Statistics for stochastic differential equations and local approximation of Crank-Nicolson method

Jun Ohkubo
Graduate School of Science and Engineering, Saitama University, 255 Shimo-Okubo, Sakura-ku, Saitama, 338-8570, Japan

Numerical evaluation of statistics plays an important role in data assimilation and filtering. When one focuses on stochastic differential equations, Monte Carlo simulations or moment closure approximations are available to evaluate the statistics. The other approach is to solve the corresponding backward Kolmogorov equation. However, a basis expansion for the backward Kolmogorov equation leads to infinite systems of differential equations, and conventional numerical methods such as the Crank-Nicolson method are not available directly. Here, a local approximation of the Crank-Nicolson method is proposed. The local approximation transforms the implicit time integration algorithm into an explicit one, which enables us to employ combinatorics for the proposed algorithm. The proposed algorithm shows a second-order convergence. Furthermore, the convergence property naturally leads to extrapolation methods; they work well to calculate a more accurate value with small steps. The proposed method is demonstrated with the Ornstein-Uhlenbeck process and the noisy van der Pol system.

I. INTRODUCTION

Numerical time integration is a common topic in various research fields of computer simulations. If we consider systems with noise, we usually focus on the statistics such as averages and variances, and so on. These statistics play an important role in data assimilation and filtering; for example, see [1]. To evaluate the statistics numerically, we usually employ Monte Carlo simulations to deal with the noisy characteristics. For example, the Euler-Maruyama approximation is famous in numerical simulations of stochastic differential equations [2]. The stochastic differential equations have many applications ranging from time-series data analysis to control problems. Furthermore, rapid evaluation of statistics is necessary for real-time processing. However, the generations of random numbers and samplings for accuracy need high computational costs.

A stochastic differential equation has a corresponding partial differential equation, i.e., a Fokker-Planck equation [3, 4]. In addition, as explained later, a different type of partial differential equation, the backward Kolmogorov equation, is also available to discuss the statistics of stochastic differential equations. Basis expansions for the Fokker-Planck equation and the backward Kolmogorov equation lead to coupled ordinary differential equations. However, the number of the derived coupled ordinary differential equations is infinite in general. Hence, we cannot directly employ conventional time integration algorithms like the Runge-Kutta method. Although truncation of the basis functions enables us to obtain approximated values, this approach is still time-consuming for high-dimensional cases. For some types of stochastic differential equations, discussions for moment-closure methods have been performed; there are many references for this topic, ranging from stock price models to population models [6]; see the review in [7]. However, this moment closure approach also needs a kind of truncation.

We have several famous numerical approaches to partial differential equations, such as spatial discretization, Galerkin methods, and collocation methods [9]. These approaches lead to coupled ordinary differential equations to approximate the original partial differential equations. Then, numerical integration methods, such as the explicit Euler method, Crank-Nicolson method, and the Runge-Kutta method, are employed. Some of them are implicit time integration algorithms, in which we need to calculate matrix inverses; the computation takes high computational costs. The authors in [10] discussed an approximation of the Crank-Nicolson method, and they proposed an explicit algorithm called the local Crank-Nicolson method. There are some demonstrations of the local Crank-Nicolson method in a simple diffusion equation [10] and Burgers’ equation [11]; we will see a brief review of the local Crank-Nicolson method later.

The focus here is the evaluation of statistics for stochastic differential equations without generating random numbers. Hence, we employ the approach based on the backward Kolmogorov equation. Since the aim is not to evaluate the complete solution of the partial differential equation, it is enough to obtain several important values for the evaluation of statistics; if one focuses on the statistics for a specific initial condition, only a part of the complete solution is enough. This approach has been studied, for example, in [12], and the usage of resolvents and combinatorics gives an efficient algorithm to evaluate the statistics for short-time ranges. Such algorithms based on combinatorics are powerful, and some recent works in statistical physics focus on them. For example, discussions based on combinatorics have been employed to compute the Mori-Zwanzig memory integral in generalizations of Langevin equations [13, 14]. However, as pointed out in [12], one of the problems of the algorithm based

* johkubo@mail.saitama-u.ac.jp
on combinatorics is the slow convergence speed for the iterative procedure. Is it possible to achieve more rapid convergence?

In the present paper, a local approximation of the Crank-Nicolson method is proposed. As stated above, the Crank-Nicolson method is an implicit time integration algorithm, which needs an inversion of the coefficient matrix. The proposition here employs the matrix inversion up to the second-order for a short-time interval, which converts the implicit algorithm into an explicit one. The explicit nature enables us to achieve the time-evolution based on combinatorics. The order of the convergence of the proposed algorithm is second. Hence, we can evaluate the statistics more rapidly and efficiently. As demonstrations, we apply the proposed algorithm to the Ornstein-Uhlenbeck process and the noisy van der Pol systems.

The construction of the present paper is as follows. Section II gives brief reviews of the backward Kolmogorov equation, algorithms based on combinatorics, resolvents, and time-splitting methods for time-integration. Section III is the main contribution of the present paper; a second-order approximation for the matrix inversion is explicitly given. Furthermore, the second-order convergence property leads to an extrapolation method to evaluate statistics with small iterative steps. The proposed method is demonstrated in Sec. IV, in which the famous Ornstein-Uhlenbeck process and a noisy van der Pol system are used. Section V gives some concluding remarks.

II. PREVIOUS WORKS

A. Backward Kolmogorov equation and statistics

Firstly, we briefly review the basics of stochastic differential equations and the corresponding forward and backward Kolmogorov equations. For details, see, for example, [3] and [4].

Consider a $D$-dimensional state space $\mathcal{M} \subseteq \mathbb{R}^D$ and a vector of stochastic variables $\mathbf{X} \in \mathcal{M}$. The time-evolution of the state vector $\mathbf{X}$ is continuous, and it obeys the following stochastic differential equation:

$$d\mathbf{X} = \mathbf{a}(\mathbf{X}, t)dt + \mathbf{B}(\mathbf{X}, t)d\mathbf{W}(t),$$  \hspace{1cm} \text{(1)}$$

where $\mathbf{a}(\mathbf{X}, t)$ is a vector of drift coefficient functions, and $\mathbf{B}(\mathbf{X}, t)$ is a matrix of diffusion coefficient functions. The components of the vector of Wiener processes $\mathbf{W}(t)$ satisfy

$$dW_i(t)dW_j(t) = \delta_{ij}dt.$$  \hspace{1cm} \text{(2)}$$

The stochastic differential equation in (1) has a corresponding Fokker-Planck equation

