



Approximating a New Class of Stochastic Differential Equations via Operational Matrices of Bernoulli Polynomials

Yadollah Ordokhani¹ · Nasrin Samadyar¹

Received: 12 April 2023 / Accepted: 22 August 2023 / Published online: 22 September 2023
© The Author(s), under exclusive licence to Shiraz University 2023

Abstract

In the current paper, we introduce an efficient methodology to solve nonlinear stochastic differential equations (SDEs) driven by variable order fractional Brownian motion (vofBm) with appropriate initial condition. SDEs have many applications throughout pure mathematics and are used to model various behaviors of stochastic models such as stock prices, random growth models or physical systems that are subjected to thermal fluctuations. The mechanism of our proposed approach, which is based on Bernoulli polynomials operational matrices, is that it first transforms the problem under consideration into a nonlinear stochastic integral equation (SIE) driven by vofBm by using given initial condition and integrating from both sides of nonlinear SDE over the interval $[0, t]$, where $t \in [0, \mathbf{T}]$. Then, operational matrices of integration (omi) based on Bernoulli polynomials are utilized to significantly reduce the complexity of solving obtained SIE through converting such SIE into a nonlinear system of algebraic equations. Error analysis and convergence of suggested technique are also analyzed under some mild conditions. It is concluded that by increasing \hat{n} , the approximate solution more accurately estimates the exact solution, where \hat{n} denotes the number of elements in the used Bernoulli vector. Finally, some test problems are included to emphasize that the introduced idea is accurate, efficient and applicable. One of the most important innovations of this paper is the numerical simulation of vofBm, which is done in two steps. In the first step, standard Brownian motion (sBm) is constructed via spline interpolation scheme and then block-pulse and hat functions are used in the second step to simulate vofBm. Moreover, the obtained numerical results are also compared with achieved results from Chebyshev cardinal wavelets (Ccw) method to confirm the superiority of the presented method respect to the previous methods. This paper finishes with presenting a real application of this model and solving it via our method.

Keywords Stochastic differential equations · Variable order fractional Brownian motion · Bernoulli polynomials · Operational matrix method · Error analysis

1 Introduction

The significant part of contents in numerical analysis field is designing and analyzing of efficient algorithms to meet demands that have been appeared in other sciences. Among the available methods, the operational matrix methods (omm) have dedicated itself a lot of researches. These

methods are based on expansion of all functions in the problem under investigation with various basic functions and using popular matrices which are named omi and operational matrix of differentiation (omd). Different functions and polynomials have been extensively utilized as basic functions by mathematicians to approximate the solution of underlying problem. For examples, Chebyshev polynomials have been employed by Heydari et al. to solve variable order fractional biharmonic equation and nonlinear Ginzburg–Landau equation in Heydari and Avazzadeh (2018) and Heydari et al. (2019), respectively. Abbasbandy et al. (2015) applied Legendre functions to estimate the solution of time fractional convection-diffusion equation. An applicable technique based on Bernoulli polynomials

✉ Yadollah Ordokhani
ordokhani@alzahra.ac.ir
Nasrin Samadyar
n.samadyar@alzahra.ac.ir

¹ Department of Mathematics, Faculty of Mathematical Sciences, Alzahra University, Tehran, Iran

together with its accuracy analysis has been introduced by Singh et al. (2018). Euler polynomials have been used as basic functions in Balci and Sezer (2016) to find the numerical solution of generalized linear Fredholm integro-differential difference equations. A numerical technique based on Bernstein polynomials has been improved by Chen et al. to estimate the solution of variable order linear cable equation in Chen et al. (2014). Finally, block-pulse functions have been proposed by Maleknejad et al. (2012), delta functions have been used by Roodaki and Almasieh (2012), triangular functions have been employed by Asgari and Khodabin (2017), hat functions have been suggested by Tripathi et al. (2013), to solve various and numerous problems of mathematics.

Fractional calculus has gained the attention of scholars as a mathematical modeling tool to describe occurred phenomena in many various disciplines. Fractional order models are more appropriate than integer order models to survey the behavior of processes with memory and hereditary features. The mathematical scientists have to focus on numerical schemes due to only a few number of such equations have analytical solution. The many efforts of researchers to estimate the solution of fractional problems have led to the development of various numerical techniques. The common methods are wavelet method (ur Rehman and Khan 2011), operational matrix methods (Mirzaee and Samadyar 2019; Rahimkhani et al. 2017), Galerkin method (Kamrani 2016), collocation method (Rahimkhani et al. 2019), finite difference method (Li et al. 2011), finite element method (Li et al. 2018), meshless method (Mirzaee and Samadyar 2019), spectral element method (Dehghan and Abbaszadeh 2018), the multistep Laplace optimized decomposition method (Maayah et al. 2022), etc.

Existence of various stochastic perturbation factors and production of powerful computing tools have led us that occurred phenomena in real life are modeled via different types of stochastic problems to reveal more accurate details in behavior description of such phenomena. In addition to no having the exact solution of such equations in many situations, it is also difficult obtaining their numerical solution. Thereby among introduced schemes in published papers, one method can play a more important role among other numerical methods if it produces more accurate results and can be extended to solve other problems. Finite difference method which has been utilized to solve linear stochastic integro-differential equations in Dareiotis and Leahy (2016) can be caused many difficulties in real life problems. For instance, discretization of domain and generating meshes is a time consuming and costly activity. Furthermore, some finite difference ideas are unconditionally stable. In the last decade, omm have been extremely used to obtain sufficiently high accuracy and alleviate

accumulation of truncated error, complexity, computational operations and CPU times. For example, it has been greatly utilized to solve stochastic Volterra–Fredholm integral equations in Khodabin et al. (2012). Stochastic Volterra integral equations has been numerically solved by this method based on block-pulse and triangular functions in Maleknejad et al. (2012) and Khodabin et al. (2013), respectively. Samadyar et al. introduced orthonormal Bernoulli polynomials and applied them to approximate solution of stochastic Itô–Volterra integral equations of Abel type in Samadyar and Mirzaee (2020). In Heydari et al. (2016), omm based on second kind Chebyshev wavelets has been suggested by Heydari et al. to achieve an accurate numerical solution of stochastic heat equation. Shifted Legendre Spectral Collocation Algorithm has been applied to investigate the existence and uniqueness of the solution of fractional stochastic integro differential equations and obtain its numerical simulation in Badawi et al. (2022). The approximate solution of fractional stochastic integro differential equations using Legendre-shifted spectral approach and Legendre Gauss spectral collocation method has been presented by Badawi et al. in the papers Badawi et al. (2023a, b), respectively.

Sheng et al. (2011) introduced vofBm in the Riemann–Liouville sense as follows:

$$B^{H(t)}(t) = \frac{1}{\Gamma(H(t) + \frac{1}{2})} \int_0^t (t-s)^{H(t)-\frac{1}{2}} dB(s), \quad (1)$$

where $H(t) \in (0, 1)$. Notice that specific situations of this process are classical fBm and sBm that obtain by considering $H(t) = H$ and $H(t) = \frac{1}{2}$, respectively. Although solving stochastic problems driven by sBm and classical fBm is difficult, there have been more works on the numerical solution of such equations rather than the numerical solution of stochastic problems driven by vofBm. Providing a flexible method together with analyzing its convergence to solve stochastic evolution equations driven by fBm has been done in Kamrani and Jamshidi (2017). Nonlinear stochastic Itô–Volterra integral equations driven by fBm have been solved by omm based on hat functions and modification of hat functions in Hashemi et al. (2017) and Mirzaee and Samadyar (2018), respectively. It is essential that mentioned that classical fBm is appropriate only for modeling mono fractal phenomena which have similar global irregularity and fixed memory. On the other hand, global self-similarity is rarely appeared and fixed scaling is only satisfied for a series of certain finite intervals. Moreover, experimental data show that scaling exponent and order of similarity have multivalued and there exist phenomena in real life which have multifractal properties with variable space and time dependent memory. Thus, vofBm is a suitable way to overcome these



limitations and it has been recently used for modeling events with variable irregularities or variable memory properties.

Suppose that $B^{H(t)}(t), t \geq 0$, be a vofBm process which is defined in Eq. (1). An differential equation of the form Heydari et al. (2019)

$$\begin{cases} dU(t) = \mu(t, U(t))dt + \sigma(t, U(t))dB^{H(t)}(t), t \in [0, 1], \\ U(0) = U_0, \end{cases} \quad (2)$$

where the functions $\mu(u, t)$ and $\sigma(u, t)$ are known smooth functions and $U(t)$ is an unknown stochastic process defined on a certain probability space $(\Omega, \mathcal{F}, \mathcal{P})$, is named nonlinear SDE driven by vofBm. The process $B(t)$ denotes a sBm defined on same probability space and U_0 is a given deterministic initial value. The functions $\mu(u, t)$ and $\sigma(u, t)$ are called the coefficients of this equation.

Equation of the form (2) is seen in modeling various problems such as signal processing (Sheng et al. 2012), geophysics (Echelard et al. 2010), financial time series (Corlay et al. 2014), but unfortunately its exact solution in many situations is not available. In the present time, it is very difficult to solve the nonlinear SDEs driven by vofBm either analytically or numerically. So, there are not many published literatures on this subject. In this paper, we introduced an efficient idea to find the numerical solution of nonlinear SDE expressed in Eq. (2). The structure of this method is such that it first transform the mentioned SDE into a corresponding SIE driven by vofBm and expand all functions in the obtained SIE with Bernoulli polynomials. In the sequel, stochastic omi driven by vofBm based on Bernoulli polynomials is derived, and then it together with ordinary omi are used to convert solving the mentioned problem into solving a set of nonlinear equations. That way, we try to find the numerical solution of this equation with higher accuracy and lower computational performance. The most important advantages of the presented method are as follows:

- Using this method, equation under consideration is converted to a system of algebraic equations which can be easily solved. Therefore, the complexity of this equation, which is caused by fractional and stochastic terms, becomes very simple.
- The unknown coefficients of the approximation of the function with these basis are easily calculated without any integration. Therefore, the computational cost of the proposed numerical method is low.
- Because of the simplicity of Bernoulli polynomials, this method is a powerful mathematical tool to solve various kinds of equations with little additional works. In other words, Bernoulli polynomials are the simple basis

functions, so the proposed method is easy to implement and it is a powerful mathematical tool to obtain the numerical solution of various kind of problems with little additional works.

- The proposed scheme is convergent. Also, when the exact solution of the problem is a polynomial of degree n , we obtain the exact solution.

