PATHSUM: A C++ and Fortran Suite of Fully Quantum Mechanical Real-Time Path Integral Methods for (Multi-)System+Bath Dynamics

Sohang Kundu[1] and Nancy Makri[1,2,3] *

[1]Department of Chemistry, University of Illinois, Urbana, Illinois 61801, USA
[2]Department of Physics, University of Illinois, Urbana, Illinois 61801, USA
[3]Illinois Quantum Information Science and Technology Center, University of Illinois, Urbana, Illinois 61801, USA

Email: nmakri@illinois.edu

Abstract

This paper reports the release of PATHSUM, a new software suite of state-of-the-art path integral methods for studying the dynamics of single or extended systems coupled to harmonic environments. The package includes two modules, suitable for system-bath problems and extended systems comprising many coupled system-bath units, and is offered in C++ and Fortran implementations. The system-bath module offers the recently developed small matrix path integral (SMatPI) and the well-established iterative quasi-adiabatic propagator path integral (iQuAPI) method for iteration of the reduced density matrix of the system. In the SMatPI module the dynamics within the entanglement interval can be computed using QuAPI, the blip sum, time evolving matrix product operators (TEMPO) or the quantum-classical path integral (QCPI) method. These methods have distinct convergence characteristics and their combination allows a user to access a variety of regimes. The extended system module provides the user with two algorithms of the modular path integral (MPI) method applicable to quantum spin chains or excitonic molecular aggregates. An overview of the methods and code structure is provided along with guidance on method selection and representative examples.

I. Introduction

The quest for understanding the intricate pathways of complex processes dominated by quantum effects has sparked intense efforts toward developing simulation algorithms. Many processes of interest involve two or more coupled electronic or spin states that interact with a large number of nuclei in a finite-temperature statistical ensemble. A variety of theoretical treatments have been developed, which by utilizing convenient but often severe assumptions and approximations are capable of capturing some important quantum dynamical effects in large chemical and biological environments and which have proven extremely valuable in chemistry. For example, Redfield theory[1,2] has led to the fundamental understanding of spin relaxation and energy transfer in the weak coupling limit. Surface hopping algorithms[3-5] have been widely used in proton-coupled electron transfer processes.[6] Quasiclassical methods, such the Wigner method,[7,8] which is alternatively known as the linearized semiclassical initial value representation[9-11] (LSC-IVR) and has also been derived by linearizing the path integral expression[12]), forward-backward semiclassical dynamics[13,14] (FBSD), and path integral Liouville dynamics[15,16] (PILD) capture important nuclear quantum effects and are suitable for large-scale molecular dynamics simulations. However, because of the major assumptions they involve, such methods are not suitable for resolving some of the most intriguing questions that surround electron, proton, or energy transfer, as well as coupled spin dynamics.

The full solution of the time-dependent Schrödinger equation remains prohibitively expensive, as the manipulation and storage of wavefunctions requires resources that formally scale exponentially with the number of degrees of freedom. In addition, accounting for thermal effects requires an unrealistically large number of separate wavefunction-based calculations. Among wavefunction-based schemes, the multiconfiguration time-dependent Hartree (MCTDH) method[17-19] along with its multilayer extension[20] has shown impressive success on molecular systems. The main advantage of the MCTDH approach is its ability to also treat anharmonic bath degrees of freedom. However, wavefunction-based methods are naturally suited to zero-temperature properties, and their extension to finite temperature by summing a large number of microcanonical results is computationally prohibitive when the environment includes a large number of low-frequency modes. Methods based on matrix product states,[21,22] such as the density matrix renormalization group[23,24] (DMRG), can be highly efficient for extended systems, but their cost increases rapidly with the size of each unit.

The path integral formulation of quantum mechanics[25,26] provides an alternative starting point, which is appealing because it does not require the use of wavefunctions. Further, the path integral offers an intuitive, classical-like picture, where paths carry interfering phases that are responsible for quantum coherence and its destruction. The classical limit emerges naturally and elegantly from this formulation.[27] In spite of its appeal, numerical evaluation of the real-time path integral (for dynamical processes) requires the summation of astronomical numbers of terms, and the use of Monte Carlo methods is plagued by a "sign problem" that arises from the highly oscillatory quantum mechanical phase.[28,29]

By replacing the time parameter by $-i\hbar\beta$ (where $\beta = 1/k_{\mathrm{B}}T$), the path integral also leads to a powerful formulation of quantum statistical mechanics,[30] offering a transparent view of quantum delocalization through the quantum-classical isomorphism.[31] The imaginary-time path integral does not suffer from a sign problem (except in the case of identical fermions) and has led to the development of efficient simulation methods for highly complex systems that employ Monte Carlo or molecular dynamics sampling.[32-34] Further, the imaginary-time path integral forms an excellent basis for the development of efficient dynamical approximations, such as centroid molecular dynamics[35] (CMD) and ring polymer molecular dynamics[36,37] (RPMD). The rigorous connection between the fully quantum mechanical result

and the RPMD approximation has recently become clear through the development of Matsubara dynamics.[38,39]

The system-bath Hamiltonian, where the system of interest is coupled to a large number of harmonic degrees of freedom, provides a convenient and relatively simple model for investigating the effects of condensed phase environments on the dynamics of a small system, but also allows a realistic and accurate description of a variety of processes. This is so because the harmonic bath can comprise molecular vibrations at the normal mode level, lattice phonons, and also in many cases can mimic the collective effects of complex environments through Gaussian response.[40-42] The path integral formulation offers a unique advantage in the case of system-bath Hamiltonians because harmonic bath degrees of freedom can be integrated out analytically at any temperature, replacing all variables associated with the bath by an influence functional.[43] However, the cost of this enormous simplification is the introduction of temporal nonlocality in the path integral expression for the system, in a way analogous to the presence of memory in the generalized Langevin equation.[44] This nonlocality prevents the stepwise evaluation of the path integral by matrix-vector multiplication techniques commonly employed in the absence of coupling to a bath,[45,46] appearing to require the summation of a number of terms that grows exponentially with propagation time. Early attempts to sample these terms by Monte Carlo methods were met with limited success. However, in the special case of a harmonic bath described by a spectral density of the Drude form, the hierarchical equations of motion (HEOM) method[47] allows efficient simulation of system-bath dynamics. Extension to more general situations requires a decomposition of the spectral density into Drude-type components,[48] but since the algorithm scales exponentially with respect to the number of such terms, this approach is not practical for simulating processes in structured environments.

Stable, numerically exact real-time path integral methods, based on the full evaluation of the system-bath path integral by quadrature[49-54] (thus circumventing the Monte Carlo sign problem) emerged in the early 1990s and became known as the quasi-adiabatic propagator path integral (QuAPI). To remedy the numerical issues encountered by directly attempting to evaluate Feynman's expression, QuAPI exploits the finite energy span of dynamical relevance to construct smooth propagators that are free of the highly oscillatory phase,[49,55] employs a physically motivated partitioning of the Hamiltonian that allows sizable time steps,[49] and introduces system-specific discrete variable representations[50] of the influence functional that minimize the number of grid points. Further, QuAPI removes the exponential scaling with the number of path integral time steps by taking advantage of the finite span of memory induced by macroscopic environments to decompose the path sum into a series of successive tensor-based steps that are evaluated by quadrature.[52-54] The QuAPI methodology, along with many additional improvements and extensions by our group and several others,[56-69] has enabled the fully quantum mechanical investigation of a variety of processes.

During the last decade, a number of further developments in real-time path integral methodology led to new, powerful tools for system-bath simulation. The blip representation[70] achieves an exponential reduction of the number of path integral variables and is now used in practically all path integral algorithms. Further, the blip-summed path integral (BSPI or BlipSum) offers a systematic filtering tool, as multi-blip paths make negligible contributions, and exploits the structure of the influence functional to sum the majority of terms by inexpensive procedures, resulting in dramatic savings in some regimes. The time-evolving matrix product operator (TEMPO) representation of the QuAPI expression,[71-75] based on the structure of the influence functional, brings the path amplitude in matrix product state (MPS) form, employing singular value decomposition (SVD) to prevent the exponential growth of the matrices, offering

a powerful algorithm for some regimes. Very recent work showed analytically that the path integral can be decomposed into small matrices of minimal size, eliminating the need for QuAPI tensors. The resulting small matrix path integral[76-78] (SMatPI) reduces the cost to that of a single QuAPI step, and the replacement of tensors by small matrices allows the treatment of systems with many states.

Further, the path integral formulation leads to a rigorous and consistent mixed quantum-classical treatment that does not encounter the well-known issues[5] encountered when Newtonian trajectories are combined with wavefunctions. The quantum-classical path integral[79-81] (QCPI) captures the interaction between the system and its environment through phases along classical trajectories and can be used for simulating nonadiabatic processes such as charge and proton transfer in solution or biological environments, without any *ad hoc* assumptions. The QCPI expression becomes fully quantum mechanical and exact in the case of a harmonic bath, offering an additional approach to system-bath dynamics which has some distinct advantages in some regimes.

Last, a modular path integral (MPI) algorithm[82-86] has been developed for extended systems, composed of many system-bath units in a one-dimensional arrangement with mostly spatially local couplings. In this case the path integral is evaluated by summing the variables of each unit after linking them to those of its neighbor and including the relevant influence functional factors. The MPI algorithm gives rise to linear scaling with system size and may be used for simulating energy transfer in long molecular aggregates with Frenkel exciton interactions or one-dimensional arrangements of spins coupled to local molecular vibrations, where the total number of states coupled to harmonic baths exceeds by far the capabilities of other methods.

These numerically exact, fully quantum mechanical algorithms offer powerful tools for simulating the dynamics of system-bath (and extended) Hamiltonians. In combination, these methods provide the needed flexibility for treating a variety of processes in diverse parameter regimes, which are not accessible to any single approach. In addition to many investigations that have been carried out with these methods, recent work reported simulations of spin chain dynamics,[87] molecular polaritons[88,89] and energy transfer in large molecular aggregates,[90,91] the FMO complex,[92,93] the B850 light harvesting ring[94,95] and the 24-bacteriochlorophyll LH2 complex.[96,97]

In this paper we describe PATHSUM, our new software suite in C++ and Fortran, which implements these fully quantum mechanical real-time path integral methods. There are two broad modules in this package: System-Bath and Extended-System, which differ in the scope of problems that they address and in the methods they utilize. The System-Bath module implements the SMatPI and i-QuAPI methodologies for iterative evaluation of the RDM beyond memory. (We use the term i-QuAPI to explicitly refer to the iterative algorithm, and reserve the term QuAPI for calculations within the entangled memory length.) The system can be coupled to common as well as local baths, and interactions within the system can have arbitrary values. Although i-QuAPI is provided, we advise the user to utilize the SMatPI method whenever possible due its significant advantages. When using the SMatPI method of iteration, the user is free to choose among QuAPI, BlipSum, TEMPO and QCPI to compute the dynamics within the entangled memory interval. The Extended-System module offers methods for systems that are too large to be handled with the System-Bath module, such as extended chains of spin-bath units or coupled molecular aggregates characterized by nearest neighbor interactions. Within this module the code implements the MPI method which has two available algorithms, depending on whether the interactions between neighboring units are diagonal (such as in the Ising Hamiltonian) or of a general form (i.e., in the Heisenberg or Frenkel

Hamiltonians) in the system basis. While the QCPI method is generally applicable to anharmonic environments, all methods in the PATHSUM package are restricted to harmonic baths.

In section II we provide an overview of the theoretical framework, describing the system-bath Hamiltonian and that of extended systems, the relevant parameters and the observables. In section III we summarize the various methods implemented in the code. The structure of the code, a concise description of the online resources available with the package, and information about its dependencies on commonly used libraries, are described in section IV. In section V we summarize the convergence parameters and the scaling of the methods, offering some guidance on how to select among the available options. In section VI we give six tutorial examples, illustrating some of the possibilities of this package, and encourage the reader to try out other alternative choices of methods across the different regimes encountered in condensed phase dynamics (e.g. weak vs. strong dissipation, or low vs. high temperature, as well as small vs. large systems). In section VII we present some concluding remarks.

II. Theoretical Framework

The PATHSUM code is currently applicable to Hamiltonians of the system-bath form, or to extended systems composed of multiple coupled system-bath units. In this section we review the types of Hamiltonians along with important terminology, introduce our notation, and describe the required input and the produced dynamical quantities.

(a) System-Bath Hamiltonian

In general, the system-bath Hamiltonian has the form

$$\hat{H} = \hat{H}_0 + \hat{H}_{\mathrm{sb}} \tag{2.1}$$

where H_0 is the Hamiltonian of the adiabatically renormalized system[49] and the second term describes the interaction between the system and the harmonic bath degrees of freedom. The system may be described by a continuous coordinate *s* through a potential function, or by a collection of discrete states. Continuous systems can be converted to discrete ones by using a discrete variable representation (DVR).[50,98,99] Thus, regardless of its original form, the system Hamiltonian is treated as a matrix of *n* basis states $|\varphi_\alpha\rangle$, $\alpha = 1,\ldots,n$, which (within the *n*-state subspace) are eigenstates of the system coordinate operator $\hat{s}$.

