

PAPER • OPEN ACCESS

Inclusion–exclusion principle for open quantum systems with bosonic bath

To cite this article: Siyao Yang *et al* 2021 *New J. Phys.* **23** 063049

View the [article online](#) for updates and enhancements.



OPEN ACCESS

RECEIVED
8 March 2021REVISED
12 May 2021ACCEPTED FOR PUBLICATION
19 May 2021PUBLISHED
17 June 2021Original content from
this work may be used
under the terms of the
[Creative Commons
Attribution 4.0 licence](#).Any further distribution
of this work must
maintain attribution to
the author(s) and the
title of the work, journal
citation and DOI.

PAPER

Inclusion–exclusion principle for open quantum systems with bosonic bath

Si Yao Yang¹ , Zhenning Cai¹ and Jianfeng Lu^{2,*} ¹ Department of Mathematics, National University of Singapore, Level 4, Block S17, 10 Lower Kent Ridge Road, 119076, Singapore² Department of Mathematics, Department of Physics, and Department of Chemistry, Duke University, Box 90320, Durham NC 27708, United States of America

* Author to whom any correspondence should be addressed.

E-mail: jianfeng@math.duke.edu**Keywords:** Dyson series, inchworm Monte Carlo method, inclusion–exclusion principle, complexity analysis

Abstract

We present two fast algorithms which apply inclusion–exclusion principle to sum over the bosonic diagrams in bare diagrammatic quantum Monte Carlo and inchworm Monte Carlo method, respectively. In the case of inchworm Monte Carlo, the proposed fast algorithm gives an extension to the work [2018 Inclusion–exclusion principle for many-body diagrammatics *Phys. Rev. B* **98** 115152] from fermionic to bosonic systems. We prove that the proposed fast algorithms reduce the computational complexity from double factorial to exponential. Numerical experiments are carried out to verify the theoretical results and to compare the efficiency of the methods.

1. Introduction

Open quantum systems, which characterize quantum systems coupled with environment, have been studied extensively for many decades, as it arises in many context including quantum optics [9], quantum computation [31], and dynamical mean field theory [20], just to list a few. The coupling between the system and the environment leads to non-Markovian evolution of the quantum state of the system. In the weak coupling limit, such evolution can be approximated by the Markovian process described by the Lindblad equation [16, 17], which simplifies the numerical simulation. In the more challenging case where memory effect has to be taken into account, a number of numerical methods have been proposed in the literature. For example, the quasi-adiabatic propagator path integral [25, 26] method assumes finite memory length and so that the path integral can be numerically computed iteratively; by assuming that the bath response function has a special form, the hierarchical equations of motion can be applied [38, 39]; the method of multiconfiguration time dependent Hartree [3] is developed based on ansatz of wave functions. While these deterministic methods require some additional modeling of the open quantum system, the bare diagrammatic quantum Monte Carlo (dQMC) method [22] applies Monte Carlo sampling to directly compute the summations and high-dimensional integrals in the Dyson series expansion of the quantum observable [37], and after applying Wick's theorem [30], this approach can be represented as the summation of all possible diagrams, each of which is determined by a finite time sequences and a partition of them into pairs. However, such technique may encounter the notorious numerical sign problem [10–12], meaning that the number of Monte Carlo samples is required to grow at least exponentially (with respect to physical time) in order to keep the accuracy of the simulation.

Recently, the inchworm Monte Carlo method [1, 12–14, 18, 32] was proposed to mitigate the numerical sign problem. It introduces bold lines as partial resummations of bare dQMC, so that the total number of diagrams can be reduced, and the sign problem is hence suppressed. This approach is further improved in [11] by writing the evolution of the bold lines as an integro-differential equation, which only requires to sum over 'linked' diagrams, so that the computational cost can be further reduced. Even after such reductions, however, as the number of points in the time sequence m increases, the total number of

diagrams still grows as a double factorial $O((m-1)!!)$. The Monte Carlo sampling of these diagrams will again contribute to the stochastic error, when a large m is needed.

One possible approach to reduce the stochastic error is to sum up all the diagrams with the same time sequence using a deterministic method. As a direct summation is prohibitive due to the large number of diagrams, it calls for designing better algorithms to circumvent the difficulty. The fermionic bath influence functional in the bare continuous-time hybridization expansion (CTHYB) [27, 28, 36, 42], which is the counterpart of bare dQMC for bosons, can be calculated in the form of a determinant [27, 42] and thus the computational cost can be reduced to $O(m^3)$. In the inchworm method, which has less severe numerical sign problem, such a method cannot be directly applied as inchworm expansion only sums over the linked diagrams and thus corresponding bath influence functional cannot be written in a determinant form as in CTHYB.

The recent work [8] tackles this challenge with an *inverted algorithm*, which takes the idea of [33] that considers the sum of all linked diagrams at once and utilizes the massive cancellations between the diagrams, leading to an exponential rather than factorial computational complexity. It has been shown that the inverted algorithm can asymptotically achieve a computational cost at $O(m^3\alpha^m)$, which is significantly smaller than the double factorial complexity for the direct summation of all linked diagrams. The work [8] also developed another algorithm based on inclusion–exclusion principle which is even more efficient in the sense that it can further reduce the constant α in the context of inchworm hybridization expansion. The inclusion–exclusion principle describes how the cardinality (or other measures) of unions of sets can be calculated, which is also well known through the Venn diagram. This principle has been applied in a number of areas to reduce the computational complexity, including set partitioning [7], counting perfect matchings [4] and computing matrix permanents [34]. The algorithm in [8] is designed by excluding all the unlinked diagrams from the set of all diagrams, where the set of all the unlinked diagrams is represented by the union of several non-disjoint sets. This allows the inclusion–exclusion principle to be applied to the summation of diagrams, resulting in significant reduction of the computational time.

In this work, we aim to generalize these efforts on fermionic cases to bosonic cases for the simulation of open quantum systems. For bare dQMC, instead of the determinant form, the bath influence functional now holds the form of the matrix hafnian [2, 5]. By the inclusion–exclusion principle, we propose a fast algorithm with computational cost $O(2^m)$, which is efficient for small values of m (around $m \leq 20$). We then further generalize the idea to inchworm method for bosonic systems so that the summation of diagrams with the same time sequence also requires only $O(\alpha^m)$ operations. A sharp estimation of α will also be provided for our algorithm.

The rest of this paper proceeds as follows: in section 2, we introduce the Dyson series and use inclusion–exclusion principle to derive an algorithm which sums over the diagrams in the Dyson series efficiently. In section 3, another fast algorithm based on inclusion–exclusion principle is designed to sum over the linked diagrams appearing in the inchworm method. An optimization for this algorithm is further proposed, and a complexity analysis is included to examine the computational cost of the optimized algorithm. Section 4 verifies these theoretical results by numerical experiments. Finally, we draw our conclusion in section 5.

2. Dyson series with inclusion–exclusion principle

We study an open quantum system described by the von Neumann equation

$$i\frac{d\rho}{dt} = [H, \rho], \quad (1)$$

where the density matrix $\rho(t)$ and the Schrödinger picture Hamiltonian H above are both Hermitian operators on the Hilbert space $\mathcal{H} = \mathcal{H}_s \otimes \mathcal{H}_b$, with \mathcal{H}_s and \mathcal{H}_b representing respectively the Hilbert spaces associated with the system and the bath of the open quantum system. The Hamiltonian H takes the form as a combination of an uncoupled Hamiltonian H_0 and a coupling term W . Here we assume that the coupling term W takes the tensor-product form, so that we have

$$H = H_0 + W := (H_s \otimes \text{Id}_b + \text{Id}_s \otimes H_b) + W_s \otimes W_b,$$

where $H_s, W_s \in \mathcal{H}_s$, $H_b, W_b \in \mathcal{H}_b$, and Id_s, Id_b are respectively the identity operators for the system and the bath.

We are interested in the evolution of the expectation for a given observable $O = O_s \otimes \text{Id}_b$ acting only on the system part, defined by

$$\langle O(t) \rangle := \text{tr}(O\rho(t)) = \text{tr}(Oe^{-itH}\rho(0)e^{itH}). \quad (2)$$

Due to the high dimensionality of the space \mathcal{H}_b , it is usually impractical to solve $e^{\pm i t H}$ directly. One feasible approach is to apply the method of quantum Monte Carlo to approximate $\langle O(t) \rangle$ numerically. Below we will first introduce the bare dQMC based on the Dyson series expansion of $\langle O(t) \rangle$, and then propose an efficient method to compute a key term in the expansion known as the bath influence functional.

2.1. Introduction to the bare diagrammatic quantum Monte Carlo method

Upon assuming the initial density matrix has the separable form $\rho(0) = \rho_s \otimes \rho_b$ where the bath ρ_b commutes with the Hamiltonian H_b , the expectation of observable $\langle O(t) \rangle$ can be represented by the following *Dyson series* (for derivation, see [11]):

$$\begin{aligned} \langle O(t) \rangle &= \sum_{m=0}^{+\infty} i^m \int_0^{2t} ds_m \int_0^{s_m} ds_{m-1} \dots \int_0^{s_2} ds_1 (-1)^{\#\{s < t\}} \\ &\times \text{tr}_s(\rho_s \mathcal{U}^{(0)}(0, s_1, \dots, s_m, 2t)) \cdot \mathcal{L}_b(s_1, \dots, s_m). \end{aligned} \quad (3)$$

Here $\#\{s < t\}$ denotes the number of s_i which are less than t and tr_s takes trace of the system degree of freedom.

In practice, one may truncate the series above at a sufficiently large \bar{M} and evaluate those high-dimensional integrals on the right-hand side using Monte Carlo integration, resulting in the bare dQMC. This requires us to evaluate the integrand in the Dyson series (3) for each sample. The explicit formula for the propagator $\mathcal{U}^{(0)}$ is given in appendix A, which contains the observable O_s and is associated with the system space. As for the *bath influence functional* \mathcal{L}_b , we assume that the Wick's theorem can be applied so that

$$\mathcal{L}_b(s_1, \dots, s_m) = \begin{cases} 0, & \text{if } m \text{ is odd;} \\ \sum_{q \in \mathcal{Q}(s)} \prod_{(s_j, s_k) \in q} B(s_j, s_k), & \text{if } m \text{ is even,} \end{cases} \quad (4)$$

where $B : \{(\tau_1, \tau_2) | 0 \leq \tau_1 \leq \tau_2\} \rightarrow \mathbb{C}$ is the two-point bath correlation and the set $\mathcal{Q}(s_1, \dots, s_m)$ is the collection of all possible ordered pairings of the time sequence (s_1, \dots, s_m) :

$$\begin{aligned} \mathcal{Q}(s_1, \dots, s_m) &= \left\{ \{(s_{j_1}, s_{k_1}), \dots, (s_{j_{m/2}}, s_{k_{m/2}})\} \mid \{j_1, \dots, j_{m/2}, k_1, \dots, k_{m/2}\} = \{1, \dots, m\}, \right. \\ &\quad \left. j_l < k_l \text{ for any } l = 1, \dots, m/2 \right\}. \end{aligned} \quad (5)$$

For example, when $m = 4$, $\mathcal{L}_b(s_1, s_2, s_3, s_4)$ is given by

$$\mathcal{L}_b(s_1, s_2, s_3, s_4) = B(s_1, s_2)B(s_3, s_4) + B(s_1, s_3)B(s_2, s_4) + B(s_1, s_4)B(s_2, s_3). \quad (6)$$

In particular, when $m = 0$, the value of $\mathcal{L}_b(\emptyset)$ is defined as 1. With such expression of bath influence functional, the right-hand side of (3) only sums over the terms with even m . We may also express (6) using many-body diagrams:

$$\mathcal{L}_b(s_1, s_2, s_3, s_4) = \text{---} \text{---} \text{---} + \text{---} \text{---} \text{---} + \text{---} \text{---} \text{---} . \quad (7)$$

In the diagrammatic representation above, each diagram refers to a product $B(\cdot, \cdot)B(\cdot, \cdot)$ where each arc connecting a pair denotes the corresponding two-point correlation.

The major challenge on evaluating $\mathcal{L}_b(s_1, \dots, s_m)$ (m is even) is that its diagrammatic representation includes in total $(m-1)!!$ diagrams, which leads to a double factorial growth in the computational cost on calculating such a bath influence functional via *direct method* (i.e., direct summation over each diagram in the expansion such as (7)). As this cost increases drastically when m gets larger, one needs to compute a given $\mathcal{L}_b(s_1, \dots, s_m)$ using the Monte Carlo method (on top the Monte Carlo sampling of (s_1, \dots, s_m)), leading to larger stochastic error. In this section, we will show how we can benefit from the well-known inclusion–exclusion principle to greatly reduce the complexity of computing the bath influence functional.

2.2. Inclusion–exclusion principle for computing $\mathcal{L}_b(s_1, \dots, s_m)$

Mathematically, the equation (4) is known to be the hafnian of an undirected graph [2]. Several fast algorithms have been introduced to compute such quantity in the recent years [5, 6, 15, 21, 24, 29]. While most algorithms aim for a good complexity for large values of m , here we are going to introduce a novel fast algorithm for computing hafnians with small m based on the inclusion–exclusion principle.

Given a function $\mu(\cdot)$ satisfying the additivity such that $\mu(\bigcup_{i=1}^n E_i) = \sum_{i=1}^n \mu(E_i)$ for any disjoint sets $\{E_i\}_{i=1}^n$, the inclusion–exclusion principle reads

$$\mu\left(S \setminus \bigcup_{i=1}^n A_i\right) = \mu(S) - \sum_{i=1}^n \mu(A_i) + \sum_{1 \leq i < j \leq n} \mu(A_i \cap A_j) - \cdots + (-1)^n \mu(A_1 \cap \cdots \cap A_n), \quad (8)$$

where S is a given finite universal set containing A_1, A_2, \dots, A_n .

The inclusion–exclusion principle plays important roles in a number of fast algorithms such as Ryser’s algorithm for matrix permanents [34] and the diagrammatic resummation of quantum impurity models [8]. To apply this principle on the evaluation of $\mathcal{L}_b(s_1, \dots, s_m)$, we set S as the collection of all combinations of $m/2$ distinct pairs from s_1, \dots, s_m and A_i as all combinations of $m/2$ distinct pairs from s_1, \dots, s_m except s_i :

$$\begin{aligned} S &= \left\{ \left((x_1, y_1), \dots, (x_{m/2}, y_{m/2}) \right) \mid x_j = s_{j_1}, y_j = s_{j_2} \in \{s_1, \dots, s_m\} \text{ with } j_1 < j_2 \right. \\ &\quad \left. \text{for any } j = 1, \dots, m/2 \right\}, \\ A_i &= \left\{ \left((x_1, y_1), \dots, (x_{m/2}, y_{m/2}) \right) \mid \right. \\ &\quad \left. x_j = s_{j_1}, y_j = s_{j_2} \in \{s_1, \dots, s_{i-1}, s_{i+1}, \dots, s_m\} \text{ with } j_1 < j_2 \text{ for any } j = 1, \dots, m/2 \right\}. \end{aligned}$$

We point out that

- For any element of S or A_i , one pair may appear multiple times, e.g. $((s_1, s_2), (s_1, s_2)) \in S$.
- The elements in S and A_i are ordered: The same pairs arranged in different orders form different elements in these sets, e.g. $((s_1, s_3), (s_1, s_2))$ and $((s_1, s_2), (s_1, s_3))$ are different elements of S .