The outline of the rest of this paper is organized as follows. Simulation of vofBm by using block-pulse and hat functions has been done in Sect. 2. The definition of Bernoulli polynomials and some of their properties are presented in Sect. 3. In Sect. 4, a numerical method for solving the nonlinear SDEs driven by vofBm is proposed. The error analysis has been investigated in Sect. 5. The numerical results and application of under consideration problem in real world are carried out in Sect. 6 and Sect. 7, respectively. Finally, the conclusion is included in the Sect. 8.

2 Simulation of vofBm

2.1 Variable Order Fractional Integral Operator

Definition 1 Suppose that $\alpha(t) \geq 0$ is a known continuous function. The Riemann–Liouville fractional integral of function $f(t)$ of variable order $\alpha(t)$ is defined as follows (Chen et al. 2014):

$$(\mathcal{I}^{\alpha(t)}f)(t) = \begin{cases} \frac{1}{\Gamma(\alpha(t))} \int_0^t (t-s)^{\alpha(t)-1} f(s) ds, & \alpha(t) > 0, \\ f(t), & \alpha(t) = 0. \end{cases} \quad (3)$$

2.2 The Block-Pulse Functions

Definition 2 (Maleknejad et al. 2012) Consider a vector of block-pulse functions with \hat{m} components over the interval $[0, \mathbf{T})$ as follows:

$$\vec{\Phi}_{\hat{m}}(t) = [\phi_1(t), \phi_2(t), \dots, \phi_{\hat{m}}(t)]^T, \quad (4)$$

where the i th components of this vector is defined by

$$\phi_i(t) = \begin{cases} 1, & (i-1)\frac{\mathbf{T}}{\hat{m}} \leq t < i\frac{\mathbf{T}}{\hat{m}}, i = 1, 2, \dots, \hat{m}. \\ 0, & \text{otherwise,} \end{cases} \quad (5)$$

Any absolutely integrable function $f(t)$, defined over the interval $[0, \mathbf{T}]$, can be expanded by \hat{m} terms of block-pulse functions as follows:

$$f(t) \simeq \overrightarrow{f_{\hat{m}}}(t) = \sum_{i=1}^{\hat{m}} f_i \phi_i(t) = \overrightarrow{F_{\hat{m}}}^T \overrightarrow{\Phi_{\hat{m}}}(t), \tag{6}$$

where $\overrightarrow{\Phi_{\hat{m}}}(t)$ is defined in Eq. (4) and $\overrightarrow{F_{\hat{m}}} = [f_1, f_2, \dots, f_{\hat{m}}]^T$. Furthermore, i th component of vector $\overrightarrow{F_{\hat{m}}}$ is calculated by the following relation

$$f_i = \frac{\hat{m}}{\mathbf{T}} \int_{(i-1)\frac{\mathbf{T}}{\hat{m}}}^{i\frac{\mathbf{T}}{\hat{m}}} f(t) \phi_i(t) dt = \frac{\hat{m}}{\mathbf{T}} \int_{(i-1)\frac{\mathbf{T}}{\hat{m}}}^{i\frac{\mathbf{T}}{\hat{m}}} f(t) dt \tag{7}$$

$$\simeq f\left(\frac{(2i-1)\mathbf{T}}{2\hat{m}}\right), \quad i = 1, 2, \dots, \hat{m}.$$

Remark 1 (Kilicman and Al Zhouh 2007) Integration and differentiation of block-pulse functions vector $\overrightarrow{\Phi_{\hat{m}}}(t)$ for l -times is approximated as follows:

$$\underbrace{\int_0^t \dots \int_0^t}_{l\text{-times}} \overrightarrow{\Phi_{\hat{m}}}(s) ds \simeq \mathbf{I}_{\hat{m}}^{(l)} \overrightarrow{\Phi_{\hat{m}}}(t), \quad \frac{d^l \overrightarrow{\Phi_{\hat{m}}}(t)}{dt^l} \simeq \mathbf{D}_{\hat{m}}^{(l)} \overrightarrow{\Phi_{\hat{m}}}(t), \tag{8}$$

where $\mathbf{I}_{\hat{m}}^{(l)}$ and $\mathbf{D}_{\hat{m}}^{(l)}$ are called block-pulse omi and omd of l -times, respectively, and are given by

$$\mathbf{I}_{\hat{m}}^{(l)} = \left(\frac{\mathbf{T}}{\hat{m}}\right)^l \frac{1}{(l+1)!} \begin{pmatrix} 1 & \zeta_1 & \zeta_2 & \dots & \zeta_{\hat{m}-1} \\ 0 & 1 & \zeta_1 & \dots & \zeta_{\hat{m}-2} \\ 0 & 0 & 1 & \dots & \zeta_{\hat{m}-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix}_{\hat{m} \times \hat{m}}, \tag{9}$$

and

$$\mathbf{D}_{\hat{m}}^{(l)} = (l+1)! \left(\frac{\hat{m}}{\mathbf{T}}\right)^l \begin{pmatrix} 1 & \zeta_1 & \zeta_2 & \dots & \zeta_{\hat{m}-1} \\ 0 & 1 & \zeta_1 & \dots & \zeta_{\hat{m}-2} \\ 0 & 0 & 1 & \dots & \zeta_{\hat{m}-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix}_{\hat{m} \times \hat{m}}, \tag{10}$$

where $\zeta_j = (j+1)^{l+1} - 2j^{l+1} + (j-1)^{l+1}$ and $\zeta_j = -\sum_{i=1}^j \zeta_i \zeta_{j-i}$ and $\zeta_0 = 1$.

2.3 The Hat Functions

Definition 3 (Hashemi et al. 2017) In a hat functions vector $\overrightarrow{\Psi_{\hat{m}}}(t) = [\psi_0(t), \psi_1(t), \dots, \psi_{\hat{m}-1}(t)]^T$ over the

interval $[0, \mathbf{T}]$ with \hat{m} components, the first component is defined as follows:

$$\psi_0(t) = \begin{cases} \frac{h-t}{h}, & 0 \leq t < h, \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

The i th component is defined as follows:

$$\psi_i(t) = \begin{cases} \frac{t-(i-1)h}{h}, & (i-1)h \leq t < ih, \\ \frac{(i+1)h-t}{h}, & ih \leq t < (i+1)h, \\ 0, & \text{otherwise,} \end{cases} \quad i = 1, 2, \dots, \hat{m}-2. \tag{12}$$

Finally, the last component is defined as follows:

$$\psi_{\hat{m}-1}(t) = \begin{cases} \frac{t-(\mathbf{T}-h)}{h}, & \mathbf{T}-h \leq t < \mathbf{T}, \\ 0, & \text{otherwise,} \end{cases} \tag{13}$$

where $h = \frac{\mathbf{T}}{\hat{m}-1}$.

Every continuous function $g(t)$ can be approximated by using \hat{m} terms of hat functions as follows:

$$g(t) \simeq \overrightarrow{g_{\hat{m}}}(t) = \sum_{i=0}^{\hat{m}-1} g_i \psi_i(t) = \overrightarrow{G_{\hat{m}}}^T \overrightarrow{\Psi_{\hat{m}}}(t), \tag{14}$$

where $\overrightarrow{G_{\hat{m}}} = [g_0, g_1, \dots, g_{\hat{m}-1}]^T$, and $g_i = g(ih)$ for $i = 0, 1, \dots, \hat{m}-1$.

Theorem 1 (Heydari et al. 2019) Consider positive continuous function $\alpha(t) : [0, \mathbf{T}] \rightarrow \mathbb{R}^+$ and hat functions vector $\overrightarrow{\Psi_{\hat{m}}}(t)$. The Riemann–Liouville fractional integration $\overrightarrow{\mathcal{I}^{\alpha(t)}}(\overrightarrow{\Psi_{\hat{m}}}(t))$ of variable order $\alpha(t)$ is represented by $(\mathcal{I}^{\alpha(t)} \overrightarrow{\Psi_{\hat{m}}})(t)$, and is estimated as follows:

$$(\mathcal{I}^{\alpha(t)} \overrightarrow{\Psi_{\hat{m}}})(t) \simeq \mathbf{L}_{\hat{m}}^{\alpha(t)} \overrightarrow{\Psi_{\hat{m}}}(t), \tag{15}$$

where $\mathbf{L}_{\hat{m}}^{\alpha(t)}$ is called fractional omi of variable order $\alpha(t)$ for hat functions and is computed as follows:

$$\mathbf{L}_{\hat{m}}^{\alpha(t)} = \begin{pmatrix} 0 & \rho_1 & \rho_2 & \dots & \rho_{\hat{m}-2} & \rho_{\hat{m}-1} \\ 0 & \varrho_{1,1} & \varrho_{1,2} & \dots & \varrho_{1,\hat{m}-2} & \varrho_{1,\hat{m}-1} \\ 0 & 0 & \varrho_{2,2} & \dots & \varrho_{2,\hat{m}-2} & \varrho_{2,\hat{m}-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \varrho_{\hat{m}-2,\hat{m}-2} & \varrho_{\hat{m}-2,\hat{m}-1} \\ 0 & 0 & 0 & \dots & 0 & \varrho_{\hat{m}-1,\hat{m}-1} \end{pmatrix}_{\hat{m} \times \hat{m}}, \tag{16}$$

where

$$\rho_j = \frac{h^{\alpha(jh)}}{\Gamma(\alpha(jh) + 2)} \left((j - 1)^{\alpha(jh)+1} + j^{\alpha(jh)} (\alpha(jh) - j + 1) \right),$$

$$j = 1, 2, \dots, \hat{m} - 1,$$

and for $i, j = 1, \dots, \hat{m} - 1$,

$$Q_{i,j} = \begin{cases} 0, & j < i, \\ \frac{h^{\alpha(jh)}}{\Gamma(\alpha(jh) + 2)}, & j = i, \\ \frac{h^{\alpha(jh)}}{\Gamma(\alpha(jh) + 2)} \left((j - i + 1)^{\alpha(jh)+1} - 2(j - i)^{\alpha(jh)+1} + (j - i - 1)^{\alpha(jh)+1} \right), & j > i, \end{cases}$$

Theorem 2 (Heydari et al. 2019) Assume that $\vec{\Phi}_{\hat{m}}(t)$ and $\vec{\Psi}_{\hat{m}}(t)$ denote the block-pulse and hat functions vectors, respectively. The vector $\vec{\Phi}_{\hat{m}}(t)$ can be estimated as follows:

$$\vec{\Phi}_{\hat{m}}(t) \simeq \mathbf{R}_{\hat{m}} \vec{\Psi}_{\hat{m}}(t), \tag{17}$$

where $\mathbf{R}_{\hat{m}} = (\mathbf{r}_{ij})$ is a matrix of order $\hat{m} \times \hat{m}$ which its elements are computed from the following formula

$$\mathbf{r}_{ij} = \phi_i((j - 1)h), \quad i, j = 1, 2, \dots, \hat{m}. \tag{18}$$

2.4 The vofBm Process Simulation

In this section, block-pulse and hat functions which are introduced in Subsects. 2.2 and 2.3 are employed to simulate the vofBm process. The strategy of constructing this stochastic process is divided into two parts. The sBm process is constructed in the first step by using the properties of this process and spline interpolation method. It is essential to mention that in this step other interpolation methods such as “linear”, “nearest”, “cubic” and “pchip” can be used instead of spline interpolation method. In the second step, first the simulated sBm process is estimated by block-pulse functions, and then, the relationship between block-pulse and hat functions is applied to obtain the vofBm process.

