

Collisional Methods

Analysis of the Unraveling of Quantum Master Equations in the Quantum Jump and
Diffusional Limits

Francesco Ciavarella - University of Padua

francesco.ciavarella@studenti.unipd.it

January 16, 2026

Contents

1	Introduction	3
1.1	Open Quantum System an Quantum Master Equation	3
1.2	Stochastic Unraveling of Quantum Master Equation	4
1.3	Collisional Method	4
1.4	Theoretical Algorithm	6
1.5	Complete Evolution and Ancilla Trace	7
2	Case Study : Exciton Dimer	7
3	Computational Implementation	8
3.1	Language Choice and Libraries	8
3.2	Structure of the <code>Main_Dynamic.py</code> Script	9
3.2.1	System Definition and Evolution Operators	9
3.2.2	Lindblad Evolution	10
3.2.3	Isolated System Evolution	10
3.2.4	Trace on Ancilla Evolution	10
3.2.5	Stochastic Trajectory	11
3.2.6	Parameters	12
3.3	Simulation Execution and Data Storage	12
4	Expected Results	13
4.1	3D Visualization: <code>Bloch_Sphere.ipynb</code>	13
4.2	2D Visualization: <code>Plot.ipynb</code>	13

1 Introduction

1.1 Open Quantum System and Quantum Master Equation

First, it is essential to frame the scope of this project within the field of *Environment-Assisted Quantum Transport (ENAQT)*. This framework focuses on evaluating the effects of the surrounding environment on open quantum systems. The latter are usually constituted of a set of qubits representing different chromophores' energy states. Such systems are typically described using the *density matrix* formalism, where we define $\rho_S = \sum_k^N |\Psi_k^S\rangle \langle \Psi_k^S|$.

This approach is fundamental as it allows for the treatment of quantum systems subject to classical statistical uncertainty (i.e. mixed states), combining quantum coherent superposition with classical probabilities.

Specifically, ENAQT explores how environment-induced phenomena can enhance the efficiency of excitonic transport across different sites (i.e., the transfer of excitation). In this context, one of the simplest environmental effects is the induction of *Decoherence*: a mechanism that facilitates the transfer of excitation toward the final target site by destroying the coherence between different sites and localizing the excitation in a single one.

In the study of open quantum systems, various equations can describe the dynamics of these sites, defined as the evolution of the system's density matrix over time.

To account for environmental effects, it would formally be necessary to study the evolution of the *total system*, i.e. comprising both the sites and the environment's degrees of freedom, and subsequently perform a *partial trace* over the latter. However, since the dimensionality of this total Hilbert space is often computationally unmanageable, the standard approach involves the use of *Dynamical Maps*. These maps allow us to obtain the evolution of the system alone while still accounting for the effect of the external environment, such as *Dephasing* or *Relaxation*.

Among the most well-known *Dynamical Maps* there are the *Redfield equation*, which is microscopically derived, and the *Lindblad equation*. The latter is based on the theory of quantum semigroups and is derived with a more mathematical approach to ensure the evolution remains *Completely Positive and Trace Preserving* (CPTP) at all times. In this work, we will specifically focus on the Lindblad master equation, which reads :

$$\dot{\rho}_S = -i[\hat{H}, \rho_S] + \sum_j \gamma_j \left(\hat{L}_j \rho_S \hat{L}_j^\dagger - \frac{1}{2} \left[\hat{L}_j^\dagger \hat{L}_j, \rho_S \right]_+ \right) = \hat{\mathcal{L}} \rho_S \quad (1)$$

where γ_j are the *Lindblad rates*, that may be different for every sites; while \hat{L}_j are the *Jump operator*, which describe the environment effect on the system; eventually $\hat{\mathcal{L}}$ represents the *Liouvillian Super Operator*.

In this specific study, we will focus exclusively on the phenomenon of *pure Dephasing*. This process is responsible for the suppression of quantum coherences between different sites, favoring excitonic transport. For our model, this mechanism is described by the jump operators

$$\hat{L}_j = \sigma_z^{(k)} \quad (2)$$

where σ_z is the Pauli matrices acting on the k -th site (represented by a qubit). This choice of operator ensures that while the off-diagonal elements of the density matrix (the coherences) decay over time, the diagonal elements (the populations) remain unaffected.

1.2 Stochastic Unraveling of Quantum Master Equation

The concept of *unraveling* refers to the decomposition of the deterministic Master Equation into a statistical ensemble of single stochastic trajectories. Instead of directly evolving the density matrix ρ_S , which describes the average state of the ensemble, one evolves a single wave function $|\psi(t)\rangle$ subject to random events, which represents the direct effect of the environment on the quantum state. The full dynamics of the density matrix is then reconstructed by averaging over a large number of these realizations:

$$\rho_S(t) \approx \frac{1}{M} \sum_{j=1}^M |\psi_j^S(t)\rangle \langle \psi_j^S(t)| \quad (3)$$

where M is the number of simulated trajectories.

