorff, and Schwab [36], which discretizes the partial integrodifferential equation by the θ-scheme in time and a wavelet Galerkin method in space.

For work which proposes and analyzes numerical methods for diffusions with jumps, we refer the reader to Liu and Li [33], where weak convergence schemes are analyzed, and Higham and Kloeden [22], [23], [24], where implicit methods for diffusion with jumps are presented, analyzing strong convergence and nonlinear stability. We refer also to Bruti-Liberati and Platen [6], [7], [8], [9], Kubilius and Platen [32], and Mikulevičius and Platen [38].

1.1. Problem's setting. Let $d \in \mathbb{N}$, $T > 0$ and let $X : [0,T] \to \mathbb{R}^d$ be a stochastic process solving the following stochastic differential equation:

$$
\begin{aligned}
X(t) = X(0) &+ \int_0^t a(\, s, X(s^-)\,)\, ds + \sum_{\ell=1}^{\ell_0} \int_0^t b^\ell(\, s, X(s^-)\,)\, dW^\ell(s) \\
&+ \int_0^t \int_{\mathbf{z}} c(\, s, X(s^-), z\,)\, p(ds, dz) \quad \forall\, t \in [0,T]
\end{aligned}
\tag{1.1}
$$

(see [19, Chap. II.2], [27, Chap. III.2c], or [25]), with initial condition $X(0) = x_0$ that we assume deterministic for simplicity (see Remark 2.1 for a random initial condition). The randomness in this equation is generated by the following:

(i) A standard ℓ_0-dimensional Wiener process

$$
W = \{W(t) = (W^1(t), \ldots, W^{\ell_0}(t)) \colon t \in [0,T]\},
$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

defined on a filtered probability space $(\Omega^W, \mathcal{F}^W, \mathbf{F}^W, P^W)$, where

$$\mathbf{F}^W = \{\mathcal{F}_t^W : t \in [0,T]\}$$

is a filtration and $\mathcal{F}_T^W = \mathcal{F}^W$.

(ii) A Poisson random measure $p(\omega, dt, dz)$ on $[0,T]\times\mathbf{Z}$, where $\mathbf{Z} \equiv \mathbb{R}^{\ell_1}\backslash\{0\}$, with $\ell_1 \in \mathbb{N}$, and deterministic compensator $q(dt,dz) = \lambda(t)\mu(t,dz)dt$. Here $\lambda\colon [0,T] \to [0,\lambda_{\max}]$ is the time intensity of jumps with $\lambda_{\max} \in (0,+\infty)$ and, for each fixed $t \geq 0$, the measure $\mu(t,dz)$ is a probability on $\mathbf{Z}$. The jump measure is defined in a filtered probability space $(\Omega^J, \mathcal{F}^J, \mathbf{F}^J, P^J)$, where $\mathbf{F}^J = \{\mathcal{F}_t^J : t \in [0,T]\}$ is a filtration and $\mathcal{F}_T^J = \mathcal{F}^J$.

As a consequence, the process X is defined on a stochastic basis $(\Omega, \mathcal{F}, \mathbf{F}, P)$, where $\Omega = \Omega^W \times \Omega^J$, $\mathbf{F} = \{\mathcal{F}_t^W \otimes \mathcal{F}_t^J : t \in [0,T]\}$, and $P = P^W \times P^J$; as usual in this framework, the Wiener process and the Poisson random measure are mutually independent. The coefficients $a\colon [0,T]\times\mathbb{R}^d \to \mathbb{R}^d$, $b\colon [0,T]\times\mathbb{R}^d \to \mathbb{R}^{d\times\ell_0}$, and $c\colon [0,T]\times\mathbb{R}^d\times\mathbf{Z} \to \mathbb{R}^d$ are assumed to be Borel functions, satisfying regularity conditions stated in Lemma 2.1.

Given a scalar function $g\colon \mathbb{R}^d \to \mathbb{R}$, a functional of the stochastic process X of the form $\theta = E\big[\, g\big(X(T)\big)\big]$, and a tolerance $\mathrm{TOL} > 0$, the goal of our work is to derive an algorithm that produces a computable approximation $\widehat{\theta}$ to θ such that

$$\big|\, \widehat{\theta} - \theta \,\big| \leq \mathrm{TOL}$$

with a probability close to one and minimal computational effort.

1.2. The jump augmented Euler–Maruyama simulation scheme. The scheme has two steps. After choosing an initial deterministic partition of the time interval $[0,T]$ we sample the jump epochs and magnitudes, to later implement the numerical method in the *augmented* grid, including the original points and the jump points. We first describe the jump structure construction.

1.2.1. Construction of the integral with respect to the Poisson random measure. Consider a sequence $(\mathbf{e}_j)_{j=1}^{\infty}$ of random variables defined on a probability space $(\Omega^J, \mathcal{F}^J, P^J)$, which are independent and have common exponential distribution with parameter one. Define an auxiliary function

$$\Lambda(t) = \int_0^t \lambda(s)\, ds, \quad t \in [0,T]. \tag{1.2}$$

Then the number of jumps of the random Poisson measure $p(dt,dz)$ in an interval $[0,t]$ is

$$\nu(t) = \max\left\{ k\colon \sum_{j=1}^{k} \mathbf{e}_j \leq \Lambda(t) \right\}$$

and the total number of jumps in $[0,T]$ is denoted by $\nu \equiv \nu(T)$. The jump times of the Poisson measure can be defined by $\tau_0 = 0$, and $\tau_k = \Lambda^{-1}(\mathbf{e}_1 + \cdots + \mathbf{e}_k)$ for $k = 1,\ldots,\nu$, and can be computed recursively by solving for τ_k in the equations

$$\mathbf{e}_k = \int_{\tau_{k-1}}^{\tau_k} \lambda(s)\, dt = \Lambda(\tau_k) - \Lambda(\tau_{k-1}) \quad \text{for} \quad k = 1,\ldots,\nu. \tag{1.3}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Once the jump times are computed, we proceed to sample the marks $\{Z_k\}$, which, conditionally on the values of the jumps times, are independent random variables distributed respectively according to $\{\mu(\tau_k, dz)\}$. Observe that the random measure $p(dt, dz)$, characterized by its intensity $q(dt, dz) = \lambda(t)\mu(t, dz)dt$, can then be constructed as

$$p(ds, dz) = \sum_{k=1}^{\nu} \delta_{(\tau_k, Z_k)}(ds, dz).$$

Here $\delta_{(t,Z)}(ds, dz)$ is a Dirac distribution at the point (t, Z). As a consequence of this construction, the stochastic integral with respect to the Poisson random measure, i.e., the last term in (1.1), can be computed as a finite sum, i.e.,

$$\int_0^t \int_{\mathbf{z}} c(\, s, X(s^-), z\,)\, p(ds, dz) = \sum_{k=1}^{\nu(t)} c\big(\tau_k, X(\tau_k^-), Z_k\big), \quad t \in [0, T].$$

Some relevant remarks concerning the jump structure and its simulation are in order. First, the number of jumps in the interval $[0, T]$ is almost surely finite, its expectation being $\Lambda(T) = \int_0^T \lambda(s)ds \le \lambda_{\max} T$. Second, since the time intensity $\lambda(t)$ is deterministic, the jump times $\{\tau_k\}$ can be directly obtained before solving for the process X; it is enough to find the function Λ by accurately integrating (1.2) and successively finding $\{\tau_k\}$ by solving (1.3). Furthermore, if λ is a constant, then we simply sample $\tau_k = (\mathbf{e}_1 + \cdots + \mathbf{e}_k)/\lambda$. The jump augmented Euler–Maruyama scheme admits a direct generalization in the case of stochastic intensity, as long as this intensity is independent of the Wiener process and it admits exact simulation. This is the case, for instance, of a regime-switching intensity process (see, for instance, [2]).

In the second approach described before, the jump intensity λ depends on the current value $X(t)$ of the process. Then it is not possible to sample the jump times separately from the process and it is necessary to implement a different algorithm. For instance, see Glasserman and Merener [20], [21], where a rejection algorithm is proposed, reducing the case of uniformly bounded state-dependent intensity $\lambda = \lambda(t, x) \le \lambda_o(t) < \lambda_{\max}$ to the case of a time-dependent intensity $\lambda = \lambda_o(t) < \lambda_{\max}$, which is our case.

Regarding the third approach described above and the possibility of adapting our methodology to the case of infinite activity, a possible approach is to separate small and large jumps, simulating the large jumps and approximating the small jumps by an additional independent Brownian motion, as suggested by Asmussen and Rosiński [3]. Nevertheless, one should be cautious, as the efficient use of the jump augmented Euler–Maruyama method assumes a moderate number of jumps, while the cut-off approximation of the small jump part by a Brownian motion may produce a high intensity compound Poisson process. This delicate tradeoff between "small" and "large" jumps is characteristic in infinite activity Lévy processes problems, should be separately examined, and is out of the scope of the present work (see [50]).

1.2.2. Jump augmented Euler–Maruyama time stepping algorithm. After constructing the jumps and its marks as in section 1.2.1, we simulate the trajectory $\overline{X}$ using the following scheme.

Jump augmented Euler–Maruyama simulation scheme.

Input: Given N, time nodes $0 = \tilde{t}_0 < \tilde{t}_1 < \cdots < \tilde{t}_N = T$, and sample jump times $0 < \tau_1(\omega) < \cdots < \tau_{\nu(\omega)}(\omega) < T$ with corresponding marks $Z_1(\omega), \ldots, Z_{\nu(\omega)}(\omega)$,

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

set the jump counter $k = 1$.

Time stepping: Consider an augmented partition given by the union

$$\{t_n(\omega)\}_{n=0}^{\kappa(\omega)} = \{\tilde{t}_n\}_{n=0}^{N} \cup \{\tau_n(\omega)\}_{n=0}^{\nu(\omega)},$$

where $\kappa(\omega) = N + \nu(\omega)$ (a.s.) is the number of time steps.

Set the initial condition $\overline{X}(t_0, \omega) = x_0$.

For time steps $n = 0, \ldots, \kappa(\omega) - 1$

Compute the remaining approximate grid values, $\overline{X}(t_{n+1}, \omega)$, by first constructing the left limit value of the approximated process,

$$\begin{aligned}\overline{X}(t_{n+1}^-, \omega) = \overline{X}(t_n, \omega) &+ a(t_n, \overline{X}(t_n, \omega))(t_{n+1} - t_n) \\ &+ \sum_{\ell=1}^{\ell_0} b^\ell(t_n, \overline{X}(t_n, \omega))(W^\ell(t_{n+1}, \omega) - W^\ell(t_n, \omega)).\end{aligned} \tag{1.4}$$

When needed, introduce the correction due to jump discontinuities:

if $(t_{n+1} = \tau_k(\omega))$ **then**

$$\overline{X}(t_{n+1}, \omega) = \overline{X}(t_{n+1}^-, \omega) + c\big(t_{n+1}, \overline{X}(t_{n+1}^-, \omega), Z_k(\omega)\big), \tag{1.5}$$

increase k to $k + 1$

else

$$\overline{X}(t_{n+1}, \omega) = \overline{X}(t_{n+1}^-, \omega), \tag{1.6}$$

end-if

end-for

Several important particular cases deserve comment. First, taking $c = 0$ we obtain the previous algorithm in [51]. For the other particular case when $a = b = 0$ the approximate process $\overline{X}$ has the same law as X, due to the form of the grid proposed in the numerical method (see [20]). A similar situation occurs when only $b = 0$; there, conditioned on the jumps information, a higher order method for ordinary differential equation integration should be used to approximate $\overline{X}$ between jump times, giving a time discretization error which is essentially negligible with respect to the statistical error.

1.3. Error control. The aim, for a given TOL > 0, is to choose time steps $\{0 = \tilde{t}_0 < \tilde{t}_1 < \cdots < \tilde{t}_N = T\}$ and the number M of independent identically distributed (i.i.d.) samples $\{\overline{X}(\cdot\,, \omega_j)\}_{j=1}^{M}$ such that the computational work, defined as M times the average number of time steps, i.e., $M \times E[\kappa]$ is minimal, constrained by the condition that the computational error, $\mathcal{E}$, defined by

$$\mathcal{E} = E[g(X(T))] - \tfrac{1}{M} \sum_{j=1}^{M} g(\overline{X}(T, \omega_j)), \tag{1.7}$$

is such that the event

$$\big|\mathcal{E}\big| \leq \text{TOL} \tag{1.8}$$

has a probability close to one. The computational error $\mathcal{E}$ naturally splits as the sum of the (deterministic) time discretization error $\mathcal{E}_T$ and the statistical error $\mathcal{E}_S$ given,

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

respectively, by

$$\mathcal{E}_T = E\big[g(X(T))\big] - E\big[g(\overline{X}(T))\big], \tag{1.9}$$

$$\mathcal{E}_S = E[g(\overline{X}(T))] - \tfrac{1}{M}\sum_{j=1}^{M} g(\overline{X}(T,\omega_j)). \tag{1.10}$$

The number M of independent realizations of $\overline{X}$ is computed from the statistical properties of $\mathcal{E}_S$. Theorems 2.2 and 3.1 state approximations for the time discretization error in terms of error contributions from each time subinterval, that is,

$$\mathcal{E}_T \approx E\left(\sum_m \rho(t_m,\omega)(\Delta t_m)^2\right)$$

(see (2.7) and (3.4)). These approximations are used to estimate the time discretization error in the adaptive algorithms that we propose. The error density $\rho(t_m,\omega_j)$ in (2.8) and (3.5), respectively, are computed through the discrete dual functions that we describe in section 1.4.

1.4. Discrete dual equations. This subsection is dedicated to the determination of the discrete dual functions $\varphi(t) \in \mathbb{R}^d$, $\varphi'(t) \in \mathbb{R}^{d\times d}$, and $\varphi''(t) \in \mathbb{R}^{d\times d\times d}$, where t is a node of the (stochastic) partition of the time interval $[0,T]$. First, we introduce some notation.

DEFINITION 1.1 (local discrete solution operator). *Write the Euler time stepping for the time nodes $t = t_0, t_1^-, t_1, t_2^-, t_2, \dots$ as*

$$\overline{X}(t_{next},\omega) = \Phi(t,\overline{X}(t)), \tag{1.11}$$

where the next computational time is given by

$$t_{next} = \begin{cases} t_n & \text{if} \quad t = t_n^-, \\ t_{n+1}^- & \text{if} \quad t = t_n, \end{cases}$$

and, correspondingly, the previous computational time is

$$t_{prev} = \begin{cases} t_{n-1} & \text{if} \quad t = t_n^-, \\ t_n^- & \text{if} \quad t = t_n. \end{cases}$$

Observe that, depending on ω and t, we may have

$$\Phi(t,x,\omega) = \begin{cases} x + a(t,x)\Delta t + b^\ell(t,x)\Delta W^\ell & \text{if } t = t_n, \\ x + c(t,x,Z_k(\omega)) & \text{if } t = t_n^- \text{ and } t_n \text{ is the } k\text{th jump time}, \\ x & \text{if } t = t_n^- \text{ and } t_n \text{ is not a jump time}. \end{cases}$$

Then, for each realization, φ, φ', and φ'' are constructed by the following algorithm.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

DUAL BACKWARD TIME STEPPING ALGORITHM.

Set the initial backward values

$$
\begin{aligned}
t &= T,\\
\varphi_i(T) &= \partial_i g\big(\overline{X}(T)\big),\\
\varphi'_{ik}(T) &= \partial_{ik} g\big(\overline{X}(T)\big),\\
\varphi''_{ikm}(T) &= \partial_{ikm} g\big(\overline{X}(T)\big).
\end{aligned} \tag{1.12}
$$

while $t > 0$ set $t = t_{prev}$

$$\varphi_i(t) = \partial_i \Phi_j(t, \overline{X}(t))\, \varphi_j(t_{next}), \tag{1.13}$$

$$\varphi'_{ik}(t) = \partial_i \Phi_j(t, \overline{X}(t))\, \partial_k \Phi_p(t, \overline{X}(t))\, \varphi'_{jp}(t_{next}) \tag{1.14}$$

$$+ \partial_{ik} \Phi_j(t, \overline{X}(t))\, \varphi_j(t_{next}), \tag{1.15}$$

$$
\begin{aligned}
\varphi''_{ikm}(t) = {} & \partial_i \Phi_j(t, \overline{X}(t)) \partial_k \Phi_p(t, \overline{X}(t)) \partial_m \Phi_r(t, \overline{X}(t)) \varphi''_{jpr}(t_{next})\\
& + \partial_{im} \Phi_j(t, \overline{X}(t)) \partial_k \Phi_p(t, \overline{X}(t)) \varphi'_{jp}(t_{next})\\
& + \partial_i \Phi_j(t, \overline{X}(t)) \partial_{km} \Phi_p(t, \overline{X}(t)) \varphi'_{jp}(t_{next})\\
& + \partial_{ik} \Phi_j(t, \overline{X}(t)) \partial_m \Phi_p(t, \overline{X}(t)) \varphi'_{jp}(t_{next})\\
& + \partial_{ikm} \Phi_j(t, \overline{X}(t)) \varphi_j(t_{next}).
\end{aligned} \tag{1.16}
$$

end-while

1.5. Adaptivity. After choosing the initial jump augmented grid, we propose two different ways of a posteriori refinement or time stepping in order to achieve the prescribed TOL: a first algorithm that refines simultaneously all the simulated trajectories, which we call *deterministic*, and a second one that refines the simulated trajectories one by one, which we call *stochastic*.

