Coding closed and open quantum systems in MATLAB: applications in quantum optics and condensed matter

Ariel Norambuena$^{1,3}$, Diego Tancara$^2$ and Raúl Coto$^1$

1 Centro de Investigación DAIITA Lab, Facultad de Estudios Interdisciplinarios, Universidad Mayor, Chile
2 Faculty of Physics, Pontificia Universidad Católica de Chile, Avda. Vicuña Mackenna 4860, Santiago, Chile
E-mail: ariel.norambuena@umayor.cl

Received 18 November 2019, revised 13 March 2020
Accepted for publication 25 March 2020
Published 9 June 2020

Abstract
We develop a package of numerical simulations implemented in MATLAB to solve complex many-body quantum systems. We focus on widely used examples that include the calculation of the magnetization dynamics for the closed and open Ising model, dynamical quantum phase transition in cavity QED arrays, Markovian dynamics for interacting two-level systems, and the non-Markovian dynamics of the pure-dephasing spin-boson model. These examples will be useful for undergraduate and graduate students with a medium or high-level background in MATLAB, and also for researchers interested in numerical studies applied to quantum optics and condensed matter systems.

Keywords: quantum mechanics, open quantum systems, MATLAB, numerical simulation, many-body quantum systems

(Some figures may appear in colour only in the online journal)

1. Introduction

Myriad current research fields in quantum optics and condensed matter demand numerical analysis of complex many-body systems (MBS). Interesting effects such as dynamical quantum phase transition (DQPT), meta-stability, and steady states, among others, can naturally emerge in MBS. These effects are observed in closed decoherence-free dynamics such as the Jaynes–Cummings [1], Jaynes–Cummings–Hubbard [2], trapped ions [3], cavity
opto-mechanics [4] or Ising models [5]. Furthermore, for open quantum systems, the environment may cause memory effects in the reservoir time scale, known as non-Markovianity (NM) [6, 7]. From the theory of open quantum systems, NM can be described either by a time-local convolutionless or a convolution master equation [8]. In most of these cases, handling these kinds of problems with analytical tools is highly demanding, and a numerical approach can appear to be the only solution. For instance, this is the case for systems with non-trivial system–environment interactions [9] and quantum systems with many degrees of freedom.

Nowadays, numerical toolboxes or open-source packages save time when dealing with analytically untreatable problems, without the requirement of a vast knowledge in computational physics. To this end, we found the Wave Packet open-source package of MATLAB [10–12], the tutorial ‘Doing Physics with MATLAB’ [13], the introductory book *Quantum Mechanics with MATLAB* [14], and the Quantum Optics Toolbox of MATLAB [15]. However, most of these does not have examples illustrating many-body effects in both closed and open quantum systems. In this work, we provide several examples implemented in MATLAB to code both closed and open dynamics in MBS. This material will be useful for undergraduate and graduate students, studying either master or even doctoral courses. More importantly, the advantage of using the matrix operations in MATLAB can be crucial to work with Fock states, spin systems, or atoms coupled to light.

MATLAB is a multi-paradigm computational language that provides a robust framework for numerical computing based on matrix operations [16], along with user-friendly coding and high performance calculations. Hence, the solutions that we provide have a pedagogical impact on the way that quantum mechanical problems can be efficiently tackled. This paper is organized as follows. In section 2 we introduce the calculation of observables for decoherence-free closed quantum MBS. We discuss the transverse Ising and the Jaynes–Cumming–Hubbard models with a particular interest in DQPTs. In section 3, we model the Markovian and non-Markovian dynamics of open quantum systems using the density matrix. The dissipative dynamics of interacting two-level systems is calculated using a fast algorithm based on eigenvalues and eigenmatrices. Finally, the non-Markovian dynamics of the pure-dephasing spin-boson model is discussed and modelled for different spectral density functions, in terms of a convolutionless master equation.

2. Closed quantum systems

In the framework of closed quantum MBS the dynamics is ruled by the time-dependent Schrödinger equation [17]:

\[ i\hbar \frac{d}{dt} |\Psi(t)\rangle = \hat{H} |\Psi(t)\rangle, \]

where $|\Psi(t)\rangle$ and $\hat{H}$ are the many-body wavefunction and Hamiltonian of the system, respectively. For a time-independent Hamiltonian, the formal solution for the wavefunction reads

\[ |\Psi(t)\rangle = \hat{U}(t) |\Psi(0)\rangle = e^{-i\hat{H}t/\hbar} |\Psi(0)\rangle, \]

where $\hat{U}(t) = \exp(-i\hat{H}t/\hbar)$ is the time propagator operator. The numerical implementation of this type of dynamics relies on the construction of the initial state $|\Psi(0)\rangle$ (vector), the time
propagator $\hat{U}(t)$ (matrix), and the system Hamiltonian $\hat{H}$ (matrix). For illustration, we introduce first two spin-1/2 particles described by the following Hamiltonian:

$$\hat{H}_{\text{spins}} = -JS^x_1 S^x_2 - B(S^x_1 + S^x_2). \quad (3)$$

In MATLAB, the above Hamiltonian can be coded as follows (using $J = 1$ and $B = 0.1J$):

```matlab
1 J = 1; % Value of the coupling J  
2 B = 0.1*J; % Value of the magnetic field  
3 Sx = [0 1; 1 0]; % S_x operator  
4 Sz = [1 0; 0 -1]; % S_z operator  
5 I = eye(2); % Identity matrix  
6 Hspins = -J*kron(Sx,Sx) - B*(kron(Sx,I) + kron(I,Sx)); % Hamiltonian
```

where `eye(2)` generates the $2 \times 2$ identity matrix and `kron(X,Y) = X \otimes Y` gives the tensor product of matrices $X$ and $Y$. Now, we illustrate how to solve the spin dynamics for the initial condition $|\Psi(0)\rangle = |\downarrow\rangle \otimes |\downarrow\rangle/\sqrt{2}$ from the initial time $t_i = 0$ to the final time $t_f = 2T$ with $T = \pi/(2B)$. We must point out that the method we will use next is not valid for time-dependent Hamiltonians. For completeness, we compute the average magnetization along the $\alpha$ direction, which is defined as

$$\langle M_\alpha \rangle = \frac{1}{N} \sum_{i=1}^{N} \langle \Psi(t)|S_\alpha^i|\Psi(t)\rangle, \quad \alpha = x, y, z, \quad (4)$$

where $N$ is the number of spins and $|\Psi(t)\rangle$ is the many-body wavefunction given in equation (2). In our particular case, $N = 2$, and thus we can write the following code to compute $\langle M_z \rangle$:

```matlab
1 down = [0 1]'; % Quantum state down = [0 1]'  
2 Psi_0 = kron(down,down); % Initial wavefunction  
3 T = 2*pi/(2*B); % Period of time  
4 Nt = 1000; % Number of steps to construct time vector  
5 t_i = 0; % Initial time  
6 tf = 2*T; % Final time  
7 dt = (tf-t_i)/(Nt-1); % Step time dt  
8 t = t_i:dt:tf; % Time vector  
9 U = expm(-1i*Hspins*dt); % Time propagator operator U(dt)  
10 SSp = (kron(Sz,I)*kron(I,Sz))/2; % Operator $S^z_1 \otimes S^z_2$  
11 Mz = zeros(size(t)); % Average magnetization  
12 for n=1:length(t) % Iteration to find Psi(t) and Mz(t)  
13    if n==1  
14      Psi = Psi_0; % Initial wavefunction  
15    else  
16      Psi = U*Psi; % Wavefunction at time t_n  
17    end  
18    Mz(n) = Psi'*SSp*Psi; % Average magnetization at time t_n  
19 end  
20 plot(t/T,real(Mz),'r-', 'LineWidth',3)  
21 xlabel('$t/T$', 'Interpreter','LaTeX','FontSize', 30)  
22 ylabel('$\langle S^z \rangle$', 'Interpreter','LaTeX','FontSize', 30)  
23 set(gca,'FontSize',21)
```

