Department of Mathematics
TUM School of Computation, Information and Technology
Technical University of Munich

Benchmarking a Tree Tensor Network Algorithm for the HOPS-Method to Simulate Open non-Markovian Quantum Sytems

Benjamin Sappler

Thesis for the attainment of the academic degree

Bachlor of Science

at the TUM School of Computation, Information and Technology of the Technical University of Munich

Supervisor:
Prof. Dr. Christian Mendl

Advisor:
Richard Milbradt

Submitted:
Munich, June 15, 2023

I hereby declare that this thesis is entirely the result of my own work except where otherwise indicated. I have only used the resources given in the list of references.

Munich, June 15, 2023 Benjamin Sappler

Zusammenfassung

Eine kurze Zusammenfassung der Arbeit auf Deutsch.

Abstract

A brief abstract of this thesis in English.

Contents

1 Introduction

Open quantum systems, ie. quantum systems that are interacting with an environment, are important for modelling many complex processes, like the decoherence of quantum computers and processes in the field of chemical physics. Most of the time, these systems cannot be described analytically, but only numerically. Thus, the field of open quantum system simulation is very important.
A popular approach for the simulation of open quantum systems is the Hierarchy Of Pure States (HOPS) [1]. With HOPS, one can integrate the non-Markovian quantum state diffusion equation and simulate non-markovian open quantum systems. Recently, it was shown that the popular Matrix Product State formalism can be used to derive a Hierarchy Of Matrix Pure States (HOMPS), which can drastically improve the memory requirements for HOPS.
The goal of this thesis is to implement and benchmark both HOPS and HOMPS in detail. I start by giving a derivation of HOPS from the non-markovian quantum state diffusion equation in section **??**. Next, I explain in detail how HOPS can be implemented and test the method on the spin boson model in section 4. In section 4, i derive and implement HOMPS, and test it on the spin boson model as well. Finally, I give a conclusion and some references on how to further improve the methods in section 5.
My implementation of both HOPS and HOMPS, including the code that was used to generate all plots in this thesis, is openly available under [2].

2 Theory

2.1 Non-Markovian Quantum State Diffusion (NMQSD)

To simulate an open quantum system, we first need to model the system, its environment, and the interaction between them. We will consider a system S linearly coupled to a bath B of harmonic oscillators. We can split the Hamiltonian of such a model into a system, bath, and interaction part

$$\hat{H} = \hat{H}_\mathrm{S} \otimes \mathbb{1}_\mathrm{B} + \mathbb{1}_\mathrm{S} \otimes \hat{H}_\mathrm{B} + \hat{H}_\mathrm{int}.$$

We assume that the bath consists of K harmonic oscillators, which couple linearly to the system. The bath Hamiltonian is then given by

$$\hat{H}_\mathrm{B} = \sum_{k=1}^{K} \nu_k \hat{a}_k^\dagger \hat{a}_k,$$

where $\hat{a}_k^\dagger$, $\hat{a}_k$ are the bosonic creation and annihilation operators of the kth harmonic oscillator, and ν_k are constants. The interaction Hamiltonian can be written as

$$\hat{H}_\mathrm{int} = \sum_{k=1}^{K} \left(\gamma_k^* \hat{L} \otimes \hat{a}_k^\dagger + \mathrm{h.c.} \right)$$

with constants γ_k. The system operator $\hat{L}$ describes the coupling of the system to the bath modes.
In the context of open systems it is useful to define the *bath correlation function*

$$\alpha(\tau) = \frac{1}{\pi} \int_0^\infty \mathrm{d}\omega S(\omega) \left[\coth\left(\frac{\omega}{2T}\right) \cos(\omega\tau) - i \sin(\tau) \right] \tag{2.1}$$

with the *spectral density* $S(\omega)$. The bath correlation function fully characterizes the influence of the environment at temperature T [3] and is connected to the constants ν_k and γ_k.
We are interested in the dynamics of the system S, which can be described in terms of the reduced density matrix

$$\rho(t) = \mathrm{tr}_\mathrm{B} \{ \rho_\mathrm{tot}(t) \},$$

where $\rho_\mathrm{tot}(t)$ is the density matrix of the total system (system and bath) at time t. $\mathrm{tr}_\mathrm{B}\{\cdots\}$ denotes the trace over all bath degrees of freedom. We assume that the total system is initially in the state

$$\rho_\mathrm{tot}(0) = \rho_\mathrm{S}(0) \otimes \rho_{\mathrm{B,\,therm}},$$

where the bath is in the thermal state

$$\rho_\mathrm{therm}^B = \frac{e^{-\hat{H}_\mathrm{B}/T}}{Z_\mathrm{B}}$$

with the partition function $Z_\mathrm{B} = \mathrm{tr}_\mathrm{B}\left\{ e^{-\hat{H}_\mathrm{B}/T} \right\}$.
The idea of Non-Markovian Quantum State Diffusion (NMQSD) is that one can obtain the reduced density matrix $\rho(t)$ from an average over pure states

$$\rho(t) = \mathbb{E}\left[|\Psi_t(z)\rangle \langle \Psi_t(z)| \right]. \tag{2.2}$$

The pure states $|\Psi_t(z)\rangle \in \mathcal{H}_\mathrm{S}$ are vectors in the system Hilbert space that depend on a complex gaussian stochastic process $z\colon t \to z_t \in \mathbb{C}$. The expectation value $\mathbb{E}[\cdots]$ can then be computed by taking

the average over different realizations of z. For NMQSD, the stochastic process must have the following properties:

$$\begin{aligned} \mathbb{E}\left[z_t\right] &= \mathbb{E}\left[z_t^*\right] = 0, \\ \mathbb{E}\left[z_t z_s\right] &= 0, \\ \mathbb{E}\left[z_t z_s^*\right] &= \alpha\left(t-s\right). \end{aligned} \tag{2.3}$$

Each pure state $|\Psi_t\left(z\right)\rangle$ starts off in the same initial state $|\Psi_{t=0}\left(z\right)\rangle = |\Psi_0\rangle$ and then evolves according to the Non-Markovian Quantum State Diffusion (NMQSD) equation [4, 5]

$$\frac{\partial}{\partial t}|\Psi_t\rangle = -i\hat{H}_{\mathrm{S}}|\Psi_t\rangle + \hat{L}z_t^*|\Psi_t\rangle - \hat{L}^\dagger \int_0^t \mathrm{d}s\alpha\left(t-s\right)\frac{\delta|\Psi_t\rangle}{\delta z_s^*}, \tag{2.4}$$

where we omitted the explicit dependency of $|\Psi_t\rangle$ on z due to brevity. It is important to realize that the NMQSD equation describes the dynamics in terms of a stochastic expectation value of pure states, whereas regular master equations involve a non-stochastic differential equation of the reduced density matrix. The advantage of the NMQSD equation is that it is often easier to work with pure states than with density matrices.

