Technical University of Munich

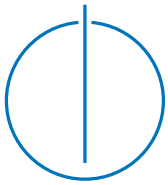
School of Computation, Information and Technology

Department of Computer Science

Bachelor's Thesis in Informatics

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

Benjamin Sappler



Technical University of Munich

School of Computation, Information and Technology

Department of Computer Science

Bachelor's Thesis in Informatics

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

Benchmarking eines Tensor Network Algorithmus für die HOPS-Methode zur Simulation offener, nicht-Markovianischer Quantensysteme

Author:	Benjamin Sappler
Supervisor:	Prof. Dr. Christian Mendl
Advisor:	Richard Milbradt, M.Sc.
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

Abstract

The dynamics of open quantum systems is of great interest for many fields in physics and chemistry. For systems strongly interacting with their environments the dynamics are inherently non-Markovian, which is notoriously difficult to study and simulate. In this thesis I implement in detail the Hierarchy of Pure States (HOPS) [1] and the recently developed Hierarchy of Matrix Product States (HOMPS) [2] methods, which can be used to simulate non-Markovian dynamics. I then benchmark both methods using the spin-boson model.

Zusammenfassung

Die Dynamik offener Quantensysteme ist in vielen Bereichen der Physik und Chemie von großer Bedeutung. Systeme, die stark mit ihrer Umgebung interagieren, besitzen eine inherent nicht-Markovianische Dynamik, was erhebliche Probleme bei der theoretischen Beschreibung und Simulation dieser Systeme verursachen kann. In dieser Arbeit implementiere ich die Hierarchy of Pure States (HOPS) [1] und die Hierarchy of Matrix Product States (HOMPS) [2] Methoden, die für die Simulation offener, nicht-Markovianischer Systeme benutzt werden können. Anschließend benchmarke ich beide Methoden mit dem Spin-Boson Modell.

Contents

1 Introduction

Open quantum systems, i.e., quantum systems that are interacting with an environment, are important for modelling many complex processes, including the decoherence of quantum computers and many processes in the field of chemical physics. Most often, these systems cannot be described analytically, but only numerically. Thus, the field of open quantum system simulation is very important.
In many cases, the dynamics of an open quantum system can be described with a Markov approximation, where the state of the environment is assumed to only depend on the current state of the system. However, in many interesting situations (for example involving short length scales and high energies), the Markov approximation fails. The environment develops a memory and depends on previous states of the system as well.
A popular approach for the simulation of open quantum systems is the Hierarchy Of Pure States (HOPS) [1]. With the HOPS, one can integrate the non-Markovian quantum state diffusion equation and simulate non-Markovian open quantum systems. Recently it was shown [2] 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 the HOPS method and the HOMPS method in detail. I start by summarizing the theory which is necessary for understanding both methods in Section 2. This includes the derivation of the HOPS from the non-Markovian quantum state diffusion equation in Section 2.2 and a brief introduction of Matrix Product States and Matrix Product Operators in Section 2.4. Next, I explain in detail how the HOPS method can be implemented and test my implementation on the spin-boson model in Section 3. In Section 4, I then derive and implement the HOMPS method, 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 [3].

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 the two. 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 [2] 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) [4, 5, 6, 7, 8] 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 $\mathcal{H}_{\mathrm{S}}$ 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 [9], but the same result is also obtained in [1, 10].
We will start by approximating the bath correlation function (BCF) (2.1) by a finite sum of exponentials

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

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 two different expansion methods.
We will work with the discrete version of the NMQSD equation [9]

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

where z_t with $t \in \mathbb{N}_0$ now is a discrete stochastic process and we introduced the time step $\Delta \in \mathbb{R}_{>0}$. One can easily see that the NMQSD equation (2.4) is recovered if the limit $\Delta \to 0$ 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, simplifying the following derivation.
We define the operator

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

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):

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

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

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

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

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

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

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

where we have omitted terms of order $O\left(\Delta^2\right)$. Taking the limit $\Delta \longrightarrow 0$, we obtain the linear HOPS equations for a system coupled to K bath modes:

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

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 is that the states are not normalized. For that reason, different realizations of the noise can produce state vectors with vastly different magnitudes. The stochastic expectation value is then dominated by the state vectors with the largest magnitudes, which causes the expectation value to converge very slowly. If a specific realization 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:

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

The non-linear HOPS equations are obtained by replacing [4]

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

and

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

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

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

2.2.3 Computing Expectation Values

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

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

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

$$\mathbb{E}\left[\langle\hat{A}\rangle_t\right] = \frac{\mathrm{tr}\left\{\rho(t)\hat{A}\right\}}{\mathrm{tr}\left\{\rho(t)\right\}} \approx \frac{1}{N}\sum_{i=1}^{N}\langle\widetilde{\Psi}_t(z_i^*)|\hat{A}|\widetilde{\Psi}_t(z_i^*)\rangle\,.$$

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:

$$\Psi_t^{(\mathbf{n})} = 0 \quad \Leftrightarrow \quad \exists k\colon n_k \geq N_{\mathrm{trunc}}. \tag{2.14}$$

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

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

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, just setting the truncated states to zero yields good results in practice. This is done throughout all computations in this thesis.

2.3 Generation of the Stochastic Process

There are multiple options to generate a gaussian stochastic process with the Properties (2.3). In the following, I will give three examples, implementations of which can be found at [3].
First, one can generate the process using a complex multivariate normal distribution, which is available in multiple numerical libraries, e.g., the `numpy` package for python. The disadvantage of this method is that computing the multivariate gaussian is slow for large stochastic processes, which are necessary for the HOPS method with small time steps.
Second, one can use the method discussed in the appendix of [11] and in the supplementary material of [2], where the integral in the bath correlation function is approximated with a sum. The stochastic process can then be generated as

$$z(t) = \sum_{j=1}^{N} \sqrt{\frac{S(\omega_j \Delta\omega)}{\pi}} \left(\sqrt{\frac{1}{2}\coth\left(\frac{\omega_j \beta}{2}\right) - \frac{1}{2}} e^{i(\omega_j t + 2\pi\Phi_{j,1})} + \sqrt{\frac{1}{2}\coth\left(\frac{\omega_j \beta}{2}\right) + \frac{1}{2}} e^{i(-\omega_j t + 2\pi\Phi_{j,2})} \right),$$

where the $\Phi_{j,1}$ and $\Phi_{j,2}$ are independent random numbers uniformly distributed in $[0, 1)$, and $\omega_j = \left(j - \frac{1}{2}\right)\Delta\omega$. This method generally works well, but introduces two additional parameters to tune (the cutoff N and step size $\Delta\omega$ for approximating the integral with the sum).
Third, one can use the *Fourier filtering* technique [12]. The idea of this method is to first generate uncorrelated gaussian white noise, then to transform it to the frequency domain using a Fourier transform, and to apply a multiplicative filter. An inverse Fourier transform is then used to go back to the time domain, creating noise with the requested correlations.
Throughout my thesis, I use he Fourier filtering technique 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** [13]: 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:

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

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 to Matrix Product States and many of the popular algorithms can be found in [14, 15].

Matrix Product States (MPS)

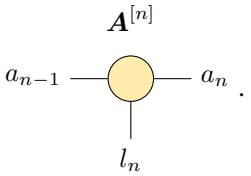
Matrix product states (MPS), also known as tensor trains, are a useful way of writing quantum states. An arbitrary many-body state for a system consisting of N subsystem (for instance N spins on a chain) can be written as

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

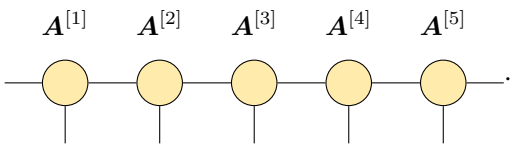
where $|l_1, l_2, \ldots, l_N\rangle \coloneqq |l_1\rangle \otimes |l_2\rangle \otimes \cdots \otimes |l_N\rangle$ are basis vectors of the many-body Hilbert space and $\Psi_{l_1,l_2,\ldots,l_N}$ are scalars. We may rewrite this state into

$$\begin{aligned} |\Psi\rangle &= \sum_{l_1,l_2,\ldots,l_N} \sum_{i_0=1}^{\chi_0} \sum_{i_1=1}^{\chi_1} \cdots \sum_{i_{N-1}=1}^{\chi_{N-1}} A^{[1],l_1}_{i_0,i_1} A^{[2],l_2}_{i_1,i_2} \cdots A^{[N],l_N}_{i_{N-1},i_0} |l_1, l_2, \ldots, l_N\rangle \\ &= \sum_{l_1,l_2,\ldots,l_N} \operatorname{tr}\left(\mathbf{A}^{[1],l_1}\mathbf{A}^{[2],l_2}\cdots\mathbf{A}^{[N],l_N}\right) |l_1, l_2, \ldots, l_N\rangle , \end{aligned} \tag{2.15}$$

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 χ_N of the tensors $\mathbf{A}^{[1]}$ and $\mathbf{A}^{[N]}$ are set to one.
The main advantage of using the MPS formulism is that one can easily approximate states by truncating the virtual 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 led to a variety of intuitive and useful algorithms, e.g., for computing ground states (DMRG [14, 16]) or time evolution (TEBD [17], TDVP [18]). There also exists an intuitive diagrammatic notation for tensor networks, where tensors are represented by shapes and their indices are represented by lines emerging from these shapes. In this notation, a single tensor $\boldsymbol{A}^{[n]}$ of rank three can be written as



Connecting lines of two tensors represents a contraction of these tensors along the specified legs. A full MPS can be drawn by connecting the virtual legs of neighbouring tensors:



Matrix Product Operators (MPO)

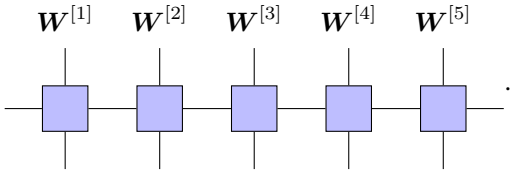
After defining Matrix Product States it is a natural next step to also write operators in the MPS formalism. A general operator $\hat{H}$ acting on the previously defined many-body system can be written as

$$\hat{H} = \sum_{l_1,l_2,\ldots,l_N} \sum_{l_1',l_2',\ldots,l_N'} H_{l_1,l_2,\ldots,l_N}^{l_1',l_2',\ldots,l_N'} \, |l_1, l_2, \ldots, l_N\rangle \, \langle l_1', l_2', \ldots, l_N'|$$