The system states may couple to a common (e.g. phonon) bath and/or to multiple local harmonic environments[100] (for example, the normal mode vibrations of each molecule in a molecular aggregate). In the case of a common bath, the system coordinate operator is expressed in the form

$$\hat{s} = \sum_{\alpha=1}^{n} \sigma_\alpha |\varphi_\alpha\rangle\langle\varphi_\alpha| \tag{2.2}$$

where σ_α are position-like parameters for the system and the system-bath coupling is

$$\hat{H}_{\mathrm{sb}} = \sum_{i=1}^{\nu}\left[\frac{\hat{p}_i^2}{2m} + \frac{1}{2}m\omega_i^2\left(\hat{q}_i - \frac{c_i\hat{s}}{m\omega_i^2}\right)^2\right] = \sum_{i=1}^{\nu}\left(\frac{\hat{p}_i^2}{2m} + \frac{1}{2}m\omega_i^2\hat{q}_i^2 - c_i\hat{s}\hat{q}_i + \frac{c_i^2\hat{s}^2}{2m\omega_i^2}\right) \tag{2.3}$$

where ν is the (finite or infinite) number of bath degrees of freedom. Eq. (2.3) may also be expressed in terms of the discrete system states as

$$\hat{H}_{\mathrm{sb}} = \sum_{i=1}^{\nu}\left(\frac{\hat{p}_i^2}{2m} + \frac{1}{2}m\omega_i^2\hat{q}_i^2 - c_i\hat{q}_i\sum_{\alpha=1}^{n}\sigma_\alpha|\varphi_\alpha\rangle\langle\varphi_\alpha| + \frac{c_i^2\sigma_\alpha^2}{2m\omega_i^2}|\varphi_\alpha\rangle\langle\varphi_\alpha|\right). \tag{2.4}$$

The bath degrees of freedom may be specified by the individual frequencies ω_i and coupling constants c_i or by a continuous spectral density function,[101]

$$J(\omega) = \frac{2\pi}{\Delta\sigma^2}\sum_{i=1}^{\nu}\frac{c_i^2}{m\omega_i}\delta(\omega-\omega_i) \tag{2.5}$$

where $\Delta\sigma$ is a characteristic distance parameter of the system (e.g., $\Delta\sigma = |\sigma_n - \sigma_1|$).

In different situations, for example in a simulation of energy transfer in a molecular aggregate within the Frenkel exciton framework,[102,103] the system consists of the singly excited electronic states of the aggregate and (some or all of) the bath modes represent the intramolecular vibrations of each monomer within the normal mode approximation, which couple to the ground and excited electronic states of the particular unit. Within the singly-excited Frenkel subspace the ground state does not participate in the dynamics, thus the normal modes of a monomer constitute a local bath that couples to the corresponding system state. The system-bath coupling has the form

$$\hat{H}_{\mathrm{sb}} = \sum_{\alpha=1}^{n}\left(\hat{h}_\alpha^{\mathrm{e}} + \sum_{\alpha'\neq\alpha}\hat{h}_{\alpha'}^{\mathrm{g}}\right)|\varphi_\alpha\rangle\langle\varphi_\alpha| \tag{2.6}$$

where

$$\hat{h}_\alpha^{\mathrm{g}} = \sum_{i=1}^{\nu}\left(\frac{\hat{p}_{\alpha i}^2}{2m} + \frac{1}{2}m\omega_{\alpha i}^2\left(\hat{q}_{\alpha i} - \frac{c_{\alpha i}\sigma_\alpha^{\mathrm{g}}}{m\omega_{\alpha i}^2}\right)^2\right), \quad \hat{h}_\alpha^{\mathrm{e}} = \sum_{i=1}^{\nu}\left(\frac{\hat{p}_{\alpha i}^2}{2m} + \frac{1}{2}m\omega_{\alpha i}^2\left(\hat{q}_{\alpha i} - \frac{c_{\alpha i}\sigma_\alpha^{\mathrm{e}}}{m\omega_{\alpha i}^2}\right)^2\right). \tag{2.7}$$

are the vibrational components of the ground and excited states. In this case the parameter $\Delta\sigma_\alpha = |\sigma_\alpha^{\mathrm{e}} - \sigma_\alpha^{\mathrm{g}}|$ characterizes the pair of ground and excited electronic states of monomer α and the coupling constants are obtained from the corresponding Huang-Rhys factors $S_{i\alpha}$ through the relation

$$c_{i\alpha} = \sqrt{2m\omega_{i\alpha}^3\hbar S_{i\alpha}/\Delta\sigma_\alpha^2}\,. \tag{2.8}$$

The Hamiltonian for local baths and also for more complex cases of correlated baths can be described by a system-bath expression in vector form, with the coupling terms being given by[100,104]

$$-\sum_{i=1}^{\nu} \hat{\mathbf{q}}_i^{\mathrm{T}} \cdot \mathbf{C}_i \cdot \hat{\mathbf{s}} \tag{2.9}$$

where the system coordinate and position parameters have been replaced by *n*-component vectors, i.e.,

$$\hat{\mathbf{s}} = \sum_{\alpha=1}^{n} \boldsymbol{\sigma}_\alpha \left|\varphi_\alpha\right\rangle\left\langle\varphi_\alpha\right|. \tag{2.10}$$

(b) Extended systems comprising many system-bath units with local couplings

The second class of Hamiltonians involves extended systems, composed of *d* system-bath units, each with one or more quantum states, in a locally one-dimensional arrangement (which could be a branched chain, a cyclic structure, or may have a more complex topology). If each system comprises a single state (e.g. the excited state of a chromophore), the composite structure is described by a system-bath Hamiltonian of *d* states with local baths. In this case the need for alternative approaches arises when *d* exceeds the system size treatable by the available system-bath methods. On the other hand, if each system has $n > 1$ states, the composite Hamiltonian involves n^d states. The important case of spin-½ molecular units with $n = 2$, which gives rise to quantum Ising and the more general Heisenberg models coupled to dissipative environments, leads to a system of 2^d states. Clearly, simulating the dynamics of such system-bath Hamiltonians is limited to a handful of spins.

Rather than considering the total number of states as a very large system coupled to local baths, a more profitable approach is to treat each system-bath unit separately, then couple the units. If the couplings between units are mostly local, fully quantum mechanical calculations are possible using the MPI methodology,[82-86] which offers linear scaling with *d*. The PATHSUM code currently treats chains of two-state systems interacting with local baths and connected by nearest-neighbor coupling terms according to the Hamiltonian

$$\hat{H}_0 = -\hbar\sum_{\alpha=1}^{d} \Omega_\alpha \hat{\sigma}_\alpha^x - J_x \sum_{\alpha=1}^{d-1} \hat{\sigma}_\alpha^x \hat{\sigma}_{\alpha+1}^x - J_y \sum_{\alpha=1}^{d-1} \hat{\sigma}_\alpha^y \hat{\sigma}_{\alpha+1}^y - J_z \sum_{\alpha=1}^{d-1} \hat{\sigma}_\alpha^z \hat{\sigma}_{\alpha+1}^z \tag{2.11}$$

where

$$\begin{aligned} \hat{\sigma}_\alpha^x &= \left|\varphi_\alpha^{\mathrm{R}}\right\rangle\left\langle\varphi_\alpha^{\mathrm{L}}\right| + \left|\varphi_\alpha^{\mathrm{L}}\right\rangle\left\langle\varphi_\alpha^{\mathrm{R}}\right| \\ \hat{\sigma}_\alpha^y &= -i\left|\varphi_\alpha^{\mathrm{R}}\right\rangle\left\langle\varphi_\alpha^{\mathrm{L}}\right| + i\left|\varphi_\alpha^{\mathrm{L}}\right\rangle\left\langle\varphi_\alpha^{\mathrm{R}}\right| \\ \hat{\sigma}_\alpha^z &= \left|\varphi_\alpha^{\mathrm{R}}\right\rangle\left\langle\varphi_\alpha^{\mathrm{R}}\right| - \left|\varphi_\alpha^{\mathrm{L}}\right\rangle\left\langle\varphi_\alpha^{\mathrm{L}}\right| \end{aligned} \tag{2.12}$$

are the three Pauli spin operators. Setting $J_x = J_y = 0$ one obtains the quantum Ising chain, while the choice $J_x = J_y = J/2,\ J_z = 0$ gives the Frenkel exciton Hamiltonian[102,103] with coupling J.

(c) Input parameters and observables

Continuous systems are specified in terms of the particle mass, a one-dimensional potential function provided by the user, and the spectral density of the (common) bath. PATHSUM immediately converts a continuous system to a discrete Hamiltonians that involves a specified number *n* of DVR and proceeds to apply the discrete system procedures.

The following input is required to specify the system-bath and extended system-bath Hamiltonians:

- The number *n* of system states.
- The $n \times n$ matrix of the system Hamiltonian H_0.
- For a system coupled to a common bath, the system coordinates $\sigma_1, \ldots, \sigma_n$.
- For a common bath, the ν pairs ω_i, c_i or the spectral density function $J(\omega)$ on a fine frequency grid (or both).
- For local baths, the set of ν_α pairs $\omega_{\alpha i}, c_{\alpha i}$ for each monomer or the set of *n* spectral density functions $J_\alpha(\omega)$ (or both).
- The reciprocal temperature, $\beta = 1/k_B T$.
- In the case of extended systems, the TLS frequency and the three coupling parameters J_x, J_y, J_z between adjacent TLS units.

The system-bath module computes the $n^2 \times n^2$ reduced density matrix (RDM) of the system,

$$\tilde{\rho}^{(N0)}_{\alpha_N^\pm \alpha_0^\pm} = \mathrm{Tr}_\mathrm{b} \left\langle \varphi_{\alpha_N^+} \middle| e^{-i\hat{H}t/\hbar} \middle| \varphi_{\alpha_0^+} \right\rangle \hat{\rho}_\mathrm{b} \left\langle \varphi_{\alpha_0^-} \middle| e^{i\hat{H}t/\hbar} \middle| \varphi_{\alpha_N^-} \right\rangle \tag{2.13}$$

at time values $N\Delta t$, where Δt is the path integral time step, for a system initial condition specified by $\alpha_0^\pm$. Here $\hat{\rho}_\mathrm{b}$ is the Boltzmann operator of the bath in equilibrium with one of the system states. In the case of a common bath, shifting the system coordinate values such that $\sigma_\alpha = 0$ places the bath in equilibrium with state φ_α. For local environments, the bath is placed in equilibrium with the ground state by setting $\sigma_\alpha^\mathrm{g} = 0$ and with the excited state φ_α by setting $\sigma_\alpha^\mathrm{e} = 0$.

In the case of an extended system, the code gives the RDM of a particular TLS, traced with respect to all other TLS units.

III. Review of PATHSUM Methods

A. QuAPI

The QuAPI method involves two components: the discretization of the path integral and the iteration that gives rise to linear scaling with the number of propagation steps.

The QuAPI discretization employs the physically motivated quasi-adiabatic propagator partitioning of the time evolution operator[49] and an optimal discretization of continuous potentials on a DVR grid.[50,98,99] The RDM is expressed in the form

$$\tilde{\rho}^{(N0)}_{\alpha_N^\pm \alpha_0^\pm} = \sum_{\alpha_{N-1}^\pm=1}^{n} \cdots \sum_{\alpha_1^\pm=1}^{n} K_{\alpha_N^\pm \alpha_{N-1}^\pm} \cdots K_{\alpha_1^\pm \alpha_0^\pm} F_{\alpha_N^\pm \ldots \alpha_1^\pm \alpha_0^\pm} \tag{3.1}$$

where $K_{\alpha_{k+1}^\pm \alpha_k^\pm}$ is the short-time propagator of the system and $F_{\alpha_N^\pm \ldots \alpha_1^\pm \alpha_0^\pm}$ is the QuAPI-discretized influence functional[43] at the given temperature, given by

$$F\left(s_N^{\pm},\ldots,s_1^{\pm},s_0^{\pm}\right)=e^{-\frac{1}{\hbar}\sum_{k'=0}^{N}\sum_{k''=0}^{k'}\left(s_{k'}^{+}-s_{k'}^{-}\right)\left(\eta_{k'k''}s_{k''}^{+}-\eta_{k'k''}^{*}s_{k''}^{-}\right)}\tag{3.2}$$

The coefficients $\eta_{k'k''}$ are computed from the spectral density function.[54] In the simplest QuAPI algorithm, the sum in Eq. (3.1) involves n^{2N} terms from each initial condition, i.e. a total of n^{2N+2} terms if all initial conditions are of interest, and is performed without additional simplifications besides (optionally) path filtering (i.e. dropping paths that are anticipated to make negligible contributions). The QuAPI sum is fully parallelized in the PATHSUM code.

