Simulation of Nonequilibrium Dynamics on a Quantum Computer

Henry Lamm[*] and Scott Lawrence[†]
Department of Physics, University of Maryland, College Park, Maryland 20742, USA

(Received 21 June 2018; revised manuscript received 6 September 2018; published 22 October 2018)

We present a hybrid quantum-classical algorithm for the time evolution of out-of-equilibrium thermal states. The method depends on classically computing a sparse approximation to the density matrix and, then, time-evolving each matrix element via the quantum computer. For this exploratory study, we investigate a time-dependent Ising model with five spins on the Rigetti Forest quantum virtual machine and a one spin system on the Rigetti 8Q-Agave quantum processor.

DOI: 10.1103/PhysRevLett.121.170501

Whether at the microscopic or the cosmological scale, a major challenge in physics is understanding the real-time evolution of nonequilibrium quantum systems. Classic examples of our limited knowledge in this area are hadronization of the quark-gluon plasma produced in heavy-ion collision and the expansion of the early Universe. While, in principle, these problems are amenable to numerical approaches on classical computers, the exponentially large state space of quantum systems coupled with the numerical sign problem in both fermionic systems [1] and real time [2] render such calculations intractable.

The promise of quantum computers is that the computational complexity of such problems can be reduced from exponential to polynomial. This potential improvement is twofold: one can represent the entanglement of quantum states directly and sign-problem free real-time calculations are possible. At present, we are restricted to fewer than 50 non-error-corrected qubits, which greatly restricts the class of problems we can attempt to simulate. Despite these present limitations, calculations in systems of interest in nuclear physics [3,4], quantum field theory [5], condensed matter [6], and quantum chemistry [7,8] have been achieved with as few as two qubits. Typically, these calculations have relied upon hybrid algorithms that couple a few-qubit quantum computer solving a problem of exponentially bad classical computational complexity to a larger classical computer.

In this paradigm, we present, in this Letter, the evolving density matrices on qubits (EρOQ) algorithm, a hybrid quantum-classical technique for computing nonequilibrium dynamics of many-body quantum systems. In particular, we show how to compute the density matrix of a Hamiltonian H_0, with inverse temperature β and, then, evolve this mixed state in real time by a different (potentially time-dependent) Hamiltonian H_1. The algorithm proceeds by computing, on a classical computer, a stochastic approximation to the density matrix $\rho = e^{-\beta H_0}$, via the density matrix quantum Monte Carlo algorithm [9]. This approximate density matrix is passed to a quantum computer element by element, which performs time evolution with a different Hamiltonian H_1, and then computes observables with the time-evolved density matrix $\rho(t) = e^{-iH_1 t}\rho e^{iH_1 t}$.

Past theoretical work on computing thermal physics with a quantum computer has focused on performing the thermal-state preparation on the quantum processor [10,11]. EρOQ differs from these approaches in allowing the computation of the thermal state to remain on the classical computer, using the quantum processor only for the classically intractable time evolution. With this division, the problem of evolving a mixed state on a quantum computer is reduced to the problem of evolving multiple pure states.

In this Letter, we implement our algorithm for a 1D Ising chain with $N \leq 5$ sites. The real-time evolution of this system has a long history of study on classical computers, starting with [12]. Since then, it has been used as a benchmark for developing time-dependent methods in quantum systems [13–16].

Below, we describe the hybrid quantum-classical algorithm EρOQ in full detail. Results using the Rigetti Forest, a quantum virtual machine (QVM) [17], and Rigetti's 8-qubit quantum processor (QPU) 8Q-Agave, are presented.

The first step of EρOQ produces a stochastic, sparse approximation to the density matrix using the density matrix quantum Monte Carlo algorithm (DMQMC) [9], which we briefly summarize here. The DMQMC is closely related to diffusion Monte Carlo methods [18], in which a population of imaginary particles called "psips" in Ref. [18]. explore the configuration space of a system through random walks in imaginary time $\beta = it$. Each psip is associated to a position basis state, and in the limit of large β, the density of psips approximates the ground state wave function. In DMQMC, the psips explore the space of basis operators, and after evolution by a finite β, the density of psips approximates the density matrix at inverse temperature β.