This approach offers different advantages such as the reduction of the computational cost, since with if N is the dimension of the Hilbert space, the density matrix ρ involves N^2 elements and its evolution scales as $O(N^2)$, instead of the wave function, that contains only N components and scales as $O(N)$. Another advantage is that the Master Equation describes only the average effect of the environment on the System Density Matrix, while in the stochastic unraveling, analyzing a single realization, it's possible to understand how effectively the environment affects the system, distinguishing between the fundamental quantum uncertainty and the statistical mixture resulting from decoherence.

In this sense it's possible to define two opposite limits, describing the environment effect on the system, which are the so called :

- **Diffusive Limit:** The environment induces small changes in the system's state but acts continuously in time, behaving as a source of random noise. This regime is physically associated with a *weak measurement* performed on the system.
- **Quantum Jump Limit:** The environment induces a strong, discontinuous modification of the system's wave function, but the probability of this event occurring is very low in a single time step. In contrast to the diffusive case, this limit corresponds to a *strong measurement* performed on the system.

1.3 Collisional Method

While there are several established algorithms for unraveling the Quantum Master Equation—such as the *Stochastic Schrödinger Equation* (SSE) or the *Monte Carlo Wave Function* (MCWF) method, this work focuses on the *Collisional Model* framework.

The primary objective is to implement and explore this approach, which reconstructs the continuous dynamics through a sequence of discrete interactions, called *Collisions*, occurring over a finite time step Δt . In this model, the environment is not treated as a continuum but is represented by a set of auxiliary units called *Ancillas*, typically modeled as qubits (two-level systems).

The dynamics proceeds as follows:

1. **Initialization:** At the beginning of each step, the System and the current Ancilla are in a *product state*, $\rho_{tot} = \rho_S \otimes \rho_A$.
2. **Collision:** They undergo a joint unitary evolution governed by a specific *Collisional Hamiltonian*, which generally creates entanglement between the System and the Ancilla.
3. **Measurement:** A measurement is performed on the Ancilla. Due to the quantum correlations established during the collision, the outcome of this measurement induces a conditional change on the System's state.
4. **Reset:** After the measurement, the current ancilla is discarded. For the next time step, the system interacts with a new, identical ancilla initialized in the same initial state, and the cycle repeats. This ensures the Markovian nature of the dynamics.

Since the probability of measuring one of the two Ancilla's states is finite, it is possible to implement a *Stochastic Algorithm* which randomly chooses the change to apply to the System, according to the Ancilla's measurement outcome. In this way, it is possible to obtain a single *Random Trajectory*.

But first of all, focus on the Collisional Methods Hamiltonian form, which in general is divided in two term :

$$\mathcal{H}_{CM} = \mathcal{H}_{Exc} + \mathcal{H}_{Collision} \quad (4)$$

where

$$\mathcal{H}_{Exc} = \mathcal{H}_{Site} + V_{Hopping} = \sum_{j=1}^N \frac{\varepsilon_j}{2} \sigma_z^j \otimes \mathbb{I}^{\otimes N} + \sum_{\langle j,j' \rangle} \frac{V_{j,j'}}{2} \left(\sigma_x^j \sigma_x^{j'} \otimes \mathbb{I}^{\otimes N} + \sigma_y^j \sigma_y^{j'} \otimes \mathbb{I}^{\otimes N} \right) \quad (5)$$

represents the *Isolated System Hamiltonian*, representing the internal dynamics of the N sites. The Hopping term specifically is what allows the effective Exciton transfer, since the Collisional term just facilitates the transport by canceling the coherence created. The summation on j runs over the different System's sites and the σ_α^j term are defined as :

$$\sigma_\alpha^j = \mathbb{I}^{\otimes j-1} \otimes \sigma_\alpha \otimes \mathbb{I}^{\otimes N-j} \quad (6)$$

The form of $\mathcal{H}_{Collision}$ specifically defines the unraveling regime. In order to recover the correct QME form, it's fundamental to correctly initialize the Ancilla's state, which is deeply related to the specific form of *Interaction Hamiltonian*, $\mathcal{H}_{Collision}$.

In this context it's possible to associate the two opposite regime, seen before, with two different configurations of the Collisional Method (i.e. form of the Collisional Hamiltonian and Ancilla's state initialization) :

Diffusive Limit

$$\mathcal{H}_{Coll} = \sum_{j=1}^N c_j \sigma_z^j \otimes \sigma_z^{a_j} \quad \text{and} \quad \rho_A = \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad (7)$$

The Ancilla is in the complete mixed state, i.e. given a reservoir of Ancillas, there's the 0.5 probability of finding it in the $|0_a\rangle$ state and so the same for $|1_a\rangle$ state; in this case the \mathcal{H}_{Coll} acts on the Ancilla as a $\sigma_z^{a_j}$, introducing a phase shift on the state.