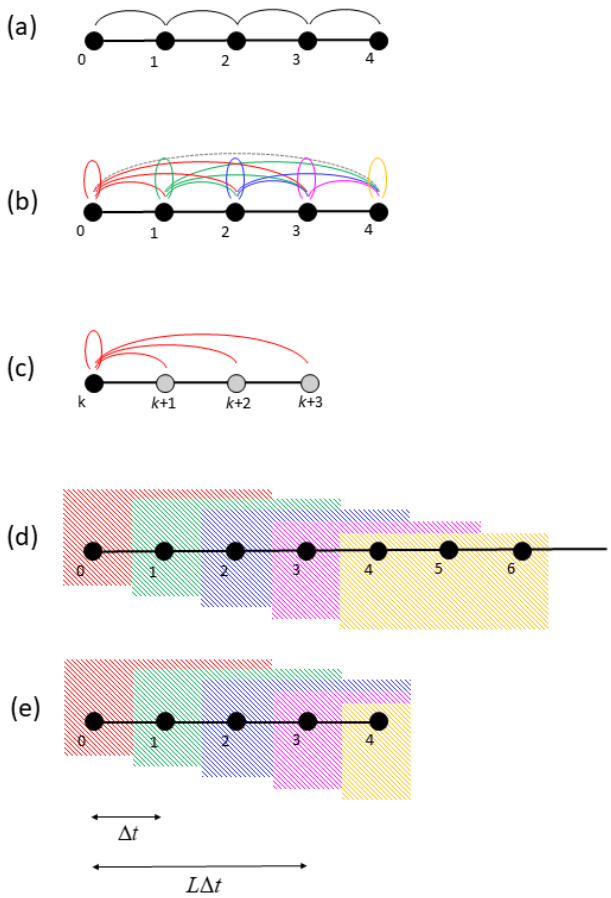

Fig. 1. Schematic illustration of the couplings among path integral variables and the i-QuAPI decomposition of the path integral for a system coupled to a harmonic bath. (a) Couplings in the absence of a bath. Only nearest-neighbor couplings are present in this case. (b) Coupling to a bath introduces non-nearest-neighbor couplings. Those connecting points separated by more than $L=3$ path integral time steps are shown as dashed lines. (c) Influence functional couplings from a single path integral variable for a memory equal to $L=3$ path integral time steps. (d) Path segments spanning the memory length for $L=3$ are shown as colored rectangles. The tensor decomposition for continued propagation connects each of these segments to those in the adjacent array after including the influence functional couplings shown in panel (c). (e) Termination of the i-QuAPI algorithm for propagation to $N=4$ time steps.

The i-QuAPI algorithm employs a tensor decomposition of Eq. (3.1),[52,53]

$$R^{(k+1)}_{\alpha^{\pm}_{k+L},\ldots,\alpha^{\pm}_{k+1}} = \sum_{\alpha^{+}_{k}=1}^{n} T_{\alpha^{\pm}_{k+L},\ldots,\alpha^{\pm}_{k}} R^{(k)}_{\alpha^{\pm}_{k+L-1},\ldots,\alpha^{\pm}_{k}} \tag{3.3}$$

where **T** is a tensor that contains the short time propagators and influence functional couplings that connect the path integral variables of the tensors $\mathbf{R}^{(k)}$ and $\mathbf{R}^{(k+1)}$ within the memory length encoded in the influence functional. The tensor multiplication is repeated until the desired time is reached, giving rise to linear scaling with the number of time steps. The tensor $\mathbf{R}^{(k)}$ has n^{2L} elements and each step in Eq. (3.3) involves n^{2L+2} operations. Once converged with respect to the time steps and the memory length included, the i-QuAPI propagation yields the fully quantum mechanical and numerically exact result for the RDM of the system-bath Hamiltonian.[54,105] The algorithm is illustrated in Figure 1. The i-QuAPI methodology has also been applied to time correlation functions.[106] The i-QuAPI code runs on a single processor.

We emphasize that finite memory requires a truly macroscopic environment with a continuous spectral density. Baths consisting of a moderate number of discrete modes cannot be treated by iterative path integral algorithms. (However, if a bath comprises a dense manifold of coupled modes over a broad frequency range, the resulting memory can be finite for practical purposes. This is discussed again in connection with the QCPI method.)

B. BlipSum

The blip representation[70] performs a change of the path integral variables to

$$\Delta\sigma_{\alpha_{k'}} = \sigma^{+}_{\alpha_{k'}} - \sigma^{-}_{\alpha_{k'}}, \quad \bar{\sigma}_{\alpha_{k'}} = \frac{1}{2}\left(\sigma^{+}_{\alpha_{k'}} + \sigma^{-}_{\alpha_{k'}}\right). \tag{3.4}$$

Time points with $\Delta\sigma_k \neq 0$ are 'blips', while those with $\Delta\sigma_k = 0$ are the 'sojourns'.[107] In the blip representation, the influence functional of Eq. (3.2) has the form

$$\exp\left(-\frac{1}{\hbar}\sum_{k'=0}^{N}\sum_{k''=0}^{k'}\Delta\sigma_{\alpha_{k'}}\left(\mathrm{Re}\,\eta_{k'k''}\Delta\sigma_{\alpha_{k''}} + 2i\,\mathrm{Im}\,\eta_{k'k''}\bar{\sigma}_{\alpha_{k''}}\right)\right). \tag{3.5}$$

The outer sum in this expression contains all forward-backward path pairs with $b = 0,1,2,\ldots N$ blips. Further, the blip-blip interaction gives rise to a decaying exponential, implying that forward-backward path pairs with many blips make vanishing contributions to the path sum.[70] The damping effect of the blip-blip interaction is most pronounced when the system-bath coupling is strong and at high temperatures. Under such favorable conditions, converged results can be obtained for tens or even hundreds of time steps, such that results for the entire period of interest may be obtained without resorting to iterative methods.[70]

Another very attractive advantage of the blip representation is the ability to evaluate the inner (sojourn) sum through a series of iterative matrix-vector multiplications. This is a consequence of the structure of the influence functional, which involves only blip-blip and blip-sojourn interactions (see Figure 2). If all blips (i.e. all path integral terms) are included, the blip sum requires $\left(n^2 - n + 1\right)^N$ operations from

each initial condition, which (at least for small values of n) implies a much lower cost compared to the path sum in the original QuAPI representation that involves n^{2N} terms. The BlipSum component is fully parallelized.

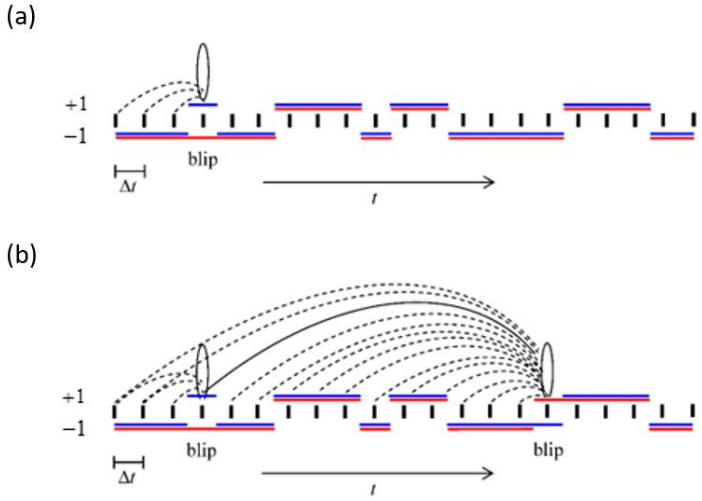

Fig. 2. Schematic illustration of influence functional interactions for two forward-backward path configurations with (a) a single blip at $k_1 = 3$ and (b) two blips at $k_1 = 3, k_2 = 15$. Red and blue lines show the coordinates of the forward and backward TLS paths. Black vertical bars indicate the time grid. Solid and dashed curves indicate blip-blip and blip-sojourn interactions, respectively. Adapted from Ref. [70].

An iterative, tensor-based version of BSPI has been developed,[108] which uses a fixed number of blips within the memory interval and thus enables propagation for long times. The PATHSUM code does not include an implementation of the iterative blip sum algorithm, as iteration is more efficient through the SMatPI decomposition. In that case the blip sum provides a systematic and efficient algorithm for generating the results required to construct the SMatPI matrices, which are used to propagate the RDM for long times.

The blip representation is used in all system-bath methods employed in PATHSUM (with the exception of the original QuAPI algorithm), providing significant gains in terms of storage and speed.

C. TEMPO

The influence functional in the QuAPI tensor has the structure of a matrix product operator[71] and the amplitude of a forward-backward path pair can be written in the form of a matrix product state[21] (MPS),

$$A_{\alpha_N^{\pm}\ldots\alpha_1^{\pm}\alpha_0^{\pm}} = \sum_{\beta_N=1}^{\beta_N^{\max}} \cdots \sum_{\beta_1=1}^{\beta_1^{\max}} \sum_{\beta_0=1}^{\beta_0^{\max}} S^{\alpha_N^{\pm}}_{\beta_N\beta_{N-1}} \cdots S^{\alpha_1^{\pm}}_{\beta_1\beta_0} S^{\alpha_0^{\pm}}_{\beta_0} \tag{3.6}$$

where $S^{\alpha_{k+1}^{\pm}}_{\beta_{k+1}\beta_k}$ are rectangular matrices and β_k are "bond indices".[21] In the exact representation of Eq. (3.6) the bond dimensions increase exponentially with the time index k, following the growth of the number of

paths, thus $\beta_N = n^{2N+12}$. However, the paths contain a great deal of redundant information. Singular value decomposition (SVD) offers an efficient mathematical tool for removing redundancy, and its use often reduces the values of the bond indices to manageable values by discarding components with singular values smaller than a cutoff θ_{SVD}.[71] The TEMPO method relies on SVD compression and also makes use of the blip representation to reduce storage.

The cost of the TEMPO algorithm[71] is dominated by the SVD operation in each step, which scales as $\left(\beta_k^{\max}\right)^3$. In some regimes, for example with weak-to-moderate system-bath coupling, the bond dimensions can remain manageable up to fairly long times, offering an efficient alternative to methods based on the explicit generation of paths.[72-74] The TEMPO algorithm does not appear to be parallelizable.

D. QCPI

Unlike the previous methods, the QCPI algorithm[79-81] is not restricted to harmonic baths, and thus can be used in simulations of nonadiabatic dynamics in solution or biological environments. Here we review the QCPI algorithm for Hamiltonians of the system-bath type, for which it provides numerically exact results. In this case the RDM is calculated from the expression[109]

$$\tilde{\rho}^{(N0)}_{\alpha_N^{\pm}\alpha_0^{\pm}} = \int d\mathbf{q}_0 \int d\mathbf{p}_0 W\left(\mathbf{q}_0,\mathbf{p}_0\right) \sum_{\alpha_{N-1}^{\pm}=1}^{n} \cdots \sum_{\alpha_1^{\pm}=1}^{n} G_{\alpha_N^{\pm}\alpha_{N-1}^{\pm}} \cdots G_{\alpha_1^{\pm}\alpha_0^{\pm}} e^{\frac{i}{\hbar}\Delta\Phi\left(\alpha_N^{\pm}\alpha_{N-1}^{\pm},\ldots,\alpha_0^{\pm}\right)} \tag{3.7}$$

where $\mathbf{q}_0,\mathbf{p}_0$ denote the coordinates and momenta of the bath degrees of freedom at $t=0$, which serve as initial conditions of analytically available classical trajectories, $W\left(\mathbf{q}_0,\mathbf{p}_0\right)$ is the bath Wigner phase space distribution,[7] $G_{\alpha_k^{\pm}\alpha_{k-1}^{\pm}}$ is the short-time propagator for a time-dependent system Hamiltonian augmented by the system-bath interaction along a chosen reference trajectory,[110] and $\Delta\Phi$ is the net forward-backward action that remains after the action of the reference trajectory has been subtracted (and included through the augmented system propagators). This phase contains all dynamical effects arising from the "back-reaction" (the trajectory state hops following a system path[109,111]) which are responsible for quantum interference as well as decoherence[112] and produces the imaginary part of the (properly discretized) influence functional, thus it can be included in ways analogous to those described earlier. The QCPI expression provides another fully quantum mechanical alternative to the dynamics of system-bath Hamiltonians.

