



Split S-ROCK Methods for High-Dimensional Stochastic Differential Equations

Yoshio Komori¹ · Kevin Burrage²

Received: 26 April 2023 / Revised: 2 September 2023 / Accepted: 7 September 2023 /
Published online: 30 October 2023
© Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract

We propose explicit stochastic Runge–Kutta (RK) methods for high-dimensional Itô stochastic differential equations. By providing a linear error analysis and utilizing a Strang splitting-type approach, we construct them on the basis of orthogonal Runge–Kutta–Chebyshev methods of order 2. Our methods are of weak order 2 and have high computational accuracy for relatively large time-step size, as well as good stability properties. In addition, we take stochastic exponential RK methods of weak order 2 as competitors, and deal with implementation issues on Krylov subspace projection techniques for them. We carry out numerical experiments on a variety of linear and nonlinear problems to check the computational performance of the methods. As a result, it is shown that the proposed methods can be very effective on high-dimensional problems whose drift term has eigenvalues lying near the negative real axis and whose diffusion term does not have very large noise.

Keywords Explicit method · Weak second order approximation · Orthogonal Runge–Kutta–Chebyshev method · Stiffness · Noncommutative noise · Itô stochastic differential equation

Mathematics Subject Classification 60H10 · 65L05 · 65L06

1 Introduction

We are concerned with stabilized explicit methods for high-dimensional stochastic differential equations (SDEs), which give weak second order approximations to the solution of SDEs. One such class for ordinary differential equations (ODEs) is the Runge–Kutta–Chebyshev

✉ Yoshio Komori
komori@phys.kyutech.ac.jp; yoshio.kom@gmail.com

Kevin Burrage
kevin.burrage@qut.edu.au; kevin.burrage@gmail.com

¹ Department of Physics and Information Technology, Kyushu Institute of Technology, 680-4 Kawazu, Iizuka 820-8502, Japan

² School of Mathematical Sciences, Queensland University of Technology, Brisbane, Australia

(RKC) methods, which are useful for stiff problems whose eigenvalues lie near the negative real axis [3, 30]. When the dimension of ODEs is very high, the class can be a powerful tool, compared with other stabilized explicit methods such as exponential Runge–Kutta (RK) methods. Recently, the class has played a significant role for the development of stabilized explicit methods for SDEs.

Abdulle and Cirilli [1] and Abdulle and Li [2] have proposed a family of explicit stochastic orthogonal Runge–Kutta–Chebyshev (SROCK) methods of strong order 1/2 and weak order 1 for noncommutative Stratonovich SDEs and for noncommutative Itô SDEs, respectively. As the first order RKC methods are embedded into their SROCK methods, the SROCK methods have extended mean square (MS) stability regions. Komori and Burrage [21] have proposed SROCK methods of weak order 2 for noncommutative Stratonovich SDEs. The second order orthogonal RKC methods proposed in [3] are embedded into the SROCK methods from [21]. We will refer to the orthogonal RKC methods as the standard ROCK2 methods. The SROCK methods in [21] have the advantage that the stability region is large along the negative real axis, but their stability region is not very wide. In order to overcome this drawback, Abdulle, Vilmart, and Zygalkis [4] have proposed a new family of weak second order SROCK methods for noncommutative Itô SDEs. We call them SROCK2 methods throughout the present paper. In the SROCK2 methods, a new family of second order orthogonal RKC methods is embedded, and it has a real parameter α . We will refer to them as the ROCK2 methods with α . The ROCK2 methods with $\alpha = 1$ reduce to the standard ROCK2 methods.

Incidentally, there is another suitable class of explicit methods for semilinear problems that have stiffness in the linear part as opposed to the nonlinear part. One such class of methods for semilinear ODEs is the family of explicit exponential RK methods [15–17]. These methods have been recently developed for SDEs [9–12, 14, 22]. Some exponential RK methods have been proposed to cope with highly oscillatory problems [9, 12]. In [23], the authors have proposed weak second order stochastic exponential RK methods for noncommutative Itô SDEs with a semilinear drift term, and showed the superiority of stochastic exponential RK methods to SROCK2 methods in terms of computational accuracy for relatively large step size in numerical experiments. This fact reduces the advantage of SROCK2 methods against exponential RK methods in high-dimensional problems.

In the present paper, we propose the split SROCK2 methods, which not only are explicit and stabilized, but also have high computational accuracy for relatively large step size. Our extension for the SROCK2 methods is very helpful to remove a step size restriction that the SROCK2 methods suffer from for computational accuracy in high-dimensional SDEs. In Sect. 2 we will briefly introduce the derivative-free Milstein–Talay (DFMT) method. In Sect. 3 we will derive the split SROCK2 methods and will give their error analysis. In Sect. 4 we will state other stabilized explicit methods based on the ROCK2 methods with α , as well as stochastic exponential RK methods. In addition, we will deal with the implementation issues of Krylov subspace projection techniques for matrix exponential functions. In Sect. 5 we will present numerical results and in Sect. 6 our conclusions.

2 Weak Second Order Stochastic RK Methods

First of all, We briefly introduce the definition of weak order of convergence and the DFMT method of weak order 2 [4]. We are concerned with the autonomous SDE

$$dy(t) = f(y(t))dt + \sum_{j=1}^m g_j(y(t))dW_j(t), \quad y(0) = y_0, \quad (1)$$

where $t \in [0, T]$ and where $f, g_j, j = 1, 2, \dots, m$, are \mathbb{R}^d -valued functions on \mathbb{R}^d , the $W_j(t), j = 1, 2, \dots, m$, are independent Wiener processes, and y_0 is independent of $W_j(t) - W_j(0)$ [7, p. 100]. In order to consider weak q -th order approximations to its solution, we make the assumptions on (1) that all moments of the initial value y_0 exist and $f, g_j, j = 1, 2, \dots, m$, are Lipschitz continuous with all their components belonging to $C_P^{2(q+1)}(\mathbb{R}^d, \mathbb{R})$. As usual, $C_P^L(\mathbb{R}^d, \mathbb{R})$ denotes the family of L times continuously differentiable real-valued functions on \mathbb{R}^d , whose partial derivatives of order less than or equal to L have polynomial growth [20, p. 474]. Let y_n denote a discrete approximation to the solution $y(t_n)$ of (1) for an equidistant grid point $t_n \stackrel{\text{def}}{=} nh$ ($n = 1, 2, \dots, M$) with step size $h = T/M < 1$ (M is a positive integer). Then we define the weak convergence of order q as follows [20, p. 327].

Definition 1 When a numerical method gives discrete approximations $y_n, n = 1, 2, \dots, M$, we say that the method is of weak (global) order q if for all $G \in C_P^{2(q+1)}(\mathbb{R}^d, \mathbb{R})$, constants $C > 0$ (independent of h) and $\delta_0 > 0$ exist such that

$$|E[G(y(T)) - E[G(y_M)]]| \leq Ch^q, \quad h \in (0, \delta_0).$$

An effective numerical method of weak order 2, the DFMT method, has been proposed [4]:

$$\begin{aligned} K_1 &= y_n + hf(y_n), & K_2 &= K_1 + \sqrt{h} \sum_{j=1}^m g_j(y_n)\xi_j, \\ y_{n+1} &= y_n + \frac{h}{2} \{f(y_n) + f(K_2)\} + H(y_n) + \tilde{H}\left(\frac{y_n + K_1}{2}, y_n\right), \end{aligned} \tag{2}$$

where

$$\begin{aligned} H(y) &\stackrel{\text{def}}{=} \frac{1}{2} \sum_{j=1}^m \left\{ g_j\left(y + h \sum_{k=1}^m g_k(y)\zeta_{kj}\right) - g_j\left(y - h \sum_{k=1}^m g_k(y)\zeta_{kj}\right) \right\}, \\ \tilde{H}(y, z) &\stackrel{\text{def}}{=} \frac{\sqrt{h}}{2} \sum_{j=1}^m \left\{ g_j\left(y + \sqrt{\frac{h}{2}} \sum_{k=1}^m g_k(z)\chi_k\right) \right. \\ &\quad \left. + g_j\left(y - \sqrt{\frac{h}{2}} \sum_{k=1}^m g_k(z)\chi_k\right) \right\} \xi_j, \end{aligned}$$

and where the χ_j and $\xi_j, j = 1, 2, \dots, m$, are discrete random variables satisfying $P(\chi_j = \pm 1) = 1/2, P(\xi_j = \pm\sqrt{3}) = 1/6$, and $P(\xi_j = 0) = 2/3$, and the $\zeta_{kj}, j, k = 1, 2, \dots, m$, are given by

$$\zeta_{kj} \stackrel{\text{def}}{=} \begin{cases} (\xi_j \xi_j - 1)/2 & (j = k), \\ (\xi_k \xi_j - \chi_k)/2 & (j < k), \\ (\xi_k \xi_j + \chi_j)/2 & (j > k). \end{cases}$$

Based on the following well-known theorem proposed by Milstein [25, 26], the proof of the weak order of convergence is obtained for (2). For details, see [4].

Theorem 1 Suppose that the numerical solutions satisfy the following conditions:

- (1) for a sufficiently large $r \in \mathbb{N}$, the moments $E[\|y_n\|^{2r}]$ exist and are uniformly bounded with respect to M and $n = 0, 1, \dots, M$;

(2) for all $G \in C_p^{2(q+1)}(\mathbb{R}^d, \mathbb{R})$, the local error estimation

$$|E[G(y(t_{n+1}))] - E[G(y_{n+1})]| \leq |K(y_n)|h^{q+1}$$

holds if $y(t_n) = y_n$, where $K \in C_p^0(\mathbb{R}^d, \mathbb{R})$.

Then, the method that gives $y_n, n = 1, 2, \dots, M$, is of weak (global) order q .

Remark 1 If (1) has diagonal noise [20, p. 348], that is, $d = m$ and

$$[g_1(y) \ g_2(y) \ \dots \ g_d(y)] = \text{diag}(g_{11}(y_1), g_{22}(y_2), \dots, g_{dd}(y_d))$$

are satisfied, then we have $\sum_{j=1}^d g_j(y)\xi_j = [g_{11}(y_1)\xi_1 \ g_{22}(y_2)\xi_2, \dots, g_{dd}(y_d)\xi_d]^T$, where g_{jj} and $y_j, j = 1, 2, \dots, d$, denote the j -th component of g_j and y , respectively. Similarly, $H(y)$ is a d -dimensional vector whose j -th component is

$$\frac{1}{2} \{g_{jj}(y_j + hg_{jj}(y_j)\zeta_{jj}) - g_{jj}(y_j - hg_{jj}(y_j)\zeta_{jj})\},$$

and $\tilde{H}(y, z)$ is a d -dimensional vector whose j -th component is

$$\frac{\sqrt{h}}{2} \left\{ g_{jj} \left(y_j + \sqrt{\frac{h}{2}} g_{jj}(z_j) \chi_j \right) + g_{jj} \left(y_j - \sqrt{\frac{h}{2}} g_{jj}(z_j) \chi_j \right) \right\} \xi_j.$$

If $g_j, j = 1, 2, \dots, m$, vanish in (1), then the problem is an ODE. Thus, we give a brief introduction to the ROCK2 methods with a free parameter α for ODEs [3]. Originally, the standard ROCK2 methods have been proposed by Abdulle and Medovikov [3]. The parameter values of the methods are given in a site¹ on the Internet. Abdulle, Vilmart, and Zygalkakis [4] have extended the standard ROCK2 methods to the following ROCK2 methods with a free parameter α :

$$\begin{aligned} K_0 &= y_n, & K_1 &= K_0 + \mu_1 \alpha h f(K_0), \\ K_i &= \mu_i \alpha h f(K_{i-1}) + (\kappa_i + 1) K_{i-1} - \kappa_i K_{i-2}, & i &= 2, 3, \dots, s-2, \\ K_{s-1} &= K_{s-2} + 2\tau_\alpha h f(K_{s-2}), \\ y_{n+1} &= K_{s-2} + \left(2\theta_\alpha - \frac{1}{2} \right) h f(K_{s-2}) + \frac{h}{2} f(K_{s-1}). \end{aligned} \tag{3}$$

Here, $\theta_\alpha \stackrel{\text{def}}{=} (1 - \alpha)/2 + \alpha\theta$, $\tau_\alpha \stackrel{\text{def}}{=} (1 - \alpha)^2/2 + 2\alpha(1 - \alpha)\theta + \alpha^2\tau$, and the μ_i, κ_i, θ , and τ are the same constants as in the standard s -stage ROCK2 method. This method achieves order 2 for ODEs regardless of the value of α .