Below we provide all these sets for $m = 4$ represented by diagrams as an example:

$$\begin{aligned} S &= \left\{ \begin{array}{c} \text{Diagram 1}, \text{Diagram 2}, \text{Diagram 3}, \text{Diagram 4}, \text{Diagram 5}, \text{Diagram 6}, \\ \text{Diagram 7}, \text{Diagram 8}, \text{Diagram 9}, \text{Diagram 10}, \text{Diagram 11}, \text{Diagram 12}, \\ \text{Diagram 13}, \text{Diagram 14}, \text{Diagram 15}, \text{Diagram 16}, \text{Diagram 17}, \text{Diagram 18}, \\ \text{Diagram 19}, \text{Diagram 20}, \text{Diagram 21}, \text{Diagram 22}, \text{Diagram 23}, \text{Diagram 24}, \\ \text{Diagram 25}, \text{Diagram 26}, \text{Diagram 27}, \text{Diagram 28}, \text{Diagram 29}, \text{Diagram 30}, \\ \text{Diagram 31}, \text{Diagram 32}, \text{Diagram 33}, \text{Diagram 34}, \text{Diagram 35}, \text{Diagram 36} \end{array} \right\}; \\ A_1 &= \left\{ \begin{array}{c} \text{Diagram 37}, \text{Diagram 38}, \text{Diagram 39}, \\ \text{Diagram 40}, \text{Diagram 41}, \text{Diagram 42}, \\ \text{Diagram 43}, \text{Diagram 44}, \text{Diagram 45} \end{array} \right\}; \quad A_2 = \left\{ \begin{array}{c} \text{Diagram 46}, \text{Diagram 47}, \text{Diagram 48}, \\ \text{Diagram 49}, \text{Diagram 50}, \text{Diagram 51}, \\ \text{Diagram 52}, \text{Diagram 53}, \text{Diagram 54} \end{array} \right\}; \\ A_3 &= \left\{ \begin{array}{c} \text{Diagram 55}, \text{Diagram 56}, \text{Diagram 57}, \\ \text{Diagram 58}, \text{Diagram 59}, \text{Diagram 60}, \\ \text{Diagram 61}, \text{Diagram 62}, \text{Diagram 63} \end{array} \right\}; \quad A_4 = \left\{ \begin{array}{c} \text{Diagram 64}, \text{Diagram 65}, \text{Diagram 66}, \\ \text{Diagram 67}, \text{Diagram 68}, \text{Diagram 69}, \\ \text{Diagram 70}, \text{Diagram 71}, \text{Diagram 72} \end{array} \right\}. \end{aligned}$$

Each diagram in the braces refers to a pairing in S or A_i whose first and second components are represented by blue and red arcs respectively. For instance, we have

$$((s_1, s_2), (s_1, s_2)) = \text{Diagram 1}, \quad ((s_1, s_2), (s_1, s_3)) = \text{Diagram 2}, \dots$$

One can observe that s_i (the point marked in green) never occurs in A_i . Regardless of the blue/red color of arcs, all diagrams of A_i are excluded from $\mathcal{Q}(s_1, s_2, s_3, s_4)$, which is the collection of the diagrams in the diagrammatic representation of $\mathcal{L}_b(s_1, s_2, s_3, s_4)$. In fact, the union of A_i contains all diagrams that are not in $\mathcal{Q}(s_1, \dots, s_m)$, and therefore $S \setminus \bigcup_{i=1}^m A_i$ is the set formed by arranging all the pairings in $\mathcal{Q}(s_1, \dots, s_m)$ in all

possible orders. Denote a set function $\mu(\cdot)$ as

$$\mu(E) := \sum_{(x_1, y_1), \dots, (x_{m/2}, y_{m/2}) \in E} B(x_1, y_1) \dots B(x_{m/2}, y_{m/2}) \text{ for } E = S, A_1, \dots, A_m$$

the left-hand side of inclusion–exclusion principle (8) is then given by

$$\mu\left(S \setminus \bigcup_{i=1}^m A_i\right) = \mathcal{L}_b(s_1, \dots, s_m) \cdot (m/2)! \quad (9)$$

Here the combinatorial factor $(m/2)!$ takes into account the possible permutations of pairs in $((s_{j_1}, s_{k_1}), \dots, (s_{j_{m/2}}, s_{k_{m/2}}))$ which all refer to the same element in the set.

On the other hand, the terms in the right-hand side of (8) are

$$\begin{aligned} \mu(S) &= \left(\sum_{1 \leq i < j \leq m} B(s_i, s_j) \right)^{m/2}, \\ \mu(A_{k_1} \cap \dots \cap A_{k_l}) &= \left(\sum_{\substack{1 \leq i < j \leq m \\ i \neq k_1, j \neq k_1 \\ \vdots \\ i \neq k_l, j \neq k_l}} B(s_i, s_j) \right)^{m/2} \text{ for } l = 1, \dots, m-2 \text{ and } 1 \leq k_1 < \dots < k_l \leq m. \end{aligned} \quad (10)$$

Note that the intersection $A_{k_1} \cap \dots \cap A_{k_l}$ does not include s_{k_1}, \dots, s_{k_l} and thus the i, j indices in the subscript of the summation can never be assigned as k_1, \dots, k_l . In addition, $A_{k_1} \cap \dots \cap A_{k_l}$ is empty for $l = m-1, m$, and therefore the corresponding values of $\mu(\cdot)$ is zero.

At this point, we can combine (10) with (9) and reach the following formula:

Theorem 1. *Given the increasing time sequence (s_1, \dots, s_m) with m being an even number, $\mathcal{L}_b(s_1, \dots, s_m)$ defined in (4) can be calculated by*

$$\begin{aligned} \mathcal{L}_b(s_1, \dots, s_m) &= \left[(Q)^{\frac{m}{2}} - \sum_{k_1=1}^m (Q_{k_1})^{\frac{m}{2}} + \sum_{1 \leq k_1 < k_2 \leq m} (Q_{k_1 k_2})^{\frac{m}{2}} - \dots \right. \\ &\quad \left. \dots + \sum_{1 \leq k_1 < \dots < k_{m-2} \leq m} (Q_{k_1 k_2 \dots k_{m-2}})^{\frac{m}{2}} \right] / \left(\frac{m}{2} \right)! \end{aligned} \quad (11)$$

where

$$Q = \sum_{1 \leq i < j \leq m} B(s_i, s_j) \quad \text{and} \quad Q_{k_1 k_2 \dots k_n} = \sum_{\substack{1 \leq i < j \leq m \\ i \neq k_1, j \neq k_1 \\ \vdots \\ i \neq k_n, j \neq k_n}} B(s_i, s_j) \quad \text{for } n = 1, \dots, m-2.$$

The equation (11) can be used to calculate $\mathcal{L}_b(s_1, \dots, s_m)$ for given values of $B(s_i, s_j)$. Again in the example of $m = 4$, this equation can be expanded as

$$\begin{aligned} &\mathcal{L}_b(s_1, s_2, s_3, s_4) \\ &= \frac{1}{2} \left[(B(s_1, s_2) + B(s_1, s_3) + B(s_1, s_4) + B(s_2, s_3) + B(s_2, s_4) + B(s_3, s_4))^2 \right. \\ &\quad - (B(s_2, s_3) + B(s_2, s_4) + B(s_3, s_4))^2 - (B(s_1, s_3) + B(s_1, s_4) + B(s_3, s_4))^2 \\ &\quad - (B(s_1, s_2) + B(s_1, s_4) + B(s_2, s_4))^2 - (B(s_1, s_2) + B(s_1, s_3) + B(s_2, s_3))^2 \\ &\quad \left. + (B(s_3, s_4))^2 + (B(s_2, s_4))^2 + (B(s_2, s_3))^2 + (B(s_1, s_4))^2 + (B(s_1, s_3))^2 + (B(s_1, s_2))^2 \right]. \end{aligned} \quad (12)$$

It can be checked by direct calculation that the cancellations among these two-point correlations will finally lead to the same result as (6) from the definition of $\mathcal{L}_b(s_1, \dots, s_m)$, which justifies the equivalence between the two approaches to evaluate the bath influence functional.

The formula (12) does not seem to hold any advantage at first glance. Indeed, for small values of m , the formula (11) requires more operations than the direct approach using the definition (4). However, the

computational cost of such inclusion–exclusion principle will become significantly cheaper when m is large, which is

$$\begin{aligned} & \binom{m}{0} \binom{m}{2} + \binom{m}{1} \binom{m-1}{2} + \cdots + \binom{m}{m-2} \binom{2}{2} \\ &= \sum_{n=0}^{m-2} \frac{m!}{n!(m-n)!} \cdot \frac{(m-n)!}{2(m-n-2)!} = \frac{m(m-1)}{2} \cdot \sum_{n=0}^{m-2} \binom{m-2}{n} \sim O(m^2 2^m). \end{aligned} \quad (13)$$

Here the binomial coefficient $\binom{m}{n}$ is the number of terms in the summation in (11) with respect to k_1, \dots, k_n , and the binomial coefficient $\binom{m-n}{2}$ corresponds to the number of terms in the definition of $Q_{k_1 \dots k_n}$. This cost grows considerably slower than the double factorial for the direct calculation of the bath influence functional. We note that the reduction of computational complexity can be compared to the Ryser formula [34] for computing the permanent of an $m \times m$ matrix, which is also derived from the inclusion–exclusion principle with the computational cost also being $O(m^2 2^m)$. In our case, we are able to further reduce the computational cost to $O(2^m)$ by calculating $Q_{k_1 k_2 \dots k_n}$ iteratively. Specifically, we define the symmetrization of the two-point correlation as

$$\bar{B}(s_i, s_j) = \begin{cases} B(s_i, s_j) & \text{if } s_i < s_j, \\ 0 & \text{if } s_i = s_j, \\ B(s_j, s_i) & \text{if } s_i > s_j. \end{cases}$$

Below we use $\bar{B} \in \mathbb{C}^{m \times m}$ to denote the symmetric matrix whose entries are $\bar{B}(s_i, s_j)$. Based on such definition, the desired bath influence functional is then computed by the hafnian [2] of the symmetric matrix \bar{B} , while Q and each $Q_{k_1 k_2 \dots k_n}$ are expressed in terms of the summation over entries of \bar{B} :

$$\begin{aligned} Q &= \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \bar{B}(s_i, s_j), \\ Q_{k_1 k_2 \dots k_n} &= \frac{1}{2} \sum_{\substack{1 \leq i \leq m \\ i \neq k_1, \dots, k_n}} \sum_{\substack{1 \leq j \leq m \\ j \neq k_1, \dots, k_n}} \bar{B}(s_i, s_j). \end{aligned}$$

Note that the constant $\frac{1}{2}$ is needed here since we have taken into each term twice in the symmetrized summation. From this definition, we can observe that $Q_{k_1 k_2 \dots k_n}$ is a half of the sum over all the entries of \bar{B} excluding the k_i th rows and columns for $i = 1, \dots, n$. Thus the value of $Q_{k_1 k_2 \dots k_n}$ can be obtained from the $Q_{k_1 k_2 \dots k_{n-1}}$ by taking away the k_n th row and column. We are then inspired to define

$$\begin{aligned} R_{k_1 k_2 \dots k_{n-1}}^{(i)} &= \sum_{j=1}^m \bar{B}(s_i, s_j) - \sum_{j=1}^{n-1} \bar{B}(s_i, s_{k_j}) \\ &= \frac{1}{2} \left(\sum_{j=1}^m \bar{B}(s_i, s_j) - \sum_{j=1}^{n-1} \bar{B}(s_i, s_{k_j}) \right) + \frac{1}{2} \left(\sum_{j=1}^m \bar{B}(s_j, s_i) - \sum_{j=1}^{n-1} \bar{B}(s_{k_j}, s_i) \right) \end{aligned} \quad (14)$$

which describes the summation over i th row and column except the k_1 th, k_2 th, \dots , k_{n-1} th entries. The underlined terms are subtracted since they do not exist in the sum $Q_{k_1 k_2 \dots k_{n-1}}$, and the diagonal entry $\bar{B}(s_i, s_i)$ is counted twice but this does not matter since it equals zero by definition. Now we can compute $Q_{k_1 k_2 \dots k_n}$ by

$$Q_{k_1 k_2 \dots k_n} = Q_{k_1 k_2 \dots k_{n-1}} - R_{k_1 k_2 \dots k_{n-1}}^{(k_n)}. \quad (15)$$

Note that this relation also holds for $n = 1$, for which the left-hand side of (14) becomes $R^{(i)}$, denoting the sum of the i th row of the matrix \bar{B} . The equation (15) reduces the computational cost of each $Q_{k_1 k_2 \dots k_n}$ to $O(1)$ once the initial value Q is given, and each $R_{k_1 k_2 \dots k_{n-1}}^{(k_n)}$ can also be obtained iteratively by only one subtraction:

$$R_{k_1 k_2 \dots k_n}^{(i)} = R_{k_1 k_2 \dots k_{n-1}}^{(i)} - \bar{B}(s_{k_n}, s_i), \quad (16)$$

which can be easily seen according to its definition.

For a more intuitive understanding, one may refer to figure 1 to visualize the procedures to compute a simple example as Q_{24} : each node is assigned with the value of the corresponding entry of the matrix \bar{B}

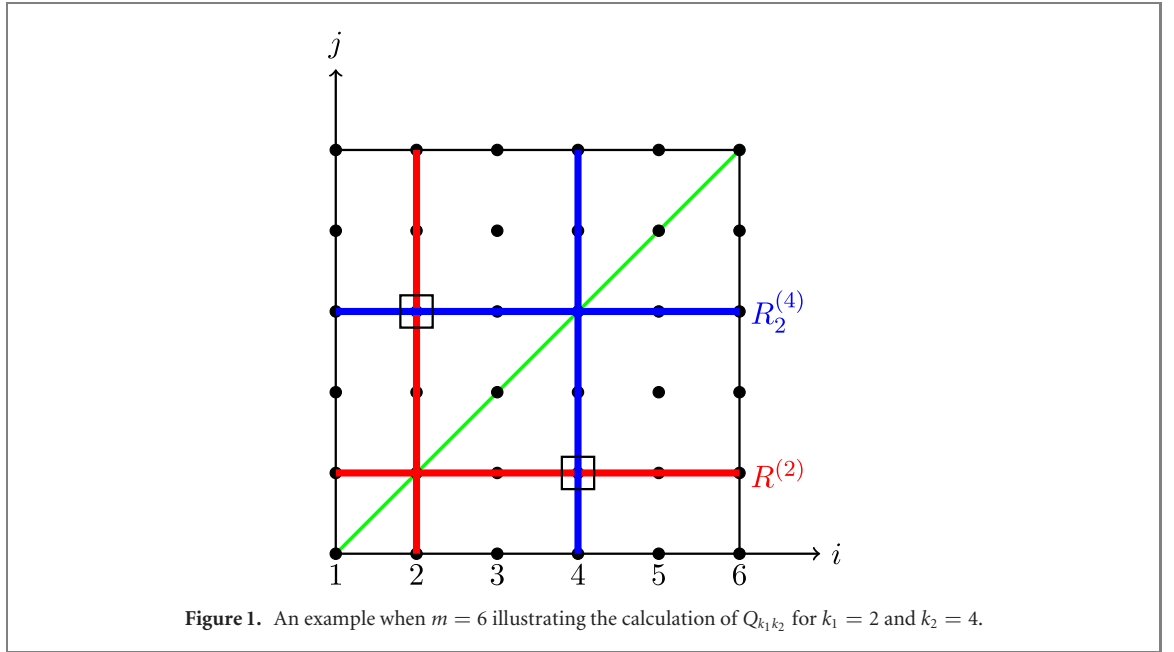


Figure 1. An example when $m = 6$ illustrating the calculation of $Q_{k_1 k_2}$ for $k_1 = 2$ and $k_2 = 4$.