The main advantage of the QCPI formulation is the inclusion of the most important part of the phase arising from the system-bath interaction in the effective system propagators $G_{\alpha_k^{\pm}\alpha_{k-1}^{\pm}}$, which are treated exactly for any value of the path integral time step. Thus, the path integral discretization needs to properly account for the remaining phase (which is associated with the quantum mechanical component of decoherence,[112] and slightly more elaborate procedures[113] are able to capture even a portion of that phase into system propagators). As a result, the QCPI expression converges with larger time steps *and* shorter memory compared to the QuAPI expression, Eq. (3.1). Further, the QCPI expression can be cast in the blip form,[109] which offers all the benefits described in part B. The phase space integral is evaluated using Monte Carlo techniques to select m_{MC} trajectory initial conditions. Because of the full quadrature evaluation of the integrals associated with the system variables, the QCPI algorithm does not encounter a sign problem.[79] The QCPI algorithm is fully parallelized by distributing trajectory initial conditions.

An iterative tensor-based decomposition of the QCPI expression has been developed.[79,113] Further, recent work[114] showed that the QCPI tensors can be eliminated by using a small matrix procedure for iteration. Effectively, QCPI results are used to generate SMatPI matrices (see the next subsection) and iteration is performed at the SMatPI cost. In favorable regimes of strong system-bath coupling, low-frequency baths and high temperature, which are challenging to QuAPI, the cost of Monte Carlo integral evaluation is offset by the gain offered by the use of a larger time step and shorter memory within the QCPI framework.[115]

In order to implement QCPI, baths described by continuous spectral densities must be discretized. An efficient "logarithmic" discretization of spectral densities is available,[42,116] which produces the desired number of discrete frequencies and coupling constants, placing the frequencies such that each mode carries the same reorganization energy (see Figure 3). As a result, the logarithmic discretization allows convergence of dynamical methods with fewer modes in comparison to spectral density discretization on a fixed interval. The number of discrete modes required for convergence increases with increasing propagation time.

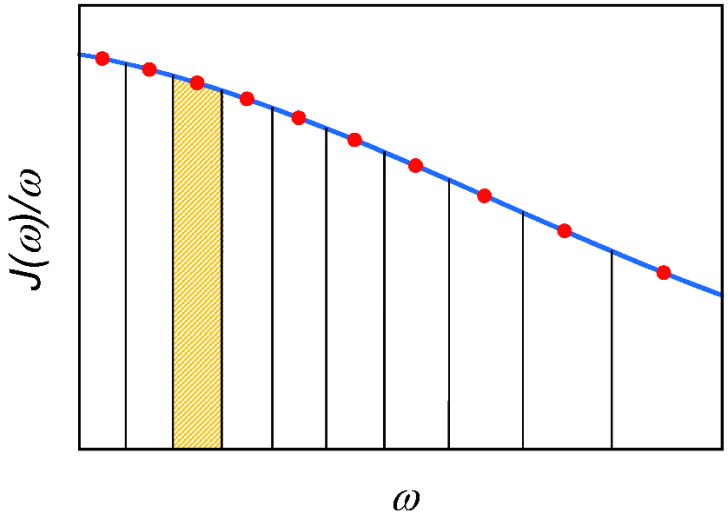

Fig. 3. Logarithmic discretization of the spectral density into modes that carry the same reorganization energy. Adapted from Ref. [116]

E. SMatPI

The SMatPI algorithm[76,77] is an analytically derived iterative decomposition of the QuAPI expression that replaces the QuAPI tensors by small matrices whose size is minimal, equal to that of the RDM. It retains the full entanglement of the path integral variables within intervals of length $r_{\max}\Delta t$, where $r_{\max}$ is the entanglement parameter which often is equal to the memory length. Figure 4 illustrates the SMatPI decomposition. At times exceeding the entanglement interval the RDM is obtained from the sum

$$\tilde{\rho}^{(N0)}_{\alpha^{\pm}_{N}\alpha^{\pm}_{0}} = \sum_{r=1}^{r_{\max}} M^{(N,N-r)}_{\alpha^{\pm}_{N}\alpha^{\pm}_{N-r}} \cdot \tilde{\rho}^{(N-r,0)}_{\alpha^{\pm}_{N-r}\alpha^{\pm}_{0}}, \quad N = r_{\max}+1,\ldots \tag{3.8}$$

The SMatPI matrices $\mathbf{M}^{(Nm)}$ have fixed dimensions $n^2 \times n^2$. These matrices are obtained by evaluating Eq. (3.1) by one of the available non-iterative methods. This process involves four full-memory, full-entanglement RDM evaluations of time length $r_{\max}\Delta t$, which can be combined and evaluated simultaneously.

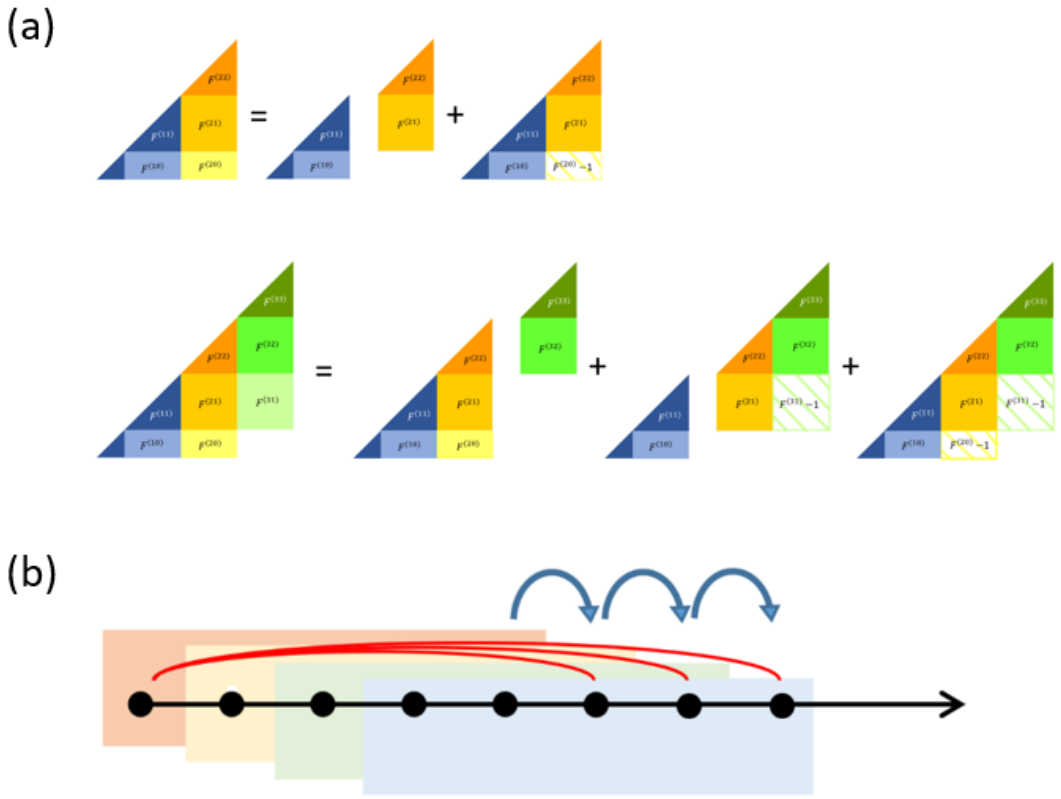

Fig. 4. (a) Schematic illustration of the influence functional factors and matrix decomposition for the case of two-step memory ($r_{\max} = 2$). The path integral time step is equal to the length of a square. Each shaded region corresponds to the area included in an influence functional factor. Triangles correspond to $F^{(kk)}$ and solid-shaded squares correspond to $F^{(kk')}$. The blue group corresponds to the factors included in $\mathbf{M}^{(10)}$, the orange regions corresponds to those included in $\mathbf{M}^{(21)}$, the yellow rectangle corresponds to $\mathbf{M}^{(20)}$, the saturated green regions correspond to $\mathbf{M}^{(32)}$, and the pale green square shows $\mathbf{M}^{(31)}$. Hatched squares and rectangles correspond to $F^{(k+2,k)} - 1$ factors. Top: decomposition of $\mathbf{U}^{(20)}$. Bottom: decomposition of $\mathbf{U}^{(30)}$. Adapted from Ref. [77] (b) Influence functional factors included beyond the entanglement interval through the extended memory procedure. Adapted from Ref. [78].

Once the SMatPI matrices have been computed, iteration involves $r_{\max}$ small matrix multiplications for each propagation step, i.e. the total cost of each step is $n^3 r_{\max}$. This makes the SMatPI algorithm *much* faster than i-QuAPI, which scales as n^{2L+2}. Perhaps most importantly, the SMatPI matrices require negligible storage in comparison to i-QuAPI tensors or the large TEMPO matrices, allowing application to multistate systems. Further, in many cases[96] the entanglement of path integral variables decays faster than the memory length, i.e. $r_{\max} \leq L$. In such situations residual memory of arbitrary length can be included

without an increase in computational effort.[78] This process is illustrated in Fig. 4. We often refer to memory exceeding the entanglement length as "extended" memory. The SMatPI algorithm has also been extended to Hamiltonians that include time-dependent fields.[117]

In addition to the path integral time step, the SMatPI algorithm employs two convergence parameters, the entanglement (or entangled memory) length $r_{\max}$ and the extended memory length L. Almost the entire cost of a calculation is associated with the full RDM calculations required to construct the SMatPI matrices. Consequently, parallelization of the calculations within the entanglement interval is very desirable, whenever possible. Once the SMatPI matrices have been constructed, iterative evaluation of Eq. (3.8) involves sequential matrix multiplications, which is extremely fast and is performed on a single processor. The following methods are available for the calculations within the entanglement interval:

1. QuAPI

As explained earlier, the cost of the non-iterative QuAPI procedure for constructing the full RDM up to the time $r_{\max}\Delta t$ scales as $n^{2r_{\max}+2}$. A variety of filtering and coarse-graining methods[56,57,60,63-65,68,118] can reduce this cost, but PATHSUM currently implements only the crudest filtering in the QuAPI module by dropping paths whose bare amplitude, combined with nearest-neighbor influence functional couplings, is smaller than a threshold θ. This threshold is often set to 0, but can be useful in very demanding calculations which can benefit from path elimination. The QuAPI path sum unit is fully parallelized with respect to the system paths.

2. BlipSum

The convergence parameters of the BlipSum component are the time step and the number b of blips. If all blips are included ($b = r_{\max}+1$), the cost scales as $\left(n^2-n+1\right)^{r_{\max}+1}$. Under strongly damped conditions (low-frequency baths with a large reorganization energy and a relatively high temperature) the number of blips can be small, i.e. $b < r_{\max}$ and sometimes $b << r_{\max}$, which leads to high efficiency. The BlipSum code is parallelized by distributing the blip configurations to multiple processors.

3. TEMPO

In some regimes (primarily weak-to-moderate dissipation) the TEMPO algorithm in the blip representation offers an efficient approach for generating results within the entanglement length, which can be used to construct SMatPI matrices. The cost is associated with the SVD procedures and is given by the third power of the maximum bond dimension, making the TEMPO algorithm attractive for systems with a small number of states. The TEMPO algorithm is not parallelizable.

The C++ module implements TEMPO by calling the ITensor libraries[119] to perform the MPO-MPS operations. These libraries are optimized and can minimize bond dimensions by performing multiple "sweeps".[120] The Fortran module does not utilize tensor libraries to perform the MPO-MPS procedure. Instead, it applies each influence functional factor and performs tensor compression by calling a standard SVD subroutine available in the LAPACK/BLAS libraries. Because of the different algorithms employed in these components the cutoff values θ_{SVD} generally are quite different in the C++ and Fortran modules.

4. QCPI

The PATHSUM code implements the QCPI expression in the form of the efficient BlipSum algorithm within the entanglement length. If all blips are retained, the cost scales as $\left(n^2-n+1\right)^{r_{\max}+1}$ multiplied by the number of trajectory initial conditions. However, since QCPI affords larger time steps and shorter memory, the

value of r_{max} tends to be considerably smaller compared to the value allowed by QuAPI-based expressions. The code is parallelized by distributing trajectory initial conditions.

F. MPI

The MPI algorithm[82-85] is ideally suited to extended systems composed of locally interacting segments with a one-dimensional topology, such as stacked molecular aggregates with Frenkel exciton interactions or chains of molecules containing interacting spins. Each molecular unit has one or more quantum (electronic or spin) states that couple to a large number of intramolecular normal mode vibrations that constitute the bath.