By carefully looking at our code, we observed that MATLAB provides an intuitive platform to simulate quantum dynamics. First, the Hamiltonian defined in equation (3) can be easily implemented if the operators (matrices $S^i_\alpha$) are defined, and the two-body interaction
Figure 1. (a) Schematic representation of the two-spin system. (b) Average magnetization along the \(z\) direction for \(J = 1\) and \(B = J/10\) and considering the initial condition \(|\Psi(0)\rangle = |\downarrow\rangle_1 \otimes |\downarrow\rangle_2\). The time is divided by the natural period \(T = 2\pi/(2B)\).

is written using the \texttt{kron()} function. The tensor product defined as \texttt{kron()} reconstructs the Hilbert space of the system (two spin-1/2 particles). Second, the time evolution only depends on the initial state (vector \(\Psi_0\)) and the time propagator \(U = \expm(-i\omega t)\), where \(\omega\) is the frequency. We remark that it is convenient to use the \texttt{expm()} function of MATLAB to calculate the exponential of a matrix. Finally, the \texttt{for loop} allows us to actualize the wavefunction \(|\Psi(t)\rangle\) at every time using the recursive relation \(\Psi(t_{n+1}) = U\Psi(t_n)\), where the term on the left-hand side is the wavefunction at time \(t_{n+1} = t_n + \Delta t\). Having numerical stability is necessary to satisfy the condition \(\Delta t \ll \hbar/(\max(\|\hat{H}\|))\). Hence, using the wavefunction we can calculate the average magnetization \(\langle M_z \rangle = \langle \hat{M}_z \rangle_{\langle \Psi(t) | \Psi(t) \rangle} \) employing the command \(M_z(n) = \texttt{Psi}'*\texttt{SSz}^*\texttt{Psi}\).

In figure 1 we plotted the expected average magnetization \(\langle M_z \rangle\) for the two-spin system. Initially, the system has a magnetization \(\langle M_z \rangle = -1\) due to the condition \(|\Psi(0)\rangle = |\downarrow\rangle_1 \otimes |\downarrow\rangle_2\), and then two characteristic oscillations are observed. The slow and fast oscillations in the signal correspond to the influence of \(B\) and \(J\), respectively.

We note that this example does not require any additional package to efficiently run the codes. In the following, all codes may be executed using the available MATLAB library and the functions introduced in each example. In the next subsection, we introduce three relevant many-body problems, namely the transverse Ising, the Rabi–Hubbard and the Jaynes–Cummins–Hubbard models. For each case we write down an algorithm to solve the many-body dynamics.

2.1. Ising model

Let us consider the following transverse Ising Hamiltonian [3]:

\[
\hat{H}_{\text{Ising}} = H_1 + H_0 = -\sum_{i \neq j}^N J_{ij} \hat{S}_i^x \hat{S}_j^x - B \sum_{i = 1}^N \hat{S}_i^z, \tag{5}
\]

where \(J_{ij}\) is the coupling matrix, \(B\) is the external magnetic field, and \(N\) is the number of spins. The spin operators \(\hat{S}_i^\alpha\) with \(\alpha = x, y, z\) and \(i = 1, \ldots, N\) are the Pauli matrices for \(S = 1/2\). The interaction between adjacent spins is modelled using \(J_{ij} = |i - j|^{\alpha}/J\), where \(\alpha \geq 0\) and \(J = (N - 1)^{-1} \sum_{i \neq j} J_{ij}\) [18, 19]. The Hamiltonian given in equation (5) can be coded as
In the above code, we set $B = J/0.42$, as was recently used in [20], to study the dynamical phase transition in trapped ions. We also consider $N = 6$ (number of spins) and $\alpha = 0.2$ (power law $J_{ij} \propto |i-j|^{-\alpha}$) to simulate the spin dynamics. In addition, we have introduced the many-body operators $S_{zi}, S_{xi},$ and $S_{xj}$ (see lines 16, 20 and 21). These operators are mathematically defined as

$$\mathbf{S}_\alpha^i = \mathbf{1} \otimes \mathbf{1} \otimes \cdots \otimes \mathbf{S}_\alpha \otimes \mathbf{1} \otimes \cdots \otimes \mathbf{1}, \quad \alpha = x, y, z,$$

where $\mathbf{1}$ is the single-particle identity matrix. The numerical implementation of the operators $\mathbf{S}_\alpha^i$ is performed using the function getSci(), which is defined as

```matlab
function Sci = getSci(sc, i, Nspins)
    Is = eye(2);
    Op_total = cell(1, Nspins);
    for site = 1:Nspins
        Op_total{site} = Is+double(eq{1,site})*(sc-Is);
    end
    Sci = Op_total{i};
    for site = 2:Nspins
        Sci = kron(Sci, Op_total{site});
    end
end
```

The ground state of the interaction Hamiltonian $H_1$ has a double degeneracy given by $H_1|\Psi_\eta\rangle = E_\eta|\Psi_\eta\rangle$, where $|\Psi_\eta\rangle$ and $|\Psi_\eta\rangle$ are the degenerate ground states and $E_\eta = E_\eta$, the corresponding energies. We consider the initial many-body state $|\Psi(0)\rangle = |\Psi_\eta\rangle \otimes \cdots \otimes |\Psi_\eta\rangle$ to simulate a quench dynamics. The initial condition and the operator $M_x$ (see equation (4)) are implemented using the following code:
To study the DQPT of this system we introduce the following rate function $\Lambda(t)$ [20]:

$$
\Lambda(t) = \min_{\eta = \rightarrow, \leftarrow} \left\{ -N^{-1} \ln(P_\eta(t)) \right\},
$$

(7)

where $P_\eta(t) = |\langle \Psi_\eta | \Psi(t) \rangle|^2$ is the probability to return to the ground state being $|\Psi(t)\rangle = \exp(-i \hat{H}_{\text{Ising}} t) |\Psi(0)\rangle$ ($\hbar = 1$). Finally, to calculate the rate function and the average magnetization $\langle M_x \rangle$, we implement the following code in MATLAB:

```matlab
1 t0 = 0; % Initial time
2 tf = 22; % Final time
3 Nt = 10000; % Number of steps
4 dt = (tf-t0)/(Nt-1); % Step time dt
5 t = t0:dt:tf; % Time vector
6 U = expm(-1i*H*dt); % Time propagator operator U(dt)
7 Lambda = zeros(size(t)); % Rate function Lambda(t)
8 Av_Mx = zeros(size(t)); % Average Magnetization <M_x(t)>
9 for n=1:length(t)
10    if n==1
11       PSI = PSI_0; % Initial wavefunction
12    else
13       PSI = U*PSI; % Wavefunction at time t_n
14    end
15    Pr = abs(Xr'*PSI)^2; % Probability state |Psi_\rightarrow><Psi_\rightarrow|
16    Pl = abs(Xl'*PSI)^2; % Probability state |Psi_\leftarrow><Psi_\leftarrow|
17    Lambda(n) = min(-N*Psi(-1)*log(Pr),-N*Psi(-1)*log(Pl));
18    Av_Mx(n) = PSI'*M_x*PSI; % Average magnetization along x-axis
19 end
20
21 figure()
22 plot(t,Bet,Lambda,'b-','Linewidth',3)
23 xlabel('$B$ $t$', 'Interpreter', 'LaTeX', 'FontSize', 30)
24 ylabel('$\langle M_x \rangle$','Interpreter','LaTeX','FontSize',30)
25 set(gca,'fontsize',21)
26 xlim([0 5])
27
28 figure()
29 box on
30 plot(t+B.real(Av_Mx),'r-', 'Linewidth',2)
31 xlabel('$B$ $t$', 'Interpreter','LaTeX','FontSize',30)
32 ylabel('angle M_x','Interpreter','LaTeX','FontSize', 30)
33 set(gca,'fontsize',21)
34 xlim([0 100])
```
Figure 2. (a) Average magnetization component $\langle M_x \rangle$ for $J/B = 0.42$. (b) Order parameter $\Lambda(t)$ given in equation (7) as function of time. The critical times $t_i^c$ ($i = 1, 2, 3$) are shown in the regions where a non-analytic signature of $\Lambda(t)$ is observed.