In the deterministic algorithm, the refinement is taken in order to achieve

$$(\Delta \tilde{t}_n)^2 |E[\rho(\tilde{t}_n, \omega)]| = \text{constant}.$$

Observe that we can only control the deterministic points $\tilde{t}_n$ and not the whole augmented grid. This is the basis of our adaptive algorithm in section 4.3.

In the stochastic algorithm the time steps are determined by the condition

$$(\Delta \tilde{t}_n)^2 |\rho(\tilde{t}_n, \omega)| = \text{constant},$$

where, in contrast to the previous algorithm, there is no expectation. This is achieved through a test performed at each interval of each realization, to decide whether to refine the given interval (see section 4.4). In this case, when a node is added, a Brownian bridge is used to obtain the value of both the Wiener process and the approximated process, $\overline{X}$, at this new intermediate time node.

Stochastic time steps are advantageous for problems with singularities at random times, but since their use entails more work per realization than the deterministic time steps they should be judiciously used. A natural application of stochastic time steps appears in the weak approximation of killed diffusions; see [15]. The optimal stochastic steps depend on the whole solution $\overline{X}(t)$, $0 < t < T$, and in particular the step $\Delta t(t)$ at time t depends also on $W(u)$ for $t < u$. In stochastic analysis the concept of *adaptedness* means, intuitively, that the values of the process at time t depend only on the events generated by the sources of randomness up to time t,

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

i.e., they do not depend on future events. In numerical analysis a method is said to be *adaptive* when the approximate solution is used to control the error, e.g., to determine the time steps. Our stochastic time stepping algorithm is in this sense adaptive and nonadapted, since the time steps $\Delta t(t)$ depend on values of $W(u)$ for $t < u$, i.e., on future values of the Wiener process. The backward evolutions (1.13)–(1.14) of the weight functions φ and φ' avoid solving for the two variables t, s present in $\frac{\partial \overline{X}(t)}{\partial x(s)}$, which appears in the forward t-evolution equation for $\frac{\partial \overline{X}(t)}{\partial x(s)}$ in the identity $\varphi_i(t_n) = \partial_j g(\overline{X}(T)) \frac{\partial \overline{X}_j(T)}{\partial x_i(t_n)}$; see (6.25). A solution with two variables s and t would require work of the order N^2 for each realization, instead of the corresponding work of the order N in Theorem 2.2.

The rest of the paper is organized as follows. Section 2 contains a first error expansion based on the dual functions and their first variations. Section 3 contains a second error expansion based on the duals and their first and second variations. Section 4 presents implementations of adaptive algorithms. Section 5 includes results from numerical experiments, and section 6 presents the proofs.

2. A first estimate of the time discretization error. In this section we present in Theorem 2.2 an error expansion in a posteriori computable form for the time discretization error $\mathcal{E}_T$ in (1.9) based on the first and second dual functions φ and φ'. All the proofs are given in section 6.

Our analysis uses the fact that the simulated process can be extended to the interval $[0, T]$. To do this we first define the piecewise constant coefficients

$$\begin{aligned} &\overline{a}(s; \overline{X}) \equiv a(t_n, \overline{X}(t_n)) \quad \forall\, s \in [t_n, t_{n+1}), \\ &\overline{b}^{\ell}(s; \overline{X}) \equiv b^{\ell}(t_n, \overline{X}(t_n)) \quad \forall\, s \in [t_n, t_{n+1}), \quad \ell = 1, \ldots, \ell_0, \end{aligned} \tag{2.1}$$

for $n = 0, \ldots, \nu - 1$, and then consider the stochastic differential equation

$$\begin{aligned} \overline{X}(t) = \overline{X}(0) &+ \int_0^t \overline{a}(s; \overline{X}) ds + \sum_{\ell=1}^{\ell_0} \int_0^t \overline{b}^{\ell}(s; \overline{X})\, dW^{\ell}(s) \\ &+ \int_0^t \int_{\mathbf{z}} c\big(s, \overline{X}(s^-), z\big)\, p(ds, dz) \quad \forall\, t \in [0, T]. \end{aligned} \tag{2.2}$$

A key point of our approach is the possibility of perfect simulation of the jump structure, avoiding the necessity of the consideration of an approximated jump coefficient $\bar{c}$. For simplicity we introduce the notation $\partial_k \equiv \frac{\partial}{\partial x_k}$, $\partial_{ki} \equiv \frac{\partial^2}{\partial x_k \partial x_i}, \ldots$, $\partial_t \equiv \frac{\partial}{\partial t}$ and use the summation convention, i.e., if the same subscript appears twice in a term, the term denotes the sum over the range of this subscript, e.g., $c_{ik}\partial_k b_j \equiv \sum_{k=1}^{d} c_{ik}\partial_k b_j$, and consequently

$$d_{ij} \equiv \frac{1}{2} \sum_{\ell=1}^{\ell_0} b_i^{\ell} b_j^{\ell} = \frac{1}{2} b_i^{\ell} b_j^{\ell}, \quad \overline{d}_{ij} \equiv \frac{1}{2} \sum_{\ell=1}^{\ell_0} \overline{b}_i^{\ell} \overline{b}_j^{\ell} = \frac{1}{2} \overline{b}_i^{\ell} \overline{b}_j^{\ell}.$$

For a derivative ∂_α the notation $|\alpha|$ is its order.

LEMMA 2.1. (a) *Assume that the following regularity conditions hold:*

(i) *The functions $a(t, x)$, $b(t, x)$, and $c(t, x, z)$ are continuous in (t, x) and are twice continuously differentiable with respect to x.*

(ii) *The partial derivatives of first and second order with respect to x of the functions a and b are uniformly bounded.*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

(iii) *The function* $c(t,x,z)$ *satisfies*

(2.3)
$$\int_0^T \int_{\mathbf{z}} \left(|c(t,x,z)| + |\nabla_x c(t,x,z)|^4 + |\nabla_x^2 c(t,x,z)|^2 \right) \lambda(t)\mu(t,dz)dt \le C,$$

where ∇_x *denotes the gradient with respect to* x*, and the constant* C *is independent of* x*.*

(iv) *The function* g *is twice continuously differentiable, and together with its partial derivatives of first and second order it is uniformly bounded.*

Then the cost to go *function, defined by*

$$u(t,x) = E\big[g(X(T)) \mid X(t) = x\big], \tag{2.4}$$

satisfies the Kolmogorov equation

(2.5)
$$\partial_t u(t,x) + a_k \partial_k u(t,x) + d_{kn}\partial_{kn} u(t,x) + \lambda(t) \int_{\mathbf{z}} [u(t,x+c(t,x,z)) - u(t,x)]\mu(t,dz) = 0,$$

with the boundary condition

$$u(T,\cdot) = g.$$

(b) *Furthermore, if the following regularity conditions are satisfied:*

(i) *the functions* $\partial_\beta a(t,\cdot)$*,* $\partial_\beta b(t,\cdot)$*, and* $\partial_\beta c(t,\cdot,z)$ *are bounded uniformly in* t *and* z *for* $1 \le |\beta| \le 8$*;*

(ii) *the jump intensity* $\lambda(t)$ *and the measure* $\mu(t,dz)$ *have continuous time derivatives; in addition, the functions* $a(\cdot,x)$*,* $b(\cdot,x)$*,* $c(\cdot,x,z)$ *have continuous and uniformly bounded first order time derivatives;*

(iii) $\int_0^T \int_{\mathbf{z}} |c(t,0,z)|^{2k}\mu(t,dz)dt < \infty$ *for all* $k \in \mathbb{N}$*;*

(iv) *the function* g *has spatial derivatives* $\partial_\beta g$*, with polynomial growth for* $|\beta| \le 8$*;*

then the function u *has continuous partial derivatives with respect to* x *up to the order* 8*, satisfying the following polynomial growth condition: for all* $i \in \{0,1,2\}$ *and* $\alpha \in \mathbb{N}^d$ *with* $i + |\alpha| \le 8$ *there exists* $p_{\alpha,i} \in \mathbb{N}$ *and* $C_{\alpha,i} > 0$ *such that*

$$\max_{0\le t\le T} \left| \partial_t^i \partial_\alpha u(t,x) \right| \le C_{\alpha,i} \left(1 + |x|^{p_{\alpha,i}} \right) \quad \forall\, x \in \mathbb{R}^d.$$

LEMMA 2.2. *Assume that the regularity conditions described in paragraph* (b) *of Lemma* 2.1 *hold. Then the time discretization error in* (1.9) *satisfies*

$$\begin{aligned} \mathcal{E}_T = &\int_0^T E\Big[\left(a_k(t,\overline{X}(t^-)) - \overline{a}_k(t;\overline{X}) \right) \partial_k u(t,\overline{X}(t^-)) \Big]\, dt \\ &+ \int_0^T E\Big[\left(d_{ij}(t,\overline{X}(t^-)) - \overline{d}_{ij}(t;\overline{X}) \right) \partial_{ij} u(t,\overline{X}(t^-)) \Big]\, dt, \end{aligned} \tag{2.6}$$

where $u = u(t,x)$ *is the cost to go function defined in* (2.4)*.*

Remark 2.1. It is possible to relax the assumption of the process departing from a fixed value $x_0 \in \mathbb{R}$. We can assume that it departs from a random variable $X(0)$, independent of the jump structure and the Wiener process, and we can assume that either this initial random variable admits perfect simulation by $\overline{X}(0)$ or it does not. Both situations easily adapt to our approach, giving (in the latter case) an additional term of the form $E[u(0,X(0)) - u(0,\overline{X}(0))]$ in the error expansion in (2.6).

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Lemma 2.2 is combined with stochastic flows to derive the a posteriori error expansion in Theorem 2.2 below. This error expansion is based on variations of the processes X and $\overline{X}$. The proof of Theorem 2.2 uses mainly the fact that the error in replacing $g(X(T))$ in Lemma 2.2 by $g(\overline{X}(T))$, in the representation (2.4) of $\partial_\alpha u$, yields the small deterministic remainder term $\int_0^T \mathcal{O}\big((\Delta t)^2\big)dt$ in (2.7) of Theorem 2.2, which is analogous to the $\mathcal{O}(N^{-2})$ term in Talay and Tubaro's expansion, [53], and needs some a priori estimate to be controlled. Lemma 2.2 can be applied to estimate this error. The second important ingredient in the proof is the Markov property of $\overline{X}$ satisfied at the discrete times t_n. Based on the fact that $\overline{X}(t_n)$ is $\mathcal{F}_{t_n}$ measurable, the nested expected values

$$E\big[a_j(t_n, \overline{X}(t_n))\partial_{x_j(t_n)} E[g(\overline{X}(T)) \mid \mathcal{F}_{t_n}]\big]$$

in (2.6) can, by the definition of φ and its implication (6.25), be decoupled to

$$E\big[a_j(t_n, \overline{X}(t_n))\varphi_j(t_n)\big],$$

which reduces the computational complexity substantially; see Lemma 6.3.

THEOREM 2.2 (first error expansion). *Suppose that a, b, g, X, and $\overline{X}$ satisfy the assumptions in Lemma* 2.2. *Then the time discretization error in* (1.9) *has the expansion*

(2.7)
$$\mathcal{E}_T = E\left[\sum_{m=0}^{N-1} \rho(\tilde{t}_m, \omega)(\Delta\tilde{t}_m)^2\right] + E\left[\sum_{m=0}^{N-1} (\Delta\tilde{t}_m)^2 \left\{\mathcal{O}(\Delta\tilde{t}_m) + \sum_{k=m}^{N-1} \mathcal{O}((\Delta\tilde{t}_k)^2)\right\}\right],$$

where the leading order error term is in computable a posteriori form:

(2.8)
$$\begin{aligned}\rho(\tilde{t}_m, \omega) \equiv{}& \frac{1}{2}\sum_{n\in\mathcal{J}_m} \big[\big(a_i(t_{n+1}, \overline{X}(t_{n+1}^-, \omega)) - a_i(t_n, \overline{X}(t_n, \omega))\big)\varphi_i(t_{n+1}^-, \omega)\big]\frac{\Delta t_n}{(\Delta\tilde{t}_m)^2}\\ &+ \frac{1}{2}\sum_{n\in\mathcal{J}_m} \big[\big(d_{ik}(t_{n+1}, \overline{X}(t_{n+1}^-, \omega)) - d_{ik}(t_n, \overline{X}(t_n, \omega))\big)\varphi'_{ik}(t_{n+1}^-, \omega)\big]\frac{\Delta t_n}{(\Delta\tilde{t}_m)^2}\end{aligned}$$

with $\mathcal{J}_m \equiv \{n : \tilde{t}_m \le t_n < \tilde{t}_{m+1}\}$, $m = 0, \dots, N-1$, and the discrete dual functions $\varphi(t_n) \in \mathbb{R}^d$ and $\varphi'(t_n) \in \mathbb{R}^{d\times d}$ defined in (1.13) *and* (1.14), *respectively.*

Remark 2.3. When implementing a deterministic time stepping algorithm based on this result the expectation of the sum of errors in (2.7) is approximated by the mean of the errors along the M simulated trajectories, i.e.,

$$E\left[\sum_{m=0}^{N-1} \rho(\tilde{t}_m, \omega)(\Delta\tilde{t}_m)^2\right] \approx \frac{1}{M}\sum_{j=1}^{M}\sum_{m=0}^{N-1} \rho(\tilde{t}_m, \omega_j)(\Delta\tilde{t}_m)^2.$$

The statistical error of this approximation can be expressed as

$$E\left[\sum_{m=0}^{N-1} \rho(\tilde{t}_m, \omega)(\Delta\tilde{t}_m)^2\right] - \frac{1}{M}\sum_{j=1}^{M}\sum_{m=0}^{N-1} \rho(\tilde{t}_m, \omega_j)(\Delta\tilde{t}_m)^2 = \int_0^T (I_M + J_M)dt,$$

where the distributions of the statistical errors $\sqrt{M}I_M$ and $\sqrt{M}J_M$ weakly converge to normal distributions with mean zero and time interval–dependent variances given

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

by

$$\mathrm{Var}\left[\sum_{n\in\mathcal{J}_m}\left(a_i(t_{n+1},\overline{X}(t_{n+1}^-))-a_i(t_n,\overline{X}(t_n))\right)\varphi_i(t_{n+1}^-)\right]=\mathcal{O}(\Delta\tilde{t}_m)$$

and

$$\mathrm{Var}\left[\sum_{n\in\mathcal{J}_m}\left(d_{ik}(t_{n+1},\overline{X}(t_{n+1}^-))-d_{ik}(t_n,\overline{X}(t_n))\right)\varphi'_{ik}(t_{n+1}^-)\right]=\mathcal{O}(\Delta\tilde{t}_m),$$

respectively. Observe that the number of realizations to determine a reliable error estimate is in general TOL^{-1}, much smaller than the—proportional to TOL^{-2}—number of realizations to approximate $E[g(X(T))]$. For more details on this and the statistical approximation of the error density ρ see Remark 2.7 in [51].

3. A second estimate of the time discretization error. In this section we present in Theorem 3.1 an error expansion in a posteriori computable form for the time discretization error $\mathcal{E}_T$ in (1.9) based on the dual functions and its first and second derivatives φ, φ', and φ''. The analysis uses the Malliavin derivative, $\partial_{W(t)}Y$, which is the first variation of a process Y with respect to a perturbation $dW(t)$, at time t of the Wiener process; cf. [46]. As we will see, although computationally more expensive than the expansion in Theorem 2.2, it gives a more precise estimation of the error, measured through its variance.

