# A HIGH-ORDER RECOMBINATION ALGORITHM FOR WEAK APPROXIMATION OF STOCHASTIC DIFFERENTIAL EQUATIONS

SYOITI NINOMIYA\* AND YUJI SHINOZAKI†

ABSTRACT. This paper presents an algorithm for applying the high-order recombination method, originally introduced by Lyons and Litterer in "High-order recombination and an application to cubature on Wiener space" (Ann. Appl. Probab. 22(4):1301–1327, 2012), to practical problems in mathematical finance. A refined error analysis is provided, yielding a sharper condition for space partitioning. Based on this condition, a computationally feasible recursive partitioning algorithm is developed. Numerical examples are also included, demonstrating that the proposed algorithm effectively avoids the explosive growth in the cardinality of the support required to achieve high-order approximations.

#### 1. Introduction

In [16, 17], it is shown that the probability measure arising in weak approximations of a stochastic differential equation (SDE) can avoid exponential growth in the cardinality of its support with respect to accuracy, by means of a method that transforms the probability measure of a given finite probability space into one with smaller support, while preserving moments up to a certain degree. This method is the subject of this paper and is referred to as the high-order recombination method. The aim of the paper is to present a theoretically guaranteed and practically feasible algorithm for the high-order recombination method, based on a refined error analysis and a sharp condition for space partitioning, and to demonstrate its applicability to real problems in mathematical finance while maintaining the theoretically predicted performance.

1.1. Weak approximation problem of SDEs. Let  $(\Omega, \mathcal{F}, P)$  be a probability space, and let  $(B^1(t), \dots, B^d(t))$  denote the d-dimensional standard Brownian motion. For notational convenience, we define  $B^0(t) = t$ . We denote by  $C_b^{\infty}(\mathbb{R}^N; \mathbb{R}^N)$  the set of  $\mathbb{R}^N$ -valued smooth functions on  $\mathbb{R}^N$  whose derivatives of all orders are bounded. Let  $I_N$  denote the identity map on  $\mathbb{R}^N$ . Let X(t, x) be an

<sup>\*</sup>Department of Mathematics, School of Science, Institute of Science Tokyo 2-12-1 Ookayama, Meguro-ku, Tokyo 152-8551 JAPAN

<sup>&</sup>lt;sup>†</sup>School of Business Administration, Hitotsubashi University Business School, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8439, Japan

 $<sup>\</sup>textit{E-mail addresses}: \verb| ninomiya@math.titech.ac.jp, yshinozaki@hub.hit-u.ac.jp.|$ 

<sup>2020</sup> Mathematics Subject Classification. 91G60, 60H35, 65C20, 65C30, 65C05, 68U20, 62P05. Key words and phrases. computational finance, stochastic differential equation, high-order discretisation method, cubature on Wiener space, weak approximation, tree-based method, Heston model.

 $\mathbb{R}^N$ -valued diffusion process defined as the solution to the Stratonovich-form SDE:

$$X(t,x) = x + \sum_{i=0}^{d} \int_{0}^{t} V_{i} I_{N}(X(s,x)) \circ dB^{i}(s),$$
(1)

where  $x \in \mathbb{R}^N$  and  $V_0, \dots, V_d$  are tangent vector fields on  $\mathbb{R}^N$ , each having coefficients in  $C_b^{\infty}(\mathbb{R}^N; \mathbb{R}^N)$ ; that is,  $V_i I_N = \begin{pmatrix} (V_i I_N)^1 & \dots & (V_i I_N)^N \end{pmatrix} \in C_b^{\infty}(\mathbb{R}^N; \mathbb{R}^N)$ ,

and the identity 
$$V_i g(y) = \sum_{j=1}^{N} (V_i I_N)^j(x) \frac{\partial g}{\partial x_j}(y)$$
 holds for all  $i \in \{0, 1, \dots, d\}$ ,

 $g \in C^{\infty}(\mathbb{R}^N)$  and  $y \in \mathbb{R}^N$ . Furthermore, the SDE (1) can be equivalently written in Itô form:

$$X(t,x) = x + \int_0^t \tilde{V}_0(X(s,x)) dt + \sum_{i=1}^d \int_0^t V_i I_N(X(s,x)) dB^i(s),$$
 (2)

where

$$(\tilde{V}_0 I_N)^k = (V_0 I_N)^k + \frac{1}{2} \sum_{i=1}^d V_i (V_i I_N)^k$$
(3)

for all  $k \in \{1, ..., N\}$ . This identity is known as the Itô–Stratonovich transformation.

In this paper, we address the weak approximation problem for SDEs; that is, the numerical evaluation of E[f(X(T,x))], where X(t,x) denotes the value of an N-dimensional diffusion process at time t and f is a  $\mathbb{R}$ -valued function defined on  $\mathbb{R}^N$  with appropriate regularity. This type of computation is crucial in many areas of applied science and engineering—for example, in the pricing and hedging of financial derivatives. There are two main approaches to solving this problem numerically: one based on partial differential equations and the other on simulations. We focus on the latter approach, particularly on a so-called tree-based simulation method, which is described in a later subsection.

- 1.2. **High-order cubature on Wiener space method.** This section adopts high-order cubature on Wiener space methods [18, 22]. In these methods, one approximates the infinite-dimensional Wiener measure with a finite-dimensional measure on a given partition of the time interval [0,T]. We denote this partition by  $\Delta = \{0 = t_0 < \cdots < t_n = T\}$ , and define  $\sharp \Delta = n$  and  $s_j = t_j t_{j-1}$ .
- 1.2.1. **cub3**: a cubature of degree 3. As a simple illustrative example of a cubature measure on Wiener space, we outline a degree 3 cubature, called **cub3**, which is proposed in [18, 24].

$$X_{t_0}^{(\operatorname{cub3},\Delta)}(\eta) = x,$$

$$X_{\lambda t_{j+1}+(1-\lambda)t_j}^{(\operatorname{cub3},\Delta)}(\eta) = \exp\left(\lambda \left(s_{j+1}V_0 + \sum_{i=1}^d \sqrt{s_{j+1}}\eta_{i,j+1}V_i\right)\right) X_{t_j}^{(\operatorname{cub3},\Delta)}(\eta)$$

$$(4)$$

where  $0 \le \lambda \le 1$ ,  $\exp(V)x$  denotes the solution at time 1 of the ordinary differential equation (ODE)

$$\frac{\mathrm{d}z_t}{\mathrm{d}t} = VI_N(z_t), \quad z_0 = x,\tag{5}$$

for a tangent vector field V on  $\mathbb{R}^N$ , and  $\eta = \{\eta_{i,j}\}_{\substack{i=1,\dots,d\\j=1,\dots,\sharp}\Delta}$  denotes a  $d\sharp\Delta$ -dimensional vector consisting of independent and identically distributed (i.i.d.) random variables satisfying  $P(\eta_{i,j}=\pm 1)=1/2$ . We also remark that for any  $s\in\mathbb{R}$ , the expression  $\exp(sV)(x)$  is well-defined by virtue of the chain rule and solves the ODE (5).

We then obtain a 3rd-order cubature measure on Wiener space  $\mu^{(\text{cub3},\Delta)}$  supported on  $\{\omega_{\eta} \mid \eta \in \{-1,1\}^{d\sharp \Delta}\}$  where

$$\omega_{\eta} = \left(X_s^{(\operatorname{cub}3,\Delta)}(\eta)\right)_{s \in [0,T]}$$

and each path  $\omega_{\eta}$  has an equal weight of  $2^{-d\sharp\Delta}$ . Therefore, the computational cost of evaluating

$$\begin{split} \mu_T^{(\text{cub3},\Delta)}(f) &= E\left[f\left(X_T^{(\text{cub3},\Delta)}\right)\right] \\ &= \sum_{\eta \in \{-1,1\}^{d\sharp \Delta}} 2^{-d\sharp \Delta} f\left(\left.\omega_\eta\right|_{s=T}\right) \end{split}$$

grows exponentially with  $\sharp \Delta$ .

1.2.2. Tree-based simulation. Discretisation methods similar to the **cub3** method, which proceed by drawing a direct product of discrete random variables  $\eta$ , are referred to as tree based simulations. This terminology stems from the structural interpretation that the support  $\left\{ \left. \omega_{\eta} \right| \eta \in \{-1,1\}^{d\sharp \Delta} \right\}$  of the measure can be viewed as a tree, with the root  $\left. \omega_{\eta} \right|_{s=0}$  and the leaves  $\left\{ \left. \omega_{\eta} \right|_{s=T} \right| \eta \in \{-1,1\}^{d\sharp \Delta} \right\}$ .

We now introduce notation for tree-based simulations. Let  $\{X^{(\mathrm{Alg},\Delta)}(t_i)\}_{i=0}^{\sharp\Delta}$  denote the discretised process obtained by a discretisation method  $\mathbf{Alg}$ , based on discrete random variables with respect to the partition  $\Delta$  of the interval [0,T]. We denote by  $\mu_{t_i}^{(\mathrm{Alg},\Delta)}$  the probability measure on  $\mathbb{R}^N$  induced by  $X^{(\mathrm{Alg},\Delta)}(t_i)$  and write it as

$$\mu_{t_i}^{(\text{Alg},\Delta)} = \sum_{j=1}^{m_i} w_j^{(i)} \delta_{p_j^{(i)}}, \tag{6}$$

where  $\left\{\left(p_j^{(i)}, w_j^{(i)}\right)\right\}_{j=1}^{m_i}$  is the set of pairs consisting of points  $p_j^{(i)} \in \mathbb{R}^N$  and the corresponding weights  $w_j^{(i)}$ . Here,  $\delta_x$  denotes the Dirac measure at  $x \in \mathbb{R}^N$ . We also denote the transition probability as

$$w^{(i)}(k,j) = P\left(X^{(\text{Alg},\Delta)}(t_{i+1}) = p_j^{(i+1)} \mid X^{(\text{Alg},\Delta)}(t_i) = p_k^{(i)}\right).$$
 (7)

We then define the one-step forward operator on the measure space as

$$\hat{Q}_{(s_{i+1})}^{(Alg,\Delta)} \left( \sum_{j=1}^{m_i} \alpha_j \delta_{p_j^{(i)}} \right) = \sum_{k=1}^{m_i} \sum_{j=1}^{m_{i+1}} \alpha_k w^{(i)}(k,j) \delta_{p_j^{(i+1)}}, \tag{8}$$

where  $\alpha_i \in \mathbb{R}$ . We also define

$$w_j^{(i+1)} = \begin{cases} P\left(X^{(\text{Alg},\Delta)}(t_1) = p_j^{(1)}\right) & \text{if } i = 0\\ \sum_{k=1}^{m_i} w^{(i)}(k,j)w_j^{(i)} & \text{if } 1 \le i \le \sharp \Delta - 1. \end{cases}$$
(9)

Finally, we obtain the measure  $\mu_T^{(Alg,\Delta)} = \hat{Q}_{(s_{\sharp\Delta})}^{(Alg,\Delta)} \circ \cdots \circ \hat{Q}_{(s_1)}^{(Alg,\Delta)} (\delta_x)$ . The process of approximating E[f(X(T,x))] by the expectation

$$\mu_T^{(Alg,\Delta)}(f) = \sum_{j=1}^{m_{\sharp\Delta}} w_j^{(\sharp\Delta)} f\left(p_j^{(\sharp\Delta)}\right), \tag{10}$$

is referred to as a weak approximation by tree-based simulation. The **cub3** method  $\mu_T^{(\text{cub3},\Delta)}$ , introduced in subsection 1.2.1, is one of the simplest examples of tree-based simulation methods. The Ninomiya–Victoir method ([24])  $\mu_T^{(\text{NV},\Delta)}$ , which serves as the subject of the numerical experiments in this paper, is introduced in subsection 2.3.

