Journal of Computational Mathematics
Vol.xx, No.x, 2024, 1–22.

http://www.global-sci.org/jcm
doi:10.4208/jcm.2402-m2023-0104

AN OVERLAPPING DOMAIN DECOMPOSITION SPLITTING ALGORITHM FOR STOCHASTIC NONLINEAR SCHRÖDINGER EQUATION[*]

Lihai Ji
Institute of Applied Physics and Computational Mathematics, Beijing 100094, China;
Shanghai Zhangjiang Institute of Mathematics, Shanghai 201203, China
Email: jilihai@lsec.cc.ac.cn

Abstract

A novel overlapping domain decomposition splitting algorithm based on a Crank-Nicolson method is developed for the stochastic nonlinear Schrödinger equation driven by a multiplicative noise with non-periodic boundary conditions. The proposed algorithm can significantly reduce the computational cost while maintaining the similar conservation laws. Numerical experiments are dedicated to illustrating the capability of the algorithm for different spatial dimensions, as well as the various initial conditions. In particular, we compare the performance of the overlapping domain decomposition splitting algorithm with the stochastic multi-symplectic method in [S. Jiang et al., Commun. Comput. Phys., 14 (2013), 393–411] and the finite difference splitting scheme in [J. Cui et al., J. Differ. Equ., 266 (2019), 5625–5663]. We observe that our proposed algorithm has excellent computational efficiency and is highly competitive. It provides a useful tool for solving stochastic partial differential equations.

Mathematics subject classification: 60H35, 35Q55, 60H15.
Key words: Stochastic nonlinear Schrödinger equation, Domain decomposition method, Operator splitting, Overlapping domain decomposition splitting algorithm.

1. Introduction

The main purpose of this work is to propose an innovative overlapping domain decomposition splitting (ODDS for short) algorithm for the stochastic nonlinear Schrödinger (NLS) equation driven by a multiplicative noise

$$\begin{cases} \mathrm{i}du = [\Delta u + \lambda |u|^2 u]dt + \varepsilon u \circ dW(t), & t \in (0,T], \\ u(0,x) = u_0(x), & x \in D \subset \mathbb{R}^d, \quad d \geq 1 \end{cases} \tag{1.1}$$

with Dirichlet boundary conditions, $\varepsilon > 0$, $\lambda \in \mathbb{R}$ and W an $L^2(D;\mathbb{R})$-valued Q-Wiener process defined on a complete filtered probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t\in[0,T]}, \mathbb{P})$. More precisely, $W(t)$ has the following Karhunen-Loève expansion:

$$W(t) = \sum_{k\in\mathbb{N}^d} Q^{\frac{1}{2}} e_k \beta_k(t), \quad t \in [0,T]$$

with$\{e_k\}_{k\in\mathbb{N}^d}$ being an orthonormal basis of $L^2(D;\mathbb{R})$, $\{\beta_k\}_{k\in\mathbb{N}^d}$ being a sequence of real-valued mutually independent and identically distributed Brownian motions, and $Qe_k = \eta_k e_k$ for $\eta_k \geq 0$,

[*] Received May 12, 2023 / Revised version received December 26, 2023 / Accepted February 19, 2024 / Published online April 29, 2024 /

$k \in \mathbb{N}^d$. For convenience, we always consider the equivalent Itô form of (1.1)

$$\mathrm{i}du = \left[\Delta u + \lambda |u|^2 u - \frac{\mathrm{i}}{2}\varepsilon^2 F_Q u\right] dt + \varepsilon u dW(t) \tag{1.2}$$

with the initial value $u(0) = u_0$ and

$$F_Q(x) := \sum_{k \in \mathbb{N}^d} \left(Q^{\frac{1}{2}} e_k(x)\right)^2.$$

In the last two decades, much progress has been made in theoretical analysis and numerical approximation for the stochastic NLS equation. To numerically inherit the charge conservation law and the geometric structure of (1.1), [8, 9, 13, 23] propose the stochastic symplectic and multi-symplectic algorithms. Particularly, the authors in [11] applies the large deviation principle to investigate the probabilistic superiority of the stochastic symplectic algorithms. Very recently, the authors in [17] discover the stochastic symplectic structure of the stochastic Schrödinger equation on Wasserstein manifold at the first time. To preserve the ergodicity of the numerical solution of (1.1), [20, 21] study the ergodic numerical approximations. To reduce the computational cost of (1.1), [22] designs a parareal algorithm and [12, 14, 25, 26] propose the splitting algorithm, respectively. For more details about other kinds of numerical approximations of the stochastic NLS equation, we refer to [2, 3, 5, 6, 10, 15, 16, 18] and references therein. These existing semi-discretizations and full discretizations for the stochastic NLS equation mentioned above are all investigated under the assumption of homogeneous or periodic boundary conditions. It is worth to point out that the soliton solution of the nonlinear dispersive wave propagation problems in a very large or unbounded domain for the stochastic NLS equation is an interesting and important subject in applications (see, e.g. [1]). However, solving high dimensional stochastic partial differential equations can be a computationally intensive task, particularly when the computational domain is very large. To simulate such systems in a moderate amount of time, we must employ high-performance computing. This motivates us to construct highly efficient and numerically stable algorithms for the d-dimensional stochastic NLS equation (1.1) in a large spatial domain with inhomogeneous or non-periodic boundary conditions.

To this end, we first apply the splitting technique of [25] to the Eq. (1.1) and get a deterministic linear PDE with random initial datum and a nonlinear stochastic PDE

$$\mathrm{i}du^{[1]} = \Delta u^{[1]} dt, \tag{1.3}$$

$$\mathrm{i}du^{[2]} = \lambda \left|u^{[2]}\right|^2 u^{[2]} dt + \varepsilon u^{[2]} \circ dW(t). \tag{1.4}$$

Then, for the subsystem (1.4) we can get the analytic solution due to the point-wise conservation law $|u(t,x)| = |u_0(x)|$. The key issues lie in the numerical approximation for the subsystem (1.3), we first discretize it based on the Crank-Nicolson scheme in the temporal direction and get a temporal semi-discretization.

To overcome the difficulties introduced by the non-periodic boundary conditions, we use the Chebyshev pseudo-spectral interpolation idea in space. To efficiently exploit modern high performance computing platforms, it is essential to design high performance algorithms. Domain decomposition method provides a useful tool to develop fast and efficient solvers for stochastic PDEs with a large number of random inputs. The non-overlapping domain decomposition method for PDEs with random coefficients is first proposed in [27] and then extended by [28] to quantify uncertainty in large-scale simulations. We refer to [19,24,29,30] and references therein for more details about the theory and applications of the domain decomposition method to PDEs with the random input.

We combine the Chebyshev interpolation idea and the overlapping domain decomposition method to approximate the temporal semi-discretization of (1.3) and thus obtain a full discretization of (1.3). The explicitness of the solution of (1.4) together with the full discretization of (1.3) gives us an ODDS algorithm for the stochastic NLS equation (1.1). Finally, several numerical examples for the stochastic NLS equation in one and two-dimensional spaces are presented to illustrate the capability of the proposed algorithm, which can be calculated high efficient. To the best of our knowledge, this is the first domain decomposition result of numerical approximations for stochastic PDEs whose the stochasticity comes from the stochastic source.

The rest of this paper is organized as follows. In Section 2, we present and analyze the ODDS algorithm for the stochastic NLS equation. In Section 2.1, we show the algorithm for the one-dimensional stochastic NLS equation. In Section 2.2, we focus on studying the ODDS algorithm for the two-dimensional case. Section 3 contains some numerical experiments for the stochastic NLS equation to demonstrate the accuracy and efficiency of the proposed algorithm. Concluding remarks are given in Section 4.

2. The ODDS Algorithm for the Stochastic NLS Equation

In this section, we devote to obtaining the ODDS algorithm for the stochastic NLS equation in one and multi-dimensional spaces.

2.1. An ODDS algorithm for the one-dimensional stochastic NLS equation

This part concentrates mainly on demonstrating an ODDS algorithm for the following stochastic nonlinear problem:

$$\mathrm{i}du = \big[u_{xx} + \lambda |u|^2 u\big]dt + \varepsilon u \circ dW(t), \quad t \in (0,T] \tag{2.1}$$

with an initial datum

$$u(0,x) = u_0(x), \quad x \in D = [x_L, x_R],$$

and the boundary conditions

$$u(t,x_L) = f(t), \quad u(t,x_R) = g(t), \quad t \in (0,T]. \tag{2.2}$$

It is well known that the stochastic NLS equation (1.1) possesses the charge conservation law under the homogeneous or periodic boundary conditions (see, e.g. [4, Proposition 4.4]), that is

$$\int_D |u(t,x)|^2 dx = \int_D |u_0(x)|^2 dx, \quad \mathbb{P}\text{-a.s.} \tag{2.3}$$

for all $t \in [0,T]$. Furthermore, if we define the Hamiltonian

$$H(u) = \frac{1}{2}\int_D |\nabla u(x)|^2 dx - \frac{\lambda}{4}\int_D |u(x)|^4 dx,$$

then the averaged energy $\mathbb{E}[H(u(t))]$ satisfies (see, e.g. [4, Proposition 4.5])

$$\mathbb{E}\big[H\big(u(t)\big)\big] = \mathbb{E}[H(u_0)] + \frac{\varepsilon^2}{2}\int_0^t \int_D |u(s,x)|^2 \sum_{k\in\mathbb{N}^d} \big|\nabla\big(Q^{\frac{1}{2}} e_k(x)\big)\big|^2 dxds. \tag{2.4}$$

In general, there is no charge conservation law or the averaged energy evolution law for the stochastic NLS equation with the boundary conditions (2.2).

(a) Operator Splitting

We use a splitting technique which was proposed in [25] to discretize (2.1) and obtain

$$\mathrm{i}du^{[1]} = u^{[1]}_{xx}dt, \tag{2.5}$$

$$\mathrm{i}du^{[2]} = \left(\lambda\big|u^{[2]}\big|^2 u^{[2]} - \frac{\mathrm{i}}{2}F_Q u^{[2]}\right)dt + \varepsilon u^{[2]} dW(t). \tag{2.6}$$

Note that for the nonlinear subsystem (2.6), we have the following useful result. We refer readers to [25, Proposition 3.1] for more details.

Proposition 2.1. *Assume the initial datum u_0 is $\mathcal{F}_0$-measurable L^2-valued random variable. Then the solution of*

$$\mathrm{i}du^{[2]} = \left(\lambda\big|u^{[2]}\big|^2 u^{[2]} - \frac{\mathrm{i}}{2}F_Q u^{[2]}\right)dt + \varepsilon u^{[2]} dW(t)$$

is given by

$$u^{[2]}(t,x) = u_0(x)\exp\big\{-\mathrm{i}\big(t\lambda|u_0(x)|^2 + \varepsilon W(t,x)\big)\big\}$$

due to $|u^{[2]}(t,x)| = |u_0(x)|$.