We shall restrict the analysis to time steps which are constructed by first sampling the jump times to augment an a priori given time discretization $\Delta\tilde{t}$, obtaining $\Delta t[0]$ (see (3.3)). Using the refinement criterion

$$|\rho(t,\omega)|\big(\Delta t(t)\big)^2<\text{constant},\tag{3.1}$$

we construct the partition $\Delta t(t)$ that satisfies

$$\Delta t(t)=\Delta t[0](t)/2^n\ \text{ for some natural number } n=n(t,\omega),$$

with an approximate error density function ρ, satisfying for $s,t\in[0,T]$ and all outcomes ω the uniform upper and lower bounds

$$\begin{aligned}\rho_{low}(\mathrm{TOL})\le|\rho(s,\omega)|&\le\rho_{up}(\mathrm{TOL}),\\ |\partial_{W(t)}\rho(s,\omega)|&\le\rho_{up}(\mathrm{TOL}).\end{aligned}\tag{3.2}$$

Here ρ_{low} and ρ_{up} are positive functions with $\mathrm{TOL}/\rho_{low}(\mathrm{TOL})\to 0$ as $\mathrm{TOL}\to 0$. For each realization successive subdivisions of the steps yield the largest time steps satisfying (3.1). The corresponding stochastic increments ΔW will have the correct distribution, with the necessary independence, if the increments ΔW related to the new steps are generated by Brownian bridges [30], i.e., the time steps are generated by conditional expected values of the Wiener process.

Let δ be a constant approximating $\frac{\mathrm{TOL}}{E[N]}$, where $E[N]$ is the expected number of steps. The analysis in this section with adaptive nonadapted time steps, satisfying (3.1)–(3.2), is based on the *stochastic time stepping algorithm* described below.

In what follows, Lemma 3.1 and Theorem 3.1 show that although the steps generated by (3.1)–(3.2) are not adapted, the method indeed converges to the correct limit, which is the same as the limit of the forward Euler method with adapted time steps.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

LEMMA 3.1 (strong convergence). *Suppose that a, b, c, g, X satisfy the assumptions in Lemma 2.2, that $\overline{X}$ is constructed by the forward Euler method, based on the stochastic time stepping algorithm above, with step sizes Δt_n satisfying (3.1)–(3.2), and that their corresponding ΔW_n are generated by Brownian bridges. Then*

$$\sup_{0\le t\le T}\sqrt{E[|X(t)-\overline{X}(t)|^2]} = \mathcal{O}\left(\sqrt{\Delta t_{\sup}}\right) = \mathcal{O}\left(\sqrt{\frac{\mathrm{TOL}}{\rho_{low}(\mathrm{TOL})}}\right) \to 0$$

as TOL $\to 0$, *where* $\Delta t_{\sup} \equiv \sup_{n,\omega} \Delta t_n(\omega)$.

STOCHASTIC TIME STEPPING ALGORITHM.

Do for M realizations ω_j, $j = 1, \dots, M$:

Sample jump times: $0 < \tau_1(\omega_j) < \cdots < \tau_{\nu(\omega_j)}(\omega_j) < T$.

Consider the initial augmented partition given by the union

$$\Delta t[0] = \{t_n(\omega_j)\}_{n=0}^{\kappa(\omega_j)} = \{\tilde{t}_n\}_{n=0}^{N} \cup \{\tau_n(\omega_j)\}_{n=0}^{\nu(\omega_j)} \tag{3.3}$$

that has $\kappa(\omega_j) = N + \nu(\omega_j)$ (a.s.) time steps.

`STEP 1`: Set $k = 0$. Start with the initial coarse mesh $\Delta t[0]$ and compute $\Delta W[0]$.

`STEP 2`: For the piecewise constant mesh function $\Delta t[k]$ with corresponding noise $\Delta W[k]$, compute $\overline{X}[k]$ and the weight function $\rho[k]$ defined in (3.4)–(3.5).

`STEP 3`: Define $r(t) \equiv |\rho[k](t)|(\Delta t[k](t))^2$ and let for all t

$$\Delta t[k+1](t) = \begin{cases} \Delta t[k](t) & \text{if } r(t) < \delta, \quad (\dagger) \\ \Delta t[k](t)/2 & \text{if } r(t) \ge \delta, \quad (\star) \end{cases}$$

and in the refinement case $(\star)$ construct $\Delta W[k+1]$ by Brownian bridges based on the already known $\Delta W[k]$.

`STEP 4`: **If** at least one step of $\Delta t[k]$ is refined by $(\star)$, increment k by 1 and **goto** Step 2. **Else** all steps of $\Delta t[k]$ satisfy $(\dagger)$; thus accept the approximation $g(\overline{X}(T, \omega_j))$ and **goto** the next realization of p and W.

Enddo

If the statistical error, $E[g(\overline{X}(T))] - \frac{1}{M}\sum_{j=1}^{M} g(\overline{X}(T, \omega_j))$, is sufficiently small **stop**, **else** restart with a larger M.

Endif

For a detailed description of the stochastic time stepping algorithm see section 4.4.

Proof of Lemma 3.1. The proof relies on the two-step simulation scheme that we construct and on the fact that the jump nodes do not contribute to the global error. Observe that $\mathcal{F}^J$ is the σ-algebra generated by the jumps and marks constructed in section 1.2.1. Consider the conditional expectation $E[|X(t) - \overline{X}(t)|^2 \mid \mathcal{F}^J]$ and apply Lemma 3.1 from [51], using also that there is no time discretization error at the jump nodes. ∎

In addition to the dual functions φ and φ' in Theorem 2.2, the new error expansion for stochastic time steps in Theorem 3.1 below also uses, for $t = t_{n+1}^-$, the discrete dual variation $\varphi''(t)$ defined in (1.16). It derives an error estimate applicable both to adaptive deterministic time steps and to the stochastic time stepping algorithm; the assumptions and the proof of the theorem focus on stochastic steps; however, a modification to deterministic time steps is straightforward. The computable error density $\tilde{\rho}$ of this error estimate can then be cut off for small and large values to satisfy (3.2). This truncation procedure is explained in formulas (4.6) and (4.15).

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

THEOREM 3.1 (second error expansion). *Suppose that the functions a, b, c, g, and the process X satisfy the assumptions in Lemma 2.2 with $X(0)$ deterministic. Let $\overline{X}$ be constructed by the forward Euler method with step sizes Δt_n satisfying (3.1)–(3.2) and let the corresponding ΔW_n be generated by Brownian bridges, following the stochastic time stepping algorithm in Lemma 3.1. Then the time discretization error defined in (1.9) has the following expansion, based on both the drift and diffusion fluxes and the discrete dual functions φ, φ', and φ'' given in (1.13)–(1.16), with computable leading order terms:*

(3.4)
$$\mathcal{E}_T = E\left[\sum_{n=0}^{\kappa-1} \tilde{\rho}(t_n,\omega)(\Delta t_n)^2\right] + \mathcal{O}\left(\Big(\frac{\mathrm{TOL}}{\rho_{low}(\mathrm{TOL})}\Big)^{1/2}\Big(\frac{\rho_{up}(\mathrm{TOL})}{\rho_{low}(\mathrm{TOL})}\Big)^{\epsilon}\right) E\left[\sum_{n=0}^{\kappa-1}(\Delta t_n)^2\right],$$

for any $\epsilon > 0$ and where

$$\begin{aligned}\tilde{\rho}(t_n,\omega) \equiv \frac{1}{2}\Big(&\big(\partial_t a_k + \partial_j a_k a_j + \partial_{ij} a_k d_{ij}\big)\varphi_k(t_{n+1}^-)\\ &+ \big(\partial_t d_{km} + \partial_j d_{km} a_j + \partial_{ij} d_{km} d_{ij} + 2\partial_j a_k d_{jm}\big)\varphi'_{km}(t_{n+1}^-)\\ &+ \big(2\partial_j d_{km} d_{jr}\big)\varphi''_{kmr}(t_{n+1}^-)\Big)\end{aligned} \tag{3.5}$$

and the terms in the sum of (3.5) are evaluated at the a posteriori known points $(t_n, \overline{X}(t_n))$, i.e.,

$$\partial_\alpha a \equiv \partial_\alpha a(t_n, \overline{X}(t_n)), \quad \partial_\alpha b \equiv \partial_\alpha b(t_n, \overline{X}(t_n)), \quad \partial_\alpha d \equiv \partial_\alpha d(t_n, \overline{X}(t_n)).$$

Observe that the constant in $\mathcal{O}$ that appears in (3.4) may not be uniform with respect to the value ϵ. Thus, in practice one chooses $\epsilon = \epsilon(\mathrm{TOL})$ to minimize the contribution of the remainder term to the error expansion (3.4).

4. Adaptive time stepping algorithms. Here, we describe two adaptive time stepping algorithms for the weak approximation problem based on the approximation error formula (3.4). They are very similar to those introduced in [45]. Algorithm D augments a mesh that is the same for all realizations with jump times that are sampled differently for each realization; its adaptive strategy is based on averaged information from the a posteriori error formula. On the other hand, Algorithm S can adapt the mesh differently for each realization. Both adaptive algorithms choose adaptively the number of realizations and the size of time steps to efficiently bound the approximation error by a prescribed error tolerance.

4.1. Computational error splitting. As we described in section 1.3 the weak approximation computational error $\mathcal{E}$ of the jump augmented Euler–Maruyama scheme naturally splits into time discretization and statistical errors, namely, $\mathcal{E} = \mathcal{E}_T + \mathcal{E}_S$. Thus, the control of the computational error is related to the combined control of the time discretization error via the choice of the time steps Δt, and of the statistical error via the choice of the number M of realizations. Therefore, we split a given computational error tolerance, $\mathrm{TOL} > 0$, into a statistical error tolerance TOL_S and a time discretization error tolerance TOL_T (see [51], [45]) by

$$\mathrm{TOL}_T = \frac{1}{3}\,\mathrm{TOL} \quad \text{and} \quad \mathrm{TOL}_S = \frac{2}{3}\mathrm{TOL}. \tag{4.1}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

4.2. Control of the statistical error. For M independent samples $\{Y(\omega_j)\}_{j=1}^M$ of a random variable Y, with $E\left[\,|Y|^6\,\right] < \infty$, define the sample average by

$$\mathcal{A}(Y;M) \equiv \frac{1}{M}\sum_{j=1}^{M} Y(\omega_j)$$

and the sample standard deviation by

$$\mathcal{S}(Y;M) \equiv \left\{\mathcal{A}(Y^2;M) - (\mathcal{A}(Y;M))^2\right\}^{\frac{1}{2}}.$$

Let $\sigma_Y \equiv \left\{E\left[\left|Y - E[Y]\right|^2\right]\right\}^{\frac{1}{2}}$ and $Z_M \equiv \frac{\sqrt{M}}{\sigma_Y}\left(\mathcal{A}(Y;M) - E[Y]\right)$ with cumulative distribution function $F_{Z_M}(x) \equiv P(Z_M \le x),\ x \in \mathbb{R}$. Let $\gamma \equiv (E[|Y - E[Y]|^3])^{1/3}/\sigma_Y < \infty$. The Berry–Essen theorem [14] gives the following estimation in the central limit theorem:

$$\sup_{x\in\mathbb{R}} |F_{Z_M}(x) - \mathcal{N}(x)| \le \frac{3}{\sqrt{M}}\gamma^3$$

for the rate of convergence of F_{Z_M} to the distribution function $\mathcal{N}$ of a normal random variable with mean zero and variance one, i.e.,

$$\mathcal{N}(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}s^2\right) ds. \tag{4.2}$$

Provided M is sufficiently large, i.e., $M \gg 36\gamma^6$, the statistical error

$$\mathcal{E}_S(Y;M) \equiv E[Y] - \mathcal{A}(Y;M)$$

satisfies, by the Berry–Essen theorem, the probability approximation

$$P\left(\left[|\mathcal{E}_S(Y;M)| \le c_0\frac{\sigma_Y}{\sqrt{M}}\right]\right) \approx 2\mathcal{N}(c_0) - 1.$$

In practice choose some constant $c_0 \ge 1.65$, so the normal distribution satisfies $1 > 2\mathcal{N}(c_0) - 1 \ge 0.901$ and the event

$$|\mathcal{E}_S(Y;M)| \le \mathtt{E}_S(Y;M) \equiv c_0\ \frac{\mathcal{S}(Y;M)}{\sqrt{M}} \tag{4.3}$$

has probability close to one, which involves the additional step to approximate σ_Y by $\mathcal{S}(Y;M)$; cf. [18]. Thus, in the computations $\mathtt{E}_S(Y;M)$ is a good approximation of the statistical error $\mathcal{E}_S(Y;M)$.

For a given $\mathrm{TOL}_S > 0$, the goal is to find M such that $\mathtt{E}_S(Y;M) \le \mathrm{TOL}_S$. The following algorithm adaptively finds the number of realizations M to compute the sample average $\mathcal{A}(Y;M)$ as an approximation to $E[Y]$. With probability close to one, depending on c_0, the statistical error in the approximation is then bounded by TOL_S.

`routine Monte-Carlo`(TOL_S, M_0; EY)
Set the batch counter $m = 1$, $M[m] = M_0$ and $\mathtt{E}_S[m] = +\infty$.
Do while ($\mathtt{E}_S[m] > \mathrm{TOL}_S$)

(4.4)
Compute $M[m]$ new samples of Y, along with the sample average $EY \equiv \mathcal{A}(Y;M[m])$, the sample standard deviation $\mathcal{S}[m] \equiv \mathcal{S}(Y;M[m])$ and the statistical error estimation $\mathtt{E}_S[m+1] \equiv \mathtt{E}_S(Y,M[m])$. Compute $M[m+1]$ by `change_`M ($M[m]$, $\mathcal{S}[m]$, TOL_S; $M[m+1]$).
Increase m by 1.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

end-do
`end of Monte-Carlo`

`routine change_`M $(M_{in}, \mathcal{S}_{in}, \mathrm{TOL}_S; M_{out})$

$$\begin{aligned} M^* &= \min\Big\{\text{integer part } \Big(\tfrac{c_0\,\mathcal{S}_{in}}{\mathrm{TOL}_S}\Big)^2,\ \mathrm{MCH}\times M_{in}\Big\},\\ n &= \text{integer part } (\log_2 M^*) + 1,\\ M_{out} &= 2^n. \end{aligned} \tag{4.5}$$

`end of change_`M

Here, M_0 is a given initial value for M, and $\mathrm{MCH} > 1$ is a positive integer parameter introduced to avoid a large new number of realizations in the next batch due to a possibly inaccurate sample standard deviation $\mathcal{S}[m]$. Indeed, $M[m+1]$ cannot be greater than $\mathrm{MCH}\times M[m]$.

4.3. Deterministic time stepping algorithm. Following closely [40] and [45], we present an adaptive algorithm based on the cut-off of the error density $\tilde{\rho}$ in (3.5) of Theorem 3.1 which is defined as

$$\rho_D^n \equiv \min\left(\max\left(\left|\frac{1}{(\Delta\tilde{t}_n)^2}\sum_{\ell\in\mathcal{J}_n}(\Delta t_\ell)^2\tilde{\rho}(t_\ell,\overline{X})\right|, \mathrm{TOL}^{\frac{1}{9}}\right), \mathrm{TOL}^{-1}\right), \quad n=1,\dots,N. \tag{4.6}$$

The error expansion in Theorem 3.1 motivates us to approximate the time discretization error by

$$|\mathcal{E}_T| \lesssim E\left[\sum_{n=1}^{N} r_n\right], \tag{4.7}$$

where the error indicator, r_n, is defined by

$$r_n \equiv \rho_D^n\,(\Delta\tilde{t}_n)^2, \quad n=1,\dots,N. \tag{4.8}$$

The main advantage of the deterministic time stepping algorithm over the stochastic time stepping algorithm is that the number M_T of realizations necessary to determine the optimal deterministic stepping scheme is considerably smaller than M, whereas, in the stochastic time stepping algorithm, a refinement of the partition is carried out in each one of the M trajectories, leading to a considerably larger amount of computational work.