Figure 3. Representation of a cavity QED array with a linear system of three interacting cavities. Inside each cavity we have a two-level system (atom) interacting with a cavity mode. The coupling between adjacent cavities is given by $J$. In the above array, the non-zero elements of the adjacent matrix are $A(1, 2) = 1$ and $A(2, 3) = 1$.

In the last code used to study dynamical phase transition in the Ising model we used the function expm() to calculate the time propagator. We remark that this function is calculated using the scaling and squaring algorithm with a Padé approximation (see example 11 in [21] for further details). As a consequence, running the code for high-dimensional matrices, i.e. high-dimensional Hilbert space, increases the consuming time. Therefore, for any many-body Hamiltonian presented in this work, part of the time will depend on the size of the Hilbert space. For instance, for a simulation with $N = 5$ and $N = 10$ spins, the presented code requires approximately 0.4 and 120 s, respectively, using a computer with 8GB RAM and an Intel Core i7 8th Gen processor. To profile a MATLAB code we strongly recommend clicking on Run and Time to improve the performance of each line in the main code.

In figure 2 we plot the average magnetization $\langle M_x \rangle$ and rate parameter $\Lambda(t)$ for the transverse Ising model. Interestingly, the non-analytic shape of $\Lambda(t)$ is recovered, as pointed out in [20]. This non-analytic behaviour prevails at different critical times $t_i^c$ for which $d\Lambda/dt|_{t=t_i^c}$ is not well defined. In this example, when $B = 0$, the minimum energy is described by the degenerate states $|\psi\rangle_0$. After applying the magnetic field $B = 0.42J$, the system tries to find these
minimum energy states. In particular, the sharp peaks in the rate parameter \( \Lambda(t) \) show that the system switches from the many-body state \( |\Psi\rangle \) to \( |\Psi\rangle \).

It is important to emphasise the required memory to simulate the dynamics of the Ising model. The dimension of the Hilbert space for \( N \) spin-1/2 particles is \( \dim_2^N = 2^N \), and considering that a double variable requires 8 bytes we found that the memory required (in GB) is given by the expression \( \dim_2^N \times 8 \times 10^{-9} \). The scaling \( \propto \dim_2^N \) is due to the Hermitian matrix structure of the Hamiltonian. For instance, to simulate 14 spins, we need at least 2.2 GB of memory. In what follows, the reader must check the computational cost of each example.

### 2.2. Quantum phase transition in cavity QED arrays

Now, we consider the calculation of a quantum phase transition (QPT) in cavity QED arrays. The dynamics of a system composed by \( L \) interacting cavities is described by the Rabi–Hubbard (RH) Hamiltonian [22]

\[
\hat{H}_{RH} = \sum_{i=1}^{L} \left[ \omega_i \hat{a}_i^\dagger \hat{a}_i + \omega_a \hat{\sigma}_x^i \hat{\sigma}_-^i + g \hat{\sigma}_x^i \left( \hat{a}_i + \hat{a}_i^\dagger \right) \right] - \sum_{i,j} A_{ij} \left( \hat{a}_i \hat{a}_j^\dagger + \hat{a}_j^\dagger \hat{a}_i \right),
\]

where we have considered all cavities to be equal, \( \omega_c \) is the cavity frequency, \( \omega_a \) is the atom frequency, \( g \) is the light-atom coupling constant, \( J \) is the photon hopping between neighbouring cavities, and \( A_{ij} \) is the adjacency matrix which takes the values \( A_{ij} = 1 \) if the cavities are connected, and \( A_{ij} = 0 \) otherwise. In figure 3 we show a representation of the system for a linear array of three cavities.

The operators acting on the two-level systems are defined as \( \sigma^+_i = \sigma^x_i + \sigma^-_i \), where \( \sigma^+_i = |e\rangle\langle g|_i \) and \( \sigma^-_i = |g\rangle\langle e|_i \), with \( |e\rangle_i \) and \( |g\rangle_i \) the excited and ground states at site \( i \). Note that we are not assuming the first rotating wave approximation (RWA) that leads to \( \hat{\sigma}^+_i (\hat{a}_i + \hat{a}_i^\dagger) \approx \hat{a}_i \hat{\sigma}^+_i + \hat{a}_i^\dagger \hat{\sigma}^-_i \). In the RWA the fast oscillating terms \( \hat{a}_i \hat{\sigma}^-_i \) and \( \hat{a}_i^\dagger \hat{\sigma}^+_i \) are neglected when the light-atom coupling is small, i.e. \( g \ll \omega_c \). To study the problem we introduce two relevant Hamiltonians, namely

\[
\hat{H}_R = \sum_{i=1}^{L} \left[ \omega_i \hat{a}_i^\dagger \hat{a}_i + \omega_a \hat{\sigma}_x^i \hat{\sigma}_-^i + g \hat{\sigma}_x^i \left( \hat{a}_i + \hat{a}_i^\dagger \right) \right],
\]

\[
\hat{H}_{JC} = \sum_{i=1}^{L} \left[ \omega_i \hat{a}_i^\dagger \hat{a}_i + \omega_a \hat{\sigma}_x^i \hat{\sigma}_-^i + g \left( \hat{a}_i \hat{\sigma}_x^i + \hat{a}_i^\dagger \hat{\sigma}_x^i \right) \right],
\]

where equations (9) and (10) are the Rabi and Jaynes–Cummings Hamiltonians, respectively. The photon operators act as follows:

\[
\hat{a}_i |n_i\rangle = \sqrt{n_i} |n_i - 1\rangle, \quad \hat{a}_i^\dagger |n_i\rangle = \sqrt{n_i + 1} |n_i + 1\rangle,
\]

where \( |n_i\rangle \) is the Fock basis of the \( i \)th cavity with \( n_i = 0, 1, 2, \ldots, N_i \), with \( N_i \) a cut-off parameter for the Hilbert space of the cavity mode. First, we define the set of variables of the problem and we introduce the interaction Hamiltonian between cavities
For simplicity, we are only considering two interacting cavities, but the main code can be changed to introduce a more complex dynamics. Furthermore, the topology of the cavity network can be modified by changing the adjacency matrix $A_{ij}$. In addition, in lines 24 and 25 we have introduced the functions acav() and sigmap(), which are required to construct the many-body operators $\hat{a}_i$ and $\hat{\sigma}_i^+$ for a system of $L$ cavities. These photon and atom operators are mathematically defined as

$$\hat{a}_i = \mathbf{1} \otimes \mathbf{1} \otimes \cdots \otimes \hat{a} \otimes \mathbf{1} \otimes \cdots \otimes \mathbf{1}, \quad \hat{\sigma}_i^+ = \mathbf{1} \otimes \mathbf{1} \otimes \cdots \otimes \hat{\sigma}^+ \otimes \mathbf{1} \otimes \cdots \otimes \mathbf{1}. \quad (12)$$

In each cavity we can use the Fock basis $|0\rangle = [1,0,0,\ldots,0]$, $|1\rangle = [0,1,0,\ldots,0]$, $|2\rangle = [0,0,1,\ldots,0]$, and so on. On this basis, we have

$$\hat{a} = \begin{pmatrix} 0 & \sqrt{1} & 0 & 0 & \ldots \\ 0 & 0 & \sqrt{2} & 0 & \ldots \\ 0 & 0 & 0 & \sqrt{3} & \ldots \\ 0 & 0 & 0 & 0 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}. \quad (13)$$

To construct a boson operator $\hat{a}$ in MATLAB we can write $\hat{a} = \text{diag}(\text{sqrt}(1:N)',1)$, where $N$ is the maximum number of bosons and $\text{diag}(X,1)$ returns a square upper diagonal matrix as shown in equation (13). The function acav() represent the operator $\hat{a}_i$, which is numerically defined as in [16], and is explicitly given by
Similarly, the function \( \text{sigmap}() \) represents the operator \( \hat{\sigma}^+ \), and is given by

\[
\text{for site } = 1:L; \\
\text{Op_total} = \text{cell}(1,L); \\
\text{for site } = 1:L; \\
\text{Op_total}(\text{site}) = \text{Is*double(eq(1,site))*(a-Is)}; \\
\text{end}
\]