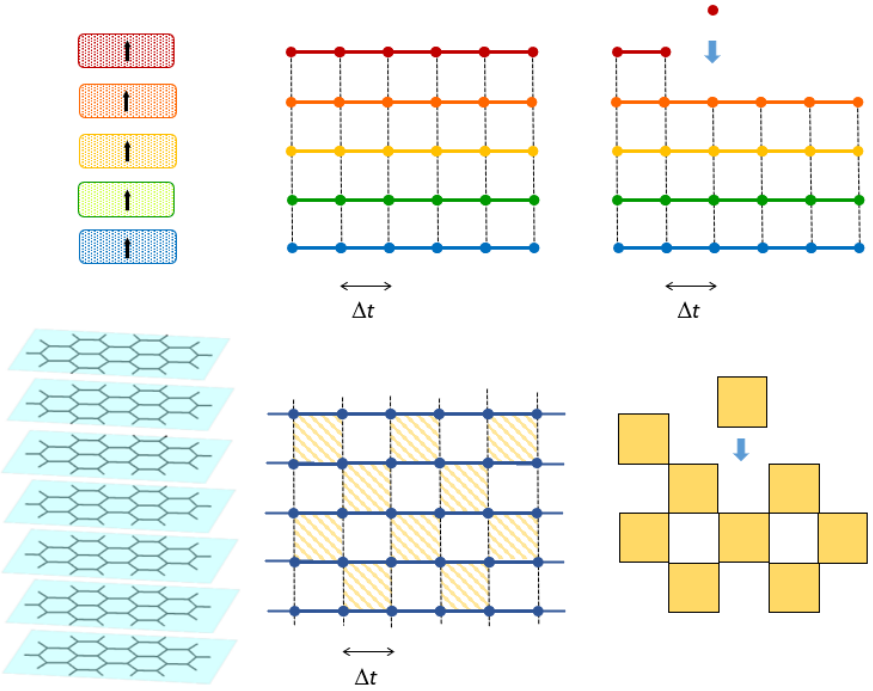

Fig. 5. Diagrammatic illustration of the MPI algorithm. The states of each unit are coupled to a bath at a finite temperature. The vertices represent the path integral variables and the dashed lines indicate couplings between units. The factorization of the MPI algorithm is also shown on the right. Top (adapted from reference [83]): Ising chain, where the coupling between adjacent monomers is diagonal in the system basis. The path integral variables of each unit are indicated with different colors in this panel. Bottom: Molecular aggregate with non-diagonal Frenkel exciton couplings. Edge units are not shown in the MPI diagram.

The MPI algorithm constructs the discretized Feynman paths in the space of the quantum states of each unit and links them to those of the adjacent unit after augmenting their amplitudes by influence functional factors arising from the local bath.[85] Once the linking is complete, the paths of the treated monomer are discarded and the process is repeated with the next pair of molecular units. The sequential nature of the MPI algorithm implies linear scaling of computational cost with aggregate size. A factorization of the MPI linking procedure[86] allows cost that scales as $nd\left(n^{2N+2}\log_n n^{2N+2}\right)$ to the leading order, where n is the number of states of each unit and d is the number of units, providing efficiency similar to that of

the fast Fourier transform (FFT) algorithm. The MPI algorithm was originally derived for diagonal interactions between units[82,83] (e.g., those of the Ising model) and has been extended to general forms including nondiagonal couplings between monomers[84] (as in the Frenkel and Heisenberg models) without a significant increase of computational cost. A SMatPI-based iterative decomposition of MPI was recently developed,[121,122] which eliminates the path storage requirements, extending calculations to long times. The PATHSUM code currently implements the non-iterative MPI algorithm for TLS-bath units, coupled through operators that are diagonal or non-diagonal in the basis of system states.

IV. Code structure

The PATHSUM code is available upon free registration at https://makrigroup.web.illinois.edu/pathsum/. A comprehensive PDF manual containing installation instructions and a documentation of the package is provided and is also available online. The C++ package depends on the LAPACK, BLAS, Armadillo and ITensor libraries,[119] while the Fortran package requires only the LAPACK and BLAS libraries. Several components of this code are embarrassingly parallelizable and utilize the "reduce" and "gather" algorithms from the *message passing interface* framework. (Note that we only refer to the modular path integral as MPI and always write the full form of "message passing interface".)

The central module or driver of the PATHSUM code reads the necessary input files that contain the various parameters and is where the user specifies which general approach is to be used. The same module performs some preliminary operations on the system Hamiltonian (such as discretizing a continuous system and computing the system propagator elements), calculates the influence functional coefficients, and prints the results computed in the peripheral modules.

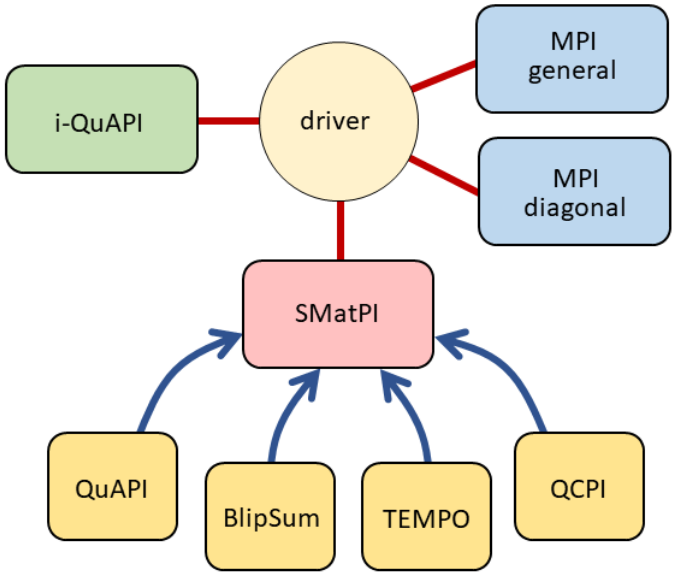

Fig. 6. Structure of the PATHSUM code. The MPI diagonal module is suitable to Ising spin chains, while the general module can be used to treat Heisenberg and Frenkel exciton Hamiltonians.

The primary method choices consist of i-QuAPI, SMatPI and MPI. If the SMatPI option is selected, the user must choose among QuAPI, BlipSum, TEMPO and QCPI for the calculations within the entanglement interval. If the propagation length is set equal to the entanglement/full memory interval, i.e., $N = r_{\max} = L$, the code produces full-memory results without resorting to iteration. If the MPI option is chosen, the user must specify the form of the Hamiltonian (diagonal for Ising spin chains or general for Heisenberg spin chains and Frenkel exciton aggregates). The overall structure of the PATHSUM code is illustrated in Figure 6.

Once the choice for the method is made, all necessary parameters for the calculation are passed on to a peripheral module of the code where the path sum is computed. For SMatPI, calculations within the entanglement interval are performed entirely in the peripheral modules. Depending on the method chosen in combination with SMatPI iteration (i.e., QuAPI, BlipSum, TEMPO or QCPI), such calculations are distributed across processors or are not parallelized. For example, calculations involving the SMatPI-QuAPI, SMatPI-BlipSum and SMatPI-QCPI combinations are distributed across as many available processors as possible with respect to the system paths, blip configurations and the trajectory initial conditions, respectively. The TEMPO-SMatPI alternative does not enable parallelization. Once the results within memory are obtained using the chosen combination, they are summed or averaged (using the "reduce" feature of the message passing interface framework for parallelized calculations), before being communicated back to the central module where the SMatPI matrices are computed and the results to long times are computed using the SMatPI method.

For the i-QuAPI and MPI choices, the entire set of results for all time points are directly obtained in the respective peripheral module before they are communicated back to the driver for printing. Additionally, the MPI method also allows parallelization with respect to the number of time points at which results are obtained. Each separate time point is a different calculation and the results for all time points are collated from different processors using the "gather" algorithm of the message passing interface framework.

V. Choosing among methods

No method is suitable for all problems. Having a broad selection of methods that work in complementary regimes allows sufficient flexibility so that converged results can be obtained for many processes of interest. Since all methods provided in PATHSUM are fully quantum mechanical and numerically exact, the only consideration in choosing a method is the computational cost, i.e. how rapidly a calculation will converge and how much array storage is required for execution.

The method-specific input parameters (which are to be varied until convergence is reached) are summarized in Table 1. In addition, the table gives the scaling of the storage and CPU requirements of each method and specifies whether or not each algorithm can be parallelized. We note that within the SMatPI module parallelization is relevant only to the operations within the entanglement interval, where most of the CPU time is spent. Once the SMatPI matrices have been constructed, propagation by matrix multiplication is extremely efficient and is performed on a single processor. In the last column we attempt to summarize the favorable regimes where (based on theoretical considerations and our experience) a method should be advantageous and relatively efficient.

Table 1. Summary of methods in PATHSUM along with their convergence parameters, storage and cost scaling, and a summary of favorable regimes for each method. The symbol n indicates the number of system states, Δt is the path itegral time step, N is the number of propagation steps desired, θ is the path acceptance threshold, $r_{\max}$ is the entangled memory (or entanglement) length, L is the extended memory length, b is the number of blips placed over all time points including $k=0$, and θ_{SVD} is the SVD cutoff parameter. The scaling of QuAPI, BlipSum and QCPI (either their full memory versions or within the SMatPI implementation) is given for $\theta=0$. The scaling of TEMPO assumes nonzero SVD cutoff parameter. If $\theta_{\mathrm{SVD}}=0$, the TEMPO bond dimension is equal to $\beta_{\max}=n^{2N+2}$ (or $\beta_{\max}=n^{2r_{\max}+2}$ in the SMatPI implementation).

method	parameters	storage	cost scaling	parallelization	favorable regimes
full-memory QuAPI	Δt, θ	n^2	n^{2N}	yes	small N and/or n
i-QuAPI	$\Delta t, r_{\max}=L$	$n^{2r_{\max}}$	$Nn^{2r_{\max}+2}$	no	high-frequency bath, weak/moderate dissipation
full-memory BlipSum	$\Delta t, b$	n^2	$\left(n^2-n+1\right)^N$	yes	strong dissipation, low-frequency bath, high temperature
full-memory TEMPO	θ_{SVD}	$n^2\beta_{\max}^2$ †	$Nn^4\beta_{\max}^3$	no	small n, weak/moderate dissipation
full-memory QCPI	$\Delta t, r_{\max}=L, m_{\mathrm{MC}}, \theta$	n^2	$m_{\mathrm{MC}}n^{2N}$	yes	strong dissipation, low-frequency bath, high temperature
SMatPI with QuAPI	$\Delta t, r_{\max}, L$, θ	$r_{\max}n^4$	$n^{2r_{\max}+2}$	yes (QuAPI)	moderately large n, intermediate/high frequency bath, weak/moderate dissipation
SMatPI with BlipSum	$\Delta t, r_{\max}, L, b$	$r_{\max}n^4$	$r_{\max}\left(n^2-n+1\right)^{r_{\max}}$	yes (BlipSum)	strong dissipation, low-frequency bath, high temperature
SMatPI with TEMPO	$\Delta t, r_{\max}, L, \theta_{\mathrm{SVD}}$	$r_{\max}n^2\beta_{\max}^2$ †	$r_{\max}n^4\beta_{\max}^3$	no	small n, weak/moderate dissipation
SMatPI with QCPI	$\Delta t, r_{\max}, L, m_{\mathrm{MC}}, \theta, \nu$	$r_{\max}n^4$	$m_{\mathrm{MC}}n^{2r_{\max}+2}$	yes (QCPI)	strong dissipation, low-frequency bath, high temperature
MPI	$\Delta t, \theta$	n^{2N}	$n\left(n^{2N}\log_n n^{2N}\right)$	yes	extended system with very large total number of states

† The pre-compressed matrices utilized in the calculation have bond dimensions given by the $\beta_{\max}$ value of the previous influence functional application multiplied by n^2, resulting in temporary matrices that are larger than those listed in the table.

VI. Examples

In this section we show some representative examples that illustrate the use of the PATHSUM code with a variety of methods. The reader is encouraged to also browse the online documentation of PATHSUM for additional details on the examples that follow. The documentation provides additional step-by-step guidance on how to converge system-bath calculations. Solely based on convenience, all examples presented below use Ohmic spectral densities of the form

$$J(\omega)=\frac{2\pi}{\Delta\sigma^2}\xi\omega e^{-\omega/\omega_c} \tag{5.1}$$

where ω_c is the cutoff frequency and the Kondo parameter ξ is a dimensionless measure of the system-bath coupling. Note that the PATHSUM code is not limited to the form of the spectral density. The user is free to provide any suitable function (which should decay to zero at high frequencies) or any number of discrete modes.

Examples 1-4 employ two-level systems (TLS, $n=2$) with the following Hamiltonian,

$$H_0=\begin{pmatrix}\varepsilon & -\hbar\Omega\\ -\hbar\Omega & -\varepsilon\end{pmatrix}. \tag{5.2}$$

where $2\hbar\Omega$ is the tunneling splitting and ε is the asymmetry parameter.

Example 1: SMatPI with QuAPI or TEMPO input; asymmetric TLS with weak dissipation

As a first example we use a mildly asymmetric TLS with $\Omega=1$ and $\varepsilon=1$, coupled to a (common) Ohmic bath with parameters $\omega_c=7.5\Omega$ and $\xi=0.1$. We set the two TLS coordinates $\sigma_1=1$, $\sigma_2=-1$, therefore $\Delta\sigma=2$, and choose the inverse temperature $\beta=5\hbar\Omega$. We select the SMatPI method with QuAPI input for the calculations within the entanglement interval.