$$\frac{\partial}{\partial t} p(\mathbf{x}, t) = \mathcal{L}_t p(\mathbf{x}, t),$$  \hspace{1cm} \text{(3)}$$

where

$$\mathcal{L}_t = -\sum_i \frac{\partial}{\partial x_i} a_i(\mathbf{x}, t) + \frac{1}{2} \sum_{i,j} \frac{\partial^2}{\partial x_i \partial x_j} [\mathbf{B}(\mathbf{x}, t)\mathbf{B}(\mathbf{x}, t)^T]_{ij}$$  \hspace{1cm} \text{(4)}$$

is the time-evolution operator for the Fokker-Planck equation. The initial condition at the initial time $t_{ini}$ is

$$p(\mathbf{x}, t_{ini}) = \delta(\mathbf{x} - \mathbf{x}_{ini}),$$  \hspace{1cm} \text{(5)}$$

where $\mathbf{x}_{ini}$ is the initial position for the stochastic differential equation, and $\delta(\cdot)$ is the Dirac delta function.

By solving the Fokker-Planck equation, we obtain a probability density function or a transition probability for $\mathbf{X}$ at time $t$. If we want to focus on the statistics or functions for the stochastic differential equations, the corresponding backward Kolmogorov equation is useful. For example, provided the initial condition is $\mathbf{x}_{ini}$, the $\alpha$-th order moment at time $t_{fin}$ is evaluated as

$$\mathbb{E}[\mathbf{x}^\alpha | \mathbf{x}_{ini}] = \int_{\mathcal{M}} \mathbf{x}^\alpha p(\mathbf{x}, t_{fin})d\mathbf{x}.$$  \hspace{1cm} \text{(6)}$$

where $\alpha_d \in \mathbb{N}_0$ for $d = 1, \ldots, D$ and $\mathbf{x}^\alpha = x_1^{\alpha_1} \cdots x_D^{\alpha_D}$. Here, introducing a short-time interval $\Delta t$ and repeating the partial integration, we have

$$\int_{\mathcal{M}} \mathbf{x}^\alpha p(\mathbf{x}, t_{fin})d\mathbf{x}$$

$$= \int_{\mathcal{M}} \mathbf{x}^\alpha \left( \exp \left( \int_{t_{ini}}^{t_{fin}} dt \mathcal{L}_t \right) \right) \delta(\mathbf{x} - \mathbf{x}_{ini})d\mathbf{x}$$

$$\simeq \int_{\mathcal{M}} \mathbf{x}^\alpha \left( e^{\mathcal{L}_{t_{fin}} \Delta t} e^{\mathcal{L}_{t_{fin} - \Delta t} \Delta t} \cdots e^{\mathcal{L}_{t_{ini}} \Delta t} \delta(\mathbf{x} - \mathbf{x}_{ini}) \right) d\mathbf{x}$$

$$= \int_{\mathcal{M}} \left( e^{\mathcal{L}_{t_{fin}} \Delta t} \cdots e^{\mathcal{L}_{t_{fin} - \Delta t} \Delta t} e^{\mathcal{L}_{t_{ini}} \Delta t} \mathbf{x}^\alpha \right) \delta(\mathbf{x} - \mathbf{x}_{ini})d\mathbf{x}$$

$$\simeq \int_{\mathcal{M}} \left( \exp \left( \int_{t_{fin}}^{t_{ini}} dt \mathcal{L}_t^{\alpha} \right) \mathbf{x}^\alpha \right) \delta(\mathbf{x} - \mathbf{x}_{ini})d\mathbf{x}$$

$$= \exp \left( \int_{t_{fin}}^{t_{ini}} dt \mathcal{L}_t^{\alpha} \right) \mathbf{x}_{ini}^{\alpha},$$  \hspace{1cm} \text{(7)}$$

where

$$\mathcal{L}_t^{\alpha} = \sum_i \alpha_i(\mathbf{x}, t) \frac{\partial}{\partial x_i} + \frac{1}{2} \sum_{i,j} \left[ \mathbf{B}(\mathbf{x}, t)\mathbf{B}(\mathbf{x}, t)^T \right]_{ij} \frac{\partial^2}{\partial x_i \partial x_j}.$$  \hspace{1cm} \text{(8)}$$

Hence, we obtain the following backward Kolmogorov equation:

$$\frac{d}{dt} \varphi^\alpha(\mathbf{x}, t) = \mathcal{L}_t^{\dagger} \varphi^\alpha(\mathbf{x}, t),$$  \hspace{1cm} \text{(9)}$$

which should be solved backwardly in time, i.e., $t_{fin} \rightarrow t_{ini}$, and the ‘initial’ condition is given as

$$\varphi^\alpha(\mathbf{x}, t_{fin}) = \mathbf{x}^\alpha.$$  \hspace{1cm} \text{(10)}$$
We obtain \( \varphi_\alpha(x, t_{\text{ini}}) \) as a solution of the backward Kolmogorov equation, and a specific value \( \varphi_\alpha(x_{\text{ini}}, t_{\text{ini}}) \) gives the expectation in (6).

For simplicity, we consider only time-independent drift and diffusion coefficient functions in the following discussions. When the coefficient functions are time-dependent, we can treat them as new state variables. By using the famous Itô formula, it is also possible to consider other types of statistics; we can see examples of such extensions in [19].

**B. Combinatorics**

In the following discussions, we only consider a stochastic differential equation with time-independent drift and coefficient functions, as denoted above. Hence, the time-evolution operator in (8) is also time-independent. Assume that \( t_{\text{ini}} = 0 \) and \( t_{\text{fin}} = T \), and we reinterpret the direction of the time-evolution for the backward Kolmogorov equation as \( 0 \rightarrow T \).

In order to solve the backward Kolmogorov equation, it is useful to consider the following expansion for \( \varphi_\alpha(x, t) \):

\[
\varphi_\alpha(x, t) = \sum_n P_\alpha(n, t)(x - x_c)^n,
\]

where \( n_d \in \mathbb{N}_0 \) for \( d = 1, \ldots, D \) and \( n = (n_1, \ldots, n_D) \). \( \{P_\alpha(n, t)\} \) are the expansion coefficients, and \( x_c \) is the center of the expansion. By using the basis expansion in (11), we have coupled ordinary differential equations for \( \{P_\alpha(n, t)\} \) instead of the partial differential equation in (9). However, note that the time-evolution operator in (8) leads to infinite systems for \( \{P_\alpha(n, t)\} \), which is intractable numerically.