As pointed out before, the error expansion derived in Theorem 3.1 is also valid with deterministic time steps. Therefore, we have some flexibility in the choice of error densities for the adaptive algorithm with deterministic time steps because we can also use the results from Theorem 2.2. There are some practical differences to mention. On the one hand, the variance of the averaged error density from (2.8) is $\mathcal{O}(\frac{1}{\Delta\tilde{t}_n M_T})$. This feature has been observed in [42] and [51], where a local filtering procedure was proposed to reduce the variance of the error density estimator. A positive feature of this error density is that it does not require the computation of the second variation, φ'', which may be computationally expensive for large d. On the other hand, the averaged error density from (3.5) has a much smaller variance $\mathcal{O}(\frac{1}{M_T})$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

which does not need filtering but requires the computation of φ''. In this work, based on the comments above and on previous experience (mainly [51]), we do not discuss further this choice and only show numerical results with adaptive deterministic time steps based on the error density (3.5).

The approximation of the time discretization error in the right-hand side of (4.7) can be separated into two parts,

$$E\left[\sum_{n=1}^{N} r_n\right] \leq \mathcal{A}\left(\sum_{n=1}^{N} r_n; M_T\right) + \left|E\left[\sum_{n=1}^{N} r_n\right] - \mathcal{A}\left(\sum_{n=1}^{N} r_n; M_T\right)\right|, \tag{4.9}$$

where the second error term in the right-hand side of (4.9) is with probability close to one asymptotically bounded by

$$\left|E\left[\sum_{n=1}^{N} r_n\right] - \mathcal{A}\left(\sum_{n=1}^{N} r_n; M_T\right)\right| \lesssim \mathtt{E}_{TS} \equiv c_0 \frac{\mathcal{S}\left(\sum_{n=1}^{N} r_n; M_T\right)}{\sqrt{M_T}} \tag{4.10}$$

and the first term defines $\mathtt{E}_{TT} \equiv \mathcal{A}(\sum_{n=1}^{N} r_n; M_T)$. Then for a given $\mathrm{TOL}_T > 0$, the goal is to construct a partition $\Delta\tilde{t}$ of $[0,T]$ with as few time steps and realizations M_T as possible, such that $\mathtt{E}_{TT} + \mathtt{E}_{TS} \leq \mathrm{TOL}_T$. To this end, first split the time discretization tolerance TOL_T into two positive parts, TOL_{TT} and TOL_{TS} for $\mathtt{E}_{TT}$ and $\mathtt{E}_{TS}$, respectively. The statistical error of the time discretization using the density (4.6) is $\mathcal{O}(\frac{\Delta t_{\sup}}{\sqrt{M_T}})$. Therefore the percentage of the tolerance, TOL, devoted to the control of the statistical time discretization error can be arbitrary small as $\Delta t_{\sup} \to 0$. In practice we choose

$$\mathrm{TOL}_{TT} = \frac{2}{3}\mathrm{TOL}_T = \frac{2}{9}\mathrm{TOL}, \qquad \mathrm{TOL}_{TS} = \frac{1}{3}\mathrm{TOL}_T = \frac{1}{9}\mathrm{TOL}. \tag{4.11}$$

The control of the statistical time discretization error determines the number of realizations M_T necessary to ensure a reliable choice of the time discretization in the deterministic time stepping algorithm. Similarly as in [45], here it is optimal to equidistribute the error contributions from different time intervals. Thus, the goal of the adaptive algorithm described below is to construct a deterministic time partition $\Delta\tilde{t}$ of $[0,T]$ such that

$$\bar{r}_n \equiv \mathcal{A}(r_n; M_T) \leq d_1 \frac{\mathrm{TOL}_{TT}}{N}, \quad n = 1, \ldots, N, \tag{4.12}$$

where $d_1 = 2$; see [45, Remark 3.9].

To achieve (4.12), start with an initial partition $\Delta\tilde{t}[1]$ and then specify iteratively a new partition $\Delta\tilde{t}[k+1]$, from $\Delta\tilde{t}[k]$, using the refinement strategy

$$\begin{aligned}
&\textbf{for} \;\; n = 1, 2, \ldots, N[k] \\
&\textbf{if } \bar{r}_n[k] \geq d_1 \tfrac{\mathrm{TOL}_{TT}}{N[k]}, \;\; \textbf{then} \text{ divide } \Delta\tilde{t}_n[k] \text{ into 2 uniform substeps.} \\
&\textbf{else} \text{ let the new step be the same as the old} \\
&\textbf{endif} \\
&\textbf{endfor}
\end{aligned} \tag{4.13}$$

until the following stopping criteria is satisfied:

$$\textbf{if } \left(\max_{1 \leq n \leq N[k]} \bar{r}_n[k] < D_1 \frac{\mathrm{TOL}_{TT}}{N[k]}\right) \;\; \textbf{then} \text{ stop.} \tag{4.14}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Here D_1 is a given constant satisfying $D_1 > \frac{2}{\hat{c}} d_1$, where $\hat{c} \approx 1/2$; see [45]. The combination of (4.9) and (4.14) asymptotically guarantees a given level of accuracy, $\mathtt{E}_{TT} < D_1 \mathrm{TOL}_{TT}$. The positive number D_1 is introduced to avoid slow convergence in case almost all $\bar{r}_n$ satisfy (4.14), as in section 4.4.

Now we are ready for the detailed definition of the adaptive algorithm with deterministic steps.

ALGORITHM D.

Initialization Choose:

1. an error tolerance, $\mathrm{TOL} \equiv \mathrm{TOL}_S + \mathrm{TOL}_{TT} + \mathrm{TOL}_{TS}$,
2. a number $N[1]$ of initial uniform steps $\Delta\tilde{t}[1]$ for $[0, T]$,
3. a number $M[1]$ of initial realizations and set $M_T[1] = M[1]$,
4. a number $d_1 = 2$ in (4.13) and $\hat{c} = 1/2 + 1/20$ to compute D_1 using $D_1 > \frac{2}{\hat{c}} d_1$,
5. a constant $c_0 \geq 1.65$ and an integer MCH≥ 2 to determine the number of realizations in (4.5), and
6. a constant $\epsilon = 1/2$ used to control the statistical error in the time discretization error estimate.

Set the iteration counter, k, for time refinement levels, to 1 and set the statistical error, $\mathtt{E}_{TS} = +\infty$ and $\bar{r}[k] = +\infty$.

Do while ($\bar{r}[k]$ violates the stopping criterion (4.14) or $\mathtt{E}_{TS} > \mathrm{TOL}_{TS}$)

Compute the sample averages and the error estimates on $\Delta\tilde{t}[k]$
by calling `Euler`. Set $M_T[k+1] = M_T[k]$ and $\Delta\tilde{t}[k+1] = \Delta\tilde{t}[k]$.

Set $\widetilde{\mathrm{TOL}}_{TS,\epsilon} = \max(\mathrm{TOL}_{TS}, \epsilon\, \mathtt{E}_{TT})$.

If $(\mathtt{E}_{TS} > \widetilde{\mathrm{TOL}}_{TS,\epsilon})$

Update $M_T[k+1]$ by `change_M` $(M_T[k], \mathcal{S}_{TS}[k], \widetilde{\mathrm{TOL}}_{TS,\epsilon}; M_T[k+1])$.

elseif ($\bar{r}[k]$ violates the stopping criterion (4.14))

For all time steps $i = 1, \ldots, N[k]$, do the refinement process (4.13)
to update $\Delta\tilde{t}[k+1]$ from $\Delta\tilde{t}[k]$.

end-if

Increase k by 1.

end-do

Compute an approximation, Eg, for $E[g(\overline{X}(T))]$ with fixed time mesh $\Delta\tilde{t} = \Delta\tilde{t}[k]$ by `Monte-Carlo`$(\mathrm{TOL}_S,\ M_T[k];\ Eg)$ in (4.4).

Accept Eg as an approximation of $E[g(X(T))]$, since the estimate of the computational error is bounded by TOL.

`routine Euler`

For each $M_{_T}[k]$ new realizations, sample jump times with their corresponding marks.

As described in section 1.2, use $\Delta\tilde{t}[k]$ to compute corresponding realizations of the Euler method.

Update the approximations of the time discretization error indicators $\bar{r}[k]$ and the statistical time discretization error $\mathtt{E}_{TS}[k]$ and compute the sample standard deviation $\mathcal{S}_{TS}[k] \equiv \mathcal{S}(g(\overline{X}(T)); M_T[k])$.

`end of Euler`

4.4. The stochastic time stepping algorithm. Now we describe an adaptive algorithm with stochastic time steps based on a cut-off of the error density $\tilde{\rho}$ introduced in (3.5) of Theorem 3.1, ρ_s, defined as

$$\rho_s^n \equiv \min\left(\max\left(|\tilde{\rho}(t_n, \overline{X})|, \mathrm{TOL}^{\frac{1}{9}}\right), \mathrm{TOL}^{-1}\right), \quad n = 1, \ldots, \kappa. \tag{4.15}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Following the error expansion in Theorem 3.1, the time discretization error is approximated by

$$|\mathcal{E}_T| \lesssim E\left[\sum_{n=1}^{\kappa} r_n\right], \tag{4.16}$$

where the error indicator, r_n, is defined by

$$r_n \equiv \rho_S^n \left(\Delta t_n\right)^2, \quad n = 1, \ldots, \kappa. \tag{4.17}$$

In this case it is optimal (cf. [45]) to equidistribute the error contributions among all time steps and all realizations. In other words, the goal of the adaptive algorithm is to construct a time partition Δt of $[0, T]$ for each realization such that

$$r_n \leq s_1 \frac{\mathrm{TOL}_T}{E[\kappa]}, \quad n = 1, \ldots, \kappa, \tag{4.18}$$

where $s_1 = 2$; see Remark 3.1 in [45]. Note that in practice the quantity $E[\kappa]$ is not known and we can only estimate it by a sample average $\mathcal{A}(\kappa; M)$ from the previous batch of realizations. The statistical error $|E[\kappa] - \mathcal{A}(\kappa; M)|$ is then bounded by $\mathtt{E}_S(\kappa; M)$, with probability close to one, by the same argument as in (4.3).

Let $\bar{\kappa}[j] \equiv \mathcal{A}(\kappa; M[j])$ be the sample average of the final number of time steps in the jth batch of $M[j]$ realizations. To achieve (4.18) for each realization, start with an initial partition $\Delta t[1]$ and then specify iteratively a new partition $\Delta t[k+1]$, from $\Delta t[k]$, using the following refinement strategy:

for each realization in the jth batch
for each time step $n = 1, \ldots, \kappa[k]$
if $r_n[k] \geq s_1 \frac{\mathrm{TOL}_T}{\bar{\kappa}[j-1]}$, **then** divide $\Delta t_n[k]$ into 2 uniform substeps.
else let the new step be the same as the old
endif
endfor
endfor (4.19)

The refinement strategy (4.19) motivates the following stopping criteria: for each realization of the jth batch

$$\textbf{if } \left(\max_{1 \leq n \leq \kappa[k]} r_n[k] < S_1 \frac{\mathrm{TOL}_T}{\bar{\kappa}[j-1]}\right) \quad \textbf{then} \quad \text{stop}, \tag{4.20}$$

where $S_1 > \frac{2}{\hat{c}}\, s_1$ with $\hat{c} \approx \frac{1}{2}$; see [45].

Now we are ready for the detailed definition of the adaptive algorithm with stochastic steps.

ALGORITHM S.

Initialization Choose:

1. an error tolerance, $\mathrm{TOL} \equiv \mathrm{TOL}_S + \mathrm{TOL}_T$,
2. a number $N[1]$ of initial uniform steps $\Delta t[1]$ for $[0, T]$ and set $\bar{\kappa} = N[1]$,
3. a number $M[1]$ of initial realizations,
4. a number $s_1 = 2$ in (4.19) and $\hat{c} = \frac{1}{2} + \frac{1}{20}$ to compute S_1 using $S_1 > \frac{2}{\hat{c}}\, s_1$, and

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

5. a constant $c_0 \geq 1.65$ and an integer MCH≥ 2 to determine the number of realizations in (4.5).

Set the iteration counter for batches $m = 1$ and the stochastic error $\mathtt{E}_S[m] = +\infty$.
Do while ($\mathtt{E}_S[m] > \mathrm{TOL}_S$)
  For realizations $j = 1, \ldots, M[m]$
    Set $k = 1$ and $r[k] = +\infty$.
    Generate the jump times and their marks $(\tau, Z) = \{(\tau_\ell, Z_\ell)\}_{\ell=1}^{\nu}$.
    Start with the initial partition $\Delta t[k]$ and generate $\Delta W[k]$.
    Compute $g(\overline{X}(T))[J]$ and $N[J]$ by `routine Control-Time-Error`.
  end-for
  Compute the sample average $Eg \equiv \mathcal{A}\left(g(\overline{X}(T)); M[m]\right)$, the sample standard deviation $\mathcal{S}[m] \equiv \mathcal{S}(g(\overline{X}(T)); M[m])$ and the a posteriori bound for the statistical error $\mathtt{E}_S[m] \equiv \mathtt{E}_S(g(\overline{X}(T)), M[m])$ in (4.3).
  if ($\mathtt{E}_S[m] > \mathrm{TOL}_S$)
    Compute $M[m+1]$ by `change_`$M(M[m], \mathcal{S}[m], \mathrm{TOL}_S; M[m+1])$, (see (4.5)) and update $\bar{\kappa} = \mathcal{A}(\kappa[J]; M[m])$, where the random variable $\kappa[J]$ is the final number of time steps on each realization.
  end-if
  Increase m by 1.
end-do
Accept Eg as an approximation of $E[g(X(T))]$, since the estimate of the computational error is bounded by TOL.

`routine Control-Time-Error`$(\Delta t[k], \Delta W[k], r[k], (\tau, Z); g(\overline{X}(T))[J], N[J])$
  Do while ($r[k]$ violates the stopping criterion (4.20))
    Compute the Euler approximation $\overline{X}[k]$ in section 1.2 and the error indicator $r[k]$ in (4.8) using the error density (4.15) on $\Delta t[k]$ with the known Wiener increments $\Delta W[k]$.
    If ($r[k]$ violates the stopping criterion (4.20))
      For time steps $i = 1, \ldots, N[k]$
        Do the refinement process (4.19) to compute $\Delta t[k+1]$ from $\Delta t[k]$ and compute $\Delta W[k+1]$ from $\Delta W[k]$ using Brownian bridges.
      end-for
    end-if
    Increase k by 1.
  end-do
  Set the number of the final level $J = k - 1$.
`end of Control-Time-Error`

Remark 4.1 (computational limitations of the jump augmented Euler scheme). Since the Euler method has weak order one, the number of discretization points needed to control the time discretization error is $\mathcal{O}(1/\mathrm{TOL})$. Therefore, as long as the product $\Lambda(T)\mathrm{TOL} \ll 1$, we can still use the jump augmented Euler method to solve the weak approximation problem with an amount of computational work that is similar to that required to solve the same problem without jumps. We still find it important to emphasize that our error approximation theory does not rely on the assumption $\Lambda(T)\mathrm{TOL} \ll 1$; the key point is that it may become too costly to apply it if this condition does not hold. Indeed, if $\Lambda(T)\mathrm{TOL} \approx 1$ (or larger), one can suggest (for overall computational efficiency reasons) using another time discretization which does not include jump times in the computational grid, for instance, one related to those discussed in [22] and [33].

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

5. Numerical experiments. This section shows numerical results from the implementation of the a posteriori error approximation formula presented in section 2 and of the adaptive algorithms described in section 4. Our implementation uses double-precision FORTRAN 77 and is an extension of the code written for the numerical experiments in [51]. For the numerical simulation of the uniform distribution $\mathcal{U}(0,1)$ and the normal distribution $\mathcal{N}(0,1)$, we implemented a double-precision modification of the functions `ran1` and `gasdev` proposed in [48], with a user-provided initial seed which must be a negative integer. In particular, we use three different seeds for the independent sources of randomness in our equations, namely, `iseed` for the simulation of the Wiener process increments, `zseed` for the simulation of the jump marks, and `tseed` for the simulation of the jump times.