Figure 7a shows the population of state 1 as a function of time, obtained from SMatPI calculations with the following combinations of time step and entangled memory length ($L=r_{\max}$): $\Delta t=0.6, r_{\max}2$, $\Delta t=0.4, r_{\max}=3$, $\Delta t=0.3, r_{\max}=4$, $\Delta t=0.2, r_{\max}=6$, $\Delta t=0.15, r_{\max}=8$ and $\Delta t=0.1, r_{\max}=12$, where the total memory length is fixed at 1.2 time units. It is clear that the largest time step that leads to well converged results is $\Delta t=0.2$. We therefore continue with this time step and vary the number of steps within memory. Fig. 7b shows the results with $r_{\max}=L=2,4,6,8,10,14$. The results are very well converged with $\Delta t=0.2$ and $r_{\max}=L=10$.

Next, we demonstrate the use of PATHSUM with TEMPO as the method for the calculations within the entanglement length. In this case convergence should be checked with respect to the SVD cutoff parameter as well, which determines the bond dimensions of the matrices. Fig. 7c shows see that the SMatPI+TEMPO results (obtained with the C++ package that uses the ITensor library[119]) are fully converged with SVD cutoff 10^{-6}.

The parameters of this example (weak dissipation and low temperature) are far from the optimal regime for rapid convergence of the blip series, and the maximum number of blips ($b=r_{\max}+1$, for all path integral variables at $k=0,1,\ldots,r_{\max}$) must be included for convergence. Nevertheless, the BlipSum module always offers an excellent alternative because of its superior scaling. Table 2 shows that even though BlipSum results with $b<r_{\max}+1$ oscillate wildly and do not appear to converge, they become identical to those with the QuAPI module once all blips have been included.

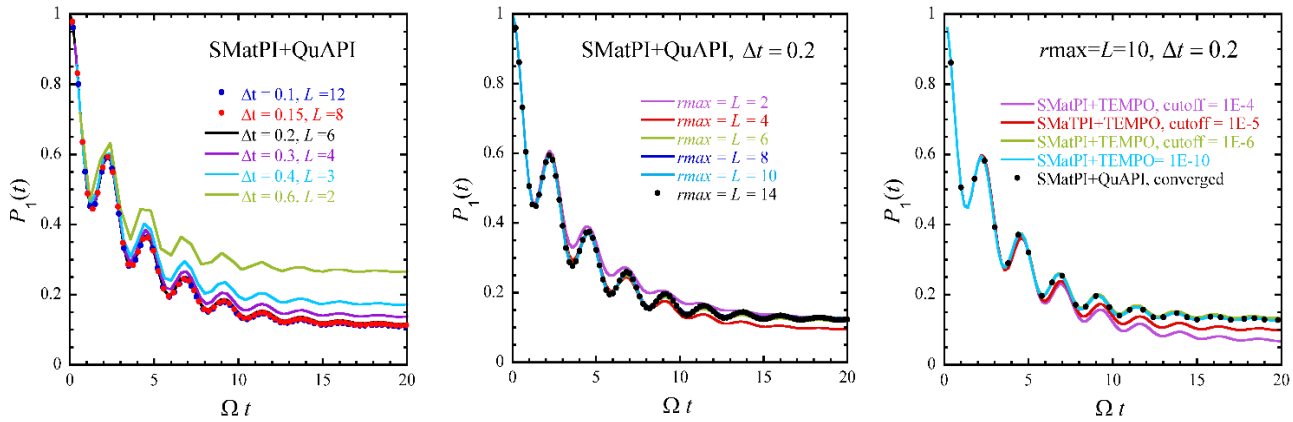

Fig. 7. Convergence of System-Bath calculation that use SMatPI with QuAPI input with respect to the path integral timestep (left) and the memory (middle). We show results for an asymmetric TLS coupled to a fast Ohmic bath at a low temperature. Converged results are obtained with $\Delta t = 0.2$ and $r_{\max} = L = 10$. The right panel shows convergence with TEMPO input.

Table 2. Comparison of SMatPI+BlipSum results at $t = 2,4$ and 6 obtained with different b values against SMatPI+QuAPI with $r_{\max} = L = 10$ and $\Delta t = 0.2$, with the parameters of example 1 (a regime unfavorable to BlipSum).

Method	$t = 2$	$t = 4$	$t = 6$
BlipSum, $b = 2$	0.32604	0.39808	0.43204
BlipSum, $b = 4$	0.35634	0.29596	0.41741
BlipSum, $b = 6$	0.45034	0.26784	0.32872
BlipSum, $b = 8$	0.54658	0.06798	0.28337
BlipSum, $b = 9$	0.57486	– 0.01805	0.17567
BlipSum, $b = 10$	0.57486	0.96078	0.99397
BlipSum, $b = 11$	0.57486	0.31921	0.20019
QuAPI	0.57486	0.31921	0.20019

Example 2: SMatPI with BlipSum and TEMPO; asymmetric TLS with strong dissipation

The next example employs a strongly asymmetric TLS described by Eq. (5.2) with $\Omega = 1$ and $\varepsilon = 5$. We set the system coordinates to $\sigma_1 = 0$, $\sigma_2 = 2$ ($\Delta\sigma = 2$), describing a bath equilibrated with respect to state 1 (the "donor" in a charge transfer reaction). The TLS is strongly coupled to a common slow bath with $\omega_c = 2$ and $\xi = 4$ at a high temperature, $\beta = 0.1$. Equilibration requires propagation for approximately 600 time steps. This combination of parameters requires a small time step and long memory, which are

challenging to older methods. SMatPI results obtained with $r_{\max} = L = 10$ do not approach the correct equilibrium value. Previous work showed that the blip series converges very rapidly in this case, and well converged results were obtained using the iterative blip sum algorithm[108] with $\Delta t = 0.075$ and $b = 3$ with $r_{\max} = L = 50$. Here we show that these results are easily and conveniently obtained using SMatPI. In Figure 8 we show converged SMatPI+BlipSum results with $r_{\max} = L = 50, b = 3$. With the present parameters, the entanglement length can be considerably shorter than the memory, so the SMatPI results with $r_{\max} = 20, L = 100, b = 4$ are indistinguishable. We also show $r_{\max} = 20, L = 100$ SMatPI results obtained with TEMPO input. The parameters of this example are challenging to TEMPO, but reaching the entanglement length $r_{\max} = 20$ is feasible (although more expensive than BlipSum). While obtaining TEMPO results over the entire propagation time or even the full memory length $L = 50 - 100$ would be difficult, SMatPI enables efficient iteration as well as the inclusion of important residual memory, which leads to nearly converged results. (Note that small errors can be seen, resulting from SVD truncation in TEMPO.) i-QuAPI calculations with the same memory produce results identical to those obtained with SMatPI but are 600 times more expensive. Similarly, a standalone TEMPO calculation which employs memory iteration analogous to i-QuAPI would be 600 times more expensive than the SMatPI+TEMPO calculation shown here.

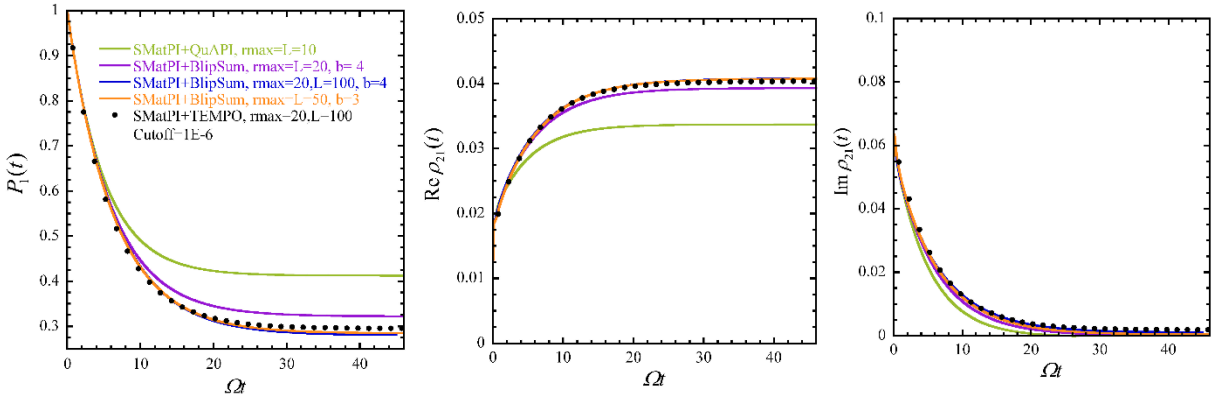

Fig. 8. System-bath results with various for an asymmetric TLS-bath system characterized by parameters of Example 1. The three panels show: (left) population of the donor state, (middle) real part of the coherence, and (right) imaginary part of the coherence, respectively. The blue and red curves are nearly identical.

Example 3: SMatPI with QuAPI, BlipSum or QCPI input; symmetric TLS with strong dissipation

This example uses a symmetric TLS with Hamiltonian of Eq. (5.2), where $\Omega = 1$ and $\varepsilon = 0$, again placing the system coordinates at 0 and 2 ($\Delta\sigma = 2$). In the present case the TLS is strongly coupled to an even slower common bath, with $\omega_c = \Omega$ and $\xi = 2$, but is now at an intermediate temperature, $\hbar\Omega\beta = 1$. This combination of parameters again generates long memory, which becomes challenging for the standalone i-QuAPI and TEMPO methods. Earlier work[78] obtained converged results SMatPI results using QUAPI input with $\Delta t = 0.25$, $r_{\max} = 18$ and extended memory $L = 100$.

Figure 9 presents SMatPI results $r_{\max}=18, L=100$ combined with QuAPI, BlipSum or QCPI input. If the BlipSum method is used instead of QuAPI to generate the SMatPI matrices, convergence is reached at $b=7$ (with the same time step $\Delta t=0.25$). Because of the lower temperature, more blips are required for converging BSPI calculations compared to Example 2, but since $b<<r_{\max}$, the BlipSum calculation is again much faster than QuAPI. Moreover, we recently showed that alternatively, if QCPI input is used, identical results are obtained (within the small Monte Carlo error from 15000 trajectory initial conditions) with a much smaller entanglement length and overall memory, $\Delta t=0.25, r_{\max}=7, L=80$. This advantage stems from the fact that QCPI incorporates the classical part of the memory exactly into the system propagator. Thus, the effective memory to be captured in the PATHSUM calculation is only that corresponding to the quantum memory, which is much shorter at this temperature. We also show a few unconverged results obtained with a smaller entanglement length in the SMatPI+QuAPI calculation, and with an insufficient number of blips.

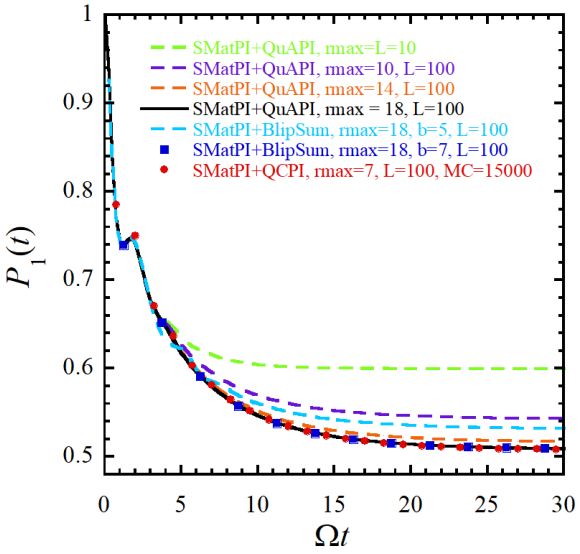

Fig. 9. Population of the donor state for a TLS coupled strongly to a slow Ohmic bath at intermediate temperatures (parameters are mentioned in the text). We show converged results obtained using the SMatPI+QuAPI, QMatPI+BlipSum and SMatPI+QCPI methods.

Example 4: SMatPI with BlipSum input; localization transition in a symmetric TLS

Next, we demonstrate the ability of the SMatPI algorithm to reproduce the localization transition of a symmetric TLS coupled to an Ohmic bath at zero temperature.[107] We use again $\Omega=1$ with $\sigma_1=1$, $\sigma_2=-1$ and choose a strongly coupled Ohmic bath characterized by $\xi=1.5$ and $\omega_c=10\Omega$ at an inverse temperature $\hbar\Omega\beta=50$, which is practically equal to zero. The SMatPI results converged easily with $\Delta t=0.05$ and $L=r_{\max}=18$. Because of the relatively large value of the entanglement parameter, the BlipSum component is more efficient than QuAPI. In spite of the strongly dissipative conditions, this regime is highly quantum mechanical, thus all blips ($b=r_{\max}+1$) must be included. Earlier expensive

calculations with the standalone TEMPO method obtained converged results with a very large memory parameter.[72]

Figure 10 shows the population of state 1 as a function of time. Following a small drop over a very short time interval, the population stabilizes at a value that is constant to four decimal places over a propagation interval of $30\,\Omega t$. Unconverged results (obtained with shorter memory or inadequate number of blips) lead to sloped population curves.

