QFlux: Adaptive Variational Quantum Algorithms for Open Systems

Victor S Batista
Yale University, Department of Chemistry and Yale Quantum Institute

Part V: From exact dilation → variational, NISQ-friendly methods

- *Two variational workflows:*
  - *A vectorized **Unrestricted Adaptive Variational Quantum Dynamics** (UAVQD) method*
  - *A trajectory-based **Stochastic Schrödinger Equation** (SSE) method*

https://qflux.batistalab.com

JCTC_V.ipynb

QFlux: Adaptive Variational Quantum Algorithms for Open Systems

Victor S Batista
Yale University, Department of Chemistry and Yale Quantum Institute

Part V: From exact dilation → variational, NISQ-friendly methods

This tutorial is based on the manuscript

QFlux: Quantum Circuit Implementations for Molecular Dynamics

Part V: QMAD, A Module for Adaptive Variational Quantum Algorithms in Open Quantum Systems

Authors:

Saurabh Shivpuje, Xiaohan Dan, Yuchen Wang, Brandon C. Allen, Delmar G. A. Cabral, Alexander V. Soudackov, Zixuan Hu, Ningyi Lyu, Eitan Geva, Victor S. Batista, and Sabre Kais

Why Variational Methods for Open Systems?

- Open systems → non-unitary evolution (Lindblad)
- Quantum hardware → unitary gates only
- Exact dilation doubles qubits + depth
- **Variational circuits:** shallow, flexible, hardware-aware

Two Complementary Variational Strategies

- UAVQD: vectorize ρ → effective Schrödinger equation
- SSE: unravel Lindblad into quantum trajectories
- Trade-off: qubits vs stochastic averaging

Classical Baseline: Lindblad Benchmarks

Two benchmarks

- ODE solvers for the Lindblad master equation
- QuTiP reference solver (mesolve)
- Two benchmarks:
  - **Amplitude damping**
  - **FMO complex**

Amplitude damping can be solved analytically

Fenna–Matthews–Olson (FMO) complex: chemically realistic energy-transfer model

Amplitude Damping: Physical Intuition

- Spontaneous emission model
- Jump operator: $\sigma^- = |0\rangle\langle 1|$ (quantum optics convention)
- Exponential decay of excited population
- Coherences decay at half the rate

Vectorizing the Lindblad Equation (UAVQD)

Vectorize the Density Matrix
Matricize the Lindbladian superoperator

- Flatten $\rho \rightarrow |v_\rho\rangle$
- Effective Hamiltonian: $H_{eff} = H_e - i\, H_a$
- Schrödinger-like evolution
- Doubles qubit count

The price: the Hilbert space dimension squares, so the qubit count doubles

Variational Ansatz for Open-System Dynamics

- Trial state:

$$|\nu_\rho(t)\rangle \approx |\phi(t)\rangle = \prod_{l=1}^{k} \mathrm{e}^{-\mathrm{i}\theta_l(t)O_l} |\psi_i\rangle$$

- Parameters $\theta_l(t)$ updated at each timestep
- Operators O_l are chosen adaptively

McLachlan's Variational Principle

$$\delta \left\| \frac{\partial |\phi(\boldsymbol{\theta}(t))\rangle}{\partial t} + \mathrm{i} H_{\mathrm{eff}} |\phi(\boldsymbol{\theta}(t))\rangle \right\|^2 = 0$$

- Minimize distance to exact evolution
- Leads to a linear system (Euler's equation): $\mathbf{M}(t)\dot{\boldsymbol{\theta}}(t) = \mathbf{V}(t)$

$$M_{kj}(t) = 2\,\mathrm{Re}\left[\frac{\partial \langle \phi(\theta(t))|}{\partial \theta_k(t)} \frac{\partial |\phi(\theta(t))\rangle}{\partial \theta_j(t)} + \langle \phi(t)| \frac{\partial |\phi(\theta(t))\rangle}{\partial \theta_k(t)} \langle \phi(t)| \frac{\partial |\phi(\theta(t))\rangle}{\partial \theta_j(t)} \right]$$

$$V_k(t) = 2\,\mathrm{Im}\left[\langle H_{\mathrm{eff}} \rangle \langle \phi(\theta(t))| \frac{\partial |\phi(\theta(t))\rangle}{\partial \theta_k(t)} + \frac{\partial \langle \phi(\theta(t))|}{\partial \theta_k(t)} H_{\mathrm{eff}} |\phi(t)\rangle \right]$$

Euler's equation is solved (classically) at each timestep

Adaptive Operator Selection

- Operator pool (`build_pool`): Pauli strings
- Append operator that most reduces error
- Unrestricted = no fixed threshold