Here, we focus on the following fact: Provided \( x_c = x_{\text{ini}} \), only a term with \( n = 0 \) in (11) remains for \( \varphi_\alpha(x_{\text{ini}}, t) \). Hence, if one wants to calculate the statistics only for the specific initial condition \( x_{\text{ini}} \), the expansion in (11) is useful because we can employ algorithms based on combinatorics for numerical evaluation as below. For example, if the time-independent drift coefficient function \( a_i(x) \) in (8) contains a term \( x_d^2 \partial x_d \), it is rewritten as

\[
x_d^2 \frac{\partial}{\partial x_d} = (x_d - x_{\text{ini},d})^2 \frac{\partial}{\partial x_d} + 2(x_d - x_{\text{ini},d})x_{\text{ini},d} \frac{\partial}{\partial x_d}
\]

\[
+ x_{\text{ini},d}^2 \frac{\partial}{\partial x_d}.
\]

The action of the first term in r.h.s of (12) to the basis in (11) gives

\[
(x_d - x_{\text{ini},d})^2 \frac{\partial}{\partial x_d} (x - x_{\text{ini}})^n = n_d (x - x_{\text{ini}})^{n'},
\]

where

\[
n' = (n_1, \ldots, n_d + 1, \ldots, n_D).
\]

Then, there is a change in the ‘discrete state’ with \( n \rightarrow n' \). From the similar discussion, it is clear that the second term in r.h.s of (12) does not change the discrete state \( n \), and the third one decreases \( n_d \) by one. This means that we have a discrete process for \( n \) instead of the partial differential equation with continuous variables in (9). Note that it is possible to extend this discussion to the duality in stochastic processes [16–19], and numerical algorithms based on combinatorics have been developed in recent studies [12, 20]. Here, there is no need to employ the discussion based on the duality, and we omit the details.

A simple way to understand the algorithms based on combinatorics is to consider the Taylor expansion of the exponential of the time-evolution operator:

\[
e^{\mathcal{L}^\dagger T} = \sum_{m=0}^{\infty} \frac{T^m}{m!} (\mathcal{L}^\dagger)^m.
\]

Here, the action of \( \mathcal{L}^\dagger \) to the basis expansion in (11) is interpreted as the transition on the discrete state \( n \). Hence, it is useful to employ the matrix expression for (8); the matrix element for \( n \rightarrow n' \) is written as

\[
[\mathcal{L}^\dagger]_{n'n} = \sum_{r=1}^{R} \gamma_r(n) \delta_{v_r,n'-n}.
\]

where \( R \) is the number of terms (events) in \( \mathcal{L}^\dagger \), \( \gamma_r(n) \) is a state-dependent coefficient for the \( r \)-th event, and \( v_r \) is a vector for the state change in \( n \) for the \( r \)-th event. The matrix expression is easily obtained from the time-evolution operator in (8); the concrete examples in Sec. IV will help to understand the expression. By using the expression, the coefficient of \( P(n = 0, T) \) in (11) is numerically computed from a sum of weights of paths from \( n_{\text{ini}} = \alpha \) to \( n = 0 \). It is possible to make the list of the paths from the discussion on combinatorics. Of course, the size of the list exponentially increases with the length of the paths. Hence, an algorithm based on dynamic programming should be employed.

**C. Resolvent**

The naive Taylor-type expansion is numerically unusable if \( \mathcal{L}^\dagger \) is an unbounded operator in a Banach space [8]. Alternatively, the following formula is suitable for the numerical calculation [12]:

\[
e^{\mathcal{L}^\dagger T} = \lim_{M \to \infty} \left( 1 - \frac{T}{M} \mathcal{L}^\dagger \right)^{-1} M,
\]

where \( (1 - \frac{T}{M} \mathcal{L}^\dagger)^{-1} \) is a resolvent of \( \mathcal{L}^\dagger \), apart from a constant factor [8]. In [12, 20], the following approxima-
tion of the resolvent was introduced:

\[
\left[ 1 - \frac{T}{M} \mathcal{L} \right]^{-1}_{n' n} = \begin{cases} 
\frac{1}{1 - \frac{T}{M} [\mathcal{L}]_{nn}} & \text{for } n' = n, \\
\frac{T}{M} \frac{[\mathcal{L}]_{n'n}}{(1 - \frac{T}{M} [\mathcal{L}]_{n'n}) (1 - \frac{T}{M} [\mathcal{L}]_{nn})} & \text{otherwise},
\end{cases}
\]

(18)

With the usage of the above approximation and algorithms based on combinatorics, we obtain reasonable numerical evaluations of statistics for stochastic differential equations. As denoted above, an algorithm based on dynamic programming is available to reduce the computational cost \cite{12}. Note that it is enough to evaluate only some finite cases for \cite{17}. That is, some acceleration and extrapolation methods are available, and hence a small \( M \) is enough. In \cite{20}, the estimation of the converged values of the target statistics was obtained for reasonable computational times if \( T \) is not so long.

D. Time-splitting methods

It is possible to reinterpret the usage of the resolvent in Sec. II C as a variant of implicit Euler methods \cite{9}. Here, assume that we fix \( M \) in \cite{17} to a large value. Then, we can see \( T/M \) as a short-time interval, which leads to a conventional time-evolution algorithm. Note that \cite{15} naturally converts the implicit nature of the algorithm to an explicit one. A similar approach has already been proposed in \cite{10} \cite{11}, i.e., the local Crank-Nicolson method. In \cite{10}, the authors applied the local Crank-Nicolson method to a simple diffusion process; an example of the local Crank-Nicolson method is briefly reviewed in Appendix A. Here, we will see only some important points of the local Crank-Nicolson method.

Note that the local Crank-Nicolson method is applicable only for a finite matrix. Hence, the discussion here is not available for the backward Kolmogorov equations. The following brief review of the local Crank-Nicolson method is given only as a hint to convert an implicit time-evolution method to the corresponding explicit one.

Assume that the coefficient matrix \( A \) for the time-evolution is split into \( S \) terms as follows:

\[
A = \sum_{i=1}^{S} A_i,
\]

(19)

where \( A_i \) is assumed to have a suitable sparsity. An example of a simple diffusion process is given in Appendix A in which \( A_i \) has a block structure. This characteristic is important for the local Crank-Nicolson method in \cite{10}.