Algorithm 1. Inclusion–exclusion principle for computing $\mathcal{L}_b(s_1, \dots, s_m)$.

```

1: set  $Q \leftarrow \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \bar{B}(s_i, s_j)$  ▷ Initial setting
2: for  $i$  from 1 to  $m$  do
3:    $R^{(i)} \leftarrow \sum_{j=1}^m \bar{B}(s_i, s_j)$ 
4:    $Q_i \leftarrow Q - R^{(i)}$ 
5:   for  $\bar{k}$  from 1 to  $i - 1$  do
6:     for  $n$  from 0 to  $\max(\min(m - 4, \bar{k} - 1), 0)$  do
7:       for  $1 \leq k_1 < k_2 < \dots < k_n < \bar{k}$  do
8:          $R_{k_1 \dots k_n \bar{k}}^{(i)} \leftarrow R_{k_1 \dots k_n}^{(i)} - \bar{B}(s_{\bar{k}}, s_i)$ 
9:          $Q_{k_1 \dots k_n \bar{k} i} \leftarrow Q_{k_1 \dots k_n \bar{k}} - R_{k_1 \dots k_n \bar{k}}^{(i)}$ 
10:      end for
11:    end for
12:  end for
13: end for
14: compute  $\mathcal{L}_b(s_1, \dots, s_m)$  according to (11) ▷ Final step
15: return  $\mathcal{L}_b(s_1, \dots, s_m)$ 

```

(with the coefficient $\frac{1}{2}$), so Q is simply the summation over all such nodes. We can reach to the desired Q_{24} by the following two steps:

- Calculate $R^{(2)}$ (summation over the red lines) and obtain Q_2 (summation over all nodes that are not on red lines) using relation (15);
- Calculate $R_2^{(4)}$ (summation over the blue lines excluding the two boxed nodes) using relation (16) and get Q_{24} (summation over all nodes that are on neither red nor blue lines) again by relation (15).

Note that the nodes on the green diagonal are all equal to zero, which explains why the double counting on the nodes of intersection on red/blue lines will not affect the result of the calculation as we have mentioned previously.

To end this section, we examine the computational cost of such procedures which are written in details as algorithm 1: the major complexity concentrates in the calculation of $R_{k_1 k_2 \dots k_n}^{(i)}$ and $Q_{k_1 k_2 \dots k_n}$ in line 8 and 9, both of which require 1 subtraction in each iteration for the total 2^m iterations, and the evaluation of the final step in line 13 whose cost is again 2^m . Other computations such as the initial settings for $R^{(i)}$ and Q need at most m^2 operations and thus are minor. Consequently, algorithm 1 has the complexity at $O(2^m)$, which is much cheaper compared to the original (13).

Compared with previous works on the computation of hafnians, this algorithm does not have the optimal time complexity. In [5], Björklund *et al* have proposed an algorithm that computes the hafnian of a $m \times m$ matrix with time complexity $O(m^3 2^{m/2})$, which requires computation of all the eigenvalues of $2^{m/2}$ matrices. Asymptotically, such an algorithm is faster than our algorithm for large m . However, according to our experiments, algorithm 1 is not slower than Björklund's algorithm up to $m = 22$ due to a relatively smaller prefactor of the overall complexity, which is sufficiently efficient as a satisfactory convergence of

Dyson series generally will not require a very large m . We have attached our MATLAB code for algorithm 1 in appendix B and one may compare the time efficiency of Björklund's algorithm with ours.

3. Inchworm Monte Carlo method with inclusion–exclusion principle

The bare dQMC method can be only applied to short-time simulation, since the variance of the integrand in the Monte Carlo method grows exponentially with simulation time, which is known as the dynamical sign problem [27, 28, 35]. One approach to alleviate the sign problem is the *inchworm Monte Carlo method* proposed in [12], which introduces the full propagator $G(s_i, s_f)$ defined by (see [11] for a derivation)

$$G(s_i, s_f) = \sum_{\substack{m=0 \\ m \text{ is even}}}^{+\infty} i^m \int_{s_i}^{s_f} ds_m \int_{s_i}^{s_m} ds_{m-1} \dots \int_{s_i}^{s_2} ds_1 \times (-1)^{\#\{s < t\}} \mathcal{U}^{(0)}(s_i, s_1, \dots, s_m, s_f) \cdot \mathcal{L}_b(s_1, \dots, s_m) \quad (17)$$

for the initial time point $s_i \in [0, 2t] \setminus \{t\}$ and the final time point $s_f \in [s_i, 2t] \setminus \{t\}$. One may compare the definition of such a full propagator with the desired expectation of observable (3) to find the relation $\langle O(t) \rangle = \text{tr}_s(\rho_s G(0, 2t))$, suggesting that we should obtain $\langle O(t) \rangle$ by studying the evolution of $G(s_i, s_f)$.

In [11, section 4], an integro-differential equation formulation for the full propagator is proposed as

$$\frac{\partial G(s_i, s_f)}{\partial s_f} = \text{sgn}(s_f - t) [iH_s G(s_i, s_f) + \sum_{\substack{m=2 \\ m \text{ is even}}}^{+\infty} i^m \int_{s_i}^{s_f} ds_{m-1} \int_{s_i}^{s_{m-1}} ds_{m-2} \dots \int_{s_i}^{s_2} ds_1 \times (-1)^{\#\{s \leq t\}} W_s \mathcal{U}(s_i, s_1, \dots, s_{m-1}, s_f) \cdot \mathcal{L}_b^c(s_1, \dots, s_{m-1}, s_f)]. \quad (18)$$

Here we recall that W_s is the perturbation associated with the system, and the functional \mathcal{U} is defined similarly to $\mathcal{U}^{(0)}$ in (17) with the bare propagator $G^{(0)}(\cdot, \cdot)$ replaced by the full propagator $G(\cdot, \cdot)$. Its formula together with some important properties of the full propagator are summarized in appendix A.

The definition of \mathcal{L}_b^c is similar to (4):

$$\mathcal{L}_b^c(s_1, \dots, s_m) = \sum_{q \in \mathcal{Q}_c(s_1, \dots, s_m)} \prod_{(s_j, s_k) \in q} B(s_j, s_k), \quad (19)$$

where \mathcal{Q}_c denotes the set of linked pairings:

$$\mathcal{Q}_c(s_1, \dots, s_m) = \{q \in \mathcal{Q}(s_1, \dots, s_m) | q \text{ is linked}\}.$$

By saying q is ‘linked’ in the diagrammatic representation, we mean that all points in a diagram are connected with each other using arcs as ‘bridges’. In the same example (7) as when $m = 4$, the second diagram on the right-hand side is considered to be linked since one may start from any of the four points and reach to any other one going through the path formed by the union of the arcs. More rigorously, this linkedness is defined as follows.

Definition 1 (Linked pairs). Two pairs of real numbers (s_1, s_2) and (τ_1, τ_2) satisfying $s_1 \leq s_2$ and $\tau_1 \leq \tau_2$ are *linked* if either of the following two statements holds:

- (a) $s_1 \leq \tau_1 \leq s_2$ and $\tau_1 \leq s_2 \leq \tau_2$.
- (b) $\tau_1 \leq s_1 \leq \tau_2$ and $s_1 \leq \tau_2 \leq s_2$.

Definition 2 (Linked sets of pairs). Given two sets of pairs q_1 and q_2 , we say q_1 and q_2 are *linked* if there exists $(s_1, s_2) \in q_1$ and $(\tau_1, \tau_2) \in q_2$ such that (s_1, s_2) and (τ_1, τ_2) are linked. We say a given set of pairs q is *linked* if it cannot be decomposed into the union of two sets of pairs that are not linked.

When two sets of pairs q_1 and q_2 are linked, we also say that q_1 is linked to q_2 and vice versa.

For example, the first diagram on the right-hand side of (7) is not linked since it can be decomposed into $q_1 \cup q_2$ where $q_1 = \{(s_1, s_2)\}$ and $q_2 = \{(s_3, s_4)\}$ and obviously q_1 is not linked to q_2 . For the same reason, the third diagram is not linked either. Therefore, only the second diagram is linked and

$$\mathcal{L}_b^c(s_1, s_2, s_3, s_4) = B(s_1, s_3)B(s_2, s_4) \quad (20)$$

which is diagrammatically expressed as

$$\text{Diagram 1} = \text{Diagram 2},$$

where \mathcal{L}_b^c is denoted by a rounded box covering time sequence (s_1, \dots, s_m) . Another example for this many-body diagrammatic representation when $m = 6$ is given below:

$$\begin{aligned} \mathcal{L}_b^c(s_1, s_2, s_3, s_4, s_5, s_6) &= \text{diagram of a rounded box with 6 points} \\ &= \text{diagram 1} + \text{diagram 2} + \text{diagram 3} + \text{diagram 4} \end{aligned} \quad (21)$$

We observe that the rounded box above only contains four linked diagrams, while the corresponding bath influence functional $\mathcal{L}_b(s_1, s_2, s_3, s_4, s_5, s_6)$ is the summation of all possible ordered pairings including diagrams like

$$\text{diagram 5}, \text{diagram 6}, \dots \quad (22)$$

which are not linked.

Compared with the Dyson series, the advantage of the equation (18) is that its series with respect to m has a faster convergence than that in (3), leading to a less severe numerical sign problem. Also, the number of diagrams in $\mathcal{L}_b^c(s_1, \dots, s_m)$ grows asymptotically as $e^{-1}(m-1)!!$ [37], which is less than the number of diagrams in $\mathcal{L}_b(s_1, \dots, s_m)$. To solve (19) numerically, one can apply Runge–Kutta type methods to discretize the time s_f , and the integrals on the right-hand side of (18) are approximated by the Monte Carlo method, especially for large m . Thus, it can be expected that most of the computational time is spent on the evaluation of $\mathcal{L}_b^c(s_1, \dots, s_m)$ defined in (19), and hence, a fast algorithm for $\mathcal{L}_b^c(s_1, \dots, s_m)$ is desirable. Below, we are going to combine the result in section 2.1 and the technique developed in [8] to accelerate the computation of \mathcal{L}_b^c for large m .

3.1. Inclusion–exclusion principle for computing rounded box $\mathcal{L}_b^c(s_1, \dots, s_m)$

Since any rounded box covering two points can be directly evaluated by the corresponding two-point correlation, we only consider the calculation of $\mathcal{L}_b^c(s_1, \dots, s_m)$ with even $m \geq 4$ in the rest of this section. In the inclusion–exclusion principle (8), we set

$$\begin{aligned} S &= \mathcal{Q}^*(s_1, \dots, s_m) \\ &:= \left\{ \{(s_{j_1}, s_{k_1}), \dots, (s_{j_{m/2}}, s_{k_{m/2}})\} \mid \{j_1, \dots, j_{m/2}, k_1, \dots, k_{m/2}\} = \{1, \dots, m\}, \right. \\ &\quad \left. k_l - j_l \geq 2 \text{ for any } l = 1, \dots, m/2 \right\} \end{aligned} \quad (23)$$

equipped with the set function

$$\mu(S) = \sum_{q \in S} \prod_{(s_j, s_k) \in q} B(s_j, s_k).$$

In (23), $\mathcal{Q}^*(s_1, \dots, s_m)$ is similarly defined as $\mathcal{Q}(s_1, \dots, s_m)$ in (5) but does not include any pair formed by two adjacent time points, i.e., we do not consider any pair (s_{j_l}, s_{k_l}) with $k_l - j_l = 2$ in S . This difference in definition is based on the fact that an ordered pairing with an arc connecting two adjacent points such as the ones in (22) will never be linked, and thus is not considered in a rounded box. Note that $S = \emptyset$ when $m = 2$, which explains why we restrict our discussion for $m \geq 4$ in this section. Upon further introducing

$$B^*(s_j, s_k) = \begin{cases} B(s_j, s_k), & \text{if } k - j > 1, \\ 0, & \text{if } k - j = 0, \end{cases}$$

it is obvious that

$$\mathcal{L}_b^*(s_1, \dots, s_m) := \mu(S) = \sum_{q \in \mathcal{Q}^*(s)} \prod_{(s_j, s_k) \in q} B(s_j, s_k) = \sum_{q \in \mathcal{Q}(s)} \prod_{(s_j, s_k) \in q} B^*(s_j, s_k). \quad (24)$$

Diagrammatically, we express a given $\mathcal{L}_b^*(s_1, \dots, s_m)$ by a rectangular box. For example when $m = 6$, we have

$$\text{diagram of a rectangular box with 6 points} = \text{diagram 1} + \text{diagram 2} + \text{diagram 3} + \text{diagram 4} + \text{diagram 5}. \quad (25)$$

Similar as the bath influence functional $\mathcal{L}_b(s_1, \dots, s_m)$, the rectangular box above contains all linked diagrams (the first four diagrams on the right-hand side). However, $\mathcal{L}_b^*(s_1, \dots, s_m)$ has only one unlinked diagram (the last diagram) and does not include any unlinked diagrams with adjacent pairs. Such idea has also been applied in [8] referred as ‘second optimization’ to eliminate some unlinked diagrams that will not be used in inchworm method. By writing $\mathcal{L}_b^*(s_1, \dots, s_m)$ as the last formula of (24), we can again apply

algorithm 1 to compute a given m -point rectangular box at the complexity of $O(2^m)$ upon setting all entries on the subdiagonal and superdiagonal of the matrix in figure 1 to be zero.