Step 1. Simulation of sBm process: Let’s start by hinting the properties of sBm process. This process is denoted by $B(t), t \in [0, \mathbf{T}]$ and has the following properties (Mirzaee and Samad-yar 2018):

- $B(0) = 0$.
- The increment $B(t) - B(s)$ where $0 \leq s < t \leq \mathbf{T}$ has normal distribution with mean 0 and variance $t - s$, i.e., $B(t) -$

$B(s) \sim \sqrt{t - s} \mathcal{N}(0, 1)$ such that $\mathcal{N}(0, 1)$ denotes normal distribution with mean 0 and variance 1.

- The increments $B(t) - B(s)$ and $B(v) - B(u)$, where $0 \leq u < v < s < t \leq \mathbf{T}$ are independent.

To construct sBm, first we choose a large positive integer number $N \in \mathbb{Z}^+$ and let $\delta t = \frac{\mathbf{T}}{N}$. Then, we consider distinguished nodal points $t_j = j\delta t$ for $j = 0, 1, \dots, N$. The first condition of sBm tell us that $B(0) = 0$ with probability 1, and the second and third conditions ensure us that $B(t_j) = B(t_{j-1}) + \delta B(t_j)$ where $j = 1, 2, \dots, N$, and $\delta B(t_j) \sim \sqrt{\delta t} \mathcal{N}(0, 1)$ is independent random variable. This procedure create a discretized sBm and then spline interpolation scheme is used to achieve a continuous function for sBm. The Matlab code for simulating sBm process over the interval $[0, 1]$ with $N = 100$ has been presented in Algorithm 1. The command “randn” has been used to create a random number with normal distribution and mean 0 and variance 1.

Step 2. Simulation of vofBm. As we mentioned in Sect. 2.2, every absolutely integrable function can be expanded in terms of block-pulse functions. So, the extension of sBm can be written as follows:

$$B(t) \simeq \vec{B}_{\hat{m}}^T \vec{\Phi}_{\hat{m}}(t), \tag{19}$$

where $\vec{B}_{\hat{m}} = [b_1, b_2, \dots, b_{\hat{m}}]^T$, $b_i \simeq B\left(\frac{(2i-1)\mathbf{T}}{2\hat{m}}\right)$ for $i = 1, 2, \dots, \hat{m}$, and the block-pulse vector $\vec{\Phi}_{\hat{m}}(t)$ has been introduced in Eq. (4). By substituting Eq. (19) into Eq. (1), we obtain

$$B^{H(t)}(t) \simeq \frac{1}{\Gamma(H(t) + \frac{1}{2})} \int_0^t (t - s)^{H(t) - \frac{1}{2}} d(\vec{B}_{\hat{m}}^T \vec{\Phi}_{\hat{m}}(s)). \tag{20}$$

From Eqs. (8) and (17), we conclude

$$B^{H(t)}(t) \simeq \frac{\vec{B}_{\hat{m}}^T \mathbf{D}_{\hat{m}}^{(1)} \mathbf{R}_{\hat{m}}}{\Gamma(H(t) + \frac{1}{2})} \int_0^t (t - s)^{H(t) - \frac{1}{2}} \vec{\Psi}_{\hat{m}}(s) ds$$

$$= \vec{B}_{\hat{m}}^T \mathbf{D}_{\hat{m}}^{(1)} \mathbf{R}_{\hat{m}} \left(\mathcal{I}^{H(t) + \frac{1}{2}} \vec{\Psi}_{\hat{m}} \right) (t). \tag{21}$$

Finally, Eq. (15) yields

$$B^{H(t)}(t) \simeq \vec{B}_{\hat{m}}^T \mathbf{D}_{\hat{m}}^{(1)} \mathbf{R}_{\hat{m}} \mathbf{L}_{\hat{m}}^{H(t) + \frac{1}{2}} \vec{\Psi}_{\hat{m}}(t). \tag{22}$$

The vofBm process simulation steps are summarized in Algorithm 2.

Algorithm 1.	Algorithm 2.
$N = 100;$	Input: $\hat{m}, H(t).$
$\mathbf{T} = 1;$	Output: $B^{H(t)}(t).$
$\delta t = \frac{\mathbf{T}}{N};$	Step 1: Compute the value of $h = \frac{\mathbf{T}}{m-1}.$
for $i = 0 : N$	Step 2: Compute the vector $\overrightarrow{B}_m = [b_1, b_2, \dots, b_m]^T.$
$t(i + 1, 1) = i \delta t;$	Step 3: Compute matrix $\mathbf{D}_m^{(l)}$ for $l = 1$ from Eq. (10).
end	Step 4: Compute matrix \mathbf{R}_m from Theorem 2.
$B = \text{zeros}(N + 1, 1);$	Step 4: Compute matrix $\mathbf{L}_m^{\alpha(t)}$ for $\alpha(t) = H(t) + \frac{1}{2}.$
for $i = 2 : N + 1$	Step 5: Compute $B^{H(t)}(t)$ from Eq. (22).
$B(i, 1) = B(i - 1, 1) + \sqrt{\delta t} \text{ randn};$	
end	
$\text{sBm} = \text{plot}(t, B);$	

3 Bernoulli Polynomials

Definition 4 (Bazm 2015) The Bernoulli polynomials satisfy in the following formula

$$\sum_{i=0}^j \binom{j+1}{i} \mathfrak{B}_i(t) = (j+1)t^j, \quad j = 0, 1, 2, \dots \tag{23}$$

Equation (23) can be written in the following matrix form

$$\mathbf{G}_n \overrightarrow{Y}_n(t) = \overrightarrow{X}_n(t), \tag{24}$$

where $\overrightarrow{X}_n(t) = [1, t, t^2, \dots, t^{n-1}]^T$, $\overrightarrow{Y}_n(t) = [\mathfrak{B}_0(t), \mathfrak{B}_1(t), \dots, \mathfrak{B}_{n-1}(t)]^T$ denotes Bernoulli basic vector, and $\mathbf{G}_n = (\mathbf{g}_{ij})$ is a lower triangular matrix of order $\hat{n} \times \hat{n}$ and

$$\mathbf{g}_{ij} = \begin{cases} \frac{1}{i} \binom{i}{j-1}, & i \geq j, \\ 0, & i < j, \end{cases} \quad i, j = 1, 2, \dots, \hat{n}. \tag{25}$$

Since all diagonal elements of matrix \mathbf{G}_n are nonzero, then

the matrix \mathbf{G}_n is nonsingular and Bernoulli basic vector can be directly calculated from

$$\overrightarrow{Y}_n(t) = \mathbf{G}_n^{-1} \overrightarrow{X}_n(t). \tag{26}$$

Every integrable function $u(t)$ can be expanded by using \hat{n} terms of Bernoulli polynomials as follows:

$$u(t) \simeq u_n(t) = \sum_{i=0}^{\hat{n}-1} u_i \mathfrak{B}_i(t) = \overrightarrow{U}_n^T \overrightarrow{Y}_n(t), \tag{27}$$

where $\overrightarrow{U}_n = [u_0, u_1, \dots, u_{\hat{n}-1}]^T$, and the component u_i is computed from the following formula

$$u_i = \frac{1}{i!} \int_0^1 \frac{d^i u(t)}{dt^i} dt, \quad i = 0, 1, \dots, \hat{n} - 1. \tag{28}$$

Theorem 3 (Bazm 2015) The integral of Bernoulli vector $\overrightarrow{Y}_n(s)$ respect to variable s over the interval $[0, t]$ can be estimated as follows:

$$\int_0^t \overrightarrow{Y}_n(s) ds \simeq \overline{\mathbf{P}}_n \overrightarrow{Y}_n(t), \tag{29}$$

where the matrix $\overline{\mathbf{P}}_n$ is named omi based on Bernoulli polynomials and is computed as follows:

$$\overline{\mathbf{P}}_n = \begin{pmatrix} -\mathfrak{B}_1(0) & 1 & 0 & \dots & 0 & 0 \\ -\frac{\mathfrak{B}_2(0)}{2} & 0 & \frac{1}{2} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ -\frac{\mathfrak{B}_{\hat{n}-1}(0)}{\hat{n}-1} & 0 & 0 & \dots & 0 & \frac{1}{\hat{n}-1} \\ -\frac{\mathfrak{B}_{\hat{n}}(0)}{\hat{n}} & 0 & 0 & \dots & 0 & 0 \end{pmatrix}_{\hat{n} \times \hat{n}}. \tag{30}$$

In the following, a theorem is stated for the first time in relation to stochastic omi driven by vofBm and then is proved by authors.