Motivated by this proposition, we get the following recursion in $T_n = (t_n, t_{n+1}]$, $t_n = n\tau$, $n \in \{0,1,\ldots,N-1\}$:

$$u^{n+1} = \exp\big\{-\mathrm{i}\big(\tau\lambda|u^n|^2 + \varepsilon\Delta W^{n+1}\big)\big\}u^n \tag{2.7}$$

with $u^0 = u_0$ and $\Delta W^{n+1} := W(t_{n+1},x) - W(t_n,x)$.

(b) Overlapping Domain Decomposition Method

Now we are in a position to approximate the linear subsystem (2.5). Denoting by p and q the real and imaginary parts of the solution $u^{[1]}$ of (2.5), which satisfy

$$dp = q_{xx}dt, \quad dq = -p_{xx}dt. \tag{2.8}$$

Applying the Crank-Nicolson method to discretize the above equations in the temporal direction yields

$$p^{n+1} = p^n + \frac{\tau}{2}\big(q^{n+1}_{xx} + q^n_{xx}\big), \quad q^{n+1} = q^n - \frac{\tau}{2}\big(p^{n+1}_{xx} + p^n_{xx}\big) \tag{2.9}$$

for all $n = 0,1,\ldots,N-1$.

Before we come to the spatial discretization of (2.9), let us introduce some basic concepts of the overlapping domain decomposition method. Let

$$V^m := \big[x^m_L, x^m_R\big], \quad x^1_L = x_L, \quad x^M_R = x_R, \quad m = 1,2,\ldots,M$$

be a uniform partition of $D = [x_L, x_R]$ with the spatial step size Δx, thus $D = \cup_{m=1}^{M} V^m$. Further, we consider a uniform partition of V^m, $m = 1, 2, \ldots, M$, with $J+1$ grid points in each fine interval, i.e.

$$x_L^m = x_0^m < x_1^m < \cdots < x_{J-1}^m < x_J^m = x_R^m, \quad m = 1, 2, \ldots, M.$$

Differing from the traditional spatial partition, here we require the last two points of the element V^m coincide with the first two points of the element V^{m+1}, that is

$$x_0^1 = x_L, \quad x_J^M = x_R, \quad x_{J-1}^m = x_0^{m+1}, \quad x_J^m = x_1^{m+1}, \quad m = 1, 2, \ldots, M-1.$$

In this situation, we remark that $\Delta x \neq (x_R - x_L)/M$. The general idea of the overlapping domain decomposition method is displayed in Fig. 2.1.

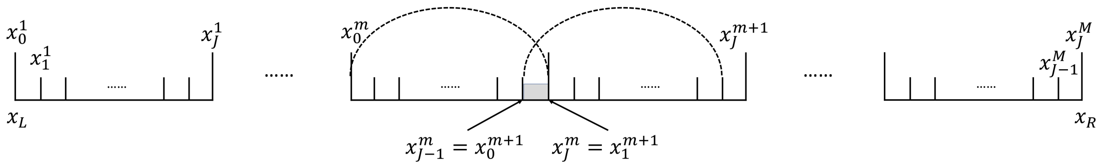

Fig. 2.1. Basic idea of the overlapping domain decomposition method.

After the above preliminaries, in element V^m, $m = 1, 2, \ldots, M$, we have

$$p^{m,n+1} = p^{m,n} + \frac{\tau}{2}\big(q_{xx}^{m,n+1} + q_{xx}^{m,n}\big), \quad q^{m,n+1} = q^{m,n} - \frac{\tau}{2}\big(p_{xx}^{m,n+1} + p_{xx}^{m,n}\big), \tag{2.10}$$

then $u^{[1],m} := p^m + \mathrm{i}q^m$ is an approximation of the solution of (2.5) over the m-th element.

To discretize these equations with the boundary conditions (2.2), we mainly use the Chebyshev-Gauss-Lobatto quadrature points (see, e.g. [7, Chapter 4]) in the interval $[-1,1]$ of the form

$$\eta_j = \cos\left(\frac{J-j}{J}\pi\right), \quad j = 0, 1, \ldots, J.$$

Since (2.10) holds for the interval $[x_L^m, x_R^m]$, in order to use the Chebyshev interpolation technique we need to introduce the following transformation:

$$\eta^m : \big[x_L^m, x_R^m\big] \;\to\; [-1,1], \quad y^m \;\mapsto\; \frac{2}{x_R^m - x_L^m} y^m - \frac{x_R^m + x_L^m}{x_R^m - x_L^m}, \tag{2.11}$$

then after a straightforward calculation we arrive at

$$\Delta x = \frac{x_R - x_L}{M + (1-M)\big(1 - \cos(\pi/J)\big)/2}.$$

Example 2.1. If $J = 2$, then

$$\Delta x = \frac{2(x_R - x_L)}{M+1}.$$

Example 2.2. If $M = 1$, then $\Delta x = x_R - x_L$. After the uniform partition of $D = [x_L, x_R]$ with $J+1$ grid points, we derive the overlapping domain decomposition method (or classical finite difference method) with the uniform spatial step size $(x_R - x_L)/J$.

Now, we present the interpolation $p^m(\eta^m,t)$ and $q^m(\eta^m,t)$ on the Chebyshev-Gauss-Lobatto collocation points $\eta^m \in [-1,1]$ given by

$$\begin{aligned} p^m(\eta^m,t) &= \sum_{j=0}^{J} \widetilde{p}_j^m(t)\phi_j^m(\eta^m) := \big(\Phi^m(\eta^m)\big)^\top P^m(t), \\ q^m(\eta^m,t) &= \sum_{j=0}^{J} \widetilde{q}_j^m(t)\phi_j^m(\eta^m) := \big(\Phi^m(\eta^m)\big)^\top Q^m(t), \end{aligned} \tag{2.12}$$

where $\Phi^m = [\phi_0^m, \cdots, \phi_J^m]^\top$ with ϕ_j being the Lagrangian interpolation function, $P^m = [\widetilde{p}_0^m, \cdots, \widetilde{p}_J^m]^\top$, and $Q^m = [\widetilde{q}_0^m, \cdots, \widetilde{q}_J^m]^\top$.

Taking the second-order partial derivatives of p^m and q^m with respect to y^m and using the transformation (2.11), it holds that

$$\begin{aligned} \frac{\partial^2 p^m(y^m,t)}{\partial y^{m,2}} &= \left(\frac{2}{x_R^m - x_L^m}\right)^2 \frac{\partial^2 p^m(\eta^m,t)}{\partial \eta^{m,2}} = \left(\frac{2}{x_R^m - x_L^m}\right)^2 (\Phi^m)^\top D^{m,(2)} P^m, \\ \frac{\partial^2 q^m(y^m,t)}{\partial y^{m,2}} &= \left(\frac{2}{x_R^m - x_L^m}\right)^2 \frac{\partial^2 q^m(\eta^m,t)}{\partial \eta^{m,2}} = \left(\frac{2}{x_R^m - x_L^m}\right)^2 (\Phi^m)^\top D^{m,(2)} Q^m, \end{aligned} \tag{2.13}$$

respectively, where $D^{m,(2)}$ is a $(J+1)\times(J+1)$ differential matrix on the element V^m. It can be checked that the entries of $D^{m,(r)}$ for any $r \geq 1$ are given by

$$[D^{m,(r)}]_{i+1,j+1} = \begin{cases} \dfrac{r}{\eta_j^m - \eta_i^m}\left(\dfrac{e_j}{e_i}[D^{m,(r-1)}]_{i+1,i+1} - [D^{m,(r-1)}]_{i+1,j+1}\right), & i \neq j, \\ -\displaystyle\sum_{k=0,k\neq i}^{J} [D^{m,(r)}]_{i+1,k+1}, & i = j \end{cases} \tag{2.14}$$

with

$$e_0 = \frac{(-1)^J}{2}, \quad e_J = \frac{1}{2}, \quad e_j = (-1)^{J-j}, \quad j = 1,2,\ldots,J-1.$$

We refer to [7,31] for more details. Particularly, the entries of $D^{m,(1)}$ are defined as

$$D_{i,j}^{m,(1)} = \begin{cases} -\dfrac{2J^2+1}{6}, & \text{if} \quad i=j=0, \\ \dfrac{2J^2+1}{6}, & \text{if} \quad i=j=J, \\ \dfrac{\eta_i^m}{2[1-(\eta_i^m)^2]}, & \text{if} \quad i=j\neq 0,J, \\ -\dfrac{c_i}{c_j}\dfrac{1}{\eta_i^m - \eta_j^m}, & \text{if} \quad i\neq j \end{cases}$$

with

$$c_0 = 2(-1)^J, \quad c_J = 2, \quad c_j = (-1)^{J-j}, \quad j = 1,2,\ldots,J-1.$$

Based on the above overlapping idea, we introduce the partition of D with the following grid points:

$$x_L = x_0^1 < x_1^1 < \cdots < x_{J-1}^1 < x_1^2 < \cdots < x_{J-1}^{m-1} < x_1^m < \cdots < x_{J-1}^M < x_J^M = x_R.$$

Denote by

$$\begin{aligned}\widetilde{\Phi} &= \left[\phi_0^1(\eta^1), \cdots, \phi_{J-1}^1(\eta^1), \phi_1^2(\eta^2), \cdots, \phi_J^M(\eta^M)\right]^\top \in \mathbb{R}^{M(J-1)+2},\\ \widetilde{P} &= \left[\widetilde{p}_0^1(t), \cdots, \widetilde{p}_{J-1}^1(t), \widetilde{p}_1^2(t), \cdots, \widetilde{p}_J^M(t)\right]^\top \in \mathbb{R}^{M(J-1)+2},\\ \widetilde{Q} &= \left[\widetilde{q}_0^1(t), \cdots, \widetilde{q}_{J-1}^1(t), \widetilde{q}_1^2(t), \cdots, \widetilde{q}_J^M(t)\right]^\top \in \mathbb{R}^{M(J-1)+2}.\end{aligned}$$

It follows from (2.12) and the fact $D = \cup_{m=1}^M V^m$ that the interpolation of functions p and q at arbitrary collocation point $x \in D$ reads as

$$p(x) \approx \widetilde{\Phi}^\top \widetilde{P}, \quad q(x) \approx \widetilde{\Phi}^\top \widetilde{Q},$$

which, along with (2.13) yields

$$\frac{d^2 p(x)}{dx^2} \approx \widetilde{\Phi}^\top \widetilde{D}^{(2)} \widetilde{P}, \quad \frac{d^2 q(x)}{dx^2} \approx \widetilde{\Phi}^\top \widetilde{D}^{(2)} \widetilde{Q}, \tag{2.15}$$

where

$$\begin{aligned}&\widetilde{D}^{(2)}_{[0:J-1,0:J]} = \frac{4}{\Delta x^2} D^{1,(2)}_{[0:J-1,0:J]},\\ &\widetilde{D}^{(2)}_{[m(J-1)+1:(m+1)(J-1),m(J-1):(m+1)(J-1)+1]} = \frac{4}{\Delta x^2} D^{m+1,(2)}_{[1:J-1,0:J]}, \quad m = 1, 2, \ldots, M-2,\\ &\widetilde{D}^{(2)}_{[(M-1)(J-1)+1:M(J-1)+1,(M-1)(J-1):M(J-1)+1]} = \frac{4}{\Delta x^2} D^{M,(2)}_{[1:J,0:J]},\end{aligned}$$

and the remaining elements of the differential matrix $\widetilde{D}^{(2)}$ are zero.