Script S.5.1: JCTC_V.ipynb

```python
from itertools import combinations, product
from qflux.variational_methods.qmad.ansatz import PauliOperator

# Define Pauli matrices
sx = np.array([[0, 1], [1, 0]])
sy = np.array([[0, -1j], [1j, 0]])
sz = np.array([[1, 0], [0, 0]])

def build_pool(nqbit):
    pauliStr = ["sx", "sz", "sy"]
    res = []
    # Iterate over combinations of qubit indices and Pauli operators
    for order in range(1, 3):
        for idx in combinations(range(1, nqbit + 1), order):
            for op in product(pauliStr, repeat=order):
                res.append(PauliOperator(op, list(idx), 1, nqbit))
    return res
```

UAVQD Example: Amplitude Damping

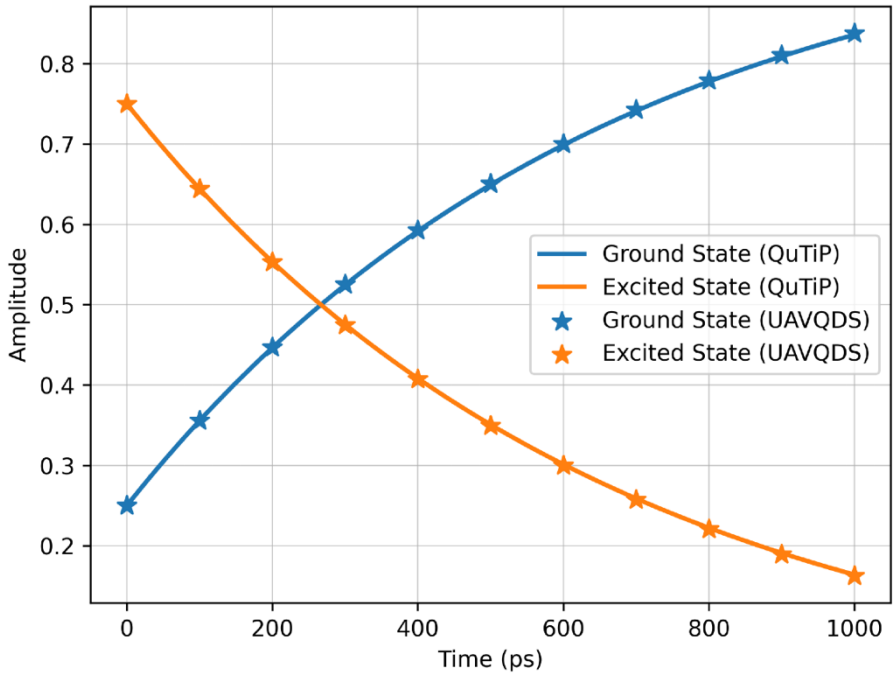
Pauli pool

Script S.6.1: JCTC_V.ipynb

- Pool: one- and two-qubit Pauli operators: $P_i, P_j \in \{X_i, Y_i, Z_i\}$

$$P_{\text{two}} = \mathrm{e}^{-\frac{\mathrm{i}\theta P_i \otimes P_j}{2}}$$

- Variational populations match numerically exact linear solver
- Shallow adaptive circuit



Why SSE Instead of Vectorization?

- UAVQD is elegant but enlarge Hilbert space: system + ancilla
- SSE keeps original Hilbert space
- Stochastic Lindblad integration = ensemble of trajectories

Stochastic Schrödinger Equation

Each trajectory evolves smoothly under an effective Hamiltonian until a random quantum jump occurs

$$\mathrm{d}|\psi_c(t)\rangle = \underbrace{\left[-\mathrm{i}H - \frac{1}{2}\sum_k \left(L_k^\dagger L_k - \langle L_k^\dagger L_k\rangle\right)\right]|\psi_c(t)\rangle\,\mathrm{d}t}_{\text{Deterministic drift}} + \underbrace{\sum_k \left[\frac{L_k|\psi_c(t)\rangle}{\langle L_k^\dagger L_k\rangle} - |\psi_c(t)\rangle\right]\mathrm{d}N_k}_{\text{Random jumps}}$$

- Jump probability $\propto \langle L^\dagger L\rangle\,\mathrm{d}t$: $\mathrm{d}p = \sum_{k=1}^{K} \frac{\langle\psi(t)|L_k^\dagger L_k|\psi(t)\rangle}{\langle\psi(t)|\psi(t)\rangle}\,\mathrm{d}t$
- If jump: apply an L_i: $\tilde{\psi}(t+\mathrm{d}t) = \frac{L_i\tilde{\psi}(t)}{\langle\tilde{\psi}(t)|L_i^\dagger L_i|\tilde{\psi}(t)\rangle}$ with $p_i = \frac{\langle\tilde{\psi}(t)|L_i^\dagger L_i|\tilde{\psi}(t)\rangle}{\sum_{k=1}^{K}\langle\tilde{\psi}(t)|L_k^\dagger L_k|\tilde{\psi}(t)\rangle}$
- Renormalize: $|\psi_j(t)\rangle = \frac{\tilde{\psi}_j(t)}{\langle\tilde{\psi}_j(t)|\tilde{\psi}_j(t)\rangle}$