The many-body wave function at time \( t \) can be obtained by propagating the initial condition \( |\Psi(0)\rangle \) using the evolution operator \( \hat{U} = \exp(-i\hat{H}RHt) \) as follows (\( \hbar = 1 \)):

\[
|\Psi(t)\rangle = \exp(-i\hat{H}RHt) |\Psi(0)\rangle,
\]

where \( |\Psi(0)\rangle \) is the many-body initial state of the system. In this particular case, we chose the Mott-insulator initial condition \( |\Psi(0)\rangle = |1, -\rangle_1 \otimes \cdots \otimes |1, -\rangle_L \), where \( |1, -\rangle_i = \cos(\theta_1)|e\rangle_i \otimes |1\rangle - \sin(\theta_1)|g\rangle_i \otimes |0\rangle \) with \( \tan(\theta_n) = 2g\sqrt{n}/\Delta \) and \( \Delta = \omega_a - \omega_c \) for the detuning. The quantum phase transition from Mott-insulator to superfluid can be studied in terms of the following order parameter [22]:

\[
\text{OP} = \frac{1}{T} \sum_{i=1}^{L} \int_0^T \left( \langle \hat{n}_i^2 \rangle - \langle \hat{n}_i \rangle^2 \right) \text{d}r,
\]

where \( T = J^{-1} \) is an appropriated large time scale to study the dynamics when \( J < g < \omega_1 \), \( \hat{n}_i = \hat{a}_i^\dagger \hat{a}_i + \hat{\sigma}_{ee} \) is the number of polaritonic excitations at site \( i \), and the expectation values are calculated as \( \langle \hat{n}_i^2 \rangle = \langle \Psi(\tau)|\hat{n}_i^2|\Psi(\tau)\rangle \). For comparison we include the rate parameter recently introduced in the Ising model [20]:

\[
\Lambda(t) = -\frac{1}{L} \log_2(P_{1-}(t)).
\]

In the context of cavity QED arrays, \( L \) is the number of cavities and \( P_{1-}(t) = |\langle \Psi(0)|\Psi(t)\rangle|^2 \) is the probability to return to the Mott-insulator state \( |\Psi(0)\rangle = |1, -\rangle_1 \otimes \cdots \otimes |1, -\rangle_L \). The following code shows the MATLAB implementation to study this QPT
In the above code we are running \( N_{\text{sim}} = 25 \) simulations of the Rabi–Hubbard and Jaynes–Cummings–Hubbard models, one for each value of detuning \( \Delta = \omega_a - \omega_c \), and the initial condition \( |\Psi(0)\rangle \) corresponds to each \( \Delta \). We also introduced a parallel calculation of a cycle for using the MATLAB function \texttt{parfor}. In the first iteration the starting parallel pool takes a longer time, but in a second execution of the main code the time is greatly reduced. In lines 11 and 13 we introduced the function \texttt{QuantumSimulationCavityArray()} which solves the Jaynes–Cummings–Hubbard and Rabi–Hubbard models for different detunings.

Furthermore, the function \texttt{QuantumSimulationCavityArray()} introduces the many-body wave function, the time evolution, and the numerical calculation of the parameters given in equations (15) and (16). For further details of the function \texttt{QuantumSimulationCavityArray()}, see appendix A.

The resulting order parameter (15) and rate function (16) are plotted in figure 4 for two interacting cavities using the Jaynes–Cummings–Hubbard and Rabi–Hubbard models, respectively. From the numerical simulation we observe different curves for the order parameter in the region \( \log(\Delta/g) > 0 \). This is because the counter rotating terms \( \hat{a}_i \hat{\sigma}^-_i \) and \( \hat{a}^+_i \hat{\sigma}^+_i \) (neglected in the RWA) are not negligible for \( g = 10^{-2} \omega_c \). Furthermore, the rate function \( \Lambda(t) \) has a remarkable non-analytic peak at \( Jt \approx 0.8 \), which is a characteristic signature of a DQPT [20, 23]. The dynamical phase transition observed in the Jaynes–Cummings–Hubbard model (red lines in figure 4(b)) shows that the probability of returning to the Mott-insulator states, \( P_1(t) = |\langle \Psi(0) | \Psi(t) \rangle|^2 \), is identically zero at the critical time \( Jt \approx 0.8 \). When \( P_1(t) = 0 \), and only for two cavities, the system is in the superfluid state, i.e. photons can freely move between cavities.
In the next section we will introduce a basic technique to address the numerical modelling of open quantum systems in the Markovian and non-Markovian regimes.

3. Open quantum dynamics

In this section, we introduce the Markovian and non-Markovian dynamics of open quantum systems. For the Markovian case, we will focus on the dynamical properties of the Lindblad master equation. We develop a fast and precise numerical method to solve the dynamics using the density matrix formalism. In the non-Markovian regime, we will examine the pure dephasing dynamics arising from the spin-boson model. We will explore memory effects by considering two different spectral density functions, say a super-ohmic and a Lorentzian model, and their effects on the time-dependent rate.

3.1. Markovian quantum master equation

Many open quantum systems can be modelled with a Markovian master equation in the weak coupling approximation [6]

$$\frac{d\rho}{dt} = \mathcal{L}_M(\rho(t)) = -i[H_s, \rho(t)] + \sum_{i=1}^{N_c} \gamma_i \left[ L_i \rho(t) L_i^\dagger - \frac{1}{2} [L_i^\dagger L_i, \rho(t)] \right],$$

where the first term in equation (17) is the conservative dynamics induced by the system Hamiltonian $H_s$. In contrast, the second term describes $N_c$ dissipative channels through operators $L_i$, which in the Markovian approximation can be associated with decay rates $\gamma_i > 0$ for $i = 1, \ldots, N_c$. The general solution of the presented master equation can be written as [5, 24]

$$\rho(t) = \sum_{k=1}^{N} c_k e^{\lambda_k t} R_k,$$

where $c_k = \text{Tr}(\rho(0)L_k)$, $\lambda_k$ are the eigenvalues of the equation $\mathcal{L}_M(R_k) = \lambda_k R_k$ and $\mathcal{L}^\dagger_M(L_k) = \lambda_k L_k$, with $R_k$ and $L_k$ satisfying the orthonormality condition $\text{Tr}(R_k L_{k'}) = \delta_{kk'}$. The general

Figure 4. (a) Order parameter given in equation (15) as a function of $\log_{10}(\Delta/g)$ for two interacting cavities using the Rabi–Hubbard (RH) and Jaynes–Cummings–Hubbard (JCH) models. We consider the values $g = 10^{-2}\omega_c$, $J = 10^{-4}\omega_c$, and $N_{\text{cut-off}} = 2$ as the cut-off parameter for the Hilbert space of each cavity. (b) Rate function $\Lambda(t)$ as a function of the dimensionless time $Jt$. This function exhibits non-analytic points at different times.
solution given in equation (18) does not apply for time-dependent master equations. Therefore, the next method must be applied to systems described by a similar Lindblad structure as shown in equation (17). To numerically solve the eigenmatrix equation $L_M(R_k) = \lambda_k R_k$ we adopt the formalism presented in [25] to rewrite the Lindblad super-operator $L_M(\rho(t))$. As an introductory example, we consider the open version of the transverse Ising model presented in section 2:

$$
\frac{d\rho}{dt} = -i[JS_x^1 S_x^2 - B(S_x^1 + S_x^2), \rho(t)] + \gamma_1 \left[ \hat{S}_z^1 \rho(t) \hat{S}_z^1 \right] - \frac{1}{2} \{ \hat{S}_z^1, \hat{S}_z^1, \rho(t) \},
$$

where $S_z^\alpha = (\hat{S}_z^\alpha - \hat{S}_z^\alpha^\dagger)/2$ for $\alpha = 1, 2$ is the lowering spin operator. In the above equation $\gamma_i$ is associated with emission processes $|\uparrow\rangle_i \rightarrow |\downarrow\rangle_i$, where $\hat{S}_z^1 |\uparrow\rangle_i = |\uparrow\rangle_i$ and $\hat{S}_z^1 |\downarrow\rangle_i = -|\downarrow\rangle_i$. We proceed as follows. To start, we rewrite the density matrix of the $N$-dimensional system as a column vector

$$
\vec{\rho}(t) = \left( \rho_{11}, \rho_{12}, \ldots, \rho_{N1}, \rho_{12}, \ldots, \rho_{N1}, \rho_{N2}, \ldots, \rho_{NN} \right)^T.
$$