Consequently, combining the temporal semi-discretization (2.9) and the Chebyshev interpolation (2.15), we can obtain the following full discretization of (2.5):

$$U^{n+1} = U^n - \frac{\mathrm{i}}{2}\tau\left(\widetilde{\Phi}^\top \widetilde{D}^{(2)} U^{n+1} + \widetilde{\Phi}^\top \widetilde{D}^{(2)} U^n\right) \tag{2.16}$$

for all $n = 0, 1, \ldots, N-1$, where $U^n = \widetilde{P}^n + \mathrm{i}\widetilde{Q}^n$ and

$$\begin{aligned}\widetilde{P}^n &= \left[\widetilde{p}_0^1(t_n), \cdots, \widetilde{p}_{J-1}^1(t_n), \widetilde{p}_1^2(t_n), \cdots, \widetilde{p}_J^M(t_n)\right]^\top,\\ \widetilde{Q}^n &= \left[\widetilde{q}_0^1(t_n), \cdots, \widetilde{q}_{J-1}^1(t_n), \widetilde{q}_1^2(t_n), \cdots, \widetilde{q}_J^M(t_n)\right]^\top.\end{aligned}$$

(c) ODDS algorithm

Based on the analytic expression (2.7) and the full discretization (2.16), we have the following algorithm to compute the numerical solution to the one-dimensional stochastic NLS equation (2.1).

Algorithm 2.1: Computing the Numerical Solution to the One-dimensional Stochastic NLS Equation.

Choose the algorithm's parameters: time interval $[0, T]$, space domain $[x_L, x_R]$, temporal step size τ, number of elements M, grid points J, orthonormal basis $\{e_k(x)\}_{k\geq 1}$ and its truncation $\{e_k(x)\}_{k=1}^K$ to determine the Q-Wiener process $\Delta W_j^{K,n+1}$.

Step 1. For each $n = 1, 2, \ldots, K-1, j = 1, 2, \ldots, M(J-1)$, take u_j^n as the initial datum, solve (2.7) on the time interval T_n and get

$$u_j^* = \exp\big\{ -\mathrm{i}\big(\tau\lambda|u_j^n|^2 + \varepsilon\Delta W_j^{K,n+1}\big)\big\}u_j^n,$$

where $\Delta W_j^{K,n+1} = W^K(t_{n+1}, x_j) - W^K(t_n, x_j)$.

Step 2. Let $U^* = (u_1^*, u_2^*, \cdots, u_{M(N-1)}^*)^\top$. For each $n = 1, 2, \ldots, N-1$, take U^* as the initial datum, solve (2.16) on the time interval T_n and get

$$U^{n+1} = U^* - \frac{\mathrm{i}}{2}\tau\big(\widetilde{\Phi}^\top \widetilde{D}^{(2)} U^{n+1} + \widetilde{\Phi}^\top \widetilde{D}^{(2)} U^*\big).$$

Step 3. On the n-th time step (at time $t_n = n\tau$), generate the Gaussian random variables $\beta_k(t_n)$. According to (2.14), compute the elements of $D^{m,(2)}$ for $m = 1, 2, \ldots, M$.

2.2. The ODDS algorithm for the multi-dimensional stochastic NLS equation

In this subsection, we present the ODDS algorithm for the d-dimensional stochastic NLS equation. Without loss of generality we restrict our discussion to the case $d = 2$. Consider the following two-dimensional stochastic nonlinear system:

$$\mathrm{i}du = \big[u_{xx} + u_{yy} + \lambda|u|^2 u\big]dt + \varepsilon u \circ dW(t), \quad t \in (0, T] \tag{2.17}$$

with an initial condition

$$u(0, x, y) = u_0(x, y), \quad x \in D = [x_L, x_R] \times [y_L, y_R],$$

and the boundary conditions

$$\begin{aligned} u(t, x_L, y) &= f_1(t), \quad u(t, x_R, y) = g_1(t), \quad t \in (0, T], \quad y \in [y_L, y_R], \\ u(t, x, y_L) &= f_2(t), \quad u(t, x, y_R) = g_2(t), \quad t \in (0, T], \quad x \in [x_L, x_R]. \end{aligned}$$

By using a similar technique as for the one-dimensional case, we split (2.17) into the linear part and the nonlinear part. To define the ODDS algorithm for the two-dimensional case, the main difference lies in dealing with the linear part

$$\mathrm{i}du = (u_{xx} + u_{yy})dt.$$

We use the local one dimensional idea to split the above equations as

$$\mathrm{i}du = u_{xx}dt, \tag{2.18}$$

$$\mathrm{i}du = u_{yy}dt. \tag{2.19}$$

Then, the algorithm developed in the previous subsection can be utilized to approximate the above four subsystems. The ODDS algorithm for the two-dimensional stochastic NLS equation (2.17) is presented as follows.

Algorithm 2.2: The ODDS Algorithm for the Two-dimensional Stochastic NLS Equation.

Choose the algorithm's parameters: time interval $[0, T]$, space domain $[x_L, x_R] \times [y_L, y_R]$, temporal step size τ, number of elements M_1 and M_2 in x, y-directions, respectively, grid points J_1 and J_2 in x, y-directions, respectively, orthonormal basis $\{e_{k_1,k_2}(x, y)\}_{k_1,k_2\geq 1}$ and its truncation $\{e_k(x, y)\}_{k_1,k_2=1}^K$ to determine the Q-Wiener process $\Delta W_{j_1,j_2}^{K,n+1}$.

Step 1. For each $n = 1, 2, \dots, N-1$, $j_1 = 1, 2, \dots, M_1(J_1-1)$ and $j_2 = 1, 2, \dots, M_2(J_2-1)$, take $u^n_{j_1,j_2}$ as the initial datum, solve (2.7) on the time interval T_n and get

$$u^*_{j_1,j_2} = \exp\big\{ -\mathrm{i}\big(\tau\lambda\big|u^n_{j_1,j_2}\big|^2 + \varepsilon\Delta W^{K_1,K_2,n+1}_{j_1,j_2}\big)\big\} u^n_{j_1,j_2},$$

where $\Delta W^{K_1,K_2,n+1}_{j_1,j_2} = W^{K_1,K_2}(t_{n+1}, x_{j_1}, y_{j_2}) - W^{K_1,K_2}(t_n, x_{j_1}, y_{j_2})$.

Step 2. Let $U^* = (u^*_{1,1}, u^*_{2,1}, \cdots, u^*_{M_1(J_1-1),1}, u^*_{1,2}, u^*_{2,2}, \cdots, u^*_{M_1(J_1-1),M_2(J_2-1)})^\top$. Take U^* as the initial datum, solve (2.18) on the time interval T_n by (2.16) and get

$$U^{**} = U^* - \frac{\mathrm{i}}{2}\tau\big(\widetilde{\Phi}^\top \widetilde{D}^{(2)} U^{**} + \widetilde{\Phi}^\top \widetilde{D}^{(2)} U^*\big).$$

Step 3. For each $n = 1, 2, \dots, N-1$, take U^{**} as the initial datum, solve (2.19) on the time interval T_n by (2.16) and get

$$U^{n+1} = U^* - \frac{\mathrm{i}}{2}\tau\big(\widetilde{\Phi}^\top \widetilde{D}^{(2)} U^{n+1} + \widetilde{\Phi}^\top \widetilde{D}^{(2)} U^{**}\big).$$

Step 4. On the n-th time step (at time $t_n = n\tau$), generate the Gaussian random variables $\beta_{k_1,k_2}(t_n)$. According to (2.14), compute the elements of $D^{m,(2)}$ for $m = 1, 2, \dots, M_1$ and $m = 1, 2, \dots, M_2$.

Remark 2.1. For the S-dimensional stochastic NLS equation with $S \geq 3$, we only need to split the linear part of the considered system into S subsystems

$$\mathrm{i}du = u_{x_s x_s} dt, \quad s = 1, 2, \dots, S,$$

and then use the similar algorithm as in Algorithm 2.2.

3. Numerical Experiments

In this section we provide several numerical examples to illustrate the accuracy and capability of the algorithms developed in the previous section under the homogeneous Dirichlet boundary conditions or the inhomogeneous Dirichlet boundary conditions.

We first present some preliminaries used throughout the following numerical implementation of Algorithms 2.1 and 2.2.

3.1. Preliminaries of the numerical implementation

For d=1, we take the eigenvalues $\{\eta_k\}_{k=1}^K$ and the orthonormal basis $\{e_k\}_{k=1}^K$ of $L^2([x_L, x_R])$ as

$$e_k(x) = \sqrt{2}\sin(k\pi x), \quad \eta_k = \frac{1}{k^3},$$

which implies

$$\Delta W^{K,n+1}_j = \sum_{k=1}^K \sqrt{\frac{2}{x_R - x_L}}\sqrt{\frac{1}{k^3}} \sin\left(\frac{k\pi(x_j - x_L)}{x_R - x_L}\right)\big(\beta_k(t_{n+1}) - \beta_k(t_n)\big).$$

Here and in what follows, we take $K = 500$.