Example 1. Consider the stochastic differential equation (1.1) with $d = 2$, $\ell_0 = \ell_1 = 1$,

$$a(t,x) = \left(-x_2,\, x_1 + \frac{1}{2}\,\lambda(t)\,x_2 \right), \quad b^1(t,x) = \left(\sqrt{\frac{\lambda(t)}{1+t}}\,\sin(x_1),\, 0 \right),$$
$$c(t,x,z) = \left(0,\, z\,\frac{\cos(x_1)}{\sqrt{t+1}} - x_2 \right)$$

and a time-dependent intensity $\lambda(t) = (1+t)^{-1}$. The distribution for the jump marks is time dependent and such that $E[Z_k^2] = 1$. In particular, we use $Z_k = \cos(2\pi\tau_k) + \sin(2\pi\tau_k)\,2\sqrt{3}\,(U_k - \frac{1}{2})$, where $\{\tau_k\}$ are the jumps constructed in section 1.2.1 and $\{U_k\}$ is a sequence of $\mathcal{U}(0,1)$ i.i.d. random variables. In this case, the inverse function Λ^{-1} is given explicitly by $\Lambda^{-1}(s) = \exp(s) - 1$. This example is a generalization of Example 5.1 in [33]. Here, we have both time-dependent jump intensity and mark distribution for the underlying Poisson process. Taking $g(x) = |x|^2$, $T = 1$, and $X(0) = (0,0)$, the exact solution of the corresponding weak approximation problem is given by the formula

$$E\big[g(X(T))\big] = |X(0)|^2 + \int_0^T \frac{\lambda(s)}{1+s}\,ds = 1 - \frac{1}{1+T} = \frac{1}{2}.$$

The value of the parameters needed in the simulations are MCH $= 10$, $c_0 = 1.65$, `iseed` $= -7$, `zseed` $= -101$, and `tseed` $= -20$. We note that the expected value of the number of jump points equals $\Lambda(T) = \Lambda(1) = \log(2) \approx 0.693$. Therefore, the computational cost of including the jump times of the process X in our discretization is fairly low; see the statistic for the sampled number of jumps, ν. In fact, the extra work to simulate the exact jump times of the process is asymptotically negligible as the required accuracy, TOL, tends to zero.

Deterministic time stepping algorithm. First, we perform overkilling runs in order to test how realistic the a posteriori error approximation of the time discretization error described in Theorem 2.2 is. The results, shown in Table 5.1, confirm that the ratio of the computational error and its computable approximation tends to 1 as the number of uniform time steps N increases. For each value of N, we choose the number of realizations M large enough in order to keep the total statistical error at the level of 1% of the size of the obtained approximation of the time discretization error. Then Table 5.2 contains the results of the Algorithm D with an adaptive choice of both the deterministic time steps and the number of realizations. The program starts with $M = 100$ and $N = 5$ subintervals as an initial uniform partition of the time interval $[0,T] = [0,1]$. The tolerance TOL $= 0.02$ is divided into $\text{TOL}_S = 0.01333$,

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

TABLE 5.1
Computing the efficiency index of the Algorithm D in Example 1.

	$c_0 = 1.65$	iseed $= -7$	$A = \frac{\mathtt{E}_{TT} - \mathtt{E}_S - \mathtt{E}_{TS}}{\mathcal{E}}$	$B = \frac{\mathtt{E}_{TT} + \mathtt{E}_S + \mathtt{E}_{TS}}{\mathcal{E}}$
N	M	$\mathtt{E}_{TT}$	$\mathtt{E}_S + \mathtt{E}_{TS}$	$[\min\{A,B\},\ \max\{A,B\}]$
5	10×10^6	-6.02×10^{-2}	5.87×10^{-4}	[1.026, 1.046]
10	50×10^6	-3.14×10^{-2}	2.33×10^{-4}	[1.019, 1.035]
20	100×10^6	-1.59×10^{-2}	1.54×10^{-4}	[1.008, 1.028]

TABLE 5.2
Adaptive choice of M and Δt with Algorithm D when TOL $= 0.02$ *in Example* 1.

Iter.	N	M	$\mathcal{E}$	$\mathtt{E}_{TT}$	$\mathtt{E}_{TS}$	$\mathtt{E}_S$
1	5	100	-2.66×10^{-2}	-5.22×10^{-2}	1.82×10^{-2}	1.44×10^{-1}
2	10	100	-9.19×10^{-3}	-2.48×10^{-2}	8.74×10^{-3}	1.35×10^{-1}
3	20	100	9.16×10^{-2}	-1.16×10^{-2}	4.25×10^{-3}	1.30×10^{-1}
4	20	1000	-1.78×10^{-2}	–	–	4.54×10^{-2}
5	20	10000	-1.77×10^{-2}	–	–	1.50×10^{-2}
6	20	14088	-1.29×10^{-2}	–	–	1.23×10^{-2}

$\mathrm{TOL}_T = 0.00444$, and $\mathrm{TOL}_{TS} = 0.00222$. When the algorithm stops, the size of the total approximation error is less than 2TOL according to the stopping criterion (4.14), and it agrees with the size of the computational error.

Stochastic time stepping algorithm. To observe the performance of the stochastic time steps Algorithm S, we apply it for different values of TOL, starting with a number of realizations $M = 100$ and a number of uniform time steps $N = 5$. Table 5.3 contains the obtained results which show that Algorithm S is also effective in giving us an approximation of the quantity of interest within the margin of 2TOL due to the criterion (4.14).

Example 2. The second example motivates the use of adaptivity by applying Algorithm D to a linear one-dimensional ($d = 1$) stochastic differential equation with time-dependent coefficients. It is interesting to observe that although here the evolution of X has many jumps, computing with Algorithm D based on jump augmented grids is still substantially more economical than doing the time stepping on a uniform mesh. The goal of the computation is given by the function $g(x) = x$, the initial condition $x_0 = 1$, and the final time $T = 1$. The coefficients of the equation have the form

$$a(t,x) = a(t)x, \quad b(t,x) = \frac{x}{\sqrt{2}}, \quad c(t,x,z) = c_* xz,$$

where $c_* > 0$ is a given constant defined below. The compensator of the jump measure is given by $q(dt,dz) = \lambda(t)\mu(dz)dt$, where $\mu(dz)$ is the uniform distribution in the interval $[-\sqrt{3},\sqrt{3}]$, and the jump intensity exhibits two different regimes,

$$\lambda(t) = \begin{cases} 0 & \text{if } 0 \le t \le 2/3, \\ 3N_J & \text{if } 2/3 < t \le 1, \end{cases}$$

with $N_J = 32^2 = 1024$. Thus, we expect in the mean $N_J = 1024$ jumps per realization. The time varying drift function is

$$a(t) = \begin{cases} 0 & \text{if } t < 1/3, \\ \frac{1}{2\sqrt{t-1/3+\mathrm{TOL}^4}} & \text{if } 1/3 \le t \le 1. \end{cases}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

TABLE 5.3
Adaptive runs with Algorithm S corresponding to different required accuracies in Example 1.

TOL	M	$\mathcal{A}(\kappa;M)$	$[\kappa_{\min},\kappa_{\max}]$	$\mathcal{S}(\kappa;M)$	$\nu_{\max}$	E_S	$\mathcal{E}$
0.040	3.2×10^3	8.3	[5, 41]	5.2	6	0.0250	−0.002
0.020	12×10^3	10.9	[5, 73]	9.2	6	0.0130	−0.006
0.010	48×10^3	20.5	[10, 146]	16.9	6	0.0066	−0.010
0.005	186×10^3	30.1	[10, 193]	32.2	7	0.0033	−0.003

Finally, the constant that defines the jump amplitude is $c_* = 1/\sqrt{N_J}$. The exact solution of the corresponding weak approximation problem is

$$E[g(X(T))] = \exp\left(\int_0^T a(t)dt\right) = \exp\left(\sqrt{T-1/3+\mathrm{TOL}^4}-\mathrm{TOL}^2\right).$$

Due to the time discontinuity of the drift function and to ensure optimal convergence of the adaptive algorithms, we modify the diffusion step in the Euler method by

$$\overline{X}(t_{n+1}^-) - \overline{X}(t_n) = a(\hat{t},\overline{X}_n)\Delta t_n + \overline{X}(t_n)\Delta W_n, \quad n=0,\dots,\kappa-1,$$

where we choose the stochastic evaluation time $\hat{t}\in\{t_n,t_{n+1}\}$ by

$$|a(\hat{t},\overline{X}_n)| = \max(|a(t_n,\overline{X}_n)|,|a(t_{n+1}^-,\overline{X}_n)|).$$

Observe that the use of $\hat{t}$ does not change the adapted nature of the Euler method. To illustrate the problem dynamics, Figure 5.1 shows three realizations of the approximate process, $\overline{X}$, and the backward dual function, $\varphi(t_n)$. The latter represents the pathwise sensitivity of $g(\overline{X}(T))$ with respect to perturbations in the initial data of X at time t_n. It coincides in this example with the first variation of $\overline{X}$, namely, $\overline{X}'(T;t_n)$. Figure 5.2 and Table 5.4 show information related to the adaptive runs for two values of the accuracy tolerance, starting in both cases with $M=10^2$ realizations and $N=10$ uniform time steps. In particular, Figure 5.2 shows the adaptive time steps needed to control the numerical error. Observe that these deterministic time steps are prior to the augmentation with the sampled jump times and are the ones we can choose to control the time discretization error. Figure 5.2 shows that although we may have more than a thousand time steps added in the interval $[2/3,1]$ due to jump times, we still need to significantly refine our deterministic time steps near the singularity at $t=1/3$. Table 5.4 shows information related to the computational effort necessary to solve the problem. There we can see that the amount of realizations needed to find our deterministic time steps, M_T, is much smaller than the amount needed to solve the problem, M. For instance, when $\mathrm{TOL}=1.0\times10^{-2}$ we need only $M_T\approx M\times10^{-3}$ realizations to control the time discretization error, making the adaptivity procedure completely inexpensive. Using the approximate error density we have also estimated the number of uniform time steps, N_U, needed to achieve the prescribed accuracy. Table 5.4 shows that N_U is approximately 100 times larger than the number of deterministic time steps, N. This fact motivates further the use of Algorithm D in this problem. Finally, we emphasize that the computational effort due to the augmentation of the time discretization with the jump times decreases as TOL decreases. Even in this case with $N_J=1024$ jump times in the mean we

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

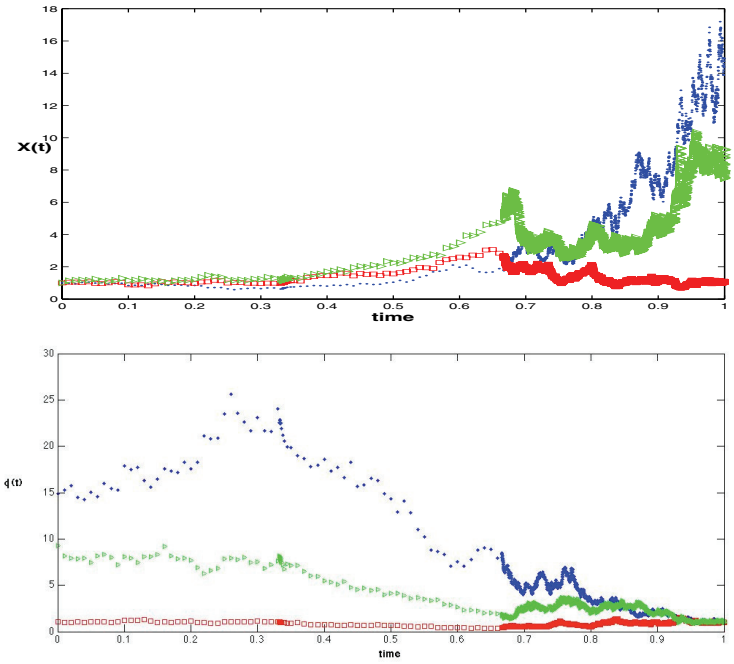


FIG. 5.1. *Three sample realizations of the processes* $\overline{X}$ *(above) and* φ *(below) obtained with modified Euler in Example* 2.

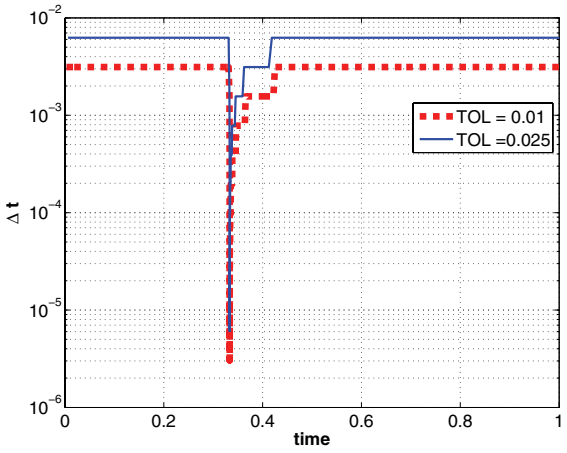


FIG. 5.2. *Final time discretizations (jumps not included) for the adaptive runs with Algorithm D and* $\mathrm{TOL} = 2.5 \times 10^{-2}$, $\mathrm{TOL} = 1 \times 10^{-2}$ *in Example* 2.

TABLE 5.4
Adaptive choice of M *and* Δt *with Algorithm D in Example* 2.

TOL	M	N	M_T	N_U	$\mathcal{E}$
2.5×10^{-2}	2.0×10^{5}	220	1.7×10^{3}	2×10^{4}	-2.0×10^{-2}
1.0×10^{-2}	1.45×10^{6}	450	1.7×10^{3}	6×10^{4}	-1.4×10^{-2}

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

anticipate that for a smaller value of TOL $\approx 10^{-3}$ the number of deterministic time steps needed to control the discretization error will be approximately 4 times larger than the mean number of jumps.

6. Proofs. This section contains the proofs postponed in the previous sections.

6.1. Proof of Lemma 2.1. The proof follows closely the proof of Corollary II.2.2 to Theorem II.2.6 (pp. 237–238 in [19]). The main two differences with this result are (i) our process has finite Lévy measure, while there a general Lévy measure is considered; (ii) our jump distribution depends on time, while there it does not. Theorem II.2.6 in [19] covers our case, and our regularity conditions on a and b are the same as in the mentioned corollary. Concerning the jumps, it is direct to see that (2.3) replaces the corresponding condition on the jump coefficient in Corollary II.2.2 in [19], following now the result from Theorem II.2.6 [19, p. 237]. The other minor difference consists of the compensation of the jump part of the differential equation (1.1). Defining

$$\alpha(t,x) = a(t,x) + \lambda(t)\int_{\mathbf{z}} c(t,x,z)\mu(t,dz),$$

we see that our equation can be written as

$$\begin{aligned} X(t) = X(0) &+ \int_0^t \alpha(\,s, X(s^-)\,)\,ds + \sum_{\ell=1}^{\ell_0}\int_0^t b^\ell(\,s, X(s^-)\,)\,dW^\ell(s) \\ &+ \int_0^t\int_{\mathbf{z}} c(\,s, X(s^-), z\,)\,\left(p(ds,dz) - q(ds,dz)\right), \end{aligned}$$

which is the form of the equation in [19]. This concludes the proof of part (a).