2.2 Hierarchy Of Pure States (HOPS)

2.2.1 linear HOPS

The NMQSD equation (2.4) cannot easily be solved numerically due to the functional derivative. However, one can bring the equation into a hierarchically structured set of differential equations, the Hierarchy of Pure States (HOPS) [1], which can then be integrated numerically. In this section, we will derive the linear HOPS equation, mainly following the derivation in [6], but the same result is also obtained in [1, 7].
We will start by approximating the bath correlation function (BCF) (2.1) by a finite sum of exponentials

$$\alpha\left(\tau\right) \approx \sum_{k=1}^{K}\alpha_k\left(\tau\right) \coloneqq \sum_{k=1}^{K} g_k e^{-\omega_k\tau},$$

with constants g_k and ω_k. The total number of terms K corresponds to the number of harmonic oscillators coupling to the system. Such an approximation of the bath correlation function is possible for many systems of interest, and a specific example is given in appendix A, where we approximate the BCF of the spin-boson model using a Matsubara expansion.
We will work with the discrete version of the NMQSD equation [6]

$$\Psi_{t+1} = \Psi_t + \Delta\cdot\left\{-iH_{\mathrm{S}} + \hat{L}z_t^* - \hat{L}^\dagger\sum_{s=0}^{t-1}\alpha\left(t\Delta - s\Delta\right)\frac{\partial}{\partial z_s^*}\right\}\Psi_t, \tag{2.5}$$

where $t, s \in \mathbb{N}_0$, we introduced the time step Δ, and $z \coloneqq \{z_1, z_2, \dots\}$ now is a discrete stochastic process. One can easily see that the NMQSD equation (2.4) is recovered if the limit $\Delta \longrightarrow 0$ is taken. The reason for using the discrete version of the NMQSD equation is that we can replace the functional derivative with an ordinary derivative, making the derivation much more intuitive.
We define the operator

$$D_k^t \coloneqq \sum_{s=0}^{t-1}\alpha_k\left(t\Delta - s\Delta\right)\frac{\partial}{\partial z_s^*} = g_k\sum_{s=0}^{t-1}e^{-\omega_k\left(t-s\right)\cdot\Delta}\frac{\partial}{\partial z_s^*}$$

and the auxillary states

$$\Psi_t^{(\mathbf{n})} \coloneqq \prod_{k=1}^{K}\left(D_k^t\right)^{n_k}\Psi_t, \tag{2.6}$$

using an index vector $\mathbf{n} \in \mathbb{N}_0^K$. The physical pure state is recovered when setting the index vector to zero, $\Psi_n = \Psi_n^{(\mathbf{0})}$. Using these definitions, we can rewrite the discrete NMQSD equation (2.5):

$$\Psi_{t+1}^{(\mathbf{0})} = \Psi_t^{(\mathbf{0})} + \Delta \cdot \left(-iH_\mathrm{S} + \hat{L} z_t^* - \hat{L}^\dagger \sum_{k=1}^{K} D_k^t\right) \Psi_t^{(\mathbf{0})} = \Psi_t^{(\mathbf{0})} + \Delta \cdot \left(-iH_\mathrm{S} + \hat{L} z_t^*\right) \Psi_t^{(\mathbf{0})} - \hat{L}^\dagger \sum_{k=1}^{K} \Psi_t^{(\mathbf{0}+\mathbf{e}_k)},$$

where $\mathbf{e}_k$ is the kth unit vector.
Our next goal is to derive an equation of motion for an arbitrary auxillary state $\Psi_t^{(\mathbf{n})}$. Using equation (2.6) we can write

$$\Psi_{t+1}^{(\mathbf{n})} = \prod_{k=1}^{K} \left(D_k^{t+1}\right)^{n_k} \Psi_{t+1}. \tag{2.7}$$

We can expand

$$D_k^{t+1} = (1 - \omega_k \cdot \Delta) \left(g_k \frac{\partial}{\partial z_t^*} + D_k^t\right) + O\left(\Delta^2\right)$$

and hence

$$\left(D_k^{t+1}\right)^{n_k} = (1 - n_k \omega_k \Delta) \left(g_k \frac{\partial}{\partial z_t^*} + D_k^t\right)^{n_k} + O\left(\Delta^2\right).$$

To further simplify equation (2.7), we can use the fact that the state at time t, $\Psi_t^{(\mathbf{n})}$, depends only on the stochastic variables $z_1, z_2, \ldots, z_{t-1}$, but not on z_t, which we can write as $\Psi_t^{(\mathbf{n})} = \Psi^{(\mathbf{n})}\left(z|_0^{t-1}\right)$. It follows that $\frac{\partial}{\partial z_t^*} \Psi_t^{(\mathbf{n})} = 0$. Using equation (2.5) we can see $\frac{\partial^2}{\partial z_t^{*2}} \Psi_{t+1}^{(\mathbf{n})} = 0$ and therefore

$$\left(D_k^{t+1}\right)^{n_k} \Psi_{t+1} = (1 - n_k \omega_k \Delta) \left(n_k g_k \left(D_k^t\right)^{n_k - 1} \frac{\partial}{\partial z_t^*} + \left(D_k^t\right)^{n_k}\right) + O\left(\Delta^2\right). \tag{2.8}$$

Inserting equations (2.5) and (2.8) into equation (2.7) and performing some additional algebra, one arrives at

$$\Psi_{t+1}^{(\mathbf{n})} = \Psi_t^{(\mathbf{n})} + \Delta \cdot \left(-iH_\mathrm{S} - \mathbf{n} \cdot \boldsymbol{\omega} + \hat{L} z_t^*\right) \Psi_t^{(\mathbf{n})} + \Delta \cdot \hat{L} \sum_{k=1}^{K} n_k g_k \Psi_t^{(\mathbf{n}-\mathbf{e}_k)} - \Delta \cdot \hat{L}^\dagger \sum_{k=1}^{K} \Psi_t^{(\mathbf{n}+\mathbf{e}_k)}.$$

Performing the limit $\Delta \longrightarrow 0$, we obtain the linear HOPS equation for a system coupled to K bath modes:

$$\frac{\partial}{\partial t} \Psi_t^{(\mathbf{n})} = \left(-iH_\mathrm{S} - \mathbf{n} \cdot \boldsymbol{\omega} + \hat{L} z_t^*\right) \Psi_t^{(\mathbf{n})} + \hat{L} \sum_{k=1}^{K} n_k g_k \Psi_t^{(\mathbf{n}-\mathbf{e}_k)} - \hat{L}^\dagger \sum_{k=1}^{K} \Psi_t^{(\mathbf{n}+\mathbf{e}_k)}. \tag{2.9}$$

Note that the HOPS equations do not contain any functional derivatives and therefore can be readily integrated numerically.

2.2.2 Non-linear HOPS

A problem of the linear HOPS equation is that the states are not normalized. This leads to different realizations of the noise producing state vectors with vastly different magnitudes. The stochastic expectation value is then dominated by the states vectors with the largest magnitude, which means that one needs to compute a lot of states until the expectation value is converged. If a specific run happens to produce a state vector with low magnitude, this will not change the result much and can be seen as wasted computation time. To fix this problem, one can derive a non-linear version of HOPS, where the density matrix of the reduced systems is computed as an expectation value over normalized states

$$\widetilde{\Psi}_t(z) \coloneqq \frac{\Psi_t(z)}{\|\Psi_t(z)\|}$$

instead,

$$\rho(t) = \mathbb{E}\left[\left|\widetilde{\Psi}_t(z)\right\rangle \left\langle\widetilde{\Psi}_t(z)\right|\right]. \tag{2.10}$$

The non-linear HOPS equations are then obtained by replacing [5]

$$\hat{L}^\dagger \to \hat{L}^\dagger - \left\langle \hat{L}^\dagger \right\rangle_t$$

and

$$z_t^* \to \tilde{z}_t^* \coloneqq z_t^* + \int_0^t \alpha^*(t-s) \left\langle \hat{L}^\dagger \right\rangle_s \mathrm{d}s \tag{2.11}$$

in the linear HOPS equation. Here, $\langle\cdot\rangle_t$ denotes the expectation value at time t, calculated using the normalized state. The full non-linear HOPS then becomes

$$\frac{\partial}{\partial t}\Psi_t^{(\mathbf{n})} = \left(-i\hat{H}_\mathrm{S} - \mathbf{n}\cdot\boldsymbol{\omega} + \hat{L}\tilde{z}_t^*\right)\Psi_t^{(\mathbf{n})} + L\sum_{k=1}^{K} n_k g_k \Psi_t^{(\mathbf{n}-\mathbf{e}_k)} - \left(\hat{L}^\dagger - \langle\hat{L}^\dagger\rangle_t\right)\sum_{k=1}^{K}\Psi_t^{(\mathbf{n}+\mathbf{e}_k)}. \tag{2.12}$$