For $d=2$, we take the eigenvalues $\{\eta_{k_1,k_2}\}_{k_1,k_2=1}^{K}$ and the orthonormal basis $\{e_{k_1,k_2}\}_{k_1,k_2=1}^{K}$ of $L^2([x_L,x_R]\times[y_L,y_R])$ as

$$e_{k_1,k_2}(x,y)=2\sin(k_1\pi x)\sin(k_2\pi y),\quad \eta_{k_1,k_2}=\frac{1}{\left(k_1^2+k_2^2\right)^2},$$

which implies

$$\begin{aligned}\Delta W_{j_1,j_2}^{K,n+1}=&\sum_{k_1,k_2=1}^{K}\frac{2}{k_1^2+k_2^2}\sqrt{\frac{1}{(x_R-x_L)(y_R-y_L)}}\sin\left(\frac{k_1\pi(x_{j_1}-x_L)}{x_R-x_L}\right)\\&\times\sin\left(\frac{k_2\pi(y_{j_2}-y_L)}{y_R-y_L}\right)\big(\beta_{k_1,k_2}(t_{n+1})-\beta_{k_1,k_2}(t_n)\big).\end{aligned}$$

Denote

$$\mathbf{U}^n=\left[\widetilde{q}_1^1(t_n),\cdots,\widetilde{q}_{J-1}^1,\widetilde{q}_1^2,\cdots,\widetilde{q}_{J-1}^M(t_n),\widetilde{p}_1^1(t_n),\cdots,\widetilde{p}_{J-1}^1,\widetilde{p}_1^2,\cdots,\widetilde{p}_{J-1}^M(t_n)\right]^\top,$$

then the full discretization (2.16) can be rewritten as an algebraic system

$$(A\otimes B+C)\mathbf{U}^{n+1}=(-A\otimes B+C)\mathbf{U}^n+F,\tag{3.1}$$

where $B=\widetilde{D}^{(2)}_{[1:M(J-1),1:M(J-1)]}$, $F\in\mathbb{R}^{2M(J-1)}$ describes the boundary conditions, and

$$A=\begin{bmatrix}-\dfrac{\tau}{2} & 0\\ 0 & \dfrac{\tau}{2}\end{bmatrix}_{2\times2},\quad C=\begin{bmatrix}0 & I\\ I & 0\end{bmatrix}_{2M(J-1)\times 2M(J-1)}.$$

We will compute (3.1) using the Matlab command Algorithm 3.1. Furthermore, once the differential matrix $D^{m,(2)}$ is known, then (3.1) provides a feasible way to solve the stochastic NLS equation. The Matlab command Algorithm 3.2 relies on (2.14) to compute the elements of $D^{m,(2)}$ for $m=1,2,\ldots,M$. Since no confusion can arise, we simply drop the superscript $m,(2)$ on $D^{m,(2)}$.

In order to demonstrate the efficiency and superiority of the proposed algorithm, we compare the ODDS algorithm with the following ones:

(1) The stochastic multi-symplectic method (SMM for short, see [23, Eq. (2.24)])

$$\begin{aligned}\mathrm{i}\Big(\delta_t^+u_{j+\frac12}^n+\delta_t^+u_{j-\frac12}^n\Big)=&\,2\delta_x^+\delta_x^-u_j^{n+\frac12}+\lambda\Big|u_{j+\frac12}^{n+\frac12}\Big|^2u_{j+\frac12}^{n+\frac12}+\lambda\Big|u_{j-\frac12}^{n+\frac12}\Big|^2u_{j-\frac12}^{n+\frac12}\\&+\varepsilon u_{j+\frac12}^{n+\frac12}\dot{W}_{j+\frac12}^{n+\frac12}+\varepsilon u_{j-\frac12}^{n+\frac12}\dot{W}_{j-\frac12}^{n+\frac12},\end{aligned}\tag{3.2}$$

where

$$\delta_t^+u^n=\frac{u^{n+1}-u^n}{\tau},\quad \delta_x^+u_j=\frac{u_{j+1}-u_j}{h_x},\quad \delta_x^-u_j=\frac{u_j-u_{j-1}}{h_x}.$$

(2) The finite difference splitting Crank-Nicolson scheme (FDSCN for short, see [14, Eq. (57)])

$$\begin{aligned}u_j^*&=u_j^n+\mathrm{i}\tau\left(\delta_x^+\delta_x^-u_j^{n+\frac12,*}+\frac{\lambda}{2}\big(\big|u_j^n\big|^2+|u_j^*|^2\big)u_j^{n+\frac12,*}\right),\\u_j^{n+1}&=\exp\big(-\mathrm{i}\varepsilon\Delta W_j^{n+1}\big)u_j^*,\end{aligned}\tag{3.3}$$

where

$$u^{n+\frac12,*}=\frac{u^n+u^*}{2}.$$

Algorithm 3.1: Code to compute the solution of a large sparse algebraic equation $Gx = b$, where $G \in \mathbb{R}^{L\times L}$ is a sparse matrix and $b \in \mathbb{R}^L$ with $L = 2M(J-1)$. The input $x0$ is an arbitrary non-zero column vector of length L.

```
function x=matrix_solve(x0,G,b,L)
r=b-G*x0; u=zeros(length(x0),1);
while max(abs(r))>0.00001
 v(:,1)=r/norm(r);
for j=1:m
 d=G*v(:,j);
for i=1:j
 H(i,j)=v(:,i)'*d;
end
 u(:)=0;
for i=1:j
 u=H(i,j)*v(:,i)+u;
end
 u=d-u; H(j+1,j)=norm(u);
if (H(j+1,j)<0.0001||j==L)
 e=zeros(j+1,1); e(1)=norm(r);
 y=pinv(H(1:j+1,1:j))*e;
 x0=x0+v(:,1:j)*y;r=b-G*x0;
 break;
end
v(:,j+1)=u/H(j+1,j);
end
end
```

Algorithm 3.2: Code to Compute the Differential Matrix $D \in \mathbb{R}^{(J+1)\times(J+1)}$.

```
function D=chebyshve_solve(J)
D=zeros(J+1,J+1);
K=(0:J)'; x=cos(pi*K/J);
c=ones(J+1,1); c(1)=2; c(J+1)=2;
for k=1:J+1
  for j=1:J+1
    if (j==1&k==1)||(j==J+1&k==J+1)
     D(k,j)=(2*J^2+1)/6;
    elseif j==k
     D(k,j)=-x(k)/2/(1-x(k)^2);
    else
     D(k,j)=c(k)/c(j)*(-1)^(k+j)/(x(k)-x(j));
    end
  end
end
D(k,j)=-D(k,j);
```

3.2. Numerical examples

After these preparations, now we concentrate on the numerical performance of the ODDS algorithms.

Example 3.1. In this example we show the soliton propagation at different instants of the following equation:

$$\mathrm{i}du = \big[u_{xx} + |u|^2 u\big]dt + \varepsilon u \circ dW(t), \quad t \in (0, T] \tag{3.4}$$

in Figs. 3.1 and 3.2 with $\varepsilon = 0.01, 0.05$ and the initial condition

$$u_0(x) = \sqrt{\frac{6}{5}}\mathrm{sech}\big(\sqrt{2}x\big)e^{\mathrm{i}x}.$$

As is stated in (2.3), Eq. (3.4) possesses the charge conservation law almost surely under the homogeneous or periodic boundary conditions. Here, we verify this result by using our algorithm

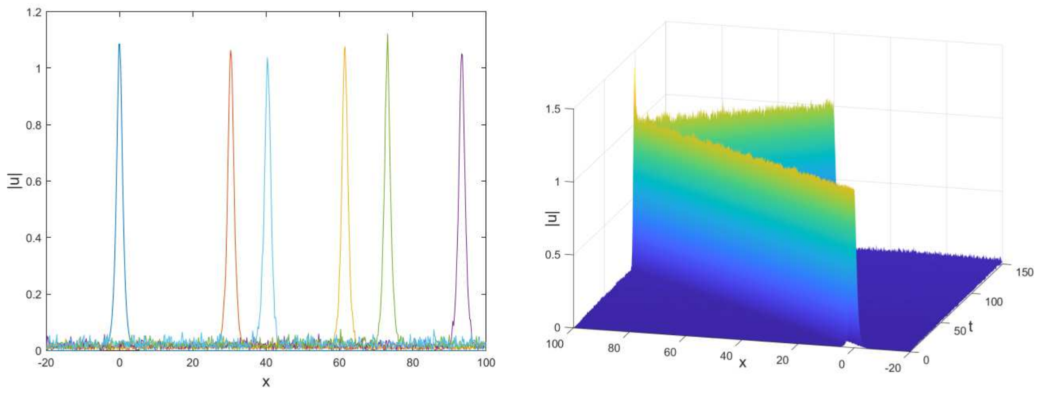

Fig. 3.1. The soliton propagation of (3.4) with inhomogeneous Dirichlet boundary conditions in $[-20, 100]$ and $\varepsilon = 0.01$. $J = 30, M = 10, T = 150, \tau = 0.015$.

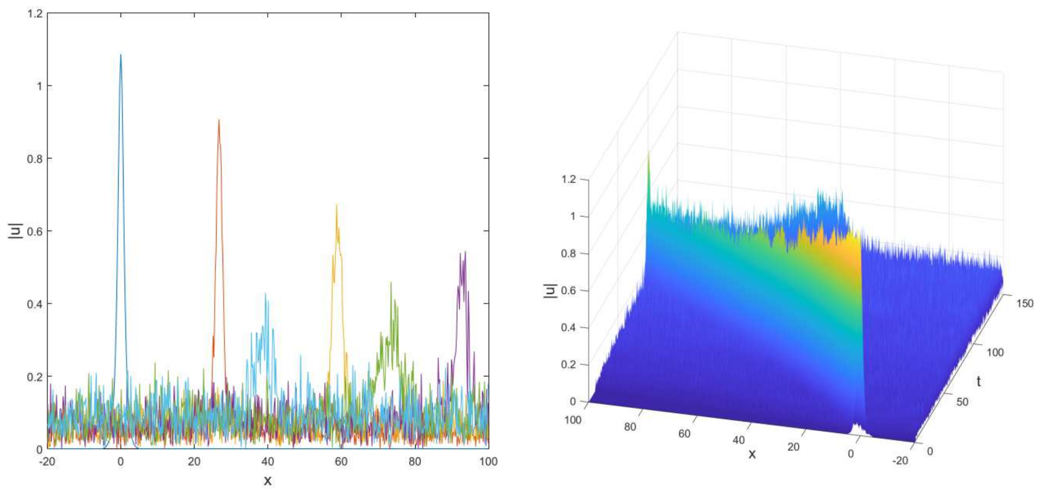

Fig. 3.2. The soliton propagation of (3.4) with inhomogeneous Dirichlet boundary conditions in $[-20, 100]$ and $\varepsilon = 0.05$. $J = 30, M = 10, T = 150, \tau = 0.015$.

with the homogeneous Dirichlet boundary conditions. Fig. 3.3 presents the evolution of the discrete charge conservation law and the conservation errors of Algorithm 2.1 with $\varepsilon = 0.01$ and 0.05. We observe a good agreement with the continuous result.