Variational SSE Workflow

SSE combined with variational dynamics

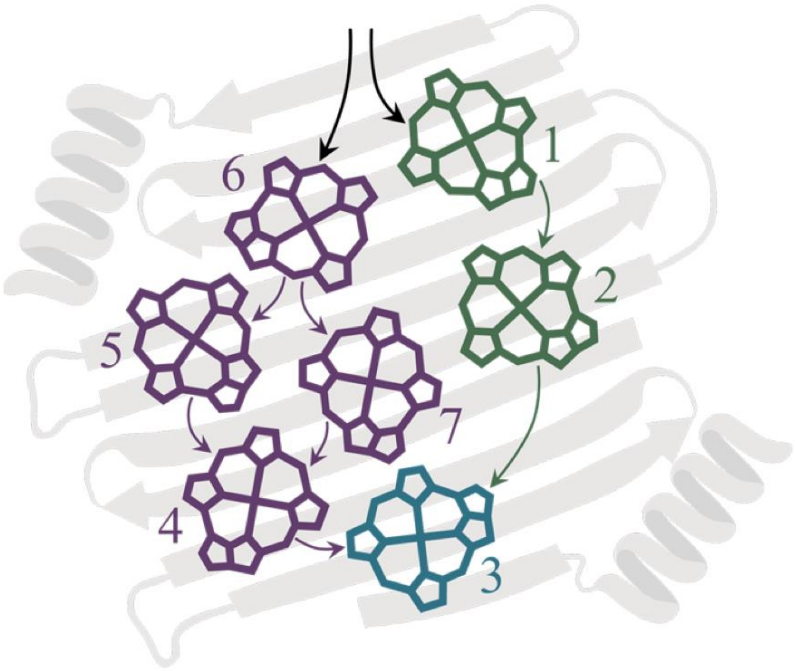
- Deterministic evolution via variational circuit
- On jump: reset ansatz parameters
- Repeat for many trajectories
- Average trajectories:

$$\rho(t) = \frac{1}{n}\sum_{j=1}^{n} |\psi_j(t)\rangle\langle\psi_j(t)|$$

FMO Complex: Physical Model

FMO complex models excitonic energy transfer in photosynthesis

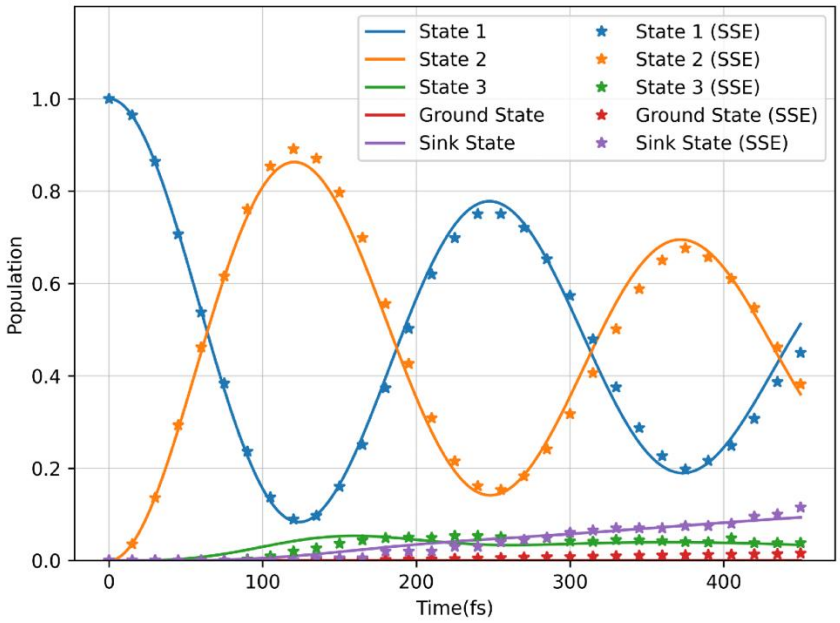
- 7 chromophores → reduced to 5 states
- Coherent couplings + dissipation
- Energy funnels to sink site



SSE Results for FMO

Example 1 — Spin-½ with Dissipation

- Excellent agreement with QuTiP
- Small errors from:
  - finite timestep
  - finite trajectories
  - truncated ansatz



Computational Trade-offs

Two complementary methods

- UAVQD:
  - deterministic
  - doubles qubits
- SSE:
  - stochastic
  - original qubits
  - many trajectories

What's Next

Memory Effects: Non-Markovian Quantum Simulations

- Non-Markovian extensions (Part VI)