1.2.3. Order of discretisation. The order of a discretisation method is defined as follows.

**Definition 1.** We call **Alg** a discretisation of order p or a pth order discretisation if there exists a constant  $C_p > 0$  such that for all  $n \in \mathbb{N}$  there exists a partition  $\Delta$  of [0,T] with  $\sharp \Delta = n$  satisfying the inequality

$$\left| E\left[ f\left( X(t,x) \right) \right] - E\left[ f\left( X^{(\mathrm{Alg},\Delta)}\left( T \right) \right) \right] \right| < C_p(\sharp \Delta)^{-p}. \tag{11}$$

In what follows, we refer to discretisation methods of order 2 or higher as high-order discretisation methods. Some specific high-order discretisation method based on cubature on Wiener space are proposed in [24, 20, 26], and the one used in this paper is introduced in subsection 2.3. In addition, research on the construction of high-order cubature measures has explored various avenues, including simple algebraic construction [8], randomised construction [9], and constructions based on Hopf algebras and unshuffle expansions [6]. For ODEs, discretisation methods of fixed order when linearly combined using weights that may include negative values can yield methods of higher order. This approach is commonly referred to as extrapolation. In the context of weak approximation for SDEs, the extrapolation of the Euler–Maruyama method is studied in [1]. Furthermore, as demonstrated in [7, 25], extrapolation methods can be combined with the method of [24] to construct discretisations of arbitrary order.

1.3. **High-order recombination.** When implementing high-order cubature measures on Wiener space, one often encounters the problem of exponential growth in the cardinality of the supp  $\left(\mu_T^{(\mathrm{Alg},\Delta)}\right)$  as the number of time partitions  $\sharp\Delta$  increases. For example, as seen in subsection 1.2.1 the cardinality of the support of  $\mu_T^{(\mathrm{cub3},\Delta)}$  is  $2^{d\sharp\Delta}$  and that of the support of  $\mu_T^{(\mathrm{NV},\Delta)}$  is  $9^{d\sharp\Delta}$ , where  $\mathbf{NV}$  refers to the second-order discretisation method introduced in subsection 2.3. In what follows, we refer to this issue as the support explosion problem.

At present the most widely used approach to overcome this problem is partial sampling such as Monte Carlo and quasi-Monte Carlo methods. Among these the technique known as TBBA is particularly effective and well suited to the tree-based simulation methods [2, 21, 19]. Furthermore in [3, 4, 22], tree-based simulation with TBBA is successfully applied to the numerical calculation of forward-backward SDEs.

High-order recombination is proposed in [17] as an alternative approach to resolving the support explosion problem. There the authors show that the cardinality

of the support can be kept polynomial in the required approximation accuracy. Several applications of high-order recombination are presented in [15]. In [27] a number of algorithms for obtaining reduced measures are discussed. The notion of reduced measure plays a significant role in the construction of recombination measures, and will be introduced in the subsequent subsection 1.3.1.

This subsection presents an overview of the procedure for constructing what we call high-order recombination measures, which form the main subject of this paper.

1.3.1. Reduced measure. We first introduce the concept of a reduced measure, which is a measure with reduced support preserving relations defined by a set of test functions. Let  $\mu$  be a discrete probability measure on a space  $\tilde{\Omega}$ , and let  $G = \{g_1, g_2, \ldots, g_{\mathcal{M}_G}\}$  be a set of test functions, where  $\mathcal{M}_G$  denotes the number of test functions and each  $g_i: \tilde{\Omega} \to \mathbb{R}$  is a measurable function.

**Definition 2.** (Reduced measure) A discrete probability measure  $\tilde{\mu}$  is said to be a reduced measure from  $\mu$  with respect to G if it satisfies the following three conditions:

- (i) supp  $(\tilde{\mu}) \subset \text{supp }(\mu)$ ,
- (ii)  $\int g(x) \,\tilde{\mu}(\mathrm{d}x) = \int g(x) \,\mu(\mathrm{d}x)$  for all  $g \in G$ ,
- (iii) card (supp  $(\tilde{\mu})$ )  $\leq \mathcal{M}_G + 1$ ,

where  $\operatorname{card}(S)$  denotes the cardinality of the set S.  $\operatorname{red}_G(\mu)$  denotes the set of reduced measures from  $\mu$  with respect to G.

In [17], the existence of a reduced measure is shown for a finite Borel measure  $\mu$  on a Polish space  $\tilde{\Omega}$ .

In this paper for discretisation methods of order m and with  $\tilde{\Omega} = \mathbb{R}^N$ , we consider the set of test functions G to consist of distinct homogeneous polynomials of degree at most m and we assume that  $g_1(x) = 1$  and  $g_2(0) = g_3(0) = \cdots = g_{\mathcal{M}_G}(0) = 0$ .

Note that condition (ii), determined by  $g_1(x) = 1$ , ensures that the reduced measure  $\tilde{\mu}$  is also a probability measure.

1.3.2. High-order recombination measures and their construction procedures. The high-order recombination measures  $\left\{\mu_{t_i}^{(\mathrm{Alg},\Delta,\mathrm{Rec})}\right\}_{i=0}^{\sharp\Delta}$  in tree-based simulation with respect to  $(\mathbf{Alg},\Delta)$  are constructed recursively as follows:

$$\mu_{t_0}^{(\text{Alg},\Delta,\text{Rec})} = \delta_{x_0},$$

$$\nu^{(i)} = \hat{Q}_{(s_i)}^{(\text{Alg},\Delta)} \left( \mu_{t_{i-1}}^{(\text{Alg},\Delta,\text{Rec})} \right),$$

$$\mu_{t_i}^{(\text{Alg},\Delta,\text{Rec})} = \sum_{k=1}^{l_i} \nu^{(i)} \left( U_k^{(i)} \right) \tilde{\nu}_k^{(i)},$$
(12)

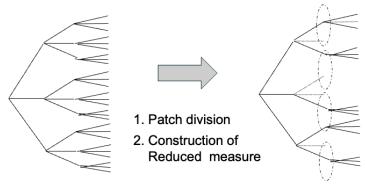
where  $\left\{U_k^{(i)}\right\}_{k=1}^{l_i}$  is a set of subsets of supp  $\left(\nu^{(i)}\right)$  satisfying

$$\bigcup_{k=1}^{l_i} U_k^{(i)} = \operatorname{supp}\left(\nu^{(i)}\right) \quad \text{and} \quad U_j^{(i)} \cap U_k^{(i)} = \emptyset \quad \text{for all } j \neq k, \tag{13}$$

and where

$$\tilde{\nu}_k^{(i)} \in \operatorname{red}_G\left(\left.\nu^{(i)}\right|_{U_k^{(i)}}\right) \quad \text{for } 1 \le k \le l_i,$$
(14)

Figure 1. Illustration of recombination method



 Toy example of Reduced measure preserving up to the 2nd-order moments

preserving up to the 2nd-order moments 
$$\begin{pmatrix} G = \{1, x, y, xy, x^2, y^2\} \\ E[X] = \hat{E}[\hat{X}], E[Y] = \hat{E}[\hat{Y}], E[X^2] = \hat{E}[\hat{X}^2], \\ E[Y^2] = \hat{E}[\hat{Y}^2], E[XY] = \hat{E}[\hat{X}\hat{Y}] \end{pmatrix}$$

$$(X, Y) = \begin{cases} (0,0) & (p_1 = 1/8) \\ (1,0) & (p_2 = 1/8) \\ (0,1) & (p_3 = 1/8) \\ (1,1) & (p_4 = 1/8) \\ (2,0) & (p_5 = 1/8) \\ (2,1) & (p_6 = 1/8) \\ (2,2) & (p_7 = 1/8) \\ (1,2) & (p_8 = 1/8) \end{cases}$$

$$(\hat{X}, \hat{Y}) = \begin{cases} (0,0) & (\hat{p}_1 = 1/6) \\ (1,0) & (\hat{p}_2 = 0) \rightarrow \text{Eliminated} \\ (0,1) & (\hat{p}_3 = 1/12) \\ (1,1) & (\hat{p}_4 = 7/24) \\ (2,0) & (\hat{p}_5 = 5/24) \\ (2,1) & (\hat{p}_6 = 0) \rightarrow \text{Eliminated} \\ (2,2) & (\hat{p}_7 = 1/6) \\ (1,2) & (\hat{p}_8 = 1/12) \end{cases}$$

$$\nu^{(i)}\Big|_{U_k^{(i)}} = \frac{1}{\nu^{(i)} \left(U_k^{(i)}\right)} \sum_{p_j^{(i)} \in U_k^{(i)}} \hat{w}_j^{(i)} \delta_{p_j^{(i)}},\tag{15}$$

which is the probability measure on  $U_k^{(i)}$  induced by  $\nu^{(i)} = \sum_{j=1}^{m_i} \hat{w}_j^{(i)} \delta_{p_j^{(i)}}$ . Henceforth a set  $\{U_k\}_{k=1}^l$  satisfying

$$\bigcup_{k=1}^{l} U_k = \operatorname{supp}(\nu^{(i)}) \quad \text{and} \quad U_j \cap U_k = \emptyset \quad \text{for all } j \neq k$$
 (16)

is referred to as a patch division of  $\operatorname{supp}(\nu^{(i)}).$  Note that this procedure is interwoven with the tree-based simulation and is carried out simultaneously. Also note that the total weight assigned to each patch remains invariant throughout the recombination procedure. We therefore refer to this as the weight of the patch  $U_k^{(i)}$  denoted by wt  $\left(U_k^{(i)}\right)$ . The following identity summarises the quantities introduced above:

$$\operatorname{wt}\left(U_{k}^{(i)}\right) = \nu^{(i)}\left(U_{k}^{(i)}\right) = \mu_{t_{i}}^{(\operatorname{Alg}, \Delta, \operatorname{Rec})}\left(U_{k}^{(i)}\right). \tag{17}$$

A simplified illustration of these procedures can be found in Figure 1. It is shown in [17] that high-order recombination can be performed in such a way that the order

of discretisation is preserved by choosing appropriate patch radii

$$rad(U_i) = \max\{\|p_k - p_{k'}\| \mid p_k, p_{k'} \in U_i\}.$$
(18)

Numerical examples of the high-order recombination method applied to practical problems appear only in [23]. It is shown numerically in that paper that, even when ad hoc patch division is employed—which has not been proved to avoid the support explosion problem—the recombination method can still be effective for certain problems.

This paper provides a detailed explanation of the algorithms corresponding to the aforementioned procedure. Specifically:

- The explicit construction of  $\hat{Q}_{(s_i)}^{(\mathrm{NV},\Delta)}$  is presented in subsection 2.3.
- An algorithm for the patch division of  $\operatorname{supp}(\nu^{(i)})$  into  $\left\{U_k^{(i)}\right\}_{k=1}^{l_i}$  is proposed in subsection 3.2.
- in subsection 3.2.

   The algorithm for finding  $\tilde{\nu}_k^{(i)} \in \operatorname{red}_G\left(\left.\nu^{(i)}\right|_{U_k^{(i)}}\right)$ —which is identical to the one used in [23]—is provided in Appendix 1.
- 1.4. Contribution of this paper. This paper presents a novel recursive patch division algorithm for high-order recombination that addresses the support explosion problem, thereby making it practically feasible for real-world applications in mathematical finance. The contributions of this paper are summarized as follows:
- An improved version of the error estimation in [17] is presented, which incorporates the patch weights wt  $\left(U_k^{(i)}\right)$ .
   A recursive algorithm is proposed that divides the support set into patches, tak-
- A recursive algorithm is proposed that divides the support set into patches, taking into account both the radii and the weights of the patches. The algorithm includes a control parameter that can be varied to change the degree of recombination. Its recursive structure enables efficient and adaptive patch division based on the refined error evaluation.
- An example is provided demonstrating the application of the algorithm to realistic financial option pricing problems. In this example the algorithm successfully avoids the support explosion problem and illustrates how adjusting the control parameter affects the degree of recombination.

The remainder of this paper is organised as follows. Section 2 introduces the necessary notation and preliminary results on high-order discretisation schemes and high-order recombination. Section 3 presents the refined error evaluation and the novel recursive patch division algorithm, which constitute the main contributions of this paper. Section 4 provides numerical results that demonstrate the performance of the proposed algorithm.