Fig. 3.4 investigates the evolution of the discrete energy of the ODDS algorithm for different values of $\varepsilon = 0.01$ and 0.05, where the blues lines denote the discrete energies over 50 trajectories, the red line represents the discrete averaged energy, and the black line shows the discrete energy in the deterministic case. We see from the numerical experiment results that the discrete averaged energy possesses a linear growth property over 50 trajectories.

Now we compare the computational costs of the ODDS Algorithm 2.1, the SMM method (3.2) and the FDSCN scheme (3.3) for one-dimensional problem (3.4) under the homogeneous Dirichlet boundary conditions. Fig. 3.5 demonstrates the computational efficiency of our ODDS algorithm in comparison with the SMM method and the FDSCN scheme. The reported CPU time is in seconds.

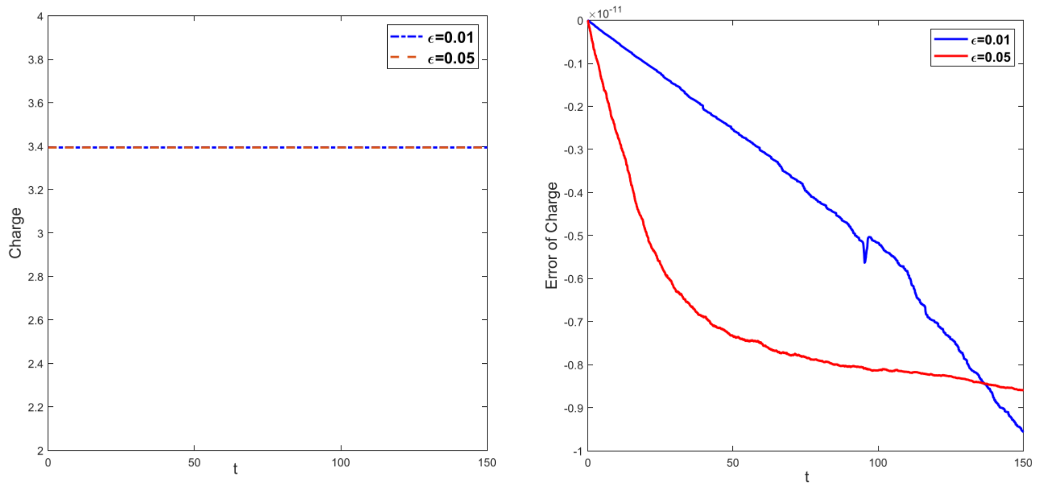

Fig. 3.3. Evolution of the discrete charge (left), and the conservation error (right), over one trajectory with $\varepsilon = 0.01, 0.05$. Homogeneous Dirichlet boundary conditions in $[-20, 100]$. J=30, M=10, T=150, $\tau = 0.015$.

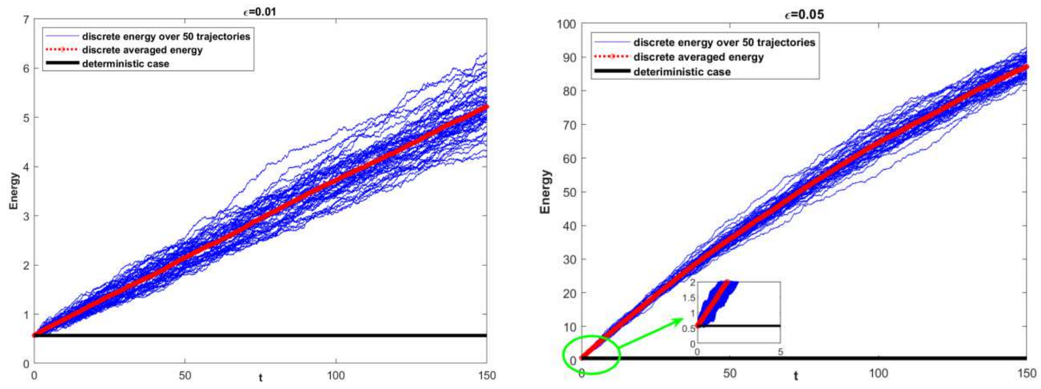

Fig. 3.4. Evolution of the discrete energies for $\varepsilon = 0.01$ (left), and $\varepsilon = 0.05$ (right). Homogeneous Dirichlet boundary conditions in $[-20, 100]$. $J = 30$, $M = 10$, $T = 150$, $\tau = 0.015$.

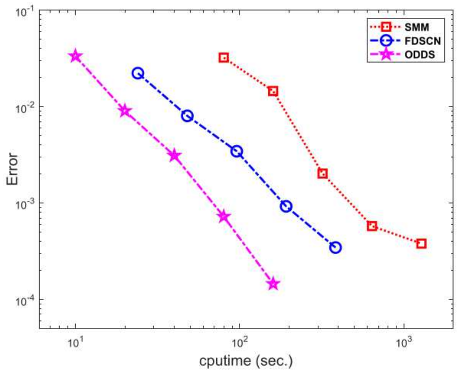

Fig. 3.5. Efficiency for the ODDS algorithm, the SMM method and the FDSCN scheme for (3.4). Homogeneous Dirichlet boundary conditions in $[-20, 100]$. $J = 30, M = 20, T = 150, \tau = 0.015$. The mesh sizes of the SMM method and the FDSCN scheme are given by $h_x = 0.2$ (i.e. $J = 600$, $M = 1$).

Example 3.2. In this example we present the double soliton collision of (3.4) with the initial condition

$$u_0(x) = \sqrt{\frac{6}{5}}\operatorname{sech}\big(\sqrt{2}x\big)e^{2ix} + \sqrt{\frac{3}{5}}\operatorname{sech}\big(\sqrt{2}(x-30)\big)e^{0.5i(x-30)}. \tag{3.5}$$

The solution is simulated with the inhomogeneous Dirichlet boundary conditions in $[-20, 150]$ along one trajectory with $\varepsilon = 0.01$. Figs. 3.6-3.8 show the double soliton collisions at different times $t = 0, 12, 60$ for the real part p, the imaginary part q and the module u, respectively.

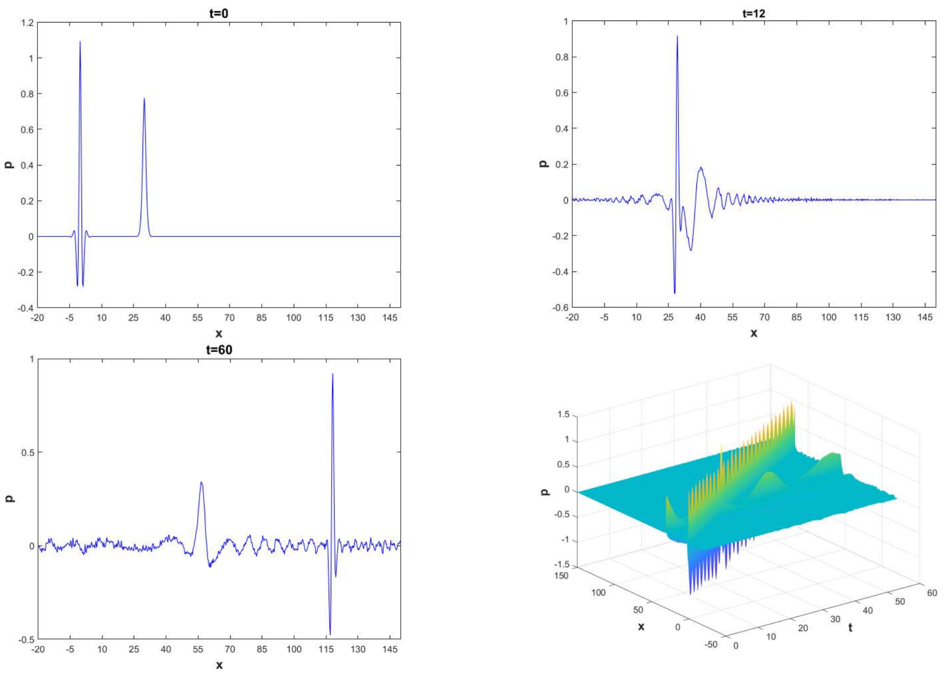

Fig. 3.6. The double soliton collision of (3.4) for the real part p with the initial condition (3.5) along one trajectory. Inhomogeneous Dirichlet boundary conditions in $[-20, 150]$. $J = 20, M = 5, T = 60$, $\tau = 0.006$.

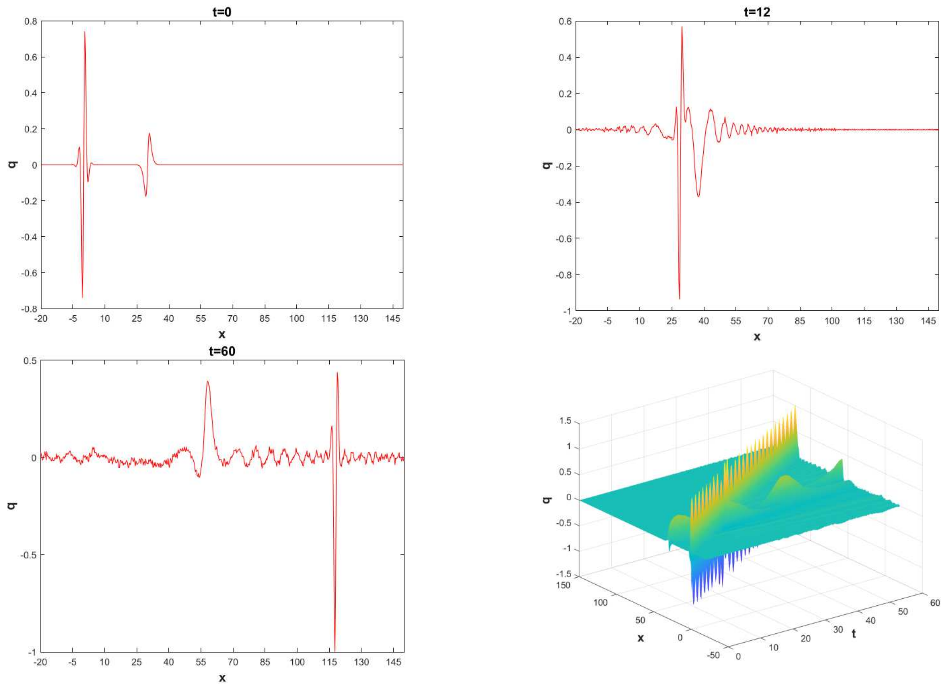

Fig. 3.7. The double soliton collision of (3.4) for the imaginary part q with the initial condition (3.5) along one trajectory. Inhomogeneous Dirichlet boundary conditions in $[-20, 150]$. J=20, M=5, T=60, $\tau = 0.006$.

