

Getting Ready in Algorithm and Software Development for the Arrival of the Quantum Computing Age (Letter of Interest to Snowmass21, Computational Frontier)

He Zhang*

Thomas Jefferson National Accelerator Facility, Newport News, VA, USA, 23606

Ji Qiang, Chad Mitchell

Lawrence Berkeley National Laboratory, Berkeley CA, USA, 94720

Ao Liu, Roman Kostin, Ben Freemire

Euclid Techlabs, LLC, Bolingbrook, IL, USA, 60440

I. Topic and status

Studies of collective effects are essential to the design and operation of modern accelerators that use high intensity beams. The particle-based numerical simulation of collective effects is usually composed of two parts: calculation of the interaction between the charged particles, and solving the dynamical equations to move the particles. The direct computation of particle interactions has a cost that scales with the square of the particle number, which is often the bottleneck when the particle number is large. Approximate algorithms [1, 2] are often employed to reduce the computational cost. Complex boundary conditions and charge distributions may further complicate the computation. The start-to-end simulation of an accelerator using realistic beams with billions or more particles remains challenging and expensive, even with state-of-the-art exascale machines. The development of the quantum computer provides new opportunities to enhance our simulation capability. Doing computations on a quantum computer requires different procedures compared with doing computations on a classical computer, and we should prepare ourselves from the perspective of algorithm and software development.

A quantum computer is a device that utilizes special properties of quantum mechanics to perform computations, which potentially can provide exponential improvement in the computational efficiency. The qubit, the quantum bit, is the fundamental unit of information in a quantum computer. Due to quantum superposition, the number of states of a quantum system with n qubits goes up to 2^n . Hence, the information stored in a quantum system scales exponentially with the number of qubits, while it scales linearly with the number of bits in a classical system. This is one important reason that quantum computing can provide higher efficiency than classical computers on some problems. Now both commercial companies and research institutes are pushing forward the techniques of building quantum computers, and the state-of-art quantum computer has above 50 qubits [3]. Currently, quantum computing is available to the public through cloud services provided by some commercial companies such as IBM [4], D-Wave [5], Amazon [6] and Microsoft [7]. Meanwhile, in academia, studies on quantum algorithm design is also booming in the past few years. The work most related to our topic is to solve linear systems [8–12] and to solve ordinary or partial differential equations [13–17] with quantum algorithms. Some papers [18, 19] discuss specifically how to solve Poisson’s equation, which is often used to model the electrostatic interaction in the beam frame when the relative speeds between the particles are low. These works provide us with the fundamental blocks to build our own simulation tools.

II. Current and future challenges

Although many papers on quantum algorithms have been published, few of them provide numerical examples besides the algorithm study, and we have not seen a real scientific problem in physics that has been solved by quantum computing. Clearly there is a gap between the development of these algorithms and their implementation. To our best knowledge, at this time there has been no implementation of quantum computing in beam dynamics simulations. We have to develop the simulation tools from scratch. We need to choose the accelerator physics problems that will benefit from quantum computing. For each problem, we need to first establish the mathematical model that is appropriate for quantum computing based on whether we can implement an existing quantum algorithm, then revise it or develop a new

*hezhang@jlab.org

one if needed.

In quantum computing, all the variables are stored in quantum states and the computation is carried out with quantum operators. Reading out the result is time-consuming. In a simulation with a large number of particles, outputting the coordinates of all particles in the 6D phase space is unpractical. It is necessary to develop new analysis tools which calculate the macroscopic parameters that describe the properties of the particle bunch, such as emittance, bunch length, phase space density functions, etc.

The quantum computer is not a replacement for the classical computer, and not all problems can benefit from quantum computing. From the user's point of view, at least in the near future, we must use a quantum computer together with a classical computer when addressing a real computational problem. So we should consider how to make our quantum simulation tools compatible with the best existing accelerator modeling programs. Ideally a protocol will be invented, through which the quantum packages could be called by the classical programs.

III. Advances needed to meet challenges

From the perspective of code development, a quantum computer simulator is needed as a code developing, debugging, and testing platform. The simulators are available for almost all mainstream programming languages, *e.g.* C/C++, Python, Java, Matlab, *etc.* There also exist some languages specially designed for quantum computing [20]. We should investigate these platforms and choose one that is free, suitable to run on cluster machines and convenient for developing and debugging. The simulators that provide an intrinsic access to cloud service quantum computing are preferred for the convenience to test run codes on real quantum computers.