The Trotter product gives

\[
\exp(TA) = \lim_{M \to \infty} \left\{ \prod_{i=1}^{S} \exp \left( T M \frac{A_i}{M} \right) \right\}^M,
\]

(20)

where \( M \) are a positive integer. The conventional Crank-Nicolson method employs the following approximation for the time-evolution operator:

\[
\exp \left( T M A_i \right) \approx \exp \left( T M \frac{A_i}{2M} \right) \exp \left( T M \frac{A_i}{2M} \right)
\]

\[
\simeq \left( I - \frac{T}{2M} A_i \right)^{-1} \left( I + \frac{T}{2M} A_i \right)
\]

(21)

In addition, the following time-evolution operator was introduced in \cite{10} to improve the numerical accuracy:

\[
\exp(TA) \simeq \frac{1}{2} \left( V_1 + V_2 \right),
\]

(22)

where

\[
V_1 = \left\{ \prod_{i=1}^{S} \left( I - \frac{T}{2M} A_i \right)^{-1} \left( I + \frac{T}{2M} A_i \right) \right\}^M,
\]

(23)

\[
V_2 = \left\{ \prod_{i=1}^{S} \left( I - \frac{T}{2M} A_{S-i} \right)^{-1} \left( I + \frac{T}{2M} A_{S-i} \right) \right\}^M.
\]

(24)

The operator \( V_2 \) corresponds to the reverse order action of splitted operators \( A_i \). In \cite{10}, the authors discussed the property of the second-order approximation in time. Furthermore, the authors rewrote the inverse matrix in a simple form. For example, consider the following case:

\[
\left[ I - \frac{T}{2M} A_i \right]^{-1} = \left[ \frac{I}{2M} R_i \right]_{I_{N-i-2}},
\]

(25)

where \( A_i \) is a \((N-1) \times (N-1)\) matrix, and \( R_i \) is a \(2 \times 2\) matrix; an explicit example is shown in Appendix A. Here, the inverse is easily calculated as

\[
\left[ I - \frac{T}{2M} A_i \right]^{-1} = \left[ \frac{I}{2M} R_i^{-1} \right]_{I_{N-i-2}}
\]

(26)

because it is easy to obtain the inverse of \(2 \times 2\) matrix \( R_i \).

The local Crank-Nicolson method in \cite{10} has the explicit property, and the stability was also discussed. However, it is easy to see that the local Crank-Nicolson method cannot be applied for infinite cases derived from the backward Kolmogorov equation. In addition, we cannot use the simple form in \cite{20} in general.

In the next section, another type of local approximation of the Crank-Nicolson method is proposed, which is available even for the infinite cases.
III. PROPOSAL OF EXPLICIT METHODS WITH COMBINATORICS

A. A local approximation of the Crank-Nicolson method

As reviewed in Secs. IIA IIB and IIC the backward Kolmogorov equation gives infinite systems of coupled ordinary differential equations, and the basis expansion in (11) enables us to use the algorithm based on combinatorics with the aid of the resolvent. Note that the inversion in (18) was the key to obtaining the algorithm based on combinatorics.

Again, note that the simple usage of the resolvent in (18) corresponds to the implicit Euler method. Hence, it is natural to use the Crank-Nicolson method to evaluate \( \exp(L^tT) \). However, the adjoint time-evolution operator \( L^\dagger \) does not have a suitable sparse property like \( L \). Hence, we cannot calculate the inversion as in (26).

The aim here is to obtain an explicit form of the resolvent:

\[
C = \left( I - \frac{T}{2M}L^\dagger \right)^{-1},
\]

which is necessary to apply the Crank-Nicolson method for the time-evolution of the backward Kolmogorov equation. However, we cannot explicitly write down the inversion because of the infinite nature of \( L^\dagger \).

Here, note that the Crank-Nicolson method has the second-order convergence in time. The fact suggests that only the second-order approximation is enough for the inverse part. This simple idea, i.e., the local approximation of the inverse up to second-order, is the key to the proposed algorithm. Then, we should write down the explicit form of the approximated inverse: Appendix B gives a detailed discussion of the derivation for finite-dimensional matrix cases. Although the mathematically rigorous proof is difficult for unbounded operators such as \( L^\dagger \), the discussion in Appendix B gives us the following approximated form of the inversion of \( (27) \):

\[
[C]_{nn} \simeq \left( 1 - h \left[ L^\dagger \right]_{nn} - h^2 \sum_{\bar{n} \neq n} \left[ L^\dagger \right]_{n\bar{n}} \left[ L^\dagger \right]_{\bar{n}n} \right)^{-1}
\]

and

\[
\begin{aligned}
[C]_{n'n} &\simeq \frac{h \left[ L^\dagger \right]_{n'n}}{\left( 1 - h \left[ L^\dagger \right]_{nn} - h^2 \sum_{\bar{n} \neq n} \left[ L^\dagger \right]_{n\bar{n}} \left[ L^\dagger \right]_{\bar{n}n} \right)} \\
&+ \sum_{\bar{n} \neq n, n'} \frac{h^2 \left[ L^\dagger \right]_{n'\bar{n}} \left[ L^\dagger \right]_{\bar{n}n}}{(1 - h \left[ L^\dagger \right]_{nn})(1 - h \left[ L^\dagger \right]_{n\bar{n})(1 - h \left[ L^\dagger \right]_{mn})},
\end{aligned}
\]

where \( h = T/(2M) \).

Equations in (28) and (29) give approximations up to the second-order in \( h \), which is enough to obtain the second-order convergence for the Crank-Nicolson method. In addition, the backward Kolmogorov equation leads to a kind of sparse structure for \( L^\dagger \) for the expansion in (11), as shown later. Hence, the computational cost is not large, and algorithms based on combinatorics are available.

B. Summary of time-evolution methods with combinatorics

Again, we here note the point of the proposal of the present paper; the focus is the numerical evaluation of a statistical value for the stochastic differential equation for an initial condition. Hence, there is no need to obtain the entire solution of the backward Kolmogorov equation; only an expansion coefficient \( P(n=0,T) \) in (11) is enough for the numerical evaluation. The fact enables us to use the algorithm based on combinatorics and the inversion formula in (28) and (29). This proposal works if \( T \) is not so long, and recent studies for data assimilation and machine learning require efficient numerical evaluation for such short-time cases as introduced in Sec. I.

The main proposal of the present paper is the local approximation of the Crank-Nicolson method. However, it is possible to consider different types of time-evolution methods. We here summarize some of them.

