

Jacobs University Bremen
Department of Physics and Earth Sciences

Generating Structured Thermal Noise for Quantum Dynamical Systems

BSc Thesis in Physics

as part of the course
CA08-200304 Thesis Physics

by

Cristian Emiliano Godínez Ramírez

Supervised by Prof. Ulrich Keinekathöfer

Second reader: Prof. Stefan Kettemann

Abstract

To understand some of the most fundamental processes responsible for life on Earth, such as photosynthesis, quantum dynamical descriptions are of utmost importance. This analysis can be performed by studying the so-called open systems through a variation of the mean-field Ehrenfest Dynamics methods, known as Numerical Integration of Schrödinger equation (NISE). A key ingredient for the calculations using NISE is the time-dependent site energies. Such fluctuations can be obtained from Molecular Dynamics (MD) simulations. However, these methods are usually computationally expensive, and do not provide the desired accuracy. An alternative approach is the generation of structured noise following the bath spectral density. This has been done previously for systems described by simple spectral functions.

In this thesis I implement an enveloping algorithm capable of generating noise with any arbitrary spectral density. Several tests are performed to verify the validity, scope and limitations of the algorithm. After finding the parameters necessary to replicate more complex spectral shapes, this noise is used in the calculation of population dynamics for a collection of test systems. These simulations provide insight on the parameter regimes and specific considerations for the implementation of the noise in these calculations, and yield useful information in the performance of NISE under the new frameworks.

The results introduced here expand the ensemble of quantum dynamical processes that can be studied, that until now was limited to those described by simple spectra or by inaccurate and expensive computations. Thus, the analysis of quantum dynamics involving more complex system-bath interactions is enabled, along with an increase in the computational feasibility and accuracy.

Contents

1	Introduction	3
2	Theory	5
2.1	NISE	5
2.2	Autocorrelation and Spectral density	7
2.3	Fourier transform	8
2.3.1	Conventions	8
2.3.2	Discrete Transform	9
2.3.3	Continuous-Discrete Relation	10
2.3.4	Fast Fourier Transform	10
2.4	Power Spectral Density	10
2.4.1	Theoretical Spectral Shapes	11
3	Implementation	14
3.1	Algorithm	14
3.2	Units	15
3.3	Noise Generation	16
3.4	Autocorrelation	18
3.5	Spectral Density	19
3.6	Population Dynamics	21
4	Results	23
4.1	Previous Benchmarks	23
4.2	New Benchmarks	25
4.3	Performance in population dynamics calculations	29
5	Conclusions	36
6	Outlook	37
7	Acknowledgments	39
A	Appendix	41
A.1	Proof of correctness	41
A.2	Constants	42
A.3	Alternative integration methods	42
A.4	Testing System: Hamiltonian and Noise generation	43
	References	44

1 Introduction

The biological process by which plants and other photosynthetic systems convert the raw energy from sunlight into stable chemical energy plays a protagonist role to power life on Earth [1]. This makes of Light-harvesting (LH) one of the main processes of energy acquisition in our planet [2]. The process starts when the photon first enters the LH complex and it excites a pigment, or to what we will refer more abstractly as “site”, and since the system encompass many of these sites, we can refer to it as a multi-site system. The exciton then can transfer from one site to the other, making its way to the reaction center where charge separation is initiated. As classical diffusion fails to describe important quantum phenomena, such as quantum coherence [3], the implementation of quantum dynamical approaches becomes ubiquitous when aiming to describe such processes [4].

Over the past decades, an effort has been made to understand the dynamics of LH systems [5], where analytical solutions have been found for simple isolated systems [6]. However, a full description is only possible when all the components are taken into account. It then becomes pertinent to study, not only the interactions between different sites, but also the effects of the surrounding environment on the system [7].

The coupling of these so-called open systems can be seen as an interaction of the quantum system with the environment acting as a thermal bath [8]. These open systems can be studied under a reduced density matrix formalism, yielding analytically correct results when the Hamiltonian and spectral densities are specified beforehand [9][10]. One of such methods is the so-called Hierarchy Equations Of Motion (HEOM) [11][12][13]. The HEOM are a set of infinitely many coupled equations that describe a Non-Markovian time evolution of a system-bath interaction. Since this method does not assume a relative strength of the system effect on the bath, known as “quantum feedback”, it can correctly describe quantum coherence in the system with arbitrary accuracy [14]. Nevertheless, this method becomes computationally unfeasible with the increase of accuracy and bath complexity.

Consequently, a less expensive alternative based on a semi-classical approach can be employed. One of the most widely used methods is the mean-field Ehrenfest dynamics that uses the time-dependant Hamiltonian of the system to solve the time dependent Schrödinger equation [15][16]. The exciton dynamics can then be obtained from the resulting wave function by averaging over an ensemble of realizations to obtain physically meaningful results [17]. In this thesis we implement a variant of the Ehrenfest dynamics based on a high temperature limit approximation and the assumption that the quantum feedback can be neglected. This method is known as Numerical Integration of Schrödinger equation (NISE). In NISE, the effects of the bath on the system are simulated through time-dependant fluctuations in the site energies. For the aforementioned assumptions, an important relation arises between the autocorrelation of these fluctuations and the spectral density of the bath Hamiltonian [2][18][19].

Generating site fluctuations that follow a desired spectral density would allow for the calculation of the time-dependant Schrödinger equation without explicitly calculating the classical bath trajectory [8]. Until now, this has only been possible for simple spectral densities, or through methods yielding inaccurate results. In this thesis, I am interested in the implementation of an algorithm based on random sampling of white Gaussian noise and the Discrete Fourier Transform (DFT). This will allow us to generate these site fluctuations with an arbitrary spectral density. Further, I will investigate its accuracy when employed in population dynamics, the parameter regimes over which it yields feasible results when compared to methods such as HEOM, and its consequent limitations. Such implementation would enable more complex and accurate calculations of system-bath interactions, resembling the ones occurring in nature, specifically those modeled by Lorentzian spectral densities.

2 Theory

2.1 NISE

Due to the size of the molecular systems we aim to study, it is useful to split the Hamiltonian describing the compound system \hat{H} into the system, bath and system-bath coupling components, \hat{H}_S , \hat{H}_B and \hat{H}_{SB} , respectively.

$$\hat{H} = \hat{H}_S + \hat{H}_B + \hat{H}_{SB} \quad (2-1)$$

We look at a system made up of M coupled sites, with site energies E_i , and coupling between sites i and j of $V_{i,j}$. This yields the time independent Hamiltonian given by [17]:

$$\hat{H}_S = \sum_i E_i |i\rangle \langle i| + \sum_{i \neq j} V_{i,j} |j\rangle \langle i| \quad (2-2)$$

One the other hand, the bath is described by a collection of harmonic oscillators ζ , given by [17]:

$$\hat{H}_B = \sum_{\zeta} \frac{\hat{p}_{\zeta}^2}{2m_{\zeta}} + \frac{m_{\zeta}\omega_{\zeta}^2\hat{x}_{\zeta}^2}{2} \quad (2-3)$$

With \hat{p}_{ζ} , \hat{x}_{ζ} , m_{ζ} and ω_{ζ} the momentum, position, mass and frequency of the bath oscillator ζ respectively. The frequencies ω_{ζ} are limited by a spectral density obtained experimentally or through MD simulations [8].

The coupling Hamiltonian between system and bath can be seen as a sum of products of a system operator, assumed to be site diagonal for excitonic systems [18], and a bath operator $\hat{\Phi}_i$ acting on the diagonal entries, thus, leaving unaltered the couplings. Then \hat{H}_{SB} is given as in [17]:

$$\hat{H}_{SB} = \sum_i \hat{\Phi}_i |i\rangle \langle i| \quad (2-4)$$

If in addition, we assume $\hat{\Phi}_i$ to be linear in the bath coordinates, i.e. [17]:

$$\hat{\Phi}_i = \sum_{\zeta} c_{i\zeta} \hat{x}_{\zeta} \quad (2-5)$$

Where the coefficients $c_{i\zeta}$ define the coupling between the site i and bath oscillator ζ as dictated by the spectral density. Thus, the Hamiltonian of the coupling becomes:

$$\hat{H}_{SB} = \sum_i \sum_{\zeta} c_{i\zeta} \hat{x}_{\zeta} |i\rangle \langle i| \quad (2-6)$$

Looking now at the time-dependant Schrödinger equation for the complete system and implementing equation 2-1, we obtain:

$$i\hbar \frac{\partial}{\partial t} |\psi(t)\rangle = \left(\hat{H}_S + \hat{H}_B + \hat{H}_{SB} \right) |\psi(t)\rangle \quad (2-7)$$

Next, we split this expression into one equation propagating the system coordinates, and a second one propagating the bath coordinates [8]. Here we assume the propagation of the bath trajectory to be independent of the system, i.e. quantum feedback can be neglected. Under this approximation, if the bath starts in equilibrium, it will remain in equilibrium [18]. And even though this might restrict our calculations to cases where the system-bath couplings are small, it allows us to calculate the bath dynamics before the quantum calculations of the system are performed. As a result, we obtain [17]:

$$i\hbar \frac{\partial}{\partial t} |\psi^B(t)\rangle = \hat{H}_B |\psi^B(t)\rangle \quad (2-8)$$

Then, after taking the expectation value with respect to the bath, equation 2-7 becomes [17]:

$$i\hbar \frac{\partial}{\partial t} |\psi^S(t)\rangle = \left(\hat{H}_S + \langle \psi^B(t) | \hat{H}_{SB} | \psi^B(t) \rangle \right) |\psi^S(t)\rangle \quad (2-9)$$

Using equations 2-2 and 2-4 and integrating out the bath coordinates, we can then define an effective Hamiltonian [17]:

$$\begin{aligned} \hat{H}_S^{eff} &= H_S + \sum_i \langle \psi^B(t) | \hat{\Phi}_i | \psi^B(t) \rangle |i\rangle \langle i| \\ &= \sum_i (E_i + \Delta E_i(t)) |i\rangle \langle i| + \sum_{i \neq j} V_{i,j} |j\rangle \langle i| \end{aligned} \quad (2-10)$$

In which the site fluctuation term $\Delta E_i(t) = \langle \psi^B(t) | \hat{\Phi}_i | \psi^B(t) \rangle$ is the result of the coupling of the environment on the system.

To determine this time-dependant fluctuations, the bath trajectory given by equation 2-8 needs to be solved simultaneously with the Schrödinger equation of the correspondent system by means of \hat{H}_S^{eff} . These trajectories usually come from MD or other classical simulations. Instead, in this thesis an enveloping algorithm is employed to generate these site fluctuations, following a desired autocorrelation function related to the spectral density of the bath. With this addition, explicitly calculating the bath trajectories is no longer needed, which results in less computationally expensive simulations.

We proceed to solve the time-dependent Schrödinger equation in small steps Δt where $\hat{H}(t)$ is assumed to be constant [8]. This is done by defining a so-called time evolution operator that takes the wave function from t to $t + \Delta t$ [17]:

$$\mathcal{U}(t + \Delta t, t) = \exp \left(-i\hat{H}(t) \frac{\Delta t}{\hbar} \right) \quad (2-11)$$

Going to the eigenbasis, this operator simplifies to:

$$\tilde{\mathcal{U}}(t + \Delta t, t) = \sum_{\alpha} \exp \left(-i\epsilon_{\alpha}(t) \frac{\Delta t}{\hbar} \right) |\alpha\rangle \langle \alpha| \quad (2-12)$$

With α the eigenstate and ϵ_α its eigenvalue. The eigenvectors of the Hamiltonian $H(t)$ can be used to define the transformation matrix $C(t)$ between site and eigenbasis. Starting from the site basis, the time evolution of the wave function is [6]:

$$\psi(t + \Delta t) = C^\dagger(t)\tilde{\mathcal{U}}(t + \Delta t, t)C(t)\psi(t) \quad (2-13)$$

Alternatively, we can define the time evolution in the eigenbasis as [6]:

$$\begin{aligned} \tilde{\psi}(t + \Delta t) &= \tilde{\mathcal{U}}(t + \Delta t, t)C(t)C^\dagger(t - \Delta t)\tilde{\psi}(t) \\ &= \tilde{\mathcal{U}}(t + \Delta t, t)S(t)\tilde{\psi}(t) \end{aligned} \quad (2-14)$$

Where $S(t) = C(t)C^\dagger(t - \Delta t)$ is the non-adiabatic coupling. The population dynamics can be obtained by taking the absolute value squared of the wave function from 2-13 or 2-14.

To attain physically meaningful results, one must verify the correct order of the eigenvalues in $C(t - \Delta t)$ and $C(t)$, and to average over different realizations of site fluctuations when calculating the trajectory of the wave function. The latter is done to obtain physically meaningful results, due to the thermodynamic nature of the fluctuations.

2.2 Autocorrelation and Spectral density

The degrees of freedom (DOF) of the bath, i.e. position and momentum, are assumed to be harmonic as discussed in the previous section. As a result, in the high temperature limit their expectation values follow Newton's classical equations of motion, just as they would do for a classical oscillator, and therefore, the quantum bath can be approximated by a classical one. While useful for our calculations, this assumption makes the thermal equilibrium distribution of the sites deviate from the expected Boltzmann distribution [16].

By looking at equation 2-4, the frequency-dependency of the coupling between site i and the thermal bath can be fully described by the spectral density defined as [20]:

$$J_i(\omega) = \frac{\hbar}{\pi} J_{CL,i}(\omega) = \frac{1}{2} \sum_{\zeta} \frac{c_{i\zeta}^2}{m_{\zeta}\omega_{\zeta}} \delta(\omega - \omega_{\zeta}) \quad (2-15)$$

Special attention must be paid to the relation between the definition used here for the spectral density J_i and the one in the Caldeira-Legett model $J_{CL,i}$ [17].