# 2. Preliminary

This section introduces the notation and prior work on high-order discretisation schemes and the recombination algorithm.

2.1. Free Lie algebras. Let  $A = \{0, 1, \dots, d\}$  and define  $A^+ = \bigcup_{k=1}^{\infty} A^k$  as the set of all words consisting of elements of A, and define  $A^* = \{\emptyset\} \cup A^+$ .  $A^*$  becomes a semigroup with identity  $\emptyset$  when endowed with the concatenating product '·', that

is, for arbitrary elements  $(a_1, \ldots, a_m)$  and  $(b_1, \ldots, b_n)$  in  $A^*$ ,

$$(a_1, \ldots, a_m) \cdot (b_1, \ldots, b_n) = (a_1, \ldots, a_m, b_1, \ldots, b_n)$$

and

$$\emptyset \cdot (a_1, \dots, a_m) = (a_1, \dots, a_m) \cdot \emptyset = (a_1, \dots, a_m).$$

To simplify the notation, we identify  $(a_1, a_2, \ldots, a_m) \in A^*$  with  $a_1 \cdot a_2 \cdots a_m$ . If there is no risk of confusion, we also omit the symbol '·'. That is, we prefer to write  $a_1 a_2 \ldots a_m$  instead of  $(a_1, a_2, \ldots, a_m)$  or  $a_1 \cdot a_2 \cdots a_m$ . For  $\alpha = a_1 a_2 \ldots a_k \in A^*$ , we define the length of  $\alpha$  by  $|\alpha| = k$ , and set  $|\emptyset| = 0$ . We also define the order of  $\alpha$  as

$$\|\alpha\| = |\alpha| + \operatorname{card}(\{1 \le i \le k \mid a_i = 0\}).$$

Let  $A_1^* = A^* \setminus \{\emptyset, 0\}$  and define

$$A^*(m) = \{ \alpha \in A^* \mid ||\alpha|| \le m \} \text{ and } A_1^*(m) = \{ \alpha \in A_1^* \mid ||\alpha|| \le m \}$$

for  $m \in \mathbb{N}$ .

Let  $\mathbb{R}\langle A \rangle$  be the free  $\mathbb{R}$ -algebra with basis  $A^*$  and  $\mathbb{R}\langle \langle A \rangle \rangle$  be its completion, that is

$$\mathbb{R}\langle A \rangle = \left\{ \sum_{k=1}^{l} r_k \alpha_k \mid l \in \mathbb{N}, r_k \in \mathbb{R}, \alpha_k \in A^* \right\}$$

and

$$\mathbb{R}\langle\langle A\rangle\rangle = \left\{\sum_{\alpha \in A^*} r_\alpha \alpha \;\middle|\; r_\alpha \in \mathbb{R}\right\}.$$

The latter is the set of all  $\mathbb{R}$ -coefficient formal series with basis  $A^*$ . The multiplication in  $\mathbb{R}\langle\langle A\rangle\rangle$  is defined by extending the product in  $\mathbb{R}\langle A\rangle$  continuously. More precisely, for such  $R,S\in\mathbb{R}\langle\langle A\rangle\rangle$  written as

$$R = \sum_{\alpha \in A^*} r_{\alpha} \alpha, \quad S = \sum_{\beta \in A^*} s_{\beta} \beta,$$

their product is given by

$$RS = \sum_{\gamma \in A^*} \left( \sum_{\substack{\alpha, \beta \in A^* \\ \alpha\beta = \gamma}} r_{\alpha} s_{\beta} \right) \gamma.$$

This product makes  $\mathbb{R}\langle\langle A\rangle\rangle$  a unital non-commutative algebra over  $\mathbb{R}$ , in which  $\mathbb{R}\langle A\rangle$  is a subalgebra. We also define  $\|R\|_2 = (\langle R, R\rangle)^{1/2}$ . The Lie bracket is defined by [R, S] = RS - SR as usual.

 $\mathcal{L}_{\mathbb{R}}(A)$  denotes the set of Lie polynomials over A, that is, the smallest  $\mathbb{R}$ -submodule of  $\mathbb{R}\langle A \rangle$  that contains A and is closed under the Lie bracket [,].

Let  $\Phi$  be the homomorphism from  $\mathbb{R}\langle A\rangle$  to the  $\mathbb{R}$ -algebra of smooth differential operators on  $\mathbb{R}^N$ , defined by

$$\Phi(\emptyset) = \text{Id}, 
\Phi(\alpha) = V_{a_1} V_{a_2} \cdots V_{a_k} \quad \text{for } \alpha = a_1 a_2 \dots a_k \in A^+.$$
(19)

Here, Id denotes the identity operator.

We remark that  $\Phi$  canonically induces a Lie algebra homomorphism from  $\mathcal{L}_{\mathbb{R}}(A)$  to the Lie algebra of vector fields generated by  $\{V_0, \ldots, V_d\}$ . We denote  $V_R = \Phi(R)$ 

for  $R \in \mathbb{R}\langle A \rangle$ , and define  $[\alpha] = [[[\ldots [[a_1, a_2], a_3], \ldots], a_{k-1}], a_k]$  for  $\alpha = a_1 a_2 \ldots a_k \in A^*$ , that is the right-nested Lie bracket of the letters in  $\alpha$ .

We remark that the following approximation holds for  $R \in \mathcal{L}_{\mathbb{R}}(A)$  by virtue of the Taylor expansion formula:

$$f(\exp(t\Phi(R))(x)) = \Phi(j_m(\exp(tR))) f(x) + O(t^{m+1}).$$
 (20)

2.2. **High-order discretisation scheme.** In this subsection, we introduce high-order discretisation schemes for SDEs and related results.

Let  $C_{\text{Lip}}(\mathbb{R}^N;\mathbb{R})$  denote the space of Lipschitz continuous functions on  $\mathbb{R}^N$ , and let  $\|\cdot\|_{\text{Lip}}$  denote the Lipschitz norm defined by

$$||f||_{\text{Lip}} = \sup_{x,y \in \mathbb{R}^N, x \neq y} \frac{|f(x) - f(y)|}{|x - y|},$$

and let  $\|\cdot\|_{\infty}$  denote the uniform norm defined by

$$||f||_{\infty} = \sup_{x \in \mathbb{R}^N} |f(x)|$$

for  $f \in C_{\text{Lip}}(\mathbb{R}^N; \mathbb{R})$ .

**Definition 3** ( $\{P_t^X\}_{t\geq 0}$ ). Let X(t,x) be the  $\mathbb{R}^N$ -valued diffusion process defined by (1). For  $f \in C_{\text{Lip}}(\mathbb{R}^N; \mathbb{R})$  and  $t \geq 0$ , define

$$(P_t^X f)(x) = E[f(X(t,x))]$$

and obtain a semigroup of linear operators  $\{P_t^X\}_{t\geq 0}$  on  $C_{\operatorname{Lip}}(\mathbb{R}^N;\mathbb{R})$ .

We begin by presenting an approximation theorem for the high-order discretisation scheme based on [11, 12, 13].

**Definition 4** (UFG condition [12]). Let  $\ell$  be an integer. The vector fields  $V_0, V_1, \ldots, V_d$  are said to satisfy the UFG condition if for all  $\alpha \in A_1^*$  there exists a family

$$\{\varphi_{\alpha,\beta} \mid \beta \in A_1^*(\ell)\} \subset C_b^{\infty}(\mathbb{R}^N; \mathbb{R})$$

such that

$$V_{[\alpha]} = \sum_{\beta} \varphi_{\alpha,\beta} V_{[\beta]}.$$

Hereafter, the integer  $\ell$  in the definition above is referred to as the degree of the UFG condition.

We next introduce a theorem from [12], which concerns the smoothness of the semigroup  $\{P_t^X\}_{t\geq 0}$  in certain directions.

**Theorem 1.** Assume that the vector fields  $V_0, V_1, \ldots, V_d$  satisfy the UFG condition and let X be the diffusion process defined in (1). Let  $\alpha_1, \alpha_2, \ldots, \alpha_k \in A_1^*$ . Then there exists a constant C > 0 such that

$$\left\| V_{[\alpha_1]} \cdots V_{[\alpha_{k'}]} P_t^X V_{[\alpha_{k'+1}]} \cdots V_{[\alpha_k]} f \right\|_{L^p(dx)} \le C t^{(\|\alpha_1\| + \dots + \|\alpha_k\|)/2} \|f\|_{L^p(dx)}.$$

A condition on approximation operators is now introduced.

**Definition 5** (m-similar operator). A Markov operator  $\tilde{Q}_{(s)}$  is said to be an m-similar operator if there exist  $\mathcal{L}_{\mathbb{R}}(A)$ -valued random variables  $L_{(s)}^{(1)}$ ,  $L_{(s)}^{(2)}$ , ..., and  $L_{(s)}^{(K)}$  such that

$$E\left[j_m\left(\exp\left(L_{(s)}^{(1)}\right)\exp\left(L_{(s)}^{(2)}\right)\cdots\exp\left(L_{(s)}^{(K)}\right)\right)\right] = E\left[j_m\left(\sum_{\alpha\in A^*}B^{\circ\alpha}(s)\,\alpha\right)\right],$$

$$E\left[\left\|j_m\left(L_{(s)}^{(k)}\right)\right\|_2^n\right] < \infty, \quad \left\langle L_{(s)}^{(k)}, (0) \right\rangle = 1 \quad \text{for all } n \in \mathbb{N}, \ k \in \{1, 2, \dots, K\},$$

and

$$\begin{split} \left( \tilde{Q}_{(s)} f \right) (x) \\ &= E \left[ f \left( \left( \exp \left( \Phi \left( L_{(s)}^{(1)} \right) \right) \exp \left( \Phi \left( L_{(s)}^{(2)} \right) \right) \cdots \exp \left( \Phi \left( L_{(s)}^{(K)} \right) \right) \right) (x) \right) \right]. \end{split}$$

The term  $B^{\circ \alpha}(s)$  appearing above is defined for  $\alpha \in A^*$  as follows:

$$B^{\circ \alpha}(t) = \begin{cases} 1 & \text{if } \alpha = \emptyset, \\ \int_0^t B^{\alpha'}(s) \circ dB^a(s) & \text{if } \alpha = \alpha' a, \ \alpha' \in A^*, \ a \in A. \end{cases}$$

The m-similar operator may be intuitively understood as an approximation of E[f(X(T,x))] up to the mth order based on the stochastic Taylor expansion of the solution to the SDE (1). We refer to m as the degree of the discretisation method. Note that this definition is a slight generalisation of the original definition given in [11, 12, 13]. The following theorem, due to Kusuoka [13], forms the foundation of high-order discretisation schemes.

**Theorem 2.** Assume that the vector fields in the SDE (1) satisfy the UFG condition, that X is the diffusion process defined by (1), and that  $Q_{(s)}$  is an m-similar operator. Then there exist positive constants C and C' depending only on m and the vector fields  $V_0, V_1, \ldots, V_d$  such that

$$||P_s^X f - Q_{(s)} f||_{\infty} \le C s^{(m+1)/2} ||f||_{\text{Lip}},$$

and

$$\left\|Q_{(s)}f\right\|_{\infty} \leq \exp(C's)\|f\|_{\infty}$$

for all  $f \in C_{\text{Lip}}(\mathbb{R}^N; \mathbb{R})$ .

Consequently, the total discretisation error is bounded as follows. Let T > 0 be fixed, and consider a partition  $\Delta$  of the interval [0,T] with  $\sharp \Delta = n$ , given by  $t_0 = 0 < t_1 < \cdots < t_n = T$ . Then there exists a constant C'', depending only on m, the vector fields  $V_0, V_1, \ldots, V_d$ , and T such that

$$\|P_T^X f - Q_{(s_n)} \circ Q_{(s_{n-1})} \circ \cdots \circ Q_{(s_1)} f\|_{\infty}$$

$$\leq C'' \|f\|_{\text{Lip}} \left( s_n^{1/2} + \sum_{j=m}^{m+1} \sum_{i=1}^{n-1} \frac{s_i^{(j+1)/2}}{(T - t_i)^{j/2}} \right), \quad (21)$$

where  $s_k = t_k - t_{k-1}$ .