To continue the inclusion–exclusion principle, we further let

$$A_V = \{q \in S \mid q \text{ has a linked component } q_V \in \mathcal{Q}^*(V)\} \quad (26)$$

given $V := (s_{i+1}, \dots, s_{i+2n})$ being a subsequence of (s_1, \dots, s_{m-1}) ,

where we have used the short-hand notation $\mathcal{Q}^*(V)$ to denote $\mathcal{Q}^*(s_{i+1}, \dots, s_{i+2n})$. By saying q_V is a *linked component* of q , we mean $q_V \subset q$ is linked but not linked to $q \setminus q_V$. Each A_V is a collection of some unlinked diagrams since each of its elements q can be decomposed as $q = q_V \cup (q \setminus q_V)$ where the two subsets are not linked to each other. Note that it is sufficient to consider point sets V including successive time points s_{i+1}, \dots, s_{i+2n} since any unlinked diagram contains at least one linked component with only successive time points, and each V should contain at least four points due to the exclusion of the diagrams with arcs connecting adjacent points. For example when $m = 10$, we have

- For $|V| = 4$:

$$\begin{aligned} \mu(A_{\{s_1, s_2, s_3, s_4\}}) &= \text{diagram 1} + \text{diagram 2} + \text{diagram 3} \\ &\quad + \text{diagram 4} + \text{diagram 5} =: \boxed{\text{diagram 6}} \text{ ,} \\ \mu(A_{\{s_2, s_3, s_4, s_5\}}) &= \text{diagram 7} + \text{diagram 8} + \text{diagram 9} + \\ &\quad \text{diagram 10} + \text{diagram 11} + \text{diagram 12} =: \boxed{\text{diagram 13}} \text{ ,} \\ &\dots\dots\dots \\ \mu(A_{\{s_6, s_7, s_8, s_9\}}) &= \text{diagram 14} + \text{diagram 15} + \text{diagram 16} + \\ &\quad \text{diagram 17} + \text{diagram 18} + \text{diagram 19} =: \boxed{\text{diagram 20}} \text{ ;} \end{aligned} \quad (27)$$

- For $|V| = 6$:

$$\begin{aligned} \mu(A_{\{s_1, \dots, s_6\}}) &= \text{diagram 21} + \text{diagram 22} \\ &\quad + \text{diagram 23} + \text{diagram 24} =: \boxed{\text{diagram 25}} \text{ ,} \\ &\dots\dots\dots \\ \mu(A_{\{s_4, \dots, s_9\}}) &= \text{diagram 26} + \text{diagram 27} \\ &\quad + \text{diagram 28} + \text{diagram 29} =: \boxed{\text{diagram 30}} \text{ ;} \end{aligned} \quad (28)$$

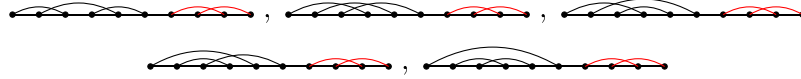
- For $|V| = 8$:

$$\mu(A_{\{s_2, \dots, s_9\}}) = \underbrace{\text{diagram 31} + \text{diagram 32} + \dots}_{27 \text{ linked diagrams within } s_2, \dots, s_9} =: \boxed{\text{diagram 33}} \text{ .} \quad (29)$$

All linked q_V (marked in red) in the corresponding A_V are eventually included in the rounded boxes which are calculated as $\mathcal{L}_b^c(V)$. The rest of points in $\{s_1, \dots, s_{10}\}$ are not linked to q_V and they build up all possible ordered pairings without any pair of adjacent two points. Therefore, we group these points in the rectangular boxes and compute them by $\mathcal{L}_b^*(\{s_1, \dots, s_{10}\} \setminus V)$. Note that some rectangular boxes may be divided by rounded boxes into several nonadjacent segments and we use ‘thin pumps’ to connect these segments above the rounded boxes to indicate that the points covered by these segments are in the same \mathcal{L}_b^* . Consequently, the diagrams on the right-hand side above are expressed as the following formulas:

$$\begin{aligned} \boxed{\text{diagram 34}} &= \mathcal{L}_b^c(s_1, s_2, s_3, s_4) \mathcal{L}_b^*(s_5, s_6, s_7, s_8, s_9, s_{10}), \\ \boxed{\text{diagram 35}} &= \mathcal{L}_b^c(s_2, s_3, s_4, s_5) \mathcal{L}_b^*(s_1, s_6, s_7, s_8, s_9, s_{10}), \\ \boxed{\text{diagram 36}} &= \mathcal{L}_b^c(s_6, s_7, s_8, s_9) \mathcal{L}_b^*(s_1, s_2, s_3, s_4, s_5, s_{10}), \\ \boxed{\text{diagram 37}} &= \mathcal{L}_b^c(s_1, s_2, s_3, s_4, s_5, s_6) \mathcal{L}_b^*(s_7, s_8, s_9, s_{10}), \\ \boxed{\text{diagram 38}} &= \mathcal{L}_b^c(s_4, s_5, s_6, s_7, s_8, s_9) \mathcal{L}_b^*(s_1, s_2, s_3, s_{10}), \\ \boxed{\text{diagram 39}} &= \mathcal{L}_b^c(s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9) \mathcal{L}_b^*(s_1, s_{10}). \end{aligned} \quad (30)$$

The union of A_V contains all the diagrams in S which are not linked. Note that in (29), we have neglected the diagram with eight-point rounded box $\mu(A_{\{s_1, \dots, s_8\}})$, which equals zero as its rectangular box is formed by the two adjacent points and thus $\mathcal{L}_b^*(s_9, s_{10}) = 0$. Moreover, in the definition of A_V in (23), we do not need to consider the case where V includes the last time point s_m . This is because, given any $V' = \{s_{m-2n+1}, \dots, s_m\}$, if we define $A_{V'}$ as the set of diagrams with a linked component including all the last $2n$ points, then each element in $A_{V'}$ can be found in at least one of the A_V 's defined in (23). For example, the five diagrams of $A_{\{s_7, s_8, s_9, s_{10}\}}$ given by



can be found in $A_{\{s_1, \dots, s_6\}}$ and $A_{\{s_2, s_3, s_4, s_5\}}$; the set $A_{\{s_5, \dots, s_{10}\}}$ has four elements which are all included in $A_{\{s_1, s_2, s_3, s_4\}}$. As a result, on the left-hand side of the inclusion–exclusion principle (8) we have

$$\mu \left(S \setminus \bigcup_V A_V \right) = \mu(\mathcal{Q}_c(s_1, \dots, s_m)) = \mathcal{L}_b^c(s_1, \dots, s_m). \quad (31)$$

On the right-hand side of (8), if the sets V_1, \dots, V_l are mutually disjoint, we have

$$A_{V_1} \cap \dots \cap A_{V_l} = \{q \in S \mid q \text{ has } l \text{ linked components } q_{V_i} \in \mathcal{Q}^*(V_i) \text{ for } i = 1, \dots, l\}; \quad (32)$$

if any V_i and V_j contain a common point s_i , then $A_{V_1} \cap \dots \cap A_{V_l} = \emptyset$, which has no contribution in the inclusion–exclusion principle. Thus, in the example of $m = 10$, the following intersections provide nonzero contribution:

$$\begin{aligned} \mu(A_{\{s_2, s_3, s_4, s_5\}} \cap A_{\{s_6, s_7, s_8, s_9\}}) &= \text{diagram} = \boxed{\text{diagram}} \\ &= \mathcal{L}_b^c(s_2, s_3, s_4, s_5) \mathcal{L}_b^c(s_6, s_7, s_8, s_9) \mathcal{L}_b^*(s_1, s_{10}), \\ \mu(A_{\{s_1, s_2, s_3, s_4\}} \cap A_{\{s_6, s_7, s_8, s_9\}}) &= \text{diagram} = \boxed{\text{diagram}} \\ &= \mathcal{L}_b^c(s_1, s_2, s_3, s_4) \mathcal{L}_b^c(s_6, s_7, s_8, s_9) \mathcal{L}_b^*(s_5, s_{10}). \end{aligned} \quad (33)$$

We have again neglected the intersection $\mu(A_{\{s_1, s_2, s_3, s_4\}} \cap A_{\{s_5, s_6, s_7, s_8\}})$ which contains the null-valued rectangular box $\mathcal{L}_b^*(s_9, s_{10})$. Now we insert (27)–(29) and (33) into (8) to get the diagrammatic representation for $\mathcal{L}_b^c(s_1, \dots, s_{10})$ using inclusion–exclusion principle as

$$\begin{aligned} \text{diagram} &= \boxed{\text{diagram}} - \boxed{\text{diagram}} - \boxed{\text{diagram}} \\ &- \boxed{\text{diagram}} - \boxed{\text{diagram}} - \boxed{\text{diagram}} - \boxed{\text{diagram}} \\ &- \boxed{\text{diagram}} - \boxed{\text{diagram}} - \boxed{\text{diagram}} - \boxed{\text{diagram}} \\ &- \boxed{\text{diagram}} + \boxed{\text{diagram}} + \boxed{\text{diagram}}. \end{aligned} \quad (34)$$

At this moment, we have obtained an indirect approach based on inclusion–exclusion principle to evaluate a given rounded box $\mathcal{L}_b^c(s_1, \dots, s_m)$: one may first expand a rounded box as (34) and then compute the (bridged) rectangular boxes using formula (11) (or more efficiently algorithm 1). Each rounded box on the right-hand side with length greater than 2 can again be evaluated by the same procedure. In the subsequent section, we will further optimize the computational complexity of the calculation by an improved algorithm.

3.2. Improved algorithm

The main idea of our improved algorithm is to combine the diagrams with the same rectangular boxes. Specifically, in the example (34), the first two diagrams in the last line have the same rectangular box, and they can be written together by the distributive law as $[\mathcal{L}_b^c(s_2, s_3, s_4, s_5) \mathcal{L}_b^c(s_6, s_7, s_8, s_9) - \mathcal{L}_b^c(s_2, \dots, s_9)] \mathcal{L}_b^*(s_1, s_{10})$. For simplicity of notations, we define the dotted box:

$$\begin{aligned} \text{dotted box} &:= \text{diagram} - \text{diagram} \\ &= \mathcal{L}_b^c(s_2, s_3, s_4, s_5) \mathcal{L}_b^c(s_6, s_7, s_8, s_9) - \mathcal{L}_b^c(s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9). \end{aligned} \quad (35)$$

In general, a dotted box with even time points represents the sum of all partitions of these time points by rounded boxes with length greater than or equal to 4, and the sign of each term depends on the number of

$$\begin{aligned}
\mathcal{L}_b^d(s_1, \dots, s_m) = & -\mathcal{L}_b^c(s_1, \dots, s_m) + \underbrace{\mathcal{L}_b^c(s_5, \dots, s_m)}_{\text{length}=m-4, i_j=4} (-1)^2 \mathcal{L}_b^c(s_1, s_2, s_3, s_4) + \dots + \\
& + \underbrace{\mathcal{L}_b^c(s_{m-5}, \dots, s_m)}_{\text{length}=6, i_j=m-6} \sum_{j=1}^{\lfloor \frac{m-6}{4} \rfloor} (-1)^{j+1} \sum_{\substack{i_1=4 \\ i_1 \text{ is even}}}^{m-4} \dots \\
& \times \sum_{\substack{i_{j-1}=i_{j-2}+4 \\ i_{j-1} \text{ is even}}}^{m-4} \mathcal{L}_b^c(s_1, \dots, s_{i_1}) \dots \mathcal{L}_b^c(s_{i_{j-1}+1}, \dots, s_{m-6}) \\
& + \underbrace{\mathcal{L}_b^c(s_{m-3}, \dots, s_m)}_{\text{length}=4, i_j=m-4} \sum_{j=1}^{\lfloor \frac{m-4}{4} \rfloor} (-1)^{j+1} \sum_{\substack{i_1=4 \\ i_1 \text{ is even}}}^{m-4} \dots \\
& \times \sum_{\substack{i_{j-1}=i_{j-2}+4 \\ i_{j-1} \text{ is even}}}^{m-4} \mathcal{L}_b^c(s_1, \dots, s_{i_1}) \dots \mathcal{L}_b^c(s_{i_{j-1}+1}, \dots, s_{m-4}).
\end{aligned}$$

By decreasing the index j in each term by 1, one may easily check these multiple summations will coincide the definition (37) for shorter dotted boxes, i.e.,

$$\begin{aligned}
\mathcal{L}_b^d(s_1, \dots, s_m) = & -\mathcal{L}_b^c(s_1, \dots, s_m) - \mathcal{L}_b^c(s_5, \dots, s_m) \mathcal{L}_b^d(s_1, s_2, s_3, s_4) - \dots - \\
& - \mathcal{L}_b^c(s_{m-5}, \dots, s_m) \mathcal{L}_b^d(s_1, \dots, s_{m-6}) - \mathcal{L}_b^c(s_{m-3}, \dots, s_m) \mathcal{L}_b^d(s_1, \dots, s_{m-4}),
\end{aligned}$$

which proves (39). \square

By comparing the number of diagrams that need to be summed up for a rounded box in the example (34) with that for a dotted box in (38), one can easily see that computing a dotted box for a large m using the above iterative method is even cheaper than computing a rounded box of the same size. Later in section 3.3, we will carry out a complexity analysis on the computational cost of these dotted boxes in the entire algorithm. Now we are ready to formulate the expansion of rounded boxes $\mathcal{L}_b^c(s_1, \dots, s_m)$ for an arbitrary even m using only dotted and rectangular boxes, and propose an optimized algorithm to compute the rounded boxes.

3.2.2. Inclusion–exclusion principle with optimization for computing rounded boxes

The example in the previous section suggests that a rounded box is computed by the summation of all possible diagrams filled up by the nonadjacent dotted boxes covering at least four points excluding the right end and a rectangular box covering rest of the time points. The formula is provided in the following theorem:

Theorem 2. Given the increasing time sequence (s_1, \dots, s_m) with $m \geq 4$ being an even number, we have

$$\begin{aligned}
\mathcal{L}_b^c(s_1, \dots, s_m) = & \mathcal{L}_b^*(s_1, \dots, s_m) + \sum_{\substack{1 \leq i_1 < j_1 \leq m \\ j_1 - i_1 \geq 4 \text{ and is even}}} \mathcal{L}_b^d(s_{i_1}, \dots, s_{j_1-1}) \mathcal{L}_b^*(\text{rest of points}) \\
& + \sum_{\substack{1 \leq i_1 < j_1 < i_2 < j_2 \leq m \\ j_1 - i_1 \geq 4 \text{ and is even} \\ j_2 - i_2 \geq 4 \text{ and is even}}} \mathcal{L}_b^d(s_{i_1}, \dots, s_{j_1-1}) \mathcal{L}_b^d(s_{i_2}, \dots, s_{j_2-1}) \mathcal{L}_b^*(\text{rest of points}) \\
& + \dots \\
& + \sum_{\substack{1 \leq i_1 < j_1 < \dots < i_k < j_k \leq m \\ j_1 - i_1 \geq 4 \text{ and is even} \\ \dots \\ j_k - i_k \geq 4 \text{ and is even}}} \mathcal{L}_b^d(s_{i_1}, \dots, s_{j_1-1}) \dots \mathcal{L}_b^d(s_{i_k}, \dots, s_{j_k-1}) \mathcal{L}_b^*(\text{rest of points}),
\end{aligned} \tag{40}$$

where $k = \lfloor \frac{m}{5} \rfloor$. The ‘rest of points’ denotes all time points in (s_1, \dots, s_m) which do not occur in the brackets of any \mathcal{L}_b^d in the same summand.