3 Novel Stabilized Explicit Methods of Weak Order 2

We want explicit methods to inherit preferable stability properties from (3) by involving its stabilization procedures, while keeping weak order 2 for any α . In order to achieve this, we propose to utilize a splitting technique. As a result, we construct novel stabilized explicit methods of weak order 2, which have not only good stability properties, but also high precision in the weak sense.

¹ <https://www.epfl.ch/labs/anmc/software/>.

3.1 Preliminary

When we consider deriving a new method of weak order 2 based on the DFMT method, the following lemma proposed in [23] can be a useful tool.

Lemma 1 For an approximate solution y_n , let y_{n+1} be given by (2) and \hat{y}_{n+1} be defined by

$$\hat{y}_{n+1} = \tilde{y}_{n+1} + \frac{h}{2} f \left(Y_1 + \sqrt{h} \sum_{j=1}^m g_j(Y_2) \xi_j \right) + H(Y_3) + \tilde{H}(Y_4, Y_5).$$

Here, we assume that the intermediate values \tilde{y}_{n+1} and $Y_i, i = 1, 2, \dots, 5$, have no random variable and satisfy $Y_4 = y_n + (h/2) f(y_n) + O(h^2), Y_i = y_n + O(h), i = 1, 2, 3, 5$, and

$$\tilde{y}_{n+1} + \frac{h}{2} f(Y_1) = y_n + h f(y_n) + \frac{h^2}{2} f'(y_n) f(y_n) + O(h^3). \tag{4}$$

(Note that the symbol $O(h^p)$ represents terms x such that $\|x\| \leq |K(y_n)|h^p$ for a $K \in C_p^0(\mathbb{R}^d, \mathbb{R})$ and a small $h > 0$.) Then, for all $G \in C_p^r(\mathbb{R}^d, \mathbb{R}) (r \geq 3)$,

$$E[G(\hat{y}_{n+1})] - E[G(y_{n+1})] = O(h^3). \tag{5}$$

The following lemma gives a splitting technique which does not violate the weak second order of convergence. A related theorem was originally proposed in [23]. Here, we write part of the statement in the theorem with an appropriate notation for the present paper.

Lemma 2 For an approximate solution y_n , let y_{n+1} be given by (2) and \hat{y}_{n+1} be defined by

$$\hat{y}_{n+1} = \Psi_{\frac{h}{2}} \left(\Phi_h \left(\Psi_{\frac{h}{2}}(y_n) \right) \right),$$

where Φ_h is the DFMT method for SDEs given by making the drift term zero, that is,

$$\Phi_h(y_n) = y_n + H(y_n) + \tilde{H}(y_n, y_n), \tag{6}$$

and where Ψ_h denotes a numerical method that at least satisfies

$$\Psi_h(y_n) = y_n + h f(y_n) + \frac{h^2}{2} f'(y_n) f(y_n) + O(h^3).$$

Then, for all $G \in C_p^r(\mathbb{R}^d, \mathbb{R}) (r \geq 3)$

$$E[G(\hat{y}_{n+1})] - E[G(y_{n+1})] = O(h^3).$$

3.2 An Extended Class of the Stabilized Explicit Methods

Now, let us put all ideas mentioned so far together and organize them in a sophisticated way to obtain a new class of the stabilized explicit methods. Then, we propose

$$\begin{aligned}
 \mathbf{K}_0 &= \mathbf{y}_n, & \mathbf{K}_1 &= \mathbf{K}_0 + \mu_1 \alpha \frac{h}{2} \mathbf{f}(\mathbf{K}_0), \\
 \mathbf{K}_i &= \mu_i \alpha \frac{h}{2} \mathbf{f}(\mathbf{K}_{i-1}) + (\kappa_i + 1) \mathbf{K}_{i-1} - \kappa_i \mathbf{K}_{i-2}, & i &= 2, 3, \dots, s-2, \\
 \tilde{\mathbf{K}}_{s-1} &= \mathbf{K}_{s-2} + \tau_\alpha h \mathbf{f}(\mathbf{K}_{s-2}), & \hat{\mathbf{K}}_{s-1} &= \mathbf{K}_{s-2} + \theta_\alpha h \mathbf{f}(\mathbf{K}_{s-2}), \\
 \mathbf{K}_s^* &= \mathbf{K}_{s-2} + \left(2\theta_\alpha - \frac{1}{2}\right) \frac{h}{2} \mathbf{f}(\mathbf{K}_{s-2}) + \frac{h}{4} \mathbf{f}(\tilde{\mathbf{K}}_{s-1}) + \mathbf{H}(\mathbf{K}_{s-2}) \\
 &+ \tilde{\mathbf{H}}(\hat{\mathbf{K}}_{s-1}, \mathbf{K}_{s-2}), & \mathbf{y}_{n+1} &= \Psi_{\frac{h}{2}}(\mathbf{K}_s^*).
 \end{aligned}
 \tag{7}$$

We name the methods (7) in our extended class as the split SROCK2 methods. In what follows, we will call them by this name.

Theorem 2 *Suppose that all moments of the initial value \mathbf{y}_0 exist and $\mathbf{f}, \mathbf{g}_j, j = 1, 2, \dots, m$, are Lipschitz continuous with all their components belonging to $C_p^6(\mathbb{R}^d, \mathbb{R})$ in (1). Then, (7) is of weak order 2 for (1).*

Proof We begin with

$$\mathbf{y}_{n+1} = \Psi_{\frac{h}{2}} \left(\Phi_h \left(\Psi_{\frac{h}{2}}(\mathbf{y}_n) \right) \right), \tag{8}$$

where Φ_h is given in (6) and $\Psi_h(\mathbf{y}_n)$ denotes \mathbf{y}_{n+1} given in (3). As mentioned after (3), since (3) achieves order 2 for ODEs regardless of the value of α , its local error is of order 3 for ODEs. Due to Lemma 2, then (8) clearly has a local error of weak order 3. In order to see that the difference between (8) and (7) is of weak order 3, let us show that the difference between $\Phi_h(\Psi_{\frac{h}{2}}(\mathbf{y}_n))$ and \mathbf{K}_s^* is of weak order 3. When we denote $\Psi_{\frac{h}{2}}(\mathbf{y}_n)$ by $\tilde{\Psi}$, we can rewrite these expressions as

$$\Phi_h \left(\Psi_{\frac{h}{2}}(\mathbf{y}_n) \right) = \tilde{\Psi} + \mathbf{H}(\tilde{\Psi}) + \tilde{\mathbf{H}}(\tilde{\Psi}, \tilde{\Psi}), \quad \mathbf{K}_s^* = \tilde{\Psi} + \mathbf{H}(\mathbf{K}_{s-2}) + \tilde{\mathbf{H}}(\hat{\mathbf{K}}_{s-1}, \mathbf{K}_{s-2}).$$

From (7) we obtain the estimates

$$\tilde{\Psi} - \mathbf{K}_{s-2} = h \left\{ \frac{1}{2} \left(2\theta_\alpha - \frac{1}{2} \right) \mathbf{f}(\mathbf{K}_{s-2}) + \frac{1}{4} \mathbf{f}(\tilde{\mathbf{K}}_{s-1}) \right\} = h\mathbf{a} + O(h^2)$$

and

$$\tilde{\Psi} - \hat{\mathbf{K}}_{s-1} = h \left\{ -\frac{1}{4} \mathbf{f}(\mathbf{K}_{s-2}) + \frac{1}{4} \mathbf{f}(\tilde{\mathbf{K}}_{s-1}) \right\} = h^2\mathbf{b} + O(h^3),$$

where $\mathbf{a} = (\theta_\alpha - 1/4)\mathbf{f}(\mathbf{y}_n) + (1/4)\mathbf{f}(\mathbf{y}_n)$ and $\mathbf{b} = (1/4)\tau_\alpha \mathbf{f}'(\mathbf{y}_n)\mathbf{f}(\mathbf{y}_n)$. Since we can rewrite $\mathbf{H}(\mathbf{Y})$ as

$$\mathbf{H}(\mathbf{Y}) = h \sum_{j,k=1}^m \mathbf{g}'_j(\mathbf{Y})\mathbf{g}_k(\mathbf{Y})\zeta_{kj} + O(h^3), \tag{9}$$

we have

$$\begin{aligned}
 & H(\tilde{\Psi}) - H(\mathbf{K}_{s-2}) \\
 &= h \sum_{j,k=1}^m \left\{ \mathbf{g}'_j(\tilde{\Psi}) \mathbf{g}_k(\tilde{\Psi}) - \mathbf{g}'_j(\mathbf{K}_{s-2}) \mathbf{g}_k(\mathbf{K}_{s-2}) \right\} \zeta_{kj} + O(h^3) = h^2 \mathbf{r}_1 + O(h^3),
 \end{aligned}$$

where $\mathbf{r}_1 = \sum_{j,k=1}^m \left\{ \mathbf{g}''_j(\mathbf{y}_n) [\mathbf{a}, \mathbf{g}_k(\mathbf{y}_n)] + \mathbf{g}'_j(\mathbf{y}_n) \mathbf{g}'_k(\mathbf{y}_n) \mathbf{a} \right\} \zeta_{kj}$. Here, we use the notation $\mathbf{g}''_j(\mathbf{y})[\cdot, \cdot]$ for the second derivative (a symmetric bilinear form) of \mathbf{g}_j at \mathbf{y} . Similarly, since we can rewrite $\tilde{\mathbf{H}}(\mathbf{Y}_1, \mathbf{Y}_2)$ as

$$\begin{aligned}
 & \tilde{\mathbf{H}}(\mathbf{Y}_1, \mathbf{Y}_2) \\
 &= \sqrt{h} \sum_{j=1}^m \mathbf{g}_j(\mathbf{Y}_1) \xi_j + \frac{h^{3/2}}{4} \sum_{j,k,l=1}^m \mathbf{g}''_j(\mathbf{Y}_1) [\mathbf{g}_k(\mathbf{Y}_2), \mathbf{g}_l(\mathbf{Y}_2)] \chi_k \chi_l \xi_j \\
 &+ \frac{h^{5/2}}{96} \sum_{j,k,l,q,r=1}^m \mathbf{g}^{(4)}_j(\mathbf{Y}_1) [\mathbf{g}_k(\mathbf{Y}_2), \mathbf{g}_l(\mathbf{Y}_2), \mathbf{g}_q(\mathbf{Y}_2), \mathbf{g}_r(\mathbf{Y}_2)] \chi_k \chi_l \chi_q \chi_r \xi_j \\
 &+ O(h^3),
 \end{aligned}$$

we have $\tilde{\mathbf{H}}(\tilde{\Psi}, \tilde{\Psi}) - H(\hat{\mathbf{K}}_{s-1}, \mathbf{K}_{s-2}) = h^{5/2} \mathbf{r}_2 + O(h^3)$, where

$$\mathbf{r}_2 = \sum_{j=1}^m \left\{ \mathbf{g}'_j(\mathbf{y}_n) \mathbf{b} + \frac{1}{2} \sum_{k,l=1}^m \mathbf{g}''_j(\mathbf{y}_n) [\mathbf{g}_k(\mathbf{y}_n), \mathbf{g}_l(\mathbf{y}_n) \mathbf{a}] \chi_k \chi_l \right\} \xi_j.$$

Here, we use the notation $\mathbf{g}^{(4)}_j(\mathbf{y})[\cdot, \cdot, \cdot, \cdot]$ for the fourth derivative (a symmetric tetra-linear form) of \mathbf{g}_j at \mathbf{y} . Thus, from $\Phi_h(\Psi_{\frac{h}{2}}(\mathbf{y}_n)) - \mathbf{K}_s^* = h^2 \mathbf{r}_1 + h^{5/2} \mathbf{r}_2 + O(h^3)$, we have

$$\begin{aligned}
 & G(\Phi_h(\Psi_{\frac{h}{2}}(\mathbf{y}_n))) - G(\mathbf{K}_s^*) \\
 &= G'(\mathbf{y}_n) (h^2 \mathbf{r}_1 + h^{5/2} \mathbf{r}_2) + h^{5/2} G''(\mathbf{y}_n) \left[\sum_{j=1}^m \mathbf{g}_j(\mathbf{y}_n) \xi_j, \mathbf{r}_1 \right] + O(h^3).
 \end{aligned}$$

Consequently, we obtain $E[G(\Phi_h(\Psi_{\frac{h}{2}}(\mathbf{y}_n)))] - E[G(\mathbf{K}_s^*)] = O(h^3)$ since $E[\mathbf{r}_1] = E[\mathbf{r}_2] = E[\xi_j \mathbf{r}_1] = 0$ ($j = 1, 2, \dots, m$).