Quantum Jump Limit

$$\mathcal{H}_{Coll} = \sum_{j=1}^N c_j \sigma_z^j \otimes \sigma_x^{a_j} \quad \text{and} \quad \rho_A = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad (8)$$

The Ancilla is always initialized in the $|0_a\rangle$ state and the \mathcal{H}_{Coll} acts on it as a $\sigma_x^{a_j}$ flipping its state.

In both cases c_j represents the *Collision Intensity* for every j -th site, a constant related to the *Lindblad Rates* γ_j by the equation:

$$c_j = \sqrt{\gamma_j/4\Delta t} \quad (9)$$

Once we've defined the $\mathcal{H}_{Collision}$, it's important to see how it affect the System's evolution. In general the wave function at a time step $t + \Delta t$ is given by the resolution of the Schrödinger Equation, which is :

$$|\Psi_S(t + \Delta t)\rangle = \mathcal{U} |\Psi_S(t)\rangle = e^{(-i\mathcal{H}_{CM}\Delta t)} |\Psi_S(t)\rangle = e^{(-i\mathcal{H}_{Exc}\Delta t)} e^{(-i\mathcal{H}_{Coll}\Delta t)} |\Psi_S(t)\rangle \quad (10)$$

where the last equivalence is valid thanks to a *Trotter-Suzuki* decomposition and allows to separate the evolution in two step:

1. Isolated evolution via \mathcal{H}_{Exc}
2. Collisional evolution via \mathcal{H}_{Coll}

Focusing on the collisional evolution we can analytically define the Evolution Operator U form and how it modifies the System wf , in the two different limits. Note that in this derivation we will refer to only one site of the System's Hilbert space:

Diffusive Limit

$$U_{collision} = \exp(-ic_j \sigma_z^j \otimes \sigma_z^{a_j} \Delta t) = \cos(c_j \Delta t) \mathbb{I}^j \otimes \mathbb{I}^a - i \sin(c_j \Delta t) \sigma_z^j \otimes \sigma_z^{a_j} \quad (11)$$

In this case, since we have to deal with a completely mixed Ancilla's Density Matrix we divide the evolution in two different results based on the Ancilla's state, including in this sense the Classical uncertainty, which is 0.5 for both the states:

$$|\Psi_0(t + \Delta t)\rangle = \left[\cos(c_j \Delta t) - i \sin(c_j \Delta t) \sigma_z^j \right] |\Psi_S(t)\rangle \otimes |0_a\rangle \quad (12)$$

$$|\Psi_1(t + \Delta t)\rangle = \left[\cos(c_j \Delta t) + i \sin(c_j \Delta t) \sigma_z^j \right] |\Psi_S(t)\rangle \otimes |1_a\rangle \quad (13)$$

Depending on the ancilla's state, the system undergoes a rotation in the opposite direction in the Hilbert space; specifically, if the ancilla is in $|0_a\rangle$ the phase shift is $+c_j \Delta t \sigma_z^j$, whereas with $|1_a\rangle$ the phase shift is $-c_j \Delta t \sigma_z^j$. This stochastic alternation between opposite phase shifts results in a quantum random walk of the system's phase. In the limit $\Delta t \rightarrow 0$, this discrete process converges to a continuous Brownian motion (Wiener process). Since $\Delta t \rightarrow 0$ the effect on the system will be very small.

Quantum Jump Limit

$$U_{collision} = \exp(-ic_j \sigma_z^j \otimes \sigma_x^{a_j} \Delta t) = \cos(c_j \Delta t) \mathbb{I}^j \otimes \mathbb{I}^a - i \sin(c_j \Delta t) \sigma_z^j \otimes \sigma_x^{a_j} \quad (14)$$

$$|\Psi(t + \Delta t)\rangle = \cos(c_j \Delta t) |\Psi_S(t)\rangle \otimes |0_a\rangle - i \sin(c_j \Delta t) \sigma_z^j |\Psi_S(t)\rangle \otimes |1_a\rangle \quad (15)$$

In this case we have a modification on the System wf only when we measure $|1_a\rangle$, but the variation isn't premultiplied by a small factor so the change will be significant. On the other hand the probability of measuring $|1_a\rangle$ is given by

$$P_{1_a} = Tr_S \left[\langle 1_a | \Psi(t + \Delta t) \rangle \langle \Psi(t + \Delta t) | 1_a \rangle \right] = \sin^2(c_j \Delta t) \quad (16)$$

So in this case we have a significant variation but that happens very rarely.