We remark that on the right-hand side of (40), the number of dotted boxes $\mathcal{L}_b^d(\dots)$ in each summand does not exceed $k = \lfloor \frac{m}{5} \rfloor$ since all dotted boxes are pairwise nonadjacent and each of them includes at least

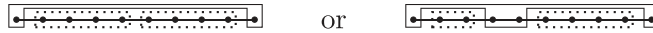
Algorithm 2. Inclusion–exclusion principle with optimization.

```

1: for  $i$  from 1 to  $m - 4$  do ▷ Initial setting
2:    $\mathcal{L}_b^c(s_i, s_{i+1}, s_{i+2}, s_{i+3}) \leftarrow B(s_i, s_{i+2})B(s_{i+1}, s_{i+3})$ 
3:    $\mathcal{L}_b^d(s_i, s_{i+1}, s_{i+2}, s_{i+3}) \leftarrow -B(s_i, s_{i+2})B(s_{i+1}, s_{i+3})$ 
4: end for
5: for  $n$  from 3 to  $\frac{m}{2} - 1$  do
6:   for  $k$  from 1 to  $m - 2n$  do ▷ Compute  $k$ th rounded and dotted segment with length  $2n$ 
7:     compute  $\mathcal{L}_b^c(s_k, \dots, s_{k+2n-1})$  according to (40) where each
        $\mathcal{L}_b^*$ (rest of points) is computed according to algorithm 1
8:     compute  $\mathcal{L}_b^d(s_k, \dots, s_{k+2n-1})$  according to (39)
9:   end for
10: end for
11: compute  $\mathcal{L}_b^c(s_1, \dots, s_m)$  according to (40) ▷ Final step
12: return  $\mathcal{L}_b^c(s_1, \dots, s_m)$ 

```

four points. For example, the diagrams



are not allowed.

Now we arrive at an optimized algorithm based on inclusion–exclusion principle to calculate a rounded box $\mathcal{L}_b^c(s_1, \dots, s_m)$: one writes the rounded box as the expansion (36) using theorem 2 and then apply algorithm 1 to calculate the rectangular part and lemma 1 for the dotted segments. To avoid repeated calculations caused by recursion, one should compute all rounded segments from short to long until the entire rounded box is obtained. Such procedures in general are described by algorithm 2.

We have now finished the implementation of inclusion–exclusion principle for computing the functional $\mathcal{L}_b^c(s_1, \dots, s_m)$. Similar as computing the bath influence functional, inclusion–exclusion principle for the rounded box is less efficient than the direct summation of all linked diagrams for a small m . However, our complexity analysis in the next section will show that the new algorithm will outperform the direct method as m becomes large. The central idea is that, when m increases, the number of diagrams in (36) will grow significantly slower than double factorial (the growth rate of the number of diagrams in the direct method).

The proposed algorithm in this section can be regarded as the bosonic version of the algorithm introduced in [8] for fermions. We have also further improved the algorithm by a more efficient scheme to compute the dotted boxes (section 3.2.1).

3.3. Complexity analysis

In this section, we will show that the computational complexity of algorithm 2 based on inclusion–exclusion principle is significantly smaller than double factorial, which is the growth rate of the direct summation over all linked diagrams.

We denote the complexities of evaluating $2n$ -point rounded and dotted segment respectively by $C_{rd}(2n)$ and $C_{dt}(2n)$. The total computational cost for $\mathcal{L}_b^c(s_1, \dots, s_m)$ using algorithm 2 can be immediately written down as

$$C_{\text{opt}}(m) = \sum_{n=2}^{m/2-1} (m-2n) \cdot (C_{rd}(2n) + C_{dt}(2n)) + C_{rd}(m), \quad (41)$$

where the last term above refers to the cost of the final step in line 11 of algorithm 2. Based on the previous calculations on the shorter segments, the computational complexity of a dotted box $C_{dt}(2n)$ is simply given by

$$C_{dt}(2n) = \underbrace{n-3}_{\text{additions}} + \underbrace{n-3}_{\text{multiplications}} = O(n) \quad (42)$$

according lemma 1. Therefore, we focus on the estimation of the complexity of rounded segments $C_{rd}(2n)$.

Inspired by the example (36), the computational cost for $\mathcal{L}_b^c(s_k, \dots, s_{k+2n-1})$ in line 7 using theorem 2 can be estimated as

$$C_{rd}(2n) \lesssim \sum_{k=0}^{n-1} a_{2n,2k} \cdot (C_b(2n-2k) + k + 1). \quad (43)$$

In the estimation above, $a_{2n,2k}$ is the number of diagrams where the total length of the dotted boxes is $2k$. $C_b(2n-2k)$ is the computational cost of a rectangular box with length $2n-2k$, which is at $O(2^{2n-2k})$ as we have discussed at the end of section 2.2. ‘ k ’ and ‘1’ respectively counts the multiplications used among the

rectangular part and dotted boxes within each diagram, and the addition between every two diagrams. For example, the last two terms in the right-hand side of (36) read

whose computational cost contributes to the $k = 4$ term in the summation (43) for $C_{rd}(10)$. In each of the $a_{10,8} = 2$ diagrams above, we compute a two-point rectangular box and need at most two multiplications (for the second diagram). We further claim that $a_{2n,2} = 0$ since a two-point dotted box including two adjacent points always has zero value, and thus there will not exist any diagram containing a two-point dotted segment in inclusion–exclusion expansion of a given rounded box.

At this point, we only need to focus on the estimation for $a_{p,2k}$ (p can be odd), which essentially is the number of nonadjacent partitions over the integers from 1 to $p - 1$ (the last point is excluded), where each dotted segment covers at least four points and the total length of all dotted segments is $2k$. The following statement provides a useful recurrence relation for the sequence $\{a_{p,2k}\}$:

Lemma 2. Given integers $p \geq 1$ and $0 \leq k \leq \lfloor \frac{p-1}{2} \rfloor$, the sequence $\{a_{p,2k}\}$ satisfies the recurrence relation

$$a_{p,2k} = a_{p-1,2k} + (a_{p-5,2k-4} + a_{p-7,2k-6} + a_{p-9,2k-8} + \cdots + a_{p-(2k-3),4}) + 1. \quad (44)$$

Proof. For simplicity, we consider the diagrams with dotted boxes only, since the rectangular boxes automatically include all the remaining points. Let the diagram

(45)

denote the set including all diagrams with a total number of $2k$ points inside an arbitrary number of non-adjacent dotted boxes in the shaded area (each dotted box must contain at least 4 points), and let the diagram

(46)

be the set of diagrams that add one dotted box as indicated to each of the diagrams in (45). For example,

Then $a_{p,2k}$ is the cardinality of the diagram set (45) or (46) if there are $p - 1$ points in the shaded area. Note that in the definitions of (45) and (46), the rightmost point in any diagram is never included in the dotted boxes, and in (46), the one point between the shaded area and the dotted box ensures that any two dotted boxes are non-adjacent. By this definition, we claim that

(47)

where all diagrams have the same length p . Since all the sets on the right-hand side are disjoint (which can be observed by focusing on the last dotted box of the diagrams), we immediately get (44) by counting the number of diagrams on both sides.

To show (47), we can take any diagram on the left-hand side, and check the status of the last second point:

- If the last second point is not included in any dotted boxes, then the diagram must belong to the first set on the right-hand side of (47);
- If the last second point is included in a $2n$ -point dotted box ($1 < n < k - 1$), then the diagram must belong to the n th set on the right-hand side of (47);

- If the last second point is included in a $2k$ -point dotted box, then this diagram must be the one in the last line of (47).

It is now clear that the left-hand side of (47) is a subset of its right-hand side. For the reverse direction, the right-hand side is obviously a subset of the left-hand side since any diagram in any set on the right-hand side of (47) has in total $2k$ points in dotted boxes, and the last point is never included into any boxes. This completes the proof of the lemma. \square

With the recurrence relation for the sequence $\{a_{p,2k}\}$, we now consider a two-variable generating function $f(x, y)$ with a Maclaurin expansion as

$$f(x, y) = \sum_{p=0}^{\infty} \sum_{q=0}^p a_{p,p-q} \cdot x^p y^q, \quad (48)$$

where we set $a_{p',0} = 1$ and $a_{p',q'} = 0$ for any nonnegative p' and odd q' . By (44), we have

$$\sum_{p=1}^{\infty} \sum_{q=1}^{\infty} [a_{p,p-q} - a_{p-1,p-q} - (a_{p-5,p-q-4} + a_{p-7,p-q-6} + \dots)] x^p y^q = 0$$

leading to

$$(f(x, y) - a_{0,0}) - xyf(x, y) - (x^5y + x^7y + \dots)f(x, y) = 0$$

and thus we obtain the explicit expression of the generating function:

$$f(x, y) = \frac{1}{1 - \left(xy + x^5y \frac{1}{1-x^2}\right)}. \quad (49)$$

Now we return to the estimation for the complexity (43), which can be further bounded by

$$\begin{aligned} C_{\text{rd}}(2n) &\lesssim \sum_{k=0}^{n-1} a_{2n,2k} \cdot (2^{2n-2k} + k + 1) \\ &\leq \sum_{k=0}^n a_{2n,2n-2k} \cdot 2^{2k} + n \cdot \sum_{k=0}^n a_{2n,2n-2k}. \end{aligned} \quad (50)$$

By (48), we see that

$$f(x, 2) = \sum_{p=0}^{\infty} \left(\sum_{q=0}^p a_{p,p-q} \cdot 2^q \right) x^p = \sum_{p=0}^{\infty} \left(\sum_{k=0}^{\lfloor p/2 \rfloor} a_{p,p-2k} \cdot 2^{2k} \right) x^p,$$

which shows that in the Maclaurin expansion of $f(x, 2)$, the coefficient of x^{2n} equals the first summation in the second line of (50). According to (49), the function $f(x, 2)$ is a rational function, so that its Maclaurin expansion can be found via the Heaviside cover-up method [40]:

$$f(x, 2) = \frac{x^2 - 1}{2x^5 - 2x^3 + x^2 + 2x - 1} = \sum_{i=1}^5 \frac{c_i}{x - x_i} = \sum_{i=1}^5 \left(-\frac{c_i}{x_i} \right) \sum_{j=0}^{\infty} \left(\frac{1}{x_i} \right)^j x^j,$$

where $x_1 \approx 0.470417$, $x_{2,3} \approx -0.970009 \pm 0.4461i$ and $x_{4,5} \approx 0.7348 \pm 0.62649i$ are the poles of $f(x, 2)$ and c_i are some constants. Therefore, asymptotically we have

$$\sum_{k=0}^n a_{2n,2n-2k} \cdot 2^{2k} = -\sum_{i=1}^5 \frac{c_i}{x_i^{2n+1}} \sim O \left(\max_{i=1,\dots,5} \left(\left| \frac{1}{x_i} \right|^{2n} \right) \right) = O \left(\left| \frac{1}{x_1} \right|^{2n} \right) \approx O(2.12577^{2n}).$$

Similarly, the second summation in the last line of (50) is the coefficient of x^{2n} in the Maclaurin expansion of $f(x, 1)$ and we can deduce that $\sum_{k=0}^n a_{2n,2n-2k} \sim O(1.44327^{2n})$. Hence,

$$C_{\text{rd}}(2n) \lesssim 2.12577^{2n} + n \cdot 1.44327^{2n} \sim O(\alpha^n) \quad \text{with } \alpha \approx 4.51891.$$

Afterwards, we insert the above upper bound back into (41) to obtain the overall complexity $C_{\text{opt}}(m)$ of algorithm 2:

$$\begin{aligned} C_{\text{opt}}(m) &\lesssim \left(\sum_{n=2}^{m/2-1} (m-2n) \cdot (\alpha^n + n) \right) + \alpha^{m/2} \\ &= \frac{\left(\frac{m}{2}\right)^3 (\alpha-1)^2 - \frac{m}{2} (6\alpha^3 + \alpha^2 - 14\alpha + 7) + 6(\alpha^{m/2+1} + \alpha^3 - \alpha^2 - 2\alpha + 1)}{3(\alpha-1)^2} + \alpha^{m/2}. \end{aligned}$$

The fraction in the last line above is asymptotically $O(\alpha^{m/2-1})$ and thus the second term dominates the upper bound. Such estimation indicates that the major computational cost of the algorithm is spent on the rectangular boxes in the final step (line 11) when calculating the longest rounded box. To summarize the analysis, we state the conclusion in the theorem below:

Theorem 3. *Given the increasing time sequence (s_1, \dots, s_m) with m being an even number, the complexity of algorithm 2 computing the entire rounded box $\mathcal{L}_b^c(s_1, \dots, s_m)$ can be bounded by*

$$C_{\text{opt}}(m) \lesssim \alpha^{m/2} \quad \text{with } \alpha \approx 4.51891. \quad (51)$$

Compared to the direct method whose computational cost grows as fast as double factorial in m , the inclusion–exclusion principle based algorithm with exponential growth rate is obviously more efficient when m is large.

4. Numerical experiments

In this section, we will first numerically verify the statements on the complexities of algorithms, and then simulate both bare dQMC and inchworm Monte Carlo method to see how we can benefit from the inclusion–exclusion principle in applications.

We consider the spin-boson model [19, 23, 41] where the Hamiltonian and perturbation operators associated to the system are

$$H_s = \epsilon \hat{\sigma}_z + \Delta \hat{\sigma}_x, \quad W_s = \hat{\sigma}_z,$$

where $\hat{\sigma}_x$ and $\hat{\sigma}_z$ are the usual Pauli matrices

$$\hat{\sigma}_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \hat{\sigma}_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

The observable of interest is set to be $O = \hat{\sigma}_z \otimes \text{Id}_b$, which meets the condition that O only acts on the system space. The initial density matrix $\rho = \rho_s \otimes \rho_b$ is given by

$$\rho_s = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad \text{and} \quad \rho_b = Z^{-1} \exp(-\beta H_b),$$

where Z is a normalizing factor chosen such that $\text{tr}(\rho_b) = 1$.