**Remark 1** (The regularity assumption on f). In the original work of Kusuoka [11, 13], Theorem 2 is proved under the assumption that  $f \in C_b^{\infty}(\mathbb{R}^N; \mathbb{R})$ . This assumption can, however, be relaxed to  $f \in C_{\text{Lip}}(\mathbb{R}^N; \mathbb{R})$ , as pointed out in Section 3 of [18].

2.3. **N–V method.** We introduce the N–V method from [24], which is a second-order discretisation scheme. Hereafter, we denote by  $(G_1 \circ G_2)g(x)$  the composition  $(G_1(G_2g))(x)$  of  $G_1$  and  $G_2$ , where  $G_1, G_2 \colon C_{\text{Lip}}(\mathbb{R}^N; \mathbb{R}) \to C_{\text{Lip}}(\mathbb{R}^N; \mathbb{R})$ .

**Theorem 3** ([24, 14]). Let  $\eta_1, \eta_2, \ldots, \eta_d$  be independent standard normal random variables and define

$$Q_{(s)}^{(\text{NV})} f(x) = E \left[ \frac{1}{2} f\left( \exp\left(sV_0\right) \exp\left(\sqrt{s}\eta_1 V_1\right) \cdots \exp\left(\sqrt{s}\eta_d V_d\right) (x) \right) + \frac{1}{2} f\left( \exp\left(\sqrt{s}\eta_d V_d\right) \cdots \exp\left(\sqrt{s}\eta_1 V_1\right) \exp\left(sV_0\right) (x) \right) \right].$$
(22)

Then  $Q_{(s)}^{(\mathrm{NV})}$  are 5-similar operators and the N-V method  $Q_{(T/n)}^{(\mathrm{NV})} \circ Q_{(T/n)}^{(\mathrm{NV})} \circ \cdots \circ Q_{(T/n)}^{(\mathrm{NV})}$  is a second-order discretisation method.

Replacing the independent standard normal random variables  $\{\eta_i\}_{i=1}^d$  in the definition above with the independent discrete random variables  $\{\hat{\eta}_i^{(5\text{th})}\}_{i=1}^d$ , we define the discrete N–V method as in [22]. Each  $\hat{\eta}_i^{(5\text{th})}$  is a discrete random variable on  $(\hat{\Omega}, \hat{\mathcal{F}}, \hat{P})$  defined by

$$\hat{P}\left(\hat{\eta}_i^{(5\text{th})} = \pm\sqrt{3}\right) = \frac{1}{6}, \ \hat{P}\left(\hat{\eta}_i^{(5\text{th})} = 0\right) = \frac{2}{3}.$$
 (23)

We note that the moments of  $\hat{\eta}_i^{(5\text{th})}$  up to the 5th order match those of the standard normal distribution.

As in subsection 1.2.2, we denote a probability measure on  $\mathbb{R}^N$  induced by  $X^{(\mathrm{Alg},\Delta)}\left(t_i\right)$  as  $\mu_{t_i}^{(\mathrm{Alg},\Delta)} = \sum_{j=1}^{m_i} w_j^{(i)} \delta_{p_j^{(i)}}$ , where  $\left\{\left(p_j^{(i)},w_j^{(i)}\right)\right\}_{j=1}^{m_i}$  is the set of pairs consisting of points  $p_j^{(i)}$  in  $\mathbb{R}^N$  and weights  $w_j^{(i)}$  assigned to them. Let  $\{\hat{\eta}_{i,l}^{(5\text{th})}\}_{i=1,\dots,n}^{i=1,\dots,n}$  be a set of independent and identically distributed discrete random variables following the same distribution as  $\hat{\eta}_i^{(5\text{th})}$ . Let  $\phi$  be a function from  $\{\pm\sqrt{3},0\}^d$  to [0,1] defined by

$$\phi(\eta) = \left(\frac{1}{6}\right)^{\operatorname{card}\{i|\eta_i = \pm\sqrt{3}\}} \left(\frac{2}{3}\right)^{\operatorname{card}\{i|\eta_i = 0\}}$$

for  $\eta=(\eta_1,\ldots,\eta_d)\in\{\pm\sqrt{3},0\}^d$ . We also define functions  $y^{(+)}$  and  $y^{(-)}$  from  $\mathbb{R}^N\times\{\pm\sqrt{3},0\}^d\times\mathbb{R}_{>0}$  to  $\mathbb{R}^N$  as

$$y^{(+)}(x,\eta,s) = \exp(sV_0) \exp(\sqrt{s}\eta_1 V_1) \cdots \exp(\sqrt{s}\eta_d V_d)(x),$$
  
$$y^{(-)}(x,\eta,s) = \exp(\sqrt{s}\eta_d V_d) \cdots \exp(\sqrt{s}\eta_1 V_1) \exp(sV_0)(x).$$

Then, the operator  $\hat{Q}_{(s_i)}^{(\mathrm{NV},\Delta)}$  between discrete measures is defined as

$$\hat{Q}_{(s_i)}^{(\text{NV},\Delta)} \left( \sum_{j=1}^{m_i} w_j^{(i)} \delta_{p_j^{(i)}} \right) \\
= \sum_{j=1}^{m_i} \sum_{\eta \in \{+\sqrt{3},0\}^d} w_j^{(i)} \frac{\phi(\eta)}{2} \left( \delta_{y^{(+)}(p_j^{(i)},\eta,s_i)} + \delta_{y^{(-)}(p_j^{(i)},\eta,s_i)} \right).$$
(24)

In this way, we construct a concrete example of  $\hat{Q}^{(\mathrm{Alg},\Delta)}_{(s_i)}$  used in this paper, as discussed in subsection 1.2.2. We then apply the recombination algorithm described in subsection 1.3.2 to this operator and obtain  $\mu_T^{(\mathrm{NV},\Delta,\mathrm{Rec})}$ .

We similarly define  $\hat{Q}_{(s_i)}(\mu)$  for an m-moment similar operator  $Q_{(s)}$ , using discrete random variables  $\hat{\eta}^{(m\text{th})}$  whose moments up to the mth order match those of the standard normal random variables. In this paper, we consider a pth order discretisation algorithm defined by a p-moment similar operator and its associated recombination measure.

Remark 2 (The number of random variables involved in the N-V method). In [24], the authors propose implementing the N-V method using Bernoulli random variables in addition to the standard normal random variables  $\eta_1, \eta_2, \ldots, \eta_d$  in order to halve the number of ODEs to be solved. In contrast, the present paper employs the N-V method without utilising Bernoulli random variables instead calculating the expectation as shown in equation (22). This approach is motivated by the need to minimise the number of support points in the discrete N-V method so as to regulate the overall computational cost, as noted in Remark 3.

**Remark 3** (Comparison between high-order discretisation methods). The N-V method offers the following advantages over other high-order discretisation methods, such as those in [20, 26]. As stated in the original paper [24], it is often more efficiently implementable because analytical solutions to the ODEs are available in certain cases, such as the Heston model [10]. Furthermore, the Feller condition [5] guarantees the positivity of processes discretised by the N-V method.

Another advantage is that, when implementing high-order discretisation methods using discrete random variables, the N-V method is economical in the sense that only d random numbers are required to approximate an SDE driven by d-dimensional Brownian motion. The numbers of random variables involved and the cardinalities of the supports of the resulting measures are summarised in Table 1.

Method	Number of $\{\hat{\eta}\}$ in $\hat{Q}_{(1/n)}$	Cardinality of the support of the measure of $\hat{Q}_{(T/n)} \circ \hat{Q}_{(T/n)} \circ \cdots \circ \hat{Q}_{(T/n)}$
N-V	d	$(2\times 3^d)^n$
N-N [20]	2d	$3^{2dn}$
$Q_{(s)}^{(7,2)}$ $(d=2)$ [26]	7	$5^{7n}$

TABLE 1. Number of random variables and support sizes for  $\hat{X}_{s,x}^{(\text{method})}$ , with  $\sharp \Delta = n$ .

2.4. **High-order recombination: approximation theorem.** In this subsection, we introduce the approximation theorem established in [17]

**Definition 6** (UH condition). The vector fields  $V_0, V_1, \ldots, V_d$  are said to satisfy the uniform Hörmandar (UH) condition if there exists  $\ell \in \mathbb{N}$  such that

$$\inf_{x,y\in\mathbb{R}^N,|y|=1}\left(\sum_{\alpha\in A_1^*(\ell)}\left\langle V_{[\alpha]}(x),y\right\rangle\right)>0$$

**Theorem 4** ([17], Theorem 19). Assume that the vector fields  $V_0, V_1, \ldots, V_d$  satisfy the UH condition. Let **Alg** be an mth-order discretisation, and let  $\Delta$  be a partition of [0,T] as  $0 = t_0 < t_1 < \cdots < t_{\sharp \Delta} = T$ . Let

$$\left\{\mu_{t_i}^{(\mathrm{Alg},\Delta,\mathrm{Rec})}, \left\{U_k^{(i)}\right\}_{k=1}^{l_i}\right\}_{i=0}^{\sharp\Delta}$$

be a set of recombination measures and patch divisions of their supports. Then the following inequality holds:

where  $u_i = \max \left\{ \operatorname{rad} \left( U_k^{(i)} \right) \mid k = 1, \dots, l_i \right\}$  and  $s_i = t_i - t_{i-1}$ .

In the following, we present an example of a condition that satisfies the assumptions of the above theorem and can be used to determine the patch division.

**Example 1** (LL patch condtion [17], Examples 21). Given a time step as

$$t_i = T\left(1 - \left(1 - \frac{j}{n}\right)^{\gamma}\right) \tag{26}$$

for some  $\gamma > 0$ . If we set

$$u_i = \left(\frac{s_i^{m+1}}{(T - t_i)^{m(1-\ell)}}\right)^{1/(2(m+1))}, \tag{27}$$

then we have

$$\|P_T^X f - \mu_T^{(Alg, \Delta, Rec)}(f)\|_{\infty} \le \frac{C_2(T)}{(\sharp \Delta)^{(m-1)/2}} \|f\|_{Lip}.$$
 (28)

In other words, the recombination does not compromise the order of the discretisation error.

We refer to the condition introduced above by Lyons–Litterer as the Lyons–Litterer patch condition, or simply the LL patch condition.

#### 3. Refined error estimation and patch division algorithms

In this section, we present the main results of this paper. First, we refine Theorem 4 and Example 1 by extending the argument in [17], leading to Theorem 5 and Example 2, which gives a sufficient condition such that the patch division satisfies the required bound on the recombination error.

3.1. Refined error estimation. Taking the weight of each patch into account yields Theorem 5 below, which refines Theorem 4 by incorporating not only the radii of the patches but also their weights.