The density matrix $\rho(\beta) = e^{-\beta H}$ may be defined as the solution to the symmetric Bloch equation

© 2018 American Physical Society

$$\frac{d\rho}{d\beta} = -\frac{1}{2}(H\rho + \rho H), \tag{1}$$

with the initial condition $\rho(0) = 1$. The symmetric formulation of the Bloch equation results in an algorithm that naturally preserves the Hermiticity of ρ, up to stochastic fluctuations. The DMQMC stochastically implements the first-order Euler difference approximation to Eq. (1), with the density matrix represented by the collection of psips. To each psip is associated a basis operator $|b_p\rangle\langle a_p|$ and a sign χ_p, determining the sign of the psip's contribution to the density matrix. The approximate density matrix $\tilde{\rho} \approx \rho$ is given by a sum over all psips: the contribution to the density matrix of each psip p is $\chi_p|b_p\rangle\langle a_p|$. Thus, $\tilde{\rho}$ is given by

$$\tilde{\rho} = \sum_p \chi_p |b_p\rangle\langle a_p|. \tag{2}$$

To simplify the communication between the quantum and classical parts of the algorithm, we choose the computational bases of the classical and quantum computers to be the same.

The evolution of Eq. (1) begins by randomly placing psips along the diagonal of the density matrix, all with the positive sign $\chi = 1$. This implements the desired initial condition for Eq. (1). The density matrix is then evolved in discrete steps of $\Delta\beta$, with $\beta/\Delta\beta$ steps taken. At each step, every psip p (which is associated with the $|b_p\rangle\langle a_p|$ term of $\tilde{\rho}$) performs four operations derived by considering a finite-time-step approximation to Eq. (1): (1) The psip may spawn a new psip on another site in the same column, $|c\rangle\langle a_p|$ where $c \neq b_p$, with probability $\frac{1}{2}|\langle c|H|b_p\rangle|\Delta\beta$. (2) Similarly, the psip may spawn a new psip onto another site in the same row, $|b_p\rangle\langle c|$ where $c \neq a_p$, with probability $\frac{1}{2}|\langle a_p|H|c\rangle|\Delta\beta$. (3) If $\langle a_p|H|a_p\rangle + \langle b_p|H|b_p\rangle > 0$, then the psip is removed from the simulation with probability $\frac{1}{2}|\langle a_p|H|a_p\rangle + \langle b_p|H|b_p\rangle|\Delta\beta$. (4) Alternatively, when $\langle a_p|H|a_p\rangle + \langle b_p|H|b_p\rangle < 0$, the psip is cloned, producing another psip on the same site. This occurs with probability $\frac{1}{2}|\langle a_p|H|a_p\rangle + \langle b_p|H|b_p\rangle|\Delta\beta$.

When the $\beta/\Delta\beta$ executions of these four steps have completed, the resulting collection of psips gives an approximation to $\rho(\beta)$ via Eq. (2). The efficiency of this algorithm is produced by the fact that $\tilde{\rho}$ may be very sparse, where the exact density matrix ρ is not. For an N-site system, the density matrix ρ has at least 2^N nonzero entries, and typically of order 2^{2N}, we expect sufficiently accurate expectation values to be obtainable with a population of psips which scales only polynomially with N. For a fixed N, expectation values computed with $\tilde{\rho}$ will converge to the exact answer as $P \to \infty$ like $1/\sqrt{P}$, where P is the number of psips used. We observed the convergence to the exact ρ for several different N.