Theorem 4 The stochastic integral of Bernoulli polynomials vector $\overrightarrow{Y}_n(s)$ respect to vofBm $B^{H(s)}(s)$ over the interval $[0, t]$ can be approximated as follows:

$$\int_0^t \overrightarrow{Y}_n(s) dB^{H(s)}(s) \simeq \overline{\mathbf{P}}_n^s \overrightarrow{Y}_n(t), \tag{31}$$

where $\overline{\mathbf{P}}_n^s$ is a matrix of order $\hat{n} \times \hat{n}$ and is called stochastic omi driven by vofBm. Also, we have $\overline{\mathbf{P}}_n^s = \mathbf{G}_n^{-1} \mathbf{A}_n \mathbf{G}_n$ where $\mathbf{A}_n = (\mathbf{a}_{ij})$ is a diagonal matrix with the following diagonal components

$$\mathbf{a}_{ij} = \left(1 - \frac{i-1}{4}\right) B^{H(0.5)}(0.5) - \left(\frac{i-1}{2^{i-1}}\right) B^{H(0.25)}(0.25), i = j = 1, 2, \dots, \hat{n}. \tag{32}$$

Proof From Eq. (26), we have

$$\int_0^t \overrightarrow{\Upsilon}_{\hat{n}}(s) dB^{H(s)}(s) = \mathbf{G}_{\hat{n}}^{-1} \int_0^t \overrightarrow{\mathbf{X}}_{\hat{n}}(s) dB^{H(s)}(s). \tag{33}$$

Using part by part integration idea, we conclude

$$\begin{aligned} \int_0^t \overrightarrow{\mathbf{X}}_{\hat{n}}(s) dB^{H(s)}(s) &= \begin{pmatrix} \int_0^t dB^{H(s)}(s) \\ \int_0^t s dB^{H(s)}(s) \\ \vdots \\ \int_0^t s^{\hat{n}-1} dB^{H(s)}(s) \end{pmatrix} \\ &= \begin{pmatrix} B^{H(t)}(t) \\ tB^{H(t)}(t) - \int_0^t B^{H(s)}(s) ds \\ \vdots \\ t^{\hat{n}-1} B^{H(t)}(t) - \int_0^t (\hat{n}-1) s^{\hat{n}-2} B^{H(s)}(s) ds \end{pmatrix} \\ &= B^{H(t)}(t) \begin{pmatrix} 1 \\ t \\ \vdots \\ t^{\hat{n}-1} \end{pmatrix} - \begin{pmatrix} 0 \\ \int_0^t B^{H(s)}(s) ds \\ \vdots \\ \int_0^t (\hat{n}-1) s^{\hat{n}-2} B^{H(s)}(s) ds \end{pmatrix}. \end{aligned}$$

Let $\overrightarrow{\Xi}_{\hat{n}}(t) = \int_0^t \overrightarrow{\mathbf{X}}_{\hat{n}}(s) dB^{H(s)}(s)$, where $\overrightarrow{\Xi}_{\hat{n}}(t) = [\varpi_0(t), \varpi_1(t), \dots, \varpi_{\hat{n}-1}(t)]^T$ and

$$\varpi_l(t) = t^l B^{H(t)}(t) - \int_0^t l s^{l-1} B^{H(s)}(s) ds, l = 0, 1, \dots, \hat{n} - 1. \tag{34}$$

The existing integral in Eq. (34) is calculated via composite trapezoidal numerical integration rule. Thus, we obtain

$$\begin{aligned} \varpi_l(t) &\simeq t^l B^{H(t)}(t) - \frac{tl}{4} \left(2\left(\frac{t}{2}\right)^{l-1} B^{H(\frac{t}{2})}\left(\frac{t}{2}\right) + t^{l-1} B^{H(t)}(t)\right) \\ &= \left(1 - \frac{l}{4}\right) t^l B^{H(t)}(t) - \left(\frac{l}{2^l}\right) t^l B^{H(\frac{t}{2})}\left(\frac{t}{2}\right) \\ &= \left(\left(1 - \frac{l}{4}\right) B^{H(t)}(t) - \left(\frac{l}{2^l}\right) B^{H(\frac{t}{2})}\left(\frac{t}{2}\right)\right) t^l, l = 0, 1, \dots, \hat{n} - 1. \end{aligned} \tag{35}$$

The values of $B^{H(t)}(t)$ and $B^{H(\frac{t}{2})}\left(\frac{t}{2}\right)$ in Eq. (35) for $0 \leq t \leq 1$ can be approximated by $\varpi := B^{H(0.5)}(0.5)$ and $\varrho := B^{H(0.25)}(0.25)$, respectively. So, we can write

$$\begin{aligned} \overrightarrow{\Xi}_{\hat{n}}(t) &\simeq \begin{pmatrix} \varpi & 0 & \dots & 0 \\ 0 & \frac{3}{4}\varpi - \frac{1}{2}\varrho & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \underbrace{\left(1 - \frac{\hat{n}-1}{4}\right)\varpi - \left(\frac{\hat{n}-1}{2^{\hat{n}-1}}\right)\varrho}_{\mathbf{A}_{\hat{n}}}\varrho \end{pmatrix} \begin{pmatrix} 1 \\ t \\ \vdots \\ t^{\hat{n}-1} \end{pmatrix} \\ &= \mathbf{A}_{\hat{n}} \overrightarrow{\mathbf{X}}_{\hat{n}}(t). \end{aligned} \tag{36}$$

Using Eqs. (24), (33) and (36), we have

$$\int_0^t \overrightarrow{\Upsilon}_{\hat{n}}(s) dB^{H(s)}(s) = \mathbf{G}_{\hat{n}}^{-1} \mathbf{A}_{\hat{n}} \overrightarrow{\mathbf{X}}_{\hat{n}}(t) = \mathbf{G}_{\hat{n}}^{-1} \mathbf{A}_{\hat{n}} \mathbf{G}_{\hat{n}} \overrightarrow{\Upsilon}_{\hat{n}}(t) = \overrightarrow{\mathbf{P}}_{\hat{n}} \overrightarrow{\Upsilon}_{\hat{n}}(t), \tag{37}$$

where $\overrightarrow{\mathbf{P}}_{\hat{n}} = \mathbf{G}_{\hat{n}}^{-1} \mathbf{A}_{\hat{n}} \mathbf{G}_{\hat{n}}$. □

4 Numerical Scheme

The nonlinear SDE (2) can be written in the following SIE form

$$U(t) = U_0 + \int_0^t \mu(s, U(s)) ds + \int_0^t \sigma(s, U(s)) dB^{H(s)}(s), t \in [0, 1]. \tag{38}$$

Let

$$\omega(t) = \mu(t, U(t)), \theta(t) = \sigma(t, U(t)). \tag{39}$$

Thus, we should solve the following SIE driven by vofBm

$$U(t) = U_0 + \int_0^t \omega(s) ds + \int_0^t \theta(s) dB^{H(s)}(s), t \in [0, 1]. \tag{40}$$

From Eqs. (39) and (40), we have

$$\begin{cases} \omega(t) = \mu\left(t, U_0 + \int_0^t \omega(s) ds + \int_0^t \theta(s) dB^{H(s)}(s)\right), \\ \theta(t) = \sigma\left(t, U_0 + \int_0^t \omega(s) ds + \int_0^t \theta(s) dB^{H(s)}(s)\right). \end{cases} \tag{41}$$

The unknown functions $\omega(t)$ and $\theta(t)$ can be expanded in terms of Bernoulli polynomials as follows:

$$\omega(t) \simeq \overrightarrow{\Omega}_{\hat{n}}^T \overrightarrow{\Upsilon}_{\hat{n}}(t), \theta(t) \simeq \overrightarrow{\Theta}_{\hat{n}}^T \overrightarrow{\Upsilon}_{\hat{n}}(t), \tag{42}$$

where $\overrightarrow{\Omega}_{\hat{n}}$ and $\overrightarrow{\Theta}_{\hat{n}}$ are Bernoulli coefficient vectors of $\omega(t)$ and $\theta(t)$, respectively. By inserting the approximations (42) into Eq. (41), we have

$$\begin{cases} \overrightarrow{\Omega}_{\hat{n}}^T \overrightarrow{\Upsilon}_{\hat{n}}(t) = \mu\left(t, U_0 + \int_0^t \overrightarrow{\Omega}_{\hat{n}}^T \overrightarrow{\Upsilon}_{\hat{n}}(s) ds + \int_0^t \overrightarrow{\Theta}_{\hat{n}}^T \overrightarrow{\Upsilon}_{\hat{n}}(s) dB^{H(s)}(s)\right), \\ \overrightarrow{\Theta}_{\hat{n}}^T \overrightarrow{\Upsilon}_{\hat{n}}(t) = \sigma\left(t, U_0 + \int_0^t \overrightarrow{\Omega}_{\hat{n}}^T \overrightarrow{\Upsilon}_{\hat{n}}(s) ds + \int_0^t \overrightarrow{\Theta}_{\hat{n}}^T \overrightarrow{\Upsilon}_{\hat{n}}(s) dB^{H(s)}(s)\right). \end{cases} \tag{43}$$

Using operational matrices of integration defined in Eqs. (29) and (31), we have

$$\begin{cases} \overrightarrow{\Omega_n^T} \overrightarrow{Y_n}(t) = \mu(t, U_0 + \overrightarrow{\Omega_n^T} \overrightarrow{P_n} \overrightarrow{Y_n}(t) + \overrightarrow{\Theta_n^T} \overrightarrow{P_n^s} \overrightarrow{Y_n}(t)), \\ \overrightarrow{\Theta_n^T} \overrightarrow{Y_n}(t) = \sigma(t, U_0 + \overrightarrow{\Omega_n^T} \overrightarrow{P_n} \overrightarrow{Y_n}(t) + \overrightarrow{\Theta_n^T} \overrightarrow{P_n^s} \overrightarrow{Y_n}(t)). \end{cases} \quad (44)$$

We consider \hat{n} Newton-Cotes collocation nodes which are calculated as follows:

$$t_l = \frac{2l + 1}{2\hat{n}}, \quad l = 0, 1, \dots, \hat{n} - 1. \quad (45)$$

By inserting collocation points t_l into Eq. (44), we obtain the following nonlinear system of $2\hat{n}$ algebraic equations and $2\hat{n}$ unknowns

$$\begin{cases} \overrightarrow{\Omega_n^T} \overrightarrow{Y_n}(t_l) = \mu(t_l, U_0 + \overrightarrow{\Omega_n^T} \overrightarrow{P_n} \overrightarrow{Y_n}(t_l) + \overrightarrow{\Theta_n^T} \overrightarrow{P_n^s} \overrightarrow{Y_n}(t_l)), \\ \overrightarrow{\Theta_n^T} \overrightarrow{Y_n}(t_l) = \sigma(t_l, U_0 + \overrightarrow{\Omega_n^T} \overrightarrow{P_n} \overrightarrow{Y_n}(t_l) + \overrightarrow{\Theta_n^T} \overrightarrow{P_n^s} \overrightarrow{Y_n}(t_l)). \end{cases} \quad (46)$$

The approximate solution of Eq. (2) is determined after solving system (46) and computing unknown vectors as follows:

$$U(t) \simeq U_{\hat{n}}(t) = U_0 + \overrightarrow{\Omega_n^T} \overrightarrow{P_n} \overrightarrow{Y_n}(t) + \overrightarrow{\Theta_n^T} \overrightarrow{P_n^s} \overrightarrow{Y_n}(t). \quad (47)$$

The process of the proposed method is described in the step-by-step in Algorithm 3.