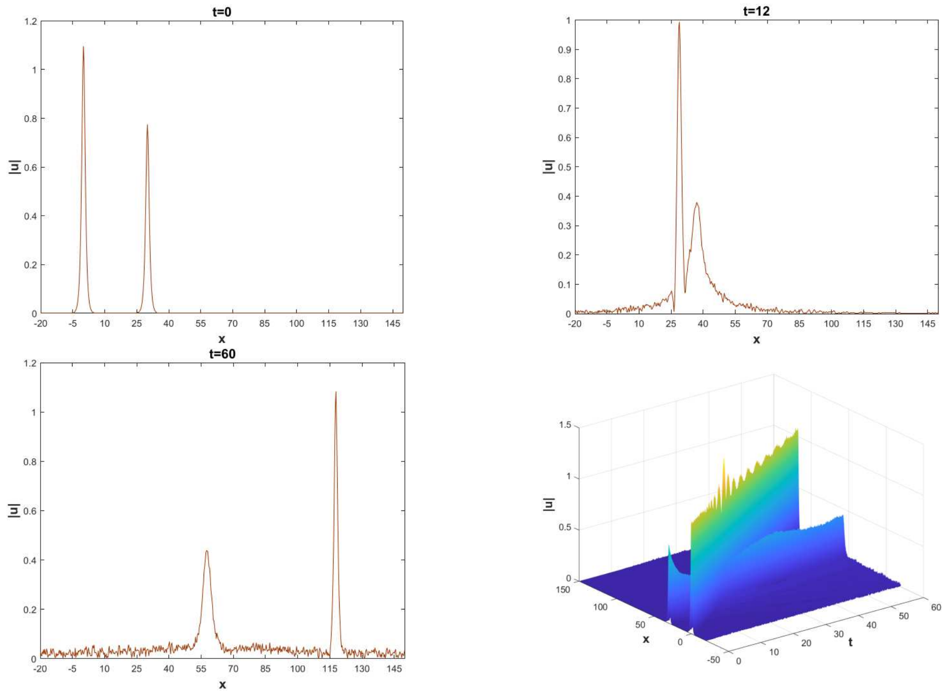

Fig. 3.8. The double soliton collision of (3.4) for the module u with the initial condition (3.5) along one trajectory. Inhomogeneous Dirichlet boundary conditions in $[-20, 150]$. $J = 20, M = 5, T = 60$, $\tau = 0.006$.

Example 3.3. In this example we consider the following two-dimensional stochastic NLS equation:

$$\mathrm{i}du = \big[u_{xx} + u_{yy} + |u|^2 u\big]dt + \varepsilon u \circ dW(t), \quad t \in (0, T]. \tag{3.6}$$

We choose the initial condition

$$u_0 = A\exp\big\{c_1x^2 + c_2y^2\big\}, \tag{3.7}$$

where A, c_1 and c_2 are constants. The solution is computed with the inhomogeneous Dirichlet boundary conditions in $[-10,10]\times[-10,10]$ with various sizes of the noise $\varepsilon = 1$, 5 and 10. The results are presented in Figs. 3.9-3.11.

Furthermore, we choose a larger spatial domain $[-60,60]\times[-60,60]$ to demonstrate the algorithm's superiority under the inhomogeneous Dirichlet boundary conditions. The results are presented in Figs. 3.12-3.14.

Example 3.4. Without loss of generality, in this numerical example we restrict our discussion to the two-dimensional stochastic NLS equation (3.6) with $\varepsilon = 1$ to show the computational efficiency of the ODDS algorithm under the homogeneous Dirichlet boundary conditions. We work in the same setting as in Example 3.3.

First, we apply the SMM method (3.2) and the FDSCN scheme (3.3) to the two-dimensional problem (3.6). Fig. 3.15 presents the computational cost of our ODDS algorithm in comparison with the SMM method and the FDSCN scheme. The reported CPU time is in seconds. From the figure, we can see that the ODDS algorithm can reduce the heavy computational load and is highly competitive.

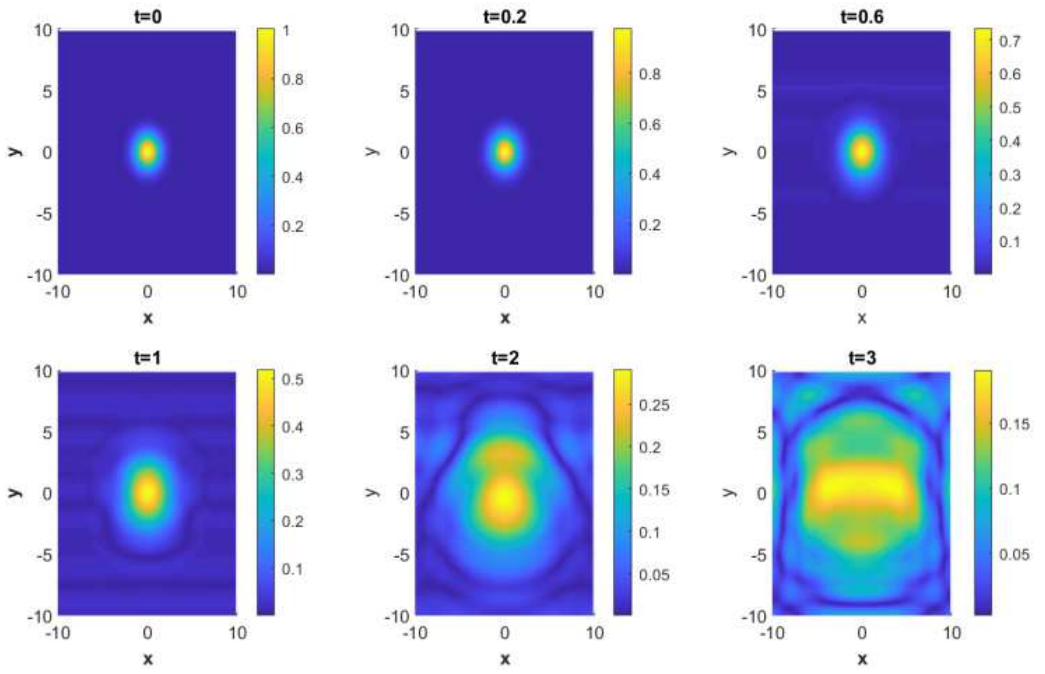

Fig. 3.9. The solution of (3.6) for the module u with the initial condition (3.7) along one trajectory. Inhomogeneous Dirichlet boundary conditions in $[-10,10]\times[-10,10]$. $A = 1, c_1 = c_2 = -1/2$, $J_1 = J_2 = 32, M_1 = M_2 = 4, T = 3, \tau = 0.01, \varepsilon = 1$.

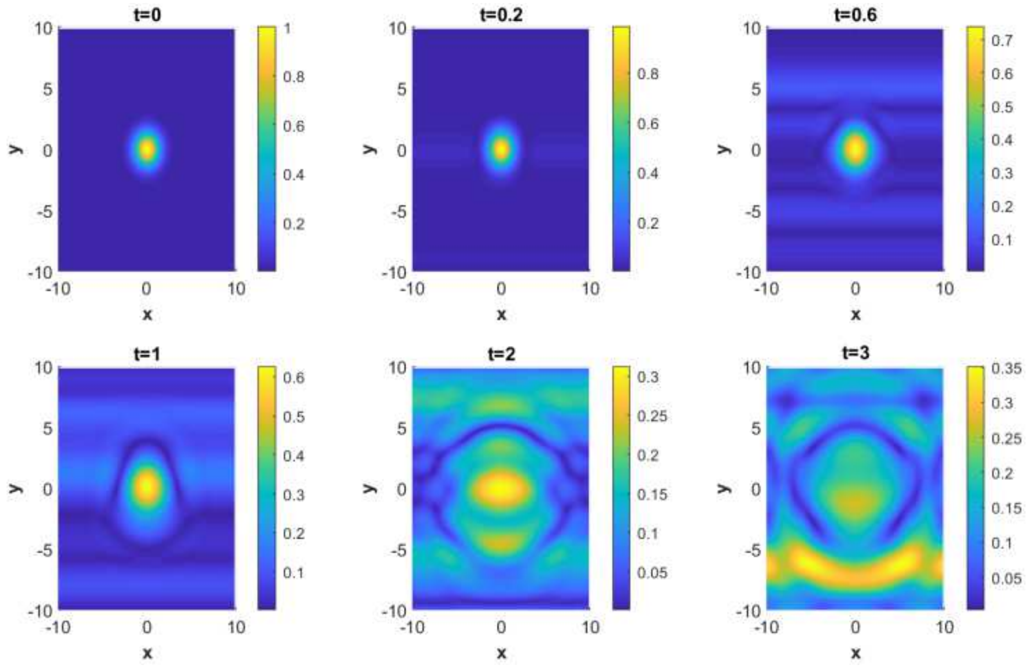

Fig. 3.10. The solution of (3.6) for the module u with the initial condition (3.7) along one trajectory. Inhomogeneous Dirichlet boundary conditions in $[-10,10]\times[-10,10]$. $A=1, c_1=c_2=-1/2$, $J_1=J_2=32, M_1=M_2=4, T=3, \tau=0.01, \varepsilon=5$.