The time-dependant fluctuations $\Delta E_i(t)$ of the site energies described in equation 2-10 can be used to calculate the autocorrelation defined as [8]:

$$C_i(t) = \langle \Delta E_i(\tau)\Delta E_i(t + \tau) \rangle \quad (2-16)$$

Denoting the inverse temperature by $\beta = 1/(k_B T)$, the relation between $C_i(t)$ and $J_i(\omega)$ is given by [4] [18] [20]:

$$J_i(\omega) = \frac{2}{\pi\hbar} \tanh\left(\frac{\beta\hbar\omega}{2}\right) \int_0^\infty C_i(t) \cos(\omega t) dt \quad (2-17)$$

and its converse:

$$C_i(t) = \hbar \int_0^\infty J_i(\omega) \coth\left(\frac{\beta\hbar\omega}{2}\right) \cos(\omega t) d\omega \quad (2-18)$$

These relations result crucial when combining the results from MD with dissipative exciton dynamics [20]. Applying the aforementioned high temperature limit, equation 2-17 becomes:

$$J_i(\omega) = \frac{\beta\omega}{\pi} \int_0^\infty C_i(t) \cos(\omega t) dt \quad (2-19)$$

Compared to equation 2-17, this expressions results in temperature-independent spectral densities [20].

The discrete autocorrelation of the site energies can be determined from a time series of these fluctuations computed at times t_l as [20]:

$$C_i(t_l) = \frac{1}{N-l} \sum_{k=1}^{N-l} \Delta E_i(t_l + t_k) \Delta E_i(t_k) \quad (2-20)$$

Where N is number of sample points taken and $l \in [0, N - 1]$

2.3 Fourier transform

In this section one of the most useful and versatile mathematical artifacts is introduced: **The Fourier Transform**

While simple periodic signals can be fully described by their frequency, amplitude and phase, often times the functions describing physical phenomena require a more elaborated approach. The Fourier transform allows us to analyze these complex functions by decomposing them into sums sinusoidal waves. Among the advantages this method provides, we are interested in its applicability to continuous, discrete, periodic and aperiodic functions.

2.3.1 Conventions

When working with Fourier transformations, it is imperative to verify the conventions used in the derivation of the formulas, specifically on the normalization employed. In the entirety of this work, the following definition of the Fourier transform is used [21]:

$$\mathcal{F}[x(t)] := X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-i\omega t} dt \quad (2-21)$$

And for its counter-part, the Inverse Fourier Transform:

$$\mathcal{F}^{-1}[X(\omega)] := x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega) e^{i\omega t} d\omega \quad (2-22)$$

2.3.2 Discrete Transform

When computing or working with real data, the focus of interest shifts from the conventional continuous functions, to their computationally-accessible discrete counterpart. Such idea is at the heart of the difference between analog and digital signal processing. Here, the term “signal”, coming from signal processing literature, will be used interchangeably with the concept of “function”.

While analytical expressions can be obtained to confirm our results, it is through discrete methods that our computers calculate, and later model, the natural phenomena described by our equations. Besides discreteness, one must consider the duration of the signal, where these can be of finite or infinite length. Several combinations can be attained depending on the nature of the data available, but on the following we will focused on the type of signals that are discrete in time and frequency domain. This will assure that they are also finite in both domains [21]. Motivated by this, I now introduce you to the discrete-version of the Fourier transform: **Discrete Fourier Transform (DFT)**.

For a discrete sequence of N evenly spaced samples of a time domain signal, the DFT is defined as [22]:

$$X[k] = \sum_{n=0}^{N-1} x[n] \exp\left(-\frac{i2\pi kn}{N}\right) \quad (2-23)$$

Where Δt represents the time step, $x[n]$ the current sample for $t = n\Delta t$, and $X[k]$ the DFT for frequency $\omega_k = k\Delta\omega$ for $k \in [0, N - 1]$ and $\Delta\omega = \frac{2\pi}{N\Delta t}$.

The inverse function is then defined as [22]:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] \exp\left(\frac{i2\pi kn}{N}\right) \quad (2-24)$$

$X[0]$ is the sum of the signal, while $X[1 : N/2]$ contains the positive-frequency terms and $X[N/2 + 1 : N - 1]$ contains the negative frequency terms in order of decreasingly negative frequency. If the input signal is real and even, its DFT is also real and even [21]. For these cases, often times only the positive frequencies (first half) of the data are taken into account when plotting and performing calculations.

Additionally, the Nyquist-Shannon sampling Theorem, or Nyquist criteria [22], defines the maximum frequency a sample signal can contain in order for it to be reconstructed. This is the so-called Nyquist Frequency and is equal to half the sampling frequency:

$$\omega_{ny} = \frac{N}{2} \Delta\omega = 2\pi \frac{1}{2\Delta t} = \frac{\pi}{\Delta t} \quad (2-25)$$

2.3.3 Continuous-Discrete Relation

A periodic signal $x(t)$ with a period T can be expressed as a Fourier series with the k^{th} coefficient defined by [23]:

$$X(\omega_k) := \frac{1}{T} \int_0^T x(t) e^{-i\omega_k t} dt, \quad k = 0, \pm 1, \pm 2, \dots \quad (2-26)$$

Now, sampling the continuous signal through a continuous-time impulse train we obtain:

$$x_s(t) := x(t) \Delta t \sum_{-\infty}^{\infty} \delta(t - n\Delta t) \quad (2-27)$$

with $\delta(t)$ the Dirac-delta function. Then, substituting this in equation 2-26 and choosing the sampling time Δt such that $N = T/\Delta t$ [23]:

$$\begin{aligned} X_s(\omega_k) &= \frac{1}{T} \int_0^T x_s(t) e^{-i\omega_k t} dt \\ &= \frac{\Delta t}{T} \sum_{n=0}^{N-1} x(n\Delta t) e^{-i\omega_k n\Delta t} \\ &= \frac{1}{N} \text{DFT}_{N,k} x(n\Delta t) \end{aligned} \quad (2-28)$$

From where the final relation between the DFT output signal and sampled continuous function is obtained as:

$$X_s(\omega_k) := \frac{1}{N} X_P \quad (2-29)$$

2.3.4 Fast Fourier Transform

While the DFT is remarkably useful to perform numerical approximations, it becomes computationally expensive with the increase of data points. From this issue, an efficient algorithm that exploits the symmetry properties of the signal to calculate the DFT was introduced by Cooley and Tukey in their 1965 paper [22][24]. With this, the time complexity of the implementations decreased from $\mathcal{O}(n^2)$ for the DFT, to $\mathcal{O}(n \log(n))$ [25]. This algorithm was coined as **Fast Fourier Transform (FFT)**.

2.4 Power Spectral Density

The enveloping algorithm uses the power spectral density $S(\omega)$, to ensure the correct frequency-domain behaviour of the energy fluctuations generated. Therefore, an analytical expression for this should be readily available. However, when working with molecular systems such as LH complex, it is rather the spectral density $J(\omega)$ the function used to describe the frequency dependence of the bath-system interactions. Due to this discrepancy, it is necessary to obtain an explicit relation between these two for its posterior implementation.

The autocorrelation $C(t)$ of an assumed ergodic signal $x(t)$ and its power spectral density $S(\omega)$ are related by the Einstein-Wiener-Khinchin theorem [26]. This theorem states that $C(t)$ is the Fourier transform pair of $S(\omega)$ as given by:

$$S(\omega) = \int_{-\infty}^{\infty} C(t)e^{-i\omega t} dt \quad (2-30)$$

In the current implementation, this relation will remain valid when the variance κ^2 of the random samples drawn satisfies $\kappa^2 = 1$. Details are shown in the appendix.

From the definition of the autocorrelation function in equation 2-16, we see that $C(t) = C(-t)$. Notice that this expression might not have a direct meaning when working with discrete, positive time series. However, this implies that the autocorrelation is an even function, which in turn guarantees that $S(\omega)$ is real [26], and allows equation 2-30 to be rewritten as:

$$S(\omega) = 2 \int_0^{\infty} C(t) \cos(\omega t) dt$$

Using this expression, together with equation 2-19, yields the desired relation:

$$\implies S(\omega) = \frac{2\pi}{\beta\omega} J(\omega) \quad (2-31)$$

2.4.1 Theoretical Spectral Shapes

Analyzing atomic transitions, along with their associated energy differences and frequencies, is at the heart of spectroscopy. This is done through the study of their spectral line shape, also known as spectral density (SD). Here, the SD is used to describe the coupling between sites and the thermal bath, as well as the autocorrelation of the energy fluctuations.

Drude

To first test the algorithm against previously established benchmarks [27], energy fluctuations following a “simple” Drude spectral density were generated. Their autocorrelation is described by an exponentially decaying function for site i as in [17]:

$$C_i(t) = \frac{2\lambda_i}{\beta} e^{-\gamma_i t} \quad (2-32)$$

Under the harmonic bath and high temperature limit assumptions discussed previously, the autocorrelation is related to the spectral density as in 2-17. Here, this yields a Drude SD:

$$J_i(\omega) = \frac{2}{\pi} \lambda_i \gamma_i \frac{\omega}{\omega^2 + \gamma_i^2} \quad (2-33)$$

Where γ_i is the inverse of the correlation time τ and relates to the width of $J(\omega)$, while λ_i stands for the **reorganization energy** of the bath and can be calculated as [17]:

$$\lambda_i = \int_0^\infty \frac{J_i(\omega)}{\omega} d\omega \quad (2-34)$$

This last equation is a general relation valid for any spectral density function.

Its power spectral density follows from 2-31:

$$S_i(\omega) = \frac{4}{\beta} \lambda_i \gamma_i \frac{1}{\omega^2 + \gamma_i^2} \quad (2-35)$$

For the posterior analysis, it is useful to calculate an expression for the maximum value and associated frequency of equation 2-33. This is given by the first derivative method as:

$$\begin{aligned} \omega_{max} &= \gamma_i \\ J_i(\omega_{max} = \gamma_i) &= \frac{\lambda_i}{\pi} \end{aligned} \quad (2-36)$$

Lorentzian

Nonetheless, experimental data and theoretical calculations demonstrate that in real physical systems, the spectral density is not generally described by merely smooth shapes as it is assumed when implementing Drude forms [20].

Experimentally calculated spectral densities can be modeled using a combination of Drude and Lorentzian functions as done by S. Maity et al. in [28]. The former, describes the low frequency modes due to coupling with the environment, implemented as an over-damped Brownian oscillator. The latter accounts for the high-frequency modes through under-damped Brownian oscillators, and its SD is given by [29]:

$$J_i(\omega) = \frac{2\hbar}{\pi} \sum_k s_k \omega_k^3 \frac{\omega \gamma_k}{(\omega_k^2 - \omega^2)^2 + \omega^2 \gamma_k^2} \quad (2-37)$$

Where s_k are the Huang-Rhys (HR) factors and ω_k the corresponding frequencies. The broadening factor (to some extended arbitrary [29]) γ_k controls the width of the individual peaks, and therefore, defines their overlap. The power spectral density is correspondingly:

$$S_i(\omega) = \frac{4\hbar}{\beta} \sum_k s_k \omega_k^3 \frac{\gamma_k}{(\omega_k^2 - \omega^2)^2 + \omega^2 \gamma_k^2} \quad (2-38)$$

For the sake of completeness, it must be mentioned that the autocorrelation of a Lorentzian follows an exponentially decaying trend, but an analytical transformation is not easily attainable.

Similarly as for the Drude model, the frequency corresponding to the maximum value for a single Lorentzian peak is calculated using the first derivative method. Here, however, the solution comes from a quartic equation and is given by:

$$\omega_{max} = \sqrt{\frac{-\sqrt{\gamma_k^4 - 4\omega_k^4\gamma_k^2} + 16\omega_k^2 - \gamma_k^2 + 2\omega_k^2}{6}} \quad (2-39)$$

Gaussian

Alternatively, real spectral densities can be approximated by Gaussian and Lorentzian functions representing the low and high frequency components of the spectrum, respectively [30].

For noise following a Gaussian SD, the autocorrelation expression is given by [31]. Under slight modifications, justified by our Fourier transform normalization, the autocorrelation is:

$$C_i(t) = \frac{\hbar\gamma_i}{\beta\sqrt{\pi}} e^{-(\gamma t)^2} \quad (2-40)$$

With spectral density:

$$J_i(\omega) = \frac{\hbar\omega}{2\pi} e^{-(\omega/2\gamma_i)^2} \quad (2-41)$$

And power spectral density described by another Gaussian function, as expected from the Fourier transform of a Gaussian:

$$S_i(\omega) = \frac{\hbar}{\beta} e^{-(\omega/2\gamma_i)^2} \quad (2-42)$$

Here, again, $\gamma = 1/\tau$ corresponds to the width of the peak and inverse of correlation time.

The \hbar was added to each function to obtain the right units. For a more detailed explanation see “Units” section.

3 Implementation

3.1 Algorithm

Avoiding the explicit calculation of the bath trajectories when computing the time-dependant Schrödinger equation allows for faster simulations. To this end, we want to generate energy fluctuations that follow an arbitrary autocorrelation and that are related to the spectral density of the thermal bath as in 2-19. This would not only decrease the computational expense, but also enable the study of more general interactions, as compared to the restricted simulations done before [13][17][27].

In the context of energy-site fluctuations generation, we will refer to these fluctuations as **Noise**, denoted by $\eta(t)$. Particularly, Noise following a desired autocorrelation is known as **Structured** or **Colored Noise**. Our goal is to generate a time series of this noise for N equally-spaced time steps with a specific autocorrelation $C(t) = \langle \eta(\tau)\eta(t + \tau) \rangle$ computed as in 2-20.