Algorithm 3.

- Input: The number \hat{n} , the vofBm $B^{H(t)}(t)$, the initial value U_0 , and the functions $\mu, \sigma : [0, 1] \times \mathbb{R} \rightarrow \mathbb{R}$.
- Output: The numerical solution of
- $$U(t) \simeq U_0 + \overrightarrow{\Omega_n^T} \overrightarrow{P_n} \overrightarrow{Y_n}(t) + \overrightarrow{\Theta_n^T} \overrightarrow{P_n^s} \overrightarrow{Y_n}(t).$$
- Step 1: Construct the vector $\overrightarrow{Y_n}(t) = [\mathfrak{B}_0(t), \mathfrak{B}_1(t), \dots, \mathfrak{B}_{\hat{n}-1}(t)]^T$ which $\mathfrak{B}_i(t)$
- for $i = 0, 1, \dots, \hat{n} - 1$ satisfied in Eq. (23).
- Step 2: Let $\omega(t) = \mu(t, U(t))$ and $\theta(t) = \sigma(t, U(t))$.
- Step 3: Define the Bernoulli coefficient vectors of $\omega(t)$ and $\theta(t)$ which are denoted by $\overrightarrow{\Omega_n}$ and $\overrightarrow{\Theta_n}$.
- Step 4: Compute the matrix $\overrightarrow{P_n}$ using Eq. (30).
- Step 5: Compute the matrix $\mathbf{G}_{\hat{n}} = (\mathbf{g}_{ij})$, which \mathbf{g}_{ij} for $i, j = 1, 2, \dots, \hat{n}$ are computed using Eq. (25).
- Step 6: Compute the matrix $\mathbf{A}_{\hat{n}} = (\mathbf{a}_{ij})$, which \mathbf{a}_{ij} for $i, j = 1, 2, \dots, \hat{n}$ are computed using Eq. (32).
- Step 7: Calculate the matrix $\overrightarrow{P_n^s} = \mathbf{G}_{\hat{n}}^{-1} \mathbf{A}_{\hat{n}} \mathbf{G}_{\hat{n}}$.
- Step 8: Consider \hat{n} Newton-Cotes collocation nodes t_l for $l = 0, 1, \dots, \hat{n} - 1$ using Eq. (45).
- Step 9: Put
- $$\begin{cases} \overrightarrow{\Omega_n^T} \overrightarrow{Y_n}(t_l) = \mu(t_l, U_0 + \overrightarrow{\Omega_n^T} \overrightarrow{P_n} \overrightarrow{Y_n}(t_l) + \overrightarrow{\Theta_n^T} \overrightarrow{P_n^s} \overrightarrow{Y_n}(t_l)), \\ \overrightarrow{\Theta_n^T} \overrightarrow{Y_n}(t_l) = \sigma(t_l, U_0 + \overrightarrow{\Omega_n^T} \overrightarrow{P_n} \overrightarrow{Y_n}(t_l) + \overrightarrow{\Theta_n^T} \overrightarrow{P_n^s} \overrightarrow{Y_n}(t_l)) \end{cases}$$
-

Algorithm 3.

Step 10: Solve the nonlinear system in Step 9 and compute the unknown vectors $\overrightarrow{\Omega_n}$ and $\overrightarrow{\Theta_n}$.

5 Error Analysis

Theorem 5 (Tohidi et al. 2013) Suppose that $f(t)$ is an infinity continuous function over the interval $[0, 1]$ and $f_{\hat{n}}(t)$ is the approximate function of $f(t)$ via Bernoulli polynomials. The following upper error bound has been achieved

$$|f(t) - f_{\hat{n}}(t)| \leq \frac{1}{(\hat{n} - 1)!} \max_{t \in [0,1]} \mathfrak{B}_{\hat{n}-1}(t) \max_{t \in [0,1]} \frac{d^{\hat{n}-1} f(t)}{dt^{\hat{n}-1}}. \quad (48)$$

Theorem 6 Assume that $\omega(t) = \mu(t, U(t))$ and $\theta(t) = \sigma(t, U(t))$ be the exact solutions of Eq. (41) such that satisfy in the Lipschitz condition, i.e.,

$$|\mu(t, U) - \mu(t, V)| + |\sigma(t, U) - \sigma(t, V)| \leq \mathcal{L}|U - V|. \quad (49)$$

Let

$$\omega_{\hat{n}}(t) = \mu(t, U_{\hat{n}}(t)), \quad \theta_{\hat{n}}(t) = \sigma(t, U_{\hat{n}}(t)), \quad (50)$$

and consider

$$\omega_{\hat{n}}^{\hat{n}}(t) = \mu_{\hat{n}}(t, U_{\hat{n}}(t)), \quad \theta_{\hat{n}}^{\hat{n}}(t) = \sigma_{\hat{n}}(t, U_{\hat{n}}(t)), \quad (51)$$

as the approximate solutions of the mentioned equation, where $\omega_{\hat{n}}^{\hat{n}}(t)$ and $\theta_{\hat{n}}^{\hat{n}}(t)$ are the estimation of $\omega_{\hat{n}}(t)$ and $\theta_{\hat{n}}(t)$ via Bernoulli polynomials. Also, suppose that the following assumption is satisfied

$$1 - \mathcal{L} - \mathcal{H}_{\hat{n}} \mathcal{L} > 0. \quad (52)$$

Then, the following upper error bound is obtained

$$|U(t) - U_{\hat{n}}(t)| \leq \frac{\mathcal{W}_{\hat{n}} + \mathcal{H}_{\hat{n}} \mathcal{T}_{\hat{n}}}{1 - \mathcal{L} - \mathcal{H}_{\hat{n}} \mathcal{L}}, \quad (53)$$

where

$$\begin{aligned} \mathcal{H}_{\hat{n}} &= \max_{t \in [0,1]} B^{H(t)}(t), \\ \mathcal{W}_{\hat{n}} &= \max_{t \in [0,1]} |\omega_{\hat{n}}(t) - \omega_{\hat{n}}^{\hat{n}}(t)| = \frac{1}{(\hat{n} - 1)!} \max_{t \in [0,1]} \mathfrak{B}_{\hat{n}-1}(t) \max_{t \in [0,1]} \frac{d^{\hat{n}-1} \omega_{\hat{n}}(t)}{dt^{\hat{n}-1}}, \\ \mathcal{T}_{\hat{n}} &= \max_{t \in [0,1]} |\theta_{\hat{n}}(t) - \theta_{\hat{n}}^{\hat{n}}(t)| = \frac{1}{(\hat{n} - 1)!} \max_{t \in [0,1]} \mathfrak{B}_{\hat{n}-1}(t) \max_{t \in [0,1]} \frac{d^{\hat{n}-1} \theta_{\hat{n}}(t)}{dt^{\hat{n}-1}}. \end{aligned} \quad (54)$$

Proof According to Eq. (49) and Theorem 5, we have

$$\begin{aligned} |\omega(t) - \omega_{\hat{n}}^{\hat{n}}(t)| &\leq |\omega(t) - \omega_{\hat{n}}(t)| + |\omega_{\hat{n}}(t) - \omega_{\hat{n}}^{\hat{n}}(t)| \leq \mathcal{L}|U(t) - U_{\hat{n}}(t)| + \mathcal{W}_{\hat{n}}, \\ |\theta(t) - \theta_{\hat{n}}^{\hat{n}}(t)| &\leq |\theta(t) - \theta_{\hat{n}}(t)| + |\theta_{\hat{n}}(t) - \theta_{\hat{n}}^{\hat{n}}(t)| \leq \mathcal{L}|U(t) - U_{\hat{n}}(t)| + \mathcal{T}_{\hat{n}}, \end{aligned} \tag{55}$$

where $\mathcal{W}_{\hat{n}}$ and $\mathcal{T}_{\hat{n}}$ have been defined in Eq. (54). On the other hand, the approximate solution of Eq. (40) is as follows:

$$U_{\hat{n}}(t) = U_0 + \int_0^t \omega_{\hat{n}}^{\hat{n}}(s) ds + \int_0^t \theta_{\hat{n}}^{\hat{n}}(s) dB^{H(s)}(s), \quad t \in [0, 1]. \tag{56}$$

Equations (40) and (56) yield

$$|U(t) - U_{\hat{n}}(t)| \leq \max_{t \in [0,1]} |\omega(t) - \omega_{\hat{n}}^{\hat{n}}(t)| + \mathcal{H}_{\hat{n}} \max_{t \in [0,1]} |\theta(t) - \theta_{\hat{n}}^{\hat{n}}(t)|, \tag{57}$$

where $\mathcal{H}_{\hat{n}} = \max_{t \in [0,1]} B^{H(t)}(t)$. From Eqs. (55) and (57), we get

$$|U(t) - U_{\hat{n}}(t)| \leq \mathcal{L}|U(t) - U_{\hat{n}}(t)| + \mathcal{W}_{\hat{n}} + \mathcal{H}_{\hat{n}}\mathcal{L}|U(t) - U_{\hat{n}}(t)| + \mathcal{H}_{\hat{n}}\mathcal{T}_{\hat{n}}. \tag{58}$$

From Eqs. (58) and (52), we conclude

$$|U(t) - U_{\hat{n}}(t)| \leq \frac{\mathcal{W}_{\hat{n}} + \mathcal{H}_{\hat{n}}\mathcal{T}_{\hat{n}}}{1 - \mathcal{L} - \mathcal{H}_{\hat{n}}\mathcal{L}}. \tag{59}$$

□

Remark 2 Equation (53) tells us that if $\hat{n} \rightarrow \infty$ then $\mathcal{W}_{\hat{n}}, \mathcal{T}_{\hat{n}}, \mathcal{H}_{\hat{n}} \rightarrow 0$, and therefore $\frac{\mathcal{W}_{\hat{n}} + \mathcal{H}_{\hat{n}}\mathcal{T}_{\hat{n}}}{1 - \mathcal{L} - \mathcal{H}_{\hat{n}}\mathcal{L}} \rightarrow 0$. It means that by increasing \hat{n} , the approximate solution $U_{\hat{n}}(t)$ tends to the exact solution $U(t)$.