For more information about the equation derivation please see the Demonstration file, in particular section 1.1 and 2.1. Moreover all the form of $|\Psi(t + \Delta t)\rangle$ can be lead back to a Lindblad evolution, as specified in section 1.3 and 2.3

1.4 Theoretical Algorithm

Concatenating the evolution given by Eq.(12) and Eq.(13) or Eq.(15), it's possible to create a Stochastic Algorithm that creates a Trajectory in the two limits; averaging then different trajectories realizations it's possible to recover the Lindblad evolution.

Quantum Jump Algorithm

1. Isolated System's evolution with $\mathcal{U}_{Exc} = \exp(-i \mathcal{H}_{Exc} \Delta t)$
2. Extraction for every different site of a Random Number α in $[0, 1]$
3. If $\alpha < \sin^2(c_j \Delta t)$ it means that there's been an effective collision that has flipped the initial Ancilla's state $|0_a\rangle$ to $|1_a\rangle$ and so the System's wf gets modified by σ_z^j on every j -th site; if $\alpha > \sin^2(c_j \Delta t)$ the system remain unchanged (i.e. application of \mathbb{I}^j)
4. Once all the sites have been processed, measure and store System's Observables at that time step (like population)
5. Repeat the algorithm for the next time step

Diffusive Algorithm

1. Isolated System's evolution with $\mathcal{U}_{Exc} = \exp(-i \mathcal{H}_{Exc} \Delta t)$
2. Extraction for every different site of a Random Number α in $[0, 1]$
3. If $\alpha < 0.5$ it means that state $|0_a\rangle$ has been measured and so the System's *wf* gets modified via EQ.(12) for every *j-th* site; otherwise state $|1_a\rangle$ has been measured and the System's *wf* gets modified via EQ.(13).
4. Once all the sites have been processed, measure and store System's Observables at that time step (like population)
5. Repeat the algorithm for the next time step

Since the update operators in both limits are unitary, the wave function norm is theoretically preserved at each step. However, to prevent numerical errors from accumulating over long simulations, it is standard practice to enforce renormalization after every time step.

1.5 Complete Evolution and Ancilla Trace

While the stochastic unraveling focuses on individual trajectories of the *wf*, the *Collisional Model* can also be formulated deterministically for the complete Density Matrix $\rho_S \otimes \rho_A$.

The evolution of the system's Density Matrix ρ_S over a time step is obtained by evolving the total state and then performing a *partial trace* over the Ancilla's degrees of freedom. This operation mathematically corresponds to averaging over all possible measurement outcomes of the ancilla.

In this way we are creating a so called *Dynamical Map* Φ , which reads:

$$\rho_S(t + \delta t) = \Phi[\rho_S(t)] = \text{Tr}_A \left[\mathcal{U}(\rho_S(t) \otimes \rho_A) \mathcal{U}^\dagger \right] \quad (17)$$

Since \mathcal{U} is unitary, this map is guaranteed to be Completely Positive and Trace Preserving (CPTP). It can be shown that in the continuous limit, i.e $\Delta t \rightarrow 0$ the discrete map defined in Eq.(17) converges to the differential Lindblad Master Equation (see section 1.3 in Demonstration file).

Thus, the Collisional Model serves as a rigorous microscopic derivation of the Lindblad dynamics.

2 Case Study : Exciton Dimer

Since this work is a preliminary study of *Collisional Methods*, the system considered is one of the simplest relevant to quantum transport: the *Exciton Dimer*. This system consists of two sites, where each site is modeled as a two-level system (qubit). The computational basis for the composite Hilbert space is given by:

$$\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\} \quad (18)$$

where the notation $|s_1 s_2\rangle$ represents the tensor product $|s\rangle_1 \otimes |s\rangle_2$, with 0 denoting the ground state and 1 the excited state.

Since the objective is to study the exciton transport between sites, the initial state is defined as one site completely in excited state and the other in ground state. The interaction between the two sites is defined by the Hopping Potential v , built to allow only exchange between the two excited states, conserving the total number of excitations. In this way the dynamics is restricted to the *single-excitation manifold* and allows us to effectively reduce the four-dimensional Hilbert space to a two-dimensional subspace, that can be represented by a single qubit built on the excited states $\{|01\rangle, |10\rangle\}$. As already said, the effect induced by the environment is just the *Pure Dephasing*, which essentially destroys the coherence between the two states, facilitating the localization of the exciton.

The key observables analyzed during the time evolution are the Excited States Populations, representing the probability of finding the Excitation in site 1 or 2. Furthermore it's possible to reconstruct the state vector in the *Bloch Sphere*, which could help visualizing the evolution in time.

The results obtained with the *Collisional Method*, both the Stochastic Trajectories and the Average Dynamics of the ρ_S (obtained tracing out the Ancilla), will be compared with the Lindblad Master Equation and the Isolated System Dynamics.