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

$$\begin{aligned} \hat{H} &= \sum_{l_1,l_2,\ldots,l_N} \sum_{l_1',l_2',\ldots,l_N'} \sum_{j_0=1}^{\chi_0} \sum_{j_1=1}^{\chi_1} \cdots \sum_{j_{N-1}=1}^{\chi_{N-1}} W_{j_0,j_1}^{[1],l_1,l_1'} W_{j_1,j_2}^{[2],l_2,l_2'} \cdots W_{j_{N-1},j_0}^{[N],l_N,l_N'} \, |l_1, l_2, \ldots, l_N\rangle \, \langle l_1', l_2', \ldots, l_N'| \\ &= \sum_{l_1,l_2,\ldots,l_N} \sum_{l_1',l_2',\ldots,l_N'} \mathrm{tr}\left(\mathbf{W}^{[1],l_1,l_1'} \mathbf{W}^{[2],l_2,l_2'} \cdots \mathbf{W}^{[N],l_N,l_N'} \right) \, |l_1, l_2, \ldots, l_N\rangle \, \langle l_1', l_2', \ldots, l_N'| \,, \end{aligned} \tag{2.16}$$

where the $\mathbf{W}^{[n]}$ are tensors of rank four. Each of the tensors of the MPO has two physical and two virtual legs. In the diagrammatic notation, an MPO can be written as



2.5 Time Evolution of Matrix Product States

One of the main problems concerning quantum systems is time evolution. The goal is to find the state at a time t, under the condition that the state at time t_0 is known and the time-dependent Schrödinger equation

$$\frac{d}{dt}\left|\Psi(t)\right\rangle = -i\hat{H}(t)\left|\Psi(t)\right\rangle$$

holds, where we have set $\hbar = 1$. There exist many algorithms implementing time evolution directly on MPS [19]. Some of these algorithms, for example the very popular Time Evolving Block Decimation (TEBD [17]), are not applicable to long-range interactions. Here, we will focus on two methods in particular, the fourth order Runge Kutta method (RK4) and the Time Dependent Variational Principle (TDVP), which work with a Hamiltonian given as an MPO and can handle long-range interactions. Implementations of both methods can be found at [3].

2.5.1 Runge-Kutta

The fourth order Runge-Kutta or RK4 method is a well known numerical method for solving linear equations. With this method, the state at time $t + \Delta t$ can be computed from the state at time t as

$$\left|\Psi(t+\Delta t)\right\rangle \approx \left|\Psi(t)\right\rangle + \frac{\Delta t}{6}\left(\left|k_1\right\rangle + 2\left|k_2\right\rangle + 2\left|k_3\right\rangle + \left|k_4\right\rangle\right),$$

where

$$\begin{aligned}
\left|k_1\right\rangle &= -i\hat{H}\left(t\right)\left|\Psi(t)\right\rangle, \\
\left|k_2\right\rangle &= -i\hat{H}\left(t+\Delta t/2\right)\left(\left|\Psi(t)\right\rangle + \frac{\Delta t}{2}\left|k_1\right\rangle\right), \\
\left|k_3\right\rangle &= -i\hat{H}\left(t+\Delta t/2\right)\left(\left|\Psi(t)\right\rangle + \frac{\Delta t}{2}\left|k_2\right\rangle\right), \\
\left|k_4\right\rangle &= -i\hat{H}\left(t+\Delta t\right)\left(\left|\Psi(t)\right\rangle + \Delta t\left|k_3\right\rangle\right).
\end{aligned}$$

For this method to work with MPSs and MPOs we have to implement two operations: Addition of two MPSs and multiplication of an MPS with an MPO. The addition of two MPSs

$$\left|\Psi\right\rangle = \sum_{l_1,l_2,\ldots,l_N} \mathrm{tr}\left(\mathbf{A}^{[1],l_1}\mathbf{A}^{[2],l_2}\cdots\mathbf{A}^{[N],l_N}\right)\left|l_1,l_2,\ldots,l_N\right\rangle$$

and

$$\left|\Phi\right\rangle = \sum_{l_1,l_2,\ldots,l_N} \mathrm{tr}\left(\mathbf{B}^{[1],l_1}\mathbf{B}^{[2],l_2}\cdots\mathbf{B}^{[N],l_N}\right)\left|l_1,l_2,\ldots,l_N\right\rangle$$

can be constructed by merging the tensors for each sub system block diagonally [20]:

$$\begin{aligned}
\left|\Psi\right\rangle + \left|\Phi\right\rangle &= \sum_{l_1,l_2,\ldots,l_N} \mathrm{tr}\left(\mathbf{C}^{[1],l_1}\mathbf{C}^{[2],l_2}\cdots\mathbf{C}^{[N],l_N}\right)\left|l_1,l_2,\ldots,l_N\right\rangle, \\
\mathbf{C}^{[n],l_n} &= \begin{pmatrix} \mathbf{A}^{[n],l_n} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}^{[n],l_n} \end{pmatrix}.
\end{aligned}$$

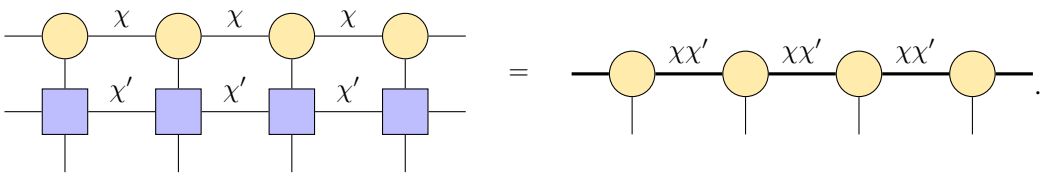
If the bond dimensions of the original two MPSs are χ_1 and χ_2, the resulting MPS has bond dimension $\chi_1 + \chi_2$.

Applying an MPO to an MPS can be done by contracting the two networks together along their physical legs. The result of applying the MPO (2.16) to the MPS (2.15) is

$$\hat{H}\left|\Psi\right\rangle = \sum_{l_1,l_2,\ldots,l_N} \mathrm{tr}\left(\mathbf{D}^{[1],l_1}\mathbf{D}^{[2],l_2}\cdots\mathbf{D}^{[N],l_N}\right)\left|l_1,l_2,\ldots,l_N\right\rangle,$$

$$D^{[n],l_n}_{k_{n-1},k_n} := D^{[n],l_n}_{(i_{n-1}j_{n-1}),(i_n j_n)} = \mathrm{reshape}\left(\sum_{l'_n} W^{[n],l_n,l'_n}_{j_{n-1},j_n} A^{[n],l'_n}_{i_{n-1},j_n}\right),$$

where the tensors $\mathbf{C}^{[n]}$ are computed by first contracting the $A^{[n]}$ and $W^{[n]}$ tensors along their physical legs and then reshaping the result, grouping together the virtual legs i_{n-1} with j_{n-1} and i_n with j_n. The operation can also be written diagrammatically:



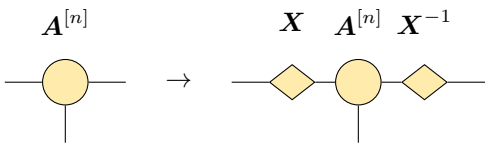
Applying an MPO to an MPS also increases the bond dimension: If the bond dimension of the original MPS is χ and the bond dimension of the MPO is χ', the bond dimension of the result will be $\chi \cdot \chi'$.
We have seen that both the addition of two MPSs and the multiplication of an MPS with an MPO increase the bond dimension. This is a problem since the computational cost of both operations scales with a power of the bond dimension. Therefore we need to truncate the MPS regularly. This can for example be done by sweeping accross the MPS, performing singular value decompositions, and keeping only the N_{trunc} largest singular values [20].

2.5.2 Time Dependent Variational Principle (TDVP)

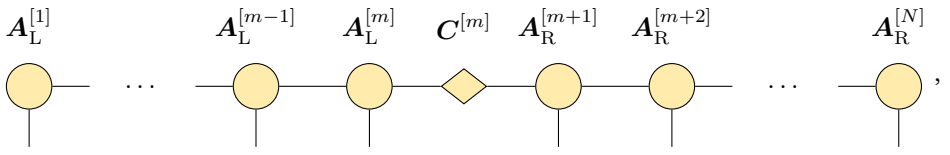
The idea of the Time Dependent Variational Principle [18, 21, 22] is to project the right hand side of the Schrödinger equation to the *tangent space* of the MPS:

$$\frac{d}{dt}\left|\Psi(t)\right\rangle_{\text{MPS}} = -i\hat{P}_{T|\Psi(t)\rangle_{\text{MPS}}}\hat{H}(t)\left|\Psi(t)\right\rangle_{\text{MPS}}. \tag{2.17}$$