**Theorem 5.** Assume that the vector fields  $V_0, V_1, \ldots, V_d$  satisfy the UH condition. Let **Alg** be an mth-order discretisation, and let  $\Delta$  be a partition of [0,T] as  $0 = t_0 < t_1 < \cdots < t_{\sharp \Delta} = T$ . Let

$$\left\{\mu_{t_i}^{(\mathrm{Alg},\Delta,\mathrm{Rec})}, \left\{U_k^{(i)}\right\}_{k=1}^{l_i}\right\}_{i=0}^{\sharp \Delta}$$

be a collection of recombined measures and patch divisions of their supports. Then, for some Q > 0 and  $s_i = t_i - t_{i-1}$ , the following inequality holds:

$$\left\| P_{T}^{X} f - \mu_{T}^{(Alg, \Delta, Rec)}(f) \right\|_{\infty} \leq \left( C_{1}(T) \left( s_{\sharp \Delta}^{1/2} + \sum_{i=1}^{\sharp \Delta - 1} \sum_{q=m+1}^{K} \frac{s_{i}^{q/2}}{(T - t_{i})^{(q-1)/2}} \right) + C_{2}(T) \sum_{i=1}^{\sharp \Delta - 1} \frac{\sum_{k=1}^{l_{i}} \left( \operatorname{rad} \left( U_{k}^{(i)} \right) \right)^{m+1} \operatorname{wt} \left( U_{k}^{(i)} \right)}{(T - t_{i})^{m/2}} \right) \|\nabla f\|_{\infty}.$$
(29)

Hereafter, we refer to the first term of the right-hand side of (29)

$$C_1(T) \left( s_{\sharp \Delta}^{1/2} + \sum_{i=1}^{\sharp \Delta - 1} \sum_{q=m+1}^K \frac{s_i^{q/2}}{(T - t_i)^{(q-1)/2}} \right) \|\nabla f\|_{\infty}$$

as the discretisation error and the second term

$$C_2(T) \sum_{i=1}^{\sharp \Delta - 1} \frac{\sum_{k=1}^{l_i} \left( \operatorname{rad} \left( U_k^{(i)} \right) \right)^{m+1} \operatorname{wt} \left( U_k^{(i)} \right)}{(T - t_i)^{m/2}} \|\nabla f\|_{\infty}$$

as the recombination error.

*Proof.* Let  $\hat{Q}_{(s)}$  be an *m*-similar operator. According to Theorem 3 of [13] there exists  $Q \in \mathbb{N}$  such that

$$\left\| P_s^X f - \hat{Q}_{(s)} f \right\|_{\infty} \le c_1 \left( s^{(m+1)/2} \|\nabla f\|_{\infty} + \sum_{q=m+1}^K s^{q/2} \|f\|_{V,q} \right)$$
(30)

and

$$\|P_s^X f - \hat{Q}_{(s)} f\|_{\infty} \le c_3 s^{1/2} \|\nabla f\|_{\infty}$$
 (31)

hold for s > 0, where  $\|\cdot\|_{V,q}$  denotes a semi-norm on  $C_b^{\infty}\left(\mathbb{R}^N;\mathbb{R}\right)$  defined by

$$||f||_{V,q} = \sum_{q'=1}^{q} \sum_{\alpha_1 \cdots \alpha_{q'} \in A^*} ||V_{[\alpha_1]} \cdots V_{[\alpha_{q'}]} f||_{\infty}.$$
 (32)

Furthermore Theorem 2 of [12] yields the following estimate:

$$\|P_{T-t}^X f\|_{V,q} \le c_2 \frac{\|\nabla f\|_{\infty}}{(T-t)^{(q-1)/2}}.$$
 (33)

Define  $\nu^{(i)} = \hat{Q}^{(\mathrm{Alg},\Delta)}_{(s_i)} \left(\mu^{(\mathrm{Alg},\Delta,\mathrm{Rec})}_{t_{i-1}}\right)$  as in (12). The total error can be decomposed as follows. Note that the operator  $P^X_T$ , the measure  $\mu^{(\mathrm{Alg},\Delta,\mathrm{Rec})}_T$ , and the intermediate measures  $\mu^{(\mathrm{Alg},\Delta,\mathrm{Rec})}_{t_i}$  and  $\nu^{(i)}$  depend on the initial value  $x_0$ .

$$\sup_{x_{0} \in \mathbb{R}^{N}} \left| P_{T}^{X} f(x_{0}) - \mu_{T}^{(\text{Alg}, \Delta, \text{Rec})}(f) \right| \\
\leq \left\| P_{T}^{X} f - \hat{Q}_{(s_{1})} P_{T-t_{1}} f \right\|_{\infty} \\
+ \sum_{i=1}^{n-1} \left\| \mu_{t_{i}}^{(\text{Alg}, \Delta, \text{Rec})} \left( P_{T-t_{i}}^{X} f \right) - \nu^{(i+1)} \left( P_{T-t_{i+1}}^{X} f \right) \right\|_{\infty} \\
+ \sum_{i=1}^{n-1} \left\| \nu^{(i)} \left( P_{T-t_{i}}^{X} f \right) - \mu_{t_{i}}^{(\text{Alg}, \Delta, \text{Rec})} \left( P_{T-t_{i}}^{X} f \right) \right\|_{\infty} .$$
(34)

The first and second terms of the right-hand side of (34), which correspond to the discretisation error, can be estimated as follows.

$$\begin{aligned} & \left\| P_{T}^{X} f - \mu_{x_{0},\Delta,\mathcal{U}}^{(1)} P_{T-t_{1}}^{X} f \right\|_{\infty} + \sum_{i=1}^{n-1} \left\| \mu_{t_{i}}^{(\mathrm{Alg},\Delta,\mathrm{Rec})} \left( P_{T-t_{i}}^{X} f \right) - \nu^{(i+1)} \left( P_{T-t_{i+1}}^{X} f \right) \right\|_{\infty} \\ & = \left\| P_{s_{1}}^{X} P_{T-t_{1}}^{X} f - \hat{Q}_{(s_{1})} P_{T-t_{1}}^{X} f \right\|_{\infty} + \sum_{i=1}^{n-1} \left\| E_{\tilde{\mu}_{x_{0},\Delta,\mathcal{U}}^{(i)}} \left( P_{s_{i+1}}^{X} - \hat{Q}_{(s_{i+1})} \right) P_{T-t_{i+1}}^{X} f \right\|_{\infty} \\ & \leq \left\| P_{s_{1}}^{X} P_{T-t_{1}}^{X} f - \hat{Q}_{(s_{1})} P_{T-t_{1}}^{X} f \right\|_{\infty} + \sum_{i=1}^{n-1} \left\| \left( P_{s_{i+1}}^{X} - \hat{Q}_{(s_{i+1})} \right) P_{T-t_{i+1}}^{X} f \right\|_{\infty} \end{aligned}$$

Then, (30), (31), (33), yield

$$\begin{aligned} & \left\| P_{T}^{X} f - \hat{Q}_{(s_{1})} \left( P_{T-t_{1}} \right) \right\|_{\infty} + \sum_{i=1}^{n-1} \left\| \mu_{t_{i}}^{(Alg,\Delta,Rec)} \left( P_{T-t_{i}}^{X} f \right) - \nu^{(i+1)} \left( P_{T-t_{i+1}}^{X} f \right) \right\|_{\infty} \\ & \leq \sum_{i=1}^{n-1} c_{1} \left( s_{i}^{(m+1)/2} \left\| \nabla P_{T-t_{i}}^{X} f \right\|_{\infty} + \sum_{q=m+1}^{K} s_{i}^{q/2} \left\| P_{T-t_{i}}^{X} f \right\|_{V,q} \right) + c_{3} s_{n}^{1/2} \| \nabla f \|_{\infty} \\ & \leq \sum_{i=1}^{n-1} c_{1} \left( s_{i}^{(m+1)/2} \left\| \nabla P_{T-t_{i}}^{X} f \right\|_{\infty} + \sum_{q=m+1}^{K} s_{i}^{q/2} \frac{\| \nabla f \|_{\infty}}{(T-t_{i})^{q/2}} \right) + c_{3} s_{n}^{1/2} \| \nabla f \|_{\infty} \\ & \leq C_{1}(T) \left( s_{n}^{1/2} + \sum_{i=1}^{n-1} \sum_{q=m+1}^{K} \frac{s_{i}^{q/2}}{(T-t_{i})^{(q-1)/2}} \right) \| \nabla f \|_{\infty}. \end{aligned}$$

The third term of (34) corresponds to the recombination error and can be bound by using Corollary 14 of [17] under the UH condition as follows. For each patch  $U_k^{(i)}$ , there exist a  $C_{U_k^{(i)}} > 0$  such that

$$\max_{q_1,\dots,q_w\in\{1,\dots,N\}} \sup_{y\in U_k^{(i)}} \left| \frac{\partial}{\partial x_{k_1}} \cdots \frac{\partial}{\partial x_{k_w}} P_t^X f(y) \right|$$

$$\leq C_{U_k^{(i)}} t^{-(w-1)l/2} \sup_{y\in U_k^{(i)}} |\nabla f(y)|_{\infty}. \quad (35)$$

Then by the Taylor expansion there exists a polynomial  $q_{U_k^{(i)}}$  of order r+1 such that

$$\sup_{y \in U_k^{(i)}} \left| P_t^X f(y) - q_{U_k^{(i)}}(y) \right| \\
\leq \operatorname{rad} \left( U_k^{(i)} \right)^{r+1} \max_{q_1 + \dots + q_N = r+1} \left| \frac{\partial^{q_1}}{\partial x_1^{q_1}} \dots \frac{\partial^{q_N}}{\partial x_N^{q_N}} f(y_0) \right| \quad (36)$$

for  $y_0 \in U_k^{(i)}$ . From the property of reduced measure, it holds that

$$\begin{split} &\sum_{i=1}^{n-1} \sup_{x_0 \in \mathbb{R}^N} \left| \nu^{(i)} \left( P_{T-t_i}^X f \right) - \mu_{t_i}^{(\mathrm{Alg}, \Delta, \mathrm{Rec})} \left( P_{T-t_i}^X f \right) \right| \\ &\leq \sum_{i=1}^{n-1} \sup_{x_0 \in \mathbb{R}^N} \sum_{k=1}^{l_i} \mathrm{rad} \left( U_k^{(i)} \right)^{m+1} \left( \max_{k_1 + \dots + k_N = m+1} \nu^{(i)} \middle|_{U_k^{(i)}} \left( \frac{\partial^{k_1}}{\partial x_1^{k_1}} \dots \frac{\partial^{k_N}}{\partial x_N^{k_N}} P_{T-t_i}^X f \right) \\ &\qquad - \max_{k_1 + \dots + k_N = m+1} \mu_{t_i}^{(\mathrm{Alg}, \Delta, \mathrm{Rec})} \middle|_{U_k^{(i)}} \left( \frac{\partial^{k_1}}{\partial x_1^{k_1}} \dots \frac{\partial^{k_N}}{\partial x_N^{k_N}} P_{T-t_i}^X f \right) \right) \\ &\leq \sum_{i=1}^{n-1} \sum_{k=1}^{l_i} \mathrm{rad} \left( U_k^{(i)} \right)^{m+1} \mathrm{wt} \left( U_k^{(i)} \right) \max_{k_1 + \dots + k_N = m+1} \sup_{y \in U_k^{(i)}} \left| \frac{\partial^{k_1}}{\partial x_1^{k_1}} \dots \frac{\partial^{k_N}}{\partial x_N^{k_N}} P_{T-t_i}^X f(y) \right| \end{split}$$

Hence from the inequality (35)

$$\sum_{i=1}^{n-1} \sup_{x_0 \in \mathbb{R}^N} \left| \nu^{(i)} \left( P_{T-t_i}^X f \right) - \mu_{t_i}^{(\text{Alg}, \Delta, \text{Rec})} \left( P_{T-t_i}^X f \right) \right|$$

$$\leq C_2(T) \sum_{i=1}^{n-1} \frac{\sum_{k=1}^{l_i} \operatorname{rad} \left( U_k^{(i)} \right)^{m+1} \operatorname{wt}(U_k^{(i)})}{(T - t_i)^{m/2}} \|\nabla f\|_{\infty}$$

holds thus.

Example 2 below stands in the same relation to Theorem 5 as Example 1 does to Theorem 4. It thus provides a refined version of the patch condition given in Example 1. This condition is hereafter referred to as the weighted LL patch condition, sometimes abbreviated as the WLL patch condition.

**Example 2** (WLL patch condition). For a set of patch divisions  $\left\{U_k^{(i)}\right\}_{\substack{i=1,\dots,\sharp\Delta,\\k=1,\dots,l_i}}$  that satisfy

$$\operatorname{rad}\left(U_{k}^{(i)}\right) = (T - t_{i})^{m/(2(m+1))} \left(l_{i} \operatorname{wt}\left(U_{k}^{(i)}\right)\right)^{-1/(m+1)} s_{i}^{1/2}, \tag{37}$$

we have the recombination error of order  $O(n^{-(m-1)/2})$ .