2.2.3 Computing Expectation Values

Both in linear and non-linear HOPS the density matrix of the system is computed by averaging over multiple realizations of pure states, see equations (2.2) and (2.10). However, mostly one is not interested in the density matrix directly, but in the expectation value of a given operator $\hat{A}$. In linear HOPS, the average is taken over unnormalized states, and thus the expectation value can be computed as

$$\left\langle \hat{A} \right\rangle_t = \frac{\mathrm{tr}\left\{\rho(t)\hat{A}\right\}}{\mathrm{tr}\left\{\rho(t)\right\}} \approx \frac{\mathrm{tr}\left\{\frac{1}{N}\sum_{i=1}^{N} |\Psi_t(z_i^*)\rangle\langle\Psi_t(z_i^*)|\,\hat{A}\right\}}{\mathrm{tr}\left\{\frac{1}{N}\sum_{i=1}^{N} |\Psi_t(z_i^*)\rangle\langle\Psi_t(z_i^*)|\right\}} = \frac{\sum_{i=1}^{N}\langle\Psi_t(z_i^*)|\,\hat{A}\,|\Psi_t(z_i^*)\rangle}{\sum_{i=1}^{N}\langle\Psi_t(z_i^*)|\Psi_t(z_i^*)\rangle}, \tag{2.13}$$

where we denote different realizations i of the stochastic process with $z_i^*, i = 0, \ldots, N$ and we used the cyclic property of the trace.
In non-linear HOPS the average is taken over normalized states $|\widetilde{\Psi}_t(z^*)\rangle$ instead, and thus it holds $\mathrm{tr}\{\rho\} = 1$. The expectation value can then be computed as

$$\left\langle \hat{A} \right\rangle_t = \frac{\mathrm{tr}\,\rho(t)\hat{A}}{\mathrm{tr}\,\rho(t)} \approx \frac{1}{N}\sum_{i=1}^{N}\langle\Psi_t(z_i^*)|\,\hat{A}\,|\Psi_t(z_i^*)\rangle \tag{2.14}$$

2.2.4 Truncation

When integrating the HOPS equations numerically, one has to truncate the hierarchy at some order, such that only a finite number of auxillary states $\Psi_t^{(\mathbf{n})}$ remain.
The most straight-forward truncation method is to set all auxillary states, for which one entry of the index vector exceeds a certain threshhold value N_trunc, to zero:

$$\Psi_t^{(\mathbf{n})} = 0 \quad \Leftrightarrow \quad \exists k\colon n_k \geq N_\mathrm{trunc}.$$

A more involved truncation method is triangular truncation, where all index vectors exceeding a given magnitude M_trunc are set to zero:

$$\Psi_t^{(\mathbf{n})} = 0 \quad \Leftrightarrow \quad \|\mathbf{n}\| \geq N_\mathrm{trunc}.$$

Instead of just setting truncated auxillary states to zero one can also use so-called *terminators* for a better approximation of the exact hierarchy. Terminators for the simple and the triangular truncation method are given in [1]. However, in practice, just setting the truncated states to zero yiels good results as well, which is also done throughout all computations in this thesis.

2.3 Generation of the Stochastic Process

There are multiple options to generate gaussian stochastic processes with the properties (2.3). In the following, I will give three examples, which can all also be found in my repository [2].
Firstly, one can generate the process using a complex multivariate normal distribution, which is e.g. implemented in the python library `numpy`. The disadvantage of this method is that computing the multivariate gaussian is slow for large stochastic processes, which are necessary if we want to compute HOPS with a small time step.
Second, one can use the method discussed in the appendix of [8] and in the supplementary material of [3], where the integral in the bath correlation function is approximated with a sum. This method generally works well, but introduces two additional parameters to tune (the cutoff and step size for approximating the integral with a sum). Lastly, one can use a *fourier filtering* technique [9]. The idea of this method is to first generate uncorrelated gaussian white noise, transform it to the frequency domain using a fourier transform, and multiplying with a filter. An inverse fourier transform is then used to go back to the time domain, creating noise with the requested correlations.
Throughout my thesis, the fourier filtering technique is used to generate stochastic processes. In the following, I will explain the method in more detail.
We start by generating complex gaussian white noise. This can for example be done with the **Box-Mueller-Wiener algorithm**: Given two random numbers $\xi_1, \xi_2 \in [0, 1]$ drawn from a uniform distribution, we can obtain the complex white noise θ as

$$\theta = \sqrt{-\log(\xi_1)} \cdot e^{2i\pi\xi_2}.$$

Our objective now is to generate a discrete stochastic process z of length N with correlations

$$\left\langle z_t z_s^* \right\rangle = \alpha(t-s) \equiv \alpha(\tau).$$

For this, we transform the white noise $\theta_t = \{\theta_1, \theta_2, \ldots, \theta_N\}$ into frequency space:

$$\hat{\theta}_k = \sum_{t=1}^{N} \theta_t e^{-\frac{2\pi i}{N} kt}.$$

Next, we construct the correlated noise in frequency space

$$\hat{z}_k \coloneqq \hat{\theta}_k \cdot \sqrt{\hat{\alpha}_k},$$

where we have introduced the fourier transformed bath correlation function

$$\hat{\alpha}_k = \sum_{n=0}^{N} \alpha(n \cdot \Delta t) e^{-\frac{2\pi i}{N} k\Delta t}$$

with the time step Δt. To obtain the correlated noise in the time domain, we simply perform an inverse fourier transform

$$z_t = \frac{1}{N} \sum_{k=1}^{N} \hat{z}_k.$$

One can show that the stochastic process z fulfills the conditions (2.3). It is important to note that, because of the symmetries of the Fourier transform, only $N/2$ of the generated values can be used; the other half are periodically correlated with the first half.

2.4 Matrix Product State (MPS) and Matrix Product Operators (MPO)

In this section, I give a brief introduction to the Matrix Product State formalism. Much more in-depth introductions of Matrix Product States and many of the popular algorithms can be found in [10, 11].

Matrix Product States (MPS)

Matrix product states (MPS), also known as tensor trains, are a useful way of writing quantum states. An arbitrary many-body state for a system consisting of N subsystem (for instance N spins on a chain) can be written as

$$|\Psi\rangle = \sum_{l_1,l_2,\ldots,l_N} \Psi_{l_1,l_2,\ldots,l_N} |l_1, l_2, \ldots, l_N\rangle ,$$

where $|l_1, l_2, \ldots, l_N\rangle \coloneqq |l_1\rangle \otimes |l_2\rangle \otimes \cdots \otimes |l_N\rangle$ are basis vectors of the many-body Hilbert space and $\Psi_{l_1,l_2,\ldots,l_N}$ are scalars. We may rewrite this state into

$$\begin{aligned} |\Psi\rangle &= \sum_{l_1,l_2,\ldots,l_N} \sum_{a_0=1}^{\chi_0} \sum_{a_1=1}^{\chi_1} \cdots \sum_{a_{N-1}=1}^{\chi_{N-1}} A^{[1],l_1}_{a_0,a_1} A^{[2],l_2}_{a_1,a_2} \cdots A^{[N],l_N}_{a_{N-1},a_0} |l_1, l_2, \ldots, l_N\rangle \\ &= \sum_{l_1,l_2,\ldots,l_N} \operatorname{tr}\left(\mathbf{A}^{[1]}\mathbf{A}^{[2]}\cdots\mathbf{A}^{[N]}\right) |l_1, l_2, \ldots, l_N\rangle , \end{aligned}$$