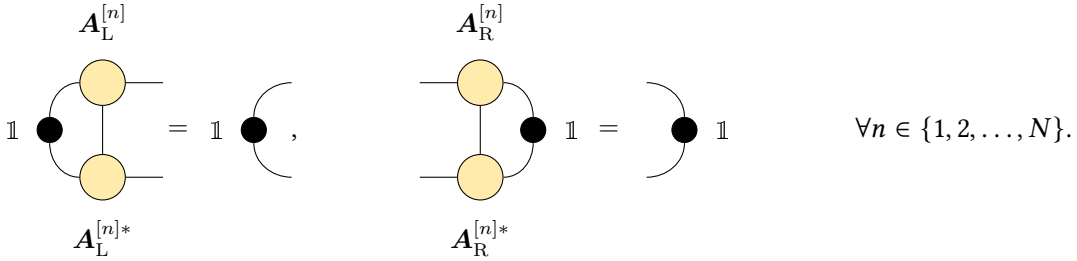
The tangent space is a manifold of MPSs that are orthogonal to the current MPS state $\left|\Psi(t)\right\rangle_{\text{MPS}}$. When integrating this equation, the resulting state never leaves the MPS manifold.
There are two versions of TDVP, one that updates MPS tensors one after the other (TDVP1), and one that updates two neighbouring MPS tensors at once (TDVP2). We will focus on TDVP2, since it allows to grow the bond dimension as needed, whereas TDVP1 assumes a fixed bond dimension.
To discuss the algorithm, we must first introduce the canonical form of MPSs. When constructing an MPS, we have an inherent gauge degree of freedom, because we can replace every tensor in the MPS



and recover the original MPS. This gauge degree of freedom can be used to transform any MPS into the *mixed canonical form*



where it holds



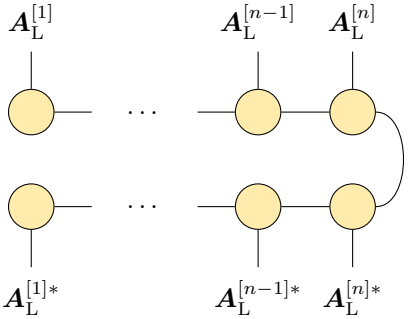
$$\forall n \in \{1, 2, \ldots, N\}.$$

$A_L^{[n]}$ and $A_R^{[n]}$ are then called *left-/right-canonical* and $C^{[m]}$ is the *orthogonality center*. Algorithms to turn any MPS into mixed canonical form and to shift the orthogonality center around can be found in [14, 15]. Assume now that the current MPS is given in mixed-canonical form. It is possible to show that the tangent space projection operator then has the form [21, 22]

$$\hat{P}_{T|\Psi(t)\rangle_{\mathrm{MPS}}} = \sum_{n=1}^{N} P_{\mathrm{L}}^{[1:n-1]} \otimes \mathbb{1}^{[n]} \otimes P_{\mathrm{R}}^{[n+1:N]} - \sum_{n=1}^{N-1} P_{\mathrm{L}}^{[1:n]} \otimes P_{\mathrm{R}}^{[n+1:N]},$$

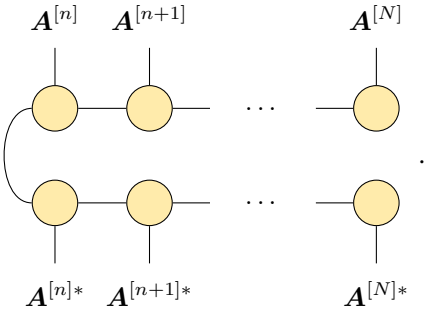
where

$P_{\mathrm{L}}^{[1:n]} =$



and

$P_{\mathrm{R}}^{[n:N]} =$

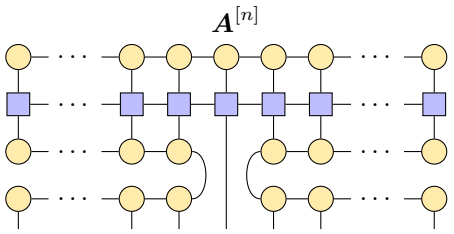


.

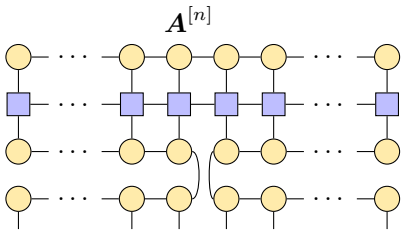
We can now write the right hand side of the time evolution Equation (2.17), up to the factor $-i$, as

$\hat{P}_{T|\Psi(t)\rangle_{\mathrm{MPS}}} \hat{H}(t) \, |\Psi(t)\rangle_{\mathrm{MPS}}$

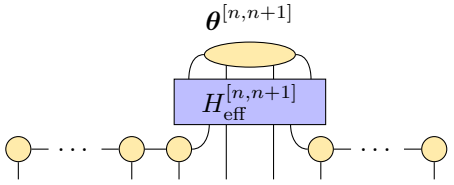
$= \sum_{n=1}^{N}$



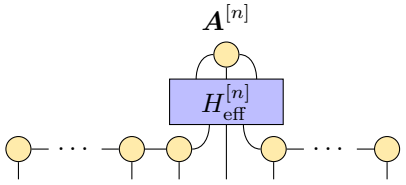
$- \sum_{n=1}^{N-1}$



$= \sum_{n=1}^{N}$



$- \sum_{n=1}^{N-1}$



,

where we introduced the two-site tensor $\theta^{[n,n+1]}$, which can be constructed by contracting the two neighbouring tensors $A^{[n]}$ and $A^{[n+1]}$. Additionally, we contracted the MPO with parts of the MPS into the effective two-site and one-site Hamiltonians $H_{\mathrm{eff}}^{[n,n+1]}$ and $H_{\mathrm{eff}}^{[n]}$.
In the following, the TDVP2 algorithm is briefly summarized. A more in-depth description can be found at [18, 21, 22]. The algorithm sweeps through the MPS, updating pairs of neighbouring tensors along the

way. A single update on the two-site tensor $\boldsymbol{\theta}^{[n,n+1]}$ during the process of sweeping from left to right is made as follows: First, we contract $\boldsymbol{\theta}^{[n,n+1]}$ with

$$\exp\left(-\frac{i\Delta t}{2} H_{\text{eff}}^{[n,n+1]}\right). \tag{2.18}$$

We then split and truncate the resulting tensor using SVD to obtain the updated tensors $\boldsymbol{A}^{[n]\prime}$ and $\boldsymbol{A}^{[n+1]\prime}$. Next, we apply

$$\exp\left(\frac{i\Delta t}{2} H_{\text{eff}}^{[n]}\right) \tag{2.19}$$

to $\boldsymbol{A}^{[n+1]\prime}$, forming $\boldsymbol{A}^{[n+1]\prime\prime}$. Finally, we can contract $\boldsymbol{A}^{[n+1]\prime\prime}$ with $\boldsymbol{A}^{[n+2]}$ to form the two-site tensor for the next step. We repeat this procedure until we arrive at the right-most tensor of the MPS. Sweeping from right to left can be done analogeously.
After sweeping through the MPS from left to right and back, we have performed a full time step Δt and obtain the updated MPS $|\Psi(t+\Delta t)\rangle_{\text{MPS}}$.
The implementation of TDVP2 is very similar to the implementation of the popular DMRG algorithm [14]; it is possible to obtain TDVP2 from DMRG by only changing a few lines of code [22]. It is worth to note that it is not necessary to compute the matrix exponentials (2.18) and (2.19) directly, which is a costly operation. Instead, we only need the result of multiplying the matrix exponential with the current state vector $\boldsymbol{\theta}^{[n,n+1]}$ or $\boldsymbol{A}^{[n+1]\prime}$, which can be computed efficiently, e.g., using the scipy function `scipy.expm_multiply`.
An efficient implementation also involves storing environment tensors of every site in a list and updating them during a sweep, such that the effective Hamiltonians do not have to be recomputed from scratch at each step.

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 often used to show the applicability of new algorithms. The system Hamiltonian of the spin-boson model is

$$\hat{H}_S = -\frac{1}{2}\Delta\hat{\sigma}_x + \frac{1}{2}\epsilon\hat{\sigma}_z, \tag{2.20}$$

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*

$$S(\omega) = \eta\frac{\omega\gamma}{\omega^2 + \gamma^2} \tag{2.21}$$

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

$$\alpha(\tau) \approx \sum_{k=0}^{K-1} \alpha_k(\tau) \coloneqq \sum_{k=0}^{K-1} g_k e^{-\omega_k\tau}.$$

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

$$\begin{aligned} g_0 &= \frac{\eta\gamma}{2}\left(\cot\left(\frac{\gamma}{2T}\right) - i\right), \quad \omega_0 = \gamma; \\ g_k &= \eta\frac{4\pi T^2 k\gamma}{4\pi^2T^2k^2 - \gamma^2}, \quad \omega_k = 2\pi Tk \quad \text{for } k \geq 1. \end{aligned} \tag{2.22}$$

At low temperatures the Matsubara expansion converges very slowly, as I will show in Section 4. Instead, one can use the Padé expansion

$$\begin{aligned} g_0 &= \eta\gamma\left(\frac{1}{\gamma\beta} - \frac{i}{2} - \sum_{j=1}^{K-1}\frac{2\tilde{\eta}_j\gamma\beta}{\tilde{\xi}_j^2 - \gamma^2\beta^2}\right), \quad \omega_0 = \gamma; \\ g_k &= \frac{2\tilde{\eta}_k\eta\gamma\tilde{\xi}_k}{\tilde{\xi}_k^2 - \gamma^2\beta^2}, \quad \omega_k = \frac{\tilde{\xi}_k}{\beta} \quad \text{for } k \geq 1. \end{aligned} \tag{2.23}$$

The constants $\tilde{\eta}_k$ and $\tilde{\xi}_k$ can be computed using the $[N-1/N]$ Padé spectrum decomposition (PDS) scheme [23] with $N = K - 1$.
The derivation of the expansion coefficients (2.22) and (2.23) is given in detail in Appendix A.

3 Implementing and Testing the HOPS method

3.1 Implementation

In this thesis we 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

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

Both the linear and the non-linear HOPS equations can be integrated by using regular numerical integration schemes, e.g., Euler or other Runge-Kutta methods. 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 $t + \Delta t$ 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:

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

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

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

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, for example with the python package `scipy.sparse`.

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),

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

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:

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

The last remaining problem is the computation of the memory term

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

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:

$$\begin{aligned} z^*_{\text{memory}}(t+\Delta t) &= \int_0^{t+\Delta t} \alpha^*(t+\Delta t - s) \left\langle \hat{L}^\dagger \right\rangle_s \mathrm{ds} = e^{-\omega^* \Delta t} \int_0^{t+\Delta t} \alpha^*(t-s) \left\langle \hat{L}^\dagger \right\rangle_s \mathrm{ds} \\ &= e^{-\omega^* \Delta t} z^*_{\text{memory}}(t) + e^{-\omega^* \Delta t} \int_t^{t+\Delta t} \alpha^*(t-s) \left\langle \hat{L}^\dagger \right\rangle_s \mathrm{ds} \\ &\approx e^{-\omega^* \Delta t} z^*_{\text{memory}}(t) + e^{-\omega^* \Delta t} \Delta t \alpha^*(0) \left\langle \hat{L}^\dagger \right\rangle_t \\ &\approx z^*_{\text{memory}}(t) - \omega^* \Delta t \, z^*_{\text{memory}}(t) + \Delta t \, g^* \left\langle \hat{L}^\dagger \right\rangle_t \end{aligned} \quad . \tag{3.2}$$

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 (2.20) with $\Delta = 1$ and $\epsilon = 0$. For the bath correlation function I use the simple expansion

$$\alpha(\tau) = ge^{-\omega\tau} \tag{3.3}$$

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]. I begin by testing the generation of the stochastic process. For this, 100, 1000, and 10000 realizations of the stochastic process are generated. The expectation value $\mathbb{E}[z_t z_0]$ is then computed and shown in Figure 3.1. One can see that the expectation value approaches the bath correlation function as the number of realizations is increased, meaning Condition (2.3) is well fulfilled.