Different methods to create trajectories following specific models have been developed previously. For instance, [27] describes an algorithm that implements a Gaussian random number generator with exponentially correlated numbers, thus achieving a Drude spectral density.

Nevertheless, as discussed previously, if we aim to model more complex physical processes, noise with a broader domain of autocorrelations should be attained. Herein I implement an enveloping algorithm proposed in [31] to obtain colored noise with **any** desired spectral density. The general algorithm is as follows:

Algorithm 1 Enveloping algorithm: Structured Noise

- 1: Generate white Gaussian noise $W(t_l)$ by sampling N times from i.i.d Gaussian variables, each with mean $\mu = 0$ and variance $= \kappa^2$
- 2: Calculate the Fourier transform of the sequence $W(t_l)$: $\mathcal{F}[W](\omega)$
- 3: Envelope, i.e. multiply the frequency spectrum by the square root of the desired power spectral density: $\sqrt{S(\omega)}\mathcal{F}[W](\omega)$
- 4: Calculate the inverse Fourier transform to obtain the noise trajectory:

$$\eta(t_l) = \mathcal{F}^{-1} \left[\sqrt{S(\omega)}\mathcal{F}[W](\omega) \right] (t_l)$$

Ensure: $S(\omega) = S(-\omega)$

See appendix for a proof of correctness of this algorithm.

3.2 Units

Surprisingly enough, one of the first challenges to overcome at the moment of implementing the different computational algorithms and methods for quantum dynamics simulations, is the correct use and consistency of the units.

Contrary to what an undergraduate physics student is exposed to, *Joules* (J) is often times not the default unit to express energy values across the different specialization areas in physics. Spectroscopists predilection is the inverse cm (cm^{-1}), motivated by the relation of the wavenumber and the energy of the associated photon. On the other hand, semiconductor experts would rather express the energy values in electronvolts (eV). Similarly, several other units can be used, e.g. fs and THz. Due to this, one ought to verify the units employed, based on the context, before any calculation.

Above all, one of the most frequent calculations is the conversion from eV to cm^{-1} , and viceversa. Using the equation that relates wavelength and energy of a photon: $E = hc/\lambda$, with E in eV, h in eV·fs, c in ms^{-1} and λ in cm^{-1} , one obtains the following relation:

$$[\text{eV}] = 1.239 \times 10^{-4} [\text{cm}^{-1}] \quad (3-43)$$

To keep the units consistent across the whole implementation, the following list was used as guidance when working with the different parameters:

- The time step for the generation of energy trajectories Δt is in the order of fs to mimic the behavior of molecular systems. Consequently, the angular frequencies employed in the noise generation and spectral density calculations have units of rad/fs and the inverse of the correlation time γ has units of 1/fs.
- One should recall that radians are **dimensionless** by definition, so expressions like $\omega^2 + \gamma^2$ can be written both with units of $1/(\text{fs})^2$ or $\text{rad}^2/(\text{fs})^2$.
- Since the autocorrelation functions have the same units as ΔE^2 , they are expressed in $(\text{eV})^2$. As a result, the spectral densities and the power spectral densities have units of eV and $(\text{eV})^2 \cdot \text{fs}$, respectively.
- To achieve the right units, the reorganization energies must be expressed in eV, the Boltzmann Constant k_B in $\text{eV} \cdot \text{K}^{-1}$, the temperature in K, and the (reduced) Planck constant in $\text{eV} \cdot \text{fs}$.
- Only when mentioned explicitly, the energy values will be changed to cm^{-1} by changing the units of the noise trajectories and the Boltzmann constant to cm^{-1} , following the calculation derived above.

The following online-tool developed by the Nanophotonics group at Rice University can be used to make direct conversions between the different energy units [32].

See appendix for a table with the numerical values for the different constants used.

3.3 Noise Generation

The noise generation algorithm, as well as the autocorrelation, spectral density and population dynamics calculations were implemented in Python 3.9 [33] using the NumPy [34], SciPy [35], mpmath [36] and PyTorch [37] libraries. The NumPy library provides the functions necessary to work with multi-dimensional arrays, as well as the random number generation and the calculation of the Fourier transformations. SciPy contains several modules used widely in scientific and technical computing, among which we implemented the 1-D interpolation method. Mpmath is a python library used for the real and complex floating-point arithmetic with arbitrary precision. Its functions for numerical integration, such as quadrature, were implemented along with the algorithm suggested when integrating highly variable functions. Finally, PyTorch’s library is build for machine learning algorithms. This was implicitly implemented in the NISE algorithm provided by Yannick Holtkamp in [8].

The noise algorithm is implemented in a “batch-manner”. This means that the generated data was stored in a 3-dimensional tensor where the second dimension corresponds to the time component of the noise trajectory, the first dimension contains the different realizations of the noise, and the third dimension contains the data for different sites. The reason for this is twofold. First, this approach allows for the use of more computationally-efficient calculations, due to the decrease in memory access operations needed. And second, this was done as an alternative to other computational methods such as “windowing” [20] for the generation of uncorrelated time samples of the noise trajectory.

Using the theory developed in the previous sections, and after defining the correspondent numerical constants and parameters for a specific shape of the power spectral density $S(\omega)$, the noise generation algorithm is implemented as:

Algorithm 2 Noise Generation

Input

Time Length: T
Realizations: $Reals$
Number of Sites: $Sites$
Time step: dt
Parameters for $S(\omega)$: S_k

Output

3D tensor containing the Noise trajectories η

Before noise generation:

- 1: Define a Power spectral density function that takes as input S_k and frequencies ω : $S_{\omega,k}$
 - 2: Define $N = \text{int}(T/dt + 1)$
 - 3: Create an empty tensor η with dimensions ($Reals \times N \times Sites$)
-

Noise generation:

- 1: Draw random samples from a normal Gaussian distribution with mean $\mu = 0$ and variance $\kappa^2 = 1$. Size of White-Gaussian noise tensor W : $((Reals * Sites) \times N)$
- 2: Calculate the DFT of matrix W . Include the correcting factor discussed in 2-29:

$$F_W = \frac{\mathcal{F}[W]}{\sqrt{dt}}$$

- 3: Generate the frequencies associated to the DFT of W : ω in rad/fs
- 4: Input ω and S_k in $S_{\omega,k}$ and Envelope the frequency spectrum by the square root of $S(\omega)$: $\sqrt{S_{\omega,k}}F_W$
- 5: Calculate the inverse Fourier transform and take the Real part to obtain the noise tensor:

$$\eta_0 = \Re \left\{ \mathcal{F}^{-1} \left[\sqrt{S_{\omega,k}} F_W \right] \right\}$$

- 6: **for** i in $[0, Sites-1]$ **do**
 - 7: Use the data of the 2D tensor with size $Reals \times N$ coming from the $(i * Reals)^{th} \rightarrow ((i + 1) * Reals)^{th}$ rows of η_0 to fill the i^{th} site in η
 - 8: **end for**
-

The DFT is implemented by calling the NumPy function `fft.fft()`. As explained in the relevant documentation [34], this function implements the Fast Fourier Transform algorithm, along with the normalization conventions explained in the theory section. The calculations of the frequencies are done in a similar fashion by using the NumPy function `fft.fftfreq()` that takes as arguments the length N of the time series and the sampling time step dt to produce the right frequencies, as defined in equation 2-23. For the implementation, these frequencies are multiplied by a factor of 2π to obtain the the correct units.

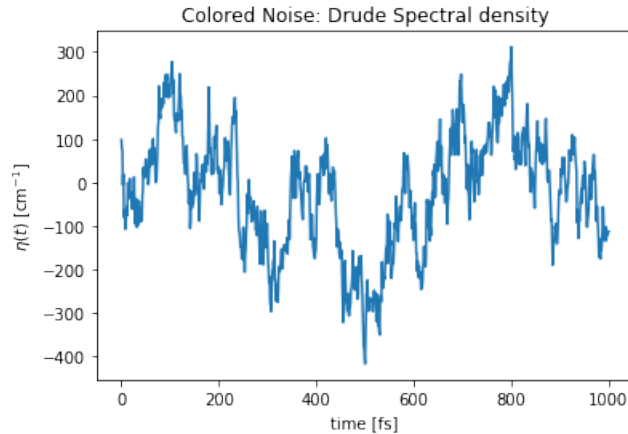


Figure 1: Single realization of the colored noise trajectory following a Drude spectral density.

The i^{th} component of the third dimension from the resulting matrix η , with dimensions

as explained above, is added to the diagonal elements of the i^{th} site Hamiltonian, for the posterior calculations of population dynamics using NISE. A more detailed explanation can be found on the “Population Dynamics” section below. This algorithm is general in essence, since all one needs to do to generate noise following the different spectral densities is to adjust the input parameters and redefine the power spectral density function $S_{\omega,k}$ accordingly. One realization of the noise trajectory generated for a single site is shown in figure 1.

3.4 Autocorrelation

The essential component of the whole algorithm, i.e. the generation of the noise, has already been explored. Fundamentally, this fluctuations can be already implemented in the NISE algorithm to calculate the population dynamics of a multi-site system. However, it is highly non-trivial to discern whether the noise generated follows the desired autocorrelation and spectral density. For this reason, I will now introduce an algorithm that employs equation 2-20 to calculate the $C(t)$ for a 2-dimensional matrix containing the noise trajectories of a single site.

Algorithm 3 Autocorrelation - Calculation

Input

2D Noise Matrix of dimensions (*reals* \times N): η_i

Output

1D Autocorrelation Array: C

Before Calculation:

Read *Reals* from length of η_i first dimension

Read N from length of η_i second dimension

- 1: **for** j in *Reals* **do**
 - 2: **for** l in N **do**
 - 3: **for** k in $N - l$ **do**
 - 4: Multiply the $(j, l + k)$ and the (j, k) elements of η_i and
 - 5: Add the result to the accumulated sum of the previous multiplications
 - 6: **end for**
 - 7: Divide the total Sum by $(N - l)$
 - 8: Assign it to the (j, l) element of C
 - 9: **end for**
 - 10: **end for**
 - 11: Average over all the different realizations (average over dimension 1)
-

Despite how cumbersome the algorithm might seem, it is nothing else but the implementation of equation 2-20 for different realizations of the noise trajectories across N time steps. For the actual implementation, the *for* loops were exchanged by vectorized methods and operations, such as “slicing”, since vector calculations are optimized for Python due to its architecture. Such implementation increased the computational speed of the algorithm significantly.

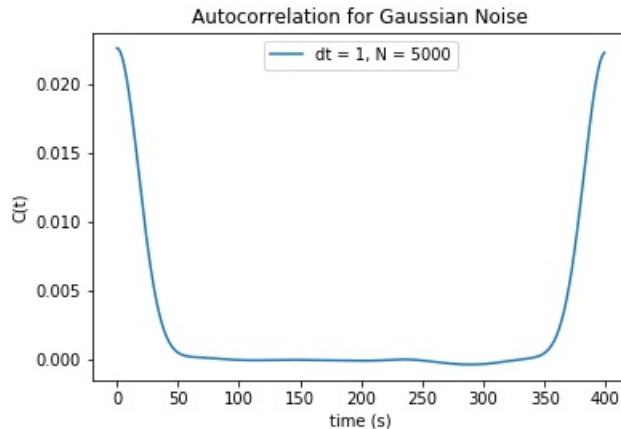


Figure 2: Complete autocorrelation $C(t)$ for noise following a Gaussian spectral density.

From the symmetry of the autocorrelation calculated, as depicted in figure 2, one can deduce that there are some additional considerations to keep in mind when performing posterior operations using the noise and the autocorrelation. This is not an unexpected result, but rather, a property predicted by the symmetries of the DFT method implemented. As explained in the theory section, the DFT of a real and even signal, is another real and **even** frequency domain signal, where the positive frequencies correspond to the first half of the array and the second half corresponds to the negative frequencies. This relatively simple fact explains our observations and justifies our next step: the use of exclusively **half** of the points generated for the noise trajectories and autocorrelations.

3.5 Spectral Density

Once the autocorrelation has been calculated and only half of the time steps are kept, the next step to follow is the study of the agreement between the theoretical spectral density and the one calculated from the colored noise generated. For this calculations, we are once again drawing upon the high temperature limit assumptions, aware of the drawbacks this carries. Such assumptions allows us to relate the already calculated autocorrelation $C(t)$ with the desired spectral density $J(\omega)$, implemented as a cosine transform times a thermal factor, according to equation 2-19. The following algorithm makes use of this relation.

Algorithm 4 Spectral Density - Calculation

Input

- 1D Autocorrelation array: C
- time step of noise trajectory used to obtain C : dt
- Accuracy degree given by integration intervals: $Inter$

Output

- 1D Spectral Density Array: J
- 1D array containing the energy values $X = \hbar\omega$ associated to the J array.

- 1: Read N_{cut} from length of C . Note that here $N_{cut} = N/2$, as argued previously.
- 2: Define fundamental frequency $d\omega$ as

$$d\omega = \frac{2\pi}{N_{cut}dt}$$

- 3: Calculate array ω containing the frequencies $\omega_n = nd\omega$. Bearing in mind the Nyquist frequency condition, the maximum allowed frequency is:

$$\omega_{max} = d\omega \frac{N_{cut}}{2} = \frac{\pi}{dt}$$

Such that $n \in [0, N_{cut}/2]$

- 4: Define the time array to use for interpolation. These should be the times corresponding to the autocorrelation points, hence: $t = kdt$, $k \in [0, N_{cut}]$
 - 5: Define a function from the interpolation of t and $C(t)$.
 - 6: Use the function obtained by interpolation, along with the whole array t and a given frequency ω_n to implement equation 2-19 with the help of a numerical integration scheme. Use $Inter$ to control the degree of accuracy of this integration, based on the oscillatory nature of the data.
 - 7: Repeat this last step for each of the frequencies ω_n in array ω to obtain $J(\omega)_n$
 - 8: Calculate array $X = \hbar\omega$ by direct multiplication.
-

Notice that while the maximum frequency in the array ω is relatively arbitrary, it still ought to follow the Nyquist condition. In fact, the Nyquist frequency $\omega_{max} = \frac{\pi}{dt}$ is the same we would have obtained had we performed the calculations with N points instead of N_{cut} . This is as expected since the Nyquist frequency depends only on the sampling rate $1/dt$.