- Taylor-type: As explained in Sec. IIB the Taylor expansion in (15) could be available, but the con-
Local approximation of the Crank-Nicolson method

• Resolvent method (implicit Euler method): The previous work based on the resolvent corresponds to the implicit Euler method, as explained above. In the resolvent method, we consider the following approximation of the time-evolution operator:

\[
\exp\left(\frac{T}{M} \mathcal{L}^t\right) \simeq \left(1 - \frac{T}{M} \mathcal{L}^t\right)^{-1},
\]

(30)

where \(M\) is assumed to be large enough. The iterative application gives the time-evolution from time 0 to \(T\). The limit of \(M \to \infty\) corresponds to the implicit Euler method. The matrix element of the inverse is given as (18). It is famous that the implicit Euler method shows the first-order convergence in time.

• Explicit Euler method: Instead of the expansion in (30), the following expansion is also available:

\[
\exp\left(\frac{T}{M} \mathcal{L}^t\right) \simeq 1 + \frac{T}{M} \mathcal{L}^t.
\]

(31)

This method does not need the approximation for the inverse; we can directly use the matrix element of \(\mathcal{L}^t\). The convergence rate in time is first-order.

• Heun method: An explicit higher-order method is available; the Heun method is given as

\[
\exp\left(\frac{T}{M} \mathcal{L}^t\right) \simeq 1 + \frac{T}{M} \mathcal{L}^t + \frac{1}{2} \left(\frac{T}{M} \mathcal{L}^t\right)^2.
\]

(32)

Note that we must perform the iterative application of (32) in order to obtain the time-evolution for time \(T\). The resolvent method and the explicit Euler method need ‘one process’ of \(\mathcal{L}^t\) for the time-evolution for \(T/M\). By contrast, the Heun method takes at most ‘two processes’ for \(T/M\). Hence, the computational cost for the Heun method for the time-evolution of \(T/M\) is larger than the above two methods. The Heun method shows second-order convergence in time.

• Local approximation of the Crank-Nicolson method (the proposed one): The Crank-Nicolson method is derived from

\[
\exp\left(\frac{T}{M} \mathcal{L}^t\right) = \exp\left(\frac{T}{2M} \mathcal{L}^t\right) \exp\left(\frac{T}{2M} \mathcal{L}^t\right)
\]

\[
\simeq \left(1 - \frac{T}{2M} \mathcal{L}^t\right)^{-1} \left(1 + \frac{T}{2M} \mathcal{L}^t\right),
\]

(33)

and the explicit matrix elements of the resolvent (inverse) are given as (28) and (29). The one step of \(T/M\) in (33) needs at most ‘three processes’ of \(\mathcal{L}^t\). This method has second-order convergence in time.

C. Extrapolation with second-order

Since we connect the usage of the resolvent with the time-splitting method, it is possible to propose extrapolation methods based on the convergence properties, as follows.

For the first-order convergence, we obtain

\[
|\hat{m}^{(1)} - \hat{m}| = C \left(\frac{T}{M^{(1)}}\right)^2,
\]

(34)

\[
|\hat{m}^{(2)} - \hat{m}| = C \left(\frac{T}{M^{(2)}}\right)^2.
\]

(35)

where \(\hat{m}^{(i)}\) is the evaluated value of the target statistic for the case with \(M^{(i)} (i = 1, 2)\). We assume that the true value is given by \(\tilde{m}\), and \(C\) is a constant.

If \(\hat{m}^{(1)} > \tilde{m}\) and \(\hat{m}^{(2)} > \tilde{m}\), the following estimation is derived:

\[
C = \frac{M^{(1)} M^{(2)}}{T (M^{(2)} - M^{(1)})},
\]

(36)

\[
\tilde{m} = \hat{m}^{(2)} - C \frac{T}{M^{(2)}}.
\]

(37)

For \(\hat{m}^{(1)} < \tilde{m}\) and \(\hat{m}^{(2)} < \tilde{m}\), the same estimation is derived. Preliminary numerical experiments show that the behavior of \(\hat{m}^{(i)}\) is similar for enough small \(T/M^{(i)}\). Hence, \(\tilde{m}\) is available.

For the case with the second-order convergence, we have

\[
|\hat{m}^{(1)} - \hat{m}| = C \left(\frac{T}{M^{(1)}}\right)^2,
\]

(38)

\[
|\hat{m}^{(2)} - \hat{m}| = C \left(\frac{T}{M^{(2)}}\right)^2.
\]

(39)

Hence, the following extrapolation method is available as similar to the first-order case:

\[
C = \frac{(M^{(1)} M^{(2)})^2}{T^2 ((M^{(2)})^2 - (M^{(1)})^2)},
\]

(40)

\[
\tilde{m} = \hat{m}^{(2)} - C \left(\frac{T}{M^{(2)}}\right)^2.
\]

(41)

IV. NUMERICAL EXPERIMENTS

A. Two examples

In order to check the convergence property of the proposed method, we use two examples.

The first example is the famous Ornstein-Uhlenbeck process:

\[
dx = -\gamma x dt + \sigma dW(t),
\]

(42)
In the numerical experiments below, we use the Ornstein-Uhlenbeck process [3]:

\[ \text{Uhlenbeck process} \quad (\sigma^2/2)\partial_x^2 x = \gamma x \]

where \( \gamma > 0 \) and \( \sigma > 0 \). The adjoint time-evolution operator \( \mathcal{L}^\dagger \) for the backward Kolmogorov equation is

\[ \mathcal{L}^\dagger = -\gamma x \frac{\partial}{\partial x} + \frac{\sigma^2}{2} \frac{\partial^2}{\partial x^2}. \]  

In order to employ the basis expansion in (11), we rewrite \( \mathcal{L}^\dagger \) as follows:

\[ \mathcal{L}^\dagger = -\gamma (x - x_{\text{ini}}) \frac{\partial}{\partial x} - \gamma x_{\text{ini}} \frac{\partial}{\partial x} + \frac{\sigma^2}{2} \frac{\partial^2}{\partial x^2}. \]  

The three terms in (14) correspond to three events shown in Table I.

The following expectation values are obtained using the established analytical solutions of the Ornstein-Uhlenbeck process [3]:

\[ \mathbb{E} \left[ x_T - x_{\text{ini}} \right] = x_{\text{ini}} e^{-\gamma T} - x_{\text{ini}}, \]  

\[ \mathbb{E} \left[ (x_T - x_{\text{ini}})^2 \right] = \frac{\sigma^2}{2 \gamma} \left( 1 - e^{-2\gamma T} \right) + x_{\text{ini}}^2 e^{-2\gamma T} - 2x_{\text{ini}}^2 e^{-\gamma T} + x_{\text{ini}}^2. \]

In the numerical experiments below, we use \( \gamma = 1.0 \) and \( x_{\text{ini}} = 1.0 \).