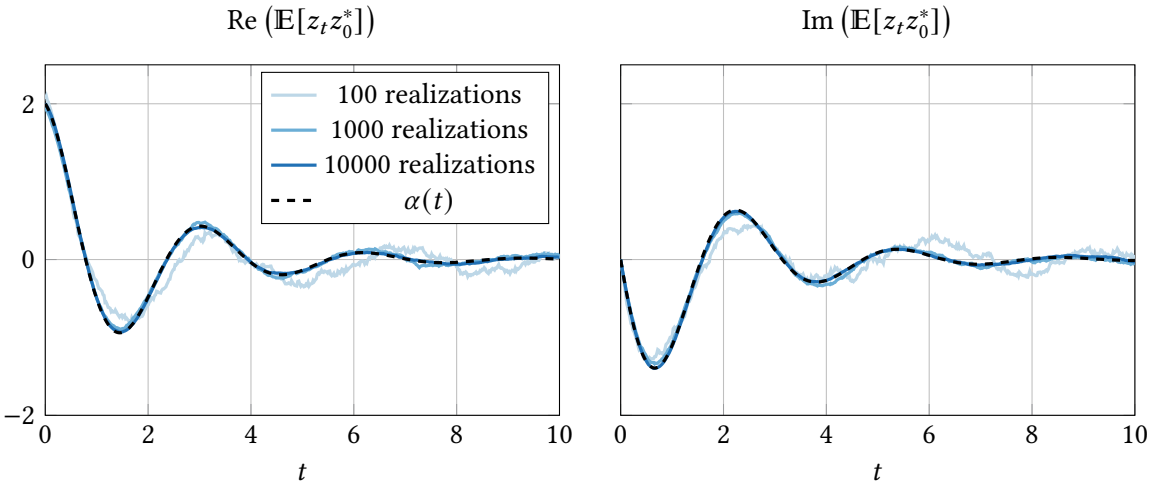


Figure 3.1 The real and imaginary parts of the expectation value $\mathbb{E}[z_t z_0^*]$ of the stochastic process used for testing HOPS with the spin-boson model are computed from 100, 1000, and 10000 realizations. One can see that the correlations approach the bath correlation function $\alpha(\tau)$ (3.3) with $g = 2$ and $\omega = 0.5 + 2i$.

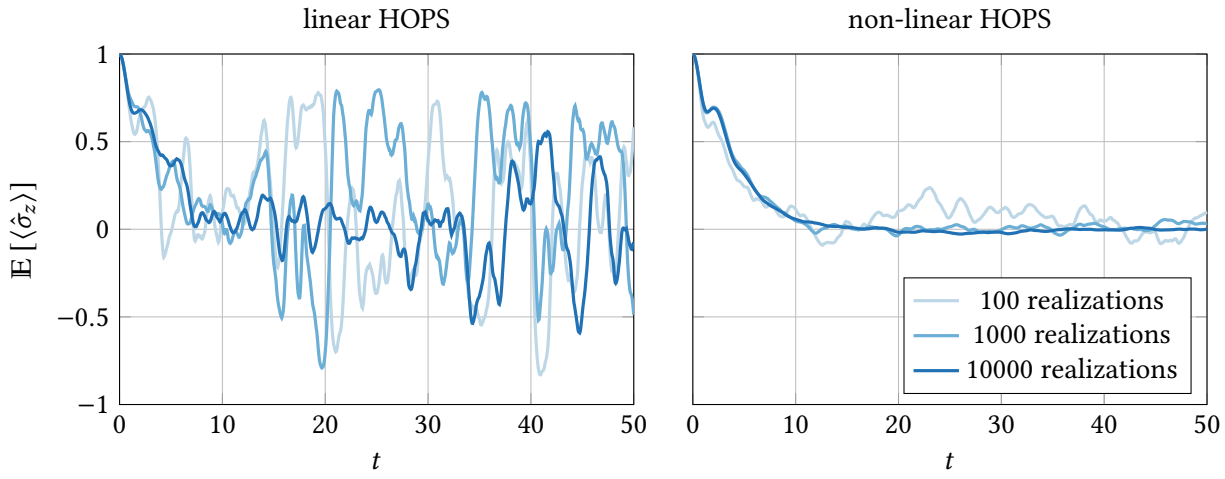


Figure 3.2 The dynamics of the spin-boson model are computed using the HOPS method. The stochastic expectation value of the $\hat{\sigma}_z$ operator is plotted against the time. The expectation value is taken over 100, 1000, and 10000 realizations of the stochastic processes. Both the linear (left) 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 $\Delta = 1$ and $\epsilon = 0$. The HOPS was truncated using simple truncation (2.14) with $N_{\text{trunc}} = 8$.

Next, I integrate both the linear and non-linear HOPS as detailed in Section 3.1. I use $N_{\text{trunc}} = 8$ and a time step of $\Delta t = 0.02$. The expectation value $\mathbb{E}[\langle \hat{\sigma}_z \rangle]$ is then computed by averaging over 100, 1000, and 1000 realizations. The result is shown in Figure 3.2. One can directly see that the linear HOPS converges much slower than the non-linear HOPS. The reason for this is that the states from different realizations of the noise have strongly different magnitudes, as can be seen in Figure 3.3. When computing the stochastic expectation value for linear HOPS (2.13), states with small magnitudes do not contribute much to the result. The expectation value is thus dominated by only a few states with large magnitudes, which leads to slow convergence. In contrast, the states in the non-linear HOPS are normalized, avoiding the problem and leading to faster convergence.
The convergence of the non-linear HOPS method with respect to the truncation dimension N_{trunc} is shown in Figure 3.4.
My results match well the ones obtained in [1].

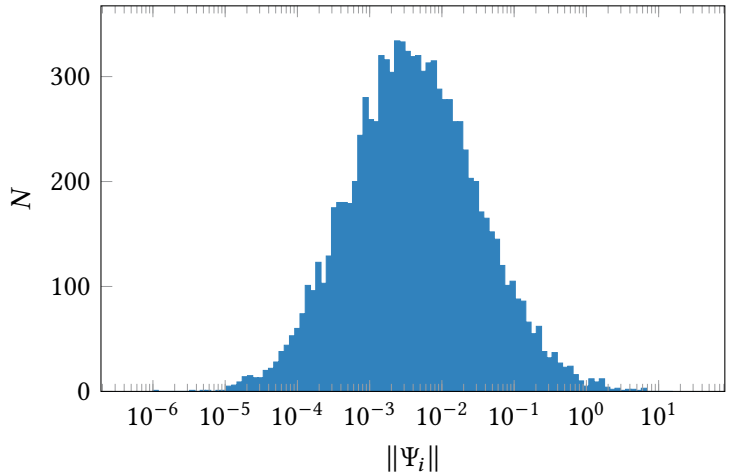


Figure 3.3 The magnitudes of states from 10000 linear HOPS realizations of the spin-boson model are shown in a histogram. There are big differences in the magnitudes of the different realizations, which leads to the problem that states with small magnitudes do not contribute much to the overall expectation value and can therefore be seen as wasted computation time.

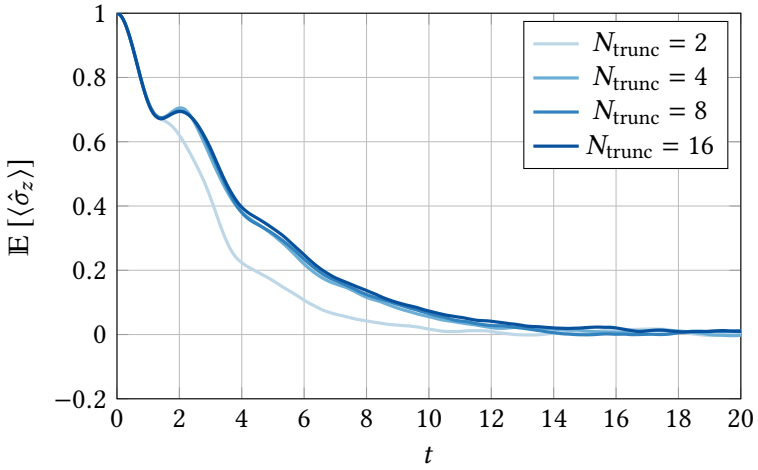


Figure 3.4 In this figure, the convergence of non-linear HOPS in the N_{trunc} parameter is shown. For different values of N_{trunc}, 10000 realizations of non-linear HOPS with the spin-boson model were computed each. The other parameters are the same as for Figure 3.2. $N_{\text{trunc}} = 8$, which was used for the full runs in Figure 3.2, is already well converged.

