Open Quantum Systems Simulation for HEP

James Kowalkowski and Adam L. Lyon*
Fermilab Quantum Institute
August 27, 2020

Quantum simulation capabilities that will be available on quantum computing devices are useful throughout many areas of physics, including digital and analog simulations for quantum field theories, neutrino nucleus scatter studies, spin systems, improved QCD background predictions for collider experiments, non-abelian gauge symmetry, and even optimization, classification and combinatorics problems that appear in analysis. These devices promise to make these studies possible on reasonable timescales even for problems that scale exponentially. Eventually, these devices will allow direct observation of quantum processes within a controlled laboratory environment, or direct coupling of quantum sensors to computing for dark matter searches.

New quantum computing devices that may be suitable for executing HEP-relevant algorithms are emerging. One example is Fermilab's system using superconducting radio-frequency cavities. This multilevel "qudit" device will exhibit extremely long coherence times and could have a major impact on the ability to run complex algorithms. A co-design process for the design and construction of the devices as well as the design of the algorithms will ensure that HEP needs are met. An important component of this process is a simulation system run on classical computers offering fast turn-around time. Such a system allows for quick testing of ideas and strategies as well as understanding the impact of device and algorithm constraints. If organized well, this system will also provide a good entry point for scientists and students that want to engage in quantum projects without the complexity of interfacing with commercial or proprietary systems.

The type of simulation codes that are applicable to the problem are categorized as "Open Quantum System Simulations". Open indicates that the device's interactions with its environment and resulting noise/decoherence are taken into account (as opposed to a Closed system where the device is isolated from the environment). With such simulations, we are able to predict behavior of the device and system under real-world conditions. Open systems simulators will typically evolve the density matrix using a master equation in Lindblad form.

Desired functionality of an open quantum system simulator at the scale required for co-design are

- Sufficient size programs - good number qubits, qudits, states.
- Performant - quick turnaround time, taking advantage of HPC libraries and features (accelerators, large memory, etc.)

- Friendly interface - Need to be able to use the system as you would a real device from a programming environment quickly and efficiently from languages including C++, Python, Julia.
- Fits in with community standard circuit-building tools - Accepts input from standard compilers and standard instruction formats such as QASM[1].
- Can be easily driven from pulse descriptions and sequences generated from standard community tools
- Useful not only for execution of digital algorithms, but also of more direct quantum simulations or experiments that might be carried out.
- Library of device Hamiltonians and device profiles to quickly get started
- Usable from cloud services and HEP batch systems.

A further important application of an open quantum system simulator is as part of a quantum device's control system. In such a system, the simulation may be used to understand the impact of noisy controls on the device as well as testing the fidelity of gates resulting from their corresponding pulses.

The dominant open quantum system simulator and toolkit is QuTiP[2]. As a Python based system, it offers an easy-to-use interface and compatibility with user-friendly tools such as Jupyter Notebooks. However, the downside of its Python roots are lack of performance for simulations of the large scale that are important for device studies.

We have partnered with ANL for a QuantISED pilot grant to bring open systems simulation capabilities to HEP to study computational and experimental devices that are expected to appear in the near future. ANL brought QuaC[3] into the project as a starting point for moving towards a scalable open systems simulator that will fulfill HEP needs. QuaC is a C library that is built on top of PETSc, a mature HPC library of mathematical functions optimized for use at HPC facilities. We have used QuaC at a modest scale on multi-core nodes.

Through this grant and the Fermilab led DOE Office of Science Quantum Center center, our plan is to begin to evolve QuaC, or a similar classical open systems simulator, to have the feature set listed above. A complete toolkit will allow HEP to study algorithms and quantum simulations in new machine settings that will appear as Quantum Information Science matures. Our desire is to have community involvement in this project. QuaC is a starting point, but something better may come about through this effort.

* Primary contact email lyon_at_fnal.gov
[1] A. Cross et. al., "Open Quantum Assembly Language", arXiv:1707.03429
[2] J. R. Johansson, P. D. Nation, and F. Nori: "QuTiP 2: A Python framework for the dynamics of open quantum systems.", Comp. Phys. Comm. **184**, 1234 (2013) [DOI: 10.1016/j.cpc.2012.11.019] (http://qutip.org)
[3] https://github.com/0tt3r/QuaC