In this implementation the interpolation was done with the help of the SciPy class `interpolate.interpld()` [35], using a *cubic* spline interpolation. This function was employed along with the `mpmath.quad()` integration scheme from `mpmath` to perform the integration relating $C(t)$ and $J(\omega)$ with the correspondent multiplication of the thermal factor $\beta\omega/\pi$. Following the standards for similar molecular dynamic simulations, our calculations assumed room temperature conditions, i.e. $T = 300$ K [17][20].

Moreover, a crucial contribution came from the integration method implemented for

highly variable functions as discussed in the `mpmath` documentation [36]. Due to the length and frequencies involved in our cosine transform, standard numerical integration schemes failed to give feasible results, often diverging from the theoretical spectral density. Alternative approaches included the built-in `fftpack.dct()` function from `scipy`, Simpson’s rule for numerical integration, and DFT based on symmetries. Details are found in appendix. However, it was the `mpmath` scheme that attained the best results, at a reasonable computational cost. The idea and results obtained are discussed in the results section.

Figure 3 shows the comparison between the theoretical and the calculated spectral density for a Drude simulation.

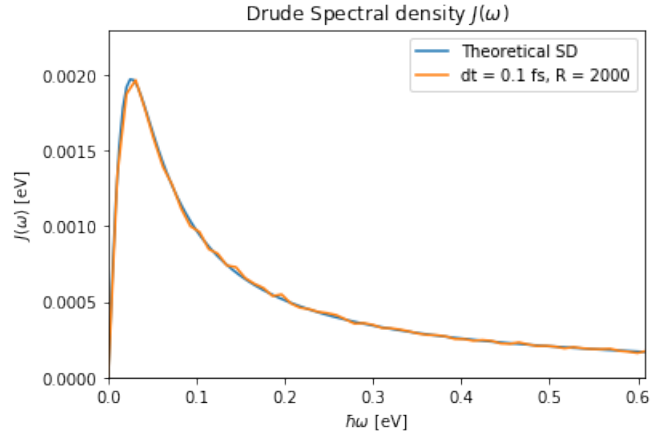


Figure 3: Spectral density for colored noise following a Drude shape. $dt = 0.1$ fs, Realizations = 2000, $T = 400$ fs.

3.6 Population Dynamics

After verifying the validity of the generated noise by comparing its autocorrelation and spectral density with their theoretical counterparts, the new aim was to use this fluctuations in the implementation of the NISE algorithm in order to calculate the population dynamics of a multi-site system.

As discussed in the theory section, the coupling of the thermal bath acting on the system can be understood through the fluctuations of site energies in the effective Hamiltonian, equation 2-10. Recalling that these fluctuations act exclusively on the site energies, we can simply add the noise tensor to the main diagonal terms of the system Hamiltonian through tensor operations to obtain the time dependent Hamiltonian. From there on, the implementation of the NISE is executed as described by Y. Holtkamp in [8]. This yields the population dynamics for the different sites.

Without the implementation of thermal corrections, in the long-time limit the system will approach an even distribution of the populations, contrary to the theoretically-correct Boltzmann distribution. The figure below depicts a population dynamics for a 4-site system

using a Hamiltonian with values within a reasonable range to resemble a real LH complex. The values were obtained from [38] and can be found in the appendix. The simulations were performed under noise following a Drude SD.

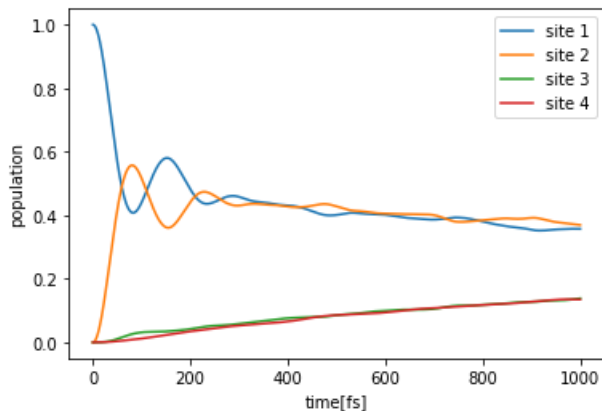


Figure 4: Population Dynamics for a 4-site system under the influence of bath fluctuations following a Drude Spectral Density

For the usage of the noise in exciton dynamics, the units of the resulting fluctuations must be adapted to show the expected results. This is done by converting from eV to cm^{-1} as explained in the Units section.

4 Results

4.1 Previous Benchmarks

Generation of structured noise has been possible before by means of different mathematical and computational mechanisms [17][27]. And while these have been tailored to generate fluctuations with rather simple spectral densities, such ad-hoc methods have settled the foundations for the implementation of more general algorithms, like the one herein presented. In the first testing-stage the enveloping algorithm underwent, the previously established benchmarks were replicated under different set of parameters.

Gaussian Generation

From here on, the number of realizations over which the energy fluctuations were averaged will be denoted as R and the total time of each trajectory will be denoted by T in fs, not to be confused with the temperature T in K.

A simple model, yet widely applicable, is the approximation of the spectral density by a Gaussian function, or rather by linear combinations of these. Here, a single Gaussian was simulated for a variety of parameters, including different time steps and total times. The results are shown in figure 5.

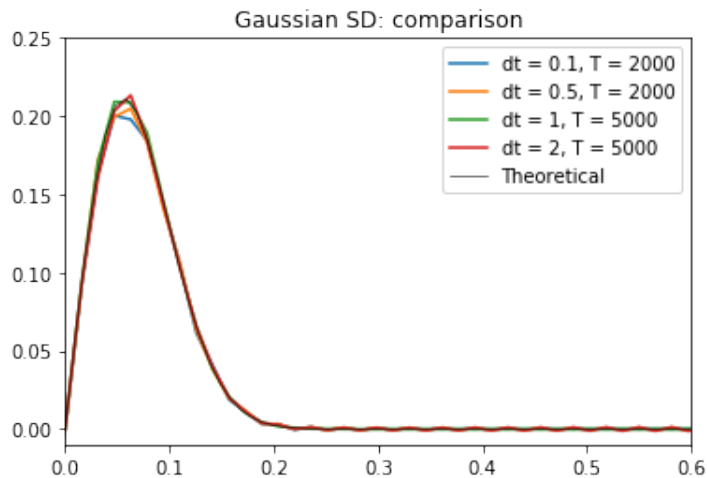


Figure 5: Numerical estimations for Gaussian spectral density generated for different set of parameters

Even though a quantitative measurement was not implemented, the accuracy of the calculated spectral density can be easily observed from the depicted comparison. The agreement was high for a big set of parameters, and for the simulation done using $dt = 0.1$, the calculated SD is indistinguishable from the theoretical function.

Drude Generation

Next, following the benchmark proposed by M. Aghtar et al. [17], a model suitable for both the density matrix formalism and wave packet dynamics approach was explored. Here, the frequency behaviour of the harmonic bath was approximated by a Drude spectral shape. Such approximation allowed to test the noise generation under conditions that resemble a real molecular system, e.g. light harvesting system.

The time-domain counterpart of a Drude function is an exponentially decaying autocorrelation as seen in 2-32. As a result, relatively small time steps were needed to obtain a high sampling rate that followed the fast decay of $C(t)$. Further, as shown in figure 6, the best results were attained when averaging over a considerable amount of realizations, as this averaged out the fluctuations coming from statistical noise.

Time steps in the order of 1 fs were implemented together with a temperature $T = 300$ K, reorganization energy $\lambda = 50 \text{ cm}^{-1}$ and autocorrelation time $\tau = 1/\gamma = 25$ fs for the comparison of the theoretical and generated spectral density. In order to minimize the benchmark proposed by M. Aghtar et al., where a 200 fs long trajectory was employed, our noise was generated using a length twice as long, to account for the symmetry in the autocorrelation coming from the DFT methods used.

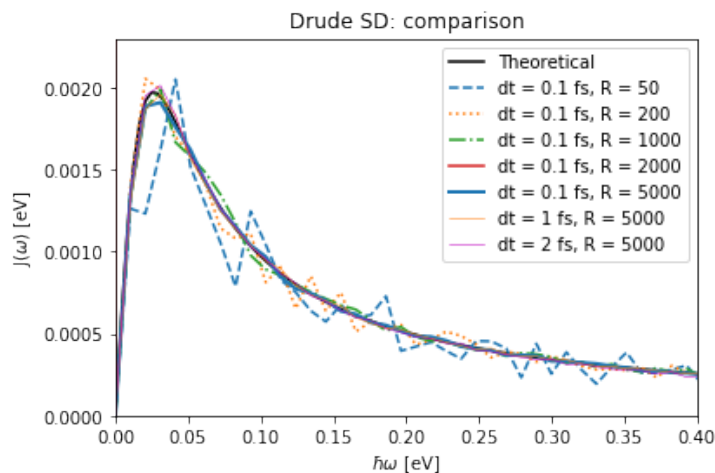


Figure 6: Numerical estimations for Drude spectral density, generated for different time steps and number of realizations

While for the calculations using Gaussian SD, a single integration interval was enough, the high frequency components inherent in the Drude model required the use of at least 20 intervals. Using equation 2-36, the maximum position of the calculated spectral densities was verified. As explained above and as depicted in figure 6, the best results for a fixed length and integration intervals were obtained for the smallest $dt = 0.1$ fs and higher number of realizations $R = 5000$, where again the calculated SD is indistinguishable from the analytical function.

4.2 New Benchmarks

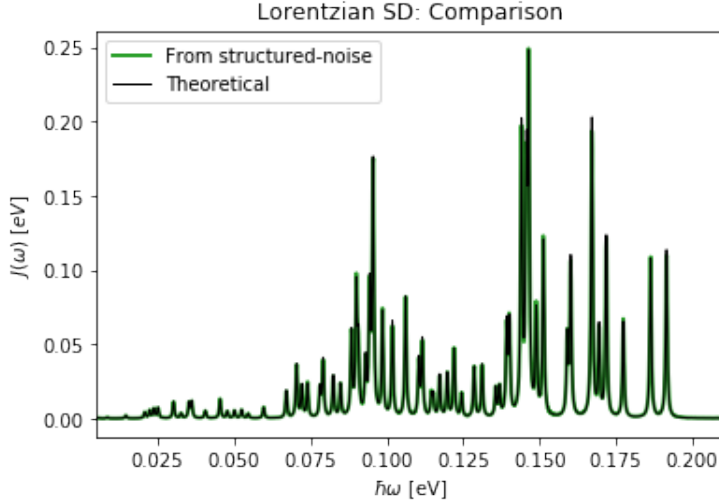


Figure 7: Comparison of Lorentzian spectral density calculated from generated Noise with the analytical function, showing an excellent agreement.

Once our algorithm was tested against previously established benchmarks, the focus shifted towards the simulation of more complex systems and parameter regimes. The high frequency component of the experimental spectral density for systems like the electron-phonon and vibronic couplings in a FMO complex, can be approximated by a Lorentzian spectral shape [29][30]. The techniques and intuition acquired through the initial tests were advantageous when implementing the pertinent changes in the source code.

Parameter-Gauging

Using the values given in [30] for the Huang-Rhys (HR) S_k factors, the respective frequencies ω_k and the broadening factor γ_k , we aimed to simulate the individual vibrational modes describing the intermolecular interactions of the system therein described. To this end, we started by gauging the parameters necessary to simulate, first, each of the peaks separately, in views of a posterior simulation of the 62 peaks all together.

To attain the optimal parameters necessary for the simulation, we proceeded to analyze the extreme cases: the lowest and the highest energy components. For the former, even small time steps yielded reasonable results, i.e. a correct peak location as verified through equation 2-39. However, fastly it became obvious that using lengths in the same order of magnitude as for Drude simulations was unsuitable if we aimed to obtain enough samples to describe the narrow peaks in our model. Such observation was supported by the relation between the sample points N , length T and time step dt : $N = T/dt + 1$, as discussed in the

implementation section. The increase in the length not only entailed higher computational times, but also resulted in a rapid increase of numerical noise that hindered the evaluation of the calculated SD. It was here when the the integration scheme was first updated to account for the highly variable functions we were dealing with. A dedicated subsection describing the method employed can be found below.

Even though the implementation of this tailored-method was inherently increasing the computational power needed even further, it allowed to virtually get rid of all the numerical noise, as long as the number of intervals was adjusted according to the complexity of the spectral density. From these computations we found that in order to reproduce the lowest energy peak, the time step could be as high as 25 fs, as given by the Nyquist criteria in equation 2-25, while the trajectory length required was as high as 30,000 fs. On the other hand, using the same techniques, we found out that to reproduce the highest peak at 1545 cm^{-1} , noise trajectories with at most 5 fs and length of at least 10,000 fs were required.