4 Implementing and Testing the HOMPS method

4.1 Implementation

Constructing the HOMPS MPO

The goal of this section is to rewrite the linear and non-linear HOPS Equations (2.9) and (2.12) in the MPS formalism. We will perform the derivation for the non-linear equation, following [2]. The linear equation can be derived analogeously.
We start by representing the full hierarchy at time t by a single quantum state

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

where the basis states $|l, n_1, n_2, \ldots, n_K\rangle$ are tensor products of system states $|l\rangle$ and auxillary *pseudo-Fock* states $|\mathbf{n}\rangle$. We choose a simple truncation condition $n_\mathrm{k} \in 0, 1, \ldots, N_\mathrm{trunc} - 1$. 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:

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

with the effective Hamiltonian

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

We now switch to the MPS formalism. We can write the full state (4.1) as an MPS

$$|\Psi_t\rangle = \sum_{l,\mathbf{n},\mathbf{i}} A^{[1],l}_{i_0,i_1} A^{[2],n_1}_{i_1,i_2} A^{[3],n_2}_{i_2,i_3}\cdots A^{[K+1]n_K}_{i_K,i_0} |l, n_1, n_2, \ldots, n_K\rangle ,$$

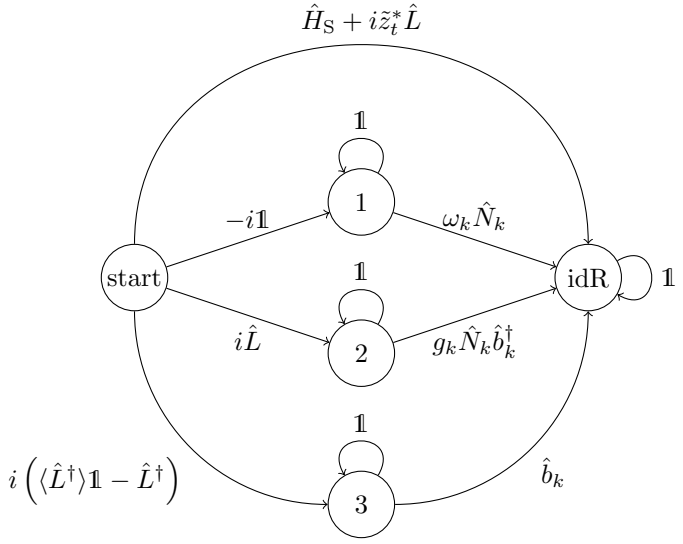


Figure 4.1 In this figure the state machine that can be used to generate the MPO (4.3) is sketched. The state machine can be constructed from Equation (4.2). For reference on how to use state machines to construct MPOs, see [24].

using $K+1$ tensors in total. next, we need to write the effective Hamiltonian (4.2) in MPO form. Here this is done with the finite state machine method discussed in [24]. By using the finite state machine depicted in Figure 4.1, we arrive at the following tensors:

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

Note that this construction is not unique. An alternative construction can be done by using the graph-based algorithm in [25].

The HOMPS with MPO (4.3) produces correct results, but becomes numerically unstable for large N_trunc. Better stability can be achieved by rescaling the auxillary states to

$$\widetilde{\boldsymbol{\Psi}}_t^{(\mathbf{n})} := \left(\prod_{k=1}^{K} n_k! |g_k|^{n_k} \right)^{-1/2} \boldsymbol{\Psi}_t^{(\mathbf{n})}$$

as proposed in [2]. The HOMPS expressed in terms of these rescaled states becomes

$$\frac{d}{dt} |\widetilde{\boldsymbol{\Psi}}_t\rangle = -i\hat{H}'_\mathrm{eff} |\widetilde{\boldsymbol{\Psi}}_t\rangle ,$$

with

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

where the raising and lowering operators have also been rescaled to

$$\hat{b}_k'^{\dagger} |\mathbf{n}\rangle := \sqrt{n_k + 1}\, |\mathbf{n} + \mathbf{e}_k\rangle ,$$
$$\hat{b}_k' |\mathbf{n}\rangle := \sqrt{n_k}\, |\mathbf{n} - \mathbf{e}_k\rangle .$$

The effective Hamiltonian in MPO form becomes

$$\begin{aligned} \boldsymbol{W}^{[1]\prime} &:= \begin{pmatrix} -i\mathbb{1} & i\hat{L} & i\left(\left\langle \hat{L}^{\dagger}\right\rangle_t \mathbb{1} - \hat{L}^{\dagger}\right) & \hat{H}_{\mathrm{S}} + i\tilde{z}_t^{*}\hat{L} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \\ \boldsymbol{W}^{[k+1]\prime} &:= \begin{pmatrix} \mathbb{1} & 0 & 0 & \omega_k \hat{N}_k \\ 0 & \mathbb{1} & 0 & \frac{g_k}{\sqrt{|g_k|}} b_k'^{\dagger} \\ 0 & 0 & \mathbb{1} & \sqrt{|g_k|}\hat{b}_k' \\ 0 & 0 & 0 & \mathbb{1} \end{pmatrix}, \quad k = 1, 2, \ldots, K, \end{aligned} \tag{4.4}$$

which can be derived similarily to (4.3).
A comparison of the default and rescaled HOMPS, highlighting the numerical stability issues, can be found in Appendix B.

Computing Expectation Values

The expectation value $\left\langle \hat{L}^{\dagger}\right\rangle_t$ can be easily computed from the MPS (4.3) by setting the index vector $\boldsymbol{n} = \mathbf{0}$:

$$\left\langle \hat{L}^{\dagger}\right\rangle_t = \frac{\langle \boldsymbol{\Psi}_t^{(\mathbf{0})} |\, \hat{L}^{\dagger}\, | \boldsymbol{\Psi}_t^{(\mathbf{0})}\rangle}{\langle \boldsymbol{\Psi}_t^{(\mathbf{0})} | \boldsymbol{\Psi}_t^{(\mathbf{0})}\rangle},$$

$$|\boldsymbol{\Psi}_t^{(\mathbf{0})}\rangle = \sum_{l,\boldsymbol{i}} A^{[1],l}_{i_0,i_1} A^{[2],0}_{i_1,i_2} A^{[3],0}_{i_2,i_3} \ldots A^{[K+1],0}_{i_K,i_0} \, |l\rangle .$$

Updating the Memory Terms

To compute the memory term

$$z^{*}_{\mathrm{memory}}(t) := \int_0^t \alpha^{*}(t-s) \left\langle L^{\dagger}\right\rangle_s \mathrm{d}s \approx \sum_{k=1}^{K} \int_0^t g_k e^{-\omega_k} \left\langle L^{\dagger}\right\rangle_s \mathrm{d}s$$

in the "shifted" noise (2.11) for $K > 1$ bath modes, we first split it into K terms

$$z^{*}_{\mathrm{memory}}(t) = \sum_{k=1}^{K} z^{*}_{\mathrm{memory},k}(t), \quad z^{*}_{\mathrm{memory},k}(t) := \int_0^t g_k e^{-\omega_k} \left\langle L^{\dagger}\right\rangle_s \mathrm{d}s.$$

Each of the terms $z^{*}_{\mathrm{memory},k}(t)$ can then be updated similar to (3.2).

4.2 Testing HOMPS with the High Temperature Spin-Boson Model

To test my implementation of the HOMPS, I start with a simple high temperature and low damping case of the spin-boson model (2.20). I use the Debye spectral density (2.21) with $T = 2$, $\gamma = 0.25$ amd $\eta = 0.5$; the parameters for the spin-boson model are chosen as $\epsilon = 2$ and $\Delta = -2$. The same parameters are used in [2] and [11]. The approximation of the bath correlation function is shown in Figure 4.2. For the used parameters one bath mode of the Matsubara approximation (2.22) is already enough to achieve a good approximation.

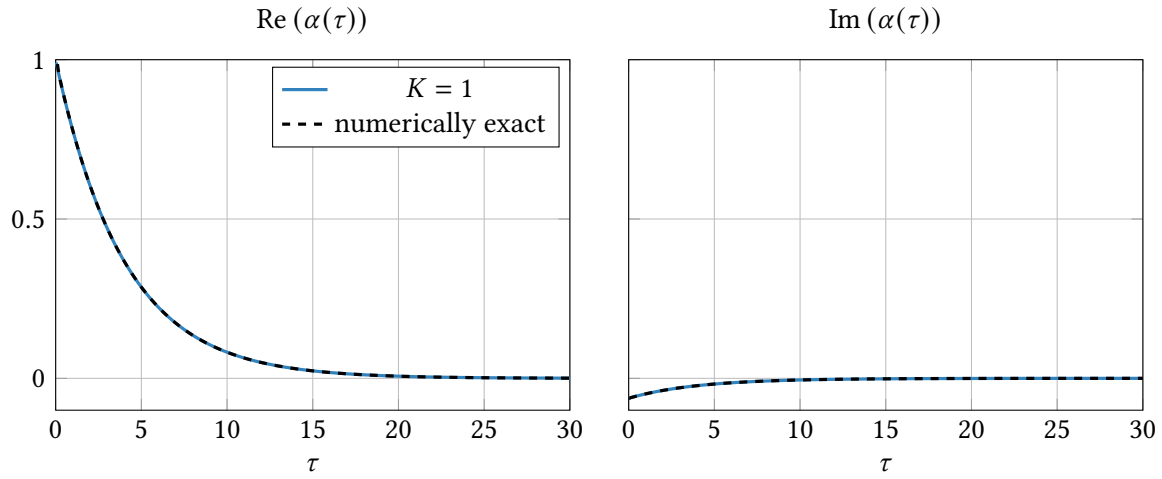


Figure 4.2 The approximation of the bath correlation function using the Debye spectral density (2.21) with $T = 2$, $\gamma = 0.25$ and $\eta = 0.5$ (high temperature, weak damping) is shown. The real and imaginary parts of the bath correlation function are shown on the left and right respectively. The numerically exact result is computed by replacing the integral with a sum. Already one term of the Matsubara approximation (2.22) is enough to converge to the numerically exact result.

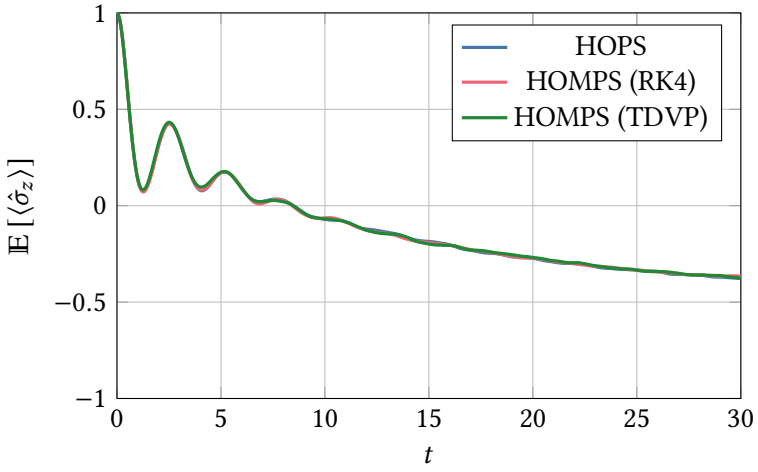


Figure 4.3 For the three different methods implemented (HOPS (RK4), HOMPS (RK4), and HOMPS (TDVP)), 10000 realizations are computed each. The stochastic expectation value $\mathbb{E}[\langle\hat{\sigma}_z\rangle]$ is then plotted against t. The parameters for the computation were chosen as $K = 1$, $N_{\text{trunc}} = 40$, $\Delta t = 0.002$.