3 Computational Implementation

Once the theoretical background is defined, we now focus on the computational implementation. The objective is the *Numerically* generation of *Stochastic Trajectories* that will be benchmarked as described above. This task is performed by the `Main_Dynamics.py` program, where the mathematical formalism of *Collisional Method* is translated into a time-discrete simulation, reproducing the iterative interaction between the System and the Ancilla reservoir, in the *Quantum Jump* and *Diffusive Limit*. For the Results visualization two dedicated Jupyter Notebooks were developed:

1. `Plot.ipynb` : based on the Python library `Matplotlib`, this program gives some examples of the 2D plot that can be made with the dynamics results; the `%matplotlib ipynb` command enables interactive data exploration directly within the notebook interface.
2. `Bloch_Sphere.ipynb` : based on the visualization tools of the Python library `Qutip`, this program generates an animated GIF representing the dynamics on the State Vector on the Bloch Sphere, providing a 3D representation of the dynamics.

3.1 Language Choice and Libraries

The simulation is implemented in *Python*, chosen for its object-oriented features to treat mathematical entities such as wave functions and operators as high-level objects. In this regard, two fundamental libraries are used:

- **Numpy** : used for linear algebra operations, allows direct representation of quantum states as vectors, Hamiltonians as matrices and provides tools for optimized operations such as matrix multiplication.
- **Qutip** (*Quantum Toolbox in Python*) : employed for specific quantum mechanical operations that require structural knowledge of the Hilbert space; this library provides complete Master Equation solver too.

Going into more detail, **Numpy** works with dense matrices and is made for generic linear algebra operation; instead **Qutip** works with `Qobj` (Quantum Object), that are sparse matrices with additional metadata describing the physical state (es bra, ket ...) and the associated Hilbert space.

For completeness, also the **Scipy** library is used, in particular its `Scipy.linalg` module, dedicated to linear algebra, from which it's imported the `expm` function, that allows to numerically build up the Evolution in time operator \mathcal{U} as matrix exponential of the Hamiltonian.

While the algorithmic logic follows a procedural paradigm, this high-level approach allows the code to remain readable and close to the mathematical formulation.

Technical note : since `Qobj` and `Np array` are incompatible, it's often used the following control line that converts a `Qobj` in a `Np array`, since the core calculations rely on **Numpy**

```
# Convert a generic variable 'var' to a dense array if it is a Qobj
var = var.full() if hasattr(var, "full") else np.array(var, dtype=complex)
# or a List element by element
List = [L_element.full() if hasattr(L_element, "full") else np.array(L_element,
dtype=complex) for L_element in List]
```

This control line checks if an object has the `.full()` method (typical of QuTiP objects) and extracts the dense matrix; otherwise, it treats the object as a standard array.

3.2 Structure of the Main_Dynamic.py Script

The `main_dynamic` script represents the core of the computational implementation. Beyond the computation of Evolution Trajectories, this code explored the **Convergence Limit** of the model, by testing different Δt values and Number of Trajectories N_{Traj} to average over. With this approach the main loop is run multiple times with different value of the two parameters, in order to eventually compare the results. The variation of Δt necessitates a dynamic update of the physical parameters: specifically, the interaction coefficients $c_j = \sqrt{\gamma_j/4\Delta t}$ values are recalculated at each Δt variation.

The program is organized into distinct logical blocks, each dedicated to a specific task; this section is dedicated to the description and explanation of the algorithmic choices and the data structures of each section.

3.2.1 System Definition and Evolution Operators

The first section is dedicated to the Hamiltonians construction and the definition of the Evolution Operator $\mathcal{U}(\Delta t)$. The first is modularized into dedicated functions:

1. `system_Hamiltonian` : constructs the System's Hamiltonian matrix based on Eq.(5) with the input parameters, defined in the `Parameters` section: `N_site` (number of sites), `E` (array of site energies, *randomly generated*) and `V_pot`, array that contains the coupling constants for every unique pair of sites (i, j) with $i < j$, organized as

$$V_{pot} = [V_{12}, V_{13}, \dots, V_{1N_{site}}, V_{23}, \dots, V_{N_{site}-1, N_{site}}]$$

in the Exciton Dimer $N_{site} = 2$ case this reduces to a single element array $V_{pot} = [V_{12}]$.

It's important noting that this definition of V_{pot} defines an *all-to-all connectivity* and may be transform in a "Nearest-Neighbor Interaction" modulating the single value.

Giving this input the function returns a Numpy matrix representing the System Hamiltonian.