As a sufficient condition for (1) in Theorem 1, it is known that the following two inequalities hold for all sufficiently small $h > 0$:

$$\|E[\mathbf{y}_{n+1} - \mathbf{y}_n \mid \mathbf{y}_n]\| \leq C(1 + \|\mathbf{y}_n\|)h, \quad \|\mathbf{y}_{n+1} - \mathbf{y}_n\| \leq X_n(1 + \|\mathbf{y}_n\|)\sqrt{h},$$

where C is a positive constant and X_n is a random variable that has moments of all orders [26, p. 102]. The smoothness and global Lipschitzness of \mathbf{g}_j ($j = 1, 2, \dots, m$) imply $\|\mathbf{g}'_j(\mathbf{y}) \mathbf{g}_k(\mathbf{y})\| \leq C(1 + \|\mathbf{y}\|)$ for a constant $C > 0$, whereas the global Lipschitzness implies $\|\mathbf{g}_j(\mathbf{y})\| \leq C(1 + \|\mathbf{y}\|)$. From these facts as well as (9) and

$$\|\Psi_{\frac{h}{2}}(\mathbf{y}) - \mathbf{y}\| \leq C_1 \|\mathbf{f}(\mathbf{y})\|h \leq C_2(1 + \|\mathbf{y}\|)h$$

for constants $C_1, C_2 > 0$, we can see that the two inequalities requested above hold for (7). Consequently, (7) is of weak order 2 for (1) by Theorem 1. □

3.3 Stabilized Methods with High Precision

As we have seen above, the split SROCK2 methods achieve weak order 2 for any α . In this section we decide an appropriate value for the parameter α on the basis of a linear error analysis.

Let us consider the linear scalar SDE given by

$$dy(t) = \lambda y(t)dt + \sigma y(t)dW(t), \quad 0 \leq t \leq T_0 < 1, \quad y(0) = y_0, \quad (10)$$

where $y_0 \neq 0$ with probability one (w.p.1) and $\lambda, \sigma \in \mathbb{R}$. When a one-step method with a step size h is applied to (10), we generally have $y_{n+1} = R(h, \lambda, \sigma, \eta) y_n$, where R is an amplification factor and η is a vector whose components are random variables appearing in the method. In [23], the authors have considered the following errors of R for a method with $h = T_0/2^k$ (k is a positive integer):

$$\bar{\mathcal{E}}(k) \stackrel{\text{def}}{=} \{\bar{R}(\lambda T_0/2^k)\}^{2^k} - e^{\lambda T_0}, \quad \hat{\mathcal{E}}(k) \stackrel{\text{def}}{=} \{\hat{R}(\lambda T_0/2^k, \sigma^2 T_0/2^k)\}^{2^k} - e^{(2\lambda + \sigma^2)T_0},$$

where $\bar{R}(\lambda h) \stackrel{\text{def}}{=} E[R]$ and $\hat{R}(\lambda h, \sigma^2 h) \stackrel{\text{def}}{=} E[R^2]$. We investigate these errors for the split SROCK2 methods (7).

If we apply (7) to (10), then we obtain

$$\begin{aligned} &R(h, \lambda, \sigma, \xi_1, \zeta_{11}) \\ &= \left\{ 1 + \theta_\alpha \lambda h + \tau_\alpha \left(\frac{1}{2} \lambda h\right)^2 \right\}^2 \left\{ P_{s-2} \left(\frac{1}{2} \alpha \lambda h\right) \right\}^2 \\ &\quad + \sigma \sqrt{h} \xi_1 (1 + \theta_\alpha \lambda h) \left\{ 1 + \theta_\alpha \lambda h + \tau_\alpha \left(\frac{1}{2} \lambda h\right)^2 \right\} \left\{ P_{s-2} \left(\frac{1}{2} \alpha \lambda h\right) \right\}^2 \\ &\quad + \sigma^2 h \zeta_{11} \left\{ 1 + \theta_\alpha \lambda h + \tau_\alpha \left(\frac{1}{2} \lambda h\right)^2 \right\} \left\{ P_{s-2} \left(\frac{1}{2} \alpha \lambda h\right) \right\}^2, \end{aligned} \quad (11)$$

where $P_0(z) = 1$, $P_1(z) = 1 + \mu_1 z$, and $P_i(z) = (1 + \kappa_i + \mu_i z)P_{i-1}(z) - \kappa_i P_{i-2}(z)$, $i = 2, 3, \dots, s$. Thus, we have

$$\begin{aligned} \hat{R}(\lambda h, \sigma^2 h) &= \left\{ 1 + \theta_\alpha \lambda h + \tau_\alpha \left(\frac{1}{2} \lambda h\right)^2 \right\}^4 \left\{ P_{s-2} \left(\frac{1}{2} \alpha \lambda h\right) \right\}^4 \\ &\quad + \sigma^2 h (1 + \theta_\alpha \lambda h)^2 \left\{ 1 + \theta_\alpha \lambda h + \tau_\alpha \left(\frac{1}{2} \lambda h\right)^2 \right\}^2 \left\{ P_{s-2} \left(\frac{1}{2} \alpha \lambda h\right) \right\}^4 \\ &\quad + \frac{1}{2} (\sigma^2 h)^2 \left\{ 1 + \theta_\alpha \lambda h + \tau_\alpha \left(\frac{1}{2} \lambda h\right)^2 \right\}^2 \left\{ P_{s-2} \left(\frac{1}{2} \alpha \lambda h\right) \right\}^4. \end{aligned}$$

Suppose that the solution to (10) is asymptotically stable in the MS. Then $2\lambda + \sigma^2 < 0$ holds. Let us define r by σ^2/λ and consider the case of $-2 < r \leq 0$ in what follows. When an $\alpha > 0$ is given to (7), for each $r \in (-2, 0]$ $\hat{\mathcal{E}}(0)$ oscillates when λT_0 moves in an interval $[l(r), 0]$ (see Figure 1), where $l(r)$ is a negative number such that $\hat{R}(l(r), r l(r)) = 1$.

If we choose $r_0 \in (-2, 0]$, then

$$\max_{\substack{l(r) < \lambda T_0 < 0 \\ r_0 \leq r \leq 0}} \left| \hat{\mathcal{E}}(0) \right| \quad (12)$$

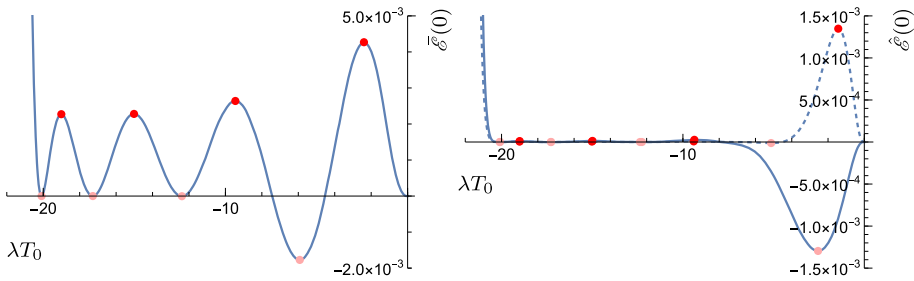


Fig. 1 Error versus λT_0 for the expectation and the second moment of the amplification factor of the split SROCK2 method with $s = 6$, $h = T_0$ and $\alpha = 2.73$ when $r = -1/2$ (solid lines) or $r = 0$ (dashed lines)

Table 1 Optimal value of α for the split SROCK2 methods

s	3	4	5	7	8, 9, 10	11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21
α	2.63	2.73	2.74	2.73	2.72	2.71
s	22, 24, 26, 28, 30, 32, 35, 38, 41, 45, 49, 53, 58, 63, 68, 74, 80, 87, 95					
α	2.72					
s	104, 114, 125, 137, 150, 165, 182, 200					
α	2.72					

gives a uniform error of the second moment of the amplification factor for the numerical method in $(l(r), 0) \times [r_0, 0]$. We decide the value of α , say $\hat{\alpha}$, such that the uniform error for a given $\hat{\alpha}$ becomes equal to

$$\min_{\alpha \geq 1} \left\{ \max_{\substack{l(r) < \lambda T_0 < 0 \\ r_0 \leq r \leq 0}} |\hat{\mathcal{E}}(0)| \right\}.$$

Here, note that if $\alpha < 1$, then the good stability properties of (3) themselves are violated.

As seen in the expression of $\hat{\mathcal{E}}(k)$, we cannot expect that methods based on (3) including other methods introduced later, have a small error $\hat{\mathcal{E}}(k)$ in general for a small $r \in (-2, 0]$, which leads to a large $\sigma^2 = r\lambda > 0$. For this, we set r_0 at $-1/2$ and seek an optimal value of α for (7) with the stage number s . The reason why we have chosen $r_0 = -1/2$, not -2 , will also be mentioned in detail later. For example, when $s = 6$, we have $\alpha = 2.73$ as an optimal value for (7). For other values of s , see Table 1. Figure 1 shows the errors of (7) with $\alpha = 2.73$. Local maxima and local minima of the errors are indicated by points. In the right-hand plot, we set $r = -1/2$ or 0. Note that $\hat{\mathcal{E}}(k)$ depends on r , but $\bar{\mathcal{E}}(k)$ does not.

Let us investigate the errors of (7) with optimal α whose value is given in Table 1. Table 2 gives (12), that is, the uniform errors of the methods with $h = T_0$. The notation \tilde{l}_s denotes the abscissa of the local minimal point of $\hat{\mathcal{E}}(0)$ that is located in the most left among the local minimal points of $\hat{\mathcal{E}}(0)$. This gives auxiliary information about stability properties for the methods. From the change of values of \tilde{l}_s , we can observe that the stability interval becomes longer as the stage number s increases. This is because the split SROCK2 methods inherit the stability properties of (3). The computational cost for the stabilization procedures depends on the number of evaluations on f . The split SROCK2 method with s stages evaluates it $2s$ times in one step. The table gives $|\tilde{l}_s|/(2s)^2$, which indicates the efficiency of the stabilization

Table 2 Uniform error of the split SROCK2 methods with optimal α

s	Error	\tilde{l}_s	$ \tilde{l}_s /(2s)^2$	s	Error	\tilde{l}_s	$ \tilde{l}_s /(2s)^2$
3	3.9×10^{-3}	-3.6	0.100	49	8.4×10^{-4}	-1428.5	0.149
6	1.3×10^{-3}	-20.1	0.140	104	8.3×10^{-4}	-6440.4	0.149
12	9.6×10^{-4}	-84.5	0.147	150	8.3×10^{-4}	-13399.3	0.149
24	8.6×10^{-4}	-341.6	0.148	200	8.3×10^{-4}	-23822.1	0.149

procedures. From the table, we can see that the value of $|\tilde{l}_s|/(2s)^2$ is approximately 0.149 when s is large.