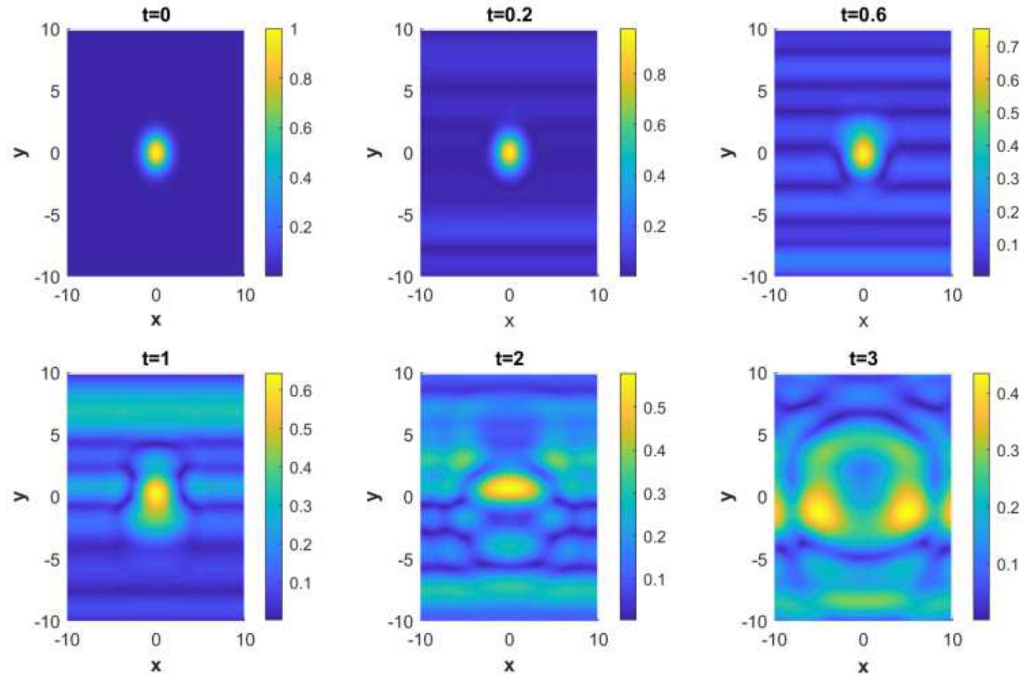

Fig. 3.11. The solution of (3.6) for the module u with the initial condition (3.7) along one trajectory. Inhomogeneous Dirichlet boundary conditions in $[-10,10]\times[-10,10]$. $A=1, c_1=c_2=-1/2$, $J_1=J_2=32, M_1=M_2=4, T=3, \tau=0.01, \varepsilon=10$.

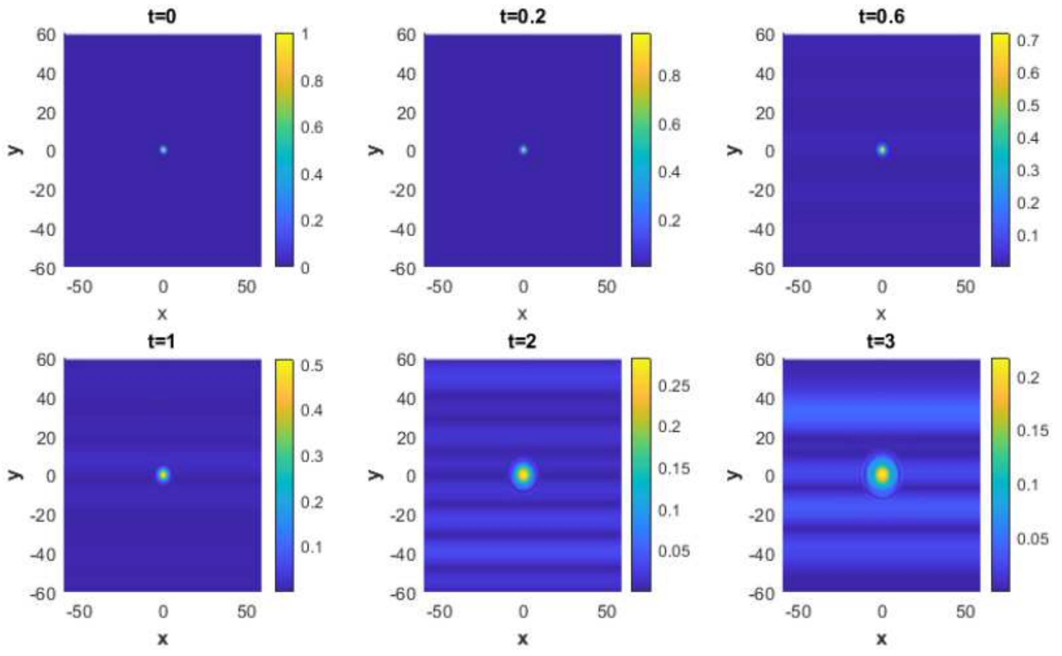

Fig. 3.12. The solution of (3.6) for the module u with the initial condition (3.7) along one trajectory. Inhomogeneous Dirichlet boundary conditions in $[-60,60]\times[-60,60]$. $A=1, c_1=c_2=-1/2$, $J_1=J_2=32, M_1=M_2=4, T=3, \tau=0.01$, $\varepsilon=1$.

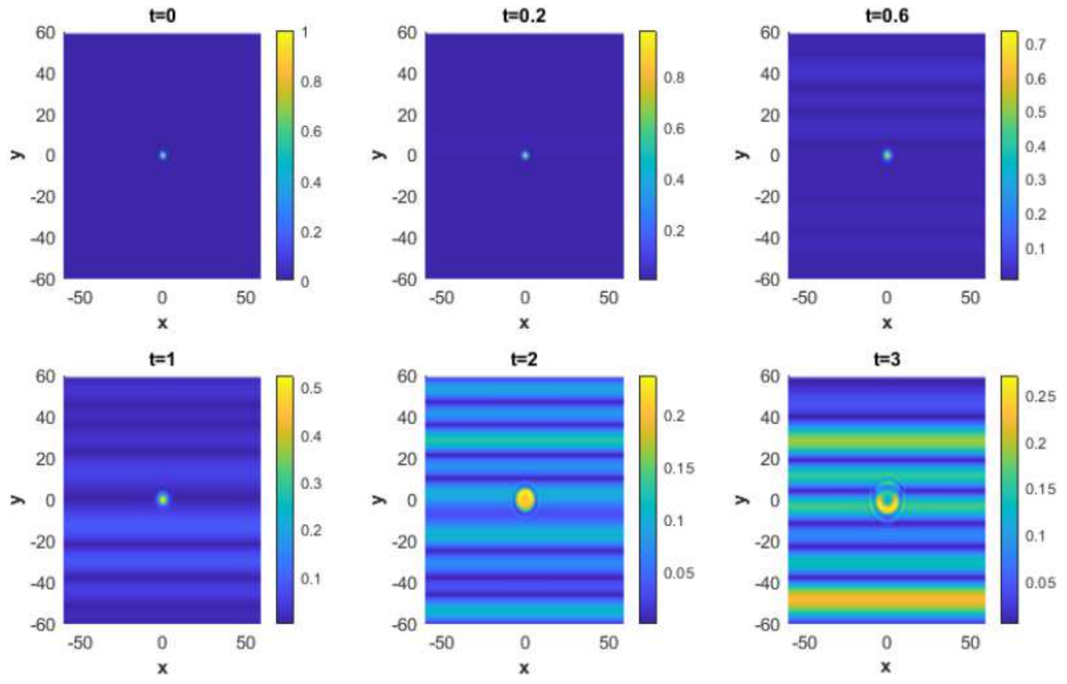

Fig. 3.13. The solution of (3.6) for the module u with the initial condition (3.7) along one trajectory. Inhomogeneous Dirichlet boundary conditions in $[-60,60]\times[-60,60]$. $A=1, c_1=c_2=-1/2$, $J_1=J_2=32, M_1=M_2=4, T=3, \tau=0.01, \varepsilon=5$.

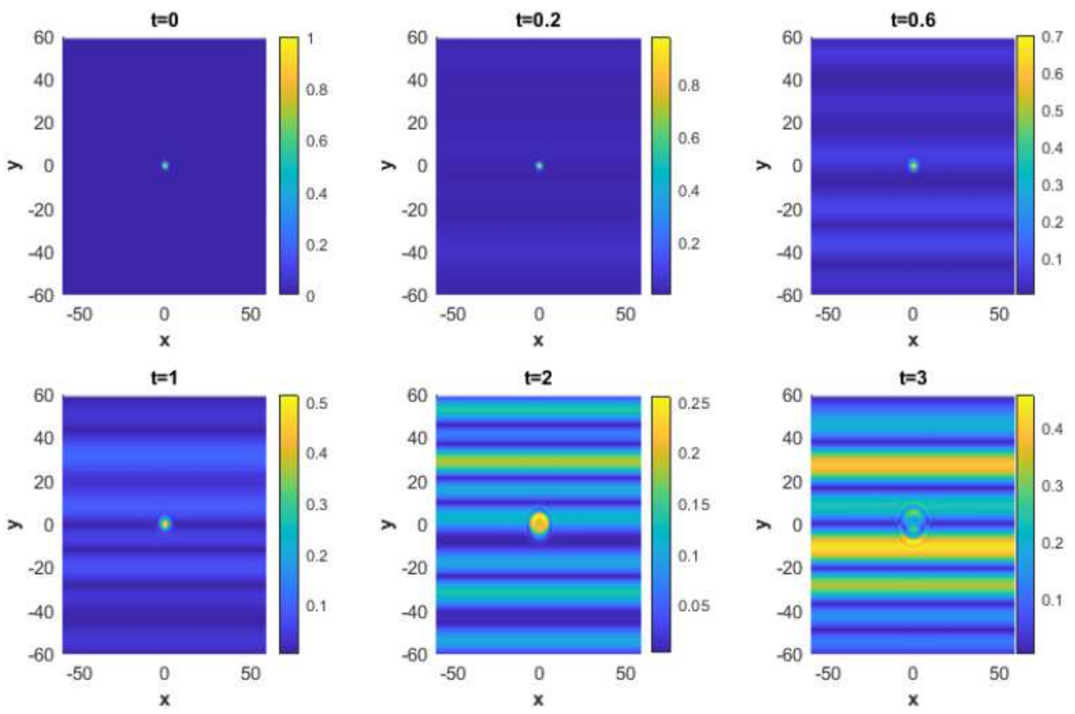

Fig. 3.14. The solution of (3.6) for the module u with the initial condition (3.7) along one trajectory. Inhomogeneous Dirichlet boundary conditions in $[-60, 60] \times [-60, 60]$. $A = 1, c_1 = c_2 = -1/2$, $J_1 = J_2 = 32, M_1 = M_2 = 4, T = 3, \tau = 0.01, \varepsilon = 10$.