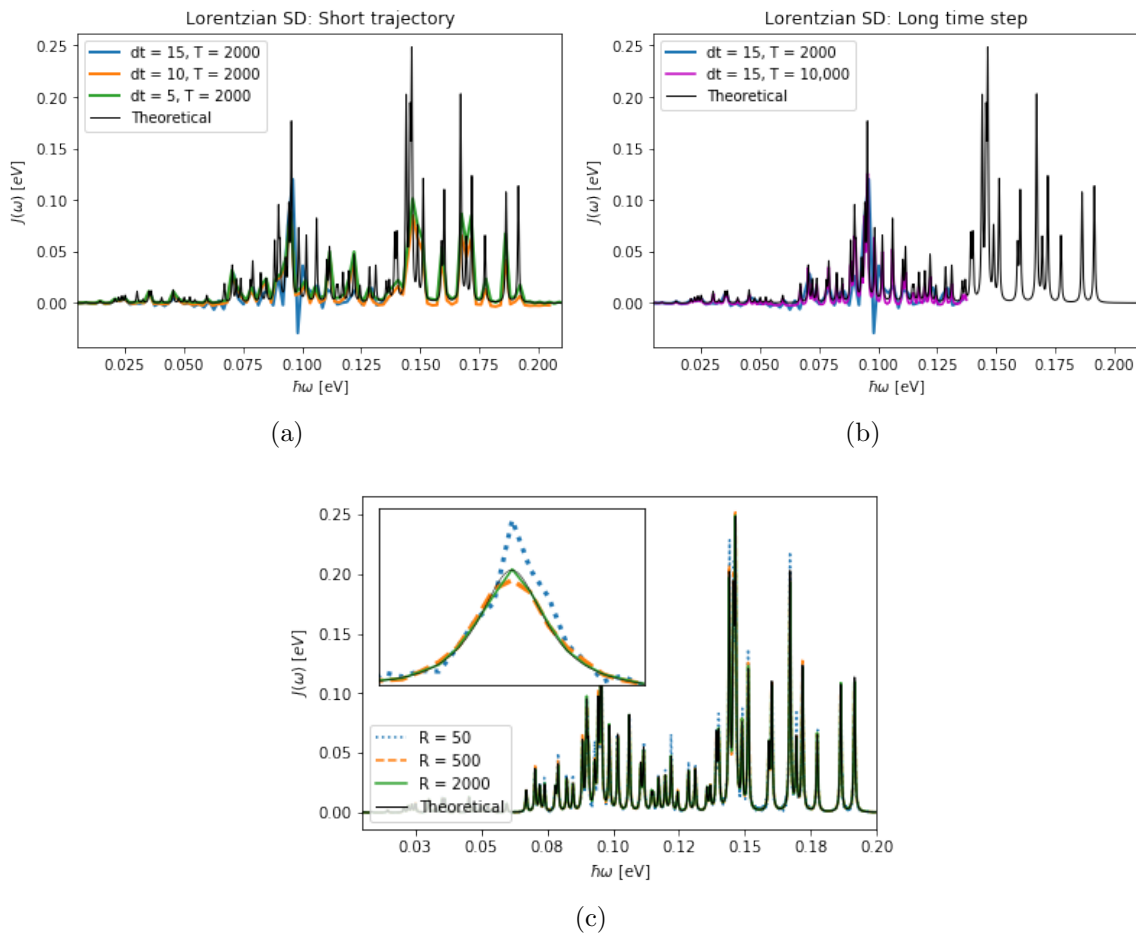


Figure 8: Dependence of Lorentzian spectral density on the time step, length of trajectory, and realizations implemented.

These observations were used as thresholds for the complete implementation of the 62 peaks as depicted in figure 8(a) and figure 8(b). First, short noise trajectories were implemented for different time steps, showing an overall right position of the peaks, but an inadequate height, due to the low amount of sampling points. Then, big time steps were employed, showing that even for relatively long trajectories, like 20,000 fs, the Nyquist criteria acted as an upper bound for the peaks we could generate. In addition, once the optimal parameters were established, these were employed with different number of realizations to find a compromise between computational cost and accuracy of the spectral density calculation, the results are shown in figure 8(c). To achieve a compromise between accuracy and computational speed, the number of realizations was set to 2000.

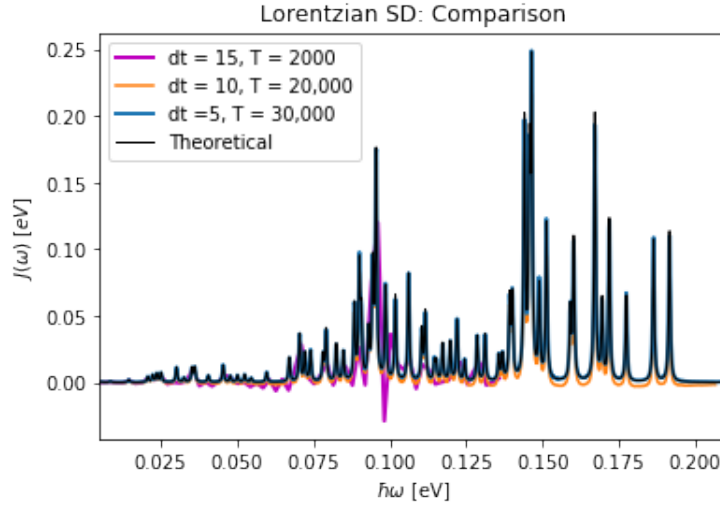


Figure 9: Numerical Calculation of Lorentzian spectral density for different set of parameters.

The discrepancies in the results for different set of parameters is better depicted in figure 9. From these observations and based on our theoretical development, the parameters yielding accurate results, while limiting the computational cost, were found as:

- time step $dt = 5$ fs
- Trajectory length $T = 30,000$ fs
- Realizations $R = 2,000$
- integration intervals = 30

These parameters were used in the generation of noise following the desired Lorentzian spectral density. Figure 10 compares this results with the theoretical function, where an *excellent* agreement is observed.

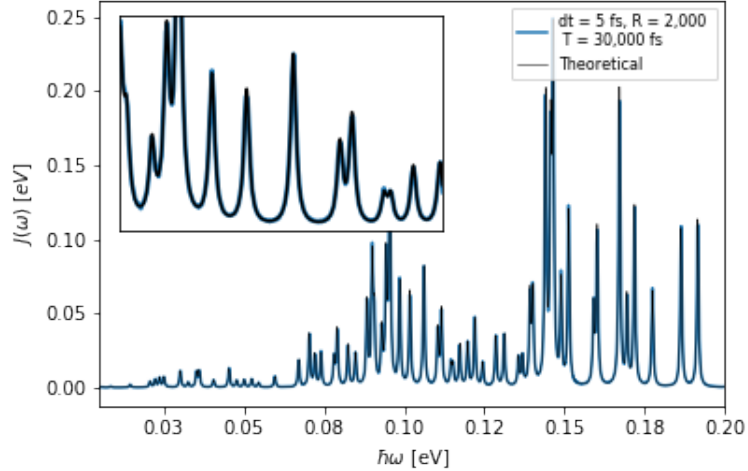


Figure 10: Lorentzian spectral density calculated for optimal parameters: $dt = 5$ fs, $T = 30,000$ fs and $R = 2000$

Highly Oscillating functions: Integration Scheme

The spectral density is calculated from the autocorrelation of the noise by effectuating a cosine transform, as dictated by equation 2-19. Various methods were devised, as explained in the implementation section, but it was the numerical integration scheme `quad()` from `mpmath` library the one employed in the final implementation, based on its performance. Due to the oscillatory nature of the functions being integrated, often times this transformation ended up diverging. As a result, the actual shape of the spectral density became indistinguishable from the numerical noise this divergence caused.

To account for this, it became imperative to implement a specialized scheme that managed highly oscillating functions. On that account, the methods in the documentation for numerical integration of `mpmath` were followed [36]. The simple, but effective, idea consists on breaking the integration into intervals where the oscillations of the integrand can be accounted by the standard routine given by `quad`. Increasing the number of intervals becomes necessary as the number of samples N gets larger, and when the complexity of the spectral density increases. As expected, the main drawback of this method lies on the increase in the computational expense with the number of intervals.

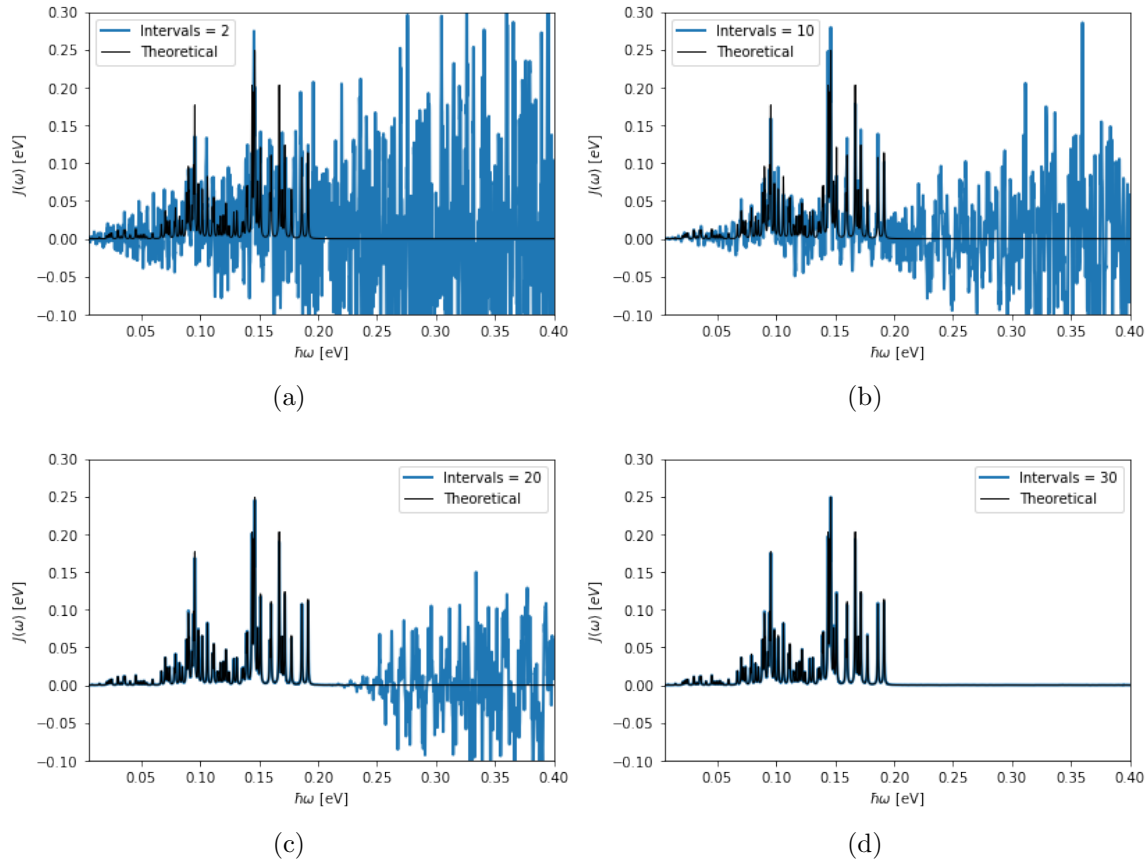


Figure 11: Lorentzian spectral density calculated using $dt = 5$ fs, $T = 30000$ fs, Reals = 2000. The results are shown for different number of integration intervals

Increasing the intervals gradually decreases the numerical noise up to the point where only the pure spectral density is left, as shown in figure 11. Here, the spectral density of 62 Lorentzian peaks was calculated from noise generated with $dt = 5$ fs, $T = 30,000$ fs, and averaging over 2000 realizations. Figure 11(a) shows the extreme case where only 2 intervals are used for the calculation, in which the integration explodes. This results is expected from the complexity of the aimed spectral density, since it consists of many narrow peaks with high amplitudes, that in the limit resemble an ensemble of Dirac delta functions. On the other end, the usage of 30 intervals yielded great results, where no undesired fluctuations appeared even for the highest frequencies, as depicted in figure 11(d). As a result, this number of intervals was used as the standard to obtain reliable results when effectuating posterior calculations.

4.3 Performance in population dynamics calculations

With the polish of the noise algorithm, and after studying its performance in the simulation of different models and parameter spaces, we went further and studied the behaviour of our noise when implemented in the calculation of population dynamics through NISE.

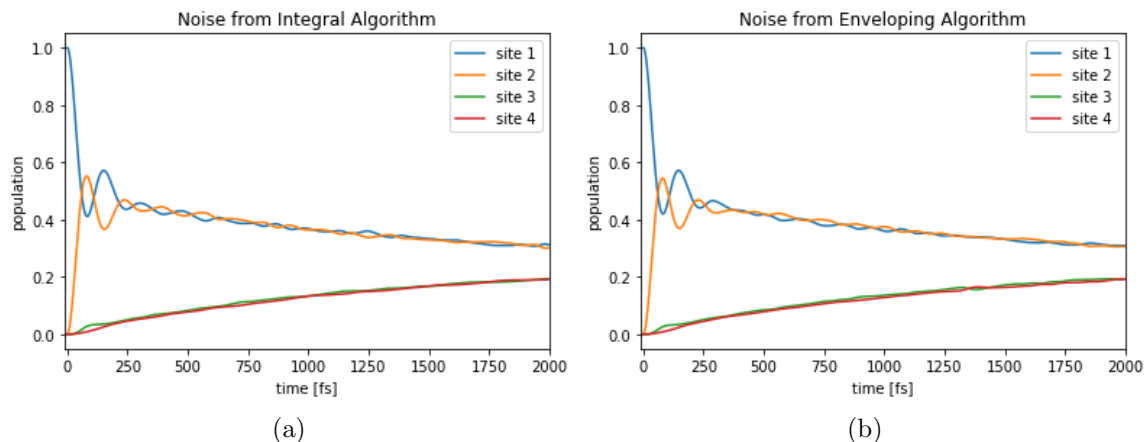


Figure 12: Population dynamics for thermal bath with Drude spectral density. Left: Noise generated by Integral algorithm. Right: Noise generated by enveloping algorithm

First, noise following a Drude SD was implemented to simulate the coupling between the environment and a 4-site system with different site energies. The values used in the Hamiltonian and noise generation can be found in the appendix. The resulting exciton dynamics from our algorithm is compared in figure 12 against the population of an integral algorithm that exclusively generates noise with exponentially decaying autocorrelation. This algorithm was first implemented in Fortran [27] and then adapted to Python by Y. Holtkamp [8]. The population densities were calculated as explained in equation 2-14, from the wavefunction obtained by the NISE algorithm [8]. From this comparison, we observe how both algorithms converge to an even distribution of the populations, as opposed to the appropriate thermal distribution for such a system given by Boltzmann’s equation. This is the result of the high temperature limit approximation.