where the $\mathbf{A}^{[n]}$ are tensors of rank three. Each tensor has a physical leg l_n with the dimension of the local subsystem and two virtual legs with bond dimension χ_{n-1} and χ_n. The superscript $[n]$ denotes the subsystem that the tensor represents. When using open boundary conditions, the bond dimensions χ_0 and χN of the tensors $\mathbf{A}^{[1]}$ and $\mathbf{A}^{[N]}$ are set to one. The big advantage of using the MPS formulism is that one can easily approximate states by truncating the physical bond dimensions using truncated singular value decomposition. If we assume that all subsystems live in D-dimensional Hilbert spaces and denote the largest allowed virtual bond dimension with N_{trunc}, we need only $N \cdot D \cdot N_{\text{trunc}}^2$ coefficients to store the approximated state, compared to D^N for the full state. Increasing the bond dimensions N_{trunc} leads to a better approximation of the exact state.
Since often one is interested in systems with a large amount N of subsystems, the MPS formalism has proven to be a very valuable tool. Furthermore, it has lead to a variety of intuitive and useful algorithms, e.g. for computing ground states (DMRG) or time evolution (TEBD, TDVP).

Matrix Product Operators (MPO)

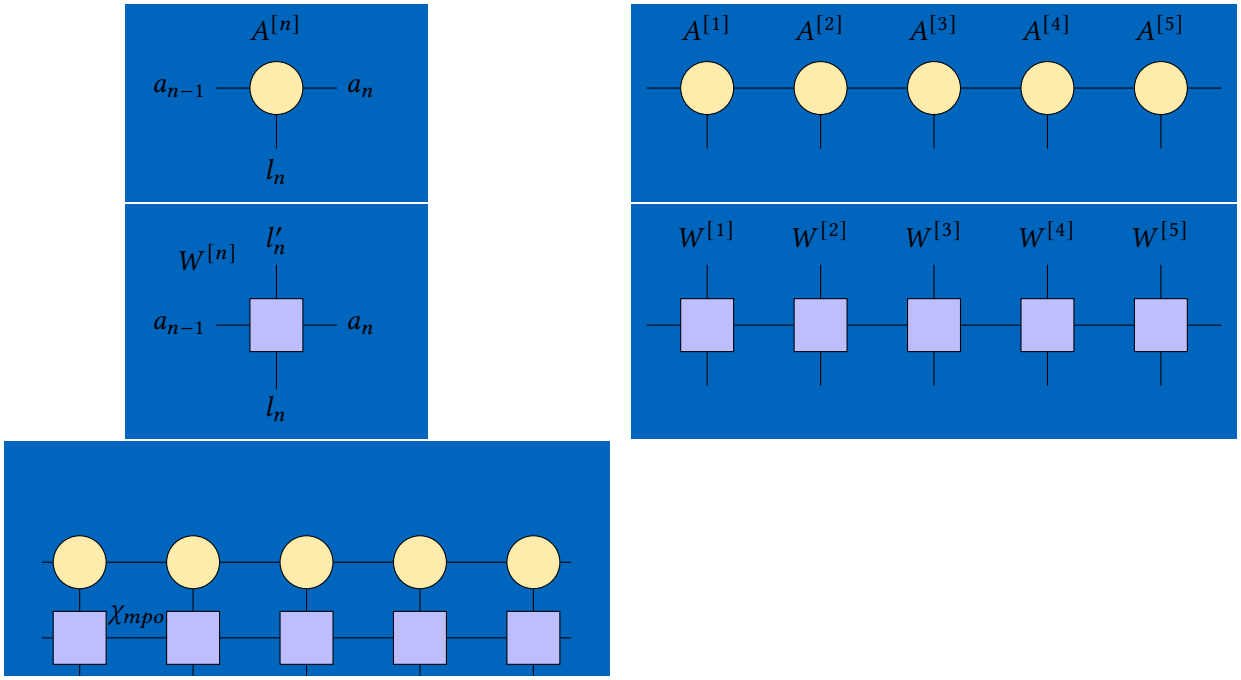
After defining Matrix Product States it is a natural next step to also write operators in the MPS formalism. A general operator $\hat{H}$ acting on the previously defined many-body system can be written as

$$\hat{H} = \sum_{l_1,l_2,\ldots,l_N} \sum_{l'_1,l'_2,\ldots,l'_N} H^{l'_1,l'_2,\ldots,l'_N}_{l_1,l_2,\ldots,l_N} |l_1, l_2, \ldots, l_N\rangle \langle l'_1, l'_2, \ldots, l'_N|$$

with the matrix elements $H^{l'_1,l'_2,\ldots,l'_N}_{l_1,l_2,\ldots,l_N}$. In the MPS formalism, the operator becomes

$$\begin{aligned} \hat{H} &= \sum_{l_1,l_2,\ldots,l_N} \sum_{l'_1,l'_2,\ldots,l'_N} \sum_{b_0=1}^{\chi_0} \sum_{b_1=1}^{\chi_1} \cdots \sum_{b_{N-1}=1}^{\chi_{N-1}} B^{[1],l_1,l'_1}_{b_0,b_1} B^{[2],l_2,l'_2}_{b_1,b_2} \cdots B^{[N],l_N,l'_N}_{b_{N-1},b_0} |l_1, l_2, \ldots, l_N\rangle \langle l'_1, l'_2, \ldots, l'_N| \\ &= \sum_{l_1,l_2,\ldots,l_N} \sum_{l'_1,l'_2,\ldots,l'_N} \operatorname{tr}\left(\mathbf{B}^{[1]}\mathbf{B}^{[2]}\cdots\mathbf{B}^{[N]}\right) |l_1, l_2, \ldots, l_N\rangle \langle l'_1, l'_2, \ldots, l'_N| , \end{aligned}$$

where the $\mathbf{B}^{[n]}$ are tensors of rank four. Each of the tensors of the MPO has two physical and two virtual legs.



2.5 Time Evolution of Matrix Product States

2.5.1 Runge-Kutta

2.5.2 The Time Dependent Variational Principle (TDVP)

2.6 The Spin Boson Model

The spin-boson model is a good model for testing and benchmarking methods for describing open quantum systems. It is widely used to show the applicability of new algorithms. The system Hamiltonian of the spin-boson model is

$$H_S = -\frac{1}{2}\Delta\hat{\sigma}_x + \frac{1}{2}\epsilon\hat{\sigma}_z, \tag{2.15}$$

where $\hat{\sigma}_x$ and $\hat{\sigma}_z$ are Pauli operators and Δ and ϵ are constants. The coupling to the bath is mediated by the operator $\hat{L} = \hat{\sigma}_z$, and the *Debye spectral density*

$$S(\omega) = \eta\frac{\omega\gamma}{\omega^2 + \gamma^2}$$

characterizes the bath correlation function (2.1). Before we can describe the spin-boson model using HOPS, we have to expand the bath correlation function as a sum of exponentials

$$\alpha(\tau) \approx \sum_{k=1}^{K}\alpha_k(\tau) \coloneqq \sum_{k=1}^{K} g_k e^{-\omega_k\tau}.$$

In the case of the Debye spectral density this can be done with a Matsubara expansion, resulting in the following expansion coefficients:

$$\begin{aligned} g_0 &= \frac{\eta\gamma}{2}\left(\cot\left(\frac{\gamma}{2T}\right) - i\right); \quad \omega_0 = \gamma, \\ g_k &= \eta\frac{4\pi T^2 k\gamma}{4\pi^2T^2k^2 - \gamma^2}; \quad \omega_k = 2\pi Tk \quad \text{for } k \geq 1. \end{aligned} \tag{2.16}$$

The derivation of these coefficients is given in appendix A.