2. `interaction_Hamiltonian_N_ancillas`: constructs the Interaction Hamiltonian both in the *Diffusive Limit* Eq.(7) and in the *Quantum Jump Limit* Eq.(8); the input required are `N_site` and `c_CM` (constant Collisional Methods), which is an array containing the value of c_j defined in Eq.(9) for every site.
3. `hamiltonian_N_ancillas` : constructs the Complete Hamiltonian, firstly expanding the System Hamiltonian in the complete Hilbert Space (i.e. System + Ancilla) and then summing it to the Interaction Hamiltonian (already construct in the total space)

It's important to focus on how formally an Hamiltonian operating on different sites is constructed : following the formal definition given in Eq.(6), i.e. any local operator acting on the j -th site can be expressed as a tensor product of the operator on the target site and identity matrices on all others. This can be computationally achieved by initializing a list of $2 \times 2\mathbb{I}$, representing the complete space, and replacing in the j -th position the operator acting on the j -th site. Finally the tensor product of the list elements is computed using the `QuTip` function `tensor`. In the case of \mathcal{H}_{Coll} the composite System has dimension $2^{2N_{site}}$ and two operators are placed simultaneously: σ_z on the j -th system site and the corresponding ancilla operator on the $(N+j)$ -th site as reported in the following example:

```
# Construction of Collisional Hamiltonian
for j in range(N_site):
    op_list = [identity(2) for _ in range(2 * N_site)] #list of identity
    op_list[j] = sigmaz() # Acts on the j site
    op_list[N_site + j] = anc_op # Acts on the j ancilla, with index N + j
    H_term = (c_CM[j] * tensor(op_list)).full() # tensor product for list's element
    H_int += H_term
```

The Evolution Operator $\mathcal{U} = \exp(-i\mathcal{H}\Delta t)$ is computationally constructed in two different ways:

- with the `Scipy.linalg` function `expm`
- Diagonalizing the \mathcal{H} matrix with the `Numpy.linalg` function `Numpy.linalg.eigh`, using the *Eigenvalues* w to build a diagonal matrix with element $\mathcal{U}_{diag} = \exp(-iw\Delta t)$ and finally, using the *Eigenvectors* V , moving back to the original basis $\mathcal{U} = V\mathcal{U}_{diag}V^\dagger$.

3.2.2 Lindblad Evolution

In this section the benchmark evolution via *Lindblad Master Equation* is calculated. The *Jump Operators* L_j and the *Lindblad rates* γ_j are defined in the `Parameters` section.

First the *Liouvillian Superoperator* is computationally construct with the `Liouvillian` function. The effective time evolution, related to Eq.(1), is implemented by the `Lindblad_evo` function. In this approach the Lindblad ME is solved in the *Liouvillian Space*, where the System's Density Matrix is vectorized and allowing the Liouvillian to be represented as a square matrix. This formulation enables the construction of the Time Evolution Operator \mathcal{U} via either the `expm` method or diagonalization, as described previously. Once the \mathcal{U} has been defined and applied the resulting update vectorized state $\rho_S(t + \Delta t)$ is stored in `rho_vec_list`, a list containing the resulting ρ_S at every time step. Finally this list is returned in vectorized form or reshaped in standard Density Matrices.

3.2.3 Isolated System Evolution

The second benchmark dynamics is the one obtained without the *Collisional Hamiltonian*, i.e. governed only by \mathcal{H}_{Exc} . The trajectory can be easily obtained evolving the System wave function with the Unitary Time Evolution Operator $\mathcal{U} = \exp(-i\mathcal{H}_{Exc}\Delta t)$. To optimize memory usage, the complete wf is not stored at every step; instead the site's Population are computed and stored at every time step in `pop_traj_isolated`, a 2D array with dimensions $[N_{Site}, N_{tSteps}]$. To extract this observables it's used the `projectors` list, which contains projectors operators on the excited states in matrix form.

For the specific case of the Exciton Dimer, the expected physical behavior is a coherent, continuous oscillation of the population between the two sites (Rabi oscillations).

3.2.4 Trace on Ancilla Evolution

The first dynamics simulated using *Collisional Methods* is the one based on the deterministic evolution of $\rho_S \otimes \rho_A$ with

$$\mathcal{U}_{CM} = \exp(-i\mathcal{H}_{CM}\Delta t) = \exp(-i(\mathcal{H}_{Exc} + \mathcal{H}_{Coll})\Delta t) \quad (19)$$

At each step, the Ancilla's degrees of freedom are traced out, as described in Section 1.5.

This process is implemented in the `compute_trace_ancilla` function. After constructing the Ancilla's Density Matrix, either in the *Quantum Jump* or *Diffusive Limit*, and the \mathcal{U}_{CM} operator, the function follows this algorithm to compute the evolution for every timestep:

1. Expansion in the System-Ancilla composite Hilbert space by tensor product between $\rho = \rho_S \otimes \rho_A$
2. Unitary Evolution of the total Density Matrix : $\rho(t + \Delta t) = \mathcal{U}_{CM}(\Delta t) \rho(t) \mathcal{U}_{CM}^\dagger(\Delta t)$
3. *Partial Trace* over the Ancilla's degree of freedom
4. Using the `projectors` list, calculation and storage of every site's Population in `pops_complete`, a 2D array with dimensions $[N_{Site}, N_{tSteps}]$

Focusing on the *Partial Trace* operation, it's computed using two important `Numpy` functions:

```
# 3 : Partial Trace on the ancilla's degree of freedom
rho_tot_reshaped = rho_tot.reshape(dim_sys, dim_anc, dim_sys, dim_anc)
rho_sys = np.trace(rho_tot_reshaped, axis1=1, axis2=3)
```

The first line reshapes the 2D density matrix into a *Block Matrix* with $N_{Site} \times N_{Site}$ blocks and indices $(i_S, \alpha_A, j_S, \beta_A)$. This structure explicitly separates the System and Ancilla indices.