The second example is the noisy van der Pol system, which has two stochastic variables:

\[ \begin{align*}
    dx_1 &= x_2 dt + \nu_{11} dW_1(t), \\
    dx_2 &= (\epsilon x_2 (1 - x_1^2) - x_1) dt + \nu_{22} dW_2(t),
\end{align*} \]  

where \( \epsilon > 0 \), \( \nu_{11} > 0 \) and \( \nu_{22} > 0 \). The noisy version of van der Pol system [21] has been employed in filtering [22] and Koopman operators [23]. The adjoint time-evolution operator \( \mathcal{L}^\dagger \) is

\[ \mathcal{L}^\dagger = x_2 \partial_{x_2} + (\epsilon x_2 (1 - x_1^2) - x_1) \partial_{x_1} + \frac{1}{2} \nu_{11} \partial_{x_1}^2 + \frac{1}{2} \nu_{22} \partial_{x_2}^2. \]  

Introducing \( x_{\text{ini}} = [x_{\text{ini},1}, x_{\text{ini},2}]^T \), we finally obtain the ten events summarized in Table I.

![FIG. 1. (Color online) Errors from the exact solutions for the second-order moment in the Ornstein-Uhlenbeck process for \( T = 1.0 \). Circle, triangle, cross, and square markers correspond to the explicit Euler method, the Heun method, the resolvent method (implicit Euler method), and the proposed local approximation of the Crank-Nicolson method, respectively.](image)

**B. Convergence rate**

Here, we check the convergence property in time. For the Ornstein-Uhlenbeck case, the evaluation of the error from the exact solutions is easy because there are the analytical solutions. For example, the second-order moment is evaluated for several different methods for \( T = 1.0 \). Figure I shows the result in which circle, triangle, cross, and square markers correspond to the explicit Euler method, the Heun method, the resolvent method (implicit Euler method), and the proposed local approximation of the Crank-Nicolson method, respectively. We can see that the Euler and resolvent methods show the first-order convergence; by contrast, the Heun and the local approximation of the Crank-Nicolson methods have the second-order convergence. Of course, several other experiments show the same convergence rate for other cases.

There is one comment. In the proposed method, we employ the second-order approximation for the resolvent (inverse matrix) in [28] and [29]. If the first-order approximation for the resolvent in [18] is employed for the Crank-Nicolson method, the convergence behavior shows the first-order one. Hence, the proposed approximation in [28] and [29] is crucial to obtain the convergence behavior of the proposed method.

Different from the Ornstein-Uhlenbeck case, we cannot obtain exact analytical solutions to the moments for the noisy van der Pol system. Hence, the fourth-order Runge-Kutta method with \( \Delta t = 10^{-10} \) is employed for \( T = 0.1 \), and the coupled ordinary differential equations for \( P_\alpha(n, t) \) are numerically solved. The finite cutoff of

---

**TABLE I. Coefficients and state-change vectors for the Ornstein-Uhlenbeck process.**

| Term | \( \gamma_v(n) \) | \( \psi_v \) |
|------|------------------|--------------|
| \( -\gamma (x - x_{\text{ini}}) \partial_x \) | \( -\gamma n \) | \([0]\) |
| \( -\gamma x_{\text{ini}} \partial_x \) | \( -\gamma n \) | \([-1]\) |
| \( (\sigma^2/2) \partial_x^2 \) | \((\sigma^2/2)n(n-1)\) | \([-2]\) |
TABLE II. Coefficients and state-change vectors for the noisy van der Pol system.

| Term                                                                 | $\gamma_r(n)$                              | $\psi_r$       |
|----------------------------------------------------------------------|--------------------------------------------|----------------|
| $(1/2)\nu_1^2 \partial_{x_1}^2$                                      | $(1/2)\nu_1^2 n_1(n_1 - 1)$               | $[-2,0]$       |
| $x_{ini,2} \partial_{x_1}$                                           | $x_{ini,2} n_1$                            | $[-1,0]$       |
| $(x_2 - x_{ini,2}) \partial_{x_1}$                                   | $n_1$                                      | $[-1,1]$       |
| $(1/2)\nu_2^2 \partial_{x_2}^2$                                      | $(1/2)\nu_2^2 n_2(n_2 - 1)$               | $[0,-2]$       |
| $-x_{ini,1} \partial_{x_2} - \epsilon (x_{ini,1})^2 x_{ini,2} \partial_{x_2} + \epsilon x_{ini,2} \partial_{x_2}$ | $-x_{ini,1} n_2 - \epsilon (x_{ini,1})^2 x_{ini,2} n_2 + \epsilon x_{ini,2} n_2$ | $[0,-1]$       |
| $-\epsilon (x_{ini,1})^2 (x_2 - x_{ini,2}) \partial_{x_2} + \epsilon (x_2 - x_{ini,2}) \partial_{x_2}$ | $-\epsilon (x_{ini,1})^2 n_2 + \epsilon n_2$ | $[0,0]$        |
| $-2\epsilon x_{ini,1} x_{ini,2} (x_1 - x_{ini,1}) \partial_{x_2} - (x_1 - x_{ini,1}) \partial_{x_2}$ | $-2\epsilon x_{ini,1} x_{ini,2} n_2 - n_2$ | $[1,-1]$       |
| $-2\epsilon x_{ini,1} (x_1 - x_{ini,1}) (x_2 - x_{ini,2}) \partial_{x_2}$ | $-2\epsilon x_{ini,1} n_2$               | $[1,0]$        |
| $-\epsilon (x_{ini,1})^2 (x_2 - x_{ini,2}) \partial_{x_2}$           | $-\epsilon x_{ini,2} n_2$                 | $[2,-1]$       |
| $(1/2) \nu_1^2 \partial_{x_1}^2$                                      | $(1/2)\nu_1^2 n_1(n_1 - 1)$               | $[-2,0]$       |

FIG. 2. (Color online) Errors from the approximated exact solutions for $E[x_1|x_2]$ in the noisy van der Pol system for $T = 0.1$. Circle, triangle, cross, and square markers correspond to the explicit Euler method, the Heun method, the resolvent method (implicit Euler method), and the proposed local approximation of the Crank-Nicolson method, respectively.

FIG. 3. (Color online) Extrapolation results for the noisy van der Pol system. The evaluated statistic is $E[x_1|x_2]$, and $T = 0.1$. Circle, triangle, cross, and square markers correspond to the raw value obtained from the resolvent method, the extrapolation in (37) for the resolvent method, the raw value obtained from the proposed method, and the extrapolation in (11) for the proposed method, respectively.