3 Implementing and testing HOPS

3.1 Implementation

In this thesis we will only implement HOPS for a single bath mode, $K = 1$. The implementation for $K > 1$ follows analogeously, but with more involved book-keeping. If we only use a single bath mode, we can use a scalar index n instead of a vector index $\mathbf{n}$ to distinguish between the different auxillary states Ψ_t^n. We can store all auxillary states in a single vector

$$\mathbf{\Psi_t} \equiv \begin{pmatrix} \Psi_t^{(0)} \\ \Psi_t^{(1)} \\ \vdots \\ \Psi_t^{(K-1)} \end{pmatrix}.$$

Both the linear and the non-linear HOPS equations can be integrated by using regular numerical integration schemes, eg. Euler or Runge-Kutta. In this thesis, I choose a Runge-Kutta method of fourth order. Given the differential equation

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{\Psi} = \mathbf{f}(t, \mathbf{\Psi}) \tag{3.1}$$

and the state $\mathbf{\Psi_t}$ at time t, the state at time $t + Deltat$ can be computed as

$$\mathbf{\Psi_{t+\Delta t}} = \mathbf{\Psi_t} + \frac{1}{6}\left(\mathbf{k_1} + 2\mathbf{k_2} + 2\mathbf{k_3} + \mathbf{k_4}\right) \cdot \Delta t$$

where

$$\begin{aligned}
\mathbf{k_1} &= f\left(t, \mathbf{\Psi_n}\right), \\
\mathbf{k_2} &= f\left(t + \frac{\Delta t}{2}, \mathbf{\Psi_t} + \Delta t \frac{\mathbf{k_1}}{2}\right), \\
\mathbf{k_3} &= f\left(t + \frac{\Delta t}{2}, \mathbf{\Psi_t} + \Delta t \frac{\mathbf{k_2}}{2}\right), \\
\mathbf{k_4} &= f\left(t + h, \mathbf{\Psi_t} + \Delta t \mathbf{k_3}\right).
\end{aligned}$$

Linear HOPS

For implementing linear HOPS it is a good idea to split the right hand side of equation (3.1) into a "linear" and a "noise" part:

$$f_k\left(t, \mathbf{\Psi_t}\right) = \left(\boldsymbol{M}_{\text{linear}} + \tilde{z}_t^* \cdot \boldsymbol{M}_{\text{noise}}\right) \mathbf{\Psi_t}, \tag{3.2}$$

where we have defined the linear propagator

$$\mathbf{M}_{\text{linear}} = \begin{pmatrix}
-i\hat{H}_{\text{S}} & -\hat{L}^\dagger & 0 & \cdots & & 0 \\
\alpha(0)\hat{L} & -i\hat{H}_{\text{S}} - \omega\mathbb{1} & -\hat{L}^\dagger & 0 & \cdots & 0 \\
0 & 2\alpha(0)\hat{L} & -i\hat{H}_{\text{S}} - 2\omega\mathbb{1} & -\hat{L}^\dagger & 0 \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots \; \vdots & \vdots
\end{pmatrix}$$

and the noise propagator

$$M_{\text{noise}} = \begin{pmatrix} \hat{L} & 0 & \cdots & \\ 0 & \hat{L} & 0 & \cdots \\ \vdots & 0 & \hat{L} & \ddots \\ & \vdots & \ddots & \ddots \end{pmatrix}.$$

These propagators can be easily derived from equation (2.9). The advantage of defining the differential equation in such a way is that an update can then be computed by simple matrix addition and multiplication. It is worth noting that most of the entries of the propagator matrices are zero, giving way to an efficient implementation using sparse matrices.

Non-linear HOPS

The non-linear HOPS can be implemented by including a non-linear propagator in the right hand side of equation (3.1),

$$f_k\left(t, \mathbf{\Psi_t}\right) = \left(\boldsymbol{M}_{\text{linear}} + \tilde{z}_t^* \cdot \boldsymbol{M}_{\text{noise}} + \left\langle \hat{L}^\dagger \right\rangle \cdot \boldsymbol{M}_{\text{non-linear}}\right) \mathbf{\Psi_t}, \tag{3.3}$$

where

$$\mathbf{M}_{\text{non-linear}} = \begin{pmatrix} 0 & \mathbb{1} & \cdots & \\ 0 & 0 & \mathbb{1} & \cdots \\ \vdots & 0 & 0 & \ddots \\ & \vdots & \ddots & \ddots \end{pmatrix}.$$

The expectation value of $\hat{L}^\dagger$ can be easily computed with the physical state $\Psi_t^{(0)}$ at time t:

$$\left\langle \hat{L}^\dagger \right\rangle_t = \frac{\langle \Psi_t^{(0)} | \, \hat{L}^\dagger \, | \Psi_t^{(0)} \rangle}{\langle \Psi_t^{(0)} | \Psi_t^{(0)} \rangle}. \tag{3.4}$$

The last remaining problem is the computation of the memory term

$$z^*_{\text{memory}}(t) \coloneqq \int_0^t \alpha^*(t-s) \left\langle \hat{L}^\dagger \right\rangle$$

in the "shifted noise" (2.11) of the non-linear HOPS. To avoid recomputing the memory term at each step, we can derive an iterative update equation using some approximations:

$$\begin{aligned} z^*_{\text{memory}}(t+dt) &= \int_0^{t+dt} \mathrm{d}s\, \alpha^*(t+\Delta t - s) \left\langle \hat{L}^\dagger \right\rangle_s = e^{-\omega^* \Delta t} \int_0^{t+dt} \mathrm{d}s\, \alpha^*(t-s) \left\langle \hat{L}^\dagger \right\rangle_s \\ &= e^{-\omega^* \Delta t} z^*_{\text{memory}}(t) + e^{-\omega^* \Delta t} \int_t^{t+dt} \mathrm{d}s\, \alpha^*(t-s) \left\langle \hat{L}^\dagger \right\rangle_s \\ &\approx e^{-\omega^* \Delta t} z^*_{\text{memory}}(t) + e^{-\omega^* \Delta t} \Delta t \alpha^*(0) \left\langle \hat{L}^\dagger \right\rangle_t \\ &\approx z^*_{\text{memory}}(t) - \omega^* \Delta t \, z^*_{\text{memory}}(t) + \Delta t \, g^* \left\langle \hat{L}^\dagger \right\rangle_t \end{aligned} \tag{3.5}$$

With this, the memory term can be easily updated using Runge-Kutta or any other numerical integration scheme.

3.2 Testing HOPS with the Spin Boson Model

To test my implementation of the HOPS, I use the spin boson model (2.15) with $\Delta = 1$ and $\epsilon = 0$. For the bath correlation function I use the simple expansion

$$\alpha(\tau) = g e^{-\omega \tau}$$

with only a single bath mode and constants $g = 2$ and $\omega = 0.5 + 2i$. The constants for the spin-boson model and the bath correlation function are taken from [1]. In figure **??**, the dynamics of the model is shown, computed using 100, 1000, and 10000 realizations of the stochastic process. Both linear and non-linear HOPS are shown. One can directly see that the linear HOPS converges much slower. The reason for this is that the difference in magnitudes of the states from different realizations of the stochastic process can be really large (see figure **??**). Because of this, most realizations do not contribute much to the stochastic expectation value, which is therefore dominated by only a few realizations with large magnitudes. In the non-linear HOPS, states are normalized, and thus each realization is weighted equally in the stochastic expectation value, leading to faster convergence.