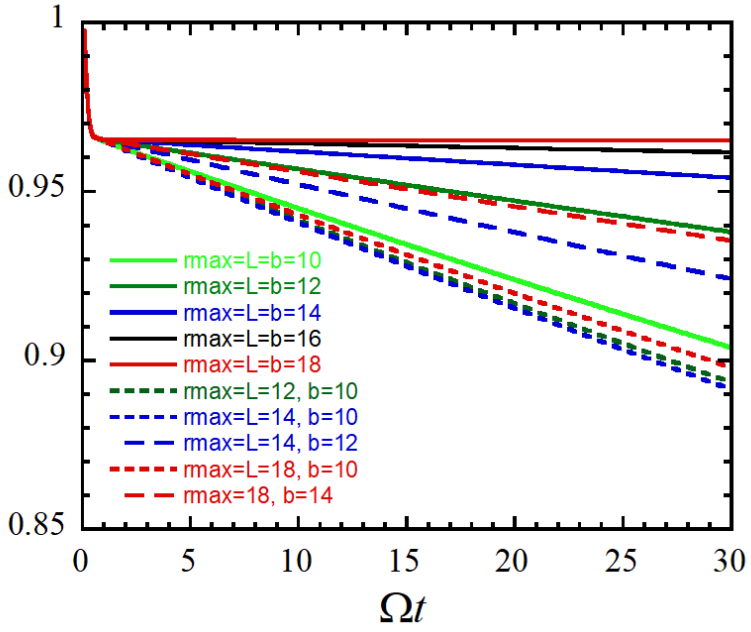

Fig. 10. Population of state 1 in a symmetric TLS coupled to an Ohmic bath with $\xi = 1.5$ and $\omega_c = 10\Omega$ at a temperature almost equal to zero. Converged SMatPI+BlipSum show the localization of the TLS.

Example 5: SMatPI with QuAPI input, 20-state system with local baths

We now move to multistate systems and present a calculation on a system with $n = 20$ states ("sites") described by the tight-binding Hamiltonian with nearest-neighbor couplings,

$$H_0 = \begin{pmatrix} 0 & -\hbar\Omega & \dots & 0 \\ -\hbar\Omega & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} \tag{5.3}$$

where $\Omega = 1$. Each site represents a singly excited electronic states of a molecular aggregate and is coupled to its own local Ohmic bath with $\omega_c = 5$ and $\xi = 0.1$ at an intermediate temperature, $\hbar\Omega\beta = 5$. While the System-Bath module does not pose any restrictions on the system Hamiltonian, in this example we use a

tight binding Hamiltonian with identical site energies and nearest-neighbor coupling parameter equal to -1. The system coordinates are set at 0 and 2 for each state, which ensures that each local bath is equilibrated with respect to the ground state of the state it is coupled to.[100] This corresponds to a Franck Condon initial excitation to the chosen initially excited state of the system. The reorganization energy between ground and excited state of each monomer is equal to $\hbar\Omega$. The calculation converges with $\Delta t = 0.25$, $r_{\max} = L = 4$ using the SMatPI algorithm with QuAPI input (see the inset of Figure 10). The converged populations of several of the 20 states of the system are shown in Fig. 11.

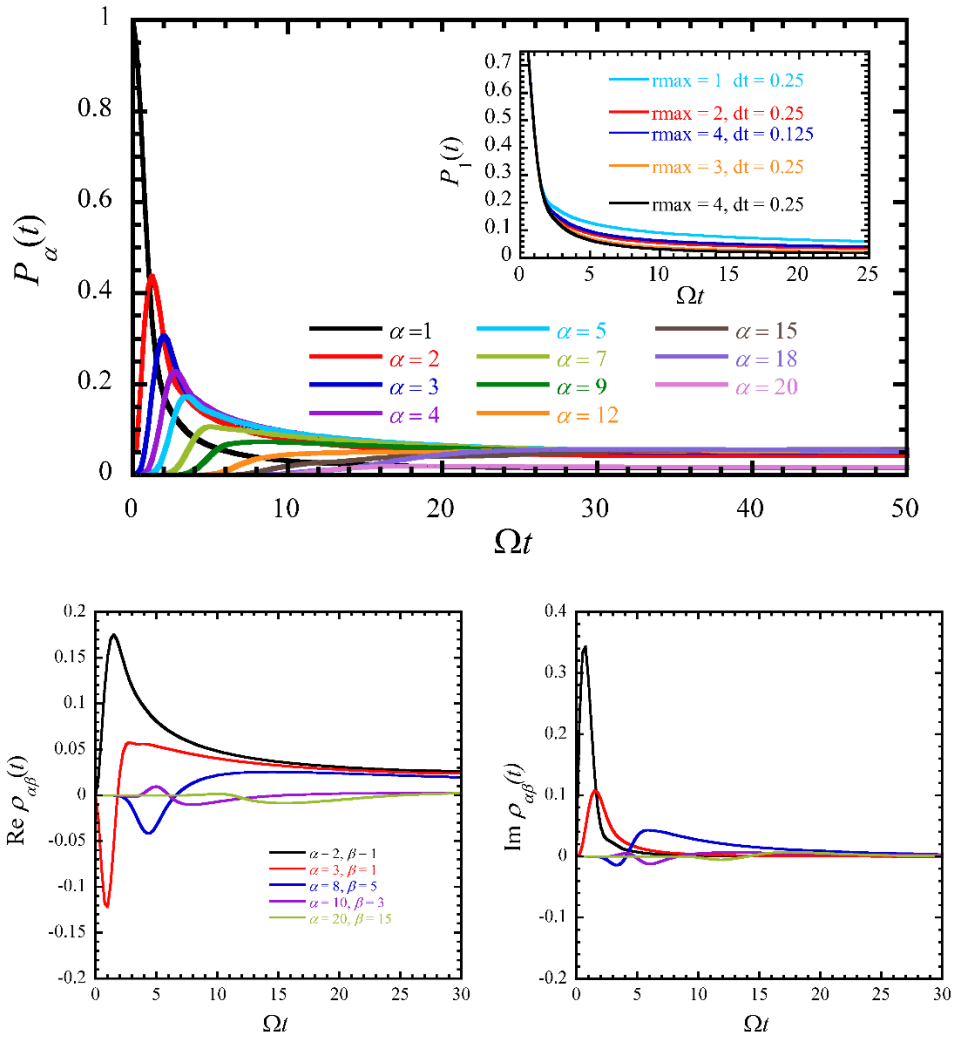

Fig. 11. Top: Populations of several states for a 20-level system coupled to identical local Ohmic baths at an intermediate temperature (see parameters mentioned in the text) obtained using the SMATPI+QuAPI method. We also show the convergence of the population of monomer 1 with respect to the path integral timestep (comparison of red and blue curves) and memory (cyan, red, orange and black curves) in the inset. Bottom: Four important coherences (left: real parts, right: imaginary parts). Only converged results obtained using the SMatPI+QuAPI method are shown.

Since the System-Bath module of PATHSUM calculates the entire RDM at all times, the user may simultaneously examine the coherences (off-diagonal elements of the RDM) between the different system states as well. Recent work[123] has shown that coherences carry important dynamical information. For example, the instantaneous rate of change of a population is given by the sum of coherences that involve the given state and all other states, weighted by Hamiltonian coupling elements.[124] The visualization of the entire RDM through coherence maps[123] (spatial snapshots of the RDM) can be used to infer the kinetic pathways in the system-bath dynamics.[125] In Figure 11 we also show the real and imaginary parts of four coherences.

Example 6: MPI for Ising chain of 10 spins coupled to Ohmic baths

Last, we study a chain of $n=10$ TLSs with Ising-type nearest neighbor couplings according to the Hamiltonian of Eq. (2.11) with $\Omega=1$ and $J_z=-0.2\Omega$. Each TLS is additionally coupled to an Ohmic bath (all Ohmic baths are assumed to be identical) with $\omega_c=5$ and $\xi=0.3$ at an intermediate temperature $\beta=1$, and the system coordinates for each TLS are kept at +1 and −1. We seek $\langle\sigma_z(t)\rangle$ for the first TLS in the chain (on either side, due to symmetry). The total number of states in the system alone is 2^{10}, taking this calculation beyond the scope of system-bath methods, thus the MPI algorithm needs to be used.

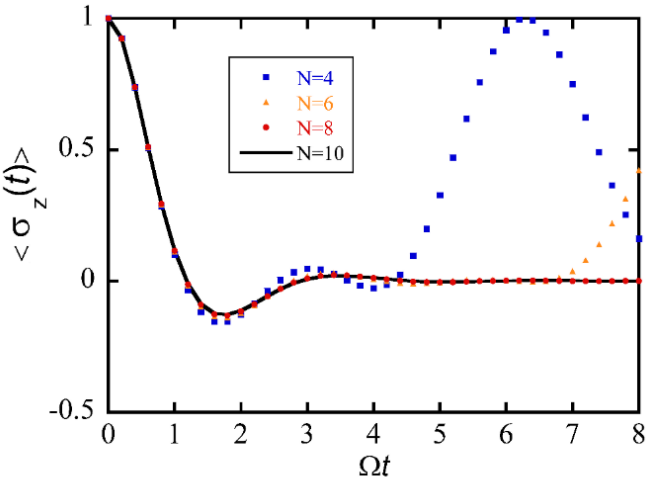

Fig. 12. $\langle\sigma_z(t)\rangle$ for the first in a chain of 10 TLSs, each coupled to a (local) Ohmic bath. We show the convergence with respect to N from an MPI calculation using the diagonal algorithm.

MPI calculations use a plotting grid of fixed time separation (set by the user). In this case, we choose the time grid with an interval of 0.2 units over the range up to 8 time units (i.e. 41 time values). The only convergence parameter that the user needs to set is N, the number of path integral time steps to be used in each MPI calculation. In Figure 12 we vary N in different calculations to test convergence. Since the value of N is fixed, the results obtained at different time points use a different path integral time step given by t/N, where t is a time value. Therefore, results at shorter times are always very accurate. Fig.

12 shows that results obtained with $N = 4$ develop small errors early on and exhibit a spurious population growth after $t \simeq 5$). Results obtained with $N = 6$ only show such behavior after $t \simeq 7$. Increasing N to 8 and to 10, leads to results that are converged over the time interval shown in the figure.

VII. Concluding Remarks

In this paper we reported the release of PATHSUM, a new software suite that implements fully quantum mechanical real-time path integral algorithms to study the dynamics of (multi-)system-bath Hamiltonians. The package has been built to be equally useful to the non-expert looking to run state-of-the-art computations that are relevant to the quantum dynamics of chemical and biological processes such as electron/proton/energy transfer or spin dynamics, without having to write their own code. The package has two modules, a System-Bath module that deals with a single system coupled to one common or many local baths, and an Extended-System module that addresses the problem of many quantum systems in an molecular aggregate coupled to their own harmonic environments. For each method that is available in the two modules of our package, we provide a concise but complete overview in this paper to guide the new user, as well as detailed references to aid the more inquisitive reader. We illustrate the use of our software through six tutorial examples across the two modules and provide insights regarding the comparative advantages of the different methods whenever possible. We also emphasize the complementarity of different methodologies available in PATHSUM, thus providing the user with access to different regimes of condensed phase quantum dynamics.

We encourage users to embrace the PATHSUM suite and help us improve it through their valuable feedback, as we continue to build different functionalities and methods into this package, as and when they are developed in the years to come. As the field of exact quantum dynamics keeps becoming increasingly viable and valuable in the research of our times, we hope this suite of exact path integral methods will help users gain new insights into the dynamics of open quantum systems.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Award CHE-1955302.