The performance of the enveloping algorithm was further investigated by reproducing the results obtained in [17] where the population dynamics for a two-level system are calculated using the NISE and HEOM approaches. Once again, the thermal coupling between the bath and the system is modeled by a Drude spectral density due to its flexibility and applicability in both formalisms. While fixing $T = 300$ K, with equal coupling between the sites $V_{12} = V_{21} = V = 100$ cm^{-1} , equal site energies $E_1 = E_2 = E$, and correlation time of $\tau = 1/\gamma = 100$ fs, the reorganization energy λ was varied. Since only the difference between the site energies plays a role in the dynamic of the systems, these were arbitrarily set to 0 cm^{-1} [17].

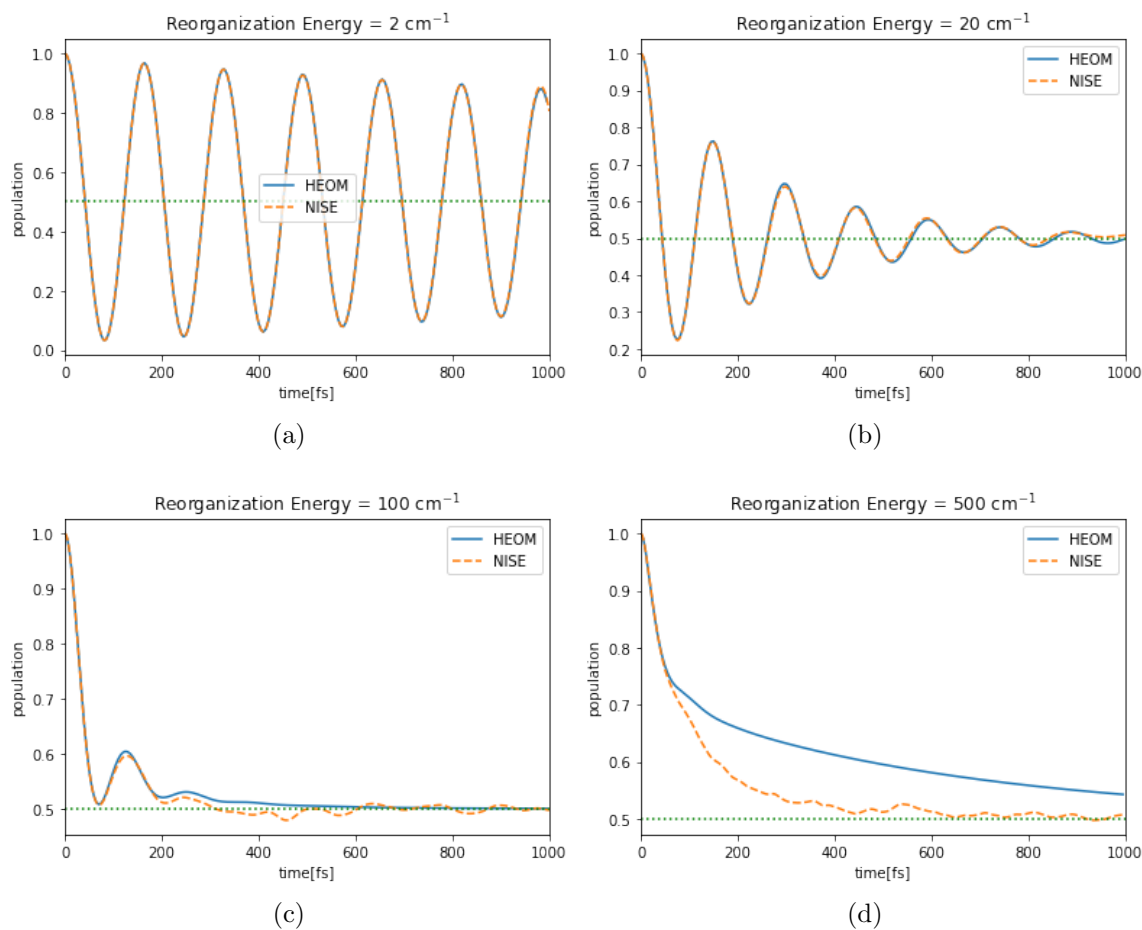


Figure 13: Comparison between HEOM and NISE when calculating the population dynamics of an excited state in a two site system with vanishing energy difference $\Delta E = 0$. Plotted for different reorganization energies λ

The population of the initially excited state is plotted against time in figure 13. From this, we notice that while for small reorganization energies NISE and HEOM produce similar results, the discrepancy for higher λ indicates that in that regime we can no longer ignore the coupling between the system and bath. Both approaches, however, achieve the evenly distributed population density predicted by the Boltzmann distribution for $\Delta E = 0$.

Additionally, a similar comparison for a system with $\Delta E = V$ is implemented as in [17]. The results of the dynamics shown in figure 14 demonstrate that for a system with different site energies, the HEOM calculations approach the theoretical distribution, while the high-temperature limit makes it impossible for NISE to diverge from the even population densities.

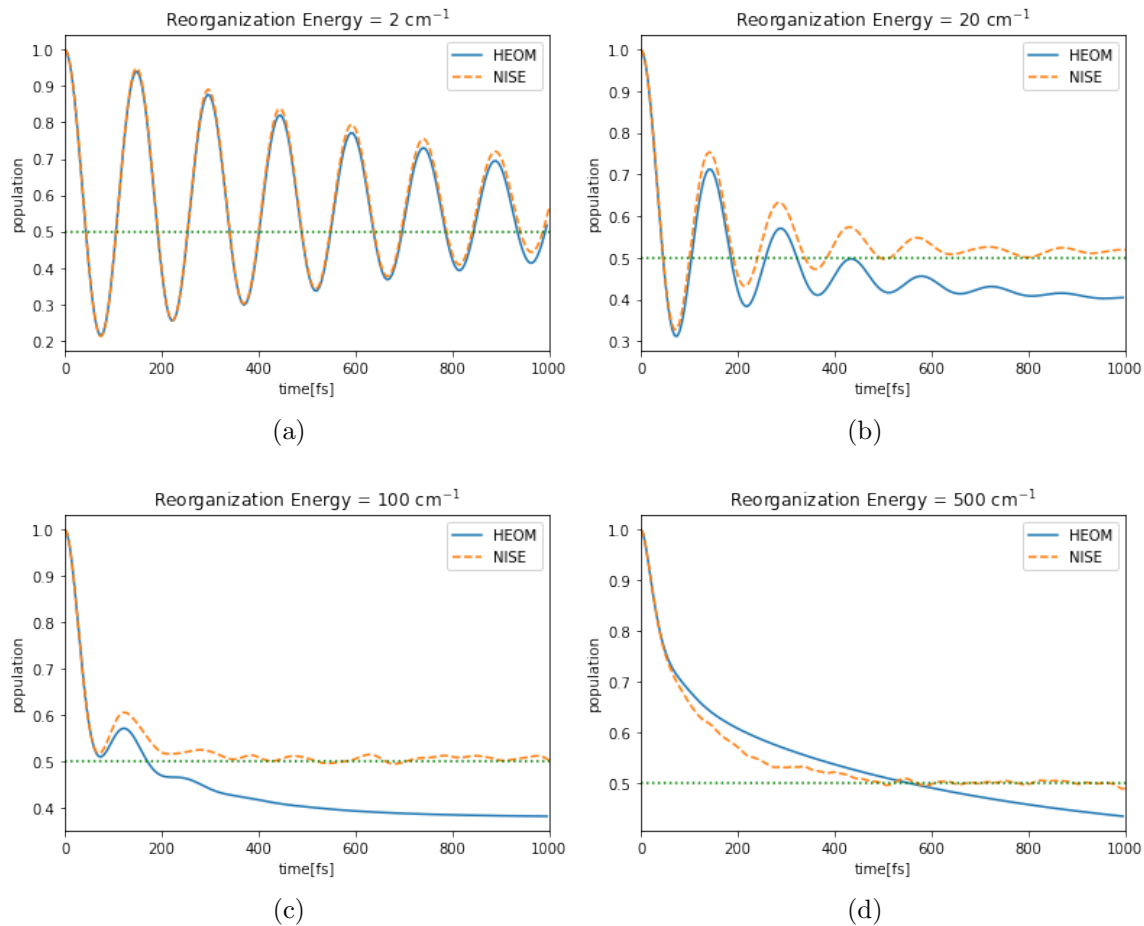


Figure 14: Comparison between HEOM and NISE when calculating the population dynamics of an excited state in a two site system with energy difference $\Delta E = V = 100 \text{ cm}^{-1}$. Plotted for different reorganization energies

The three set of simulations reliably reproduced the results previously established. Therefore, confirming the validity of the enveloping algorithm in the calculation of exciton dynamics, particularly for a bath following a Drude spectral density.

Reorganization energy dependence

Going a step further by implementing the structured noise with a Lorentzian SD in the calculation of exciton dynamics, provided additional information on the limitations and accuracy of the NISE approach and the noise generator itself.

Motivated by the discrepancies encountered between the results of NISE and HEOM when employing frequencies in the vicinity of the system's transition frequency, we studied the behaviour of the two methods when varying the reorganization energy. For this purpose, we used a system described by the same Hamiltonian employed in the calculations of figure 13. The transition frequency can be calculated from the diagonal entries of the system

Hamiltonian in the eigenbasis \tilde{H} :

$$\tilde{H} = \begin{bmatrix} \epsilon_1 & 0 \\ 0 & \epsilon_2 \end{bmatrix} \quad (4-44)$$

Here, we obtained $\hbar\omega_{12} = \epsilon_2 - \epsilon_1 = 200 \text{ cm}^{-1}$. Consequently, aiming to stay away from this frequency, while still obtaining a reasonable damping, we opted for a thermal bath described by a single Lorentzian peak with HR factor $S_k = 0.008$ and frequency $\omega_k = 284 \text{ cm}^{-1}$. Based on the tests performed for the simulation of Lorentzian spectral densities, the noise was generated using $dt = 1 \text{ fs}$, $T = 10,000 \text{ fs}$ and $R = 2,000$.

As shown in figure 15, and as explained by the proportionality between the reorganization energy and S_k , the long-time behaviour of the population dynamics in our test system differed from one method to the other. This discrepancy became more notorious with the increase of the reorganization energy (increase of S_k). For the case of $S_k = 0.012$, the results from the two approaches started to diverge even for small times.

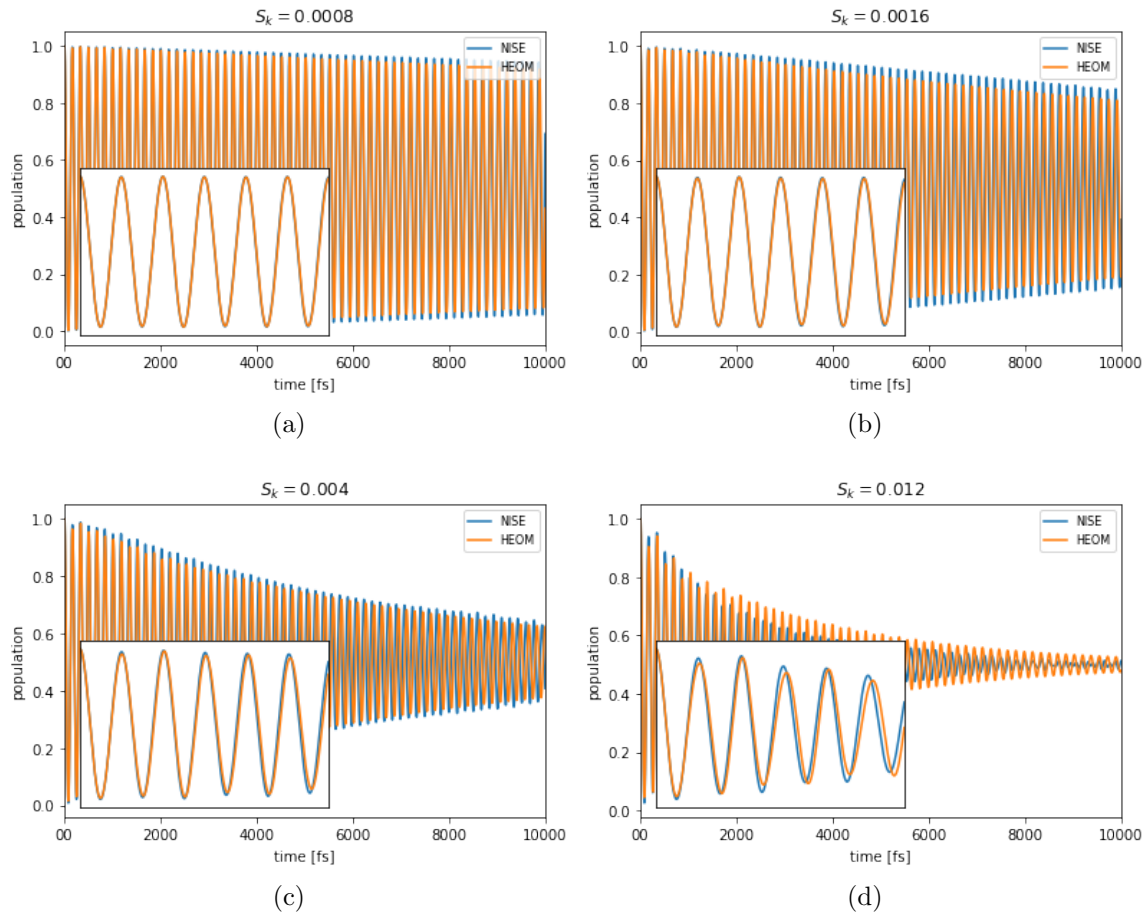


Figure 15: Population dynamics of a two site system under a Lorentzian-Bath. Comparison between HEOM and NISE for different reorganization energies

This result confirms, once again, that the assumption of low system-bath coupling breaks down for large values of reorganization energy. Thus, resulting in discrepant population dynamics calculated from NISE and HEOM. An additional observation drawn from these simulations is the effect that having resonant frequencies, i.e. frequencies in the vicinity of the system's transition frequency, has on our calculations. Where specially the NISE method fails to depict the rapid transitions of the exciton.