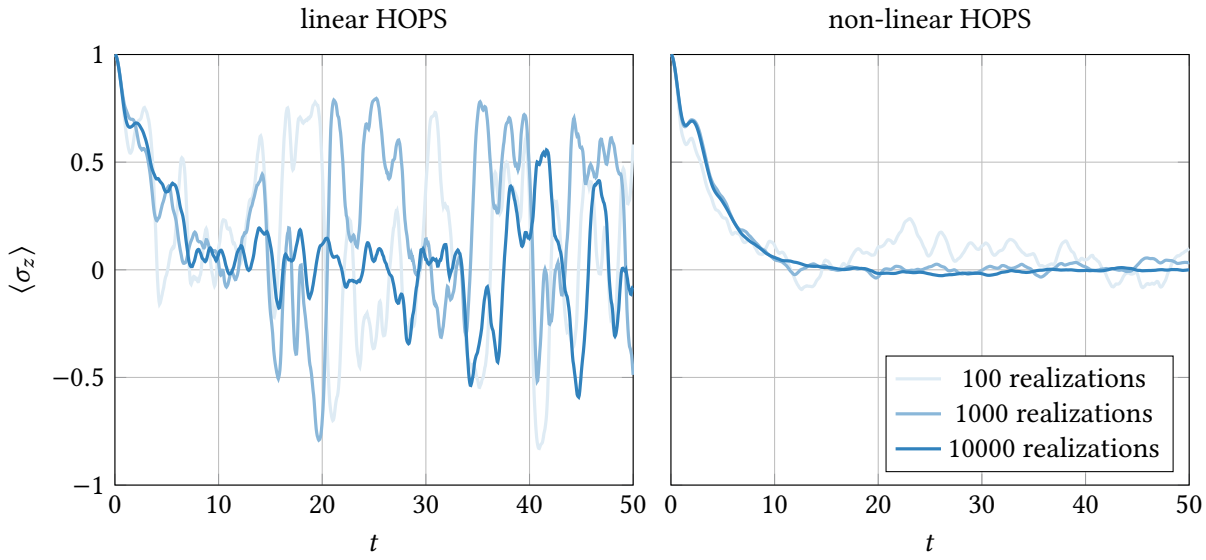


Figure 3.1 In this figure, the dynamics of the spin-boson model are computed using the HOPS. The stochastic expectation value of the σ_z operator is plotted against the time. The expectation value is taken over 100, 1000, and 10000 realizations of the stochastic processes. Both the linear (left) and the non-linear (right) HOPS was used. The time steps for all realizations was chosen as $\Delta t = 0.05$. The parameters for the spin-boson model are given in the text. The HOPS was truncated using simple truncation with $N_{\text{trunc}} = 8$.

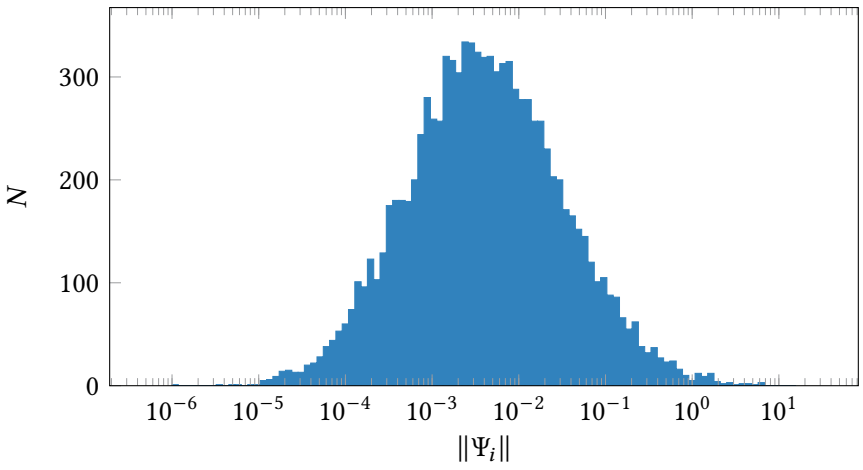


Figure 3.2 The magnitudes of states from 10000 linear HOPS realizations of the spin-boson model are shown in a histogram. There are big differences in the magnitudes of the different realizations, which leads to the problem that states with small magnitudes do not contribute much to the overall expectation value and can therefore be seen as wasted computation time.

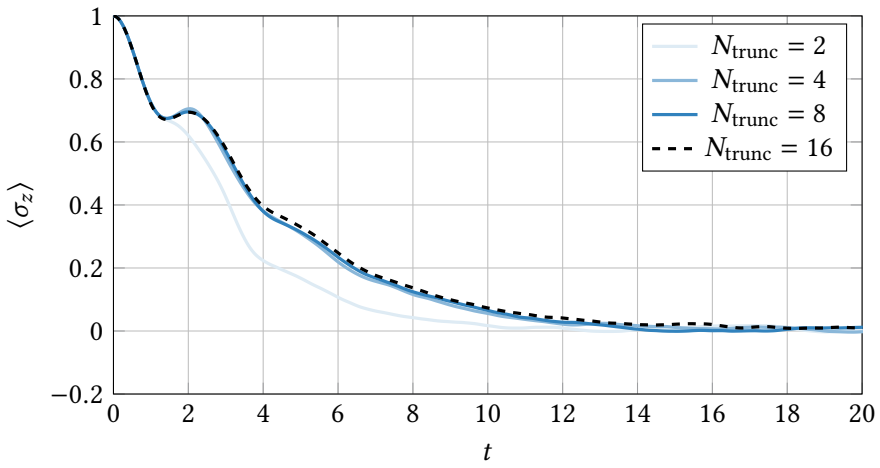


Figure 3.3 In this figure, the convergence of non-linear HOPS in the N_{trunc} parameter is shown. For different values of N_{trunc}, 1000 realizations of non-linear HOPS with the spin-boson model were computed each. The other parameters are the same as for figure **??**. $N_{\text{trunc}} = 8$, which was used for the full runs, is already well converged.

4 Implementing and testing HOMPS

4.1 Implementation

Matrix Product States

We now want to rewrite the linear and non-linear HOPS equations (2.9, 2.12) in such a way that we can use them together with the MPS formalism. We will do the derivation for the non-linear equation, following reference [3]. The linear equation can be derived analogeously.
We start by representing the full hierarchy at time t by a single quantum state

$$|\Psi_t\rangle = \sum_{l,\mathbf{n}} \Psi_t^{l,(\mathbf{n})} |l, n_1, n_2, \ldots, n_K\rangle . \tag{4.1}$$

The HOPS equations then become

$$\frac{\partial}{\partial t}\Psi_t^{l,(\mathbf{n})} = \left(-i\hat{H}_\mathrm{S} - \mathbf{n}\cdot\boldsymbol{\omega} + \hat{L}\tilde{z}_t^*\right)\Psi_t^{l,(\mathbf{n})} + L\sum_{k=1}^{K} n_k g_k \Psi_t^{l,(\mathbf{n}-\boldsymbol{e}_k)} - \left(\hat{L}^\dagger - \langle\hat{L}^\dagger\rangle_t\right)\sum_{k=1}^{K}\Psi_t^{l,(\mathbf{n}+\boldsymbol{e}_k)},$$

where $\mathbf{e}_k$ again denotes the kth unit vector. Next, we define the ladder operators

$$\begin{aligned}\hat{b}_k^\dagger |\mathbf{n}\rangle &:= |\mathbf{n}+\mathbf{e}_k\rangle \\ \hat{b}_k |\mathbf{n}\rangle &:= |\mathbf{n}-\mathbf{e}_k\rangle\end{aligned}$$

and the number operator

$$\hat{N}_k |\mathbf{n}\rangle := n_k |\mathbf{n}\rangle$$

and use them to write an update equation for the full state:

$$\frac{\partial}{\partial t}|\Psi_t\rangle = -i\hat{H}_\mathrm{eff}|\Psi_t\rangle ,$$

with the effective Hamiltonian

$$\begin{aligned}\hat{H}_\mathrm{eff} = \hat{H}_\mathrm{S}\otimes\mathbb{1} - i\sum_{k=1}^{K}\omega_k\cdot\mathbb{1}\otimes\hat{N}_k + i\tilde{z}_t^*\cdot\hat{L}\otimes\mathbb{1} \\ + i\sum_{k=1}^{K} g_k\cdot\hat{L}\otimes\hat{N}_k\hat{b}_k^\dagger - i\sum_{k=1}^{K}\left(\hat{L}^\dagger - \langle\hat{L}^\dagger\rangle_t\right)\otimes\hat{b}_k.\end{aligned} \tag{4.2}$$

