Building a Software Toolchain for Quantum Networks

M. Goerz[1,2], N. Tezak[1], R. Hamerly[1], H. Mabuchi[1], R. Balu[2]

[1]Stanford University [2]Army Research Lab, Adelphi, MD

Army High Performance Computing Research Center

Goals

Build a collection of *open-source software packages* working together to enable a complete toolchain for the *description, analysis, and simulation of quantum networks*.
The toolchain enables the design and synthesis of novel computational devices operating at or near quantum energy scales. Different levels of abstraction and approximation allow for an efficient workflow appropriate to the particular system and application.

Applications

- Hybrid quantum networks for scalable quantum computing / communication
- Quantum memories with autonomous error correction
- Ultra-low power photonic circuits for classical logic and alternative computational schemes (analog, neuromorphic, distributed, probabilistic computation)
- Quantum and classical sensor networks

Toolchain

cirq (python)
Widget for circuit schematics
https://github.com/ntezak/cirq

gEDA

QNET (python)

QNET.jl (julia)
Julia interface to QNET

Design and analysis of photonic circuit models
- QHDL model
- SLH formalism
- symbolic quantum algebra
- circuit component library
- visualization
- drive numerical backends

https://github.com/mabuchilab/qnet

jupyter
sympy
matplotlib

QMANIFOLD (python)
- Symbolic tools for transforming open quantum system models to a moving basis parametrized by semi-classical coordinates.
- Numerical tools for performing quantum diffusion simulations.

QMANIFOLD.jl (julia)
Julia interface to QMANIFOLD

QTensors.jl (julia)
Application of sparse tensor product operators to quantum states

QuTiP

clusterjob (python)
Create and manage jobs on a HPC cluster
https://github.com/goerz/clusterjob

QSD (C++)
Moving basis solver for quantum trajectories:
- diffusive unravelings of the ME
- accumulate trajectories to obtain expectation values

https://github.com/mabuchilab/qsd-mpi

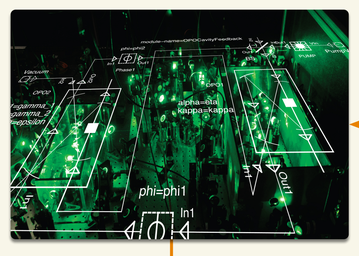
Wigner SDE
Semiclassical simulation of optical circuits

QHDLC (python) **QHDLJ** (julia)

CLQG/QHDLM
Finding optimal linear quantum feedback controllers to solve a coherent LQG problem

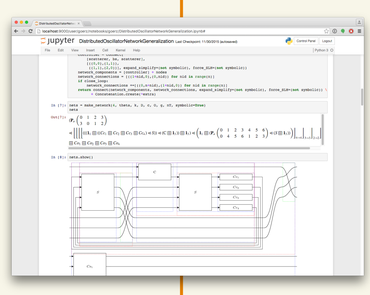
Description

- QHDL: describe physical circuit in terms of components and connections [1]
- SLH formalism: QSDE for network can be derived algebraically from components [2,3]
- ABCD parametrization of QSDE for linear quantum feedback networks
- Semi-classical Wigner-function-based SDE for non-linear coupled mode theory



Analysis

- Model reduction
  - adiabatic elimination
  - coherent manifold approach
- Steady-states, semi-classical fixpoints



Simulation

- Quantum Dynamics
  - Master equation (ensemble average)
  - Stochastic quantum trajectories (partial or full measurement of output channels) [7]
  - Coupled quantum-classical simulation through Manifold Tracking Simulation [4]

Simulation with HPC supercomputing resources

- Semi-Classical Wigner-SDE integration [5,6]: exploit localization in phase-space

Optimization

- Solution to quantum coherent LQG control problem. Used in [8]

References

[1] N. Tezak et al. Phil. Trans. R. Soc. A 370, 5270 (2012)
[2] J. Gough, M. James. Commun. Math. Phys. 287, 1109 (2009)
[3] J. Gough, M. James. IEEE Trans. Autom. Control 54, 2530 (2009)
[4] N. Tezak et al., *Quantum information geometry and localized quantum dynamics*. In preparation

References

[5] C. Santori et al. Phys. Rev. Appl. 1, 054005 (2015)
[6] R. Hamerly, H. Mabuchi. Phys. Rev. A 92, 023819 (2015)
[7] R. Schack, T. Brun, Comp. Phys. Comm. 102, 210 (1997)
[8] R. Hamerly, H. Mabuchi. Phys. Rev. Let. 109, 173602 (2012)