In this vector representation, the master equation (17) takes the vector form

$$
\dot{\vec{\rho}}(t) = L_H \vec{\rho}(t).
$$

The full matrix associated with the open evolution can be decomposed as $L = L_{4H} + L_{4\text{dis}}$, where $L_{4H}$ and $L_{4\text{dis}}$ account for the Hamiltonian and dissipative dynamics, respectively. $L_{4H}$ can be coded as follows:

```matlab
1 dim = 4; % Dimension of the total Hilbert space
2 I = eye(dim); % Identity matrix for the total Hilbert space
3 J = 1; % Value of the coupling term J
4 B = 0.1*1; % Value of the magnetic field B
5 Sz = [0 0 0 1]; % Sz operator for one spin
6 Sx = [1 0 0 -1]; % Sx operator for one spin
7 I = eye(2); % Identity matrix for one spin 1/2
8 H = -J kron(Sx,Sx) - B kron(Sz, I) kron(1, Sx); % Hamiltonian of the system
9 L_H = -1i kron(1, H) + 1i kron(1, Is); % Lindblad operator L_H
```

By including the dissipative term $L_{\text{dis}} = \sum_i L_{4i}$, the total Lindblad operator can be written as

```matlab
1 gamma_1 = 0.1*1; % Decay rate gamma_1
2 gamma_2 = 0.5*1; % Decay rate gamma_2
3 S.minus = [0 1 0]; % Decaying operator of the particle 1
4 L1 = kron(S.minus, I); % Overlapping operator of the particle 1 in the total Hilbert space
5 DL1 = gamma_1 kron(0*1, L1) - 0.5 kron(S.minus, L1) + 0.5 kron(1*1, S.minus, L1);
6 L2 = kron(S.minus, I); % Overlapping operator of the particle 2 in the total Hilbert space
7 DL2 = gamma_2 kron(0*1, L2) - 0.5 kron(S.minus, L2) + 0.5 kron(1*1, S.minus, L2);
8 L.dis = DL1 + DL2; % Total Lindblad operator L.dis
9 L = L_H + L.dis; % Total Lindblad operator
```

As the system consists of two interacting qubits, the density matrix $\rho(t)$ has a $4 \times 4$ size. As a consequence, $\vec{\rho}(t)$ has exactly 16 elements. This implies that we have 16 eigenvalues associated with $R_k$ and $L_k$ matrices. However, the eigenvalues and eigenmatrices must be sorted with the same criteria. To do this, we introduce the function sortingEigenvalues(); see appendix B. By choosing the initial condition $\rho(0) = |\Psi(0)\rangle\langle\Psi(0)|$, with $|\Psi(0)\rangle = |\downarrow\rangle_1 \otimes |\downarrow\rangle_2$, we calculate the average magnetization $\langle M_z \rangle = \text{Tr}(M_z \rho(t))$ and we solve the open dynamics. Thus, the last part of the code reads as
Figure 5. (a) Schematic representation of the dissipative Ising model. (b) Average magnetization along the $z$ direction for $J = 1$ and $B = J/10$, and considering the initial condition $\rho(0) = |\Psi(0)\rangle\langle\Psi(0)|$, with $|\Psi(0)\rangle = |↓\rangle \otimes |↓\rangle$. The time is divided by the natural period $T = 2\pi/(2B)$. The envelope (dashed line) is calculated using the eigenvalue with the largest negative part $\lambda_R^N = -0.04$.

The numerical stability of the presented method crucially depends on the value of the step size $d\tau$. The condition to have numerical stability is given by $d\tau \ll 1/\text{max}_k(|\lambda_k|)$, where $\lambda_k$ is the eigenvalues associated with the equation $L(\rho_k) = \lambda_k \rho_k$. In the above example we have chosen $d\tau = 0.0419$ since $\text{max}_k(|\lambda_k|) = 2.2003$. For the rest of the examples we must ensure this condition.

In figure 5 we plotted the expected average magnetization $\langle M_z \rangle$ for the dissipative two-spin system. In comparison with the non-dissipative case (see figure 1) the open Ising model shows a dissipative signal for $\langle M_z \rangle$. The envelope of this signal can be recognized as the exponential factor $\exp(\lambda_R^N t)$ with $N = 16$ in our case. This dissipative behaviour is a consequence of the losses introduced in the Markovian master equation.
3.2. Two-level system coupled to a photon reservoir

In this subsection, we applied the previous algorithm to a different system, i.e. a two-level system interacting with thermal photons. We consider the following Markovian master equation for the atom-field interaction [6]:

$$\frac{d\rho}{dt} = \frac{\Omega}{2}[\hat{\sigma}^+ + \hat{\sigma}^-, \rho(t)] + \gamma_0(N_{ph} + 1) \left[\hat{\sigma}^- \rho(t) \hat{\sigma}^+ - \frac{1}{2} [\hat{\sigma}^+ \hat{\sigma}^+, \rho(t)] + \gamma_0 N_{ph} \left[\hat{\sigma}^+ \rho(t) \hat{\sigma}^+ - \frac{1}{2} [\hat{\sigma}^+ \hat{\sigma}^+, \rho(t)] \right] \right], \quad (21)$$

where $\Omega$ is the optical Rabi frequency, $N_{ph}$ is the mean number of photons at thermal equilibrium, and $\hat{\sigma}^+ = |e\rangle\langle g| = \sigma_\uparrow$. To solve the open dynamics of the reduced two-level system we introduce the parameters of the system and define the Lindbladian, as follows:

Next, we numerically compute the excited state $p_e(t) = \langle e|\rho(t)|e \rangle$ and the physical observable $\langle \hat{\sigma}^+(t) \rangle$.
The numerical method used to solve the Markovian dynamics of the two-level system coupled to photons is based on the general solution presented in equation (18). After defining the Lindblad operator in line 15 (see source code), we must find the eigenmatrices and eigenvalues using the same function sortingEigenvalues() presented in appendix B. The value of the tolerance parameter TOL (line 1 of the above code) is chosen in order to have a sufficient numerical precision to discriminate the real and imaginary parts of the most similar eigenvalues \(\lambda_k\). In lines 30–33 of the source code, we calculated the density operator using the command

\[
\rho(t) = \rho_0 + c_k^* \exp(\lambda(k) \cdot t(n)) \cdot R_k, \quad c_k = \text{trace}(\rho_0 \cdot L_k), \quad \lambda(k) \text{ are the eigenvalues, } R_k \text{ are the right eigenmatrices, and } \rho_0 \text{ is the initial density matrix.}
\]

At zero temperature \((N_{ph} = 0)\), we have the following exact solutions:

\[
p_e(t) = \frac{\Omega^2}{\gamma_0^2 + 2\Omega^2} \left[ 1 - e^{-\frac{3\gamma_0 t}{4\mu}} \left( \cos \mu t + \frac{3\gamma_0}{4\mu} \sin \mu t \right) \right],
\]

(22)

\[
\langle \hat{\sigma} + (t) \rangle = \frac{-i\Omega \gamma_0}{\gamma_0^2 + 2\Omega^2} \left[ 1 - e^{-\frac{3\gamma_0 t}{4\mu}} \left( \cos \mu t + \left( \frac{3\gamma_0}{4\mu} \cos \mu t \right) \sin \mu t \right) \right],
\]