The first term in $\tilde{\mathcal{E}}(0)$ is equal to the first term in the right-hand side of (11) if $h = T_0$. In the term, the polynomial $\{1 + \theta_\alpha \lambda h + \tau_\alpha (\lambda h/2)^2\} P_{s-2}(\alpha \lambda h/2)$ for $h = T_0$ approximates $e^{\lambda T_0/2}$, and as a result, its square approximates $e^{\lambda T_0}$ in the interval $[-1/2, 0]$ such as the error is shown in the left-hand plot of Fig. 1.

Also in $\hat{\mathcal{E}}(0)$, the square approximates $e^{\lambda T_0}$ in the interval. In addition, the other polynomials $(1 + \theta_\alpha \lambda h) P_{s-2}(\alpha \lambda h/2)$ and $P_{s-2}(\alpha \lambda h/2)$ for $h = T_0$ approximate $e^{\lambda T_0/2}$ less precisely than the above mentioned polynomial when $|\lambda T_0| \ll 1$, but their squares serve together with the square of the above mentioned polynomial to overcome the increase by $\sigma^2 h$ and $(\sigma^2 h)^2/2$ for $h = T_0$. As a result, the error $\hat{\mathcal{E}}(0)$ behaves as shown in the right-hand plot of Fig. 1.

Remark 2 The naïve split method (8) gives \hat{R} as follows:

$$\hat{R}(\lambda h, \sigma^2 h) = \left\{ 1 + \sigma^2 h + \frac{1}{2} (\sigma^2 h)^2 \right\} \left\{ 1 + \theta_\alpha \lambda h + \tau_\alpha \left(\frac{1}{2} \lambda h \right)^2 \right\}^4 \left\{ P_{s-2} \left(\frac{1}{2} \alpha \lambda h \right) \right\}^4.$$

Although the differences from \hat{R} for (7) are only the replacements of $(1 + \theta_\alpha \lambda h)^2$ or 1 with $\{1 + \theta_\alpha \lambda h + \tau_\alpha (\lambda h/2)^2\}^2$, the replacements strengthen the increase in \hat{R} when $|\lambda h|$ is large. Thus, (8) is also of weak order 2, but unfortunately it cannot have good stability properties especially when s is large. In addition, note that making a change in one $\Psi_{h/2}$ in (8) leads to a replacement of $\{1 + \theta_\alpha \lambda h + \tau_\alpha (\lambda h/2)^2\}^2 \times \{P_{s-2}(\alpha \lambda h/2)\}^2$ in \hat{R} .

4 Other Stabilized Explicit Methods of Weak Order 2

As competitors to the split SROCK2 methods, we first state other stabilized explicit methods based on (3). Next, we introduce an exponential RK method of weak order 2 for SDEs [23]. When the dimension of the SDEs is very large, the exponential RK method needs techniques in order to calculate the product of a matrix exponential function and a vector efficiently. In the end of the section, we consider the implementation of Krylov subspace projection techniques related to a matrix exponential function φ_2 .

4.1 Other Methods Based on the ROCK2 Methods with a Parameter α

By combining (2) and (3), Abdulle, Vilmart, and Zygalakis [4] have proposed the following class of methods:

$$\begin{aligned}
 \mathbf{K}_i &= \mu_i \alpha h \mathbf{f}(\mathbf{K}_{i-1}) + (\kappa_i + 1) \mathbf{K}_{i-1} - \kappa_i \mathbf{K}_{i-2}, \quad i = s - 1, s, \\
 \mathbf{K}_{s-1}^* &= \mathbf{K}_{s-2} + 2\tau_\alpha h \mathbf{f}(\mathbf{K}_{s-2}) + \sqrt{h} \sum_{j=1}^m \mathbf{g}_j(\mathbf{K}_s) \xi_j, \\
 \mathbf{y}_{n+1} &= \mathbf{K}_{s-2} + \left(2\theta_\alpha - \frac{1}{2}\right) h \mathbf{f}(\mathbf{K}_{s-2}) + \frac{h}{2} \mathbf{f}(\mathbf{K}_{s-1}^*) + \mathbf{H}(\mathbf{K}_s) \\
 &\quad + \tilde{\mathbf{H}}(\mathbf{K}_{s-1}, \mathbf{K}_s),
 \end{aligned}
 \tag{13}$$

where $K_i, i = s - 3, s - 2$, are the same as those in (3), and have determined the value of α such that $\mathbf{K}_{s-1} = \mathbf{y}_n + (h/2)\mathbf{f}(\mathbf{y}_n) + O(h^2)$ holds. For the value of α , (13) achieves weak order 2. Note Lemma 1. The methods are called the SROCK2 methods. When (13) is applied to (10), we have

$$\begin{aligned}
 \hat{R}(\lambda h, \sigma^2 h) &= \{1 + 2\theta_\alpha \lambda h + \tau_\alpha (\lambda h)^2\}^2 \{P_{s-2}(\alpha \lambda h)\}^2 \\
 &\quad + \sigma^2 h \left\{P_{s-1}(\alpha \lambda h) + \frac{1}{2} \lambda h P_s(\alpha \lambda h)\right\}^2 \\
 &\quad + \frac{1}{2} (\sigma^2 h)^2 \{P_s(\alpha \lambda h)\}^2.
 \end{aligned}
 \tag{14}$$

For a given s , (7) contains almost twice as many computational procedures as (13). For this, as a competitor to (7) with $h = T_0$, we choose (13) with $h = T_0/2$ when the both methods have the same stage number s . Figure 2 shows the errors of (13) when $s = 6$ and $k = 1$ in a similar manner to Fig. 1. From the comparison of Figs. 1 and 2, we have the following remarks. First, the split SROCK2 method with $h = T_0$ can achieve very small errors, compared with the SROCK2 method with $h = T_0/2$. Second, the SROCK2 method with $h = T_0/2$ has a much longer stability interval than the split SROCK2 method with $h = T_0$. Also for other stages, this stability property is a strength of the SROCK2 methods, whereas low precision is a weakness of them.

Incidentally, Table 3 gives the uniform errors of the SROCK2 methods with $h = T_0$. As the methods are of weak order 2, the replacement of $h = T_0$ with $h = T_0/2$ is expected to

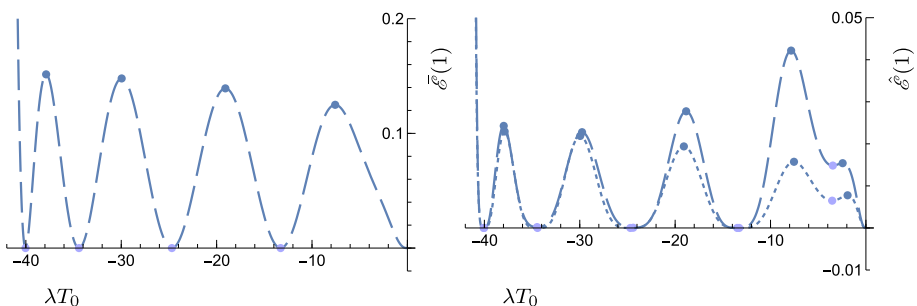


Fig. 2 Error versus λT_0 for the expectation and the second moment of the amplification factor of the SROCK2 method ($s = 6, k = 1, \alpha = 1.37$) with $h = T_0/2$ when $r = -1/2$ (long-dashed lines) or $r = 0$ (dashed lines)

Table 3 Uniform errors of the SROCK2 methods

s	Error	\tilde{l}_s	$ \tilde{l}_s /(s + 1)^2$	s	Error	\tilde{l}_s	$ \tilde{l}_s /(s + 1)^2$
3	9.7×10^{-1}	-5.5	0.344	49	4.6×10^{-2}	-1070.9	0.428
6	2.0×10^{-1}	-20.0	0.408	104	4.8×10^{-2}	-4721.1	0.428
12	6.1×10^{-2}	-71.7	0.424	150	4.9×10^{-2}	-9761.9	0.428
24	4.2×10^{-2}	-267.5	0.428	200	5.0×10^{-2}	-17294.8	0.428

decrease errors by a factor of 1/4 approximately. Even if we take this into account, we can say that the split SROCK2 methods with $h = T_0$ have smaller errors than the SROCK2 methods with $h = T_0/2$. Here, note that the value of errors in the table indicates a local maximum for $\hat{\mathcal{E}}(0)$, not for $\hat{\mathcal{E}}(1)$. The SROCK2 method with s stages evaluates the function f $s + 1$ times in one step. The table gives $|\tilde{l}_s|/(s + 1)^2$, which indicates the efficiency of the stabilization procedures. From the table, we can see that the value of $|\tilde{l}_s|/(s + 1)^2$ is approximately 0.428 when s is large.

The first term in $\bar{\mathcal{E}}(1)$ for (13) is equal to the first term in the right-hand side of (14) if $h = T_0/2$. Thus, $\bar{\mathcal{E}}(0)$ for (7) and $\bar{\mathcal{E}}(1)$ for (13) have the same expression except the value of α . The difference in α makes the difference of the left-hand plots in Figs. 1 and 2. In addition, in $\hat{\mathcal{E}}(1)$ for (13), the polynomials $P_{s-1}(\alpha\lambda h) + \lambda h P_{s-2}(\alpha\lambda h)/2$ and $P_s(\alpha\lambda h)$ for $h = T_0/2$ approximate $e^{\lambda T_0/2}$, and the fourth powers of them serve to overcome the increase by $(\sigma^2 h)^2$ and $(\sigma^2 h)^4/4$ for $h = T_0/2$. Remember that $\hat{\mathcal{E}}(0)$ for (7) has lower order terms with respect to $\sigma^2 h$. This difference as well as the difference in α makes the difference of the right-hand plots in Figs. 1 and 2.

In Fig. 2, the following are also remarkable. When λT_0 is sufficiently small, the first term is dominant over the second term $e^{\lambda T_0}$ in $\bar{\mathcal{E}}(k)$. For example, since $e^{-3} \approx 5.0 \times 10^{-2}$, in the left-hand plot of Fig. 2 the curve shows almost only the first term in $\bar{\mathcal{E}}(1)$ when $\lambda T_0 < -3$. Similarly, when λT_0 is sufficiently small and r is close to 0, the first term is dominant over the second term $e^{(2\lambda + \sigma^2)T_0} = e^{(2+r)\lambda T_0}$ in $\hat{\mathcal{E}}(k)$. For example, since $e^{(2+r)\lambda T_0} = e^{-6} \approx 2.5 \times 10^{-3}$ if $r = -1/2$ and $\lambda T_0 = -4$, in the right-hand plot of Fig. 2 the curve shows almost only the first term in $\hat{\mathcal{E}}(1)$ when $\lambda T_0 < -4$.

Remark 3 On the basis of the idea of dense output in RK methods, due to Lemma 1 we can obtain another class of weak second order methods based on the ROCK2 methods with a free parameter α . However, unfortunately the methods lead to much worse precision than the split SROCK2 methods and much worse stability properties than the SROCK2 methods.

We are concerned with weak second order methods, but there is a class of excellent stabilized explicit methods of weak order one [5]. Let us call them the SK-ROCK methods and give a brief remark on the methods.