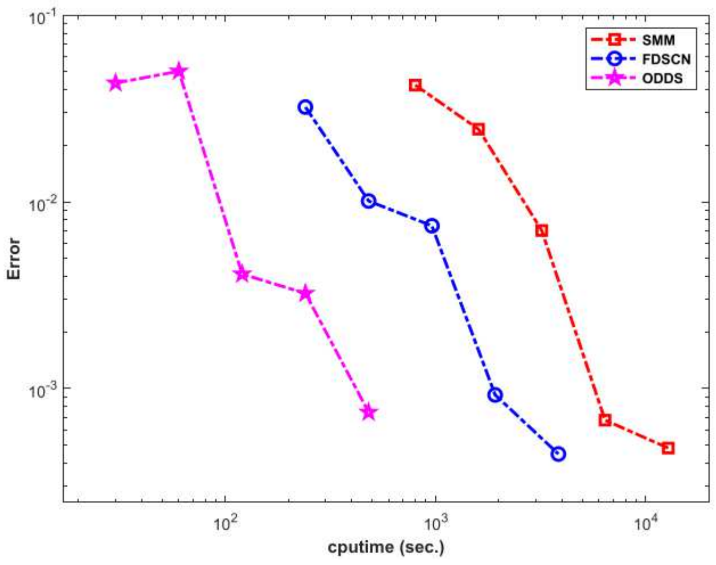

Fig. 3.15. Efficiency for the ODDS algorithm, the SMM method and the FDSCN scheme for (3.6) along one trajectory. Homogeneous Dirichlet boundary conditions in $[-10, 10]\times[-10, 10]$. A=1, c_1=c_2=$-1/2$, $J_1 = J_2 = 32, M_1 = M_2 = 4, T = 3, \tau = 0.01, \varepsilon = 1$. The mesh sizes of the SMM method and the FDSCN scheme are given by $h_x = h_y = 5/32$ (i.e. $J_1 = J_2 = 128, M_1 = M_2 = 1$).

Example 3.5. We conclude this section with the mean-square convergence order in the temporal direction of the proposed ODDS algorithm for the one-dimensional stochastic NLS equation (2.1) with the initial condition $u_0 = \sin(\pi x)$.

To compute the mean-square error, we run P independent trajectories $u^p(t,\cdot)$ and $u^{p,n}(\cdot)$

$$Err := \left(\mathbb{E}\Big[\big\|u(T,\cdot) - u^N(\cdot)\big\|_{l^2}^2\Big]\right)^{\frac{1}{2}} = \left(\frac{1}{P}\sum_{p=1}^{P}\big\|u^p(T,\cdot) - u^{p,N}(\cdot)\big\|_{l^2}^2\right)^{\frac{1}{2}}.$$

We take time $T = 1/4, [x_L, x_R] = [-1, 1]$ and $P = 500$. The reference solution is computed by the ODDS algorithm with small temporal step size $\tau = 2^{-10}$. The number of trajectories $P = 500$ is sufficiently large for the statistical errors not to significantly hinder the mean-square errors. The mean-square error is plotted in Table 3.1. The observed rates of convergence of the ODDS algorithm in time is close to $0.5\sim 1$. It is meaningful to give the mean-square convergence analysis theoretically in the future work.

Table 3.1: Mean-square errors of the ODDS algorithm for $\lambda = 1$ and $\varepsilon = 0.01$.

τ	Err	Order
2^{-4}	5.1163E-1	–
2^{-5}	2.6093E-1	0.97
2^{-6}	1.2133E-1	1.10
2^{-7}	7.0089E-2	0.79
2^{-8}	5.2949E-2	0.40
2^{-9}	2.7614E-2	0.93

4. Concluding Remarks

The calculation of stochastic NLS equation is an interesting and important problem. One of the classical techniques is by the operator splitting. In this work, we have developed a high efficient ODDS algorithm to solve the stochastic NLS equation with a multiplicative noise by combining the splitting technique. Several numerical examples are presented to illustrate the capability of the algorithm. Although not considered in this work, this algorithm is flexible for the coupled stochastic NLS equation, the stochastic wave equation and the stochastic Maxwell equations, and has excellent computational efficiency.

One difficult and challenging future work is the mean-square convergence analysis of the ODDS algorithm. On one hand, the difficulty in the error analysis of the proposed algorithm for the stochastic NLS equation in general is the lack of regularity of its solution. On the other hand, one general approach to analyze the convergence order of the overlapping and non-overlapping domain decomposition algorithms is the formal language theory, however it is not a generic property. We do not know whether the numerical algorithms of the stochastic partial differential equation possess the formal language property. This is an interesting and challenging topic for the future study.

Acknowledgments. The author is very grateful to the anonymous referees for their valuable comments and suggestions on the improvement of the presentation of this paper.

The research of Lihai Ji is partly supported by the National Natural Science Foundation of China (Grant Nos. 12171047, 11971458).

References

[1] O. Bang, P.L. Christiansen, F. If, K.Ø. Rasmussen, and Y.B. Gaididei, White noise in the two-dimensional nonlinear Schrödinger equation, *Appl. Anal.*, **57** (1995), 3–15.

[2] M. Barton-Smith, A. Debussche, and L. Di Menza, Numerical study of two-dimensional stochastic NLS equations, *Numer. Methods Partial Differ. Equ.*, **21** (2005), 810–842.

[3] A. de Bouard and A. Debussche, A stochastic nonlinear Schrödinger equation with multiplicative noise, *Comm. Math. Phys.*, **205** (1999), 161–181.

[4] A. de Bouard and A. Debussche, The stochastic nonlinear Schrödinger equations in H^1, *Stochastic Anal. Appl.*, **21** (2003), 97–126.

[5] A. de Bouard and A. Debussche, A semi-discrete scheme for the stochastic nonlinear Schrödinger equation, *Numer. Math.*, **96** (2004), 733–770.

[6] A. de Bouard and A. Debussche, Weak and strong order of convergence of a semidiscrete scheme for the stochastic nonlinear Schrödinger equation, *Appl. Math. Optim.*, **54** (2006), 369–399.

[7] J.P. Boyd, *Chebyshev and Fourier Spectral Methods*, Springer-Verlag, 1989.

[8] C. Chen and J. Hong, Symplectic Runge-Kutta semidiscretization for stochastic Schrödinger equation, *SIAM J. Numer. Anal.*, **54** (2016), 2569–2593.

[9] C. Chen, J. Hong, and L. Ji, Mean-square convergence of a symplectic local discontinuous Galerkin method applied to stochastic linear Schrödinger equation, *IMA J. Numer. Anal.*, **37** (2017), 1041–1065.

[10] C. Chen, J. Hong, and A. Prohl, Convergence of a θ-scheme to solve the stochastic nonlinear Schrödinger equation with Stratonovich noise, *Stoch. Partial Differ. Equ. Anal. Comput.*, **4** (2016), 274–318.

[11] C. Chen, J. Hong, D. Jin, and L. Sun, Large deviations principles for symplectic discretizations of stochastic linear Schrödinger equation, *Potential Anal.*, **59** (2023), 971–1011.

[12] J. Cui and J. Hong, Analysis of a splitting scheme for damped stochastic nonlinear Schrödinger equation with multiplicative noise, *SIAM J. Numer. Anal.*, **56** (2018), 2045–2069.

[13] J. Cui, J. Hong, Z. Liu, and W. Zhou, Stochastic symplectic and multi-symplectic methods for nonlinear Schrödinger equation with white noise dispersion, *J. Comput. Phys.*, **342** (2017), 267–285.

[14] J. Cui, J. Hong, Z. Liu, and W. Zhou, Strong convergence rate of splitting schemes for stochastic nonlinear Schrödinger equations, *J. Differ. Equ.*, **266** (2019), 5625–5663.

[15] J. Cui, S. Liu, and H. Zhou, Wasserstein Hamiltonian flow with common noise on graph, *SIAM J. Appl. Math.*, **83** (2023), 484–509.

[16] J. Cui, S. Liu, and H. Zhou, Optimal control for stochastic nonlinear Schrödinger equation on graph, *SIAM J. Control Optim.*, **61** (2023), 2021–2042.

[17] J. Cui, S. Liu, and H. Zhou, Stochastic Wasserstein Hamiltonian flows, *J. Dyn. Diff. Equat.*, (2023). `https://doi.org/10.1007/s10884-023-10264-4`

[18] J. Cui and L. Sun, Stochastic logarithmic Schrodinger equations: Energy regularized approach, *SIAM J. Math. Anal.*, **55** (2023), 3044–3080.

[19] A. Desaia, M. Khalilb, C. Pettitc, D. Poireld, and A. Sarkara, Scalable domain decomposition solvers for stochastic PDEs in high performance computing, *Comput. Methods Appl. Mech. Engrg.*, **335** (2018), 194–222.

[20] J. Hong and X. Wang, *Invariant Measures for Stochastic Nonlinear Schrödinger Equations: Numerical Approximations and Symplectic Structures*, in: *Lecture Notes in Mathematics*, Vol. 2251, Springer, 2019.

[21] J. Hong, X. Wang, and L. Zhang, Numerical analysis on ergodic limit of approximations for stochastic NLS equation via multi-symplectic scheme, *SIAM J. Numer. Anal.*, **55** (2017), 305–327.

[22] J. Hong, X. Wang, and L. Zhang, Parareal exponential θ-scheme for longtime simulation of stochastic Schrödinger equations with weak damping, *SIAM J. Sci. Comput.*, **41** (2019), B1155–B1177.

[23] S. Jiang, L. Wang, and J. Hong, Stochastic multi-symplectic integrator for stochastic nonlinear

Schrödinger equation, *Commun. Comput. Phys.*, **14** (2013), 393–411.

[24] G. Lin, A.M. Tartakovsky, and D.M. Tartakovsky, Uncertainty quantification via random domain decomposition and probabilistic collocation on sparse grids, *J. Comput. Phys.*, **229** (2010), 6995–7012.

[25] J. Liu, Order of convergence of splitting schemes for both deterministic and stochastic nonlinear Schrödinger equations, *SIAM J. Numer. Anal.*, **51** (2013), 1911–1932.

[26] J. Liu, A mass-preserving splitting scheme for the stochastic nonlinear Schrödinger equations with multiplicative noise, *IMA J. Numer. Anal.*, **33** (2013), 1469–1479.

[27] A. Sarkar, N. Benabbo, and R. Ghanem, Domain decomposition of stochastic PDEs: Theoretical formulations, *Internat. J. Numer. Methods Engrg.*, **77** (2009), 689–701.

[28] W. Subber and A. Sarkar, A domain decomposition method of stochastic PDEs: An iterative solution techniques using a two-level scalable preconditioner, *J. Comput. Phys.*, **257** (2014), 298–317.

[29] R. Tipireddy, P. Stinis, and A.M. Tartakovsky, Basis adaptation and domain decomposition for steady-state partial differential equations with random coefficients, *J. Comput. Phys.*, **351** (2017), 203–215.

[30] R. Tipireddy, P. Stinis, and A.M. Tartakovsky, Stochastic basis adaptation and spatial domain decomposition for partial differential equations with random coefficients, *SIAM/ASA J. Uncertain. Quantification*, **6** (2018), 273–301.

[31] B.D. Welfert, Generation of pseudospectral differentiation matrices I, *SIAM J. Numer. Anal.*, **34** (1997), 1640–1657.