6 Numerical Results

Two numerical examples are solved in this section via proposed method to check applicability and computational efficiency of the suggested technique. In reporting the values of absolute errors (AEs) for some used Bernoulli polynomials \hat{n} , we pursue two goals. The first goal is that the presented theoretical results in Sect. 5 are numerically investigated in this section, and the second goal is that presented numerical method is compared with Ccw method (Heydari et al. 2019).

Example 1 (Heydari et al. 2019) Consider the following SDE driven by vofBm

$$\begin{cases} dU(t) = v^2 \cos(U(t)) \sin^3(U(t)) dt - v \sin^2(U(t)) dB^{H(t)}(t), \\ U(0) = U_0, \end{cases} \tag{60}$$

such that its exact solution is given by

$$U(t) = \operatorname{arccot}(vB^{H(t)}(t) + \cot(U_0)). \tag{61}$$

- The presented method in the Sect. 4 has been employed to solve this example for three values $\hat{n} = 6, 8, 10$ and two functions $H(t) = 0.5 + 0.3 \sin(\pi t)$ and $H(t) = 0.6 - 0.2 \exp(-2t)$, and obtained AEs at some nodal points are reported in Tables 1 and 2, respectively. Other variable parameters are considered as $\mathbf{T} = 1, N = 50, \hat{m} = 100$, and $U_0 = v = \frac{1}{20}$. It is mentioned in Heydari et al. (2019) that the values of AEs obtained by considering $M = 3$ and $k = 3, 4, 5$. So, authors have applied Ccw vector with $2^k M = 24, 48, 96$ elements which this caused to generating larger matrices and more computations. The results of these tables reveal that our suggested method is more accurate and efficient than Ccw method, and indicate the effect of \hat{n} on the approximate solution. The value of \hat{n} have inverse relation with the values of AEs, and by increasing \hat{n} AEs decrease.
- Also, the behavior of AEs by considering mentioned parameters is illustrated in Fig. 1. The truth of our claim that the values of AEs decrease by increasing Bernoulli vector's size, can be more clearly seen from this diagram.
- For investigating the effect of initial value and v on the values of AEs, we consider the rest of the parameters as $N = 100, \hat{m} = 100, \hat{n} = 6, \mathbf{T} = 1, H(t) = 0.3 + 0.2 \cos(-3t)$, and run the Matlab code for different values of $U_0 = \frac{1}{10}, \frac{1}{20}, \frac{1}{30}$ and $v = \frac{1}{20}$. Once again, we consider the value of $U_0 = \frac{1}{20}$ fixed and run programming code for various values of $v = \frac{1}{10}, \frac{1}{20}, \frac{1}{30}$. The obtained results are reported in Table 3 and are plotted in Fig. 2 which demonstrate that there is a direct relation between initial value and the values of AEs. Furthermore, the values of AEs are also related on v directly.

Example 2 (Heydari et al. 2019) Consider the following SDE driven by vofBm

$$\begin{cases} dU(t) = v^2 U(t)(U^2(t) - 1) dt + v(1 - U^2(t)) dB^{H(t)}(t), \\ U(0) = U_0, \end{cases} \tag{62}$$

such that its exact solution is given by

$$U(t) = \frac{(1 + U_0) \exp(2vB^{H(t)}(t)) + U_0 - 1}{(1 + U_0) \exp(vB^{H(t)}(t)) - U_0 + 1}. \tag{63}$$

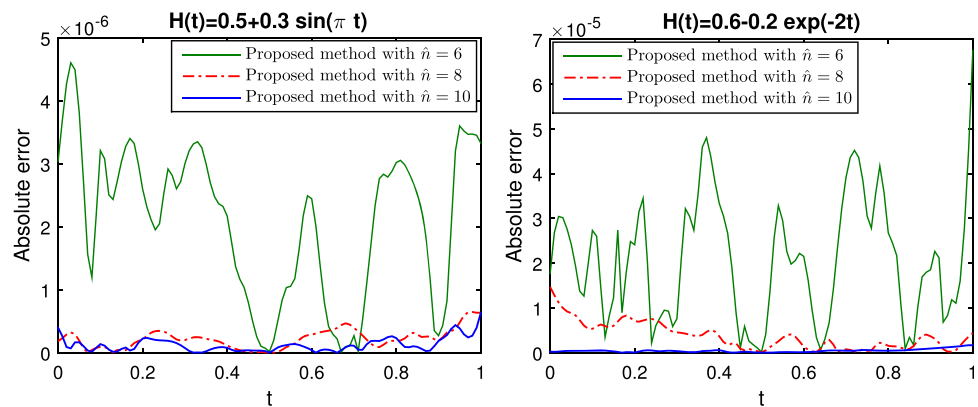
Table 1 Comparison of AEs of our method and Ccw method for Example 1 with $H(t) = 0.5 + 0.3 \sin(\pi t)$

t_l	Ccw method			Our method		
	$k = 3$	$k = 4$	$k = 5$	$\hat{n} = 6$	$\hat{n} = 8$	$\hat{n} = 10$
0.2	2.7463×10^{-6}	3.8793×10^{-6}	1.4045×10^{-6}	2.5981×10^{-6}	2.1483×10^{-7}	2.4771×10^{-7}
0.4	6.2365×10^{-6}	2.4425×10^{-7}	3.9707×10^{-7}	2.1713×10^{-6}	1.1339×10^{-7}	7.6957×10^{-8}
0.6	1.3982×10^{-6}	4.9411×10^{-7}	3.4729×10^{-7}	2.4453×10^{-6}	2.8758×10^{-7}	4.0530×10^{-8}
0.8	3.2472×10^{-6}	3.1180×10^{-6}	9.6406×10^{-7}	3.0202×10^{-6}	1.8635×10^{-7}	2.3194×10^{-7}
1.0	2.8907×10^{-6}	2.5445×10^{-6}	3.9374×10^{-7}	3.3302×10^{-6}	6.3521×10^{-7}	6.3279×10^{-7}

Table 2 Comparison of AEs of our method and Ccw method for Example 1 with $H(t) = 0.6 - 0.2 \exp(-2t)$

t_l	Ccw method			Our method		
	$k = 3$	$k = 4$	$k = 5$	$\hat{n} = 6$	$\hat{n} = 8$	$\hat{n} = 10$
0.2	5.0008×10^{-6}	2.9191×10^{-6}	1.8266×10^{-6}	2.4119×10^{-5}	7.0514×10^{-6}	2.8825×10^{-8}
0.4	2.8066×10^{-5}	2.6119×10^{-6}	1.0112×10^{-6}	3.3385×10^{-5}	3.6715×10^{-6}	5.6289×10^{-7}
0.6	5.1922×10^{-6}	2.1989×10^{-6}	1.5571×10^{-6}	2.1511×10^{-5}	1.3684×10^{-6}	7.2972×10^{-8}
0.8	5.0562×10^{-6}	4.9267×10^{-6}	1.7644×10^{-7}	2.6766×10^{-5}	1.5293×10^{-6}	4.6274×10^{-7}
1.0	6.1019×10^{-6}	2.6561×10^{-6}	1.7949×10^{-6}	6.7492×10^{-5}	4.3734×10^{-6}	1.6665×10^{-6}

Fig. 1 The graph of AEs for Example 1 with two selected $H(t)$



- The presented method in the Sect. 4 has been employed to solve this example for three values $\hat{n} = 8, 10, 12$ and two functions $H(t) = 0.7 + 0.2 \sin(\pi t)$ and $H(t) = 0.7 - 0.25 \exp(-t)$, and obtained AEs at some nodal points are reported in Tables 4 and 5, respectively. Other variable parameters are considered as $\mathbf{T} = 1, N = 50, \hat{m} = 100, U_0 = 0.01$, and $\nu = \frac{1}{30}$. It is mentioned in Heydari et al. (2019) that the values of AEs obtained by considering $M = 2$ and $k = 4, 5, 6$. So, authors have applied Ccw vector with $2^k M = 32, 64, 128$ elements which this caused to generating larger matrices and more computations. The results of these tables reveal that our suggested method is more accurate and efficient than Ccw method, and indicate the effect of \hat{n} on the approximate solution. The value

of \hat{n} have inverse relation with the values of AEs, and by increasing \hat{n} AEs decrease.

- Also, the behavior of AEs by considering mentioned parameters is illustrated in Fig. 3. The truth of our claim that the values of AEs decrease by increasing Bernoulli vector's size, can be more clearly seen from this diagram.
- For investigating the effect of initial value and ν on the values of AEs, we consider the rest of the parameters as $N = 100, \hat{m} = 100, \hat{n} = 8, \mathbf{T} = 1, H(t) = 0.4 + 0.3t^2$, and run the Matlab code for different values of $U_0 = 0.01, 0.005, 0.001$ and $\nu = \frac{1}{30}$. Once again, we consider the value of $U_0 = 0.01$ fixed and run programming code for various values of $\nu = \frac{1}{20}, \frac{1}{30}, \frac{1}{40}$. The obtained results are reported in Table 6 and are plotted in Fig. 4 which

Table 3 Investigating influence of the values U_0 and ν on the values of AEs

t_l	$\nu = \frac{1}{20}$			$U_0 = \frac{1}{20}$		
	$U_0 = \frac{1}{10}$	$U_0 = \frac{1}{20}$	$U_0 = \frac{1}{30}$	$\nu = \frac{1}{10}$	$\nu = \frac{1}{20}$	$\nu = \frac{1}{30}$
0.2	3.4867×10^{-4}	8.1241×10^{-5}	5.1581×10^{-5}	9.6260×10^{-5}	4.1608×10^{-5}	1.4877×10^{-5}
0.4	2.4509×10^{-4}	8.8081×10^{-6}	1.9126×10^{-5}	7.3460×10^{-5}	2.7682×10^{-5}	1.7641×10^{-5}
0.6	4.0395×10^{-4}	6.6373×10^{-5}	2.0259×10^{-6}	1.7593×10^{-4}	2.4037×10^{-5}	8.4145×10^{-6}
0.8	3.7852×10^{-4}	1.3575×10^{-4}	2.9688×10^{-5}	3.4780×10^{-5}	4.0285×10^{-5}	3.9365×10^{-5}
1.0	2.4989×10^{-4}	1.9926×10^{-4}	6.3116×10^{-5}	3.0326×10^{-4}	1.9249×10^{-4}	3.7012×10^{-5}

Fig. 2 Investigating the effect of U_0 (left) and parameter ν (right) on the values of AEs