The second line computes the trace contraction over the Ancilla indices (axis 1 and 3), effectively summing the diagonal elements of the Ancilla subspace for every System configuration. This translates mathematically into:

$$\rho_{S_{i_S, j_S}} = \text{Tr}_A [\rho] = \sum_{\alpha_A = \beta_A} \rho_{Tens} [i_S, \alpha_A, j_S, \beta_A] \quad (20)$$

Remark : Note that in this case the *Trotter-Suzuki approximation* on the Time Evolution Operator is **not** applied, i.e. it is construct with the total \mathcal{H}_{CM} in Eq.(4). With this approach the trajectory obtained is exact (except for the time discretization error) and not affected by the error introduced by the *Trotter-Suzuki approximation* : $\mathcal{O}(\Delta t^2/m^2)$ at first order.

3.2.5 Stochastic Trajectory

This section is dedicated to the computation of the *Stochastic Trajectory* generated by the *Collisional Methods*. Before dealing with the algorithm, it's fundamental to define the quantities calculated by the algorithm and their storage structure:

- **pop_traj_sample** : 3D array with dimension $[N_{Site}, N_{tSteps}, N_{SampleTraj}]$; stores the population of each site at every time step for 3 independent sampled trajectory.
- **avg_pop_traj** : 2D array with dimension $[N_{Site}, N_{tSteps}]$; stores the average value over all different trajectories of each site's Population at every time step.
- **sample_x, sample_y, sample_z** : 2D arrays with dimension $[N_{tSteps}, N_{SampleTraj}]$; store from 3 sampled trajectories the Components of the *Bloch State Vector* projected onto the subspace of the Single-Exciton states and computed via the **compute_Bloch_Sphere** function, valid specifically for the Exciton Dimer case.
- **avg_x, avg_y, avg_z** : 1D arrays with dimension $[N_{tSteps}]$; stores average value over different trajectories of the *Bloch State Vector* components.
- **count** : 1D array with dimension $[N_{Trajectories}]$; stores the cumulative number of measurement of $|1_a\rangle$ that occurred during the entire trajectory, for each trajectory.
- **avg_count** : float, represents the average value of count calculated over all N_{Traj} Trajectories.

The Evolution Algorithm relies on the logic described in Section 1.4. The simulation allows switching between the *Quantum Jump* and *Diffusive Limit* by simply modifying the **method** parameter.

These two regimes differ in:

- the Probability of measuring the state $|1_a\rangle$
- the Evolution applied to the state's *wf*

The steps of the Algorithm for a Single Trajectory are:

1. Evolution of the complete *wf* with only Site's Evolution Operator $\mathcal{U}_{Exc}(\Delta t) = \exp(-i \mathcal{H}_{Exc} \Delta t)$
2. Extraction of a Random Number α in $[0, 1]$ for each System's site
3. If $\alpha < \textit{probability condition}$ defined by the dynamics limit, apply to the selected site's state the evolution related to the measurement of $|1_a\rangle$. Otherwise apply the evolution defined by the measurement of $|0_a\rangle$ (see Eq.(12) and Eq.(13) or Eq.(15))
4. Once all the sites have been processed, Normalization of the resulting complete state
5. Measurement and storage of the site Populations (using the **Projectors** list) and the Bloch State Vector components (using the **compute_Bloch_Sphere**) in the appropriate arrays.
6. Repeat the algorithm for the next time step, until the total *Dynamics Time* is covered.

7. Once the Single Trajectory is completed, restart the Algorithm for a new Independent Trajectory.
8. Finally , once all trajectories are computed and stored, compute the Average Value over different trajectories at every time step, for the Populations and *Bloch State Vector* components.