Length-dependent fluctuations

The unexpected results coming from a 1000 fs-long trajectory implemented in our test system shed some light on another important factor to take into account when implementing the noise algorithm for NISE calculations: The trajectory of the populations seemed to be dependent on the length of the structured noise generated.

In response, this dependency of the exciton dynamics was studied by calculating the standard deviation of energy fluctuations generated using different time lengths. The standard deviation is plotted in figure 16 against different lengths T . The calculations were done by using:

$$\sigma_T = \sqrt{(\eta_T)^2} \quad (4-45)$$

Where the standard deviation σ_T was then averaged over 2000 different realizations for every T .

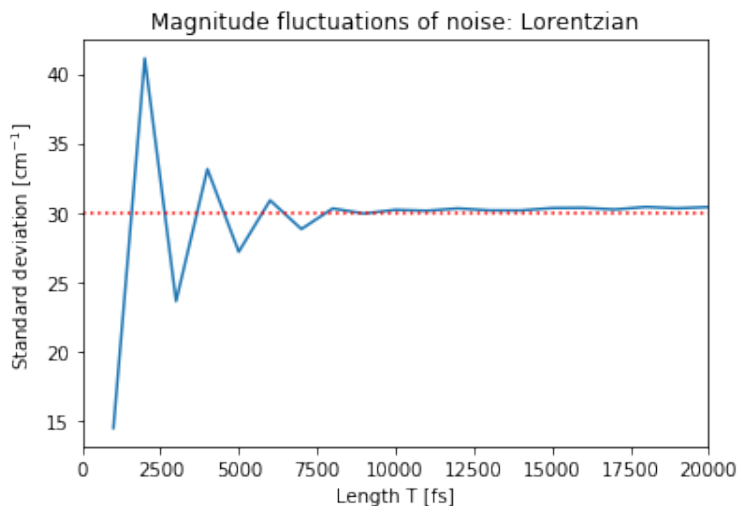


Figure 16: Variation of the magnitude in the structured Noise in dependence of length of trajectory T

The results show that the deviation stabilizes after a certain threshold, one which coincides with the minimum length necessary to attain an accurate simulation of the spectral density calculated from the noise. As a rule of thumb, it is necessary to verify the correct reproduction of the desired spectral density, when deciding the parameters to employ in the noise generation used in the calculation of the population dynamics through NISE.

5 Conclusions

The study of some of the fundamental processes responsible for life on Earth, like the one carried by Light Harvesting complexes, require a quantum dynamical description, where the system and the surrounding environment need to be taken into account. The mean-field Ehrenfest Dynamics methods, based on a semi-classical approach, is an alternative to study these so-called open systems [15][16]. A variation coined as Numerical Integration of Schrödinger Equation (NISE) is of special interest throughout this work. In this, the time-dependent Schrödinger equation is solved using the time-dependent Hamiltonian of the system, where the time-dependent term corresponds to the energy site fluctuations and appears as a result of the effect of the bath on the system [8][17]. With the two main approximations introduced by NISE, a high temperature limit and the neglect of quantum feedback, the autocorrelation of the site fluctuations is related to the spectral density of the bath by a cosine transform. This simple, yet crucial, relation allows for the calculation of the exciton dynamics, when the energy fluctuations are known.

These fluctuations can be obtained through computationally expensive Molecular Dynamics (MD) or Quantum Mechanic/Molecular Mechanic (QM/MM) simulations [20][28][39]. However, despite their expensive implementation, often times they do not provide the desired degree of accuracy when replicating the experimental spectral densities. An alternative method is the use of structured noise generators for the particular scenarios where the spectral density of the bath can be approximated by simple spectral shapes, e.g. Drude and Gaussian functions [8][17]. This thesis introduced a noise-generating algorithm, based on the sampling of white Gaussian noise and the implementation of Fourier transformations [31], that is capable to generate fluctuations following any desired spectral density. With it, we expected to broaden the calculation of exciton dynamics to systems ruled by more general system-bath interactions.

Using the proposed algorithm, we successfully replicated the previously established benchmarks in the simulation of spectral densities. And while the implementation of the Drude and Gaussian spectral shapes was focused on single peaks, the extensibility of the calculations lies entirely on computational considerations, such as the increase of integration intervals, as well as the use of newly gauged parameters. The extension of these simulations to more complex spectral densities was explored through Lorentzian functions. On this regard, a spectral density made up of 62 peaks was simulated from experimental data, with the high frequency components approximated by Lorentzian peaks [30]. Several tests were performed for different time steps, length and realizations of the noise generated. Additionally, we implemented an integration scheme tailored for highly oscillating functions, where an increase in the complexity and accuracy of the desired spectral densities resulted in an increase in the integration intervals. The insight provided by these tests was used along with theoretical considerations, such as the nyquist criteria and sampling frequency necessary for the highest and lowest frequencies, resulting in an outstanding agreement between the theoretical and calculated spectral density.

The spectral density simulations were useful, not only to verify the validity and scope of the algorithm, but also to establish the parameter regimes that can be employed when the noise is used in further calculations. Particularly, the thresholds needed to obtain reasonable SD shapes were taken into account when computing the exciton dynamics. When these requirements were not fulfilled, the agreement between the populations obtained from NISE and HEOM was rather poor. These observations were supported by the dependency of the fluctuations magnitude on the length of the trajectories generated. For our concrete example, the variations in the magnitude decreased after 10,000 fs, which coincides with the minimum required length necessary to reproduce the Lorentzian peak in question. Additionally, the results of the two approaches for the calculation of population dynamics was assessed under different reorganization energies. The results confirmed the limitations of the NISE method when implemented in settings with high system-bath coupling, as an unavoidable consequence of the underlying assumptions. Another important observation arose from the use of frequencies close to the transition frequency of the system Hamiltonian. In these situations, the exciton dynamics show a phenomena known as quantum revival, and which the NISE approach failed to handle. Such effects can be attenuated with the change in reorganization energy and broadening factors, in order to modify the damping magnitude.

6 Outlook

With the algorithms and simulations explored in this thesis, we have developed the tools needed to describe systems governed by a wide variety of spectral densities. The Δ FLN spectra of a FMO complex, for instance, can be approximated by using Gaussian and Lorentzian peaks for the low and high frequency components, respectively [30]. Alternatively, the spectral density of the LHCII complex can be modeled by a combination of Drude functions, for the low frequency modes, and Lorentzian functions, for the high frequency modes [28]. Furthermore, the essence of the algorithm herein implemented is the possibility of generating noise with any arbitrary spectral density, resulting in an even more extensive set of systems that can be studied through the methods here explored.

Even though additional computational artifacts, like windowing the trajectories [17], were not implemented, the batch approach and operations used, allowed for the noise trajectories to be averaged over many realizations. Due to the optimization in vectorized operations using Python, these simulations were attained at a relatively low computational cost. Specially, when comparing the results and performance for the simulations here implemented with those from MD or other classical approaches, the improvement in accuracy and processing times becomes evident [28]. There is, however, still a big room for improvement in the implementation of the enveloping algorithm. Possible speed-up methods can come from the use of parallel computing techniques employed in the manipulation of the noise tensors. On the other hand, advanced computational techniques, such as machine learning, could be implemented when gauging the parameters necessary to simulate new spectral densities. This would decrease the testing-times, while improving the accuracy, and thus, allowing for the calculation of a wider domain of spectral densities. An algorithm generating stochastic processes following a given autocorrelation has also been implemented in [40]. There,

however, the sampled random variables are **complex** Gaussian distributed. Further, the implementation does not seem to take into account the thermal correction factors shown in equation 2-31. An in-depth evaluation of the algorithm therein proposed can be done to compare its performance and applicability in our calculations against the enveloping algorithm here proposed.

The introduction of this algorithm opens new ways of studying quantum dynamical systems than until now, were limited to those described by simple SD or inaccurate computations using MD or QM/MM. With this, the study of population dynamics of systems ruled by system-bath interactions with a higher complexity is enabled, accompanied by an improved computational feasibility and accuracy. I am confident the work provided here will open new horizons in the study of some of the most fundamental processes powering life on earth, like photosynthesis. The thorough understanding of the dynamics involved in these processes, would be of utmost importance when aiming to use the same principles in a diverse regime of applications, such as solar energy conversion, design of electronic circuits, and development of medicines and drugs [41].

7 Acknowledgments

I would like to thank my supervisor Prof. Ulrich Kleinekathöfer for his guidance, support, and amiability, not only through the thesis process, but since the very first moment I became his student. Likewise, I would like to thank Yannick Holtkamp for the uncountable times his insight helped me to move the project forward, for the hours spent helping me prepare for the different stages of the thesis, and for sparking my interest in the project and its applications. To both of you, thank you for inspiring me to discover more about the beauty behind the computational physics field.

I would also like to thank my friends, in particular, those who were part of my three years at university. Thank you for making this process a lot more enjoyable and thank you for helping me grow both academically and personally. I will always be grateful for the moments I spent with you.

Finally, I would like to thank my family, and specially my parents for being my motivation and my greatest support. Thank you for your love and encouragement, despite being so far away. This achievement is also yours.

Thank you,

Cristian Emiliano Godínez Ramírez

Statutory Declaration

Family Name, Given/First Name	Godinez Ramirez, Cristian Emiliano
Matriculation number	30002010
Kind of thesis	Bachelor

English: Declaration of Authorship

I hereby declare that the thesis submitted was created and written solely by myself without any external support. Any sources, direct or indirect, are marked as such. I am aware of the fact that the contents of the thesis in digital form may be revised with regard to usage of unauthorized aid as well as whether the whole or parts of it may be identified as plagiarism. I do agree my work to be entered into a database for it to be compared with existing sources, where it will remain in order to enable further comparisons with future theses. This does not grant any rights of reproduction and usage, however.

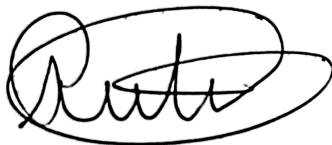
The Thesis has been written independently and has not been submitted at any other university for the conferral of a PhD degree; neither has the thesis been previously published in full.

German: Erklärung der Autorenschaft (Urheberschaft)

Ich erkläre hiermit, dass die vorliegende Arbeit ohne fremde Hilfe ausschließlich von mir erstellt und geschrieben worden ist. Jedwede verwendeten Quellen, direkter oder indirekter Art, sind als solche kenntlich gemacht worden. Mir ist die Tatsache bewusst, dass der Inhalt der Thesis in digitaler Form geprüft werden kann im Hinblick darauf, ob es sich ganz oder in Teilen um ein Plagiat handelt. Ich bin damit einverstanden, dass meine Arbeit in einer Datenbank eingegeben werden kann, um mit bereits bestehenden Quellen verglichen zu werden und dort auch verbleibt, um mit zukünftigen Arbeiten verglichen werden zu können. Dies berechtigt jedoch nicht zur Verwendung oder Vervielfältigung.

Diese Arbeit wurde in der vorliegenden Form weder einer anderen Prüfungsbehörde vorgelegt noch wurde das Gesamtdokument bisher veröffentlicht.

May 17th, 2021



A Appendix

The sections included here serve for deeper explanations or additional information of ideas discussed in the main text.