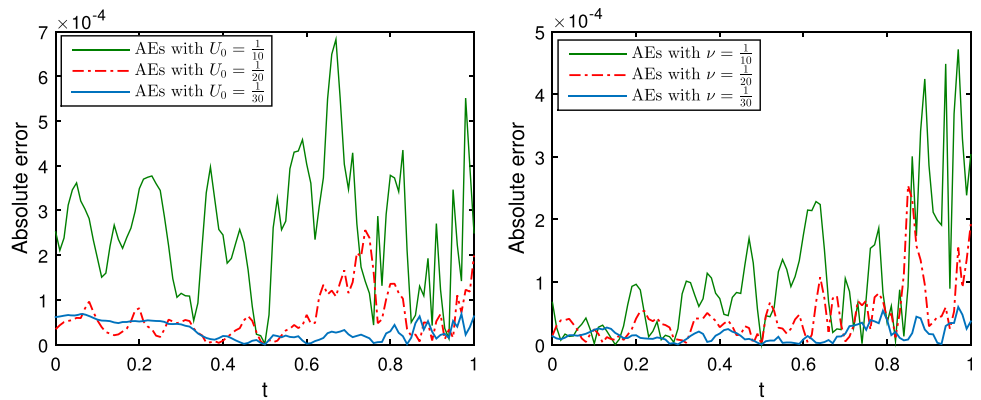


Table 4 Comparison of AEs of our method and Ccw method for Example 2 with $H(t) = 0.7 + 0.2 \sin(\pi t)$

t_l	Ccw method			Our method		
	$k = 4$	$k = 5$	$k = 5$	$\hat{n} = 8$	$\hat{n} = 10$	$\hat{n} = 12$
0.2	3.2850×10^{-5}	2.4033×10^{-4}	1.4634×10^{-5}	2.2231×10^{-4}	2.5254×10^{-5}	7.0943×10^{-6}
0.4	7.7123×10^{-5}	1.2578×10^{-5}	8.2132×10^{-6}	2.0816×10^{-4}	1.8391×10^{-5}	2.3021×10^{-6}
0.6	1.1454×10^{-4}	5.1047×10^{-5}	4.6802×10^{-5}	1.0375×10^{-4}	1.8600×10^{-5}	1.0191×10^{-5}
0.8	3.7300×10^{-6}	3.6659×10^{-6}	2.9566×10^{-6}	1.3822×10^{-4}	6.6835×10^{-5}	6.1411×10^{-6}
1.0	1.0563×10^{-3}	5.0603×10^{-4}	3.0478×10^{-4}	4.5514×10^{-5}	1.6431×10^{-4}	1.4041×10^{-6}

Table 5 Comparison of AEs of our method and Ccw method for Example 2 with $H(t) = 0.7 - 0.25 \exp(-t)$

t_l	Ccw method			Our method		
	$k = 4$	$k = 5$	$k = 5$	$\hat{n} = 8$	$\hat{n} = 10$	$\hat{n} = 12$
0.2	1.0811×10^{-4}	6.9681×10^{-4}	1.2367×10^{-4}	2.9124×10^{-4}	1.9955×10^{-4}	2.0255×10^{-5}
0.4	2.4154×10^{-4}	1.4780×10^{-4}	3.6764×10^{-4}	3.5388×10^{-4}	5.3817×10^{-5}	4.6241×10^{-5}
0.6	4.9532×10^{-4}	8.0529×10^{-5}	5.7225×10^{-4}	5.6374×10^{-4}	7.2291×10^{-5}	9.7434×10^{-6}
0.8	1.9303×10^{-4}	1.6314×10^{-4}	6.4132×10^{-5}	1.6360×10^{-4}	8.9914×10^{-5}	2.6797×10^{-5}
1.0	1.9402×10^{-3}	8.4869×10^{-4}	7.1071×10^{-4}	7.4854×10^{-4}	7.1312×10^{-6}	4.2582×10^{-5}

Fig. 3 The graph of AEs for Example 2 with two selected $H(t)$

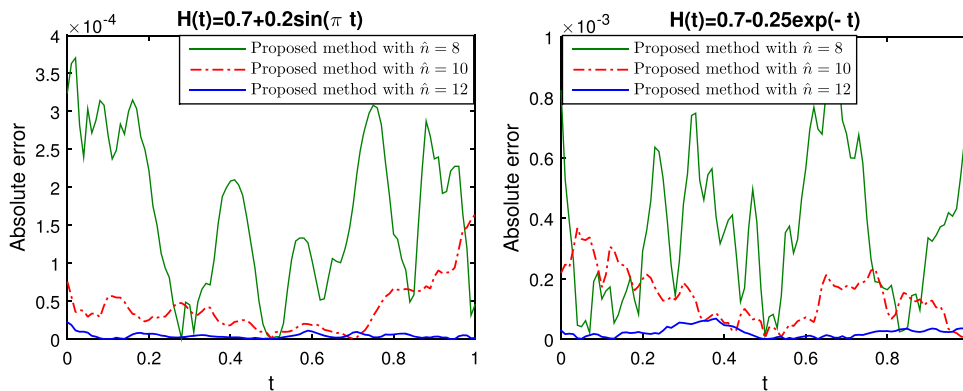
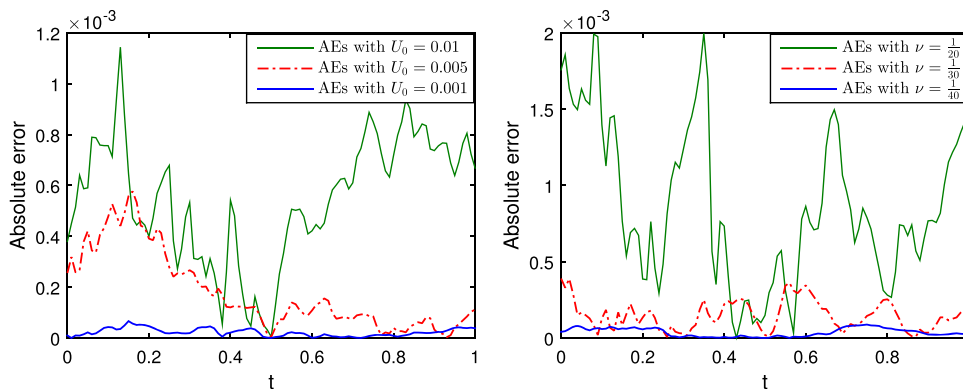


Table 6 Investigating influence of the values U_0 and ν on the values of AEs

t_i	$\nu = \frac{1}{30}$			$U_0 = 0.01$		
	$U_0 = 0.01$	$U_0 = 0.005$	$U_0 = 0.001$	$\nu = \frac{1}{20}$	$\nu = \frac{1}{30}$	$\nu = \frac{1}{40}$
0.2	3.9930×10^{-4}	3.9096×10^{-4}	4.6791×10^{-5}	3.8651×10^{-4}	1.3383×10^{-4}	5.6613×10^{-5}
0.4	5.4163×10^{-4}	1.2033×10^{-4}	2.0930×10^{-5}	7.3352×10^{-4}	2.2147×10^{-4}	7.3401×10^{-6}
0.6	4.2137×10^{-4}	1.0810×10^{-4}	6.8061×10^{-7}	8.6288×10^{-4}	3.4467×10^{-4}	9.9247×10^{-6}
0.8	7.1816×10^{-4}	3.9517×10^{-5}	8.0232×10^{-6}	2.8120×10^{-4}	2.5354×10^{-4}	6.3250×10^{-5}
1.0	6.6846×10^{-4}	1.1204×10^{-4}	3.8278×10^{-5}	1.5401×10^{-3}	2.2873×10^{-4}	2.4994×10^{-5}

Fig. 4 Investigating the effect of U_0 (left) and parameter ν (right) on the values of AEs



demonstrate that there is a direct relation between initial value and the values of AEs. Furthermore, the values of AEs are also related on ν directly.

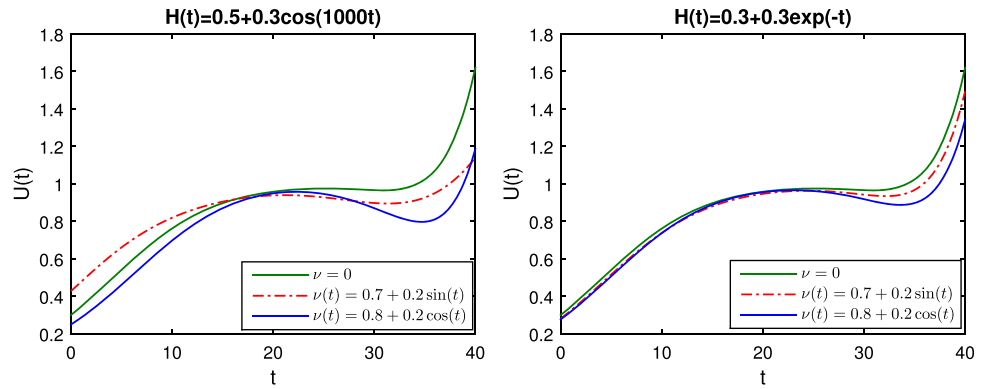
$$\begin{cases} dU(t) = rU(t)\left(1 - \frac{U(t)}{\tau}\right)dt, \\ U(0) = U_0. \end{cases} \tag{64}$$

In Eq. (64), $U(t)$, $\tau > 0$ and r denote the population size at time t , carrying capacity of environment, and population growth rate, respectively. As we know, the rate of population growth is uncertain in real world and it can be perturbed by white noise process $\zeta(t)$ as $r \rightarrow r + \nu\zeta(t)$ where $\zeta(t) = \frac{dB(t)}{dt}$ and ν is a constant number. So, the classical logistic equation (64) is transformed to the following stochastic logistic equation

7 Application in Real World

One of the most well-known equation in the ecology science is logistic equation which has the following form

Fig. 5 Simulation of stochastic logistic equation behavior via suggested method



$$\begin{cases} dU(t) = rU(t)\left(1 - \frac{U(t)}{\tau}\right)dt + vU(t)\left(1 - \frac{U(t)}{\tau}\right)dB(t), \\ U(0) = U_0. \end{cases} \tag{65}$$

Since the rate of population growth depends on time t , it is better to study non-autonomous form of stochastic logistic equation which obtain from $r \rightarrow r(t) + v(t)\xi(t)$. Furthermore, stochastic form of logistic equation driven by vofBm is introduced as follows:

$$\begin{cases} dU(t) = r(t)U(t)\left(1 - \frac{U(t)}{\tau}\right)dt + v(t)U(t)\left(1 - \frac{U(t)}{\tau}\right)dB^{H(t)}(t), \\ U(0) = U_0. \end{cases} \tag{66}$$

The introduced method in Sect. 4 is applied to solve Eq. (66) for $H(t) = 0.5 + 0.3 \cos(1000t)$ and $H(t) = 0.3 + 0.3 \exp(-t)$ and obtained results are plotted in Fig. 5. Also, other parameters are considered as $N = 100, \hat{m} = 100, \hat{n} = 8, U_0 = 0.3, r(t) = 0.2, \tau = 1$ and three values $v(t) = 0, v(t) = 0.8 + 0.2 \cos(t)$ and $v(t) = 0.7 + 0.2 \sin(t)$.