Remark 4 The SK-ROCK methods have a free parameter η to obtain good stability properties. In fact, they have enjoyed excellent stability properties in some numerical experiments [5], when the parameter has a typical value, $\eta = 0.05$. However, they suffer from the error of the amplification factor. When $s = 6$ and $\eta = 0.05$, we obtain the left-hand plot in Fig. 3. Even if we increase the value of η , they still suffer from the error, especially for large $\lambda T_0 < 0$. For example, see the right-hand plot in the figure.

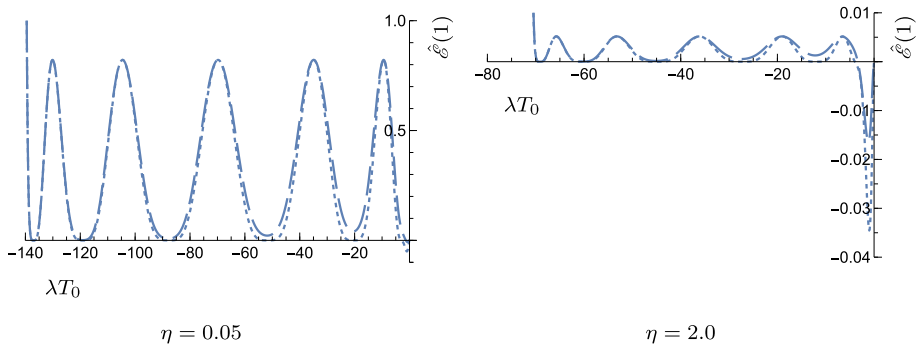


Fig. 3 Error versus λT_0 for the second moment of the amplification factor of the SK-ROCK methods ($s = 6, k = 1, \eta = 0.05, 2.0$) with $h = T_0/2$ when $r = -1/2$ (long-dashed lines) or $r = 0$ (dashed lines)

4.2 Exponential RK Methods and the Implementation of Krylov Subspace Projection Techniques for Matrix Exponential Functions

If the drift term is given as a semilinear drift term $Ay + f_0(y)$ in (1), where A is a $d \times d$ matrix and f_0 is an \mathbb{R}^d -valued nonlinear function on \mathbb{R}^d , then we can apply the following stochastic exponential RK method to (1):

$$\begin{aligned}
 y_{n+1} = & Y_1 + \frac{h}{3}\varphi_2(hA) \left\{ 4f_0\left(Y_2 + \frac{(6 + \sqrt{6})\sqrt{h}}{10} \sum_{j=1}^m g_j(Y_2)\xi_j\right) \right. \\
 & + f_0\left(Y_3 + \frac{(3 - 2\sqrt{6})\sqrt{h}}{5} \sum_{j=1}^m g_j(Y_2)\xi_j\right) \\
 & \left. - 5f_0(y_n) \right\} + e^{\frac{h}{2}A} \left(H(Y_2) + \tilde{H}(Y_2, Y_2) \right),
 \end{aligned} \tag{15}$$

where

$$\begin{aligned}
 Y_1 &= e^{hA} y_n + h\varphi_1(hA)f_0(y_n), \quad Y_2 = e^{\frac{h}{2}A} y_n + \frac{h}{2}\varphi_1\left(\frac{h}{2}A\right) f_0(y_n), \\
 Y_3 &= Y_1 + 2h\varphi_2\left(\frac{h}{2}A\right) \{f_0(Y_2) - f_0(y_n)\} + 2h\varphi_2(hA) \{f_0(Y_2) - f_0(y_n)\}.
 \end{aligned}$$

The matrix exponential functions are defined by

$$\varphi_1(Z) \stackrel{\text{def}}{=} Z^{-1}(e^Z - I) = \sum_{k=1}^{\infty} \frac{1}{k!} Z^{k-1}, \quad \varphi_2(Z) \stackrel{\text{def}}{=} Z^{-2}(e^Z - I - Z) = \sum_{k=2}^{\infty} \frac{1}{k!} Z^{k-2},$$

where Z, I stand for a $d \times d$ matrix, the $d \times d$ identity matrix, respectively. As the method is of weak order 2 and deterministic order 3 [23], let us call this the SERKW2D3 method in what follows.

If A is diagonalizable and its dimension is not large, for example, 100, then the matrix exponential functions in the method are easy to calculate through a similarity transformation based on eigenvectors [27, p. 23]. In addition, once they are calculated for a given step size h , we can use them for all trajectories. Thus, in this case the SERKW2D3 method can be an effective and powerful method. However, if the dimension of A is very large, the situation

drastically changes. The computation based on the similarity transformation is not useful. We need another computational method for the matrix exponential functions.

For a large sparse matrix A , let us suppose that A is symmetric since we can rewrite f as $(A + A^T)/2 + f_0(\mathbf{y}) + (A - A^T)/2$. The symmetric property will lead to computational savings for a matrix exponential function times a vector.

In order to calculate Y_3 efficiently and precisely, let us derive a recursive formula for φ_2 . From the definition of φ_1 and φ_2 , we have

$$\begin{aligned} (t + \Delta t)^2 \varphi_2((t + \Delta t)A)\mathbf{q} &= \int_0^{t+\Delta t} (t + \Delta t - s)\varphi_1((t + \Delta t - s)A)\mathbf{q}ds \\ &= e^{\Delta t A} t^2 \varphi_2(tA)\mathbf{q} + t \Delta t \varphi_1(\Delta t A)\mathbf{q} + (\Delta t)^2 \varphi_2(\Delta t A)\mathbf{q} \end{aligned}$$

for $t, \Delta t > 0$, where $\mathbf{q} = \mathbf{f}_0(Y_2) - \mathbf{f}_0(y_n)$. Similarly, for any $\mathbf{w} \in \mathbb{R}^d$,

$$\begin{aligned} \Delta t \varphi_1(\Delta t A)(A\mathbf{w} + \mathbf{q} - t^{-1}\mathbf{w}) &= e^{\Delta t A} \mathbf{w} + \Delta t \varphi_1(\Delta t A)(\mathbf{q} - t^{-1}\mathbf{w}) - \mathbf{w}, \\ (\Delta t)^2 \varphi_2(\Delta t A)(A\mathbf{w} + \mathbf{q}) &= \Delta t \varphi_1(\Delta t A)\mathbf{w} - \Delta t \mathbf{w} + (\Delta t)^2 \varphi_2(\Delta t A)\mathbf{q}. \end{aligned}$$

If we set $\mathbf{w} = t\varphi_2(tA)\mathbf{q}$, then, from the above equalities we have

$$\begin{aligned} (t + \Delta t)^2 \varphi_2((t + \Delta t)A)\mathbf{q} &= (\Delta t)^2 \varphi_2(\Delta t A)(A\mathbf{w} + \mathbf{q}) + t \Delta t \varphi_1(\Delta t A)(A\mathbf{w} + \mathbf{q}) - \Delta t \varphi_1(\Delta t A)\mathbf{w} + (t + \Delta t)\mathbf{w}. \end{aligned}$$

We can calculate $2h\varphi_2((h/2)A)\mathbf{q}$ firstly and $2h\varphi_2(hA)\mathbf{q}$ secondly in Y_3 , using this relationship repeatedly. Note that the Δt does not always need to be fixed in the calculations of Y_3 .

Our next concern is to calculate $(\Delta t)^2 \varphi_2(\Delta t A)(A\mathbf{w} + \mathbf{q})$ and $\Delta t \varphi_1(\Delta t A)(A\mathbf{w} + \mathbf{q})$. For this, we adopt Krylov subspace projection techniques. Let us set $\mathbf{u} = A\mathbf{w} + \mathbf{q}$ and denote by $V_{\rho+1} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{\rho+1}]$ and $H_\rho = [h_{i,j}]$, the orthonormal basis of the Krylov subspace $\text{Span}\{\mathbf{u}, A\mathbf{u}, \dots, A^\rho \mathbf{u}\}$ and the ρ -dimensional symmetric tridiagonal matrix resulting from Lanczos process, where ρ is a small positive integer compared to d . For our purpose, Theorem 2 in [28] gives the following approximation

$$(\Delta t)^2 \varphi_2(\Delta t A)\mathbf{u} \approx \beta V_{\rho+1} \begin{bmatrix} \tilde{\Phi}_{2,\rho} \mathbf{e}_1 \\ h_{\rho+1,\rho} \mathbf{e}_\rho^\top \tilde{\Phi}_{3,\rho} \mathbf{e}_1 \end{bmatrix} \tag{16}$$

for a sufficiently small $\Delta t > 0$, where $\beta = \|\mathbf{u}\|$, $h_{\rho+1,\rho} = \|A\mathbf{v}_\rho - \sum_{i=\rho-1}^0 h_{i,\rho} \mathbf{v}_i\|$, $\tilde{\Phi}_{i,\rho} = (\Delta t)^i \varphi_i(\Delta t H_\rho)$, and \mathbf{e}_i denotes the ρ -dimensional unit vector whose i -th component is 1. Note that if the minimal degree of \mathbf{u} is some integer $l (\leq \rho)$, then an invariant subspace is found and the above approximation is replaced with the equality $(\Delta t)^2 \varphi_2(\Delta t A)\mathbf{u} = \beta V_l \tilde{\Phi}_{2,l} \tilde{\mathbf{e}}_1$, where $\tilde{\mathbf{e}}_1$ is the l -dimensional unit vector whose 1st component is 1 [28].

Now, our next concern is to calculate $\tilde{\Phi}_{2,\rho} \mathbf{e}_1$ and $\tilde{\Phi}_{3,\rho} \mathbf{e}_1$ efficiently. Theorem 1 in [28] gives the following equality:

$$\exp(\Delta t \tilde{H}_{\rho+4}) = \begin{bmatrix} \exp(\Delta t H_\rho) & \tilde{\Phi}_{1,\rho} \mathbf{e}_1 & \tilde{\Phi}_{2,\rho} \mathbf{e}_1 & \cdots & \tilde{\Phi}_{4,\rho} \mathbf{e}_1 \\ \mathbf{0}_\rho^\top & 1 & \frac{\Delta t}{1!} & \cdots & \frac{(\Delta t)^3}{3!} \\ \mathbf{0}_\rho^\top & 0 & 1 & \ddots & \vdots \\ \mathbf{0}_\rho^\top & 0 & 0 & \ddots & \frac{\Delta t}{1!} \\ \mathbf{0}_\rho^\top & 0 & 0 & \cdots & 1 \end{bmatrix}, \tag{17}$$

where $\mathbf{0}_\rho$ denotes the ρ -dimensional zero column vector and

$$\tilde{H}_{\rho+4} = \begin{bmatrix} H_\rho & \mathbf{e}_1 & \mathbf{0}_\rho & \mathbf{0}_\rho & \mathbf{0}_\rho \\ \mathbf{0}_\rho^\top & 0 & 1 & 0 & 0 \\ & & 0 & 1 & 0 \\ & & & 0 & 1 \\ 0 & & & & 0 \end{bmatrix} \in \mathbb{R}^{(\rho+4) \times (\rho+4)}.$$

This shows that $\exp(\Delta t \tilde{H}_{\rho+4})$ gives $\tilde{\Phi}_{2,\rho} \mathbf{e}_1$ and $\tilde{\Phi}_{3,\rho} \mathbf{e}_1$, and it is also helpful for us to determine such a small Δt that (16) is valid.

The procedure to determine Δt is given as follows [28]. When we set two errors by $err1 = \beta |h_{\rho+1,\rho} \mathbf{e}_1^\top \tilde{\Phi}_{3,\rho} \mathbf{e}_1|$ and $err2 = \beta |h_{\rho+1,\rho} \mathbf{e}_1^\top \tilde{\Phi}_{4,\rho} \mathbf{e}_1| \|A \mathbf{v}_{\rho+1}\|$ for an initial Δt , we have an error estimation err_{loc} by Algorithm 2 in [28]. We repeatedly carry out a procedure to obtain an acceptable step size Δt . For details, see [28]. Incidentally, in our numerical experiments we calculate $\exp(\Delta t \tilde{H}_{\rho+4})$ in (17) by a method based on Padé approximations [28].