C. Extrapolation results

The second proposition of the present paper, i.e., the extrapolation based on the convergence property, is numerically checked. Figure 3 shows the estimated values for $E[x_1|x_2]$; circle, triangle, cross, and square markers correspond to the raw value obtained from the resolvent method, the extrapolation in (37) for the resolvent method, the raw value obtained from the proposed method, and the extrapolation in (11) for the proposed method, respectively. The approximated exact solution obtained from the fourth-order Runge-Kutta method with $\Delta t = 10^{-10}$ is $2.030 \times 10^{-5}$. We can see that the extrapolations give rapid convergence compared with the raw values.
D. Computational time

The main aim of the present paper is the proposition of the local approximation of the Crank-Nicolson method and the extrapolation method. Hence, the computational time is briefly mentioned here. For the noisy van der Pol system, the coupled ordinary differential equations for $P_\alpha(n, t)$ give the numerical evaluation of moments. However, the number of equations is infinite in general, and we need some finite cutoff, as stated above. For the case with $T = 0.1$ and the parameters above, the cutoff with $n_1, n_2 < 15$ was enough. Despite the small cutoff size, $\Delta t = 10^{-10}$ needs high computational time. The computational time was about 40 hours with Intel(R) Xeon(R) Gold 6128 CPU, 3.40GHz.

The experiments for the proposed algorithms were performed on MacBookAir with an M1 processor. In the proposed local approximation of the Crank-Nicolson method, a case with $M = 30$ takes about 24 seconds. That with $M = 20$ needs about 3 seconds, and that with $M = 10$ is finished in less than 0.2 seconds. The increase of $M$ needs exponential increases in the computational time. However, the proposed extrapolation method enables us to use small $M$, and the proposed algorithm will work effectively for short $T$ cases.

V. CONCLUSIONS

The local approximation of the Crank-Nicolson method is proposed, which shows the second-order convergence in time. The key of the proposed method is the local approximation of the resolvent (inverse matrix), which enables us to convert the implicit algorithm to an explicit one. Hence, we can employ the algorithm based on combinatorics. The numerical demonstration shows that the proposed algorithm actually shows the second-order of convergence in time. In addition, the convergence property naturally gives the extrapolation method, which reduces the computational costs.

As denoted in the introduction, the naive numerical algorithm based on combinatorics has been developed in the context of the duality in stochastic processes. The naive method corresponds to the resolvent method in the present paper. Although the convergence of the resolvent method is slow, the proposed method shows more rapid convergence. This fact leads to an efficient numerical estimation of statistics without any samplings.

Although the power of the local approximation of the Crank-Nicolson method is demonstrated in the present paper, there are some remaining studies. For example, the stability of the numerical algorithms should be discussed in the future. Although the original Crank-Nicolson method shows famous stable behavior, the approximation of the resolvent (inverse matrix) in [28] and [29] would vary the stability property. In addition, although the proposed algorithm based on combinatorics works well when the number of variables is small, a large number of variables takes exponentially large computational times. Hence, we need some approximation. Such approximation studies are also important for future works.

Appendix A: An example of local Crank-Nicolson method for finite cases

In [10], the following simple diffusion equation was discussed:

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2},$$  \hspace{1cm} (A1)

where $\alpha \in \mathbb{R}_+$, $x \in [0, 1]$. The initial and the boundary conditions are given as

$$u(x, 0) = f(x), \quad \text{for } x \in (0, 1),$$ \hspace{1cm} (A2)

$$u(0, t) = 0, \quad u(1, t) = 0 \quad \text{for } t \geq 0. \hspace{1cm} (A3)$$

The conventional centered difference method leads to

$$\frac{dU(t)}{dt} = \frac{\alpha}{(\Delta x)^2} AU(t),$$ \hspace{1cm} (A4)

where $U(t) = [u(x_1, t), u(x_2, t), \ldots, u(x_{N-1}, t)]^T$, $\Delta x = 1/N$, and $x_i = i\Delta x$ ($i = 1, \ldots, N-1$). The coefficient matrix $A$ is given as

$$A = \begin{bmatrix}
-2 & 1 & 0 & 0 & 0 & \cdots & 0 \\
1 & -2 & 1 & 0 & 0 & \cdots & 0 \\
0 & 1 & -2 & 1 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & \cdots & \cdots & 1 & -2 & 1 & 0 \\
0 & \cdots & \cdots & 0 & 1 & -2 & 1 \\
0 & \cdots & \cdots & 0 & 0 & 1 & -2 \\
\end{bmatrix}. \hspace{1cm} (A5)$$

Here, we introduce the following matrices $A_i$:

$$A_1 = \begin{bmatrix}
-2 & 1 & 0 & \cdots & 0 \\
1 & -1 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & \cdots & \cdots & 0 & 0 \\
\end{bmatrix}, \hspace{1cm} (A6)$$

$$A_{N-1} = \begin{bmatrix}
0 & \cdots & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & \cdots & \cdots & -1 & 1 \\
0 & 0 & 1 & -2 \\
\end{bmatrix}, \hspace{1cm} (A7)$$
and

$$A_i = \begin{bmatrix} 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(A8)

for \(i = 2, \ldots, N-2\). Then, the coefficient matrix in (A5) is split into

$$A = \sum_{i=1}^{N-1} A_i.$$  \hspace{1cm} (A9)

The decomposition of \(A\) is suitable for the local Crank-Nicolson method in [10], as shown in Sec. [11].

**Appendix B: Examples of approximation of the inverse matrix**

The original \(L^1\) is infinite-dimensional for the basis expansion of [11]. Hence, it is difficult to give mathematically rigorous discussions to approximate the resolvent in (27). Here, we discuss a finite example with a \(4 \times 4\) matrix, which is helpful to obtain the expressions in [28] and [29].