(23)

where \(\mu = \sqrt{\Omega^2 - (\gamma/4)^2}\). In figure 6 we plotted the population \(p_e(t) = \langle e|\rho(t)|e \rangle\) and the imaginary part of \(\langle \hat{\sigma} + \rangle\) for \(N_{ph} = 0\). We define the above functions, and we plot both exact and numerical solutions:

We observed good agreement between the numerical and the exact solutions. Beyond the time evolution, the steady state of the system is a very useful piece of information. For example, when the system reaches the steady state \((t \gg \gamma_0^{-1})\), the excited state and coherence can be...
Figure 6. (a) Population of the excited state $p_e(t) = \langle e | \rho(t) | e \rangle$ for a two-level system interacting with a photon reservoir at zero temperature. In both curves we use $N = 0$, $\gamma_0 = 0.2\Omega$ and $\Omega = 1$. (b) Imaginary part of the observable $\langle \hat{\sigma}_+ \rangle$ as a function of time. In both plots the red (solid) and blue (dashed) lines correspond to the numerical and exact calculations, respectively.

found: $\rho_{ee}^{ss} = \Omega^2 / (\gamma_0^2 + 2\Omega^2)$ and $\rho_{eg}^{ss} = i\Omega\gamma_0 / (\gamma_0^2 + 2\Omega^2)$. In our case, $\Omega = 1$ and $\gamma_0 = 0.2\Gamma$ resulting in $\rho_{ee}^{ss} = 0.4902$ and $\rho_{eg}^{ss} = 0.0980i$. The numerical steady state is defined as

$$\rho_{ss}^{num} = c_1 R_1,$$  \hspace{1cm} (24)

This solution corresponds to the particular case in which $\dot{\rho} = 0$. By looking our numerical simulations we obtain

$$\rho_{ss}^{num} = \begin{pmatrix} 0.4902 & 0.0980i \\ -0.0980i & 0.5098 \end{pmatrix},$$  \hspace{1cm} (25)

which exactly matches the theoretical predictions. Further extensions of this code to multiple level systems can be realized by changing the Hamiltonian and the dissipative contributions.

3.3. Time-local quantum master equation

In the literature, the time-local quantum master equation in the secular approximation is presented as \[7\]

$$\frac{d\rho}{dt} = \mathcal{L}_{NM}\rho(t) = -i[H_s, \rho(t)] + \sum_{i=1}^{N_c} \gamma_i(t) \left[ \hat{L}_i \rho(t) \hat{L}_i^\dagger - \frac{1}{2} \{ \hat{L}_i^\dagger \hat{L}_i, \rho(t) \} \right],$$  \hspace{1cm} (26)

where $\gamma_i(t)$ are the time-dependent rates associated with the operators $\hat{L}_i$. These rates can be obtained using a microscopic derivation ruled by the Hamiltonian describing the system–reservoir interaction \[6\]. In order to numerically solve this type of equation we adopt a different approach with respect to the Markovian case. To solve the non-Markovian dynamics we employed a predictor corrector integrator method \[16, 27\]. In the next section, we illustrate the main ideas by solving the pure-dephasing dynamics of the spin-boson model.
3.4. Pure-dephasing model and non-Markovianity

We consider the Hamiltonian for the pure dephasing spin-boson model ($\hbar = 1$):

$$H = \frac{\omega_{eg}}{2} \sigma_z + \sum_k \omega_k \hat{b}_k^\dagger \hat{b}_k + \frac{\tilde{\sigma}}{2} \sum_k \left( g_k \hat{b}_k + g_k^\ast \hat{b}_k^\dagger \right),$$  

(27)

where $\omega_{eg}$ is the bare frequency of the two-level system and $\omega_k$ are the boson frequencies. The exact time-local master equation in the interaction picture is [26]

$$\frac{d\rho}{dt} = \frac{\gamma(t)}{2} \left[ \sigma_z \rho_s(t) \sigma_z^\dagger - \{ \sigma_z^\dagger \sigma_z, \rho(t) \} \right] = \frac{\gamma(t)}{2} \left[ \sigma_z \rho_s(t) \sigma_z - \rho(t) \right].$$  

(28)

The system-environment interaction is fully determined by the time-dependent dephasing rate ($\hbar = 1$):

$$\gamma(t) = \int_0^\infty J(\omega) \coth \frac{\omega}{2k_B T} \sin(\omega t) d\omega,$$  

(29)

where $J(\omega) = \sum_k |g_k|^2 \delta(\omega - \omega_k)$ is the spectral density function (SDF), $k_B$ is the Boltzmann constant and $T$ is the reservoir temperature. We solve the dynamics for the following SDFs:

$$J_1(\omega) = \alpha \omega^{1-s} e^{-\omega/\omega_c},$$  

(30)

$$J_2(\omega) = \frac{\Gamma/2}{\left( \frac{\omega}{\omega_0} + 1 \right)^2 \left( \omega - \omega_0 \right)^2 + \left( \Gamma/2 \right)^2},$$  

(31)

where $J_1(\omega)$ was originally introduced in the context of dissipative two-level systems [28]. Physically, $\alpha$ is the system-environment coupling strength, $s \geq 0$ is a parameter, and $\omega_c$ is the cut-off frequency. Usually, three cases are defined: (i) $0 < s < 1$ (sub-ohmic), $s = 1$ (ohmic), and $s > 1$ (super-ohmic). The spectral density function $J_2(\omega)$ comes from the dynamics of quantum dots [29] but also can describe localized phonons in colour centres in diamond [9].

We then coded the spectral density functions $J_1(\omega)$ and $J_2(\omega)$:
In figure 7(a) we plotted the SDFs $J_1(\omega)$ (red) and $J_2(\omega)$ (blue) given in equations (30) and (31) for $s = 2.5$ and $\omega_c = 0.1$. The super-ohmic function $J_1(\omega)$ reaches a maximum around $\omega \approx 0.25$ and quickly decreases due to the cut-off frequency term $\exp(-\omega/\omega_c)$. In contrast, the SDF $J_2(\omega)$ is strongly localized at the frequency $\omega_0 = 2$ with a full width at half maximum equal to $\Gamma = 0.1$. We solve the non-Markovian dynamics starting from the initial condition

$$\rho(0) = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix},$$

and the code reads as

```matlab
1 Nt = 5000; % Number of points for time
2 ti = 0; % Initial time
3 tf = 100; % Final time
4 dt = (tf-ti)/(Nt-1); % Step time dt
5 t = ti:dt:tf; % Time vector
6 Psi_0 = [1 1]/sqrt(2); % Initial wavefunction
7 rho_01 = Psi_0*Psi_0'; % Initial density matrix for J_1(\omega)
8 rho_02 = rho_01; % Initial density matrix for J_2(\omega)
9 p11 = zeros(size(t)); % Matrix element rho_11
10 p22 = zeros(size(t)); % Matrix element rho_22
11 p12 = zeros(size(t)); % Matrix element rho_12
12 p21 = zeros(size(t)); % Matrix element rho_21
```

The time-dependent rates $\gamma_1(t)$ ($J_1(\omega)$) and $\gamma_2(t)$ ($J_2(\omega)$) can be numerically solved using the following integration method:

```matlab
1 wa = ones(size(t))*w; % Auxiliary frequency vector
2 Jw = ones(size(t))*J1; % Auxiliary J_1 vector
3 Jw = ones(size(t))*J2; % Auxiliary J_2 vector
4 t = t*ones(size(w)); % Auxiliary time vector
5 T = 0.001*w; % Temperature
6 gamma1 = sum(Jw.*sin(wa.*t).*coth(wa/T/2)),ddw; % Rate gamma_1(t)
7 gamma_1_t = alpha*w.*gamma1.*sin(satan(\omega_c*t))./(1-(\omega_c*t).^2)./(\omega_c*t); % Rate gamma_1(t)
8 gamma2 = sum(Jw.*sin(wa.*t).*coth(wa/T/2)),ddw; % Rate gamma_2(t)
```

