Spin-Boson Quantum Simulation

User Documentation

1. Introduction

This code simulates the quantum dynamics of a hybrid light-matter system. It models two optical cavities (photon modes) coupled to a chain of interacting quantum spins. The simulation tracks how energy flows between light and matter over time.

2. File Structure

run.py - Main simulation script. Execute this file to run the simulation.

hamiltonians.py - Builds the Hamiltonian (energy operator) for the system.

states.py - Creates initial quantum states (photon and spin states).

evolution.py - Performs time evolution of the density matrix.

observables.py - Computes physical observables from the quantum state.

3. Requirements

Install the required Python packages:

```
pip install numpy scipy
```

4. How to Run

1. Place all Python files in the same directory.
2. Open a terminal and navigate to that directory.
3. Run the command:

```
python run.py
```

4. The simulation will print progress information and save results to files.

5. Parameters

System Size:

• n_max: Maximum number of photons per cavity (default: 2)
• N_spins: Number of spins in the chain (default: 8)

Photon Frequencies:

- omega1: Frequency of cavity mode 1 (default: 1.0)
- omega2: Frequency of cavity mode 2 (default: 1.5)

Coupling Strengths:

- g1: Coupling between cavity 1 and spins (default: 0.5)
- g2: Coupling between cavity 2 and spins (default: 0.5)

Spin Interactions (XXZ Model):

- J: Exchange coupling. J < 0 is ferromagnetic, J > 0 is antiferromagnetic (default: -1.0)
- delta: Anisotropy parameter. delta = 1 is Heisenberg, delta > 1 is Ising-like (default: 2.0)

Initial State:

- T_spin: Temperature for thermal spin state (default: 2.0)
- Photons start in Bell state: (|0,0■ + |1,1■)/√2

Time Evolution:

- t_list: Array of time points from 0.0 to 5.0 with step 0.05

6. Output Files

Density Matrices:

File: rho_evolved_Nspins{N}_nmax{n}_J{J}_delta{d}_T{T}.npz

Contains the full quantum state (density matrix) at each time step. This is a large file containing all the information about the system.

Observables:

File: observables_Nspins{N}_nmax{n}_J{J}_delta{d}_T{T}.npz

Contains:
• times: Array of time points
• n1_avg: Average photon number in cavity 1 at each time
• sz_spin0: Z-component of first spin at each time

7. Loading and Analyzing Results

To load the saved results in Python:

```
import numpy as np

# Load observables
data = np.load("observables_Nspins8_nmax2_J-1.0_delta2.0_T2.0.npz")
times = data['times']
n1 = data['n1_avg']
sz = data['sz_spin0']

# Print some values
print("Time points:", times[:5])
print("Photon number:", n1[:5])
print("Spin z:", sz[:5])
```

To plot the results:

```
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 4))

plt.subplot(1, 2, 1)
plt.plot(times, n1)
plt.xlabel('Time')
plt.ylabel('Photon number')
plt.title('Cavity 1 Occupation')

plt.subplot(1, 2, 2)
plt.plot(times, sz)
plt.xlabel('Time')
plt.ylabel('Spin z-component')
plt.title('First Spin')
```

```
plt.tight_layout()
plt.savefig('results.png')
plt.show()
```

8. Understanding the Physics

Photon Number ■n■■:

This measures the average number of photons in cavity 1. If this value changes over time, it means photons are being absorbed or emitted by the spins. Oscillations indicate coherent energy exchange between light and matter (Rabi oscillations).

Spin Z-Component ■σ■■■:

This measures whether the first spin points up (+1), down (-1), or is in a superposition (values between -1 and +1). Changes in this value show how the spin state evolves due to interactions with photons and neighboring spins.

9. Modifying the Simulation

Change Parameters:

Edit the parameter values at the top of run.py. Be careful with n_max and N_spins as increasing them significantly increases computation time and memory usage.

Different Initial Photon State:

To use a product state instead of Bell state, change:

```
from states import product_photons
rho_photon = product_photons(n_max, n1_init=1, n2_init=0)
```

Different Observable:

To measure a different spin or component:

```
# Measure x-component of spin 3
sx_spin3 = spin_at_site(rho, 3, 'x', n_max, N_spins)

# Measure second photon mode
n1, n2 = photon_numbers(rho, n_max, N_spins)
```

10. Troubleshooting

Simulation is too slow:

• Reduce N_spins (try 4 or 6 instead of 8)
• Reduce n_max (keep at 2)
• Increase time step (use 0.1 instead of 0.05)

Memory error:

• The Hilbert space dimension is $n\_max^2 \times 2^{N\_spins}$
• For N_spins=8, n_max=2: dimension = 1024
• For N_spins=10, n_max=3: dimension = 9216 (much larger!)

Warning about trace:

Small deviations (less than 1e-6) are normal numerical errors. If deviations are large, reduce the time step.

11. Notes

This code uses exact diagonalization which limits the system size. For larger systems, consider tensor network methods or other approximations. The simulation assumes closed system dynamics (no dissipation or decoherence).