Let us define

$$1 - hA = \begin{bmatrix} 1 - ha_{11} & -ha_{12} & -ha_{13} & -ha_{14} \\ -ha_{21} & 1 - ha_{22} & -ha_{23} & -ha_{24} \\ -ha_{31} & -ha_{32} & 1 - ha_{33} & -ha_{34} \\ -ha_{41} & -ha_{42} & -ha_{43} & 1 - ha_{44} \end{bmatrix},$$

(B1)

where \(h\) is assumed to be small. The inverse matrix is denoted as

$$C = (1 - hA)^{-1}.$$ \hspace{1cm} (B2)

Using the determinant and adjugate, \(C\) is calculated as follows [24]:

$$C = \frac{1}{\det(1 - hA)} \text{adj}(1 - hA).$$ \hspace{1cm} (B3)

After tedious calculations (the \texttt{sympy} package [25] was employed), we have the following expression for \(c_{11}\):

$$c_{11} = \frac{1}{1 - ha_{11} - h^2a_{12}a_{21} - h^2a_{13}a_{31} - h^2a_{14}a_{41} + \mathcal{O}(h^3)}.$$ \hspace{1cm} (B4)

The other diagonal matrix element has the similar form, i.e.,

$$c_{ii} \approx \left(1 - ha_{ii} - h^2 \sum_{j \neq i} a_{ij}a_{ji} \right)^{-1}.$$ \hspace{1cm} (B5)

Using the following Taylor expansion,

$$\frac{1}{1 - x} = 1 + x + x^2 + \mathcal{O}(x^3),$$ \hspace{1cm} (B6)

we can confirm that the expression (B3) is the second order in terms of \(h\). From these discussions, It is easy to guess [28] for \(L^1\).

In order to obtain the non-diagonal elements, for example, we focus on the following identity:

$$\begin{aligned}
\frac{ha_{23}}{(1 - ha_{22} - h^2a_{21}a_{12} - h^2a_{23}a_{32} - h^2a_{24}a_{42}) (1 - ha_{33} - h^2a_{31}a_{13} - h^2a_{34}a_{43})} \\
+ \frac{h^2a_{21}a_{13}}{(1 - ha_{22} - h^2a_{21}a_{12} - h^2a_{23}a_{32} - h^2a_{24}a_{42}) (1 - ha_{33} - h^2a_{31}a_{13} - h^2a_{34}a_{43}) (1 - ha_{11} - h^2a_{14}a_{41})} \\
+ \frac{h^2a_{24}a_{43}}{(1 - ha_{22} - h^2a_{21}a_{12} - h^2a_{23}a_{32} - h^2a_{24}a_{42}) (1 - ha_{33} - h^2a_{31}a_{13} - h^2a_{34}a_{43}) (1 - ha_{11} - h^2a_{14}a_{41})} \\
= \frac{g_{23}^{\text{numer}}}{g_{23}^{\text{denom}}} + \mathcal{O}(h^3). \hspace{1cm} (B7)
\end{aligned}$$

For example, tedious calculations show the matrix element \(c_{23}\) is denoted as

$$c_{23} = \frac{g_{23}^{\text{numer}}}{g_{23}^{\text{denom}}} + \mathcal{O}(h^3).$$ \hspace{1cm} (B8)
Then, we have

\[
   c_{ij} \approx \frac{h_{aij}}{1 - h_{aii} - h^2 \sum_{j \neq i} a_{ij} a_{ji}} \left( 1 - h_{aij} - h^2 \sum_{j \neq i} a_{ij} a_{ji} \right) + \frac{h^2 a_{ik} a_{kj}}{(1 - h_{aii})(1 - h_{akk})(1 - h_{ajj})}.
\]

(B9)

Hence, (29) is obtained for the \( L^\dagger \) case.

Note that it is possible to neglect some terms in the denominators of the r.h.s. in (B9) because we need the second-order approximation. However, these terms increase numerical accuracy, while the additional computational costs are low. Of course, it could be possible to include second-order terms in \( h \) to each parenthesis in the denominator of the final term in (B9). However, the addition of these terms needs high computational costs. We should pay attention to the balance between accuracy and computational cost. Hence, the proposed expression in (29) would be adequate. If there is an efficient technique to implement the second-order terms, we will have higher accuracy; this is one of the future works.

ACKNOWLEDGMENTS

This work was supported by JST FOREST Program (Grant Number JPMJFR216K, Japan).

[1] G. Evensen, Data assimilation: the ensemble Kalman filter, 2nd edition (Springer, Berlin Heidelberg, 2009).
[2] P.E. Kloeden and E. Platen, Numerical solution of stochastic differential equations (Springer, Berlin, 1992).
[3] C. Gardiner, Stochastic methods: A handbook for the natural and social sciences, 4th edition (Springer, Berlin Heidelberg, 2009).
[4] H. Risken, The Fokker-Planck equation: methods of solution and applications, 2nd edition (Springer, Berlin Heidelberg, 1989).
[5] H. Singer, Comp. Stat. 21, 385 (2006).
[6] A.J. Ekanayake and L.J.S. Allen, Stoc. Anal. Appl. 28, 907 (2010).
[7] C. Kuehn, in Control of Self-Organizing Nonlinear Systems, edited by E. Schöll, S. Klapp and P. Hövel (Springer, Cham, 2016).
[8] T. Kato, Perturbation theory for linear operators (Springer-Verlag, Berlin, 1995).
[9] P. Knabner and L. Angermann, Numerical methods for elliptic and parabolic partial differential equations, 2nd edition (Springer, Cham, 2021).
[10] A. Abduwali, M. Sakakihara and H. Niki, Hiroshima Math. J. 24, 1 (1994).
[11] P. Huang, A. Abduwali, Comp. Math. with Appl. 59, 2452 (2010).
[12] J. Ohkubo, J. Stat. Mech. 2021, 013401 (2021).
[13] G. Amati, H. Meyer and T. Schilling J. Stat. Phys. 174, 219 (2019).
[14] Y. Zhu and D. Venturi, J. Stat. Phys. 178, 1217 (2020).
[15] J. Ohkubo, J. Phys. Soc. Jpn. 89, 044004 (2020).
[16] T.M. Liggett, Interacting Particle Systems (Classics in Mathematics), Reprint of the 1985 edition (Springer, Berlin, 2005).
[17] C. Giardinà, J. Kurchan, F. Redig and K. Vafayi, J. Stat. Phys. 135, 25 (2009).
[18] J. Ohkubo, J. Phys. A: Math. Theor. 46, 375004 (2013).
[19] S. Jansen and N. Kurt, Probab. Surveys 11, 59 (2014).
[20] J. Ohkubo, J. Phys. A: Math. Theor. 55, 224007 (2022).
[21] B. Van der Pol, London, Edinburgh Dublin Phil. Mag. J. Sci. 2, 978 (1926).
[22] S. Lakshmivarahan and D. J. Stensrud, IEEE Control Syst. Mag. 29, 34 (2009).
[23] N. Ćrnarić-Zic, S. Mačešić and I. Mezić, J. Nonlinear Sci. 30, 2007 (2020).
[24] L. Beilina, E. Karchevskii and M. Karchevskii, Numerical Linear Algebra: Theory and Applications (Springer, Cham, 2017).
[25] https://www.scipy.org/.