3.2.6 Parameters

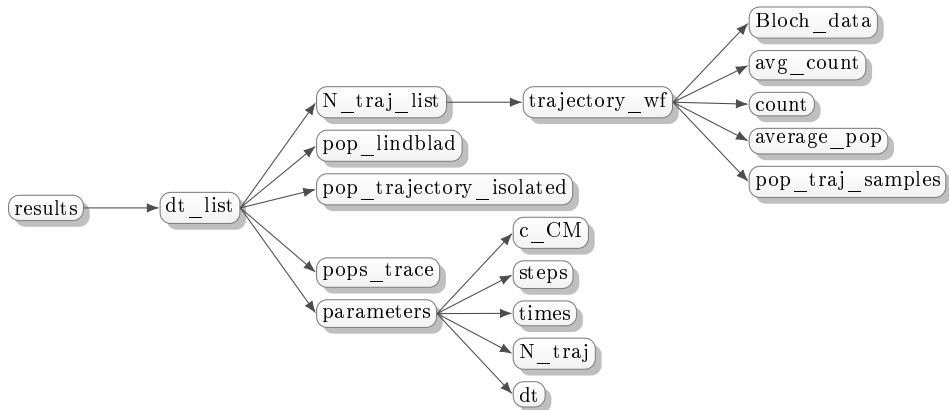
This section initializes all physical constants and simulation variables required for the dynamics. The parameters are organized into four logical categories:

1. System Parameters:
 - **N_site**: Number of system sites ($N = 2$).
 - **E**: Site's energies initialized with random Gaussian disorder around the value 1.5.
 - **V_array**: Hopping potential between sites, as defined Section 3.2.1.
2. Dynamics Parameters:
 - **dt_list**: List of different time steps Δt that defines the related **steps_list** and **times_list**.
 - **N_traj_list**: List of Number of Trajectories.
 - **tf**: Final simulation time.
3. Interaction Constants:
 - **g_j**: List of Lindblad dephasing rate γ_j for every j -th site.
 - **c_CM_list**: The collisional strength coefficients, recalculated for each Δt as $c = \sqrt{\gamma/4\Delta t}$.
4. Initial States and Operators:
 - **psi_sys_initial**: System initialized in the state $|01\rangle$.
 - **rho_anc**: Ancilla state, defined as Pure ($|0\rangle\langle 0|$) for QJ or Mixed ($\mathbb{I}/2$) for Diffusive limit.
 - **L_j** : List of Lindblad Jump Operator
 - **projectors**: List of measurement operators (P_{10}, P_{01}) used to extract populations.
5. Dynamics Limit:

the **INTERACTION_LIMIT** parameter defines the **method** variable, allowing the simulation to be performed in either the *Quantum Jump* or *Diffusive Limit*.

3.3 Simulation Execution and Data Storage

The simulation iterates over Δt and N_{Traj} values, with nested loops, in order to analyze *Convergence Properties*. To optimize storage, full trajectory are retained for only the first three samples per configuration. All outcomes are organized in the **results Dictionary**, structured as follows:



4 Expected Results

Once the computational data has been generated and stored in the **results** Dictionary, the analysis is performed using two dedicated Jupyter Notebooks:

4.1 3D Visualization: Bloch_Sphere.ipynb

This notebook provides a geometrical interpretation of the dynamics within the Single-Exciton subspace, mapping the state evolution onto the **Bloch Sphere**. The expected visualization highlights the fundamental difference between Single Realizations and Average over Trajectories:

- **Ensemble Average (Green Vector):** This Vector represents the density matrix $\rho_S(t)$. Due to the decoherence induced by the environment, its length is expected to decrease over time (shrinking inside the sphere).
- **Single Trajectories (Blue Points):** These represent the pure states $|\psi(t)\rangle$ of individual stochastic Trajectory. They must always lie on the surface of the sphere (norm = 1), as the evolution remains unitary.

The visual behavior of the blue points distinguishes the two dynamical limits:

- In the *Quantum Jump Limit*, the state mostly follows the Coherent Unitary Rotation but occasionally undergoes sudden, discontinuous "jumps".
- In the *Diffusive Limit*, the Trajectory is continuous but "noisy". The state performs a Brownian-like motion on the sphere's surface, fluctuating constantly around the ideal unitary path.

4.2 2D Visualization: Plot.ipynb

The second notebook focuses on the 2D plotting of populations and convergence metrics. Based on the generated data, the analysis focuses on four main comparisons:

1. *"Comparison Lindblad, Trace, Avg Trajectories"* : verify the validity of the method. The Average of the Stochastic Trajectories and the Partial Trace method must perfectly overlap with the benchmark solution provided by the Lindblad Master Equation.
2. *"Comparison Trajectories: Collisional vs Isolated"* : visualizes the effect of \mathcal{H}_{Coll} . While the Isolated System shows perfect Rabi oscillations, the Collisional Trajectory exhibit always present noise (Diffusive) or occasionally jump (Quantum Jump) .
3. *"Comparison different N_{traj} "* : demonstrates that as the trajectories number increases (e.g., from 100 to 10,000), the residual noise in the average population decreases, converging to the analytical Lindblad result.
4. *"Comparison different dt "*: validates the continuous limit. It show that as $\Delta t \rightarrow 0$, the Discrete Collisional Dynamics correctly reproduce the continuous differential equations of the system.