8 Conclusion and Future Works

The studied model in this paper is a nonlinear SDE driven by vofBm that it has no analytical solution in many situations. On the other hand, the complexity of this model is so great that until now only one numerical method has been proposed to solve it. In order to solve this model, first we have derived stochastic omi driven by vofBm, then this operator together with ordinary omi based on Bernoulli polynomials are used to convert mentioned model to a nonlinear system of algebraic equations. The obtained system is solved via Newton’s numerical method and the approximate solution of this equation is achieved. In Sect. 5, we theoretically proved that by increasing the number of used Bernoulli polynomials \hat{n} , the approximate solution tends to the exact solution. The effect of the number of used Bernoulli polynomials \hat{n} , initial value U_0 ,

and constant coefficient v , on the AEs values have been investigated in Sect. 6. Also, the presented method has been compared with Ccw method in the same section to confirm the superiority of our method respected to previous methods. The numerical results for different values of \hat{n} are reported in order to is established that by increasing \hat{n} the approximate solution converges to the exact solution. Reported numerical results in Sect. 6 confirm that one can obtain accurate approximate solution even by using small number of basic functions and performing few calculation efforts. Also, numerical results demonstrate that the values of U_0 and v have an direct relationship with the values of AEs, i.e., by reducing the values of U_0 and v , the AEs values are also reduced. It seems that the values of N and \hat{m} are affected on the error values which investigation of this fact is recommended for future research works.

Funding The authors have not disclosed any funding.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Abbasbandy S, Kazem S, Alhuthali MS, Alsulami HH (2015) Application of the operational matrix of fractional-order Legendre functions for solving the time-fractional convection-diffusion equation. *Appl Math Comput* 266:31–40

Asgari M, Khodabin M (2017) Computational method based on triangular operational matrices for solving nonlinear stochastic differential equations. *Int J Nonlinear Anal Appl* 8(2):169–179

Badawi H, Arqub OA, Shawagfeh N (2023) Well-posedness and numerical simulations employing Legendre-shifted spectral approach for Caputo-Fabrizio fractional stochastic integro differential equations. *Int J Mod Phys C* 34(06):2350070

Badawi H, Arqub OA, Shawagfeh N (2023) Stochastic integrodifferential models of fractional orders and Leffler nonsingular kernels: well-posedness theoretical results and Legendre Gauss

- spectral collocation approximations, *Chaos. Solit Fract* X 10:100091
- Badawi H, Shawagfeh N, Arqub OA (2022) Fractional conformable stochastic integro differential equations: Existence, uniqueness, and numerical simulations utilizing the shifted Legendre spectral collocation algorithm. *Math Probl Eng*
- Balcı MA, Sezer M (2016) Hybrid Euler-Taylor matrix method for solving of generalized linear Fredholm Integro-differential difference equations. *Appl Math Comput* 273:33–41
- Bazm S (2015) Bernoulli polynomials for the numerical solution of some classes of linear and nonlinear integral equations. *J Comput Appl Math* 275:44–60
- Chen Y, Liu L, Li B, Sun Y (2014) Numerical solution for the variable order linear cable equation with Bernstein polynomials. *Appl Math Comput* 238:329–341
- Corlay S, Lebovits J, Véhel JL (2014) Multifractional stochastic volatility models. *Math Finance* 24(2):364–402
- Dareiotis K, Leahy JM (2016) Finite difference schemes for linear stochastic integro-differential equations. *Stoch Proc Appl* 126(10):3202–3234
- Dehghan M, Abbaszadeh M (2018) A Legendre spectral element method (SEM) based on the modified bases for solving neutral delay distributed-order fractional damped diffusion-wave equation. *Math Meth Appl Sci* 41(9):3476–3494
- Echelard A, Véhel JL, Barrière O (2010) Terrain modeling with multifractional Brownian motion and self-regulating processes, International conference on computer vision and graphics, Berlin Heidelberg: Springer
- Hashemi B, Khodabin M, Maleknejad K (2017) Numerical solution based on hat functions for solving nonlinear stochastic Itô-Volterra integral equations driven by fractional Brownian motion. *Mediterr J Math* 14(1):1–15
- Heydari MH, Avazzadeh Z (2018) An operational matrix method for solving variable-order fractional biharmonic equation. *Comput Appl Math* 37(4):4397–4411
- Heydari MH, Hooshmandasl MR, Badri Loghmani G, Cattani C (2016) Wavelet Galerkin method for solving stochastic heat equation. *Int J Comput Math* 93.9:1579–1596
- Heydari MH, Atangana A, Avazzadeh Z (2019) Chebyshev polynomials for the numerical solution of fractal-fractional model of nonlinear Ginzburg-Landau equation. *Eng, Comput (In Press)*
- Heydari MH, Avazzadeh Z, Mahmoudi MR (2019) Chebyshev cardinal wavelets for nonlinear stochastic differential equations driven with variable-order fractional Brownian motion. *Chaos Soliton Fract* 124:105–124
- Kamrani M (2016) Convergence of Galerkin method for the solution of stochastic fractional integro differential equations. *Optik* 127:10049–10057
- Kamrani M, Jamshidi N (2017) Implicit Euler approximation of stochastic evolution equations with fractional Brownian motion. *Commun Nonlinear Sci Numer Simulat* 44:1–10
- Khodabin M, Maleknejad K, Rostami M, Nouri M (2012) Numerical approach for solving stochastic Volterra-Fredholm integral equations by stochastic operational matrix. *Comput Math Appl* 64(6):1903–1913
- Khodabin M, Maleknejad K, Hossoini Shekarabi F (2013) Application of triangular functions to numerical solution of stochastic Volterra integral equations. *IAENG Int J Appl Math* 43(1):1–9
- Kilicman A, Al Zhou ZA (2007) Kronecker operational matrices for fractional calculus and some applications. *Appl Math Comput* 187.1:250–265
- Li C, Zhao Z, Chen Y (2011) Numerical approximation of nonlinear fractional differential equations with subdiffusion and superdiffusion. *Comput Math Appl* 62(3):855–875
- Li M, Gu XM, Huang C, Fei M, Zhang G (2018) A fast linearized conservative finite element method for the strongly coupled nonlinear fractional Schrödinger equations. *J Comput Phys* 358:256–282
- Maayah B, Moussaoui A, Bushnaq S, Arqub OA (2022) The multistep Laplace optimized decomposition method for solving fractional-order coronavirus disease model (COVID-19) via the Caputo fractional approach. *Demonstr Math* 55(1):963–977
- Maleknejad K, Khodabin M, Rostami M (2012) Numerical solution of stochastic Volterra integral equations by a stochastic operational matrix based on block pulse functions. *Math Comput Model* 55(3):791–800
- Maleknejad K, Khodabin M, Rostami M (2012) A numerical method for solving m-dimensional stochastic Itô-Volterra integral equations by stochastic operational matrix. *Comput Math Appl* 63(1):133–143
- Mirzaee F, Samadyar N (2018) Numerical solution of nonlinear stochastic Itô-Volterra integral equations driven by fractional Brownian motion. *Math Meth Appl Sci* 41(4):1410–1423
- Mirzaee F, Samadyar N (2018) Using radial basis functions to solve two dimensional linear stochastic integral equations on non-rectangular domains. *Eng Anal Bound Elem* 92:180–195
- Mirzaee F, Samadyar N (2019) Numerical solution based on two-dimensional orthonormal Bernstein polynomials for solving some classes of two-dimensional nonlinear integral equations of fractional order. *Appl Math Comput* 344–345:191–203
- Mirzaee F, Samadyar N (2019) On the numerical solution of fractional stochastic Integro-differential equations via meshless discrete collocation method based on radial basis functions. *Eng Anal Bound Elem* 100:246–255
- Rahimkhani P, Ordokhani Y, Babolian E (2017) Application of fractional-order Bernoulli functions for solving fractional Riccati differential equation. *Int J Nonlinear Anal Appl* 8(2):277–292
- Rahimkhani P, Ordokhani Y, Lima PM (2019) An improved composite collocation method for distributed-order fractional differential equations based on fractional Chelyshkov wavelets. *Appl Numer Math* 145:1–27
- Roodaki M, Almasieh H (2012) Delta basis functions and their applications to systems of integral equations. *Comput Math Appl* 63:100–109
- Samadyar N, Mirzaee F (2020) Orthonormal Bernoulli polynomials collocation approach for solving stochastic Itô-Volterra integral equations of Abel type. *Int J Numer Model* 33(1):1–14
- Sheng H, Sun H, Chen Y, Qiu T (2011) Synthesis of multi fractional gaussian noises based on variable-order fractional operator. *Signal Process* 91(7):1645–1650
- Sheng H, Chen YQ, Qiu TS (2012) Fractional processes and fractional-order signal processing: techniques and applications. Springer, New York
- Singh S, Patel VK, Singh VK, Tohidi E (2018) Application of Bernoulli matrix method for solving two-dimensional hyperbolic telegraph equations with Dirichlet boundary conditions. *Comput Math Appl* 75(7):2280–2294
- Tohidi E, Bhrawy AH, Erfani K (2013) A collocation method based on Bernoulli operational matrix for numerical solution of generalized pantograph equation. *Appl Math Model* 37:4283–4294
- Tripathi MP, Baranwal VK, Pandey RK, Singh OP (2013) A new numerical algorithm to solve fractional differential equations

based on operational matrix of generalized hat functions. Commun Nonlinear Sci Numer Simulat 18(6):1327–1340

ur Rehman M, Khan RA (2011) The Legendre wavelet method for solving fractional differential equations. Commun Nonlinear Sci Numer Simul 16(11):4163–4173

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.