Assume a Ohmic spectral density, the two-point correlation function is formulated as

$$B(\tau_1, \tau_2) = \sum_{l=1}^L \frac{c_l^2}{2\omega_l} \left[\coth\left(\frac{\beta\omega_l}{2}\right) \cos(\omega_l \Delta\tau) - i \sin(\omega_l \Delta\tau) \right],$$

where $\Delta\tau$ is the time difference on the Keldysh contour defined as

$$\Delta\tau = \begin{cases} \tau_2 - \tau_1, & \text{if } \tau_1 \leq \tau_2 < t, \\ \tau_1 - \tau_2, & \text{if } t \leq \tau_1 \leq \tau_2, \\ 2t - \tau_1 - \tau_2, & \text{if } \tau_1 < t \leq \tau_2. \end{cases}$$

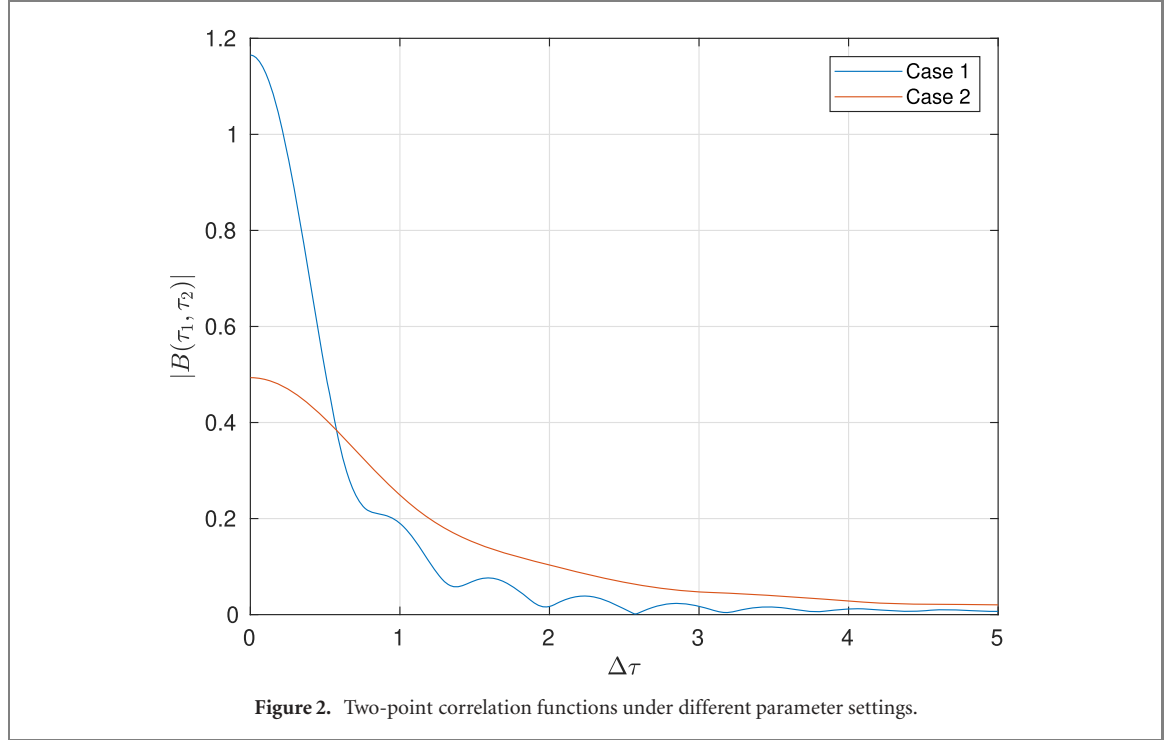
and the coupling intensity c_l and frequency of each harmonic oscillator ω_l are given by

$$c_l = \omega_l \sqrt{\frac{\xi \omega_c}{L} [1 - \exp(-\omega_{\text{max}} / \omega_c)]}, \quad \omega_l = -\omega_c \ln \left(1 - \frac{l}{L} [1 - \exp(-\omega_{\text{max}} / \omega_c)] \right), \quad l = 1, \dots, L.$$

In our experiments, we will study two examples with the parameter settings listed below in table 1. As one can observe from figure 2, $B(\tau_1, \tau_2)$ under the two parameter settings both decay to zero for large time

Table 1. Parameter settings for spin-boson model.

Parameters	Case 1	Case 2
Kondo parameter, ξ	0.4	0.1
Inverse temperature, β	5	0.2
Primary frequency, ω_c	2.5	1
Maximum frequency, ω_{\max}	4	4
Energy difference, ϵ	1	1
Frequency of spin flipping, Δ	1	1
Number of modes, L	400	400

**Figure 2.** Two-point correlation functions under different parameter settings.

difference, which guarantees the convergence of the Dyson series as well as the infinite series in the inchworm integro-differential equation (18). In case 2, the decay of $|B(\tau_1, \tau_2)|$ is slower than case 1, leading to a slower convergence of the Dyson series and the inchworm series. It can then be expected that larger m needs to be included in the simulation of case 2.

4.1. Numerical experiments for computational complexity

In this section, we compare the wall clock time on evaluating given $\mathcal{L}_b(s_1, \dots, s_m)$ and $\mathcal{L}_b^c(s_1, \dots, s_m)$ using direct summation and algorithms 1 and 2 based on the inclusion–exclusion principle. The experiments are carried out using MATLAB on Intel Xeon CPU X5650 and the results for the time consumed may vary for different hardware, programming languages and implementation details. Since the operation counts do not depend on the value of $B(\tau_1, \tau_2)$, we will only use the parameters for case 1 in our test throughout this section.

The computational time for a various choice of m is plotted in figure 3. In the left panel, we compare direct method with algorithm 1 for computing a given bath influence functional. As predicted, the efficiencies of the two algorithms are comparable for small order m . Starting from $m = 12$, however, due to the double factorial growth in complexity, the time cost for the direct method becomes obviously larger than the inclusion–exclusion principle whose growth rate is only exponential as $O(2^m)$ according to our discussion at the end of section 2. The right panel of figure 3 compares algorithm 2 with the direct summation over the linked diagrams to compute a rounded box. Algorithm 2 outperforms the direct method when $m \geq 16$. The dotted line represents our estimation of the growth rate $O(\alpha^{m/2})$. We can observe that the curve of the inclusion–exclusion principle gradually becomes parallel to the dotted line, as indicates that our estimation of the computational complexity is sharp.

We would also like to discuss the memory cost of the algorithms. The direct method is out of memory for our machine once entering the red region (i.e., $m > 20$) and thus the results are not presented. In our simulation, to implement the direct method efficiently, we first generate all the linked diagrams and store

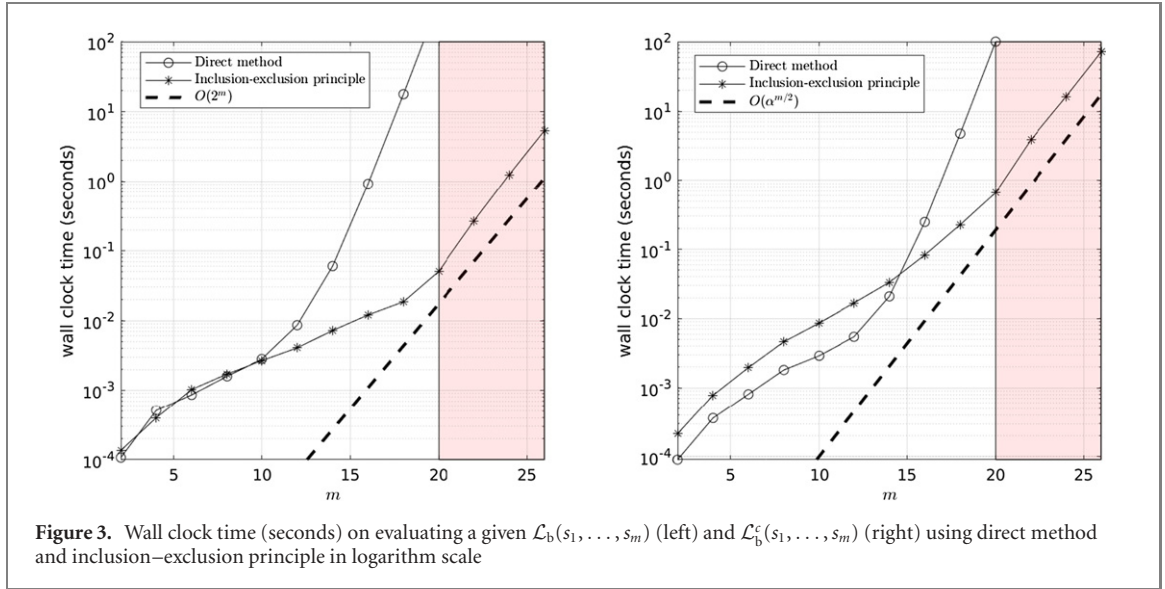


Figure 3. Wall clock time (seconds) on evaluating a given $\mathcal{L}_b(s_1, \dots, s_m)$ (left) and $\mathcal{L}_b^c(s_1, \dots, s_m)$ (right) using direct method and inclusion–exclusion principle in logarithm scale

their configurations in the memory, so that the computational time presented in figure 3 can be minimized. To store the diagrams, we use a matrix of size $A_m \times m$ to record the indices of the time sequence in all linked diagrams. Here A_m represents the number of diagrams, and m denotes the length of the rounded box. For example, all the diagrams included in $\mathcal{L}_b^c(s_1, s_2, s_3, s_4, s_5, s_6)$ (see (21)) are stored in the following 4×6 matrix

$$\begin{pmatrix} 1 & 3 & 2 & 5 & 4 & 6 \\ 1 & 4 & 2 & 5 & 3 & 6 \\ 1 & 4 & 2 & 6 & 3 & 5 \\ 1 & 5 & 2 & 4 & 3 & 6 \end{pmatrix},$$

where each row of the matrix describes the pairing of time points (arcs) in one diagram on the right-hand side of (21). However, the size of this matrix grows quickly as m increases due to the double factorial growth of the number of diagrams. For example, when the length of a rounded box reaches $m = 22$, the size of matrix turns out to be $434\,226\,3000 \times 22$. Even if we use the uint8 data type in MATLAB (the smallest unsigned integer type that takes only one byte) to store the matrix, the total memory cost is around 89G, which is beyond the capacity of most machines. A workaround is to further compress the matrix using more compact storage patterns, or generate the diagrams during the summation. Both approaches will cause additional operations so that the computation may be further slowed down.

As a comparison, the major memory cost for inclusion–exclusion principle concentrates in the temporary storage of complex-valued $Q_{k_1 k_2 \dots k_5}$ and $R_{k_1 k_2 \dots k_5}$ appearing in theorem 1, which grows only as an exponential and the memory requirement is at most $16 \times \frac{2^{m+1}}{1000^3}$ G (only 0.1344 G for $m = 22$). Here 16 refers to the number of bytes for a double-precision complex number, and 2^{m+1} refers to the total number of entries in $Q_{k_1 k_2 \dots k_5}$ and $R_{k_1 k_2 \dots k_5}$. As a result, a longer diagram can be computed using the algorithm based on the inclusion–exclusion principle. Such a memory issue for the direct method also exists when computing $\mathcal{L}_b(s_1, \dots, s_m)$ since the number of diagrams given by a bath influence functional is even larger than that in a rounded box of the same size.

4.2. Numerical simulations for open quantum systems

We now implement several numerical simulations on the observable $\langle \hat{\sigma}_z(t) \rangle$ using bare dQMC and inchworm Monte Carlo method respectively, in which inclusion–exclusion principle will be used to evaluate large rectangular and rounded boxes. We would like to check if and how frequently we will encounter the scenarios when the order m has to be chosen very large during the simulations to show the necessity of exploiting the inclusion–exclusion principle.

4.2.1. Numerical methods

We first introduce the numerical methods for our simulations. In particular, we will discuss the details of the implementation of bare dQMC. For the more complicated inchworm equation, we only provide our Monte Carlo sampling method, which is novel in this work. One may refer to [10, 11] for the general framework of the full implementation.

In the Dyson series (3), m should be chosen as even due to the Wick's theorem for the bath influence functional. Moreover, the $m = 0$ term in the Dyson series does not contain any time points and thus no

Monte Carlo sampling is needed when applying bare dQMC. Therefore, we may take out this term and rewrite (3) as

$$\begin{aligned} \langle O(t) \rangle &= \text{tr}(\rho_s e^{itH_s} O_s e^{-itH_s}) + \sum_{\substack{m=2 \\ m \text{ is even}}}^{+\infty} i^m \int_0^{2t} ds_m \int_0^{s_m} ds_{m-1} \dots \int_0^{s_2} ds_1 \\ &\quad \times (-1)^{\#\{s < t\}} \text{tr}_s(\rho_s \mathcal{U}^{(0)}(0, s_1, \dots, s_m, 2t)) \cdot \mathcal{L}_b(s_1, \dots, s_m). \end{aligned}$$

To approximate the infinite series in the above formula using Monte Carlo integration, we need sample

- A positive even number m ;
- A sequence of times: $0 \leq s_1 \leq s_2 \leq \dots \leq s_m \leq 2t$.

Once m is chosen, the time sequence (s_1, s_2, \dots, s_m) can be generated by drawing a sample from the uniform distribution $U([0, 2t]^m)$ and then sorting the sequence. In our previous works [10, 11], instead of sampling the even number m , we simply truncated the series (3) at $m = \bar{M}$ and use the same number of samples for each m . In the current paper, we would propose a heuristic approach to take samples of m . Ideally, the probability of m should be proportional to the absolute value of the integral in (3). Since such a function is not available, we make the following approximations:

- Ignore the term $\text{tr}_s(\dots)$ representing the system part;
- Use the uniform distribution of s_1, \dots, s_m to represent the value of $\mathcal{L}_b(s_1, \dots, s_m)$ in all cases.

Thus the distribution of m becomes

$$\begin{aligned} \mathbb{P}_t(m = 2M) &= \frac{1}{\lambda_0} \int_0^{2t} ds_{2M} \int_0^{s_{2M}} ds_{2M-1} \dots \int_0^{s_2} ds_1 |\mathcal{L}_b(\tau, 2\tau, \dots, (2M-1)\tau, 2M\tau)| \\ &= \frac{(2t)^{2M}}{\lambda_0 (2M)!} \cdot |\mathcal{L}_b(\tau, 2\tau, \dots, (2M-1)\tau, 2M\tau)| \quad \text{for } M = 1, 2, \dots, M_{\max}, \end{aligned} \quad (52)$$

where $\tau = \frac{2t}{2M+1}$ and λ_0 is given such that the normalization $\sum_{M=1}^{M_{\max}} \mathbb{P}_t(m = 2M) = 1$ holds. Here we set M_{\max} to be the maximum value of M in order to prevent m from being too large, which may cause unnecessary huge computational cost in the evaluation of the bath influence functional. Thereafter, the bare dQMC approximates the observable $\langle \hat{\sigma}_z(t) \rangle$ as

$$\begin{aligned} \langle \hat{\sigma}_z(t) \rangle &\approx \text{tr}(\rho_s e^{itH_s} \hat{\sigma}_z e^{-itH_s}) + \frac{1}{N_s} \sum_{j=1}^{N_s} \frac{(2t)^{m^{(j)}}}{(m^{(j)})!} \times (\mathbb{P}_t(m = m^{(j)}))^{-1} \times (-1)^{\#\{s^{(j)} < t\}} \\ &\quad \times \text{tr}_s(\rho_s \mathcal{U}^{(0)}(0, s_1^{(j)}, \dots, s_{m^{(j)}}^{(j)}, 2t)) \cdot \mathcal{L}_b(s_1^{(j)}, \dots, s_{m^{(j)}}^{(j)}) \quad \text{for } m^{(j)} \sim \text{i.i.d.} \mathbb{P}_t, \end{aligned} \quad (53)$$

where N_s is the number of samples, and the quantities with superscript (j) denote the j th sample.