With the approximate density matrix $\tilde{\rho}$ determined, time-dependent expectation values are evaluated on a quantum processor. A time-dependent expectation value is given by

$$\langle \mathcal{O}(t)\rangle = \frac{\mathrm{Tr}\mathcal{O}e^{-iH_1 t}\rho e^{iH_1 t}}{\mathrm{Tr}\rho}, \tag{3}$$

where H_1, the Hamiltonian used for time evolution, is distinct from the H_0 Hamiltonian used to define the density matrix. Substituting the Hermitized approximate density matrix $\rho \to \frac{1}{2}(\tilde{\rho} + \tilde{\rho}^\dagger)$, we see that the expectation value may be approximated by a sum over psips

$$\langle \mathcal{O}(t)\rangle \approx \frac{1}{\mathrm{Tr}\tilde{\rho}} \sum_p \mathrm{Tr}\left(\frac{1}{2}\mathcal{O}e^{-iH_1 t}[\chi_p|b_p\rangle\langle a_p| + \bar{\chi}_p|a_p\rangle\langle b_p|]e^{iH_1 t}\right). \tag{4}$$

From Eq. (4), it can be seen that the decomposition of the density matrix into psips allows one to time-evolve each psip independently as a pure state, avoiding the difficulty of constructing a mixed state on a quantum processor.

psips for which $a_p = b_p$ are termed "diagonal." Expectation values $\langle a_p|\mathcal{O}(t)|a_p\rangle$ of diagonal psips may be evaluated straightforwardly on a quantum computer because they can be represented easily as a pure state. In contrast, nondiagonal psips must be diagonalized before evaluation on a quantum processor. For real charges χ_p, a Hermitized psip is diagonal in the basis $|u_p\rangle = (1/\sqrt{2})[|a_p\rangle + |b_p\rangle]$, $|v_p\rangle = (1/\sqrt{2})[|a_p\rangle - |b_p\rangle]$. Such states are easily prepared on a quantum computer—we use a single ancillary qubit that is discarded after state preparation. Working in this basis (a different basis for each psip), the contribution to $\langle\mathcal{O}(t)\rangle$ of the nondiagonal psips becomes

$$\langle O(t)\rangle \approx \frac{1}{\mathrm{Tr}\tilde{\rho}} \sum_p [\langle u_p|e^{iH_1 t}Oe^{-iH_1 t}|u_p\rangle - \langle v_p|e^{iH_1 t}Oe^{-iH_1 t}|v_p\rangle]. \tag{5}$$

In this form, the expectation value is a sum of quantities, each amenable to computation with a quantum computer. For a given set of psips specifying $\tilde{\rho}$, a separate instance of a general program is run on the quantum processor for each psip. Each program contains the same code for time evolution and measurement, but a different sequence of operations for preparing the pure states. For nondiagonal psips, two programs must be executed, one for $|u_p\rangle$ and one for $|v_p\rangle$, while the diagonal psips require only one. Each program has the following steps: (1) Prepare the state $|u_p\rangle$ (or $|v_p\rangle$); (2) Time-evolve with H_1 for a fixed time t via Trotterization;

(3) Measure $\mathcal{O}$, and any other observables of interest simultaneously.

For nearly all Hamiltonians of physical interest, the diagonal basis of the Hamiltonian is not efficiently accessible, and the time-evolution operator $e^{iH_1 t}$ must be approximated by Trotterization. This is accomplished by decomposing the Hamiltonian into terms easily diagonalized: $H_1 = H_x + H_z$. The time-evolution operator is then $e^{iH_1 t} = (e^{iH_x \Delta t} e^{iH_z \Delta t})^{t/(\Delta t)} + O(\Delta t)$. In the case of Eq. (6), we Trotterize with $H_x = -\mu_x \sum_i \sigma_x^{(i)}$ and $H_z = -J_z \sum_{\langle ij \rangle} \sigma_z^{(i)} \sigma_z^{(j)} - \mu_z \sum_i \sigma_z^{(i)}$.

In this Letter, the observable of interest (transverse magnetization) may be measured by changing basis from the z to the x basis (a rotation of each qubit), and measuring all qubits simultaneously.