3.2. Patch division algorithm. In this subsection, we present a recursive algorithm for constructing a set of patche divisions  $\left\{U_k^{(i)}\right\}_{\substack{i=1,\ldots,\sharp\Delta\\k=1,\ldots,l_i}}$  that satisfies (37),

based on the probability measure  $\nu^{(i)} = \sum_{j=1}^{m_i} \hat{w}_j^{(i)} \delta_{p_j^{(i)}}$ . To this end, we consider a measure  $\bar{\nu} = \sum_{j=1}^m w_j \delta_{p_j}$  on  $\mathbb{R}^N$  and define a penalty function  $\Psi_i$  as

$$\Psi_{i}(\bar{\nu}) = \operatorname{rad}(U^{\bar{\nu}}) - (T - t_{i})^{m/(2(m+1))} s_{i}^{1/2} \left(\operatorname{wt}(U^{\bar{\nu}})\right)^{1/(m+1)}, \tag{38}$$

where  $U^{\bar{\nu}} = \operatorname{supp}(\bar{\nu}) = \{p_j\}_{j=1}^m$ . The penalty function corresponds to the difference between the radius and the WLL patch condition.

Starting from  $\nu^{(i)} = \sum_{j=1}^{m_i} \hat{w}_j^{(i)} \hat{\delta}_{p_j^{(i)}}$ , we recursively execute the following steps 1– 3 to construct a collection of patches based on the penalty function. More precisely, given a measure  $\bar{\nu}$ , we perform Steps 1–3 to divide it into two measures  $\bar{\nu}_1$  and  $\bar{\nu}_2$ , and continue the procedure recursively for each measure until the penalty function  $\Psi_i(\bar{\nu}_i)$  becomes positive:

$$\Psi_i(\bar{\nu}_i) > 0 \quad (j = 1, 2).$$

If the penalty function becomes positive for a measure  $\bar{\nu}_i$ , we define the support of  $\bar{\nu}_i$  as a patch  $U_{i}^{(i)}$ , and increment k by one.

Step 1. Calculate the projections of the nodes onto the least-squares direction We set the starting point c of the projected vector as the centre of mass of the given nodes. Thus we first calculate the centre of mass  $c = (c^{(1)} \cdots c^{(N)}) \in$  $\mathbb{R}^N$ , that is

$$c = \sum_{j=1}^{m} w_j p_j. \tag{39}$$

Then, we set the end point of the projected vector  $e \in \mathbb{R}^N$  as the minimiser of weighted squared error, that is

$$e = \arg\min_{c \in \mathbb{R}^N} \sum_{j=1}^m w_j \| p_j - c \|_{\mathbb{R}^N}.$$
 (40)

Step 2. Sort the nodes  $\{(p_j, w_j)\}_{j=1,\dots,m}$  in order of inner product with the projected vector, that is

$$\langle p_j - c, e - c \rangle$$
 (41)

to  $\{(p_{s(j)},w_{s(j)})\}_{j=1,\dots,m}$ . Step 3. Divide the nodes into two patches so that each patch has an equal number of nodes. Namely, divide  $\bar{\nu}$  into two measures  $\bar{\nu}_1$  and  $\bar{\nu}_2$  as

$$\bar{\nu}_1 = \sum_{j=1}^{\lfloor m/2 \rfloor} w_{s(j)} \delta_{p_{s(j)}}, \text{ and } \bar{\nu}_2 = \sum_{j=\lfloor m/2 \rfloor + 1}^m w_{s(j)} \delta_{p_{s(j)}}.$$
(42)

Implementation advantages. The above algorithm produces a patch division that satisfies the condition in (37), while minimising the total number of patches as far as possible. Moreover, due to its recursive nature, the division can be performed efficiently. This contributes to reducing the overall computational cost of the recombination method.

Figures 2 and 3 show an example of patch division. This example corresponds to the pricing of an Asian option under the Heston model, discussed in subsection 4.1, with the number of partitions n=6 and the fifth time step  $t_5=(5/6)T$ . Here the node size indicates its weight, and the colour shows the patch to which it belongs. We observe that nodes with smaller weights—that is, those represented by smallersized circles—are grouped into a single large patch.

# 4. Numerical results

To discuss the practical efficiency of the high-order recombination, we conduct the following numerical experiments.

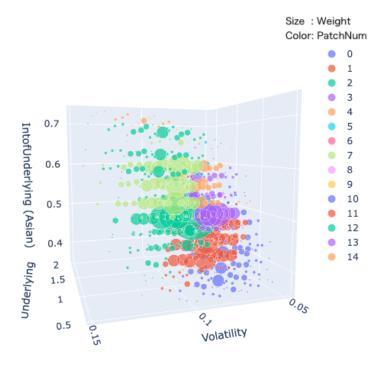


FIGURE 2. Example of patch division

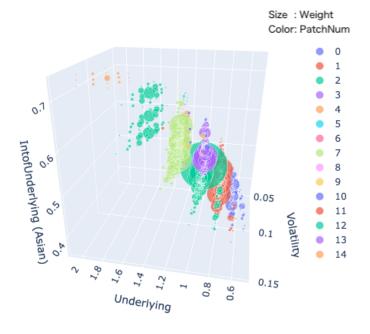


FIGURE 3. Example of patch division (from a different angle)

4.1. Convergence of Discretisation error. First, to verify the theoretical converges in practical problems, we consider the pricing an Asian call option under the Heston model, which is also considered in [24, 20, 26]. Let  $Y^{(1)}(t,x)$  be the price process and  $Y^{(2)}(t,x)$  be the volatility process as follows.

$$dY^{(1)}(t,x) = \mu Y^{(1)}(t,x) dt + Y^{(1)}(t,x) \sqrt{Y^{(2)}(t,x)} dB^{(1)}(t),$$
  

$$dY^{(2)}(t,x) = \alpha(\theta - Y^{(2)}(t,x)) dt + \beta \sqrt{Y^{(2)}(t,x)} \left(\rho dB^{(1)}(t) + \sqrt{1-\rho^2} dB^{(2)}(t)\right)$$

where  $x = (x_1, x_2) \in (\mathbb{R}_{>0})^2$ ,  $-1 \le \rho \le 1$ . We set the parameters  $\alpha, \beta, \theta$  to satisfy the Feller condition:

$$2\alpha\theta - \beta^2 > 0 \tag{43}$$

to ensure the existence and uniqueness of a solution to the CIR process  $Y^{(2)}(t,x)$  [5]. Note that when the CIR process is discretised using the N–V method, the positivity can be guaranteed under the following condition:

$$4\alpha\theta - \beta^2 > 0, (44)$$

which is weaker as a condition on  $\beta$  than (43). This condition for the positivity can be derived the discretisation formula presented in [24]. The payoff of the Asian call option on this asset with maturity T and strike K is  $\max(Y^{(3)}(T,x)/T-K,0)$  where

$$Y^{(3)}(t,x) = \int_0^t Y^{(1)}(s,x) \ ds.$$

Let  $Y(t,x) = {}^t(Y^{(1)}(t,x),Y^{(2)}(t,x),Y^{(3)}(t,x))$ . Here we change the coordinate of  $Y^{(2)}$  for numerical stability of the simulation. Then Y(t,x) is the solution of the Stratonovich form SDEs

$$Y(t,x) = \sum_{i=0}^{2} \int_{0}^{t} W_{i}(Y(s,x)) \circ dB^{i}(s)$$

where

$$W_{0}\begin{pmatrix} y_{1} \\ y_{2} \\ y_{3} \end{pmatrix} = \begin{pmatrix} y_{1} \left(\mu - y_{2}/2\right) - \beta \rho y_{1}/4 \\ \alpha \left(\theta - y_{2}\right) - \beta^{2}/4 \\ y_{1} \end{pmatrix},$$

$$W_{1}\begin{pmatrix} y_{1} \\ y_{2} \\ y_{3} \end{pmatrix} = \begin{pmatrix} y_{1}\sqrt{y_{2}} \\ \beta \rho \sqrt{y_{2}} \\ 0 \end{pmatrix},$$

$$W_{2}\begin{pmatrix} y_{1} \\ y_{2} \\ y_{3} \end{pmatrix} = \begin{pmatrix} 0 \\ \beta \sqrt{y_{2}(1 - \rho^{2})} \\ 0 \end{pmatrix}.$$

We set  $T=1, K=1.05, \mu=0.05, \alpha=2.0, \beta=0.1, \theta=0.09, \rho=0.3$ , and  $(x_1, x_2)=(1.0, 0.09)$  and take the true value as follows:

$$E_P[f(Y(T,x))] = E_P[\max(Y_3(T,x)/T - K,0)] = 0.06068740243939$$

This value is obtained by the third order method [26] and the quasi-Monte Carlo with the number of partitions n = 100 and the number of sample points  $M = 10^9$ . For more verification, refer to Section 4 of [26].

Figure 4 shows the convergences of the total error

$$\left| E_{\mu_{x,\Delta,\mathcal{U}}^{(n)}} \left[ f\left(Y^{(n)}(T,x)\right) \right] - E_P \left[ f\left(Y(T,x)\right) \right] \right|,$$

which is the difference between the expectations under the recombined measure  $\mu_{x,\Delta,\mathcal{U}}^{(n)}$  and the original measure P, with respect to the numbers of partition n. Here, we set a time partition  $\Delta$  as the even partition, that is  $s_i = 1/n$ , and a set of radii as in Corollary 2. See Remark 5 below on the reasons for adopting the even partition here.

The errors in the E–M and N–V method with and without the recombination are plotted. Figure 5 shows the growth of the number of support. As clearly shown in Figure 4, the theoretical convergences of error can be attained in each method, whether we use the recombination or not. If we do not use the recombination, that is the whole tree calculation, only executable up to around n=8, these are limits due to the computational complexity. On the other hand, if we use the recombination, we can calculate more larger numbers of partitions n, and we obtain the theoretical order. In fact, Figure 5 clearly shows that the recombination enables us to suppress the exponential growth of the number of support to the polynomial growth, O(n) to  $O(n^2)$  in our example. This indicates that the proposed recombination algorithm enables us to conquer the problem of the escalating computational complexity in this practical problem.

Remark 4 (Heston model and the UH condition). The Heston model do not satisfy the UH condition which we assume in Theorem 5. Nevertheless, the numerical result demonstrates that our algorithm works for the Heston model case. In fact, the following considerations generally justify the numerical result. In the Heston model the positivity of the discretised process by the N-V method ensures that  $\operatorname{supp}(\nu^{(i)}) \subset (\mathbb{R}_{>0})^3 \coloneqq \{(y_1, y_2, y_3) \mid y_1, y_2, y_3 > 0\}$ , where  $\nu^{(i)}$  is an intermediate measure defined in (12). This indicates that for any patch division  $\{U_k^{(i)}\}_{k=1,\ldots,l_i}$  of  $\operatorname{supp}(\nu^{(i)})$ 

there exists an  $\varepsilon > 0$  such that  $\bigcup_{k=1,...,l_i} U_k^{(i)} \subset \{(y_1,y_2,y_3) \mid y_1,y_2,y_3 > \varepsilon\}$ . Moreover, in the Heston model, the set of vectors  $\{W_1(x),W_2(x),[W_0,W_1](x)\}$  spans the linear space  $\mathbb{R}^3$  for each  $x \in (\mathbb{R}_{>0})^3$ . Thus, restricting the domain of  $P_t^X f$  to each patch  $U_k^{(i)}$  ensures the UH condition, which implies the UFG condition. Hence, Theorem 2 of [12] yields a constant  $C_{U_k^{(i)}}$  such that (35) holds, and

$$\max_{\substack{i=1,\ldots,n\\k=1,\ldots,l_i}} C_{U_k^{(i)}} < \infty.$$

However, this  $\varepsilon$  depends on the discretisation and recombination algorithm, so we cannot know the value of  $\varepsilon$  before executing the algorithm. Therefore, to fully justify our algorithm for the Heston model case, we need to prove Theorem 5 under a weaker condition than the UH condition such as the UFG condition, which is a subject for future work.