In order to study the evolution of the observable $\langle \hat{\sigma}_z(t) \rangle$ in the time interval $[0, T]$, we will need to compute all $\langle \hat{\sigma}_z(nh) \rangle$ for $n = 1, 2, \dots, T/h$ given the time step h (the initial value is $\langle \hat{\sigma}_z(0) \rangle = \text{tr}(\hat{\sigma}_z \rho_s) = 1$ according to the definition (2)). Therefore, we need to first generate \mathbb{P}_t for all $t = h, 2h, \dots, T$, which requires the calculation of long \mathcal{L}_b including up to $2M_{\max}$ time points. This can be time-consuming when the time step h is small, and it is also unnecessary for short time simulations where a large $m^{(j)}$ is unlikely to be sampled. To improve the efficiency of simulations, we consider a more accessible distribution to approximate \mathbb{P}_t . In (52), we insert the definition of the bath influence functional and reach to

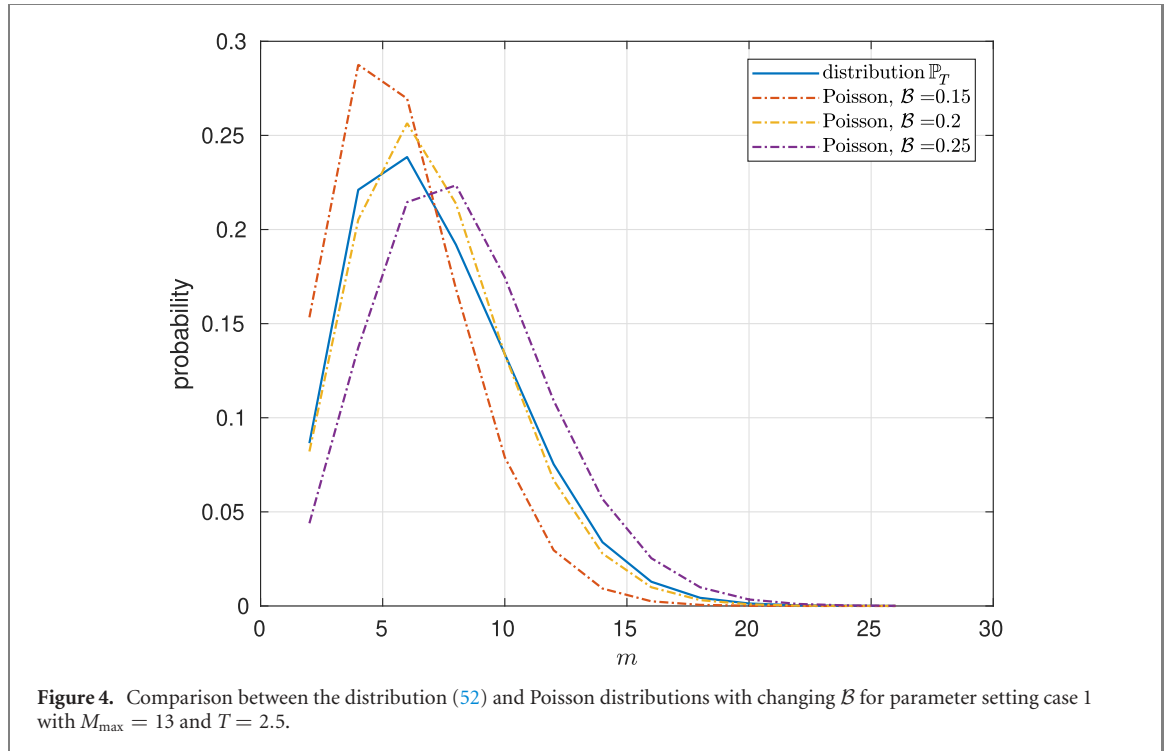
$$\mathbb{P}_t(m = 2M) = \frac{(2t)^{2M}}{\lambda_0 (2M)!} \cdot \sum_{q \in \mathcal{Q}(s)} \prod_{(s_j, s_k) \in q} B(s_j, s_k) = \frac{(2\mathcal{B}t^2)^M}{\lambda_0 M!},$$

where the constant \mathcal{B} is some average of the two-point correlation. Inspired by this formulation, we set \mathcal{B} to be a constant and choose the probability mass function to be $\tilde{\mathbb{P}}_t(m = 2M) = \lambda_1^{-1} (2\mathcal{B}t^2)^M / M!$, where λ_1 is chosen such that the normalization $\sum_{M=1}^{M_{\max}} \tilde{\mathbb{P}}_t(m = 2M) = 1$ holds. Thus each sample $m^{(j)}$ can be drawn based on the Poisson distribution. More precisely, we sample m according to

$$\frac{m^{(j)}}{2} - 1 \sim \text{Pois}(2\mathcal{B}t^2) \text{ and } \begin{cases} \text{accept } m^{(j)}, & \text{if } m^{(j)} \leq 2M_{\max}, \\ \text{reject } m^{(j)}, & \text{if } m^{(j)} > 2M_{\max}. \end{cases} \quad (54)$$

Note that the ‘ -1 ’ is needed on the left-hand side above since a standard Poisson distribution samples nonnegative integers from 0 while M begins with 1.

It remains only to set a suitable value for $\mathcal{B} \in (0, \max |B|)$. Here we simply select \mathcal{B} such that the probability mass function of m is close to (52) for $t = T$. For example in figure 4, one can compare the three



probability mass functions of Poisson distributions with different \mathcal{B} for the numerical example case 1 to see that the yellow dashed-dotted line gives a satisfactory approximation to \mathbb{P}_t . Poisson distributions with some other \mathcal{B} are plotted as references, which are comparatively far away from the target blue line. Therefore, we set $\mathcal{B} = 0.2$ for the Poisson distribution in case 1.

To end this section, we provide a brief discussion on the key procedures of the sampling method in the implementation of the inchworm Monte Carlo method. In (18), the partial derivative $\partial/\partial s_f$ on the left-hand side is discretized by a certain time integrator such as Heun's method. As for the right-hand side, similar to the bare dQMC, we need to approximate the infinite series by sampling an even number m and the time sequence (s_1, \dots, s_{m-1}) at every time step. The time sequence is again sampled according to the uniform distribution $U([s_i, s_f]^m)$, and the probability mass function of m is analogous to (52):

$$\mathbb{P}_t(m = 2M) = \frac{(2t)^{2M-1}}{\lambda_0'(2M-1)!} \cdot |\mathcal{L}_b^c(\tau, 2\tau, \dots, (2M-1)\tau, 2M\tau)| \quad \text{for } M = 1, 2, \dots, M_{\max}, \quad (55)$$

where $\tau = \frac{t}{M}$. To avoid the expensive computations of the long rounded boxes \mathcal{L}_b^c , we also sample each $m^{(i)}$ applying the Poisson distribution (54) in practice, where the choice of \mathcal{B} is subject to a satisfactory approximation to the distribution (55), which is set to be $\mathcal{B} = 0.2$ and $\mathcal{B} = 0.3$ for case 1 and case 2, respectively. We refer the readers to figure 5 for a comparison between the Poisson distribution and the distribution (55) for $t = T$.

4.2.2. Numerical results

With the numerical methods introduced, we are now ready to present the results of our simulations on the time evolution of the observable $\langle \hat{\sigma}_z(t) \rangle$. We are particularly interested in the convergence of $\langle \hat{\sigma}_z(t) \rangle$ computed by both bare dQMC and inchworm method w.r.t. the order m . Specifically, we first perform the simulation with the series in (17) or (18) truncated at $m = \bar{M}$, and plot the real part of $\langle \hat{\sigma}_z(t) \rangle$ up to $T = 2.5$. Note that due to the numerical error, the computed $\langle \hat{\sigma}_z(t) \rangle$ may contain a nonzero imaginary part. We hope to observe the convergence of these results to the numerical solution using our approach introduced in section 4.2.1, which justifies our numerical method. In our simulation, the Poisson distribution is truncated at $M_{\max} = 13$, so that the maximum value of m is 26.

Figure 6 plots the numerical results for parameter setting case 1 using bare dQMC. We set the time step to be $h = 0.1$ and compute $\langle \hat{\sigma}_z(nh) \rangle$ for each $n = 1, \dots, 25$. For the result with m sampled by Poisson distribution, each $\langle \hat{\sigma}_z(nh) \rangle$ is calculated based on $N_s = 10^8$ Monte Carlo samples. As for the results with fixed truncation \bar{M} , we evaluate each m -dimensional integral in (3) using $N_s = 2 \times 10^7$ Monte Carlo samples. One can observe that the curve of observable tends to converge as \bar{M} grows. However, significant difference can still be observed between the results for $\bar{M} = 6$ and $\bar{M} = 8$, indicating that larger m needs to be taken into account to get reliable results, and thus considering m also as a random variable turns out to

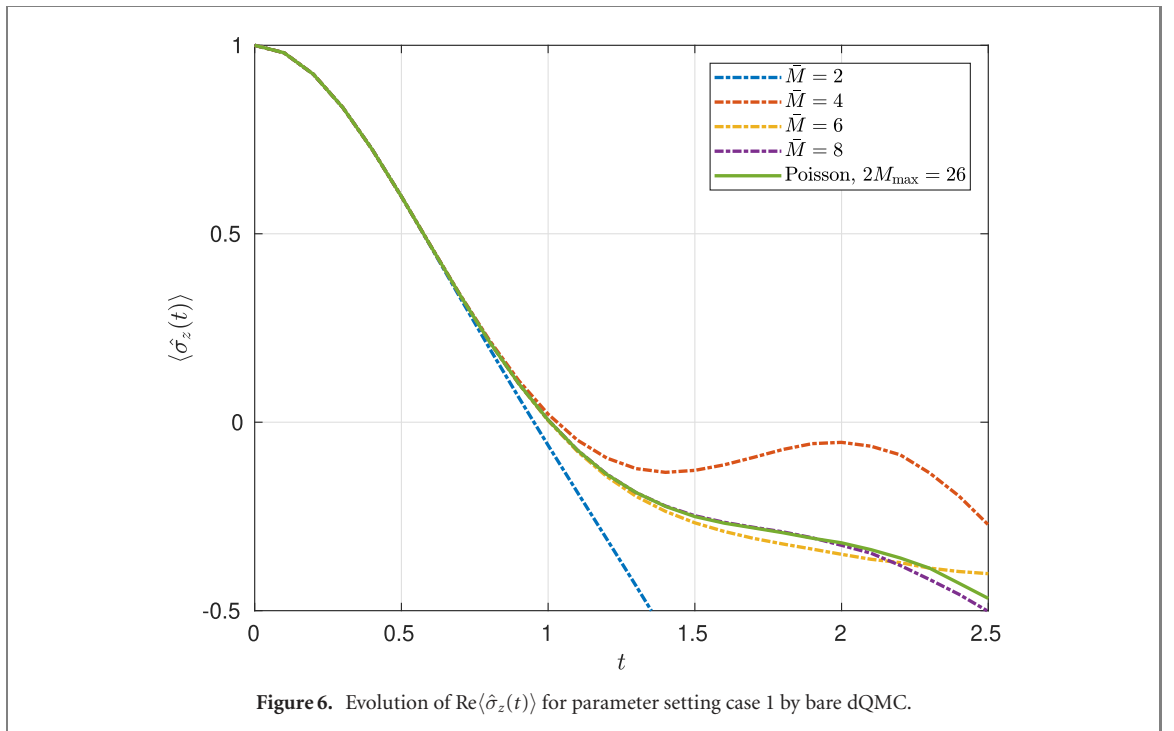
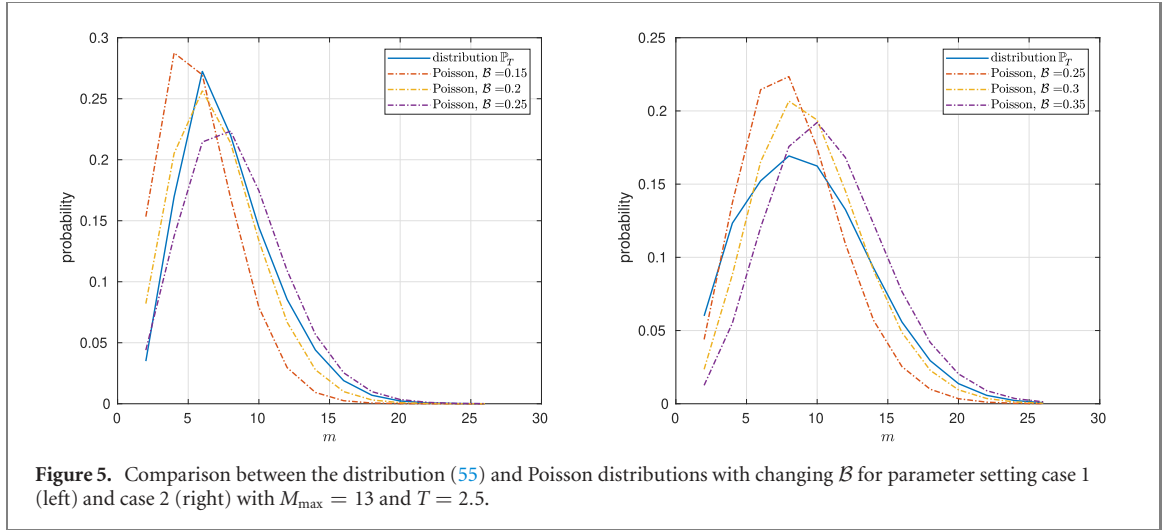


Table 2. Large m sampled by Poisson distribution in the simulation for case 1 by bare dQMC.

m	20	22	24	26
$\#\{m^{(j)} = m\}$	195 123	44 427	9197	1820

be an efficient way to find suitable number of samples. With this approach, larger m will be encountered in the simulation, and we have listed in table 2 the number of large m (within the red region of figure 3) sampled by the Poisson distribution, which also represents the number of m -point bath influence functionals evaluated in the entire simulation. For example, we need to compute 1820 independent $\mathcal{L}_b(s_1^{(j)}, \dots, s_{26}^{(j)})$ for Monte Carlo integration using inclusion–exclusion principle. The evaluation of such high-order bath influence functionals is hardly feasible using the direct method.