Part (b) is a consequence of the representation of the derivatives in terms of the fluxes, by application of the chain rule, as developed in formulas (6.9)–(6.12). We apply the Cauchy–Schwarz inequality to each summand in each representation to obtain the estimates based on the respective polynomial growth of the function g and its derivatives, and the boundedness of the moments of the process X and its variations. Since the full proof is somewhat lengthy, we do not include it here. ∎

6.2. Proof of Lemma 2.2. Based on Lemma 2.1 we know that the cost to go function $u(t,x)$ is the solution of the Kolmogorov backward equation (2.5), satisfying the regularity conditions described in (iv) of part (b) of Lemma 2.2. The Itô formula for processes with jumps (cf. [25, p. 66]) applied to (2.2) gives

$$\begin{aligned} u(T,\overline{X}(T)) = u(0,X(0)) &+ \int_0^T \left[\,\partial_t u(t,\overline{X}(t^-)) + \overline{a}_i(t;\overline{X})\,\partial_i u(t,\overline{X}(t^-))\,\right]\,dt \\ &+ \int_0^T \overline{d}_{ij}(t;\overline{X})\,\partial_{ij}u(t,\overline{X}(t^-))\,dt \\ &+ \int_0^T \overline{b}_i^\ell(t;\overline{X})\,\partial_i u(t,\overline{X}(t^-))\,dW^\ell(t) \\ &+ \int_0^T\int_{\mathbf{z}} \left[\,u\big(t,\overline{X}(t^-) + c(t,\overline{X}(t^-),z)\big) - u(t,\overline{X}(t^-))\,\right]\,p(dt,dz), \end{aligned}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

which is written equivalently as

$$\begin{aligned}
u(T,\overline{X}(T)) = u(0,X(0)) &+ \int_0^T \left[\, \partial_t u(t,\overline{X}(t^-)) + \overline{a}_i(t;\overline{X})\, \partial_i u(t,\overline{X}(t^-)) \,\right] dt \\
&+ \int_0^T \overline{d}_{ij}(t;\overline{X})\, \partial_{ij} u(t,\overline{X}(t^-))\, dt \\
&+ \int_0^T \overline{b}_i^{\ell}(t;\overline{X})\, \partial_i u(t,\overline{X}(t^-))\, dW^{\ell}(t) \\
&+ \int_0^T \lambda(t) \int_{\mathbf{z}} \left[\, u\big(t,\overline{X}(t^-) + c(t,\overline{X}(t^-),z)\big) - u(t,\overline{X}(t^-)) \,\right] \mu(t,dz)dt \\
&+ \int_0^T \int_{\mathbf{z}} [u(t,\overline{X}(t^-) + c(t,\overline{X}(t^-),z)) - u(t,\overline{X}(t^-))]\, \big(p(dt,dz) - q(dt,dz)\big),
\end{aligned}$$

where the last equality is obtained by the compensation of the integral with respect to p (remember that $q(dt,dz) = \lambda(t)\mu(t,dz)dt$). Now, use (2.5) to substitute $\partial_t u(t,\overline{X}(t^-))$ in the latter formula to obtain

$$\begin{aligned}
u(0,X(0)) = g(\overline{X}(T)) &+ \int_0^T \left[\, a_i(t,\overline{X}(t^-)) - \overline{a}_i(t;\overline{X}) \,\right] \partial_i u(t,\overline{X}(t^-))\, dt \\
&+ \int_0^T \left[\, d_{ij}(t,\overline{X}(t^-)) - \overline{d}_{ij}(t;\overline{X}) \right] \partial_{ij} u(t,\overline{X}(t^-))\, dt \\
&- \int_0^T \overline{b}_i^{\ell}(t;\overline{X})\, \partial_i u(t,\overline{X}(t^-))\, dW^{\ell}(t) \\
- \int_0^T \int_{\mathbf{z}} &\left[\, u(t,\overline{X}(t^-) + c(t,\overline{X}(t^-),z)) - u(t,\overline{X}(t^-)) \right] \big(p(dt,dz) - q(dt,dz)\big).
\end{aligned} \tag{6.1}$$

The expected value of the last two integrals is zero, the first one by the martingale property of Itô integrals and the second one due to the fact that $p(dt,dz) - q(dt,dz)$ is a compensated random measure (i.e., a martingale measure). Using the representation of u in (2.4) we obtain $E\big[u(0,X(0))\big] = E\big[g(X(T))\big]$. Therefore, taking expected values in both sides of (6.1) we arrive at the error representation (2.6). □

6.3. Proof of Theorem 2.2. The main content of Theorem 2.2 is the replacement of the (noncomputable) error estimate in Lemma 2.2 by an error expansion with computable leading order term. For this purpose the derivatives of the expected value

$$\partial_\alpha u(t,x) = \partial_\alpha E\big[g(X(T))|X(t) = x\big]$$

appearing in the integral in Lemma 2.2 are approximated by the corresponding derivatives of

$$\bar{u}(t,x) \equiv E[g(\overline{X}(T)) \mid \overline{X}(t) = x],$$

which depends on the simulated solution $\overline{X}$. The proof is divided into three steps:

(i) in Lemma 6.1 we estimate the quadrature error;
(ii) in Lemma 6.2 we bound the error in replacing $\partial_\alpha u$ by $\partial_\alpha \bar{u}$ with the use of stochastic flows and its variations;
(iii) in Lemma 6.3 we use the discrete dual functions φ and φ' (which solve the backward evolution problems; see (1.13)–(1.14) in section 1.4) to derive a computable representation of $\partial_\alpha \bar{u}$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Step 1. We begin with the estimation of the error of the quadrature approximation.

LEMMA 6.1. *Let* $\mathcal{J}_m \equiv \{n : \tilde{t}_m \leq t_n < \tilde{t}_{m+1}\}$ *for* $m = 0, \ldots, N-1$. *If the assumptions in Lemma* 2.2 *hold, then the following quadrature error estimates hold:*

$$\int_{\tilde{t}_m}^{\tilde{t}_{m+1}} E\big[\big(a_i(t, \overline{X}(t^-)) - \overline{a}_i(t; \overline{X})\big)\, \partial_i u(t, \overline{X}(t^-))\big] dt = Q_m(a_i) + \mathcal{O}\big((\Delta \tilde{t}_m)^3\big)$$

and

$$\int_{\tilde{t}_m}^{\tilde{t}_{m+1}} E\big[\big(d_{ij}(t, \overline{X}(t^-)) - \overline{d}_{ij}(t; \overline{X})\big)\partial_{ij} u(t, \overline{X}(t^-))\big] dt = \widetilde{Q}_m(d_{ij}) + \mathcal{O}\big((\Delta \tilde{t}_m)^3\big)$$

for $m = 0, \ldots, N-1$, *where*

$$Q_m(a_i) \equiv E\left[\sum_{n \in \mathcal{J}_m} \frac{\Delta t_n}{2} \left[a_i(t_{n+1}, \overline{X}(t_{n+1}^-)) - a_i(t_n, \overline{X}(t_n))\right] \partial_i u(t_{n+1}, \overline{X}(t_{n+1}^-))\right]$$

and

$$\widetilde{Q}_m(d_{ij}) \equiv E\left[\sum_{n \in \mathcal{J}_m} \frac{\Delta t_n}{2} \left[d_{ij}(t_{n+1}, \overline{X}(t_{n+1}^-)) - d_{ij}(t_n, \overline{X}(t_n))\right] \partial_{ij} u(t_{n+1}, \overline{X}(t_{n+1}^-))\right].$$

Proof. Denote by $\mathcal{F}^J$ the σ-algebra generated by the jumps and marks in $[0, T]$ constructed in section 1.2.1. Observe that in $[t_n, t_{n+1})$ the conditioned process $\overline{X}$ has no jump discontinuities, and introduce the auxiliary functions

$$\begin{aligned} \mathcal{H}(t, \overline{X}(t^-)) &\equiv \left[\, d_{ij}(t, \overline{X}(t^-)) - \overline{d}_{ij}(t; \overline{X}) \,\right] \partial_{ij} u(t, \overline{X}(t^-)), \\ \overline{\mathcal{H}}_n(t) &\equiv \frac{t - t_n}{\Delta t_n}\, \mathcal{H}(t_{n+1}, \overline{X}(t_{n+1}^-)), \quad n = 0, \ldots, \kappa - 1. \end{aligned}$$

Then we have

$$\begin{aligned} \int_{\tilde{t}_m}^{\tilde{t}_{m+1}} E[\mathcal{H}(t, \overline{X}(t^-))]\, dt &= E\left[\sum_{n \in \mathcal{J}_m} \int_{t_n}^{t_{n+1}} E\big[\mathcal{H}(t, \overline{X}(t^-))|\mathcal{F}^J\big]\, dt\right] \\ &= \widetilde{Q}_m(d_{ij}) + E\left[\sum_{n \in \mathcal{J}_m} \int_{t_n}^{t_{n+1}} E\big[\mathcal{H}(t, \overline{X}(t^-)) - \overline{\mathcal{H}}_n(t)|\mathcal{F}^J\big]\, dt\right]. \end{aligned} \tag{6.2}$$

Observing that the function $E\big[\,\overline{\mathcal{H}}_n(t)|\mathcal{F}^J\,\big]$ is the linear interpolant of the smooth function $E\big[\,\mathcal{H}(t, \overline{X}(t^-))|\mathcal{F}^J\big]$ at the nodes $\{t_n, t_{n+1}\}$, a standard interpolation error

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

estimate yields

$$\begin{aligned}\left|\int_{t_n}^{t_{n+1}} E[\mathcal{H}(t,\overline{X}(t^-)) - \overline{\mathcal{H}}_h(t)|\mathcal{F}^J]\,dt\right| \\ \leq \frac{(\Delta t_n)^2}{8}\int_{t_n}^{t_{n+1}}\left|\frac{d^2}{dt^2}E[\mathcal{H}(t,\overline{X}(t^-))|\mathcal{F}^J]\right|dt.\end{aligned} \tag{6.3}$$

Denoting $\mathcal{L}\mathcal{H} \equiv \partial_t\mathcal{H}+\overline{a}_i\partial_i\mathcal{H}+\overline{d}_{ij}\partial_{ij}\mathcal{H}$, Itô's formula and the conditions in Lemma 2.1 show that there exist constants K_1, K_2 and $p \in \mathbb{N}$ such that

$$\begin{aligned}\frac{d}{dt}E[\mathcal{H}(t,\overline{X}(t^-))|\mathcal{F}^J] &= E[\mathcal{L}\mathcal{H}(t,\overline{X}(t^-))|\mathcal{F}^J] \\ &\leq K_1 + K_2\,E[\max_{t\in[0,T]}|\overline{X}_t|^{2p}|\mathcal{F}^J], \\ \frac{d^2}{dt^2}E[\mathcal{H}(t,\overline{X}(t^-))|\mathcal{F}^J] &= E[\mathcal{L}^2\mathcal{H}(t,\overline{X}(t^-))|\mathcal{F}^J] \\ &\leq K_1 + K_2\,E[\max_{t\in[0,T]}|\overline{X}_t|^{2p}|\mathcal{F}^J],\end{aligned} \tag{6.4}$$

as all the functions involved in the derivatives of $\mathcal{H}(t,x)$ in the computation of $\mathcal{L}\mathcal{H}$ and $\mathcal{L}^2\mathcal{H}$ have polynomial growth, and $\overline{X}$ has bounded moments of all order, uniformly with respect to Δt_{max} (from this fact follows the finiteness of the moments of the supremum of the process). The combination of (6.2), (6.3), and (6.4) proves the estimate of the diffusion term in the lemma. The estimate of the drift term follows analogously. ∎

Step 2. In the second step of the proof the derivative $\partial_\alpha u$ and its approximation $\partial_\alpha \bar{u}$ are evaluated, respectively, through expected values of *stochastic flows* of X and $\overline{X}$ and its *variations*, which we now introduce. Let

$$\delta_{ik} \equiv \begin{cases} 0, & i \neq k, \\ 1, & i = k. \end{cases}$$

The following equation for the first variation of the process X at time $s > t$ holds:

$$\begin{aligned} dX'_{ij}(s) &= \partial_k a_i(s,X(s^-))\,X'_{kj}(s^-)\,ds + \partial_k b_i^\ell(s,X(s^-))\,X'_{kj}(s^-)\,dW^\ell(s) \\ &\quad + \int_{\mathbf{z}} \partial_k c_i(s,X(s^-),z)\,X'_{kj}(s^-)\,p(ds,dz), \\ X'_{ij}(t) &= \delta_{ij}. \end{aligned} \tag{6.5}$$

Similarly, for the second variation of the process X at time $s > t$ we have

$$\begin{aligned} dX''_{ijn}(s) &= \left[\partial_k a_i(s,X(s^-))\,X''_{kjn}(s^-) + \partial_{kr}a_i(s,X(s^-))\,X'_{kj}(s^-)\,X'_{rn}(s^-)\right]ds \\ &\quad + \left[\partial_k b_i^\ell(s,X(s^-))\,X''_{kjn}(s^-)\right. \\ &\qquad \left. + \partial_{kr}b_i^\ell(s,X(s^-))\,X'_{kj}(s^-)\,X'_{rn}(s^-)\right]dW^\ell(s) \\ &\quad + \int_{\mathbf{z}}\left[\partial_k c_i(s,X(s^-),z)\,X''_{kjn}(s^-)\right. \\ &\qquad \left. + \partial_{kr}c_i(s,X(s^-),z)\,X'_{kj}(s^-)\,X'_{rn}(s^-)\right]p(ds,dz), \\ X''_{ijn}(t) &= 0. \end{aligned} \tag{6.6}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

For the third variation of the process X at time $s > t$ we have

$$
\begin{aligned}
dX'''_{ijnm}(s) = \Big[& \partial_k a_i(s, X(s^-))\, X'''_{kjnm}(s^-) \\
& + \partial_{kr} a_i(s, X(s^-))\, X'_{kj}(s^-)\, X''_{rnm}(s^-) \\
& + \partial_{kr} a_i(s, X(s^-))\, X'_{kn}(s^-)\, X''_{rjm}(s^-) \\
& + \partial_{kr} a_i(s, X(s^-))\, X'_{km}(s^-)\, X''_{rjn}(s^-) \\
& + \partial_{krv} a_i(s, X(s^-))\, X'_{kj}(s^-)\, X'_{rn}(s^-)\, X'_{vm}(s^-) \Big]\, ds \\
+ \Big[& \partial_k b_i^\ell(s, X(s^-))\, X'''_{kjnm}(s^-) \\
& + \partial_{kr} b_i^\ell(s, X(s^-))\, X'_{kj}(s^-)\, X''_{rnm}(s^-) \\
& + \partial_{kr} b_i^\ell(s, X(s^-))\, X'_{kn}(s^-)\, X''_{rjm}(s^-) \\
& + \partial_{kr} b_i^\ell(s, X(s^-))\, X'_{km}(s^-)\, X''_{rjn}(s^-) \\
& + \partial_{krv} b_i^\ell(s, X(s^-))\, X'_{kj}(s^-)\, X'_{rn}(s^-)\, X'_{vm}(s^-) \Big]\, dW^\ell(s) \\
+ \int_{\mathbf{z}} \Big[& \partial_k c_i(s, X(s^-))\, X'''_{kjnm}(s^-) \\
& + \partial_{kr} c_i(s, X(s^-), z)\, X'_{kj}(s^-) X''_{rnm}(s^-) \\
& + \partial_{kr} c_i(s, X(s^-), z)\, X'_{kn}(s^-)\, X''_{rjm}(s^-) \\
& + \partial_{kr} c_i(s, X(s^-), z)\, X'_{km}(s^-)\, X''_{rjn}(s^-) \\
& + \partial_{krv} c_i(s, X(s^-), z)\, X'_{kj}(s^-)\, X'_{rn}(s^-)\, X'_{vm}(s^-) \Big]\, p(ds, dz), \\
X'''_{ijnm}(t) = & \, 0,
\end{aligned}
\tag{6.7}
$$

and similarly for the fourth variation of the process X at time $s > t$:

$$
dX''''_{ijnmp} = \ldots, \qquad X''''_{ijnmp}(t) = 0. \tag{6.8}
$$

Let $I^{d\times d}$ be the $d \times d$ identity matrix. The equations above imply the representation of the derivatives of expectations with stochastic flows (cf. [19], [49], and [52]) as follows: for the first order derivatives

$$
\partial_k u(t,x) = E\big[\, \partial_i g(X(T))\, X'_{ik}(T) \mid X^{I}(t) = (x, I^{d\times d}) \,\big] \tag{6.9}
$$

with $X^{I} \equiv (X, X')$; for the second order derivatives

$$
\begin{aligned}
\partial_{kn} u(t,x) = E\big[& \partial_i g(X(T))\, X''_{ikn}(T) \\
& + \partial_{ir} g(X(T))\, X'_{ik}(T)\, X'_{rn}(T) \mid X^{II}(t) = (x, I^{d\times d}, 0) \,\big]
\end{aligned}
\tag{6.10}
$$

with $X^{II} \equiv (X, X', X'')$; for the third order derivatives

$$
\begin{aligned}
\partial_{knm} u(t,x) = E\big[& \partial_i g(X(T))\, X'''_{iknm}(T) + \partial_{ir} g(X(T))\, X'_{ik}(T)\, X''_{rnm}(T) \\
& + \partial_{ir} g(X(T))\, X'_{in}(T)\, X''_{rkm}(T) \\
& + \partial_{irv} g(X(T))\, X'_{ik}(T)\, X'_{rn}(T)\, X'_{vm}(T) \\
& + \partial_{ir} g(X(T))\, X'_{im}(T)\, X''_{rkn}(T) \mid X^{III}(t) = (x, I^{d\times d}, 0, 0) \big]
\end{aligned}
\tag{6.11}
$$

with $X^{III} \equiv (X, X', X'', X''')$; and, analogously, for the fourth order derivatives

$$
\partial_{knmp} u(t,x) = \ldots. \tag{6.12}
$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Let $Y = (X, X', X'', X''', X'''')^T$. Then the system consisting of (1.1) and (6.5)–(6.8) can be written as

$$\begin{aligned} dY(t) &= A(t, Y(t^-))\, dt + B^\ell(t, Y(t^-))\, dW^\ell(t) \\ &\quad + \int_{\mathbf{z}} C(t, Y(t^-), z)\, p(dt, dz), \quad t > t_0, \\ Y(t_0) &= (x, I^{d\times d}, 0, 0, 0)^T. \end{aligned} \tag{6.13}$$

Furthermore, in order to simplify the writing of the formulas (6.9)–(6.12), we introduce below additional notation:

$$\begin{aligned} f_i(Y(T)) &\equiv \partial_k g(X(T))\, X'_{ki}(T), \\ f_{ij}(Y(T)) &\equiv \partial_k g(X(T))\, X''_{kij}(T) + \partial_{kn} g(X(T))\, X'_{ki}(T)\, X'_{nj}(T), \\ f_{ijm}(Y(T)) &\equiv \partial_k g(X(T))\, X'''_{kijm}(T) + \partial_{kn} g(X(T))\, X'_{ki}(T)\, X''_{njm}(T) \\ &\quad + \partial_{kn} g(X(T))\, X'_{kj}(T)\, X''_{nim}(T) \\ &\quad + \partial_{kn} g(X(T))\, X'_{km}(T)\, X''_{nji}(T) \\ &\quad + \partial_{knv} g(X(T))\, X'_{ki}(T)\, X'_{nj}(T)\, X'_{vm}(T), \\ f_{ijmn}(Y(T)) &\equiv \dots. \end{aligned} \tag{6.14}$$

The Euler approximation of Y, the solution of (6.13), is denoted by

$$\overline{Y} = \big(\overline{Y}^0, \overline{Y}^1, \overline{Y}^2, \overline{Y}^3, \overline{Y}^4\big)^T$$

and can be extended as the solution of the stochastic differential equation with piecewise constant drift and diffusion fluxes

$$d\overline{Y} = \overline{A}(t; \overline{Y})\, dt + \overline{B}^\ell(t; \overline{Y})\, dW^\ell(t) + \int_{\mathbf{z}} C(t, \overline{Y}, z)\, p(dt, dz), \tag{6.15}$$

defined as in (1.4), (1.6), and (2.2), with coefficients defined as in (2.1).