We will use the z basis as our computational basis. For a diagonal psip, the state $|a_p\rangle$ is prepared by beginning with $|00\cdots\rangle$, and performing a NOT gate on qubit i if the spin at site i is down in the z basis. For a nondiagonal psip, we introduce an ancillary qubit to be placed in the state $|0\rangle + e^{i\theta_p}|1\rangle$ (if preparing $|u_p\rangle$) or $|0\rangle - e^{i\theta_p}|1\rangle$ (if preparing $|v_p\rangle$. This is done by initializing the ancillary to $|0\rangle$, and applying a Hadamard gate followed by a phase rotation of either θ_p or $\theta_p + \pi$. With the ancillary qubit prepared, the ith qubit is flipped with a CNOT gate, conditional on the ancillary. Finally, the ancillary is disentangled from the rest of the system via further CNOT gates and discarded.

Once each psip has been evaluated by the quantum processor, the results are summed together (on the classical computer) via Eq. (5) to calculate the expectation value of the thermal state.

In additional to the classical polynomial scaling of this algorithm, each psip corresponds to one or two calculations on the quantum computer. Thus, the number of calculations required on the quantum computer is expected to be polynomial in N as well. Because of Trotterization, each calculation on the quantum computer requires $O(T/\Delta t)$ gates, with a constant of proportionality dependent on the Hamiltonian being simulated.

As a demonstration of the algorithm, we simulate a 1D time-dependent Ising spin chain with one coupling constant and two independent magnetic fields [12–16]. The general Hamiltonian for this class of system is

$$H(t) = -J_z(t) \sum_{\langle ij \rangle} \sigma_z^{(i)} \sigma_z^{(j)} - \mu_x(t) \sum_i \sigma_x^{(i)} - \mu_z(t) \sum_i \sigma_z^{(i)}, \quad (6)$$

where $J_z(t)$ is the coupling constant between the z axis aligned spin component of nearest neighbors, and $\mu_x(t)$ and $\mu_z(t)$ denote time-dependent magnetic fields aligned with the x and z axes, respectively. We take the spin chain to have periodic boundary conditions. In this Letter, we will work in units where the inverse temperature is $\beta = 1$, and restrict ourselves to a constant coupling $J_z(t) = 1$ and longitudinal magnetic field $\mu_z(t)$ which is 0 for the $N = 5$ system and 1 for the $N = 1$. The transverse magnetic field is permitted to be time-dependent.

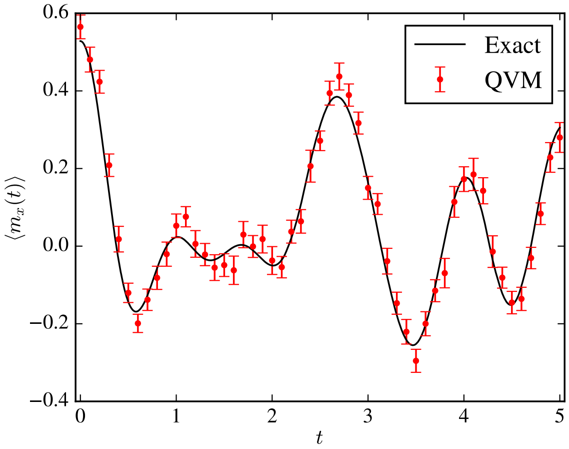

FIG. 1. The transverse magnetization $\langle m_x(t) \rangle$ for a $N = 5$ site spin chain with coupling $J_z = 1$, and an initial $\mu_x(0) = 1$ and $\beta = 1$, which is evolved with $\mu_x(t > 0) = -1$. Results from the Forest QVM are shown by red circles and the exact result is denoted by the solid black line.

The time-dependent observable we measure is the average transverse magnetization, given by

$$\langle m_x(t) \rangle \equiv \frac{1}{N} \sum_i \sigma_x^{(i)}(t). \quad (7)$$

As discussed in the previous section, this quantity is easily measured on the quantum processor.

For the purposes of this exploratory study, we compute $\langle m_x(t) \rangle$ for two cases: the $N = 5$ spin chain on the Rigetti Forest QVM to empirically test the algorithm's correctness, and the single-spin case on the Rigetti 8Q-Agave quantum computer to study the sources of uncertainty arising in a physical quantum processor.