As for the simulations by inchworm Monte Carlo method, we refer to figure 7 for the numerical results with both parameter settings case 1 and case 2. The time step is again set as $h = 0.1$, while the number of samples is chosen as a relatively smaller $N_s = 10^5$ (N_s denotes the total number of samples used in the simulation by Poisson distribution, and the number of samples used for each $(m-1)$ -dimensional integral in the simulations with fixed truncation) since the numerical error of inchworm method is generally smaller than that of classic Dyson series [10]. For case 1, the curve with fixed \bar{M} becomes almost identical to $\bar{M} = 6$

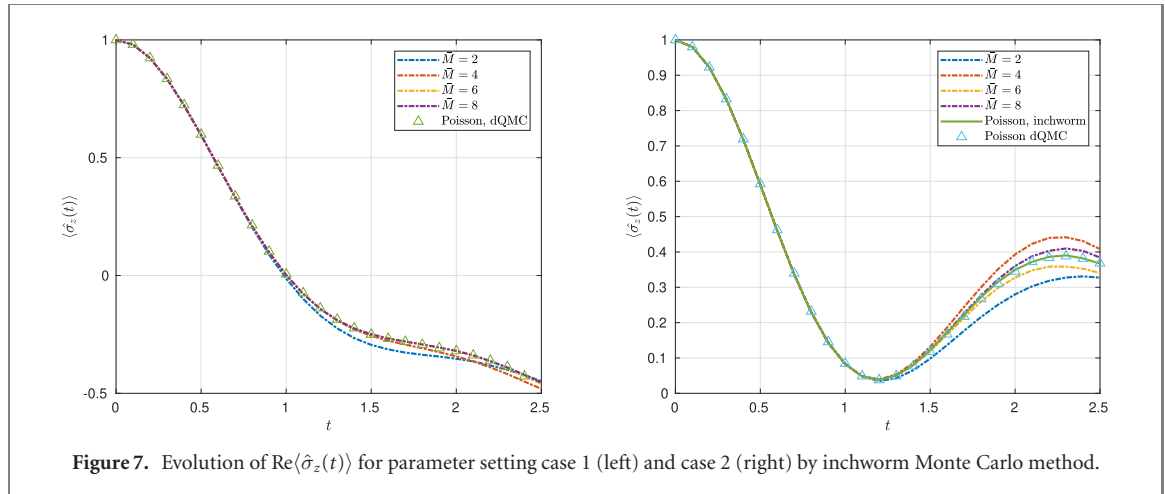


Figure 7. Evolution of $\text{Re}\langle \hat{\sigma}_z(t) \rangle$ for parameter setting case 1 (left) and case 2 (right) by inchworm Monte Carlo method.

Table 3. Large m sampled by Poisson distribution in the simulation for case 2 by inchworm Monte Carlo method.

m	20	22	24	26
$\#\{m^{(j)} = m\}$	17 710	5336	1492	395

thanks to the rapid convergence of inchworm method. The result by bare dQMC using Poisson distribution (same as the solid line in figure 6) is given as a reference. This indicates that inchworm Monte Carlo method can provide a satisfactory approximation to the exact solution with a small truncation \bar{M} and hence outperform bare dQMC for this set of parameters. As no larger m is needed, there is no need to apply the inclusion–exclusion principle in this case. However in case 2, the inchworm Monte Carlo method also suffers from slow convergence due to the slow decay of the two-point correlation function (see figure 2). The right panel of figure 7 shows that the discrepancy between $\bar{M} = 6$ and $\bar{M} = 8$ is still noticeable. With the adaptive choice of m , we are able to obtain results in good agreement with the reference results provide by bare dQMC with 10^8 samples for each $\langle \hat{\sigma}_z(nh) \rangle$. Again, we list in table 3 the number of samples involving large m in this experiment to show that inclusion–exclusion principle is indispensable to the calculation of long rounded boxes that direct method cannot deal with.

5. Conclusion

We have proposed fast algorithms based on inclusion–exclusion principle to sum diagrams appearing in the bare dQMC and inchworm Monte Carlo method. For bare dQMC, we have developed a formula to efficiently evaluate the bosonic bath influence functional at the cost of $O(2^m)$. Note that in the fermionic case, the bath influence functional becomes a determinant [27, 42], while in the bosonic case, the computational cost is higher, but it turns out that the computational complexity is lower than the Ryser’s algorithm for matrix permanents. For the inchworm method, our algorithm calculating the sum over linked diagrams can be considered as an extension to the work [8] which deals with the fermionic quantum impurity models. By a detailed complexity analysis, we have proved that the new algorithm reduces the computational cost from the original double factorial to exponential. More precisely, we estimate the computational complexity as $O(\alpha^{m/2})$ where $\alpha \approx 4.51891$, which has been also verified by our numerical experiments. Moreover, numerical simulations for the spin-boson model have been implemented to show the advantages of our approaches.

Funding

ZC was supported by the Academic Research Fund of the Ministry of Education of Singapore under Grant No. R-146-000-291-114. The work of JL was supported in part by the National Science Foundation via grants DMS-2012286 and CHE-2037263.

Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

Appendix A. Formulas of functionals

A.1. Definition of $\mathcal{U}^{(0)}$

The system related functional $\mathcal{U}^{(0)}$ in the Dyson series (3) is defined by

$$\mathcal{U}^{(0)}(0, s_1, \dots, s_m, 2t) = G_s^{(0)}(s_m, 2t) W_s G_s^{(0)}(s_{m-1}, s_m) W_s \dots W_s G_s^{(0)}(s_1, s_2) W_s G_s^{(0)}(0, s_1),$$

where

$$G_s^{(0)}(s_i, s_f) = \begin{cases} e^{-i(s_f - s_i)H_s}, & \text{if } s_i \leq s_f < t, \\ e^{-i(s_i - s_f)H_s}, & \text{if } t \leq s_i \leq s_f, \\ e^{-i(t - s_f)H_s} O_s e^{-i(t - s_i)H_s}, & \text{if } s_i < t \leq s_f. \end{cases}$$

A.2. Definition of \mathcal{U}

The functional \mathcal{U} in the integro-differential equation (18) is given by

$$\mathcal{U}(s_i, s_1, \dots, s_{m-1}, s_f) = G(s_{m-1}, s_f) W_s G(s_{m-2}, s_{m-1}) W_s \dots W_s G(s_1, s_2) W_s G(s_i, s_1),$$

where the full propagator $G(s_i, s_f)$ is defined by

$$G(s_i, s_f) = \begin{cases} \text{tr}_b(\rho_b G_b^{(0)}(s_f, 2t) e^{-i(s_f - s_i)H} G_b^{(0)}(0, s_i)), & \text{if } s_i \leq s_f < t, \\ \text{tr}_b(\rho_b G_b^{(0)}(s_f, 2t) e^{-i(s_i - s_f)H} G_b^{(0)}(0, s_i)), & \text{if } t \leq s_i \leq s_f, \\ \text{tr}_b(\rho_b G_b^{(0)}(s_f, 2t) e^{i(s_f - t)H} O e^{-i(t - s_i)H} G_b^{(0)}(0, s_i)), & \text{if } s_i < t \leq s_f \end{cases}$$

with the propagator associated with the bath

$$G_b^{(0)}(s_i, s_f) = \begin{cases} e^{-i(s_f - s_i)H_b}, & \text{if } s_i \leq s_f < t, \\ e^{-i(s_i - s_f)H_b}, & \text{if } t \leq s_i \leq s_f, \\ e^{-i(2t - s_i - s_f)H_b}, & \text{if } s_i < t \leq s_f. \end{cases}$$

The full propagator $G(s_i, s_f)$ satisfies

- Jump condition:

$$\lim_{s_f \rightarrow t^+} G(s_i, s_f) = O_s \lim_{s_f \rightarrow t^-} G(s_i, s_f);$$

$$\lim_{s_i \rightarrow t^-} G(s_i, s_f) = \lim_{s_i \rightarrow t^+} G(s_i, s_f) O_s.$$

- Boundary condition: $G(s_f, s_f) = \text{Id}$.

Appendix B. MATLAB code for computing the hafnian

Below we provide our MATLAB code to compute the hafnian of a symmetric matrix B . The input matrix needs to be a symmetric square matrix with all diagonal entries being zero.

```
function v = hafnian(B)
m = size(B,1);
R = zeros(1,2^m-1);
Q = zeros(1,2^m); Q(1) = sum(B, 'all') / 2;
sgn = zeros(2^m,1); sgn(1) = 1;
for i=1:m
    idx = 2^(i-1);
    R(idx) = sum(B(i,:));
    for kn=1:i-1
        j = idx + 2^(kn-1);
        R(j:2*j-idx-1) = R(idx:j-1) - B(i, kn);
    end
    Q(idx+1:2*idx) = Q(1:idx) - R(idx:2*idx-1);
    sgn(idx+1:2*idx) = -sgn(1:idx);
end
v = Q.(m/2) * sgn / factorial(m/2);
```

ORCID iDs

Siyao Yang  <https://orcid.org/0000-0002-6651-6224>

Zhenning Cai  <https://orcid.org/0000-0002-7086-7983>

Jianfeng Lu  <https://orcid.org/0000-0001-6255-5165>

References

- [1] Antipov A E, Dong Q, Kleinhenz J, Cohen G and Gull E 2017 Currents and green's functions of impurities out of equilibrium: results from inchworm quantum Monte Carlo *Phys. Rev. B* **95** 085144
- [2] Barvinok A 1999 Polynomial time algorithms to approximate permanents and mixed discriminants within a simply exponential factor *Random Struct. Algorithms* **14** 29–61
- [3] Beck M, Jackle A, Worth G A and Meyer H D 2000 The multiconfiguration time-dependent hartree (mctdh) method: a highly efficient algorithm for propagating wavepackets *Phys. Rep.* **324** 1–105
- [4] Björklund A 2012 Counting perfect matchings as fast as Ryser *Proc. of the 2012 Annual ACM-SIAM Symp. on Discrete Algorithms* ed Y Rabani pp 914–21
- [5] Björklund A, Gupt B and Quesada N 2019 A faster hafnian formula for complex matrices and its benchmarking on a supercomputer *ACM J. Exp. Algorithmics* **24** 1–17
- [6] Björklund A and Husfeldt T 2008 Exact algorithms for exact satisfiability and number of perfect matchings *Algorithmica* **52** 226–49
- [7] Björklund A, Husfeldt T and Koivisto M 2009 Set partitioning via inclusion–exclusion *SIAM J. Comput.* **39** 546–63
- [8] Boag A, Gull E and Cohen G 2018 Inclusion–exclusion principle for many-body diagrammatics *Phys. Rev. B* **98** 115152
- [9] Breuer H and Petruccione F 2007 *The Theory of Open Quantum Systems* (Oxford: Oxford University Press)
- [10] Cai Z, Lu J and Yang S 2020 Numerical analysis for inchworm Monte Carlo method: sign problem and error growth (arXiv:2006.07654)
- [11] Cai Z, Lu J and Yang S 2020 Inchworm Monte Carlo method for open quantum systems *Commun. Pure Appl. Math.* **73** 2430–72
- [12] Chen H-T, Cohen G and Reichman D R 2017 Inchworm Monte Carlo for exact non-adiabatic dynamics. I. Theory and algorithms *J. Chem. Phys.* **146** 054105
- [13] Chen H-T, Cohen G and Reichman D R 2017 Inchworm Monte Carlo for exact non-adiabatic dynamics. II. Benchmarks and comparison with established methods *J. Chem. Phys.* **146** 054106
- [14] Cohen G, Gull E, Reichman D R and Millis A J 2015 Taming the dynamical sign problem in real-time evolution of quantum many-body problems *Phys. Rev. Lett.* **115** 266802
- [15] Cygan M and Pilipczuk M 2015 Faster exponential-time algorithms in graphs of bounded average degree *Inf. Comput.* **243** 75–85
- [16] Davies E B 1974 Markovian master equations *Commun. Math. Phys.* **39** 91–110
- [17] Davies E B 1976 Markovian master equations. II *Math. Ann.* **219** 147–58
- [18] Dong Q, Krivenko I, Kleinhenz J, Antipov A E, Cohen G and Gull E 2017 Quantum Monte Carlo solution of the dynamical mean field equations in real time *Phys. Rev. B* **96** 155126
- [19] Duan C, Tang Z, Cao J and Wu J 2017 Zero-temperature localization in a sub-ohmic spin-boson model investigated by an extended hierarchy equation of motion *Phys. Rev. B* **95** 214308
- [20] Gull E, Millis A J, Lichtenstein A I, Rubtsov A N, Troyer M and Werner P 2011 Continuous-time Monte Carlo methods for quantum impurity models *Rev. Mod. Phys.* **83** 349–404
- [21] Kan R 2008 From moments of sum to moments of product *J. Multivariate Anal.* **99** 542–54
- [22] Keldysh L V 1964 Diagram technique for nonequilibrium processes *Zh. Eksp. Teor. Fiz.* **47** 1515–27
- [23] Mac Kernan D, Ciccotti G and Kapral R 2002 Surface-hopping dynamics of a spin-boson system *J. Chem. Phys.* **116** 2346–53
- [24] Koivisto M 2009 *Partitioning into Sets of Bounded Cardinality* (Berlin: Springer) pp 258–63
- [25] Makri N 1995 Numerical path integral techniques for long time dynamics of quantum dissipative systems *J. Math. Phys.* **36** 2430–57
- [26] Makri N, Sim E, Makarov D E and Topaler M 1996 Long-time quantum simulation of the primary charge separation in bacterial photosynthesis *Proc. Natl Acad. Sci.* **93** 3926–31
- [27] Mühlbacher L and Rabani E 2008 Real-time path integral approach to nonequilibrium many-body quantum systems *Phys. Rev. Lett.* **100** 176403
- [28] Mühlbacher L and Rabani E 2009 Diagrammatic Monte Carlo simulation of nonequilibrium systems *Phys. Rev. B* **79** 035320
- [29] Nederlof J 2009 Fast polynomial-space algorithms using Möbius inversion: improving on Steiner tree and related problems *Automata, Languages and Programming* (Berlin: Springer) pp 713–25
- [30] Negele J W and Orland H 1998 *Quantum Many-Particle Systems: Advanced Book Classics* (Boulder: Westview Press)
- [31] Nielsen M A and Chuang I L 2010 *Quantum Computation and Quantum Information* 10th Anniversary edn (Cambridge: Cambridge University Press)
- [32] Ridley M, Singh V N, Gull E and Cohen G 2018 Numerically exact full counting statistics of the nonequilibrium Anderson impurity model *Phys. Rev. B* **97** 115109
- [33] Rossi R 2017 Determinant diagrammatic Monte Carlo algorithm in the thermodynamic limit *Phys. Rev. Lett.* **119** 045701
- [34] Ryser H J 1963 *Combinatorial Mathematics* 1 edn vol 14 (Washington, DC: Mathematical Association of America)
- [35] Schiró M 2010 Real-time dynamics in quantum impurity models with diagrammatic Monte Carlo *Phys. Rev. B* **81** 085126
- [36] Schiró M and Fabrizio M 2009 Real-time diagrammatic Monte Carlo for nonequilibrium quantum transport *Phys. Rev. B* **79** 153302
- [37] Stein P R and Everett C J 1978 On a class of linked diagrams II. Asymptotics *Discrete Math.* **21** 309–18
- [38] Tanimura Y 1990 Nonperturbative expansion method for a quantum system coupled to a harmonic-oscillator bath *Phys. Rev. A* **41** 6676–87
- [39] Tanimura Y and Kubo R 1989 Time evolution of a quantum system in contact with a nearly Gaussian–Markoffian noise bath *J. Phys. Soc. Japan* **58** 101–14
- [40] Thomas G and Finney R 1988 *Calculus and Analytic Geometry* 7th edn (Reading, MA: Addison-Wesley)

- [41] Wang H 2000 Basis set approach to the quantum dissipative dynamics: application of the multiconfiguration time-dependent Hartree method to the spin-boson problem *J. Chem. Phys.* **113** 9948–56
- [42] Werner P, Comanac A, Medici L, Troyer M and Millis A J 2006 Continuous-time solver for quantum impurity models *Phys. Rev. Lett.* **97** 076405