An important consequence of the Euler method is that the variation and the Euler discretization commute. This yields for each α the representation

$$\begin{aligned} &\partial_\alpha u(t,x) - \partial_\alpha \bar{u}(t,x) \\ &\qquad = E\big[f_\alpha(Y(T)) - f_\alpha(\overline{Y}(T)) \mid Y(t) = \overline{Y}(t) = (x, I^{d\times d}, 0, 0, 0)^T\big]. \end{aligned} \tag{6.16}$$

Now we rely on Lemma 2.2 applied to the process Y to obtain the representation

$$\begin{aligned} &E\big[f_\alpha(Y(T)) - f_\alpha(\overline{Y}(T)) \mid Y(t) = \overline{Y}(t) = (x, I^{d\times d}, 0, 0, 0)^T\big] \\ &\quad = \int_t^T E\Big[(A - \overline{A})_k\, \partial_k \nu^\alpha(s, \overline{Y}(s^-)) \,\Big|\, \overline{Y}(t) = (x, I^{d\times d}, 0, 0, 0)^T\Big]\, ds \\ &\qquad + \int_t^T E\Big[(D - \overline{D})_{kn}\, \partial_{kn} \nu^\alpha(s, \overline{Y}(s^-)) \,\Big|\, \overline{Y}(t) = (x, I^{d\times d}, 0, 0, 0)^T\Big]\, ds \end{aligned} \tag{6.17}$$

to be used in Lemmas 6.2 and 6.3 below, where the cost to go function ν^α is, for each f_α in (6.14), defined as

$$\nu^\alpha(t, y) = E\big[f_\alpha(Y(T)) \mid Y(t) = y\big].$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

We use the notation $D_{kn} = \frac{1}{2}B_k^\ell B_n^\ell$ and $\overline{D}_{kn} = \frac{1}{2}\overline{B}_k^\ell \overline{B}_n^\ell$, and the corresponding Kolmogorov backward equation for $t > T$ is

$$\begin{aligned}\mathcal{L}^Y \nu^\alpha &\equiv \partial_t \nu^\alpha + A_k \partial_k \nu^\alpha + D_{kn}\partial_{kn}\nu^\alpha \\ &\quad + \lambda(t)\int_{\mathbf{z}} [\nu^\alpha(t, Y + C(t,Y,z)) - \nu^\alpha(t,Y)]\,\mu(t,dz) = 0,\\ \nu^\alpha(T,\cdot) &= f_\alpha.\end{aligned}$$

We are ready for the second step.

LEMMA 6.2 (approximation of $\partial_\alpha u$). *Let the piecewise constant mesh function Δt be defined by $\Delta t(s) \equiv \Delta t_n$ for $s \in [t_n, t_{n+1})$ and $n = 0, \ldots, \kappa - 1$. Suppose that the assumptions in Lemma 2.2 hold. Then the discretization errors of the stochastic flows, for $|\alpha| \le 4$, satisfy*

$$\partial_\alpha(u - \bar{u})(t_n, \overline{X}(t_n)) = \int_{t_n}^T E[\mathcal{O}(\Delta t(s)) \mid \mathcal{F}_{t_n}]\, ds = \mathcal{O}(\Delta \tilde{t}_{max}). \tag{6.18}$$

Furthermore

$$\begin{aligned}E\Big[\,\big[a_i(t_{n+1}, \overline{X}(t_{n+1}^-)) - a_i(t_n, \overline{X}(t_n))\big]\,\partial_i(u-\bar{u})(t_{n+1}, \overline{X}(t_{n+1}^-)) \mid \mathcal{F}^J\Big] \\ = \Delta t_n \int_{t_{n+1}}^T \mathcal{O}(\Delta t(s))\, ds\end{aligned} \tag{6.19}$$

and

$$\begin{aligned}E\Big[\,\big[\,d_{ij}(t_{n+1}, \overline{X}(t_{n+1}^-)) - d_{ij}(t_n, \overline{X}(t_n))\,\big]\,\partial_{ij}(u-\bar{u})(t_{n+1}, \overline{X}(t_{n+1}^-)) \mid \mathcal{F}^J\,\Big] \\ = \Delta t_n \int_{t_{n+1}}^T \mathcal{O}(\Delta t(s))\, ds.\end{aligned} \tag{6.20}$$

Proof. The combination of (6.16) and (6.17) gives

$$\begin{aligned}\partial_\alpha(u-\bar{u})(t_n, \overline{X}(t_n)) &= \int_{t_n}^T E\big[(A_i - \overline{A}_i)\,\partial_i \nu^\alpha(s, \overline{Y}(s^-)) \mid \mathcal{F}_{t_n}\big]\, ds \\ &\quad + \int_{t_n}^T E\big[(D_{ij} - \overline{D}_{ij})\,\partial_{ij}\nu^\alpha(s, \overline{Y}(s^-)) \mid \mathcal{F}_{t_n}\big]\, ds.\end{aligned} \tag{6.21}$$

Now, for $t_m \le t < t_{m+1}$, introduce the notation

$$h(t, \overline{Y}(t)) \equiv (A_i - \overline{A}_i)\partial_i \nu^\alpha(t, \overline{Y}(t)) + (D_{ij} - \overline{D}_{ij})\partial_{ij}\nu^\alpha(t, \overline{Y}(t))$$

and

$$\begin{aligned}\mathcal{L}_0 w(t,y) &\equiv \big(\partial_t w + \overline{A}_n \partial_n w + \overline{D}_{kn}\partial_{kn} w\big)(t,y),\\ \mathcal{L}^{\overline{Y}} w(t,y) &\equiv \mathcal{L}_0 w(t,y) + \lambda(t)\int_{\mathbf{z}} [w(t, y + C(t,y,z)) - w(t,y)]\mu(t,dz).\end{aligned}$$

Observe that $h(t_m, \overline{Y}(t_m)) = 0$, and apply Itô's formula in the interval $[t_m, t]$ to obtain

$$E[h(t, \overline{Y}(t^-)) \mid \mathcal{F}_{t_m}] = \int_{t_m}^t E[\mathcal{L}^{\overline{Y}} h(s, \overline{Y}(s^-)) \mid \mathcal{F}_{t_m}] ds = \mathcal{O}(\Delta t_m), \quad t_m \le t < t_{m+1}, \tag{6.22}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

where the bound follows as in (6.4). Combining (6.22) with (6.21) proves (6.18).

The estimate (6.20) follows similarly by defining now

$$\tilde{h}(t,\overline{X}(t)) \equiv (d_{ij}-\overline{d}_{ij})\partial_{ij}(u-\bar{u})(t,\overline{X}(t)).$$

Then, as in (6.3)–(6.4), the Itô's formula shows

$$E[\tilde{h}(t,\overline{X}(t)) \mid \mathcal{F}^J] = \int_{t_n}^{t} E[\mathcal{L}_0\tilde{h}(s,\overline{X}(s^-)) \mid \mathcal{F}^J]ds, \quad t_n \le t < t_{n+1}.$$

The final step to proving (6.20) is to establish

$$E[\mathcal{L}_0\,\tilde{h}(s,\overline{X}(s^-)) \mid \mathcal{F}^J] = \int_{s}^{T} \mathcal{O}(\Delta t(\tau))d\tau.$$

The function $\mathcal{L}_0\,\tilde{h}(s,\overline{X}(s^-))$ splits into the two types of terms $h_1 \equiv (d-\overline{d})v$ and $h_2 \equiv v\partial_\alpha(u-\bar{u})$, with smooth functions v of $(s,\overline{X}(s))$. The Itô formula again shows that

$$E[h_1(s,\overline{X}(s^-)) \mid \mathcal{F}^J] = \int_{t_n}^{s} E[\mathcal{L}_0 h_1(\tau,\overline{X}(\tau^-)) \mid \mathcal{F}^J]d\tau = \mathcal{O}(\Delta t_n), \quad t_n \le s < t_{n+1}.$$

Moreover (6.18) implies

$$E[h_2(s,\overline{X}(s^-)) \mid \mathcal{F}^J] = \int_{s}^{T} \mathcal{O}(\Delta t(\tau))d\tau,$$

and consequently (6.20) holds. The estimate (6.19) of the drift terms follows analogously. □

Step 3. The last step is to show a result on the representation of the derivative of $\bar{u}$ using discrete duals.

LEMMA 6.3. *Suppose that the assumptions in Lemma* 2.2 *hold. Then the dual functions* φ *and* φ', *defined in* (1.13) *and* (1.14), *satisfy for* $t = t_n$ *and* $t = t_{n+1}^-$

$$\partial_i\bar{u}(t,\overline{X}(t)) = E[\varphi_i(t) \mid \mathcal{F}_t], \tag{6.23}$$

$$\partial_{ij}\bar{u}(t,\overline{X}(t)) = E[\varphi'_{ij}(t) \mid \mathcal{F}_t]. \tag{6.24}$$

Proof. Equations (2.2), (6.5), and (6.9) show that the first variation of the Euler approximation $\overline{X}$ is in fact equal to the Euler approximation of the first variation X', and consequently

$$\partial_i\bar{u}(t,\overline{X}(t)) = E[\partial_j g(\overline{X}(T))\overline{X}'_{ji}(T;t) \mid \mathcal{F}_t],$$

where $\overline{X}'_{ji}(s;t)$ $(s > t)$, is the Euler approximation (6.15) of X' with initial data $\overline{X}'_{ji}(t;t) = \delta_{ji}$.

For $t = \ldots, t_n, t_{n+1}^-, \ldots$, let

$$\varphi_i(t) \equiv \partial_{x_i(t)} g(\overline{X}(T)),$$

i.e.,

$$\varphi_i(t) = \partial_j g(\overline{X}(T))\overline{X}'_{ji}(T;t). \tag{6.25}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

We prove inductively that $\varphi_i(t)$ is the solution of the corresponding problem in (1.12)–(1.13). Since (1.12) is trivially true it remains to prove the inductive step. By the chain rule we have

$$\partial_k g(\overline{X}(T))\overline{X}'_{ki}(T;t) = \partial_k g(\overline{X}(T))\overline{X}'_{kj}(T;t_{next})\overline{X}'_{ji}(t_{next};t)$$

or, in other words,

$$\varphi_i(t) = \varphi_j(t_{next})\partial_i \Phi_j(t,\overline{X}(t)), \tag{6.26}$$

which is equivalent to (1.12)–(1.13), which is what we wanted to prove.

The equality (6.25) implies that

$$\partial_{ij}\bar{u}(t,\overline{X}(t)) = E\big[\partial_{x_j(t)}\varphi_i(t) \mid \mathcal{F}_t\big].$$

The next step is to verify that the first variation of φ,

$$\varphi'_{ij}(t) \equiv \partial_{x_j(t)}\varphi_i(t) \equiv \frac{\partial \varphi_i(t;\overline{X}(t)=x)}{\partial x_j}, \tag{6.27}$$

satisfies the backward recursive equation (1.14). First, differentiate the equation (6.26) to obtain

$$\begin{aligned} \varphi'_{ik}(t) &= \partial_i\Phi_j\partial_{x_k(t)}\varphi_j(t_{next}) + \partial_k\partial_i\Phi_j\varphi_j(t_{next}), \quad t<T,\\ \varphi'_{ik}(T) &= \partial_{ik}g(\overline{X}(T)). \end{aligned} \tag{6.28}$$

Observe that problem (1.12)–(1.16) shows that $\varphi(t_{next})$ depends only on the point values

$$\{\overline{X}(s):\ t_{next}\le s\le T\},$$

so that

$$\partial_{x_k(t)}\varphi_j(t_{next}) = \partial_{x_p(t_{next})}\varphi_j(t_{next})\partial_{x_k(t)}\overline{X}_p(t_{next}). \tag{6.29}$$

Finally, the definitions of $\overline{X}$ and Φ in (2.2) and (1.11) imply

$$\partial_k\Phi_p(t,\overline{X}(t)) = \partial_{x_k(t)}\overline{X}_p(t_{next}), \tag{6.30}$$

which together with (6.28)–(6.29) proves that φ' satisfies the recursive equation

$$\begin{aligned} \varphi'_{ik}(t) &= \partial_i\Phi_j\ \ \varphi'_{jp}(t_{next})\partial_k\Phi_p(t,\overline{X}(t)) + \partial_k\partial_i\Phi_j\ \ \varphi_j(t_{next}), \quad t<T,\\ \varphi'_{ik}(T) &= \partial_{ik}g(\overline{X}(T)), \end{aligned} \tag{6.31}$$

which is equivalent to (1.12)–(1.14). ∎

Remark 6.1. The measurability of $\big(a_i(t_{n+1}^-,\overline{X}(t_{n+1}^-)) - \overline{a}_i(t_n,\overline{X}(t_n))\big) \in \mathcal{F}_{t_{n+1}^-}$ proves that for $t = t_{n+1}^-$ and any random variable β we have

$$\begin{aligned} E\big[\big(a_i(t,\overline{X}(t)) - \overline{a}_i(t_n,\overline{X}(t_n))\big)E[\beta\mid\mathcal{F}_t]\big] &= E\big[E[\big(a_i(t,\overline{X}(t)) - \overline{a}_i(t_n;\overline{X})\big)\beta\mid\mathcal{F}_t]\big]\\ &= E\big[\big(a_i(t,\overline{X}(t)) - \overline{a}_i(t_n;\overline{X})\big)\beta\big], \end{aligned}$$

and in a completely similar way we obtain the same result for the diffusion terms, i.e.,

$$E\big[\big(d_{ij}(t,\overline{X}(t)) - \overline{d}_{ij}(t_n;\overline{X})\big)E[\beta\mid\mathcal{F}_t]\big] = E\big[\big(d_{ij}(t,\overline{X}(t)) - \overline{d}_{ij}(t_n;\overline{X})\big)\beta\big].$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