In (15), we can calculate the product of $(h/3)\varphi_2(hA)$ and a vector in a similar way. We can also calculate the product of the matrix exponential or the matrix exponential function φ_1 and a vector in a similar but simpler way.

Remark 5 For the product of the matrix exponential and a vector in the case that the matrix is of moderate dimension and is not sparse, some researchers have proposed more efficient algorithms than the Krylov subspace projection techniques. See [6, 18] and references therein.

5 Numerical Experiments

We derived the split SROCK2 methods of weak order 2 that are expected to achieve better precision than the SROCK2 methods for SDEs with not very large noise. In order to confirm the performance of the methods, we investigate the expectation and the second moment of the solution of SDEs in our numerical experiments.² In all numerical experiments, we will use the Mersenne twister algorithm [24] to generate pseudorandom numbers.

The first example is the following 2-dimensional damped nonlinear Kubo oscillator [11, 32]

$$d\mathbf{y}(t) = \mathbf{f}(\mathbf{y}(t))dt + \sum_{j=1}^2 \mathbf{g}_j(\mathbf{y}(t))dW_j(t), \quad \mathbf{y}(0) = \mathbf{y}_0 \text{ (w.p.1)}, \tag{18}$$

where $t \in [0, 1]$,

$$\begin{aligned} \mathbf{f}(\mathbf{y}) &= \lambda \begin{bmatrix} 0 & -1 \\ 1 & \alpha \end{bmatrix} \mathbf{y} + \begin{bmatrix} 0 & -\frac{1}{5}(y_1 + y_2)^5 \\ \frac{1}{5}(y_1 + y_2)^5 & 0 \end{bmatrix} \mathbf{y}, \\ \mathbf{g}_j(\mathbf{y}) &= \sigma_j \begin{bmatrix} 0 & -1 \\ 1 & \beta_j \end{bmatrix} \mathbf{y} + \begin{bmatrix} 0 & -q_j(\mathbf{y}) \\ q_j(\mathbf{y}) & 0 \end{bmatrix} \mathbf{y}, \end{aligned}$$

$q_1(\mathbf{y}) = 0$, $q_2(\mathbf{y}) = \frac{1}{3}(y_1 + y_2)^3$, and $\lambda, \sigma_j, \alpha, \beta_j \in \mathbb{R}$. As we do not know the exact expectation and second moment of the solution, let us seek approximations to them by the SERKW2D3 method with $h = 2^{-7}$. We refer to these approximations instead of the exact

² For an implementation of the methods, we utilize the parameter values in a Fortran code, `rectp.f`, obtained from <http://anmc.epfl.ch/Pdf/srock2.zip>.

expectation and second moment. As another competitor, we add the SSDFMT method, which is a stochastic Strang splitting method of weak order 2 [23].

We set $\lambda = 2$, $\sigma_1 = -\alpha = 1/2$, $\sigma_2 = -\beta_1 = 1/5$, $\beta_2 = -1/10$ as well as $y_0 = [1/2 \ 1/2]^T$ [31], and simulate 1024×10^6 independent trajectories for a given h , and seek a numerical approximation to the expectation of $y(1)$ or to the second moment of each element of $y(1)$, that is, $[E[(y_1(1))^2] \ E[(y_2(1))^2]]^T$. The results are indicated in Fig. 4. As the solution is a vector, the Euclidean norm is used. The solid and long-dashed lines denote the split SROCK2 and SROCK2 methods with $s = 3$, respectively, whereas the dashed and long-dash-dotted lines denote the SSDFMT and SERKW2D3 methods. Here and in what follows, the dotted line is a reference line with slope 2. All the methods show the theoretical order of convergence. The split SROCK2 method with $s = 3$, the SSDFMT method and the SERKW2D3 method show much smaller errors than the SROCK2 method.

The second example is the following 2-dimensional linear SDE [29]

$$d\mathbf{y}(t) = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \mathbf{y}(t)dt + \begin{bmatrix} 0 & \sigma \\ \sigma & 0 \end{bmatrix} \mathbf{y}(t)dW(t), \quad \mathbf{y}(0) = \mathbf{y}_0 \text{ (w.p.1)}, \quad (19)$$

where $t \in [0, 1]$ and $\lambda_1, \lambda_2, \sigma \in \mathbb{R}$. As the matrices in the drift and diffusion terms are not simultaneously diagonalizable if $\lambda_1 \neq \lambda_2$, the SDE cannot be transformed to decoupled scalar SDEs. Thus, the behaviour of errors is predicted to differ from the results in Sect. 3.3. As it is linear, we obtain ODEs with respect to the expectation and second moment of the solution, and we can solve them for the exact expectation and second moment.

Let us set $y_0 = [1 \ 1]^T$, $\lambda_1 = -100$, $\lambda_2 = -1$, and $\sigma = 1/2$. We simulate 1024×10^6 independent trajectories for a given h , and seek a numerical approximation to the expectation of $y(1)$ or to the second moment of each element of $y(1)$. The results are indicated in Fig. 5. In order to solve the SDE numerically stably with reasonable cost when a step size h is given, we set the smallest stage number for each of the methods to solve it stably. As a result, we set $s = 3, 4, 5, 7$, and 10 for the split SROCK2 and SROCK2 methods when $h = 1/32, 1/16, 1/8, 1/4$, and $1/2$, respectively. As in the previous experiment, the solid, long-dashed, and dashed lines denote the split SROCK2, SROCK2, and SSDFMT methods. The results by the SERKW2D3 method are omitted since they are the same as those by the SSDFMT method.

As (19) is a linear SDE, we can predict that the two exponential methods, that is, the SERKW2D3 and SSDFMT methods perform well. In fact, their errors are very small especially in the approximation to the expectation, and the errors seem to come almost from

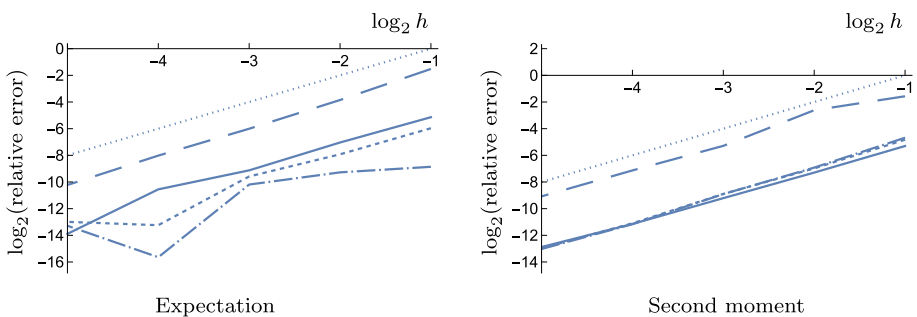


Fig. 4 Log-log plots of the relative error versus h for the expectation and second moment in (18). Solid lines: split SROCK2 ($s = 3$); long-dashed lines: SROCK2 ($s = 3$); dashed lines: SSDFMT; long-dash-dotted lines: SERKW2D3; dotted lines: reference line with slope 2

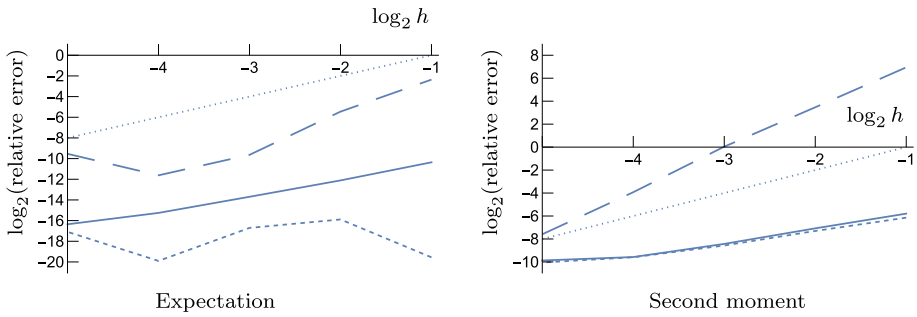


Fig. 5 Log-log plots of the relative error versus h for the expectation and second moment in (19). Solid lines: split SROCK2 ($s = 3, 4, 5, 7, 10$); long-dashed lines: SROCK2 ($s = 3, 4, 5, 7, 10$); dashed lines: SSDFMT; dotted lines: reference line with slope 2

statistical errors. The split SROCK2 methods with several stages follow after them and show smaller errors than the SROCK2 methods in the approximation to the expectation. In contrast to this, the relative error versus h for the second moment is almost the same between the split SROCK2 and the two exponential methods. In addition, the split SROCK2 methods with several stages show much smaller errors than the SROCK2 methods in the approximation to the second moment when h is large.

In this example, the two exponential methods have shown the best performance since the dimension of the SDE is small and especially the drift term has a diagonal matrix. However, when the dimension of SDEs is very large, the methods have high computational cost for matrix exponential functions unless some techniques, such as Krylov subspace projection, are provided in order to calculate them efficiently. If the SDEs come from a spatial discretization of multidimensional stochastic partial differential equations, for example, their dimension can be very large.

A multidimensional stochastic Burgers equation has been studied in [8]. As a third example, we consider the following 2-dimensional stochastic Burgers equation with zero Dirichlet boundary conditions

$$\begin{aligned}
 \frac{\partial u(t, x, y)}{\partial t} &= \gamma \left(\frac{\partial^2 u}{\partial x^2}(t, x, y) + \frac{\partial^2 u}{\partial y^2}(t, x, y) \right) \\
 &\quad + u(t, x, y) \frac{\partial u}{\partial x}(t, x, y) + v(t, x, y) \frac{\partial u}{\partial x}(t, x, y) \\
 &\quad + \beta_1 \frac{\partial^3 \tilde{W}_1}{\partial t \partial x \partial y}(t, x, y), \\
 \frac{\partial v(t, x, y)}{\partial t} &= \gamma \left(\frac{\partial^2 v}{\partial x^2}(t, x, y) + \frac{\partial^2 v}{\partial y^2}(t, x, y) \right) \\
 &\quad + u(t, x, y) \frac{\partial v}{\partial x}(t, x, y) + v(t, x, y) \frac{\partial v}{\partial x}(t, x, y) \\
 &\quad + \beta_2 \frac{\partial^3 \tilde{W}_2}{\partial t \partial x \partial y}(t, x, y), \quad (t, x, y) \in [0, 1/2] \times [0, 1] \times [0, 1], \\
 [u(0, x, y) \ v(0, x, y)] &= [u_0(x, y) \ v_0(x, y)] \text{ (w.p.1),} \\
 (x, y) &\in [0, 1] \times [0, 1],
 \end{aligned}
 \tag{20}$$

where $\tilde{W}_i(t, x, y)$, $i = 1, 2$, are independent Brownian sheets on $[0, 1/2] \times [0, 1] \times [0, 1]$, and where $\gamma > 0$ and $\beta_1, \beta_2 \in \mathbb{R}$ are parameters. We discretize the space intervals by $N + 2$ equidistant points x_i and y_k , $i, k = 0, 1, \dots, N + 1$, and define an \mathbb{R}^{N^2} -vector valued function $U(t)$ by $[\tilde{u}_1(t)^\top \tilde{u}_2(t)^\top \dots \tilde{u}_N(t)^\top]^\top$, where $\tilde{u}_i(t) = [u(t, x_1, y_i) \ u(t, x_2, y_i) \ \dots \ u(t, x_N, y_i)]^\top$. In addition, we define another vector function $V(t)$ similarly, by replacing $\tilde{u}_i(t)$ and $u(t, \cdot, \cdot)$ with $\tilde{v}_i(t)$ and $v(t, \cdot, \cdot)$, respectively. Then, the application of finite differences [4, 13] to (20) yields the $2N^2$ -dimensional SDE with diagonal and additive noise