Next, I compare the different methods of integrating the HOMPS. In Figure 4.3, 10000 realizations of the hierarchy are computed using RK4 and TDVP2 and compared with the HOPS method. The three methods produce similar dynamics. In the following, I use RK4 for integrating the HOMPS since it is slightly faster than TDVP2 in my implementation.
In Figure 4.4a the stochastic expectation value of $\hat{\sigma}_z$ is computed with the HOMPS using 100, 1000, and 10000 realizations of the stochastic process. The HOMPS are integrated using RK4 with $N_{\text{trunc}} = 40$ and a time step of $\Delta t = 0.06$. The truncation threshhold for the truncation of singular values is set to zero. This is possible since the maximum possible bond dimension for $K = 1$ bath mode is $\chi_{\text{max}} = \min\left(\dim\left(\mathcal{H}_S\right), N_{\text{trunc}}\right)$, where $\dim\left(\mathcal{H}_S\right)$ is the dimension of the system Hilbert space. In the case of the spin-boson model it holds $\dim\left(\mathcal{H}_S\right) = 2$. Therefore, the virtual bond dimension cannot exceed $\chi_{\text{max}} = 2$ and we do not need to truncate any singular values.
The convergence of the hierarchy with respect to the truncation dimension N_{trunc} is shown in Figure 4.4b. The results match well the ones obtained in [1] and [11].

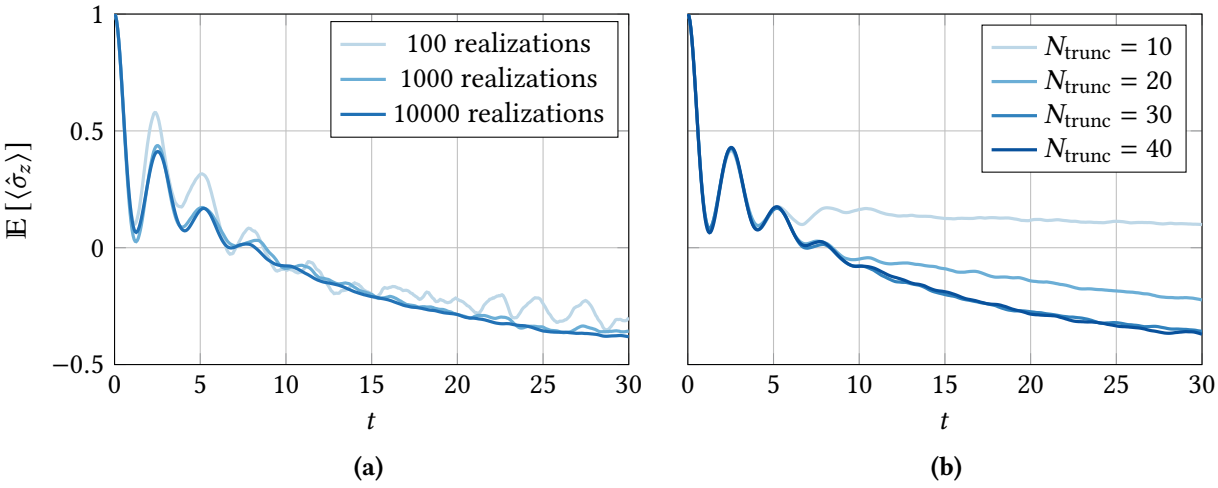


Figure 4.4 In this figure, the HOMPS is integrated using RK4 with a time step of $\Delta t = 0.06$. (a) 100, 1000, and 10000 realizations, each computed with $N_{\text{trunc}} = 40$. (b) For different values for N_{trunc}, 10000 realizations were computed each. $N_{\text{trunc}} = 40$ is well converged.

4.3 Testing HOMPS with the Low Temperature Spin-Boson Model

As a second test, I use a low temperature and high damping case of the spin-boson model (2.20). I again use the Debye spectral density and set the parameters to $T = 0.02$, $\gamma = 5$, $\eta = 0.5$, $\epsilon = 2$, $\Delta = -2$, which are also used in [1, 11]. The approximations of the bath correlation function using the Matsubara summation and Padé approximation methods are shown in Figures 4.5a and 4.5b respectively. At low temperatures the Padé approximation converges a lot faster than the Matsubara summation. The following computations are done with $K = 13$ terms of the Padé approximation, which yields a sufficiently converged BCF.

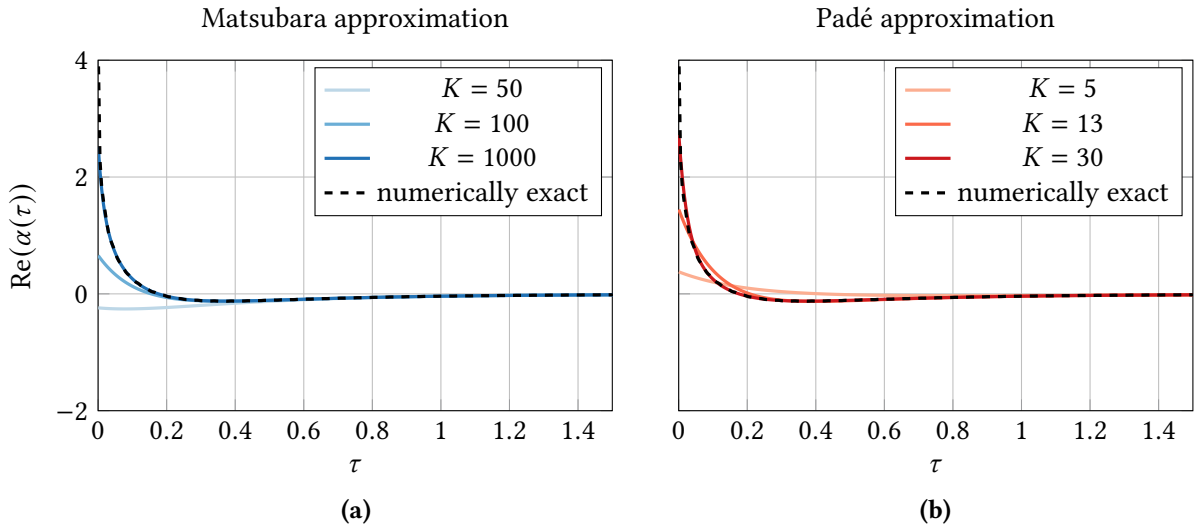


Figure 4.5 The approximation of the bath correlation function using the Debye spectral density (2.21) with $T = 0.02$, $\gamma = 5$ and $\eta = 0.5$ (low temperature, strong damping) is shown. The real part of the bath correlation function is approximated using the Matsubara approximation (2.22) and the Padé approximation (2.23) on the left and right respectively. The numerically exact result is computed by replacing the integral with a sum. One can see that the Padé approximation converges a lot faster than the Matsubara approximation. The imaginary part of the bath correlation function is not shown, as it is already well converged using $K = 1$ terms of either approximation.

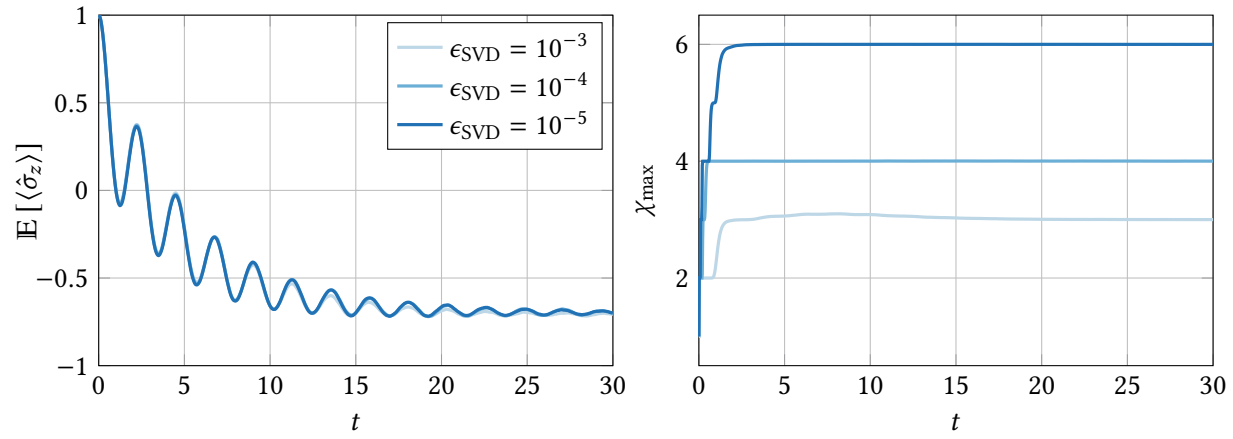


Figure 4.6 The HOMPS equations of the low temperature spin-boson model are integrated using different truncation threshholds ϵ_{SVD} for the singular value decomposition. The integration method used is RK4 with a time step of $\Delta t = 0.02$, $K = 13$ terms of the Padé approximation, and $N_{\text{trunc}} = 9$. The parameters for the spin-boson model are given in the text. On the left, the stochastic expectation value $\mathbb{E}[\langle\hat{\sigma}_z\rangle]$ is computed over 10000 realizations. Already at $\epsilon_{\text{SVD}} = 10^{-4}$, the method is well converged. On the right, the maximal bond dimension averaged over 10000 realizations is shown. Smaller truncation threshholds ϵ_{SVD} lead to larger bond dimensions, giving rise to higher computation times.