From the perspectives of accelerator physics, quantum computing packages that perform simulations for selected accelerator problems should be developed, validated and benchmarked. Once enough quantum computing power is available, we will be able to implement these packages on real research problems. We would seek open source licenses for these packages and host them on GitHub for the convenience of developers and users.

Implementing quantum computing in a specific field needs active contributions from the field experts. It is hard to believe that all the development could be finished by only a small group of people. So we should build and enlarge a community of accelerator physicists who are familiar with quantum computing and have the enthusiasm to work together on this topic.

References

- [1] Vay, J.-L., Grote, D., Cohen, R., and Friedman, A., “Novel methods in the particle-in-cell accelerator code-framework warp,” *Computational Science & Discovery*, Vol. 5, No. 1, 2012, p. 014019.
- [2] Zhang, H., and Berz, M., “The fast multipole method in the differential algebra framework,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, Vol. 645, No. 1, 2011, pp. 338–344.
- [3] Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J. C., Barends, R., Biswas, R., Boixo, S., Brandao, F. G., Buell, D. A., et al., “Quantum supremacy using a programmable superconducting processor,” *Nature*, Vol. 574, No. 7779, 2019, pp. 505–510.
- [4] “IBM Quantum Experience,” <https://www.ibm.com/quantum-computing/technology/experience/>, 2020 (accessed August 27, 2020).
- [5] “Leap²,” <https://www.dwavesys.com/>, 2020 (accessed August 27, 2020).
- [6] “Amazon Quantum Solutions Lab,” <https://aws.amazon.com/quantum-solutions-lab/>, 2020 (accessed August 27, 2020).
- [7] “Azure Quantum,” <https://azure.microsoft.com/en-us/services/quantum/>, 2020 (accessed August 27, 2020).
- [8] Harrow, A. W., Hassidim, A., and Lloyd, S., “Quantum algorithm for linear systems of equations,” *Physical review letters*, Vol. 103, No. 15, 2009, p. 150502.
- [9] Clader, B. D., Jacobs, B. C., and Sprouse, C. R., “Preconditioned quantum linear system algorithm,” *Physical review letters*, Vol. 110, No. 25, 2013, p. 250504.
- [10] Childs, A. M., Kothari, R., and Somma, R. D., “Quantum algorithm for systems of linear equations with exponentially improved dependence on precision,” *SIAM Journal on Computing*, Vol. 46, No. 6, 2017, pp. 1920–1950.
- [11] Bravo-Prieto, C., LaRose, R., Cerezo, M., Subasi, Y., Cincio, L., and Coles, P. J., “Variational quantum linear solver: A hybrid algorithm for linear systems,” *arXiv preprint arXiv:1909.05820*, 2019.
- [12] Lee, Y., Joo, J., and Lee, S., “Hybrid quantum linear equation algorithm and its experimental test on ibm quantum experience,” *Scientific reports*, Vol. 9, No. 1, 2019, p. 4778.
- [13] Leyton, S. K., and Osborne, T. J., “A quantum algorithm to solve nonlinear differential equations,” *arXiv preprint arXiv:0812.4423*, 2008.
- [14] Berry, D. W., “High-order quantum algorithm for solving linear differential equations,” *Journal of Physics A: Mathematical and Theoretical*, Vol. 47, No. 10, 2014, p. 105301.
- [15] Arrazola, J. M., Kalajdzievski, T., Weedbrook, C., and Lloyd, S., “Quantum algorithm for nonhomogeneous linear partial differential equations,” *Physical Review A*, Vol. 100, No. 3, 2019, p. 032306.
- [16] Childs, A. M., Liu, J.-P., and Ostrander, A., “High-precision quantum algorithms for partial differential equations,” *arXiv preprint arXiv:2002.07868*, 2020.
- [17] Xin, T., Wei, S., Cui, J., Xiao, J., Arrazola, I., Lamata, L., Kong, X., Lu, D., Solano, E., and Long, G., “Quantum algorithm for solving linear differential equations: Theory and experiment,” *Physical Review A*, Vol. 101, No. 3, 2020, p. 032307.
- [18] Cao, Y., Papageorgiou, A., Petras, I., Traub, J., and Kais, S., “Quantum algorithm and circuit design solving the Poisson equation,” *New Journal of Physics*, Vol. 15, No. 1, 2013, p. 013021.
- [19] Wang, S., Wang, Z., Li, W., Fan, L., Wei, Z., and Gu, Y., “Quantum Fast Poisson Solver: the algorithm and modular circuit design,” *arXiv preprint arXiv:1910.09756*, 2019.
- [20] “Q# Development Kit,” <https://www.microsoft.com/en-us/quantum/development-kit>, 2020 (accessed August 27, 2020).