References

1. A. G. Redfield, IBM Journal of Research and Development **1**, 19-31 (1957).
2. W. T. Pollard, A. K. Felts and R. A. Friesner, in *Advances in Chemical Physics* (1996), Vol. 93, pp. 77-134.
3. J. C. Tully and R. K. Preston, J. Chem. Phys. **55**, 562-572 (1971).
4. J. C. Tully, J. Chem. Phys. **93**, 1061-1071 (1990).
5. J. C. Tully, The Journal of Chemical Physics **137** (22), 22A301 (2012).
6. S. Hammes-Schiffer, Acc. Chem. Res. **42**, 1881-1889 (2009).
7. E. Wigner, Chem. Phys. **5**, 720 (1937).
8. E. J. Heller and R. C. Brown, J. Chem. Phys. **75**, 1048-1050 (1981).
9. H. Wang, X. Sun and W. H. Miller, J. Chem. Phys. **108**, 9726-9736 (1998).
10. X. Sun, H. Wang and W. H. Miller, J. Chem. Phys. **109**, 7064-7074 (1998).
11. W. H. Miller, J. Phys.Chem. **103**, 9384-9387 (1999).
12. J. A. Poulsen, G. Nyman and P. J. Rossky, J. Chem. Phys. **119** (23), 12179-12193 (2003).
13. J. Shao and N. Makri, J. Phys. Chem. A **103**, 7753-7756 (1999).
14. N. Makri, A. Nakayama and N. Wright, J. Theor. Comp. Chem. **3**, 391-417 (2004).
15. J. Liu, J. Chem. Phys. **140** (2014).
16. J. Liu and Z. Zhang, J. Chem. Phys. **144**, 034307 (2016).
17. H.-D. Meyer, U. Manthe and L. S. Cederbaum, Chem. Phys. Lett. **165**, 73-78 (1990).
18. M. H. Beck, A. Jäckle, G. A. Worth and H.-D. Meyer, Phys. Rep. **324**, 1-105 (2000).
19. G. A. Worth, H. D. Meyer, H. Köppel, L. S. Cederbaum and I. Burghardt, International Reviews in Physical Chemistry **27** (3), 569-606 (2008).
20. H. Wang and M. Thoss, The Journal of Chemical Physics **119** (3), 1289-1299 (2003).
21. R. Orús, Annals of Physics **349**, 117-158 (2014).
22. J. Ren, Z. Shuai and G. Kin-Lic Chan, Journal of Chemical Theory and Computation **14** (10), 5027-5039 (2018).
23. S. R. White, Phys. Rev. Lett. **69**, 2863 (1992).
24. U. Schollwöck, Annals of Physics **326**, 96-192 (2011).
25. R. P. Feynman, Rev. Mod. Phys. **20**, 367-387 (1948).
26. R. P. Feynman and A. R. Hibbs, *Quantum Mechanics and Path Integrals*. (McGraw-Hill, New York, 1965).
27. L. S. Schulman, *Techniques and applications of path integration*. (John Wiley and Sons, New York, 1981).
28. J. D. Doll, D. L. Freeman and T. L. Beck, Adv. Chem. Phys. **78**, 61-127 (1990).
29. N. Makri, Comp. Phys. Comm. **63**, 389-414 (1991).
30. R. P. Feynman, *Statistical Mechanics*. (Addison-Wesley, Redwood City, 1972).
31. D. Chandler and P. G. Wolynes, J. Chem. Phys. **74**, 4078-4095 (1981).
32. K. Binder and D. W. Heermann, *Monte Carlo simulation in statistical physics*. (Springer-Verlag, 1988).
33. D. M. Ceperley, Rev. Mod. Phys. **67**, 279-355 (1995).
34. T. E. Markland and M. Ceriotti, Nature Reviews Chemistry **2** (3), 0109 (2018).
35. G. A. Voth, Adv. Chem. Phys. **XCIII**, 135 (1996).
36. S. Habershon, D. E. Manolopoulos, T. E. Markland and T. F. Miller, Annu. Rev. Phys. Chem. **64**, 387-413 (2013).
37. N. Ananth, Annual Review of Physical Chemistry **73** (1), 299-322 (2022).
38. T. J. H. Hele, M. J. Willatt, A. Muolo and S. C. Althorpe, J. Chem. Phys. **142**, 191101 (2015).
39. S. Althorpe, The European Physical Journal B **94** (2021).

40. R. A. Marcus and N. Sutin, Biochim. Biophys. Acta **811**, 265-322 (1985).
41. D. Chandler, *Introduction to Modern Statistical Mechanics*. (Oxford University Press, New York, 1987).
42. N. Makri, J. Phys. Chem. **103**, 2823-2829 (1999).
43. R. P. Feynman and F. L. Vernon, Ann. Phys. **24**, 118-173 (1963).
44. R. Zwanzig, J. Stat. Phys. **9**, 215-220 (1973).
45. D. Thirumalai, E. J. Bruskin and B. J. Berne, J. Chem. Phys. **79**, 5063-5069 (1983).
46. B. J. Berne and D. Thirumalai, Annu. Rev. Phys. Chem. **37**, 401-424 (1986).
47. A. Ishizaki and Y. Tanimura, Journal of the Physics Society Japan **74** (12), 3131-3134 (2005).
48. H. Liu, L. Zhu, S. Bai and Q. Shi, The Journal of Chemical Physics **140** (13), 134106 (2014).
49. N. Makri, Chem. Phys. Lett. **193**, 435-444 (1992).
50. M. Topaler and N. Makri, Chem. Phys. Lett. **210**, 448 (1993).
51. D. E. Makarov and N. Makri, Chem. Phys. Lett. **221**, 482-491 (1994).
52. N. Makri and D. E. Makarov, J. Chem. Phys. **102**, 4600-4610 (1995).
53. N. Makri and D. E. Makarov, J. Chem. Phys. **102**, 4611-4618 (1995).
54. N. Makri, J. Math. Phys. **36**, 2430-2456 (1995).
55. N. Makri, J. Phys. Chem. **97**, 2417-2424 (1993).
56. E. Sim and N. Makri, Chem. Phys. Lett. **249**, 224-230 (1996).
57. E. Sim and N. Makri, Comp. Phys. Commun. **99**, 335-354 (1997).
58. A. A. Golosov, R. A. Friesner and P. Pechukas, J. Chem. Phys. **110**, 138-146 (1999).
59. A. A. Golosov, R. A. Friesner and P. Pechukas, J. Chem. Phys. **112**, 2095-2105 (2000).
60. E. Sim, J. Chem. Phys. **115**, 4450-4456 (2001).
61. D. Segal, D. R. Reichman and A. J. Millis, Phys. Rev. B **76**, 195316 (2007).
62. P. Nalbach, J. Eckel and M. Thorwart, New Journal of Physics **12**, 065043 (2010).
63. R. Lambert and N. Makri, Mol. Phys. **110**, 1967-1975 (2012).
64. N. S. Dattani, AIP Advances **2**, 012121 (2012).
65. N. Makri, Mol. Phys. **110**, 1001-1007 (2012).
66. P. L. Walters, T. Banerjee and N. Makri, J. Chem. Phys. **143**, 074112 (2015).
67. H.-G. Duan, A. G. Dijkstra, P. Nalbach and M. Thorwart, Physical Review E **92** (4), 042708 (2015).
68. M. Richter and B. P. Fingerhut, J. Chem. Phys. **146**, 214101 (2017).
69. Y. Sato, The Journal of Chemical Physics **150** (22), 224108 (2019).
70. N. Makri, J. Chem. Phys. **141**, 134117 (2014).
71. T. P. Straatsma, B. W. Lovett and P. Kirton, New Journal of Physics **19**, 093009 (2017).
72. A. Strathearn, P. Kirton, D. Kilda, J. Keeling and B. W. Lovett, Nature Communications **9**, 3322 (2018).
73. M. R. Jørgensen and F. A. Pollock, Phys Rev Lett **123** (24), 240602 (2019).
74. D. Gribben, A. Strathearn, J. Iles-Smith, D. Kilda, A. Nazir, B. W. Lovett and P. Kirton, Physical Review Research **2** (1), 013265 (2020).
75. A. Bose and P. L. Walters, The Journal of Chemical Physics **156** (2), 024101 (2022).
76. N. Makri, J. Chem. Phys. **152**, 041104 (2020).
77. N. Makri, J. Chem. Theory and Comput. **16**, 4038–4049 (2020).
78. N. Makri, J. Chem. Theory and Comput. **17**, 1-6 (2021).
79. R. Lambert and N. Makri, J. Chem. Phys. **137**, 22A553 (2012).
80. R. Lambert and N. Makri, J. Chem. Phys. **137**, 22A552 (2012).
81. N. Makri, International Journal of Quantum Chemistry **115**, 1209-1214 (2015).
82. N. Makri, J. Chem. Phys. **148**, 101101 (2018).
83. N. Makri, J. Chem. Phys. **149**, 214108 (2018).
84. S. Kundu and N. Makri, J. Chem. Phys. **151**, 074110 (2019).

85. S. Kundu and N. Makri, J. Chem. Phys. **153**, 044124 (2020).
86. S. Kundu and N. Makri, Mol. Phys. **119**, e1797200 (2021).
87. R. Dani and N. Makri, J. Chem. Phys. **155**, 234705 (2021).
88. K. B. Arnardottir, A. J. Moilanen, A. Strashko, P. Törmä and J. Keeling, Phys Rev Lett **125** (23), 233603 (2020).
89. G. E. Fux, E. P. Butler, P. R. Eastham, B. W. Lovett and J. Keeling, Phys Rev Lett **126** (20), 200401 (2021).
90. S. Kundu and N. Makri, Annual Review of Physical Chemistry **73** (1), 349-375 (2022).
91. R. Dani and N. Makri, The Journal of Physical Chemistry C **126** (25), 10309-10319 (2022).
92. H.-G. Duan, V. I. Prokhorenko, R. J. Cogdell, K. Ashraf, A. L. Stevens, M. Thorwart and R. J. D. Miller, Proceedings of the National Academy of Sciences, 201702261 (2017).
93. H.-G. Duan, P. Nalbach, R. J. D. Miller and M. Thorwart, Photosynthesis Research **144** (2), 137-145 (2020).
94. S. Kundu and N. Makri, J. Phys. Chem. Lett. **11**, 8783-8789 (2020).
95. A. Bose and P. L. Walters, Journal of Chemical Theory and Computation **18** (7), 4095-4108 (2022).
96. S. Kundu, R. Dani and N. Makri, J. Chem. Phys. **157**, 115101 (2022).
97. S. Kundu, R. Dani and N. Makri, Science Advances **8** (43), eadd0023 (2022).
98. Z. Bacic and J. C. Light, Annu. Rev. Phys. Chem. **40**, 469-498 (1989).
99. J. Echave and D. C. Clary, J. Chem. Phys. **190**, 225-230 (1992).
100. N. Makri, J. Chem. Phys. **158**, 144107 (2023).
101. A. O. Caldeira and A. J. Leggett, Physica A **121**, 587-616 (1983).
102. J. Frenkel, Phys. Rev. **37**, 17 (1931).
103. V. May and O. Kühn, *Charge and energy transfer dynamics in molecular systems*, 3rd ed. (Wiley, 2011).
104. N. Makri, J. Phys. A **56**, 144001 (2023).
105. N. Makri, J. Phys. Chem. **102**, 4414-4427 (1998).
106. J. Shao and N. Makri, J. Chem. Phys. **116**, 507-514 (2002).
107. A. J. Leggett, S. Chakravarty, A. T. Dorsey, M. P. A. Fisher, A. Garg and W. Zwerger, Rev. Mod. Phys. **59**, 1-85 (1987).
108. N. Makri, J. Chem. Phys. **146**, 134101 (2017).
109. N. Makri, Faraday Discuss. **195**, 81-92 (2016).
110. T. Banerjee and N. Makri, J. Phys. Chem. **117**, 13357-13366 (2013).
111. F. Wang and N. Makri, J. Chem. Phys. **150**, 184102 (2019).
112. N. Makri, Chem. Phys. Lett. **593**, 93-103 (2014).
113. P. L. Walters and N. Makri, J. Chem. Phys. **144**, 044108 (2016).
114. S. Kundu and N. Makri, J. Phys. Chem. Lett. **13**, 3492–3498 (2022).
115. S. Chatterjee and N. Makri, J. Phys. Chem. B (2019).
116. P. L. Walters, T. C. Allen and N. Makri, J. Comput. Chem. **38**, 110-115 (2017).
117. N. Makri, The Journal of Physical Chemistry A **125** (48), 10500-10506 (2021).
118. N. S. Dattani, Comp. Phys. Comm. **184**, 2828-2833 (2013).
119. M. Fishman, S. R. White and E. M. Stoudenmire, (2022).
120. S. Paeckel, T. Köhler, A. Swoboda, S. R. Manmana, U. Schollwöck and C. Hubig, Annals of Physics **411**, 167998 (2019).
121. N. Makri, Journal of Chemical Theory and Computation **17**, 3825-3829 (2021).
122. N. Makri, Phys. Chem. Chem. Phys. **23**, 12537-12540 (2021).
123. R. Dani and N. Makri, The Journal of Physical Chemistry B **126** (45), 9361-9375 (2022).
124. R. Dani and N. Makri, The Journal of Physical Chemistry Letters **13** (34), 8141-8149 (2022).

125. R. Dani, S. Kundu and N. Makri, The Journal of Physical Chemistry Letters **14**, 3835-3843 (2023).