Because now multiple bath nodes are used, the MPS needs to be truncated in order to keep computation times managable. In Figure 4.6, the HOMPS equations are integrated using different values for the truncation threshhold ϵ_{SVD}. After each Singular Value Decomposition, singular values smaller than the truncation threshhold are omitted. Lower threshhold values lead to larger bond dimensions, which drastically increases computation times. For the spin-boson model, a truncation threshhold of $\epsilon_{\text{SVD}} = 1.e-3$ yields good results and leads to a maximal bond dimension of $\chi_{\text{max}} \approx 2$. Finally, I integrate the HOMPS equations using 100, 1000, and 10000 realizations of the stochastic process in Figure 4.7. The parameters used are $K = 13$, $N_{\text{trunc}} = 9$, and $\epsilon_{\text{SVD}} = 10^{-3}$; The integration method is RK4. The results match well the ones obtained in [1, 11]. Note that the effect of the noise is low compared to the high temperature case (compare figure 4.4a and 4.7).

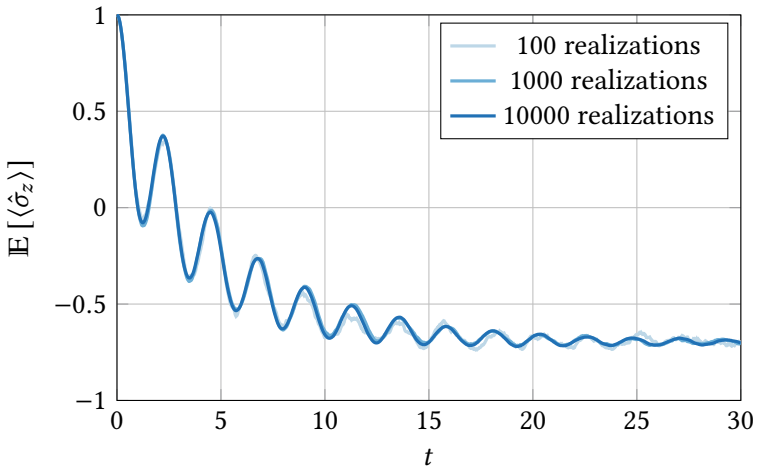


Figure 4.7 In this figure, the HOMPS equations of the low temperature spin-boson model are integrated using RK4 with a time step of $\Delta t = 0.02$, $K = 13$ terms of the Padé approximation, and $N_{\text{trunc}} = 9$. The parameters for the spin-boson model are given in the text. The stochastic expectation value is computed using 100, 1000, and 10000 realizations of the stochastic process.

5 Conclusion

As we have seen, the HOPS method is able to relieably simulate the dynamics of non-Markovian open quantum systems. If one wants to use multiple bath modes, the HOMPS method is a good strategy to minimize memory requirements by compressing the quantum state into Matrix Product form. We discussed two integration methods, RK4 and TDVP2, which are both viable for solving the HOMPS equations. There are however some additional steps that can be made to further improve the methods.
First, there exists an alternative realization of the non-Markovian stochastic Schrödinger equation [11], where, in contrast to (2.3), a stochastic process with non-zero correlations $\mathbb{E}[z_t z_s] \neq 0$ is used. This leads to a smaller bath correlation function, reducing the amplitude of the non-Markovian memory term at high temperatures. For models that need many terms for approximating the bath correlation function sufficiently well, this can be a crucial improvement.
Second, one has to consider how systems consisting of multiple subsystems, i.e., many-body systems, should be treated in the HOMPS method. Because of the exponential growth of the Hilbert space, using one tensor for the complete many-body system is inefficient. It is therefore better to split the system into smaller subsystems and to represent each subsystem by a single tensor. A question that arises is how to connect the different tensors. One idea is to use an MPS, repeating a structure where each physical tensor is followed by multiple bath mode tensors, which is done in [2]. Alternatively, it could be beneficial to use a tree tensor network [26] instead. The physical subsystems would then be represented by rank-4 tensors, where two legs are used to connect to the neighbouring subsystems, one leg is the physical leg, and the last leg is connected to an MPS representing the bath modes. TDVP2 can be adapted for tree tensor networks [26]. This approach would help to further reduce the memory requirements for computing the non-Markovian dynamics of open many-body systems.
In conclusion, I believe that the HOPS and HOMPS methods are very useful for simulating open quantum systems and will be widely used for studying new systems and comparing experiments to theory in the future.

A Approximating the BCF of the Spin-Boson model

In this section we derive the coefficients (2.22) and (2.23) that can be used to approximate 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*

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

as a sum of exponentials

$$\alpha(\tau) \approx \sum_{k=0}^{K-1} g_k e^{-\omega_k\tau}.$$

We first note that the integrand $I(\omega,\tau)$ is symmetric with respect to ω; $I(-\omega,\tau) = I(\omega,\tau)$. Therefore, we can extend the integral over the negative real axis:

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

Next, we write the sine and cosine in terms of complex exponentials and use the identity $\coth(x) = 2f^{\mathrm{Bose}}(2x) - 1$ with the *Bose function*

$$f^{\mathrm{Bose}}(x) \coloneqq \frac{1}{1-e^{-x}}$$

to arrive at

$$\alpha(\tau) = \frac{1}{\pi}\int_{-\infty}^\infty S(\omega)\left[f^{\mathrm{Bose}}(\omega\beta) - 1\right]\mathrm{d}\omega. \tag{A.1}$$

In the following, we will derive two different approximations of this integral.

Matsubara frequency summation

The Matsubara frequency summation uses the residue theorem to turn the integral (A.1) into an infinite sum. This sum then has to be truncated to get an approximation with a finite number of terms. The resulting approximation works well at high temperatures, but converges only slowly at low temperatures. Consider the line integral in Figure A.1. Because of the exponential term $e^{i\omega\tau}$, the contribution along L_2 vanishes as we take the limit $R \to \infty$. This then leaves us with

$$\begin{aligned}\alpha(\tau) &= \frac{1}{\pi}\oint_\xi S(\omega)\left[f^{\mathrm{Bose}}(\omega\beta) - 1\right]\mathrm{d}\omega = 2i\sum_{\{\omega_j\}}\mathrm{Res}\left(S(\omega)e^{-i\omega\tau}\left[f^{\mathrm{Bose}}(\omega\beta) - 1\right];\omega_j\right)\\ &= 2i\sum_{\{\omega_j'\}}\mathrm{Res}\left(S(\omega);\omega_j'\right)e^{i\omega_j'\tau}\left[f^{\mathrm{Bose}}(\omega_j'\beta) - 1\right] + 2i\sum_{\{\omega_j''\}}\mathrm{Re}\left(f^{\mathrm{Bose}}(\omega\beta);\omega_j''\right)S(\omega_j'')e^{i\omega_j''\tau},\end{aligned}$$

where ω_j are the poles of $S(\omega)\left[f^{\mathrm{Bose}}(\omega\beta) - 1\right]$, ω_j' the poles of $S(\omega)$, ω_j'' the poles of $f^{\mathrm{Bose}}(\omega\beta)$, and we have used the residue theorem. We also assumed that $\omega_i' \neq \omega_j''$ for all i,j, which is the case for almost all γ,β. Next, we need to compute all poles and residues. The Debye spectral density has simple poles at $\omega_j' = \pm i\gamma$, with residues

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

The poles of the Bose function lie on the imaginary axis, $\omega_j'' = 2\pi i n/\beta$ with $n \in \mathbb{Z}$, and are again simple poles with residues

$$\mathrm{Re}\left(f^{\mathrm{Bose}}(\omega\beta); \frac{2\pi i n}{\beta}\right) = \lim_{\omega \to 2\pi i n/\beta} \frac{(\omega - 2\pi i n/\beta)}{1 - e^{-\omega\beta}} = \lim_{x \to 0} \frac{x}{1 - e^{-\beta x - 2\pi i n}} = \lim_{x \to 0} \frac{1}{\beta e^{-\beta x}} = \frac{1}{\beta},$$

where we substituted $x = \omega - 2\pi i n/\beta$ and used the rule of L'Hopital.
With this, we are now equipped to solve the integral:

$$\alpha(\tau) = \frac{\eta\gamma}{2} e^{-\gamma\tau} \left[\cot(\gamma\beta) - i\right] e^{-\gamma\tau} + \sum_{k=1}^{\infty} \eta \frac{4\pi T^2 k\gamma}{4\pi^2 T^2 k^2 - \gamma^2} e^{-2\pi T k \tau} = \sum_{k=0}^{\infty} g_k e^{-\omega_k \tau}$$

with the coefficients (2.22). The result is similar to the one obtained in [27] (up to a different normalization).

Padé spectrum decomposition

The Padé spectrum decomposition is a method to approximate the Bose function $f_{\mathrm{Bose}}(x)$ by a finite sum [23]. We will use the Padé approximant $[N-1/N]$ with $N \equiv K-1$ to rewrite

$$\alpha(\tau) \approx \frac{1}{\pi} \int_{-\infty}^{\infty} S(\omega) e^{i\omega\tau} \left[\frac{1}{\omega\beta} - \frac{1}{2} + \sum_{k=1}^{K-1} \frac{2\tilde{\eta}_k \omega_\beta}{(\omega\beta)^2 + \tilde{\xi}_k^2}\right] \mathrm{d}\omega,$$

where the derivation of the constants $\tilde{\eta}_k$ and $\tilde{\xi}_k$ can be found in [23]. We can now proceed in a similar way to the Matsubara frequency summation. We will again consider the line integral along the contour in Figure A.1 and use the residue theorem. The poles and residues of the new terms are

$$\mathrm{Res}\left(\frac{1}{\omega\beta}; 0\right) = \frac{1}{\beta}$$

and

$$\mathrm{Res}\left(\frac{2\tilde{\eta}_k \omega\beta}{(\omega\beta)^2 + \tilde{\xi}_k^2}; \pm i\frac{\tilde{\xi}_k}{\beta}\right) = \frac{\tilde{\eta}_k}{\beta}.$$

A bit of algebra yields the final result

$$\alpha(\tau) \approx \eta\gamma \left[\frac{1}{\gamma\beta} - \frac{i}{2} - \sum_{k=1}^{K-1} \frac{2\tilde{\eta}_k \gamma\beta}{\tilde{\xi}_k^2 - \gamma^2\beta^2}\right] e^{-\gamma\tau} + \sum_{k=1}^{K-1} \frac{2\tilde{\eta}_k \eta\gamma\tilde{\xi}_k}{\tilde{\xi}_k^2 - \gamma^2\beta^2} e^{-\frac{\tilde{\xi}_k}{\beta}\tau} = \sum_{k=0}^{K-1} g_k e^{-\omega_k \tau}$$

with the coefficients (2.23).