Finally, we can now go to the MPS formalism. We can write the full state (4.1) as an MPS

$$|\Psi_t\rangle = \sum_{l,\mathbf{n},\mathbf{a}} A^{[1],l}_{a_0,a_1} A^{[2],n_1}_{a_1,a_2} A^{[3],n_2}_{a_2,a_3}\cdots A^{[K+1]n_K}_{a_K,a_0} |l, n_1, n_2, \ldots, n_K\rangle ,$$

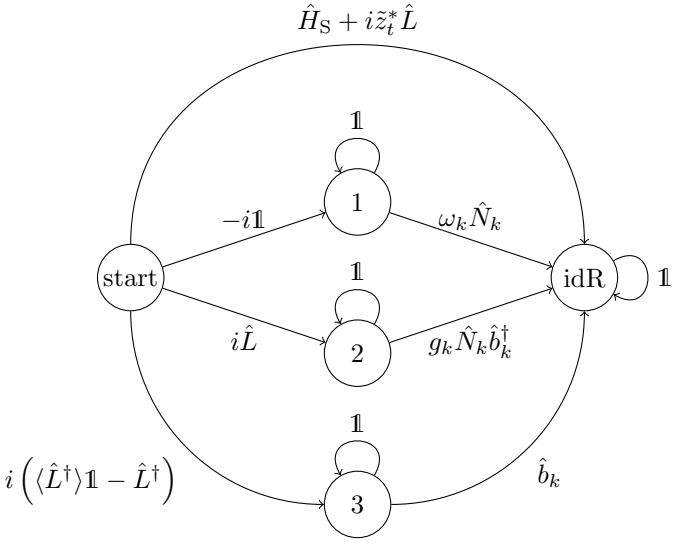


Figure 4.1 In this figure the state machine that can be used to generate the MPO (4.3) is sketched. The state machine can be constructed from equation (4.2). For reference on how to use state machines to construct MPOs, see [12]

using $K+1$ tensors in total. The last thing we still need to do is to write the effective Hamiltonian (4.2) in MPO form. This can for example be done by using the finite state machine method discussed in [12]. By using the finite state machine depicted in figure **??**, we arrive at the following tensors:

$$\begin{aligned} \boldsymbol{W}^{[1]} &:= \begin{pmatrix} -i\mathbb{1} & i\hat{L} & i\left(\langle \hat{L}^\dagger \rangle \mathbb{1} - \hat{L}^\dagger\right) & \hat{H}_\mathrm{S} + i\tilde{z}_t^* \hat{L} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \\ \boldsymbol{W}^{[k+1]} &:= \begin{pmatrix} \mathbb{1} & 0 & 0 & \omega_k \hat{N}_k \\ 0 & \mathbb{1} & 0 & g_k \hat{N}_k \hat{b}_k^\dagger \\ 0 & 0 & \mathbb{1} & \hat{b}_k \\ 0 & 0 & 0 & \mathbb{1} \end{pmatrix}, \quad k = 1, 2, \ldots, K. \end{aligned} \tag{4.3}$$

4.2 Benchmarking HOMPS

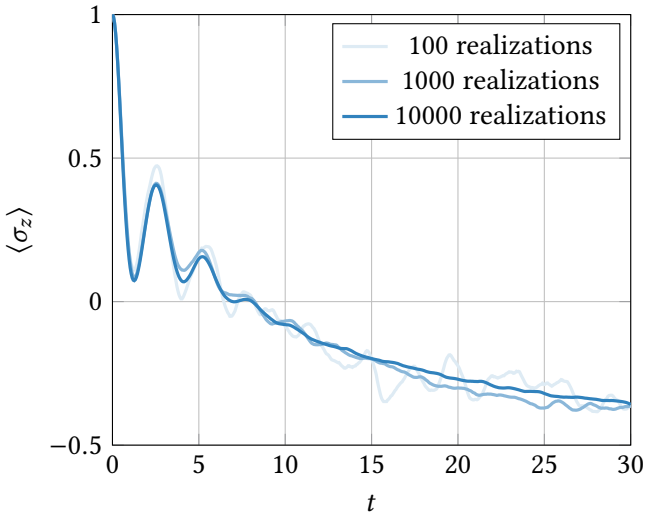


Figure 4.2 test

5 Conclusion

A Approximating the BCF of the Spin-Boson model

In this section we want to derive the result (2.16), that is coefficients g_j and w_j such that the bath correlation function

$$\alpha(\tau) = \frac{1}{\pi}\int_0^\infty S(\omega)\left[\coth\left(\frac{\omega}{2T}\right)\cos(\omega\tau) - i\sin(\omega\tau)\right]\mathrm{d}\omega = \int_0^\infty I(\omega,\tau)\mathrm{d}\omega$$

with the *Debye spectral density*

$$S(\omega) = \eta\frac{\omega\gamma}{\omega^2+\gamma^2}$$

can be approximated as a sum of exponentials

$$\alpha(\tau) \approx \sum_{j=0}^{N_{BCF}} g_j e^{-\omega_j\tau}.$$

To compute the expansion coefficients g_j and ω_j, we will use a *Matsubara expansion.* First we note that the integrand $I(\omega,\tau)$ is symmetric with respect to ω, $I(-\omega,\tau) = I(\omega,\tau)$. Hence, we can extend the integral over the negative real axis:

$$\alpha(\tau) = \frac{1}{2}\int_{-\infty}^\infty I(\omega,\tau)\mathrm{d}\omega.$$

We further split the integral into a real and an imaginary part $\alpha(\tau) = a(\tau) + ib(\tau)$, where

$$a(\tau) = \frac{1}{2\pi}\int_{-\infty}^\infty S(w)\coth\left(\frac{\omega}{2T}\right)\cos(\omega\tau)\mathrm{d}\omega$$

and

$$b(\tau) = -\frac{1}{2\pi}\int_{-\infty}^\infty S(w)\sin(\omega\tau)\mathrm{d}\omega.$$