Remark 5 (Optimality of even partitioning). Throughout these numerical experiments, we use the even partition, that is  $s_i = T/n$ , which is proved to be optimal for the original continuous versions of N-V and N-N methods in [14] and not for the discrete versions. However, our numerical experiments show that it also works for the discretised N-V method, as similarly demonstrated in [22].

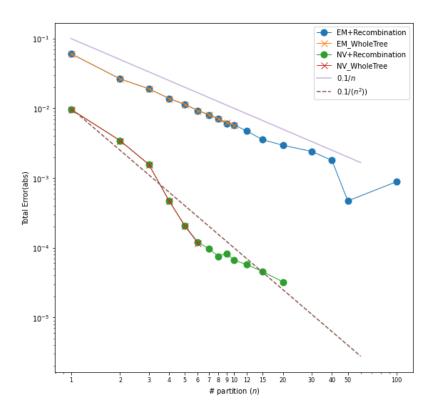
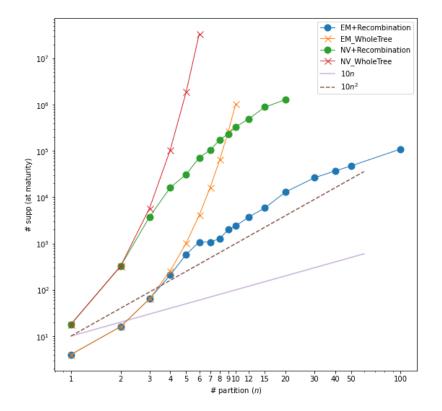


FIGURE 4. Total Error (Heston)



4.2. Effect of the patch division algorithm on recombination error. To investigate how patch division affects recombination error, we conduct further validation. The comparison includes the specific algorithm described in the previous subsection, which we hereafter refer to as Recursive\_WLL, as well as other alternative patch division strategies.

We introduce a scaling parameter  $\lambda>0$  into the penalty function defined as follows:

$$\Psi_i^{(\lambda)}(U) = \text{rad}(U) - \lambda (T - t_i)^{m/(2(m+1))} s_i^{-1/2} \left( \text{wt}(U) \right)^{1/(m+1)}. \tag{45}$$

For a fixed number of partitions n, we vary  $\lambda$  and compute the recombination errors

$$\left| \mu_T^{(\text{NV},\Delta,\text{Rec},\lambda)} \left( f \right) - \mu_T^{(\text{NV},\Delta)} \left( f \right) \right|, \tag{46}$$

that is the difference between the expectation under the recombined measure  $\mu_T^{(\text{NV},\Delta,\text{Rec},\lambda)}$  constructed using the penalty function (45), and that under the 'un-recombined' measure  $\mu_T^{(\text{NV},\Delta)}$ , defined as the full tree-based measure defined by (10) without applying recombination.

We compare the algorithm Recursive\_WLL with concentric circle dividing (CC), random dividing (Rand), and recursive path dividing with the Litterer–Lyons condition (Recursive\_LL). Note that the theoretical patch division condition is not necessarily satisfied in CC and Rand.

Figure 6 shows the recombination errors in pricing the Asian option under the Heston model, using the same setting as in the previous subsection. Here we fix the number of partitions at n=8. Among the methods compared, Recursive\_WLL achieves the smallest recombination error, followed by Recursive\_LL. The random dividing method yields moderate accuracy, while CC results in the largest error among the four.

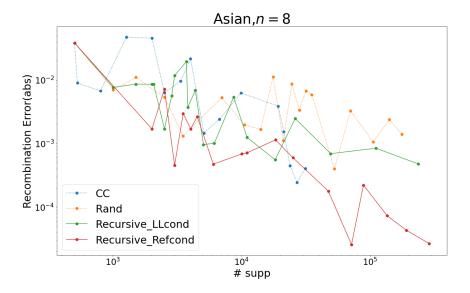


Figure 6. Recombination Error (Heston)

In addition, to investigate in which cases the patch conditions have a greater influence on the recombination error, we conduct a validation on the spiky payoff cases. Setting  $\left(\tilde{X}(T,x),\tilde{Y}(T)\right)=(Y_3(T,x)/Y_1(0,x),Y_2(T,x)/Y_2(0,x))$  and

$$f(\tilde{x}, \tilde{y}) = h - \frac{h}{r} \sqrt{(x - x_c)^2 + (y - y_c)^2},$$

where h is the height, r is the radius, and  $(x_c, y_c)$  is the centre of cone. We set the height as  $h = 10r^{-2}$  so that the volume of the cones remain constant. In this validation, We consider the following 80 cone payoffs in this coordinate.

- Radius: 5.0, 4.0, 3.0, 2.0, 1.5, 1.2, 1.0, 0.9
- Centre of cone: (0,0), (1,0), (0,1), (1,1), (2,0), (0,2), (2,2), (3,0), (0,3), (3,3)

Figure 7 shows some of above payoffs and nodes at the maturity, which indicate the above settings are exhaustive. Fixing the number of partitions n=6,8, we compare the recombination errors with respect to the number of supports at the maturity in each patch division method.

Figures 8 and 9 summarize the averages of slopes

$$\frac{\left|E_{\mu_{x,\Delta,\mathcal{U},\kappa}^{(n)}}\left[f\left(Y(^{(n)}T,x)\right)\right]-E_{\lambda_{x,\Delta,\mathcal{U}}^{(n)}}\left[f\left(Y(T,x)\right)\right]\right|}{\operatorname{card}\left(\operatorname{supp}\left(\mu_{x,\Delta,\mathcal{U},\kappa}^{(n)}\right)\right)}$$

with its confidence intervals. These graphs indicate that our patch division algorithm works well for most cases.

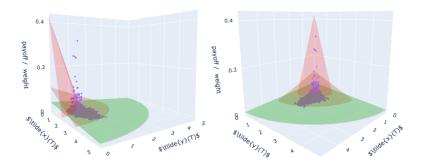


FIGURE 7. Spiky payoff

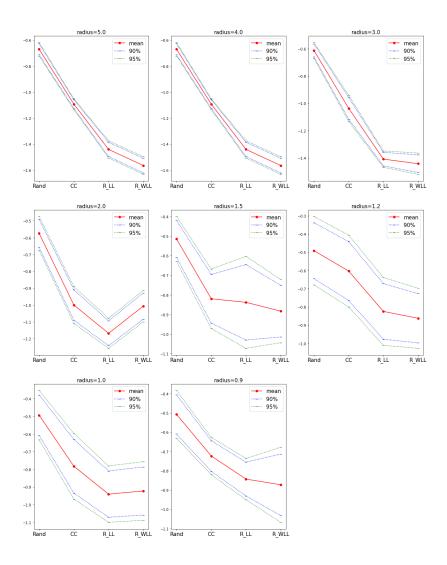


FIGURE 8. Average of slopes and its confidence interval

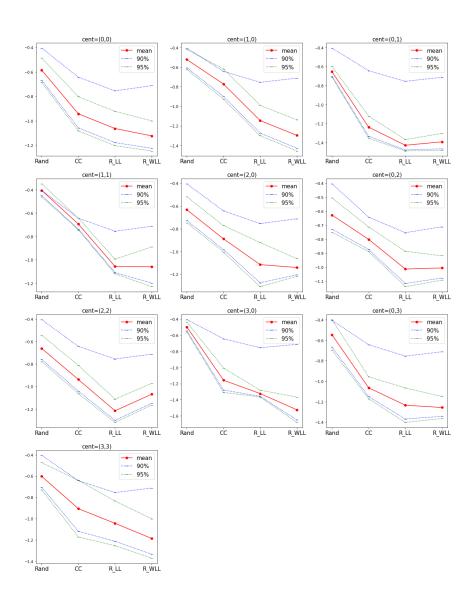


FIGURE 9. Average of slopes and its confidence interval

# Appendix1: Construction of a reduced measure

Let  $\nu$  be an induced probability measure on a patch U, defined in (15).  $\mathcal{N}_{\nu}$ denotes the number of nodes included in the patch U. Let  $G = \{g_1, g_2, \dots, g_{\mathcal{M}_G}\}$ be a set of test functions, where  $\mathcal{M}_G$  is the number of test functions and each  $g_i: \tilde{\Omega} \to \mathbb{R}$  is a measurable function. Hereafter,  $x^{(j)}$  denotes the jth coordinate of x, that is  $x = (x^{(1)}, \dots, x^{(N)})$ .

(1) Calculate the basis of kernel of the test function matrix. We define  $\mathcal{M}_G \times \mathcal{N}_{\nu}$  matrix  $A_0$  as follows:

$$A_{0} = \begin{pmatrix} g_{1}(y_{1}) & g_{1}(y_{2}) & \dots & g_{1}(y_{\mathcal{N}_{\nu}}) \\ g_{2}(y_{1}) & & \dots & & & & \\ \vdots & \vdots & \ddots & & \vdots \\ g_{\mathcal{M}_{G}}(y_{1}) & & \dots & g_{\mathcal{M}_{G}}(y_{\mathcal{N}_{\nu}}) \end{pmatrix}$$

$$(47)$$

Then, we calculate  $ker(A_0)$  using the singular value decomposition. Since  $\dim(\ker(A_0)) = \mathcal{N}_{\nu} - \mathcal{M}'_G$  where  $\mathcal{M}'_G = \operatorname{rank}(A_0)$ , we can choose a basis for  $\ker(A_0)$  consisting of  $\mathcal{N}_{\nu} - \mathcal{M}'_G$  elements denoted as  $\{v_{0,1}, v_{0,2}, \dots, v_{0,\mathcal{N}_{\nu} - \mathcal{M}'_G}\}$ We also let  $\beta_0 = {}^t(\phi_1 \quad \cdots \quad \phi_{\mathcal{N}_{\nu}}) \in [0, 1]^{\mathcal{N}_{\nu}}.$ 

(2) Eliminate the nodes to be recombined and update the weights.

Determine the set of nodes  $\{\}_{i=1}^{N_{\nu}-\mathcal{M}'_{G}}$  that are to be eliminated by the procedures specified for each  $i=1,\ldots,\mathcal{N}_{\nu}-\mathcal{M}'_{G}$  as follows:

(2-1) Determine a node to be eliminated by specifying the number of all nodes to be eliminated.

$$\alpha_{i} = \min_{1 \le j \le \mathcal{N}_{\nu} - i + 1} \left\{ \frac{\beta_{i-1}^{(j)}}{\left(v_{(i-1),1}\right)^{(j)}} \,\middle|\, \left(v_{(i-1),1}\right)^{(j)} > 0 \right\}$$
(48)

We also let

$$e_{i} = \underset{1 \leq j \leq \mathcal{N}_{\nu} - i + 1}{\operatorname{arg \, min}} \left\{ \frac{\beta_{i-1}^{(j)}}{\left(v_{(i-1),1}\right)^{(j)}} \, \middle| \, \left(v_{(i-1),1}\right)^{(j)} > 0 \right\},\tag{49}$$

that is 
$$\alpha_i = \frac{\beta_{i-1}^{(e_i)}}{(v_{(i-1),1})^{(e_i)}}.$$

(2-2) Eliminate the node  $(y_{e_i}, \phi_{e_i})$ . Eliminating the  $e_i$ th column of  $A_{i-1}$ , we obtain  $A_i$  as an  $\mathcal{M}_G \times (\mathcal{N}_{\nu} - i)$ matrix. Then, we update the probability measure by defining  $\beta_i \in$  $[0,1]^{\mathcal{N}_{\nu}-i}$  as follows:

$$\beta_i^{(j)} = \begin{cases} \beta_{i-1}^{(j)} - \alpha_i \left( v_{(i-1),1} \right)^{(j)} & \text{for } j = 1, \dots, e_i - 1, \\ \beta_{i-1}^{(j+1)} - \alpha_i \left( v_{(i-1),1} \right)^{(j+1)} & \text{for } j = e_i, \dots, \mathcal{N}_{\nu} - i. \end{cases}$$
(50)