The proof of Theorem 2.2 is now concluded by combining Lemmas 6.1, 6.2, and 6.3, Remark 6.1, and the central limit theorem to estimate I_M and J_M. We have

$$\begin{aligned}
&\int_{\tilde{t}_m}^{\tilde{t}_{m+1}} E\big[\big(a_i(t,\overline{X}(t^-)) - \overline{a}_i(t;\overline{X})\big)\,\partial_i u(t,\overline{X}(t^-))\big]dt \\
&\quad= E\left[\sum_{n\in\mathcal{J}_m} \frac{\Delta t_n}{2}\left[a_i(t_{n+1},\overline{X}(t_{n+1}^-)) - a_i(t_n,\overline{X}(t_n))\right]\partial_i u(t_{n+1},\overline{X}(t_{n+1}^-))\right] \\
&\qquad\quad+ \sum_{m=0}^{N-1}\mathcal{O}\big((\Delta\tilde{t}_m)^3\big) \\
&\quad= E\left[\sum_{n\in\mathcal{J}_m} \frac{\Delta t_n}{2}\left[\,a_i(t_{n+1},\overline{X}(t_{n+1}^-)) - a_i(t_n,\overline{X}(t_n))\,\right]\partial_i\bar{u}(t_{n+1},\overline{X}(t_{n+1}^-))\right] \\
&\qquad\quad+ \sum_{m=0}^{N-1} E\left[\sum_{n\in\mathcal{J}_m}(\Delta t_n)^2\int_{t_{n+1}}^{T}\mathcal{O}(\Delta t(s))\,ds\right] + \sum_{m=0}^{N-1}\mathcal{O}\big((\Delta\tilde{t}_m)^3\big) \\
&\quad= E\left[\sum_{n\in\mathcal{J}_m} \frac{\Delta t_n}{2}\left[\,a_i(t_{n+1},\overline{X}(t_{n+1}^-)) - a_i(t_n,\overline{X}(t_n))\,\right]E\big[\varphi_i(t_{n+1-})\,\big|\,\mathcal{F}_{t_{n+1}^-}\big]\right] \\
&\qquad\quad+ \sum_{m=0}^{N-1} E\left[\sum_{n\in\mathcal{J}_m}(\Delta t_n)^2\int_{t_{n+1}}^{T}\mathcal{O}(\Delta t(s))\,ds\right] + \sum_{m=0}^{N-1}\mathcal{O}\big((\Delta\tilde{t}_m)^3\big) \\
&\quad= E\left[\sum_{n\in\mathcal{J}_m} \frac{\Delta t_n}{2}\left[\,a_i(t_{n+1},\overline{X}(t_{n+1}^-)) - a_i(t_n,\overline{X}(t_n))\,\right]\varphi_i(t_{n+1-})\right] \\
&\qquad\quad+ \sum_{m=0}^{N-1} E\left[\sum_{n\in\mathcal{J}_m}(\Delta t_n)^2\int_{t_{n+1}}^{T}\mathcal{O}(\Delta t(s))\,ds\right] + \sum_{m=0}^{N-1}\mathcal{O}\big((\Delta\tilde{t}_m)^3\big).
\end{aligned}$$

The expansion of the diffusion term appearing in (2.8) follows analogously. ∎

6.4. Proof of Theorem 3.1. We consider the difference

$$g(X(t)) - g(\overline{X}(t))$$

and apply Theorem 3.3 from [51], using also the fact that there is no time discretization error at the jump nodes. To this end, denote the set of stochastic time nodes by $\mathcal{J} \equiv \{0 = t_0, t_1^-, t_1, t_2^-, t_2, \ldots, t_N = T\}$ and recall that the notation

$$\overline{X}(t_{next}) = \Phi(\overline{X}(t)), \quad t \in \mathcal{J}, \tag{6.32}$$

introduced in Definition 1.1 denotes one step with the Euler method. Write similarly one step with the exact solution

$$X(t_{next}) = \hat{\Phi}(X(t)), \quad t \in \mathcal{J}. \tag{6.33}$$

Introduce the notation $X^t \equiv X(t)$ and $\overline{X}^t \equiv \overline{X}(t)$. Now verify the representation

$$g(X(T)) - g(\overline{X}(T)) = \sum_{n=0}^{\kappa-1}(\hat{\Phi}(\overline{X}(t_n)) - \Phi(\overline{X}(t_n)))_i\tilde{\varphi}_i(t_{n+1}^-), \tag{6.34}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

where the weight functions are defined recursively by linear backward recursion:

$$\begin{aligned}\tilde{\varphi}_i(T) &= \int_0^1 \partial_i g(sX(T) + (1-s)\overline{X}(T))ds,\\ \tilde{\varphi}_i(t) &= \left(\int_0^1 \partial_i \hat{\Phi}_j(sX(t) + (1-s)\overline{X}(t))ds\right)\tilde{\varphi}_j(t_{next}),\ t \in \mathcal{J}.\end{aligned} \tag{6.35}$$

To verify (6.34), first observe that by construction of the Euler method at every jump point $t = t_n^-$ there is no local error in computing the next $\overline{X}$ value at time t_n, i.e.,

$$\hat{\Phi}(\overline{X}^t) - \Phi(\overline{X}^t) = 0,$$

so (6.34) is equivalent to

$$g(X(T)) - g(\overline{X}(T)) = \sum_{t\in\mathcal{J}} (\hat{\Phi}(\overline{X}^t) - \Phi(\overline{X}^t))_i \tilde{\varphi}_i(t_{next}). \tag{6.36}$$

Then telescoping cancellation gives

$$g(X(T)) - g(\overline{X}(T)) = \sum_{t\in\mathcal{J}} \big((X^{t_{next}} - \overline{X}^{t_{next}})_i \tilde{\varphi}_i(t_{next}) - (X^t - \overline{X}^t)_i \tilde{\varphi}_i(t)\big). \tag{6.37}$$

Use the definitions (6.32) and (6.33) and split the first term in the sum of (6.37) into

$$(\hat{\Phi}(X^t) - \hat{\Phi}(\overline{X}^t) + \hat{\Phi}(\overline{X}^t) - \Phi(\overline{X}^t))_i \tilde{\varphi}_i(t_{next}).$$

The two first terms above and the last term in the sum of (6.37) combine to zero by (6.35):

$$(\hat{\Phi}(X^t) - \hat{\Phi}(\overline{X}^t))_i \tilde{\varphi}_i(t_{next})(X^t - \overline{X}^t)_i \tilde{\varphi}_i(t),$$

which proves (6.34).

The next step is to use the Malliavin derivative to analyze the expectation of the representation (6.34) by studying the dependence of $\overline{X}$ and $\tilde{\varphi}$ on a small increment dW. This follows exactly the lines of the proof of Theorem 3.3 from [51], and it is not reproduced here. □

Remark 6.2. The assumptions in Theorems 2.2 and 3.1 can be relaxed. For instance, the results of Theorem 2.2 also hold when λ and μ are piecewise continuously differentiable, provided that (i) they have a finite number of discontinuities; (ii) those discontinuity times are included in the time discretization; and (iii) for each time subinterval between discontinuities, the first derivatives of λ and μ are bounded and have lateral limits.

Acknowledgment. The authors are indebted to the anonymous referees for their useful comments regarding the manuscript.

REFERENCES

[1] L. Andersen and J. Andreasen, *Jump-diffusion processes: Volatility smile fitting and numerical methods for options pricing*, Rev. Deriv. Res., 4 (2000), pp. 231–262.

[2] S. Asmussen, F. Avram, and M.R. Pistorius, *Russian and American put options under exponential phase-type Lévy models*, Stochastic Process. Appl., 109 (2004), pp. 79–111.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

[3] S. Asmussen and J. Rosiński, *Approximations of small jumps of Lévy processes with a view towards simulation*, J. Appl. Probab., 38 (2001), pp. 482–493.
[4] I. Babuška, A. Miller, and M. Vogelius, *Adaptive methods and error estimation for elliptic problems of structural mechanics*, in Adaptive Computational Methods for Partial Differential Equations (College Park, MD, 1983), SIAM, Philadelphia, PA, 1983, pp. 57–73.
[5] R. Becker and R. Rannacher, *A feed-back approach to error control in finite element methods: Basic analysis and examples*, East-West J. Numer. Math., 4 (1996), pp. 237–264.
[6] N. Bruti-Liberati and E. Platen, *On the Strong Approximation of Jump-Diffusion Processes*, Research paper 157, Quantitative Finance Research Center, University of Technology Sydney, Sydney, Australia, 2005.
[7] N. Bruti-Liberati and E. Platen, *On Weak Predictor-Corrector Schemes for Jump-Diffusion Processes in Finance*, Research paper 179, Quantitative Finance Research Center, University of Technology Sydney, Sydney, Australia, 2006.
[8] N. Bruti-Liberati and E. Platen, *Strong approximations of stochastic differential equations with jumps*, J. Comput. Appl. Math., 205 (2007), pp. 982–1001.
[9] N. Bruti-Liberati and E. Platen, *Approximation of jump diffusions in finance and economics*, Comput. Econom., 29 (2007), pp. 283–312.
[10] P. Carr, H. Geman, D.B. Madan, and M. Yor, *The fine structure of asset returns: An empirical investigation*, J. Business, 75 (2002), pp. 305–332.
[11] R. Cont and P. Tankov, *Financial Modelling with Jump Processes*, Chapman Hall/CRC Financ. Math. Ser., Chapman & Hall/CRC, Boca Raton, FL, 2004.
[12] R. Cont and P. Tankov, *Non-parametric calibration of jump-diffusion option pricing models*, J. Comput. Finance, 7 (2002), pp. 1–49.
[13] D. Duffie, D. Filipović, and W. Schachermayer, *Affine processes and applications in finance*, Ann. Appl. Probab., 13 (2003), pp. 984–1053.
[14] R. Durett, *Probability: Theory and Examples*, 3rd ed., Duxbury Advanced Series, Brooks/Cole-Thomson Learning, Belmont, CA, 2005.
[15] A. Dzougoutov, K.-S. Moon, E. von Schwerin, A. Szepessy, and R. Tempone, *Adaptive Monte Carlo algorithms for stopped diffusion*, in Multiscale Methods in Science and Engineering, B. Enguist, P. Lötstedt, and O. Runborg, eds., Lect. Notes Comput. Sci. Eng. 44, Springer-Verlag, Berlin, 2005, pp. 59–88.
[16] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson, *Introduction to adaptive methods for differential equations*, Acta Numer., 4 (1995), pp. 105–158.
[17] E.F. Fama, *Portfolio analysis in a stable Paretian market*, Management Sci., 11 (1965), pp. 404–419.
[18] G.S. Fishman, *Monte Carlo. Concepts, Algorithms and Applications*, Springer Ser. Oper. Res., Springer-Verlag, New York, 1996.
[19] I.I. Gikhman and A.V. Skorokhod, *The Theory of Stochastic Processes* III, Monographs in Probability Theory and Mathematical Statistics, Nauka, Moscow, 1975 (in Russian).
[20] P. Glasserman and N. Merener, *Numerical solution of jump-diffusion LIBOR market models*, Finance Stoch., 7 (2003), pp. 1–27.
[21] P. Glasserman and N. Merener, *Convergence of a discretization scheme for jump-diffusion processes with state-dependent intensities*, Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci., 460 (2004), pp. 111–127.
[22] D.J. Higham and P.E. Kloeden, *Numerical methods for nonlinear stochastic differential equations with jumps*, Numer. Math., 101 (2005), pp. 101–119.
[23] D.J. Higham and P.E. Kloeden, *Convergence and stability of implicit methods for jump-diffusion systems*, Int. J. Numer. Anal. Model., 3 (2006), pp. 125–140.
[24] D.J. Higham and P.E. Kloeden, *Strong convergence rates for backward Euler on a class of nonlinear jump-diffusion problems*, J. Comput. Appl. Math., 205 (2007), pp. 949–956.
[25] N. Ikeda and S. Watanabe, *Stochastic Differential Equations and Diffusion Processes*, 2nd ed., North-Holland Math. Library 24, North-Holland, Amsterdam, 1989.
[26] K. Itô, *Essentials of Stochastic Processes*, Transl. Math. Monogr. 231, translated from the 1957 Japanese original by Y. Itô, AMS, Providence, RI, 2006.
[27] J. Jacod and A.N. Shiryaev, *Limit Theorems for Stochastic Processes*, Grundlehren Math. Wiss. 288, Springer-Verlag, Berlin, 1987.
[28] C. Johnson, *Error estimates and adaptive time-step control for a class of one-step methods for stiff ordinary differential equations*, SIAM J. Numer. Anal., 25 (1988), pp. 908–926.
[29] C. Johnson and A. Szepessy, *Adaptive finite element methods for conservation laws based on a posteriori error estimates*, Comm. Pure Appl. Math., 48 (1995), pp. 199–234.
[30] I. Karatzas and S.E. Shreve, *Brownian Motion and Stochastic Calculus*, Grad. Texts in Math. 113, Springer-Verlag, New York, 1988.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

[31] P.E. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equations*, Appl. Math. 23, Springer-Verlag, New York, 1992.
[32] K. Kubilius and E. Platen, *Rate of weak convergence of the Euler approximation for diffusion processes with jumps*, Monte Carlo Methods Appl., 8 (2002), pp. 83–96.
[33] X.Q. Liu and C.W. Li, *Weak approximations and extrapolations of stochastic differential equations with jumps*, SIAM J. Numer. Anal., 37 (2000), pp. 1747–1767.
[34] B. Mandelbrot, *The variation of certain speculative prices*, J. Business, 36 (1963), pp. 394–419.
[35] G. Maruyama, *Continuous Markov processes and stochastic equations*, Rend. Circ. Mat. Palermo, 4 (1955), pp. 48–90.
[36] A.-M. Matache, T. von Petersdorff, and C. Schwab, *Fast deterministic pricing of options on Lévy driven assets*, Math. Model. Numer. Anal., 38 (2004), pp. 37–71.
[37] R.C. Merton, *Option pricing when the underlying stock returns are discontinuous*, J. Financial Econom., 3 (1976), pp. 125–144.
[38] R. Mikulevičius and E. Platen, *Time discrete Taylor approximations for Itô processes with jump component*, Math. Nachr., 138 (1988), pp. 99–104.
[39] G.N. Milstein and M.V. Tretyakov, *Stochastic Numerics for Mathematical Physics*, Sci. Comput., Springer-Verlag, Berlin, Heidelberg, 2004.
[40] K.-S. Moon, E. von Schwerin, A. Szepessy, and R. Tempone, *An adaptive algorithm for ordinary, stochastic and partial differential equations*, in Recent Advances in Adaptive Computation, Z.-C. Shi, Z. Chen, T. Tang, and D. Yu, eds., Contemp. Math. 383, AMS, Providence, RI, 2005, pp. 325–343.
[41] K.-S. Moon, E. von Schwerin, A. Szepessy, and R. Tempone, *Convergence rates for an adaptive dual weighted residual finite element algorithm*, BIT, 46 (2006), pp. 367–407.
[42] K.-S. Moon, A. Szepessy, R. Tempone, and G.E. Zouraris, *Hyperbolic differential equations and adaptive numerics*, in Theory and Numerics of Differential Equations (Durham, 2000), J.F. Blowey, J.P Coleman, and A.W. Craig, eds., Universitext, Springer-Verlag, Berlin, 2001, pp. 231–280.
[43] K.-S. Moon, A. Szepessy, R. Tempone, and G.E. Zouraris, *A variational principle for adaptive approximation of ordinary differential equations*, Numer. Math., 96 (2003), pp. 131–152.
[44] K.-S. Moon, A. Szepessy, R. Tempone, and G.E. Zouraris, *Convergence rates for adaptive approximation of ordinary differential equations*, Numer. Math., 96 (2003), pp. 99–129.
[45] K.-S. Moon, A. Szepessy, R. Tempone, and G.E. Zouraris, *Convergence rates for adaptive weak approximation of stochastic differential equations*, Stochastic Anal. Appl., 23 (2005), pp. 511–558.
[46] D. Nualart, *The Malliavin Calculus and Related Topics*, Probab. Appl., Springer-Verlag, New York, 1995.
[47] E. Platen, *An approximation method for a class of Itô processes with jump component*, Litovsk. Mat. Sb., 22 (1982), pp. 124–136.
[48] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipes in FORTRAN. The Art of Scientific Computing*, 2nd ed., Cambridge University Press, Cambridge, UK, 1992.
[49] P. Protter, *Stochastic Integration and Differential Equations. A New Approach*, Appl. Math. 21, Springer-Verlag, Berlin, 1990.
[50] P. Protter and D. Talay, *The Euler scheme for Lévy driven stochastic differential equations*, Ann. Probab., 25 (1997), pp. 393–423.
[51] A. Szepessy, R. Tempone, and G.E. Zouraris, *Adaptive weak approximation of stochastic differential equations*, Comm. Pure Appl. Math., 54 (2001), pp. 1169–1214.
[52] D.W. Strook, *Lectures on Topics in Stochastic Differential Equations*, TATA Institute of Fundamental Research, Bombay, India, 1982.
[53] D. Talay and L. Tubaro, *Expansion of the global error for numerical schemes solving stochastic differential equations*, Stochastic Anal. Appl., 8 (1990), pp. 483–509.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.