A.1 Proof of correctness

From [31], the following section contains the proof that noise generated using the enveloping algorithm indeed follows the desired power spectral density $S(\omega)$, and hence, the spectral density $J(\omega)$. Here, we assume that $S(\omega) = S(-\omega)$. The averages are performed over the different noise realizations.

$$\begin{aligned}
 \mathcal{F}[\langle \eta(t)\eta(0) \rangle](\omega) &= \int e^{-i\omega t} dt \langle \mathcal{F}^{-1} [\sqrt{S} \cdot \mathcal{F}[W]](t) \cdot \mathcal{F}^{-1} [\sqrt{S} \cdot \mathcal{F}[W]](0) \rangle \\
 &= \int e^{-i\omega t} dt \left\langle \int \frac{d\omega'}{2\pi} e^{i\omega' t} \sqrt{S(\omega')} \mathcal{F}[W](\omega') \cdot \int \frac{d\omega''}{2\pi} e^{i\omega'' t} \sqrt{S(\omega'')} \mathcal{F}[W](\omega'') \right\rangle \\
 &= \int e^{-i\omega t} dt \int \frac{d\omega'}{2\pi} \int \frac{d\omega''}{2\pi} e^{i\omega' t} \sqrt{S(\omega')S(\omega'')} \langle \mathcal{F}[W](\omega') \mathcal{F}[W](\omega'') \rangle
 \end{aligned}$$

The remaining expectation value can be computed explicitly by reducing it to the defining δ correlation function of white noise.

$$\begin{aligned}
 \langle \mathcal{F}[W](\omega') \mathcal{F}[W](\omega'') \rangle &= \int dt_1 \int dt_2 e^{-i\omega' t_1} e^{-i\omega'' t_2} \langle W(t_1) W(t_2) \rangle \\
 &= \int dt_1 \sqrt{\kappa^2} e^{-i(\omega' + \omega'') t_1} \\
 &= \sqrt{\kappa^2} \cdot 2\pi \delta(\omega' + \omega'')
 \end{aligned}$$

This resolves the ω'' integral and allows to use the symmetry of $S(\omega)$:

$$\begin{aligned}
 \mathcal{F}[\langle \eta(t)\eta(0) \rangle](\omega) &= \int e^{-i\omega t} dt \int \frac{d\omega'}{2\pi} e^{i\omega' t} \sqrt{\kappa^2} \sqrt{S(\omega')S(-\omega')} \\
 &= \mathcal{F}[\mathcal{F}^{-1}[\sqrt{\kappa^2 S(\omega')}] (t)](\omega) \\
 &= \sqrt{\kappa^2} S(\omega) \quad \square
 \end{aligned} \tag{1-46}$$

This shows that when $\kappa^2 = 1$, the Fourier transform of the autocorrelation of η is the power spectral density $S(\omega)$.

A.2 Constants

This section contains the constants used in the different implementations and calculations.

Constant	Numerical Value	Units
c	2.99792458×10^8	$\text{m}\cdot\text{s}^{-1}$
\hbar	0.658211951	$\text{eV}\cdot\text{fs}$
k_B	8.6173303×10^{-5}	$\text{eV}\cdot\text{K}^{-1}$
k_B	0.6950389	$\text{cm}^{-1}\cdot\text{K}^{-1}$

The Boltzmann constant was included twice, since the units were adjusted when calculating spectral densities from noise generated in eV or cm^{-1} .

A.3 Alternative integration methods

For the calculation of the spectral density, the cosine transform was attempted via different methods other than the one finally implemented. All of them ended up being not used, due to reasons explained here:

- **DCT From Python:** Since the relation between $J(\omega)$ and $C(t)$ is by a cosine transform, the function `fftpack.dct()` from the SciPy library was used to implement the discrete cosine transform (DCT) [35]. The problems with this approach were two-folded: First, from the documentation, it was not entirely clear the frequencies used for the calculations. Secondly, while the general shape was correct, a scaling factor appeared in all the simulations. If these two problems could be overcome, the DCT function would be a good alternative as the calculations are faster than with an interpolation method.
- **“Manual” Integration:** Using integration routines provided in the SciPy library, such as the *Simpson* scheme, the spectral density could be obtained. This, however, only worked for low frequencies, as for higher components, the integral diverged.
- **Fourier Transform Properties:** As shown in the derivation of 2-31, the Fourier transform of the autocorrelation $C(t)$ and the spectral density $J(\omega)$ are related as:

$$J(\omega) = \frac{\beta\omega}{2\pi} \Re\{\mathcal{F}[C(t)]\}$$

With:

$$\Re\{\mathcal{F}[C(t)]\} = 2 \int_0^{\infty} C(t) \cos(\omega t) dt$$

Based, on this, the Fourier transform was calculated using the functions provided by NumPy [34], and the real part was multiplied by the thermal factor. This method gave the right shape, but with a wrong scaling factor and a non-so smooth result.

A.4 Testing System: Hamiltonian and Noise generation

The 4×4 Hamiltonian used in the comparison between the different structured-noise algorithms, as shown in figure 12, is the following:

$$H = \begin{bmatrix} 10 & -87.7 & 5.5 & -5.9 \\ -87.7 & 130 & 30.8 & 8.2 \\ 5.5 & 30.8 & -190 & -53.5 \\ -5.9 & 8.2 & -53.5 & -80 \end{bmatrix} \quad (1-47)$$

And was obtained from [38].

Additionally, the parameters employed for the noise generation are:

Parameter	Numerical Value	Units
λ	35	cm^{-1}
$1/\gamma$	25	fs
dt	1	fs
T	2000	fs
R	1000	-

The graph generated in the implementation section, where the population dynamics were discussed, was generated using the same parameters, with the difference that the length used there was instead $T = 1000$ fs

References

- [1] R. E. Blankenship, *Molecular Mechanisms of Photosynthesis*. New York: John Wiley & Sons, 2014, ISBN: 978-1-118-79696-2.
- [2] C. Olbrich, J. Strümpfer, K. Schulften, and U. Kleinekathöfer, “Theory and simulation of the environmental effects on fmo electronic transitions”, *J. Phys. Chem*, vol. 2, no. 1771–1776, 2011.
- [3] M. I. Mallus, “On dephasing and exciton transfer in light-harvesting complexes”, Ph.D. dissertation, 2017. [Online]. Available: <http://urn-resolving.de/urn:nbn:de:gbv:579-opus-1007420>.
- [4] A. Damjanović, I. Kosztin, U. Kleinekathöfer, and K. Schulten, “Excitons in a photosynthetic light-harvesting system: A combined molecular dynamics, quantum chemistry, and polaron model study”, *Phys. Rev. E*, vol. 65, 3 Mar. 2002.
- [5] R. J. Cogdell, A. Gall, and J. Köhler, “The architecture and function of the light-harvesting apparatus of purple bacteria: From single molecules to in vivo membranes”, *Quarterly Reviews of Biophysics*, vol. 39, no. 3, pp. 227–324, 2006.
- [6] “Dynamics of isolated and open quantum systems”, in *Charge and Energy Transfer Dynamics in Molecular Systems*. John Wiley & Sons, Ltd, 2011, ch. 3, pp. 67–190, ISBN: 9783527633791.
- [7] A. Mortaza, U. Kleinekathöfer, C. Curutchet, and B. Mennucci, “Impact of electronic fluctuations and their description on the exciton dynamics in the light-harvesting complex pe545”, *The Journal of Physical Chemistry B*, vol. 121, no. 6, pp. 1330–1339, 2017.
- [8] Y. Holtkamp, “Enhancing temperature corrections in open system quantum dynamics simulations with machine learning”, M.S. thesis, 2020.
- [9] U. Weiss, *Quantum Dissipative Systems* -. Singapur: World Scientific, 1999, ISBN: 978-9-810-24092-9.
- [10] R. J. Silbey, “Principles of nonlinear optical spectroscopy by shaul mukamel”, *Journal of the American Chemical Society*, vol. 118, no. 50, pp. 12 872–12 872, 1996.
- [11] B. González-Soria, F. Delgado, and A. Anaya-Morales, “Predicting entanglement and coherent times in FMO complex using the HEOM method”, *Journal of Physics: Conference Series*, vol. 1730, no. 1, p. 012 033, Jan. 2021.
- [12] A. Ishizaki and Y. Tanimura, “Multidimensional vibrational spectroscopy for tunneling processes in a dissipative environment”, *J. Chem Phys.*, vol. 123, no. 1, p. 014 503, 2005.
- [13] A. Ishizaki and Y. Tanimura, “Quantum dynamics of system strongly coupled to low-temperature colored noise bath: Reduced hierarchy equations approach”, *Journal of the Physical Society of Japan*, vol. 74, no. 12, pp. 3131–3134, 2005.
- [14] A. Ishizaki and G. R. Fleming, “Unified treatment of quantum coherent and incoherent hopping dynamics in electronic energy transfer: Reduced hierarchy equation approach”, *J. Chem. Phys.*, vol. 130, no. 234111, 2009.
- [15] D. Marx and J. Hutter, “Modern methods and algorithms of quantum chemistry”, *NIC Series*, vol. 3, no. 329–477, 2000.

- [16] P. V. Parandekar and J. C. Tully, “Detailed balance in ehrenfest mixed quantum-classical dynamics”, *Journal of Chemical Theory and Computation*, vol. 2, no. 2, pp. 229–235, 2006.
- [17] M. Aghtar, J. Liebers, J. Strümpfer, K. Schulten, and U. Kleinekathöfer, “Juxtaposing density matrix and classical path-based wave packet dynamics”, *The Journal of Chemical Physics*, vol. 136, no. 214101, 2012.
- [18] V. May, O. Kuhn, O. Kühn, and J. W. bibinitperiod S. Inc, *Charge and Energy Transfer Dynamics in Molecular Systems* -, 3rd ed. Weinheim: Wiley-VCH, 2011, ISBN: 978-3-527-40396-7.
- [19] A. Damjanović, I. Kosztin, U. Kleinekathöfer, and K. Schulten, “Excitons in a photosynthetic light-harvesting system: A combined molecular dynamics, quantum chemistry, and polaron model study”, *Physical Review E*, vol. 65, no. 031919, Jul. 2002.
- [20] M. Aghtar, J. Strümpfer, J. Olbrich, K. Schulten, and U. Kleinekathöfer, “The fmo complex in a glycerol–water mixture”, *The Journal of Physical Chemistry B*, vol. 117, no. 24, pp. 7157–7163, 2013.
- [21] A. V. Oppenheim and A. S. Willsky, *Signals and Systems - Pearson New International Edition*. Harlow: Pearson Education Limited, 2013, ISBN: 978-1-292-02590-2.
- [22] Q. Kong, T. Siau, and A. Bayen, *Python Programming and Numerical Methods - A Guide for Engineers and Scientists*. Amsterdam, Boston: Academic Press, 2020, ISBN: 978-0-128-19550-5.
- [23] J. O. Smith, *Mathematics of the Discrete Fourier Transform (DFT) - With Audio Applications*, 2nd ed. Stanford, California 94305 USA: Center for Computer Research in Music and Acoustics (CCRMA), 2007.
- [24] J. W. Cooley and J. W. Tukey, “An algorithm for the machine calculation of complex fourier series”, *Math. Comp*, vol. 19, no. 297-301, 1965.
- [25] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering, Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2019, ISBN: 978-1-108-42209-3 Hardback.
- [26] A. V. Oppenheim and G. C. Verghese, *Class notes for: Introduction to communication, control and signal processing*, Mar. 2010.
- [27] I. Gatland, R. Roy, and G. Vemuri, “Fast, accurate algorithm for numerical simulation of exponentially correlated colored noise”, *Phys Rev A Gen Phys*, vol. 38, no. 5938-5940, 1988.
- [28] S. Maity, V. Daskalakis, M. Elstner, and U. Kleinekathöfer, “Multiscale qm/mm molecular dynamics simulations of the trimeric major light-harvesting complex ii”, (in press).
- [29] S. Maity, B. Bold, J. Dahyabhai, M. Sokolov, T. Kubar, M. Elstner, and U. Kleinekathöfer, “Dftb/mm molecular dynamics simulations of the fmo light-harvesting complex”, SUPPLEMENTAL INFORMATION.
- [30] M. Rätsep and A. Freiberg, “Electron–phonon and vibronic couplings in the fmo bacteriochlorophyll a antenna complex studied by difference fluorescence line narrowing”, *Journal of Luminescence*, vol. 127, no. 1, pp. 251–259, 2007, ISSN: 0022-2313.

- [31] M. David, B. Coish, and F. Fehse, “Monte-carlo analysis of dephasing in non-adiabatic state conversions”, Internship Report - McGill University, 2019.
- [32] R. U. Halas Nanophotonics Group, *Energy unit conversions*, Available at <http://halas.rice.edu/conversions> (2021).
- [33] P. S. Foundation, *Python language reference, version 3.9*, Available at <https://www.python.org/> (2021).
- [34] C. R. Harris, K. J. Millman, S. J. van der Walt, *et al.*, “Array programming with NumPy”, *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. DOI: [10.1038/s41586-020-2649-2](https://doi.org/10.1038/s41586-020-2649-2). [Online]. Available: <https://doi.org/10.1038/s41586-020-2649-2>.
- [35] P. Virtanen, R. Gommers, T. E. Oliphant, *et al.*, “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”, *Nature Methods*, vol. 17, pp. 261–272, 2020. DOI: [10.1038/s41592-019-0686-2](https://doi.org/10.1038/s41592-019-0686-2).
- [36] F. Johansson *et al.*, *Mpmath: A Python library for arbitrary-precision floating-point arithmetic (version 1.2.08)*, <http://mpmath.org/>, Jan. 2021.
- [37] A. Paszke, S. Gross, F. Massa, *et al.*, “Pytorch: An imperative style, high-performance deep learning library”, in *Advances in Neural Information Processing Systems 32*, Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.
- [38] NanoHUB, *Exciton dynamics lab for light harvestig complexes (gpu-heom)*, Available at <https://nanohub.org/tools/gpuheompop> (2021).
- [39] S. Maity, B. M. Bold, J. D. Prajapati, M. Sokolov, T. Kubař, M. Elstner, and U. Kleinekathöfer, “Dftb/mm molecular dynamics simulations of the fmo light-harvesting complex”, *The Journal of Physical Chemistry Letters*, vol. 11, no. 20, pp. 8660–8667, 2020, PMID: 32991176. DOI: [10.1021/acs.jpcllett.0c02526](https://doi.org/10.1021/acs.jpcllett.0c02526).
- [40] R. Hartmann and W. T. Strunz, “Exact open quantum system dynamics using the hierarchy of pure states (hops)”, *Journal of Chemical Theory and Computation*, vol. 13, no. 12, pp. 5834–5845, 2017, PMID: 29016126. DOI: [10.1021/acs.jctc.7b00751](https://doi.org/10.1021/acs.jctc.7b00751).
- [41] D. Gust, *Why study photosynthesis*, Available at <https://live-bioenergy.ws.asu.edu/content/why-study-photosynthesis> (2021).