$$\begin{aligned} dU(t) &= \left\{ \gamma(N + 1)^2 AU(t) + \frac{N + 1}{2} [U(t) * B_1 U(t) + V(t) * B_2 U(t)] \right\} dt \\ &\quad + \beta_1(N + 1)dW_1(t), \\ dV(t) &= \left\{ \gamma(N + 1)^2 AV(t) + \frac{N + 1}{2} [U(t) * B_1 V(t) + V(t) * B_2 V(t)] \right\} dt \\ &\quad + \beta_2(N + 1)dW_2(t), \end{aligned} \tag{21}$$

where $W_1(t)$, $W_2(t)$ are N^2 -dimensional independent standard Wiener processes, and where A, B_1, B_2 are $N^2 \times N^2$ matrices given in the following and vectors use $*$ as component wise multiplication. In order to express A, B_1, B_2 concretely, let us introduce the notation

$$\text{tridiag}(C_1, D, C_2) = \begin{bmatrix} D & C_2 & & & \\ C_1 & D & C_2 & & \\ & \ddots & \ddots & \ddots & \\ & & C_1 & D & C_2 \\ & & & C_1 & D \end{bmatrix} \in \mathbb{R}^{(kN) \times (kN)}$$

for $C_1, D, C_2 \in \mathbb{R}^{k \times k}$, which can be used to give a tridiagonal matrix or a block tridiagonal matrix when $k = 1$ or $k = N$, respectively. In (21), $A = \text{tridiag}(I, T, I)$, $B_1 = \text{tridiag}(O, S, O)$, $B_2 = \text{tridiag}(-I, O, -I)$, where $T = \text{tridiag}(1, -4, 1)$, $S = \text{tridiag}(-1, 0, 1)$, and I and O denote the $N \times N$ identity and zero matrices. If we use expressions in Remark 1 for (21), then $H(y)$ is a $2N^2$ -dimensional zero vector, and $\tilde{H}(y, z)$ is a $2N^2$ -dimensional vector whose j -th component is $\beta_1(N + 1)\sqrt{h}\xi_j$ ($j = 1, 2, \dots, N^2$) or $\beta_2(N + 1)\sqrt{h}\xi_j$ ($j = N^2 + 1, N^2 + 2, \dots, 2N^2$).

In this example, let us set $N = 127$, $\beta_1 = \beta_2 = 1/(N + 1)$, $\gamma = 1/10$, $u_0(x, y) = 4y(1 - y) \sin(\pi x)$, and $v_0(x, y) = \sin(\pi x) \sin^2(2\pi y)$, and simulate 1000 independent trajectories for a given h . Then, the dimension of the SDE is 32258. In order to solve the SDE numerically stably with reasonable cost when a step size h is given, we set the smallest stage number for each of the SROCK2 and split SROCK2 methods to solve it stably. As a result, we set $s = 4, 5$, and 7 for the SROCK2 methods when $h = 1/2048, 1/1024$, and $1/512$, respectively, whereas for the split SROCK2 method we set $s = 80$ when $h = 1/4$. We also investigate the SERKW2D3 and SSDFMT methods for $h = 1/4$. Taking computational time into account, we set $\rho = 100$ and 30 for Krylov subspace projection techniques in the SERKW2D3 and SSDFMT methods, respectively.

The results are indicated in Figs. 6 and 7. In Fig. 6, we show numerical approximations to the expectation of $u(t, x, y)$ at $t = 1/2$. The left-hand plot shows an approximate mean by the split SROCK2 method with $s = 80$ for $h = 1/4$, whereas the right-hand plot shows an approximate mean by the SROCK2 method with $s = 7$ for $h = 1/512$. The difference between them is clearly visible. Although we have omitted plots given by the SROCK2 method with $s = 5$ for $h = 1/1024$, its difference from the left-hand plot is still visible. If we

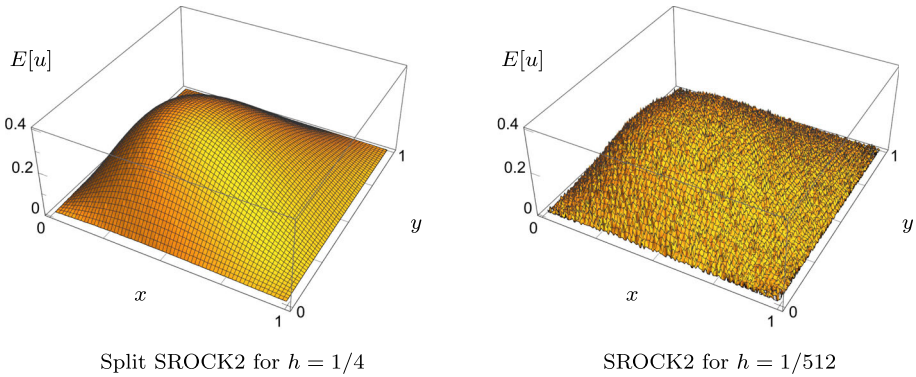


Fig. 6 Approximations to the expectation of u in (21) by the split SROCK2 method with $s = 80$ for $h = 1/4$ or the SROCK2 method with $s = 7$ for $h = 1/512$ over 1000 trajectories

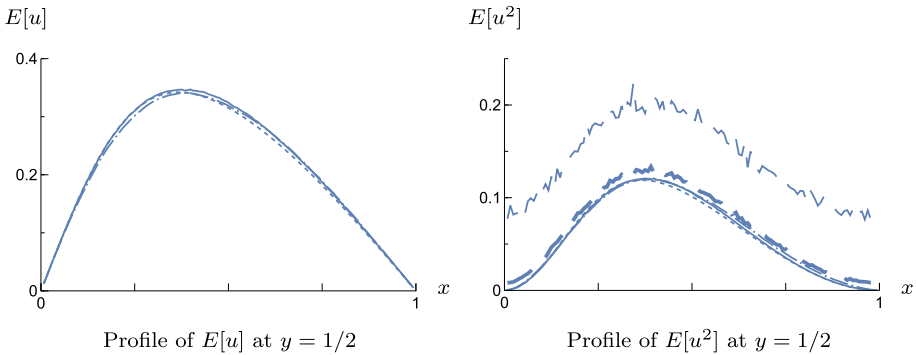


Fig. 7 Approximations to the expectation and second moment of u at $y = 1/2$ over 1000 trajectories in (21). Solid line: split SROCK2 ($s=80, h=1/4$); dashed line: SSDFMT ($h=1/4$); long-dash-dotted line: SERKW2D3 ($h=1/4$); long-dashed line: SROCK2 ($s=7, h=1/512$); thick long-dashed line: SROCK2 ($s=5, h=1/1024$)

choose a smaller step size (for example, $h = 1/2048$ ($s = 4$)) for the SROCK2 method, then such a difference completely becomes invisible. These facts imply that when we approximate an expectation over not very large number of trajectories by the SROCK2 method with not sufficiently small h , the approximate mean might be largely contaminated by errors. On the other hand, we can get other plots by the SERKW2D3 and SSDFMT methods for $h = 1/4$, which are almost the same as the left-hand plot, and the difference between them is invisible. Thus, in order to check differences among all the methods in detail, we give Fig. 7. In the left-hand plot, we can see a slight difference between the split SROCK2 method for $h = 1/4$ and each of the SERKW2D3 and SSDFMT methods for $h = 1/4$, but if we set $h = 1/8$ for the SERKW2D3 and SSDFMT methods, then the difference becomes invisible. Instead of omitting approximations to $E[u]$ given by the SROCK2 methods, we show approximations to $E[u^2]$ given by them in the right-hand plot. We can see that their approximations are heavily influenced by errors. Especially, a line corresponding to the approximation by the SROCK2 method with $s = 7$ and $h = 1/512$ looks serrated.

In the left-hand plot of Fig. 6, we would like to emphasize that the smoothness does not come from the fact that for the split SROCK2 method the noise term is evaluated fewer times. In fact, we can get a smooth plot by the split SROCK2 method with $s = 28$ and $h = 1/32$.

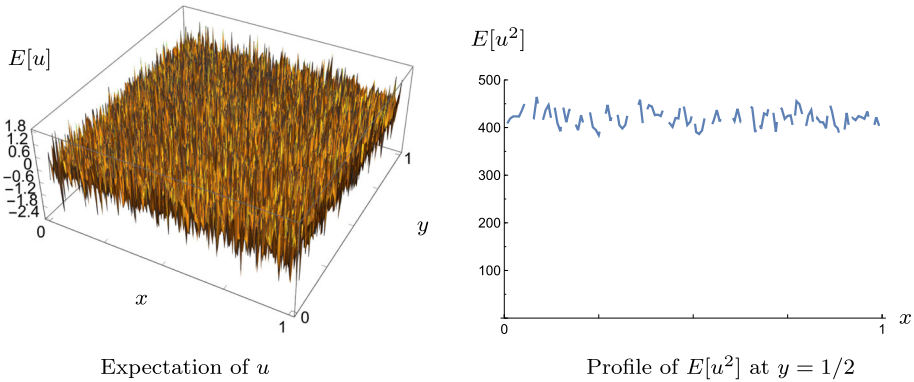


Fig. 8 Approximations to the expectation of u and the second moment of u at $y = 1/2$ in (21) by the SROCK2 method with $s = 30$ for $h = 1/32$ over 1000 trajectories

Table 4 Comparisons of computational cost in one step and one trajectory

Method	n_e	n_r
SROCK2 with s stages	$s + 5m + 1$	$2m$
Split SROCK2 with s stages	$2s + 5m$	$2m$

In contrast to this, the SROCK2 methods cannot solve the SDE numerically stably for a larger h than $1/32$. In addition, even for $h = 1/32$, the SROCK2 methods need a larger stage number $s = 30$ than 28 to avoid a numerical explosion. Figure 8 gives a result by the SROCK2 method with $s = 30$ and $h = 1/32$. The method can avoid a numerical explosion, but it is heavily influenced by computational errors

Remark 6 This is a typical example where the SROCK2 methods suffer from errors. Now, the eigenvalues in the linear part of the drift term are distributed in a wide interval. Thus, λh 's for them are also distributed in a wide interval when a large step size h is given. As a result, the SROCK2 methods are heavily influenced by the errors due to wildly distributed λh 's. This fact is suggested in Fig. 2, although the present SDE has additive noise, not multiplicative noise.

From all results in the present experiment, we can see that the split SROCK2 method for $h = 1/4$ has better precision than not only the SERKW2D3 and SSDFMT methods for $h = 1/4$, but also the SROCK2 method for $h = 1/1024$, and especially say that the SROCK2 methods suffer from precision when a sufficiently small step size such as $h = 1/2048$ is not given, even if they can solve the SDE numerically stably.

Incidentally, Table 4 indicates comparisons of computational cost for each of the SROCK2 and split SROCK2 methods in one step and one trajectory. In the table, n_e and n_r stand for the number of evaluations on d -dimensional functions in the drift or diffusion coefficients, and the number of generated pseudorandom numbers, respectively. Note that if the SDEs have diagonal noise, then n_e reduces to a smaller number obtained by the substitution of $m = 1$ due to Remark 1. In the above example, since the SROCK2 method has $s = 4$ and $h = 1/2048$ for the SDE with diagonal noise, the method evaluates the drift or diffusion coefficients 5120 times up to $t = 1/4$. On the other hand, the split SROCK2 method with $s = 80$ and $h = 1/4$ evaluates them merely 165 times up to $t = 1/4$. From a comparison between Tables 2 and 3, we can see that the efficiency of the stabilization procedures for the

Table 5 Average computational time (second) per 1000 trajectories simulation up to $t = 1/2$

SROCK2 with $s = 4, h = 1/2048$	Split SROCK2 with $s = 80, h = 1/4$
870.6 (SD 1.1)	11.7 (SD 0.6)
SERKW2D3 with $\rho = 100, h = 1/4$	SSDFMT with $\rho = 30, h = 1/4$
1058.9 (SD 1.3)	263.8 (SD 1.4)

SROCK2 methods is 2.87 times higher than that for the split SROCK2 methods. However, this numerical experiment shows that the split SROCK2 method with $s = 80$ and $h = 1/4$ has 0.03 times fewer number of the function evaluations than the SROCK2 method with $s = 4$ and $h = 1/2048$, whereas they have invisible errors in the plots of $E[u]$ and $E[u^2]$.