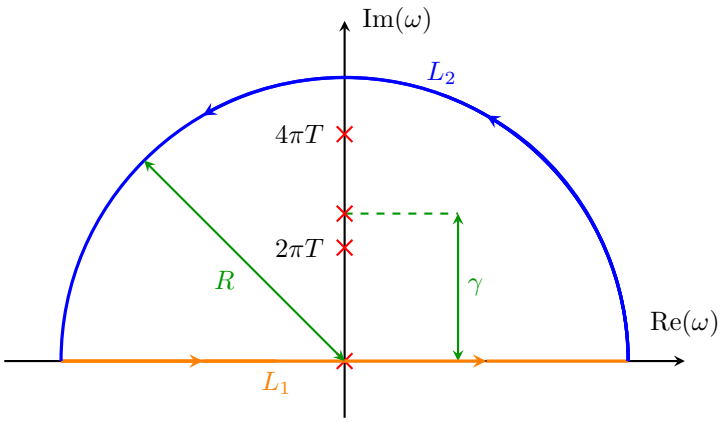


Figure A.1 In this figure, the line integral for computing the expansion of the bath correlation function is shown.

B Stability of the default and rescaled HOMPS

The non-rescaled HOMPS (4.3) has numerical stability problems for higher values of N_{trunc}. To illustrate the problem, I plot the expectation value $\mathbb{E}\left[\langle\hat{\sigma}_z\rangle_t\right]$ of the high temperature spin-boson model in Figure B.1, using both the non-rescaled HOMPS (4.3) and the rescaled HOMPS (4.4). For a better comparison, the stochastic process was set to $z(t) = 0$ for both methods. A high truncation value of $N_{\text{trunc}} = 40$ was used. One can see that the non-rescaled HOMPS diverges after a short time and does not produce the correct behaviour.

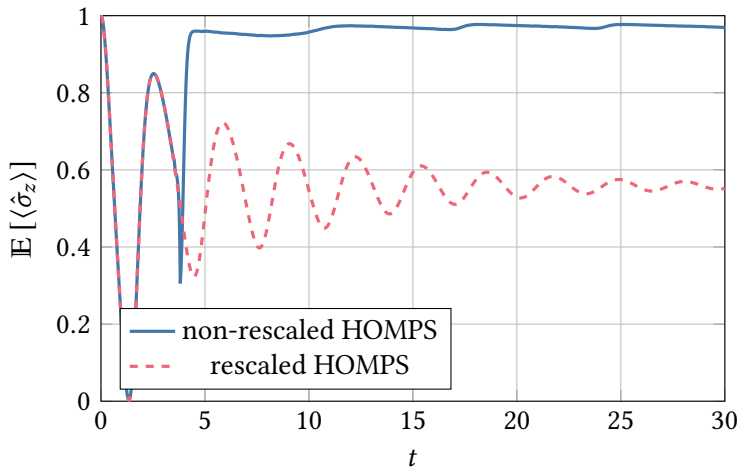


Figure B.1 The expectation value $\mathbb{E}\left[\langle\hat{\sigma}_z\rangle\right]$ is computed for both the non-rescaled and the rescaled HOMPS. I use the high temperature spin-boson model with the same parameters as in Figure 4.4a, but only compute a single realization whithout noise ($z(t) = 0$). One can see that the non-rescaled version of HOMPS is unstable, diverging after a short time.

The reason for this instability could be the fact that different auxillary states have vastly different magnitudes in the non-rescaled version of HOMPS. In Figure B.2, I plot the magnitudes of the different auxillary states of both the non-rescaled and rescaled HOMPS against time. One can see that the non-rescaled version produces magnitudes that differ by up to 15 orders of magnitudes, whereas in the rescaled version they differ only up to 7 orders of magnitude. In the HOMPS method addition of tensors is performed, which could lead to cancellation, a well-known limitation of floating point arithmetic. The rescaling "normalizes" the auxillary states and increases numerical stability.

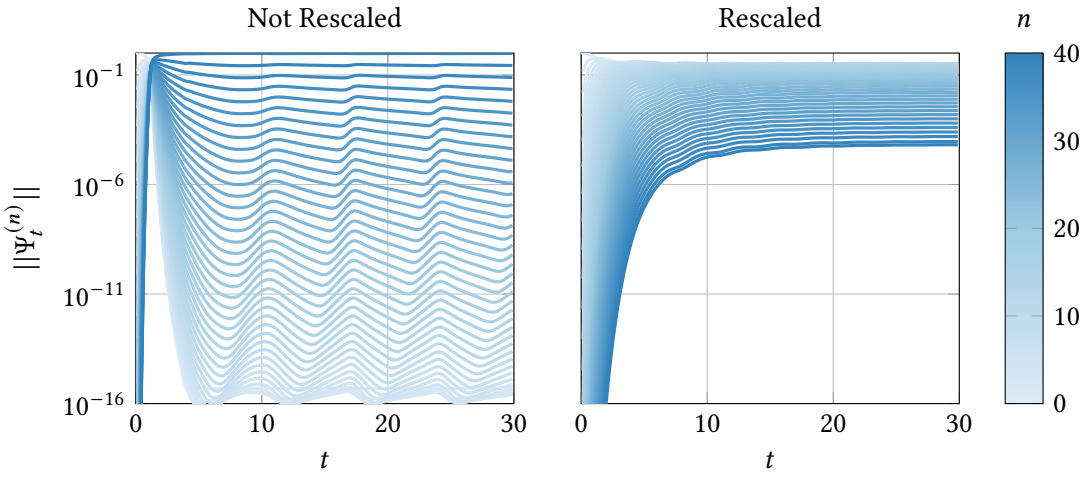


Figure B.2 In this figure, the magnitudes of different HOMPS auxillary states $\Psi_t^{(n)}$ are plotted against t. I use the high temperature spin-boson model with the same parameters as in Figure 4.4a, but only compute a single realization whithout noise ($z(t) = 0$). One can see that the magnitudes of different auxillary states take on vastly more widespread values when using the non-rescaled version of HOMPS than when using the rescaled version.

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 Oct. 2014), p. 150403.

[2] 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 Mar. 2022), p. L030202.

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

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

[5] 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.

[6] W. T. Strunz, L. Diósi, and N. Gisin. "Open System Dynamics with Non-Markovian Quantum Trajectories". In: *Phys. Rev. Lett.* 82 (9 Mar. 1999), pp. 1801–1805.

[7] I. Percival. "Quantum State Diffusion". In: *Quantum State Diffusion, by Ian Percival, pp. 198. ISBN 0521620074. Cambridge, UK: Cambridge University Press, January 1999.* (Jan. 1999).

[8] W. T. Strunz. "Linear quantum state diffusion for non-Markovian open quantum systems". In: *Physics Letters A* 224.1 (1996), pp. 25–30. ISSN: 0375-9601.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] G. E. P. Box and M. E. Muller. "A Note on the Generation of Random Normal Deviates". In: *The Annals of Mathematical Statistics* 29.2 (1958), pp. 610 –611.

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

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

[16] *Density Matrix Renormalization Group Algorithm (DMRG).* URL: https://tensornetwork.org/mps/algorithms/dmrg/ (visited on 06/12/2023).

[17] *The time-evolution block-decimation (TEBD).* URL: https://tensornetwork.org/mps/algorithms/timeevo/tebd.html (visited on 06/12/2023).

[18] *The time-dependent variational principle (TDVP).* URL: https://tensornetwork.org/mps/algorithms/timeevo/tdvp.html (visited on 06/05/2023).

[19] S. Paeckel et al. “Time-evolution methods for matrix-product states”. In: *Annals of Physics* 411 (Dec. 2019), p. 167998.

[20] W. Li, J. Ren, and Z. Shuai. “Numerical assessment for accuracy and GPU acceleration of TD-DMRG time evolution schemes”. In: *The Journal of Chemical Physics* 152.2 (Jan. 2020). 024127. ISSN: 0021-9606. eprint: https://pubs.aip.org/aip/jcp/article-pdf/doi/10.1063/1.5135363/16659217/024127_1_online.pdf.

[21] L. Vanderstraeten, J. Haegeman, and F. Verstraete. “Tangent-space methods for uniform matrix product states”. In: *SciPost Phys. Lect. Notes* (2019), p. 7.

[22] J. Haegeman et al. “Unifying time evolution and optimization with matrix product states”. In: *Phys. Rev. B* 94 (16 Oct. 2016), p. 165116.

[23] J. Hu et al. “Padé spectrum decompositions of quantum distribution functions and optimal hierarchical equations of motion construction for quantum open systems”. In: *The Journal of Chemical Physics* 134.24 (June 2011). 244106. ISSN: 0021-9606. eprint: https://pubs.aip.org/aip/jcp/article-pdf/doi/10.1063/1.3602466/15438865/244106_1_online.pdf.

[24] 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 Apr. 2016), p. 155139.

[25] J. Ren, W. Li, T. Jiang, and Z. Shuai. “A general automatic method for optimal construction of matrix product operators using bipartite graph theory”. In: *The Journal of Chemical Physics* 153.8 (Aug. 2020). 084118. ISSN: 0021-9606. eprint: https://pubs.aip.org/aip/jcp/article-pdf/doi/10.1063/5.0018149/15579678/084118_1_online.pdf.

[26] A. Holzner, A. Weichselbaum, and J. von Delft. “Matrix product state approach for a two-lead multilevel Anderson impurity model”. In: *Phys. Rev. B* 81 (12 Mar. 2010), p. 125126.

[27] 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.

Resources for Plotting

[1] M. H. Cynthia Brewer and T. P. S. University. *COLORBREWER 2.0.* URL: https://colorbrewer2.org/ (visited on 06/09/2023).

[2] J. Goedhart. *Color blind friendly palettes for data visualization.* URL: https://thenode.biologists.com/data-visualization-with-flying-colors/research/ (visited on 06/09/2023).