Without the additional sources of error inherent in a QPU, we are able to access larger systems on the QVM. We evolve the $N = 5$ spin system with the Hamiltonian described by Eq. (6) with $\mu_x(t = 0) = 1$ and $\mu_x(t > 0) = -1$. The longitudinal magnetic field is $\mu_z = 0$. For this calculation we use a Trotterization time step of $\Delta t = 0.1$. The imaginary time step was $\Delta\beta = 0.04$ for evolving the psips with 5000 initial psips. Shown in Fig. 1 is $\langle m_x(t) \rangle$, in statistical agreement with the exact result.

When run on an ideal quantum processor, as simulated by Rigetti Forest, EρOQ has two sources of uncertainty, both statistical: the approximation of ρ by a finite number of psips, and the intrinsic measurement noise on the quantum processor. These sources of error are easily accounted for with standard methods such as bootstrapping as we do in this Letter. Note, though, that the errors are correlated since the same set of psips (i.e., the same approximation to the density matrix) is used for all values of t.

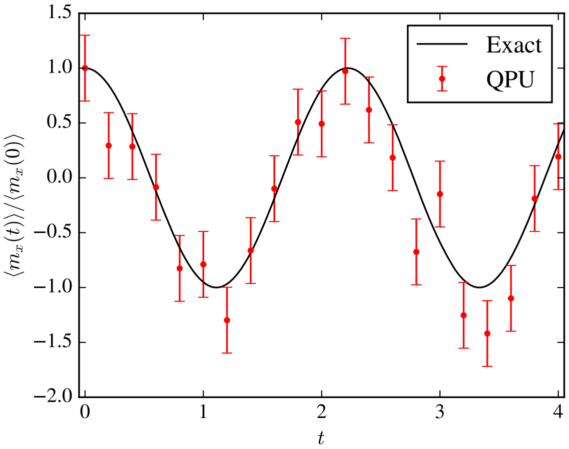

FIG. 2. The rescaled (see text) transverse magnetization $\langle m_x(t)\rangle/\langle m_x(0)\rangle$ for a single spin, with initial $\mu_x(0) = \mu_z(0) = 1$ and $\beta = 1.0$, which is evolved with $\mu_x(t > 0) = -1$. The results from Rigetti's 8Q-Agave QPU are shown in red circles while the exact result is denoted by the solid black line.

We use the eight-qubit quantum processor 8Q-Agave to simulate a single spin, thermalized in a transverse magnetic field $\mu_x(t = 0) = 1$, and time evolved in a flipped magnetic field $\mu_x(t = 0) = -1$. The longitudinal magnetic field is taken to be constant: $\mu_z = 1$. For this calculation we use a Trotterization time step of $\Delta t = 0.2$. The imaginary time step was $\Delta\beta = 0.04$, with 1000 initial psips. The results of this execution of the algorithm are presented in Fig. 2, again in good agreement with the exact result.

The physical 8Q-Agave, unlike the simulated Forest, is not an ideal quantum processor, and has several additional sources of error that must be accounted for. Most prominently, measurements have so-called readout noise. When measuring a qubit, there is some probability that the opposite state will be read, instead. If one assumes this readout noise is symmetric between the two states and independent of the gates used before a measurement is taken (empirically the case at our level of precision), this reduces the measured magnitude of $\langle m_x(t)\rangle$ by a constant factor, which can be corrected for by rescaling. In Fig. 2, we rescale $\langle m_x(t)\rangle$ by $\langle m(0)\rangle$, which appears to sufficiently remove the effect of readout noise.

Other sources of error, more difficult to correct for, are also present. For instance, when a parametrized gate (such as a one-qubit phase gate) is requested with angle θ, the actual gate implemented may have angle $\theta + \epsilon(\theta)$, producing a systematic bias in all results using that value of θ. This and other unanticipated sources of systematic error may be accounted for by performing a calibration run with a simpler Hamiltonian (diagonal in the computational basis). For this Letter, we use $H_1' = -\mu_z\sigma_z$: the error bars estimated for Fig. 2 are the quadrature average of the difference between the simulated results for H_1' and the exact answer.

