Variational Steady State Simulation for Collective Dephasing

Tanmoy Pandit

Abstract

This repository implements a variational method to approximate the non-equilibrium steady state (NESS) of a dissipative quantum many-body system subject to collective dephasing. The method is based on the McLachlan variational principle, where the trace norm of the Lindblad time derivative $\mathrm{Tr}|\dot{\rho}|$ is minimized over a variational class of product states. The code is built using the `QuTiP` library for open quantum system simulation and uses `SciPy`'s optimization routines to perform parameter updates.

Motivation and Context

Understanding steady states in dissipative quantum systems is a central challenge in modern quantum optics, non-equilibrium statistical mechanics, and quantum information. Analytical solutions are rare, and brute-force numerical simulation of the full Liouvillian quickly becomes intractable for large systems. Variational methods offer a tractable alternative by projecting the complex dynamics onto a low-dimensional ansatz manifold.

In this work, inspired by the approach proposed by H. Weimer [PRL 114, 040402 (2015)], we variationally minimize the norm of the time derivative of the density matrix within a chosen variational family. This allows us to approximate the NESS without directly integrating the master equation.

Physical Model

We consider a minimal model of two qubits with collective dephasing, driven by a transverse field and coupled via a $Z \otimes Z$ Ising-type interaction. The total Hamiltonian is:

$$H = \frac{W}{2}(X_1 + X_2) + \frac{V}{4} Z_1 Z_2 + H_{\text{eff}},$$

where H_{eff} represents a mean-field correction derived from the current variational state to account for interaction feedback. The dissipation is modeled via a set of collapse operators:

$$L_k = \sigma_k^-, \quad \text{for } k = 1, 2,$$

which represent spontaneous emission or local amplitude damping.

Variational Ansatz and Cost Function

Each qubit state is parametrized on the Bloch sphere:

$$\rho_i(\vec{\alpha}_i) = \frac{1}{2}\left(\mathbb{I} + \alpha_{i,x} X + \alpha_{i,y} Y + \alpha_{i,z} Z\right),$$

ensuring positivity via the constraint $\|\vec{\alpha}_i\|^2 \leq 1$. The full two-qubit state is approximated as a tensor product:

$$\rho_{12} = \rho_1 \otimes \rho_2.$$

The variational cost function is:

$$\mathcal{C}[\vec{\alpha}] = \mathrm{Tr}|\dot{\rho}_{12}| = \mathrm{Tr}\left|-i[H, \rho_{12}] + \sum_k \left(L_k \rho_{12} L_k^\dagger - \frac{1}{2}\{L_k^\dagger L_k, \rho_{12}\}\right)\right|,$$

which is minimized using the Sequential Least Squares Quadratic Programming (SLSQP) method, subject to Bloch norm constraints.

Code Structure

- **Operator Initialization:** Pauli matrices and identity operators are defined for tensor product construction.
- **Cost Function `fun(alpha)`:** Builds the Hamiltonian and Lindbladian using the variational parameters, then computes the trace norm of $\dot{\rho}$.
- **Callback Tracking:** Stores optimization history of the cost function and variational parameters.
- **Optimization Routine:** Calls `scipy.optimize.minimize()` with inequality constraints to enforce physicality of ρ.
- **Plotting Utilities:** Plots the cost function convergence and evolution of the variational parameters.

Results and Interpretation

The notebook demonstrates:

1. Convergence of the cost function $\mathrm{Tr}|\dot{\rho}| \to 0$ as the system approaches the steady state.
2. Stabilization of the variational parameters $\vec{\alpha}$ indicating physical consistency.
3. Expected symmetry between qubits in the final ρ under uniform parameters and interactions.

This approach can be generalized to larger systems or more sophisticated variational manifolds (e.g., matrix product operators or neural-network states).

Reference

This code is based on the variational framework proposed in:

- H. Weimer, *Variational Principle for Steady States of Dissipative Quantum Many-Body Systems*, Phys. Rev. Lett. **114**, 040402 (2015).
  DOI: 10.1103/PhysRevLett.114.040402

Dependencies

- Python 3.7+
- `qutip` – Quantum Toolbox in Python
- `numpy` – Numerical arrays
- `scipy` – Optimization routines
- `matplotlib` – Plotting

Future Directions

Future extensions may include:

- Scaling the method to larger spin chains using product states or tensor networks.
- Introducing correlated variational ansatz with entanglement.
- Exploring different noise models (e.g., collective emission, phase damping).
- Benchmarking against exact diagonalization and MPDO simulations.