(2-3) Update the basis of kernel of the test function matrix. For  $l = 1, ..., \mathcal{N}_{\nu} - \mathcal{M}'_G - i$ , we define

$$d_{i,l+1} = \frac{\left(v_{(i-1),(l+1)}\right)^{(e_i)}}{\left(v_{(i-1),1}\right)^{(e_i)}},$$

$$v_{i,l} = v_{(i-1),(l+1)} - d_{i,l+1}v_{(i-1),1} \in \mathbb{R}^{\mathcal{N}_{\nu}},$$

$$(v_{i,l})^{(j)} = \begin{cases} (v_{i,l})^{(j)} & \text{for } j = 1, \dots, e_i - 1\\ (v_{i,l})^{(j+1)} & \text{for } j = e_i, \dots, \mathcal{N}_{\nu} - i \end{cases}$$
(51)

**Example 3** (An Illutrative example). Let us give a simple example. Let X, Y be 2-dimensional discrete random variables on  $\Omega = \{\omega_0, \ldots, \omega_7\}$  such that

$$(X,Y)(\omega_0) = (0,0), \quad (X,Y)(\omega_1) = (1,0), \quad (X,Y)(\omega_2) = (0,1), \quad (X,Y)(\omega_3) = (1,1), \quad (X,Y)(\omega_4) = (2,0), \quad (X,Y)(\omega_5) = (2,1), \quad (X,Y)(\omega_6) = (2,2), \quad (X,Y)(\omega_7) = (1,2)$$

and  $\tilde{Q}$  be a probability measure on  $\Omega$  such that  $\tilde{Q}(\omega_0) = \cdots = \tilde{Q}(\omega_7) = 1/8$ . From  $\tilde{Q}$ , find its reduced measure Q such that the moments of this (X,Y) is invariant up to the second order, that is to say  $\tilde{Q}$  is the reduced measure from Q with respect to  $G = \{g_i(X,Y)\}_{i=1}^6$  where  $\{g_i\}_{i=1}^6 \subset \mathbb{R}[x,y]$  is defined as

$$g_1(x,y) = x^0 y^0 = 1$$
,  $g_2(x,y) = x$ ,  $g_3(x,y) = y$ ,  $g_4(x,y) = x^2$ ,  $g_5(x,y) = y^2$ ,  $g_6(x,y) = xy$ .

The problem reduces to finding a basis of Ker A of the following matrix A, where the ith column and the jth row corrrespond to the state  $\omega_{i-1}$  and the monomial  $g_j$  respectively:

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 2 & 2 & 2 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 2 & 2 \\ 0 & 1 & 0 & 1 & 4 & 4 & 4 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 4 & 4 \\ 0 & 0 & 0 & 1 & 0 & 2 & 4 & 2 \end{pmatrix}.$$

Then we see that  $Ker(A) = \mathbb{R}v_1 + \mathbb{R}v_2$  where

$$v_1 = {}^t (-1 \quad 2 \quad 1 \quad -2 \quad -1 \quad 1 \quad 0 \quad 0),$$
  
 $v_2 = {}^t (2 \quad -3 \quad -2 \quad 2 \quad 1 \quad 0 \quad -1 \quad 1).$ 

If we set

$$Q = \tilde{Q} - (1/8)v_1 - (1/24)v_2$$
  
=  ${}^{t}(1/6 \ 0 \ 1/12 \ 7/24 \ 5/24 \ 0 \ 1/6 \ 1/12),$ 

we see that Q satisfies

$$\begin{split} E^{\tilde{Q}}[X] &= E^{Q}[X], \quad E^{\tilde{Q}}[Y] = E^{Q}[Y], \\ E^{\tilde{Q}}[X^{2}] &= E^{Q}[X^{2}], \quad E^{\tilde{Q}}[XY] = E^{Q}[XY], \quad E^{\tilde{Q}}[Y^{2}] = E^{Q}[Y^{2}]. \end{split}$$

APPENDIX2: PSEUDO CODE OF THE HIGH-ORDER RECOMBINATION ALGORITHM Let  $\Delta = \{0 = t_0 < \dots < t_n = T\}$  be a partition of time [0, T].

```
: Number of partitions : n,
                    Parameters of SDE : x_0, V_0, \dots, V_d
                    Option parameters: maturity T, payoff function f
                  : E_{\mu_{t,r}^{(\mathrm{Alg},\Delta,\mathrm{Rec})}} f(X^{(\mathrm{NV},\Delta,\mathrm{Rec})}(T))
 1 Struct Node contains
                                   /* Realized value X^{(\mathrm{NV},\Delta,\mathrm{Rec})}(t_i)(\omega_i\in\Omega_i) */
        values;
                           /* Weight (Prob. measure of this node P(\omega_i) */
        weight;
                                                   /* patch number of this node */
       patchnum;
 4
 5 end
 6 Function Main():
        leaves \leftarrow array of Node;
        nextleaves \leftarrow array of Node;
        leaves[0].value \leftarrow x_0;
                                                         /* Set the initial value */
 9
        leaves[0].weight \leftarrow 1.0;
10
        for i \leftarrow 1 to n do
11
            nextleaves \leftarrow \mathtt{OneStepFwd}(leaves, i, n, T, V_0, \dots, V_d);
12
            numpatch \leftarrow 0;
13
            PatchDiv (nextleaves, numpatch);
14
            nextleaves \leftarrow \texttt{Recombination} \ (nextleaves);
15
            leaves \leftarrow nextleaves;
16
17
        nextleaves \leftarrow \mathtt{OneStepFwd}(leaves, i, n, T, V_0, \dots, V_d);
18
        value \leftarrow Payoff(nextleavesleaves, f);
19
20
        return value;
 1 Function OneStepFwd(leaves, i, n, T, V_0, \dots, V_d); /* Calculate \hat{X}^{(t_{i+1}, x_0)}
     from \hat{X}^{(t_i,x_0)}, using (24) */
 2:
        nextleaves \leftarrow array of Node;
 3
        forall node in leaves;
                                              /* Generate RVs \{\eta_i\}, solve ODEs */
 4
        do
 5
            for k \leftarrow 1 to 2 \times 3^d do
 6
                nextval \leftarrow Solve ODEs along with V_0, \ldots, V_d;
                                                                                  /* (24) */
 7
                nextleaves.node \leftarrow nextval;
 8
            end
 9
        \mathbf{end}
10
        return next leaves
11
```

```
1 PatchDiv(leaves,*numpatch)
                                                                       /* (19) */
2 Cond = calc\_cond(leaves);
s if Cond == True then
       forall node in leaves do
 4
          leaves.node.patchnum \leftarrow numpatch;
 5
 6
       end
      *numpatch \leftarrow *numpatch + 1;
 7
s end
9 else
      SortNodes (leaves);
10
      leftleaves \leftarrow leaves[numnode/2];
11
      rightleaves \leftarrow leaves[numnode/2:numnode];
12
      PatchDiv (leftleaves, numpatch);
13
      PatchDiv (rightleaves, numpatch);
14
15 end
16 Function SortNodes(leaves):
      g \leftarrow \text{gravitypoint}(leaves);
                                                                       /* (39) */
17
       e \leftarrow \text{endpoint}(leaves);
                                                                       /* (40) */
18
      forall node in leaves do
19
20
          node.order \leftarrow innerprod(node.value, q, e);
                                                                       /* (41) */
21
      end
       QuickSort(leaves, leaves.node.order);
                                                   /* Using external library
```

### References

- [1] Vlad Bally and Denis Talay. The law of the euler scheme for stochastic differential equations. Probability theory and related fields, 104(1):43–60, 1996.
- [2] Dan Crisan and Terry Lyons. Minimal entropy approximations and optimal algorithms. *Monte Carlo Methods and Applications*, 8(4):343–356, 2002.
- [3] Dan Crisan and Konstantinos Manolarakis. Solving backward stochastic differential equations using the cubature method: Application to nonlinear pricing. SIAM Journal on Financial Mathematics, 3(1):534–571, 2012.
- [4] Dan Crisan and Konstantinos Manolarakis. Second order discretization of backward SDEs and simulation with the cubature method. *Annals of Applied Probability*, 24(2):652–678, 2014.
- [5] William Feller. Two Singular Diffusion Problems. Annals of Mathematics, 54(1):173–182, 1951.
- [6] Emilio Ferrucci, Timothy Herschell, Christian Litterer, and Terry Lyons. High-degree cubature on wiener space through unshuffle expansions. arXiv preprint arXiv:2411.13707, 2024.
- [7] Takehiko Fujiwara. Sixth order methods of Kusuoka approximation. PreprintSeries, Graduate School of Mathematical Sciences, Univ. Tokyo, 2006.
- [8] Lajos Gyurkó and Terry Lyons. Efficient and practical implementations of cubature on Wiener space. In Stochastic analysis 2010, pages 73-111. Springer, 2011.
- [9] Satoshi Hayakawa and Ken'ichiro Tanaka. Monte carlo construction of cubature on wiener space. Japan Journal of Industrial and Applied Mathematics, 39(2):543-571, 2022.
- [10] Steven Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Review of financial studies*, 6(2):327–343, 1993.
- [11] Shigeo Kusuoka. Approximation of expectation of diffusion process and mathematical finance. In Proceedings of Final Taniguchi Symposium, Nara, volume 31 of Advanced Studies in Pure Mathematics, pages 147–165, 2001.
- [12] Shigeo Kusuoka. Malliavin calculus revisited. J. Math. Sci. Univ. Tokyo, 10(2):261–277, 2003.

- [13] Shigeo Kusuoka. Approximation of expectation of diffusion processes based on Lie algebra and Malliavin calculus. Advances in Mathematical Economics, 6:69–83, 2004.
- [14] Shigeo Kusuoka. Gaussian K-scheme: justification for KLNV method. Advances in Mathematical Economics, 17:71–120, 2013.
- [15] Wonjung Lee and Terry Lyons. The adaptive patched cubature filter and its implementation. Communications in Mathematical Sciences, 2016.
- [16] Christian Litterer. The Signature in Numerical Algorithms. PhD thesis, University of Oxford, 2008.
- [17] Christian Litterer, Terry Lyons, and Others. High order recombination and an application to cubature on Wiener space. The Annals of Applied Probability, 22(4):1301–1327, 2012.
- [18] Terry Lyons and Nicolas Victoir. Cubature on Wiener Space. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 460, pages 169–198, 2004.
- [19] Mariko Ninomiya. Application of the Kusuoka approximation with a tree-based branching algorithm to the pricing of interest-rate derivatives under the HJM model. LMS Journal of Computation and Mathematics, 13:208–221, 2010.
- [20] Mariko Ninomiya and Syoiti Ninomiya. A new higher-order weak approximation scheme for stochastic differential equations and the Runge–Kutta method. Finance and Stochastics, 13(3):415–443, 2009.
- [21] Syoiti Ninomiya. A partial sampling method applied to the Kusuoka approximation. *Monte Carlo methods and Applications*, 9(1):27–38, 2003.
- [22] Syoiti Ninomiya and Yuji Shinozaki. Higher-order Discretization Methods of Forward-backward SDEs Using KLNV-scheme and Their Applications to XVA Pricing. Applied Mathematical Finance, 26(3):257–292, 2019.
- [23] Syoiti Ninomiya and Yuji Shinozaki. On implementation of high-order recombination and its application to weak approximations of stochastic differential equations. 2021.
- [24] Syoiti Ninomiya and Nicolas Victoir. Weak approximation of stochastic differential equations and application to derivative pricing. Applied Mathematical Finance, 15(2):107–121, 2008.
- [25] Kojiro Oshima, Josef Teichmann, and Dejan Velušček. A new extrapolation method for weak approximation schemes with applications. Annals of Applied Probability, 22(3):1008–1045, 2012.
- [26] Yuji Shinozaki. Construction of a third-order K-scheme and its application to financial models. SIAM Journal on Financial Mathematics, 8(1):901–932, 2017.
- [27] Maria Tchernychova. Carathéodory cubature measures. PhD thesis, University of Oxford, 2016.