Expanding the sine and cosine in terms of exponentials and using the antisymmetry of the hyperbolic cotangent and the Debye spectral density, we can write

$$a(\tau) = \frac{1}{2\pi}\int_{-\infty}^\infty S(\omega)\coth\left(\frac{\omega}{2T}\right)e^{i\omega\tau},$$

$$b(\tau) = -\frac{1}{2\pi i}\int_{-\infty}^\infty S(\omega)e^{i\omega\tau}.$$

To solve the integrals, we will use the residual theorem. Consider the line integral in figure **??**. Because of the exponential $e^{i\omega\tau}$, the contribution along L_2 vanishes as we take the limit $R\to\infty$. This leaves us with

$$\begin{aligned}
a(\tau) &= \frac{1}{2\pi}\oint_\xi S(\omega)\coth\left(\frac{\omega}{2T}\right)e^{i\omega\tau}\mathrm{d}\omega = i\sum_{\{\omega_j\}}\mathrm{Res}\left(S(\omega)\coth\left(\frac{\omega}{2T}\right);\omega_j\right)e^{i\omega_j\tau} \\
&= i\sum_{\{\omega_j'\}}\mathrm{Res}\left(S(\omega);\omega_j'\right)\coth\left(\frac{\omega_j'}{2T}\right)e^{i\omega_j'\tau} + i\sum_{\{\omega_j''\}}\mathrm{Res}\left(\coth\left(\frac{\omega}{2T}\right);\omega_j''\right)S(\omega_j'')e^{i\omega_j''\tau}, \\
b(\tau) &= -\frac{1}{2\pi}\oint_\xi S(\omega)e^{i\omega\tau}\mathrm{d}\omega = -\sum_{\{\omega_j'\}}\mathrm{Res}\left(S(\omega);\omega_j'\right)e^{i\omega_j'\tau},
\end{aligned}$$

where ω_j are the poles of $S(\omega)\coth\left(\frac{\omega}{2T}\right)$, ω_j' the poles of $S(\omega)$, ω_j'' the poles of $\coth\left(\frac{\omega}{2T}\right)$, and we used the residue theorem. We also assumed $\omega_j' \neq \omega_b''$ for all a, b, which is the case for almost all γ and T. Next, we

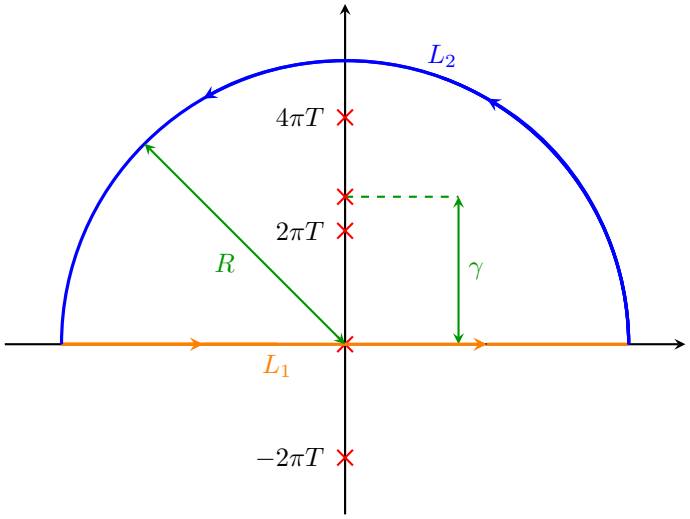


Figure A.1 In this figure, the line integral for computing the expansion of the bath correlation function is shown.

need to compute the poles and residues. The Debye spectral density has simple poles at $\omega_j' = \pm i\gamma$, with residuals

$$\text{Res}\left(S(\omega); \pm i\gamma\right) = \lim_{\omega \to i\gamma} (\omega \mp i\gamma) S(\omega) = \frac{\eta\gamma}{2}.$$

The poles of the hyperbolic cotangent $\coth\left(\frac{\omega}{2T}\right)$ lie on the imaginary axis, $\omega_j'' = 2\pi i T a$ with $a \in \mathbb{Z}$, and are again simple poles with residuals

$$\text{Res}\left(\coth\left(\frac{\omega}{2T}\right); 2\pi i T j\right) = \lim_{\omega \to 2?piiTj} (\omega - 2\pi i T j) \coth\left(\frac{\omega}{2T}\right) = \lim_{\lambda \to 0} 2T\lambda \coth(\lambda) = 2T,$$

where we used the identities $\coth(x + i\pi n) = \coth(x)$ for $n \in \mathbb{Z}$ and $\lim_{x \to 0} x \coth(x) = 1$.
With this, we are now equipped to solve the integrals:

$$b(\tau) = -\frac{\eta\gamma}{2} e^{-\gamma\tau},$$

$$a(\tau) = i\frac{\eta\gamma}{2} \coth\left(\frac{i\gamma}{2T}\right) e^{-\gamma T} + i \sum_{j=1}^{\infty} 2T\, S(2\pi i T j)\, e^{-2\pi T j \tau}.$$

Putting the results together, we can write

$$\alpha(\tau) = a(\tau) + ib(\tau) = \sum_{j=0}^{\infty} g_j e^{-\omega_j \tau}$$

with the coefficients (2.16). The result is similar to the one obtained in [13] (up to a different normalization).

Bibliography

[1] D. Suess, A. Eisfeld, and W. T. Strunz. "Hierarchy of Stochastic Pure States for Open Quantum System Dynamics". In: *Phys. Rev. Lett.* 113 (15 2014), p. 150403.

[2] B. Sappler. *HOPS and HOMPS implementation.* 2023. URL: https://github.com/SnackerBit/HOPS.

[3] X. Gao, J. Ren, A. Eisfeld, and Z. Shuai. "Non-Markovian stochastic Schrödinger equation: Matrix-product-state approach to the hierarchy of pure states". In: *Phys. Rev. A* 105 (3 2022), p. L030202.

[4] L. Diósi and W. T. Strunz. "The non-Markovian stochastic Schrödinger equation for open systems". In: *Physics Letters A* 235.6 (1997), pp. 569–573. ISSN: 0375-9601.

[5] L. Diósi, N. Gisin, and W. T. Strunz. "Non-Markovian quantum state diffusion". In: *Phys. Rev. A* 58 (3 1998), pp. 1699–1712.

[6] R. Hartmann and W. T. Strunz. "Open Quantum System Response from the Hierarchy of Pure States". In: *The Journal of Physical Chemistry A* 125.32 (2021). PMID: 34353022, pp. 7066–7079. eprint: https://doi.org/10.1021/acs.jpca.1c03339.

[7] R. Hartmann and W. T. Strunz. "Exact Open Quantum System Dynamics Using the Hierarchy of Pure States (HOPS)". In: *Journal of Chemical Theory and Computation* 13.12 (2017). PMID: 29016126, pp. 5834–5845. eprint: https://doi.org/10.1021/acs.jctc.7b00751.

[8] K. Song, L. Song, and Q. Shi. "An alternative realization of the exact non-Markovian stochastic Schrödinger equation". In: *The Journal of Chemical Physics* 144.22 (June 2016). 224105. ISSN: 0021-9606. eprint: https://pubs.aip.org/aip/jcp/article-pdf/doi/10.1063/1.4953244/15513202/224105_1_online.pdf.

[9] J. Garcia-Ojalvo and J. Sancho. "Noise in Spatially Extended Systems". In: *Physical review. E, Statistical physics, plasmas, fluids, and related interdisciplinary topics* 49 (May 1994), pp. 2769–2778.

[10] U. Schollwöck. "The density-matrix renormalization group in the age of matrix product states". In: *Annals of Physics* 326.1 (2011), pp. 96–192.

[11] J. Hauschild and F. Pollmann. "Efficient numerical simulations with Tensor Networks: Tensor Network Python (TeNPy)". In: *SciPost Physics Lecture Notes* (2018).

[12] J. Motruk, M. P. Zaletel, R. S. K. Mong, and F. Pollmann. "Density matrix renormalization group on a cylinder in mixed real and momentum space". In: *Phys. Rev. B* 93 (15 2016), p. 155139.

[13] Q. Shi et al. "Electron transfer dynamics: Zusman equation versus exact theory". In: *The Journal of Chemical Physics* 130.16 (Apr. 2009). 164518. ISSN: 0021-9606. eprint: https://pubs.aip.org/aip/jcp/article-pdf/doi/10.1063/1.3125003/15646257/164518_1_online.pdf.