In this work, we have presented EρOQ, a hybrid classical-quantum algorithm for simulating out-of-equilibrium dynamics of thermal quantum systems, applying it to a simple system on both a quantum virtual machine and a quantum processor. EρOQ first computes an approximation of the density matrix on a classical computer, evading the need to compute thermal physics or prepare a mixed state on a quantum computer. The density matrix is then passed to a quantum processor to compute the time evolution, thus, avoiding the sign problem associated with real-time calculations on a classical computer.

Going forward, this algorithm could be applied to problems of greater physical interest. While the hadronization of the quark-gluon plasma or reheating in the early Universe will require larger quantum processors than exist at present, any thermal quantum system for which a single pure state can be evolved on a quantum computer is accessible at no additional quantum computing complexity. The nonlinear response of low-dimensional systems like spin chains and graphene as well as the response of small nuclei to neutrino scattering [3,4] should be possible on near-future resources. In order to do this, a better characterization of the errors present on today's physical quantum computers will be necessary—a general concern for all quantum algorithms.

H. L. and S. L. are supported by the U.S. Department of Energy under Contract No. DE-FG02-93ER-40762. The authors would further like to thank Rigetti for their assistance and access to their resources, Forest and 8Q-Agave.

[*]hlamm@umd.edu
[†]srl@umd.edu

[1] M. Troyer and U.-J. Wiese, Phys. Rev. Lett. **94**, 170201 (2005).

[2] A. Alexandru, G. Basar, P. F. Bedaque, S. Vartak, and N. C. Warrington, Phys. Rev. Lett. **117**, 081602 (2016).

[3] E. F. Dumitrescu, A. J. McCaskey, G. Hagen, G. R. Jansen, T. D. Morris, T. Papenbrock, R. C. Pooser, D. J. Dean, and P. Lougovski, Phys. Rev. Lett. **120**, 210501 (2018).

[4] A. Roggero and J. Carlson, arXiv:1804.01505.

[5] N. Klco, E. F. Dumitrescu, A. J. McCaskey, T. D. Morris, R. C. Pooser, M. Sanz, E. Solano, P. Lougovski, and M. J. Savage, Phys. Rev. A **98**, 032331 (2018).

[6] A. Macridin, P. Spentzouris, J. Amundson, and R. Harnik, Phys. Rev. Lett. **121**, 110504 (2018).

[7] B. P. Lanyon, J. D. Whitfield, G. G. Gillett, M. E. Goggin, M. P. Almeida, I. Kassal, J. D. Biamonte, M. Mohseni, B. J. Powell, M. Barbieri *et al.*, Nat. Chem. **2**, 106 (2010).

[8] J. I. Colless, V. V. Ramasesh, D. Dahlen, M. S. Blok, M. E. Kimchi-Schwartz, J. R. McClean, J. Carter, W. A. de Jong, and I. Siddiqi, Phys. Rev. X **8**, 011021 (2018).

[9] N. S. Blunt, T. W. Rogers, J. S. Spencer, and W. M. C. Foulkes, Phys. Rev. B **89**, 245124 (2014).

[10] F. G. S. L. Brandao and M. J. Kastoryano, arXiv:1609.07877.

[11] E. Bilgin and S. Boixo, Phys. Rev. Lett. **105**, 170405 (2010).

[12] F. Carboni and P. M. Richards, Phys. Rev. **177**, 889 (1969).

[13] D. Gobert, C. Kollath, U. Schollwöck, and G. Schütz, Phys. Rev. E **71**, 036102 (2005).

[14] M. Žnidarič, T. Prosen, and P. Prelovšek, Phys. Rev. B **77**, 064426 (2008).

[15] S. R. White and A. E. Feiguin, Phys. Rev. Lett. **93**, 076401 (2004).

[16] A. J. Daley, C. Kollath, U. Schollwöck, and G. Vidal, J. Stat. Mech. (2004) P04005.

[17] R. S. Smith, M. J. Curtis, and W. J. Zeng, arXiv:1608.03355.

[18] J. B. Anderson, J. Chem. Phys. **65**, 4121 (1976).