For comparison, we introduced the exact solution [32]

$$\gamma_1^{\text{true}}(t) = \alpha \omega_c \gamma(s) \frac{\sin[\text{satan}(\omega_c t)]}{[1 + (\omega_c t)^2]^{3/2}}.$$ (33)
Figure 7. (a) Spectral density functions $J_1(\omega)$ and $J_2(\omega)$. For the spectral density functions we set the values $\alpha = 0.5$, $s = 2.5$, $\omega_\xi = 0.1$, $J_0 = 0.2$, $\omega_0 = 2$, and $\Gamma = 0.1$. (b) Time-dependent rates $\gamma_1(t)$ and $\gamma_2(t)$ associated with $J_1(\omega)$ and $J_2(\omega)$, respectively. The rates are calculated at $T = 10^{-3}\omega_0$ (low-temperature). (c) Coherence function $C_1(t)$ and $C_2(t)$ associated with $\gamma_1(t)$ and $\gamma_2(t)$, respectively. (d) Degree of non-Markovianity $N_\gamma(t) = (1/2)\int_0^t (|\gamma_\tau| - \gamma(t)) d\tau$.

The time-dependent rates $\gamma_1(t)$ and $\gamma_2(t)$ are illustrated in figure 7(b), where the black dashed curve is the theoretical result given in equation (33). These time-dependent rates are calculated at low temperature $T = 10^{-3}\omega_0$. The damped oscillations of $\gamma_2(t)$ are a consequence of the strong interaction with the localized mode $\omega_0$. In fact, the periods of the signal are approximately given by $T \approx 2\pi/\omega_0 \approx 3.14$. Meanwhile, the rate $\gamma_1(t)$ is positive in the time region $0 \leq t \leq 30.8$, while for $t > 30.8$ the curve asymptotically reaches a constant negative value. Thus, the negative region of $\gamma_1(t)$ is established by the condition $3 \tan^{-1}(\omega_\xi t) > \pi$ leading to the critical time $t_{\text{crit}} = \tan(\pi/s)/\omega_\xi \approx 30.8$, in good agreement with the numerical calculations.

In order to quantify the degree of NM we use the following measure [30, 31]:

$$N_\gamma(t) = \frac{1}{2} \int_0^t (|\gamma_\tau| - \gamma(t)) d\tau,$$

(34)

where $\gamma_\tau(t)$ is the canonical rate when the master equation is written in the form $\dot{\rho} = \gamma_\tau(t) [L_z, \rho(t)] + (1/2)\{L_z^2, \rho(t)\}$ with $\text{Tr}(L_z^2) = 1$ [33]. From equation (28) we recognize $L_z = \sigma_z/\sqrt{2}$ and therefore we have $\gamma_\tau(t) = \gamma(t)$. Finally, we calculate the NM measure introduced in equation (34) and the coherence $C(t) = \sum_{i\neq j} |\rho_{ij}(t)| = 2|\rho_{eq}(t)|$ [34]:
The coherence functions associated with $\rho_1(t)$ and $\rho_2(t)$ are shown in figure 7(c) for the initial condition $\rho(0) = |\Psi(0)\rangle \langle \Psi(0)|$ with $|\Psi(0)\rangle = (|e\rangle + |g\rangle)/\sqrt{2}$. The super-ohmic spectral density function $J_1(\omega)$ induces a monotonic decreasing behaviour in the coherence while the localized model introduces oscillations. These oscillations can be understood as a back-flow of quantum information between the system and the environment. Finally, the degree of NM $\gamma(t)$ is calculated and shown in figure 7(d). As expected, the localized model evidences a high degree of NM at any time in comparison with the super-ohmic model. This can be explained in terms of the fast oscillations observed in the rate $\gamma_2(t)$ and the coherence $C_2(t)$. Further extensions of this code can be easily completed by assuming different terms in the Lindbladian and modifying the model for the environment.
4. Concluding remarks

In summary, we have developed selected examples illustrating how to code many-body dynamics in relevant quantum systems using the high-level matrix calculations in MATLAB. We oriented our discussion and examples to the fields of quantum optics and condensed matter, and we expect that the codes shown here will be valuable for graduate students and researchers that are beginning to work in these fields. Moreover, the simplicity of the codes will allow the reader to extend them to other similar problems. We propose that these codes can be used as a starting toolkit for research projects involving closed and open quantum systems.

Acknowledgments

The authors would like to thank Fernando Crespo for the valuable codes in Python for comparison. A N acknowledges financial support from Universidad Mayor through the Postdoctoral fellowship. R C and D T acknowledge financial support from Fondecyt Iniciación Grant No. 11180143.

Appendix A. Quantum simulation of cavity QED arrays

In this appendix, we introduce the function QuantumSimulationCavityArray() used to solve the many-body dynamics of the interacting cavities:

```matlab
1 function [DP,P1n]=QuantumSimulationCavityArray(w, D,g,Wphoton,L,a,Pw,Kex,Kexpp,t, Model)
2 HJC = hamiltonianBP(t=1); %Hamiltonian
3 for t=1:L
4 switch Model
5 case 'Jaynes-Cummings'
6 HJC = HJC + wc*A(i)^+*A(i) + (D+w)*a*p(i)+p(i)^+*g*(a(i)+a(i)^+)*a(i);
7 case 'Rabi'
8 HJC = HJC + wc*A(i)^+*A(i) + (D+w)*a*p(i)+p(i)^+*g*(a(i)+a(i)^+)*a(i)^+;
9 end
10 end
11 % Total Hamiltonian
12 H = HJC + Hphpp;
13 dt = t(2)-t(1), % Step time dt
14 U = exp(-i*H*dt); % Time propagator with step dt
15 up = [1 0]; % Initial state
16 down = [0 1]; % Ground state for the two-level system
17 Fock = eye(Wphoton+1); % Fock states
18 phi_l,ph = cos(theta)*exp(i*theta*x)*sin(theta)*exp(-i*theta*x);
19 psi_0 = psi_0*exp(i*phi_l)*exp(-i*phi_0) % Initial state
20 phi_1,ph = cos(theta)*exp(i*theta*x)*sin(theta)*exp(-i*theta*x);
21 psi_1,ph = psi_1*exp(i*phi_l)*exp(-i*phi_0) % Initial state
22 psi_1,ph = norm(psi_1,phi_1); % Many body wavefunction
23 for k=1:L-1
24 psi_1,ph = norm(psi_1,phi_1); % Many body wavefunction
25 end
26 psi_1,ph = norm(psi_1,phi_1); % Many body wavefunction
27 if x=1
28 PSI = PSI*N; % Initial wavefunction
29 end
30 PSI = PSI*N; % Initial wavefunction
31 end
32 d1 = 0;
33 for i=1:L
34 d1 = d1 + PSI(i)*psi_0(i)^*psi_0(i)*psi_0(i)^*psi_0(i)^*psi_0(i)^*psi_0(i)^;
35 end
36 PSI = abs(PSI*N)*PSI*N; % Ground state probability
37 end
38 end
39 DP = max(d1,T); % Order parameter
40 end
```

First, the function determines if the model corresponds to the Jaynes–Cummings or the Rabi Hamiltonian. In code line 13, we introduce the time propagator operator used to solve the...
dynamics. The initial many-body wavefunction is calculated in the for loop, see lines 21–23. Finally, the code computes the wavefunction at any time $t$, where the return probability to the Mott-insulator state $P_1(t) = |\langle \Psi(0)|\Psi(t)\rangle|^2$ and the order parameter OP (see equation (15)), are calculated as the outputs.