Although we are concerned with weak second order methods, we give a brief comment on the SK-ROCK methods, corresponding to Remark 4. When $\eta = 2.0$, the SK-ROCK method with $s = 11$ and $h = 1/128$ gives the same profiles as those by the split SROCK2 method with $s = 80$ and $h = 1/4$ in Fig. 7. It is known that the SK-ROCK methods work very well for SDEs with additive noise [5]. In spite of it, this numerical experiment shows that the split SROCK2 method with $s = 80$ and $h = 1/4$ has 0.43 times fewer number of the function evaluations than the SK-ROCK method with $s = 11$ and $h = 1/128$, whereas they have invisible errors in the plots of $E[u]$ and $E[u^2]$.

We have also investigated computational time. The numerical experiments were carried out on Microsoft Windows 10 Pro 64-bit Operating System with an Intel Xeon W-2133 CPU @3.60GHz and 64GB RAM. We used Intel C and Fortran compilers in Intel oneAPI 2022. We simulated 16 batches of 1000 trajectories for each method. The average and standard deviation (SD) of time are indicated in Table 5. The split SROCK2 method clearly shows the best performance. Krylov subspace projection techniques are used in the SERKW2D3 and SSDFMT methods. As the SSDFMT method uses a smaller dimension ρ of a Krylov subspace since it does not have φ_1 and φ_2 functions, it recorded much shorter computational time than the SERKW2D3 method.

Incidentally, as we mentioned in Remark 5, in [18] the authors have proposed another efficient algorithm based on Taylor series for the product of the matrix exponential and a vector, and they [18] and some researchers [19] have indicated that it could work much faster than the Krylov subspace projection techniques. For this, we have tested it, which is called the `expmvtay2` algorithm. As the algorithm is slightly more sensitive than the Krylov subspace projection techniques, we chose $h = 1/8$ and carried out simulations up to $t = 1/2$. On average, the SSDFMT method with the `expmvtay2` algorithm took 1491.0 s (SD 4.8), and the SSDFMT method with the Krylov subspace projection techniques took 409.9 s (SD 4.1) per 1000 trajectories simulation. Thus, we can see that the Krylov subspace projection techniques are more efficient than the algorithm in this high-dimensional problem.

6 Concluding Remarks

We have derived a class of split SROCK2 methods that have a free parameter α and which achieve weak order 2 for noncommutative Itô SDEs. To decide the value of α , we have provided a linear error analysis on one-step methods for SDEs. On the basis of the analysis, we have decided the optimal value of α for each stage s such that the errors on the expectation and the second moment of the amplification factor of the methods become small if the noise term of the SDE is not very large. As a result, we have shown that the split SROCK2 methods have much smaller errors than the SROCK2 methods in the linear error analysis.

In order to check the computational accuracy and performance of the methods, we carried out three numerical experiments. Although the selection for α was based on an error analysis in a linear case, the numerical experiments suggest that it is valid also in other cases. In the first experiment, the damped nonlinear Kubo oscillator was considered. The experiment confirmed that the values of α we decided are appropriate for the split SROCK2 methods. In fact, the split SROCK2 method with $s = 3$ showed smaller errors than the SROCK2 method with $s = 3$. In the second experiment, we dealt with a linear SDE which cannot be transformed to decoupled scalar SDEs. The experiment showed the superiority of the split SROCK2 methods over the SROCK2 methods in the approximation to not only the expectation, but also the second moment when the step size is large. In the third experiment, we considered a multidimensional stochastic Burgers equation with a multidimensional space-time white noise, which leads to a very high-dimensional SDE with a high-dimensional, diagonal, and additive noise after finite differences. This experiment showed again the superiority of our method over the SROCK2 methods in terms of computational accuracy for relatively large time-step size. In addition, the method was superior to the stochastic exponential RK methods in terms of computational time.

A significant advantage of the methods based on the ROCK2 methods is that they can numerically stably cope with high-dimensional stiff SDEs whose drift term has eigenvalues lie near the negative real axis, by just increasing the stage number. In some types of SDEs such as the third example, however, the SROCK2 methods require a very small step size for computational accuracy, not for numerical stability. The split SROCK2 methods do not have such a step size restriction.

Finally, we should make some remarks. We have considered the case of $-1/2 \leq r \leq 0$ in the error analysis. If r becomes close to -2 , then $\max_{I(r) < \lambda T_0 < 0} |\hat{\mathcal{E}}(0)|$ becomes close to 1 for the methods based on the ROCK2 methods, whereas $\hat{\mathcal{E}}(0) = 0$ for $\lambda T_0 = 0$ ($\sigma^2 = r\lambda$). This means that we have to choose h such that $|\lambda h|$ is sufficiently small for accuracy if we apply the methods in which the ROCK2 methods are embedded, to SDEs with large noise even if they are linear SDEs. This is a disadvantage of these methods, compared with other methods such as stochastic exponential RK methods. On the other hand, stochastic exponential RK methods also have a disadvantage, namely the computational cost for high-dimensional SDEs. As we have seen in Sect. 4.2, even when a step size h is large such as $h = 1/4$, we need small Δt 's to calculate the products of matrix exponential functions and vectors precisely, which substantially increase the computational cost.

Acknowledgements The authors would like to thank referees, Professor Chi-Wang Shu and Professor David Cohen for their comments which helped to improve the earlier versions of this paper.

Funding This work was partially supported by JSPS Grant-in-Aid for Scientific Research 17K05369.

Data Availability Data and program codes are available in https://github.com/yosh-komori/supplementary_info_files_2023.

Declararions

Conflict of interest The authors have not disclosed any competing interests.

References

1. Abdulle, A., Cirilli, S.: S-ROCK: Chebyshev methods for stiff stochastic differential equations. *SIAM J. Sci. Comput.* **30**(2), 997–1014 (2008)
2. Abdulle, A., Li, T.: S-ROCK methods for stiff Itô SDEs. *Commun. Math. Sci.* **6**(4), 845–868 (2008)
3. Abdulle, A., Medovikov, A.A.: Second order Chebyshev methods based on orthogonal polynomials. *Numer. Math.* **90**, 1–18 (2001)
4. Abdulle, A., Vilmart, G., Zygalkakis, K.C.: Weak second order explicit stabilized methods for stiff stochastic differential equations. *SIAM J. Sci. Comput.* **35**(4), A1792–A1814 (2013)
5. Abdulle, A., Almuslimani, I., Vilmart, G.: Optimal explicit stabilized integrator of weak order 1 for stiff and ergodic stochastic differential equations. *SIAM/ASA J. Uncertain. Quantif.* **6**(2), 937–964 (2018)
6. Al-Mohy, A.H., Higham, N.J.: A new scaling and squaring algorithm for the matrix exponential. *SIAM J. Matrix Anal.* **31**(3), 970–989 (2009)
7. Arnold, L.: *Stochastic Differential Equations: Theory and Applications*. John Wiley & Sons, New York (1974)
8. Brzeźniak, Z., Goldys, B., Neklyudov, M.: Multidimensional stochastic Burgers equation. *SIAM J. Math. Anal.* **46**(1), 871–889 (2014)
9. Cohen, D.: On the numerical discretisation of stochastic oscillators. *Math. Comput. Simul.* **82**(8), 1478–1495 (2012)
10. Cohen, D., Sigg, M.: Convergence analysis of trigonometric methods for stiff second-order stochastic differential equations. *Numer. Math.* **121**(1), 1–29 (2012)
11. Debrabant, K., Kværnø, A., Mattsson, N.C.: Runge-Kutta Lawson schemes for stochastic differential equations. *BIT Numer. Math.* **61**(2), 381–409 (2021)
12. Debrabant, K., Kværnø, A., Mattsson, N.C.: Lawson schemes for highly oscillatory stochastic differential equations and conservation of invariants. *BIT Numer. Math.* **62**(4), 1121–1147 (2022)
13. Dyöngy, I.: Lattice approximations for stochastic quasi-linear parabolic partial differential equations driven by space-time white noise I. *Potential Anal.* **9**, 1–25 (1998)
14. Erdoğan, U., Lord, G.J.: A new class of exponential integrators for SDEs with multiplicative noise. *IMA J. Appl. Math.* **39**(2), 820–846 (2019)
15. Hochbruck, M., Ostermann, A.: Explicit exponential Runge-Kutta methods for semilinear parabolic problems. *SIAM J. Numer. Anal.* **43**(3), 1069–1090 (2005)
16. Hochbruck, M., Ostermann, A.: Exponential integrators. *Acta Numer.* **19**, 209–286 (2010)
17. Hochbruck, M., Lubich, C., Selhofer, H.: Exponential integrators for large systems of differential equations. *SIAM J. Sci. Comput.* **19**(5), 1552–1574 (1998)
18. Ibáñez, J., Alonso, J.M., Alonso-Jordá, P., Defez, E., Sastre, J.: Two Taylor algorithms for computing the action of the matrix exponential on a vector. *Algorithms* **15**(2), 48 (2022)
19. Kamm, K., Pagliarani, S., Pascucci, A.: Numerical solution of kinetic SPDEs via stochastic Magnus expansion. *Math. Comput. Simul.* **207**, 189–208 (2023)
20. Kloeden, P.E., Platen, E.: *Numerical Solution of Stochastic Differential Equations*. Springer-Verlag, Berlin (1999). (**Corrected Third Printing**)
21. Komori, Y., Burrage, K.: Weak second order S-ROCK methods for Stratonovich stochastic differential equations. *J. Comput. Appl. Math.* **236**(11), 2895–2908 (2012)
22. Komori, Y., Burrage, K.: A stochastic exponential Euler scheme for simulation of stiff biochemical reaction systems. *BIT* **54**(4), 1067–1085 (2014)
23. Komori, Y., Cohen, D., Burrage, K.: Weak second order explicit exponential Runge-Kutta methods for stochastic differential equations. *SIAM J. Sci. Comput.* **39**(6), A2857–A2878 (2017)
24. Matsumoto, M., Nishimura, T.: Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.* **8**(1), 3–30 (1998)
25. Milstein, G.N.: Weak approximation of solutions of systems of stochastic differential equations. *Theory Probab. Appl.* **30**(4), 750–766 (1986)
26. Milstein, G.N., Tretyakov, M.V.: *Stochastic Numerics for Mathematical Physics*. Springer-Verlag, Berlin (2004)

27. Moler, C., Loan, C.V.: Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Rev.* **45**(1), 3–49 (2003)
28. Sidje, R.B.: Expokit: a software package for computing matrix exponentials. *ACM Trans. Math. Softw.* **24**(1), 130–156 (1998)
29. Tocino, A., Senosiain, M.J.: Mean-square stability analysis of numerical schemes for stochastic differential systems. *J. Comput. Appl. Math.* **236**(10), 2660–2672 (2012)
30. van der Houwen, P.J., Sommeijer, B.P.: On the internal stability of explicit, m -stage Runge-Kutta methods for large m -values. *ZAMM Z. Angew. Math. Mech.* **60**, 479–485 (1980)
31. Yang, G., Burrage, K., Burrage, P., Ding, X.: How to choose an appropriate initial condition to simulate stochastic differential equations stably. In: AIP Conference Proceedings (to appear)
32. Yang, G., Burrage, K., Komori, Y., Burrage, P.M., Ding, X.: A class of new Magnus-type methods for semi-linear non-commutative Itô stochastic differential equations. *Numer. Algorithms* **88**(4), 1641–1665 (2021)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.