Appendix B. Sorting eigenvalues of the Markovian master equation

In this appendix, we introduce the function sortingEigenvalues() used to solve the Markovian dynamics presented in section 3.1. The function is defined as

```matlab
function [R_sort, L_sort, lambda_sort] = sortingEigenvalues(dim, TOL, 1)

[R, L] = sig(L); % Right eigenvectors and eigenvalues
diag(L); % Left eigenvectors and eigenvalues
diag(R); % Right eigenvalues written as a vector
diag(L); % Left eigenvalues written as a vector
int.R = sepval(dim*dim, 2);

count = 1;
for n = 1:dim
   % Sorting of eigenvalues
   x = sig(R(n));
   for m = 1:dim
      R(n, m) = abs(x(m));
      abs(x(m));
      [int.R(count, :) - m];
      count = count + 1;
   end
end

% Final sorting
R = sig(L);
L = L1; % Left eigenvalues
R = R1; % Right eigenvalues
L = L1; % Left eigenvalues
R = R1; % Right eigenvalues

% Sorting of eigenvalues
L = L1; % Left eigenvalues
R = R1; % Right eigenvalues

% Final R_sort and L_sort matrices
R_sort = cell(1, length(lambda_sort));
L_sort = cell(1, length(lambda_sort));

for n = 1:length(lambda_sort)
   % Normalized left eigenvectors
   BSort(k) = reshape(L1, k, dim, dim);
   R = B_sort(k);
   Ck = trace(L(k)*R);
   R = R/sqrt(Ck);
   % Normalized right eigenvectors
   L_sort(n) = L1;
   end
end
```

The above function returns the sorted eigenmatrices $R_k$, $L_k$ and eigenvalues $\lambda_k$ following a descending order for the real part of the eigenvalues: $0 = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$, where the $N$ eigenvalues are decomposed as $\lambda_k = \lambda_k^R + i \lambda_k^I$. The first eigenvalue $\lambda_1 = 0$ corresponds to the steady state of the system, where $L_1 = 1$ [5, 24]. In addition, the eigenvalues with $k > 1$ satisfy the condition $\lambda_k^R < 0$ leading to dissipation terms $\propto e^{\lambda_k^R t}$ in the general solution defined in equation (18). The eigenvalue with the largest negative real part ($\lambda_N^R$) defines the envelope $e^{\lambda_N^R t}$ of the experimental observables (see figure 1(b)). The normalized right ($R_k$) and left ($L_k$) eigenmatrices are calculated using the relations $R_k = R_k/sqrt(C_k)$ and $L_k = L_k/sqrt(C_k)$, where $C_k = trace(L_k*R_k)$ is the normalization factor. Using the normalized matrices $R_k$ and $L_k$ we can compute any physical observable.

**ORCID iDs**

Ariel Norambuena © https://orcid.org/0000-0001-9496-8765
References

[1] Jaynes E T and Cummings F W 1963 Comparison of quantum and semiclassical radiation theories with applications to the beam maser Proc. IEEE 51 89
[2] Schmidt S and Blatter G 2009 Strong coupling theory for the Jaynes–Cummings–Hubbard model Phys. Rev. Lett. 103 086403
[3] Porras D and Cirac J I 2004 Effective quantum spin systems with trapped ions Phys. Rev. Lett. 92 207901
[4] Aspelmeyer M, Kippenberg T J and Marquardt F 2014 Cavity optomechanics Rev. Mod. Phys. 86 1391
[5] Rose D C, Macieszczak K, Lesanovsky I and Garrahan J P 2016 Metastability in an open quantum Ising model Phys. Rev. E 94 052132
[6] Breuer H and Petruccione F 2002 The Theory of Open Quantum Systems (Oxford: Oxford University Press)
[7] Vega I and Alonso D 2017 Dynamics of non-Markovian open quantum systems Rev. Mod. Phys. 89 015001
[8] Filipov S and Chruścinski D 2018 Time deformations of master equations Phys. Rev. A 98 022123
[9] Norambuena A, Maze J R, Rabl P and Coto R 2020 Quantifying phonon-induced non-Markovianity in color centers Phys. Rev. A 101 022110
[10] Schmidt B and Lorenz U 2017 WavePacket: A Matlab package for numerical quantum dynamics. I: Closed quantum systems and discrete variable representations Comput. Phys. Commun. 213 223
[11] Schmidt B and Hartmann C 2018 WavePacket: A Matlab package for numerical quantum dynamics. II: Open quantum systems, optimal control, and model reduction Comput. Phys. Commun. 228 229
[12] Schmidt B, Klein R and Araujo L C 2019 WavePacket: A Matlab package for numerical quantum dynamics. III: Quantum-classical simulations and surface hopping trajectories J. Comput. Chem. 40 2677
[13] Cooper I 2012 Doing Physics with Matlab: Quantum Mechanics, Schrodinger Equation, Time Independent Bound States http://physics.usyd.edu.au/teach_res/mp/quantum/doc/schrodinger.pdf
[14] Chelikowsky R 2019 Introductory Quantum Mechanics with MATLAB: For Atoms, Molecules, Clusters and Nanocrystals (Weinheim: Wiley-VCH)
[15] Tan S M 2012 Quantum Optics Toolbox for MATLAB https://github.com/jevonlongdell/qotoolbox
[16] Korsch H J and Rapedius K 2016 Computations in quantum mechanics made easy Eur. J. Phys. 37 055410
[17] Shrödinger E 1926 An undulatory theory of the mechanics of atoms and molecules Phys. Rev. 28 1049
[18] Zunkovic B, Heyl M, Knap M and Silva A 2016 Dynamical quantum phase transitions in spin chains with long-range interactions: merging different concepts of non-equilibrium criticality Phys. Rev. Lett. 120 130601
[19] Halimeh J C and Zauner-Staubner V 2017 Dynamical phase diagram of spin chains with long-range interactions Phys. Rev. B 96 134427
[20] Jurcevic P, Shen H, Hauke P, Maier C, Brydges T, Hempel C, Lanyon B P, Heyl M, Blatt R and Roos F 2017 Direct observation of dynamical quantum phase transitions in an interacting many-body system Phys. Rev. Lett. 119 080501
[21] Moler C and Van Loan C 2003 Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later SIAM Rev. 45 3–49
[22] Figueroa J, Rogan J, Valdivia J A, Kiwi M, Romero G and Torres F 2018 Nucleation of superfluid-light domains in a quenched dynamics Sci. Rep. 8 15603
[23] Wang Q and Bernal F P 2019 Excited-state quantum phase transition and the quantum-speed-limit time Phys. Rev. A 100 022118
[24] Macieszczak K, Guţă M, Lesanovsky I and Garrahan J P 2016 Towards a theory of metastability in open quantum dynamics Phys. Rev. Lett. 116 240404
[25] Navarrete-Benlloch C 2015 Open systems dynamics: Simulating master equations in the computer (arXiv:1504.05266)
[26] Luczka J 1990 Spin in contact with thermostat: exact reduced dynamics Physica A 167 919
[27] Press W H, Teukolsky S A, Vetterling W T and Flannery B P 2007 Numerical Recipes 3rd edn (Cambridge: Cambridge University Press)
[28] Leggett A J, Chakravarty S, Dorsey A T, Fisher M P, Garg A and Zwerger W 1987 Dynamics of the dissipative two-state system Rev. Mod. Phys. 59 1
[29] Wilson-Rae I and Imamoğlu A 2002 Quantum dot cavity-QED in the presence of strong electron-phonon interactions Phys. Rev. B 65 235311
[30] Rivas A, Huelga S F and Plenio M B 2010 Entanglement and non-Markovianity of quantum evolutions Phys. Rev. Lett. 105 050403
[31] Rivas A, Huelga S F and Plenio M B 2014 Quantum non-Markovianity: characterization, quantification and detection Rep Prog. Phys. 77 094001
[32] Haikka P, Johnson T H and Maniscalco S 2013 Non-Markovianity of local dephasing channels and time-invariant discord Phys. Rev. A 87 010103(R)
[33] Hall M J, Cresser J D, Li L and Andersson E 2014 Canonical form of master equations and characterization of non-Markovianity Phys. Rev. A 89 042120
[34] Baumgratz T, Cramer M and Plenio M B 2014 Quantifying coherence Phys. Rev. Lett. 113 140401