Quantum Simulation of a Quantum Harmonic Oscillator

Joshua Peirce
Northeastern University Boston, MA

Abstract
This paper demonstrates how one can simulate a quantum harmonic oscillator on a quantum computer. A model is built of the quantum circuit on a classical computer and demonstrates that it works and gives some loose figures of accuracy. It simulates a quantum harmonic oscillator for half an oscillation, and achieve 99.9% overlap with the optimal wave function in the span of the computational basis. Lastly, some of the advantages and limitations of using quantum computing technologies to simulate physical systems in the future are discussed.

1 Introduction

Quantum computing promises to speed up the computing of numerous tasks, primarily a superpolynomial speedup to factoring, a quadratic speedup to unstructured search, as well as varying speedups to the simulation of physical systems. For the latter use, the speedup is due to the quantum computer's ability to simulate multiple differential equations in the same operation. For even the simplest quantum systems, those with only two states, due to the principle of superposition, for each additional quantum two-state particle (qubit), the number of differential equations doubles[2]. However, on sufficient quantum hardware, one can take advantage of this fact to simulate 2^n classical states using n qubits. The demonstration of this technique is the focus of this paper.

2 The Physical System

2.1 System parameters

The quantum harmonic oscillator (QHO) is a quantum system with a permanent restorative force proportional to the displacement of the object from equilibrium. In the position basis, its Hamiltonian is

$$H = \frac{\hat{p}^2}{2m} + \frac{1}{2}m\omega\hat{x}^2 \tag{1}$$

where $\hat{p}$ and $\hat{x}$ represent the position and momentum operators, respectively. According to the Schrödinger equation,

$$i\hbar\frac{d}{dt}\left|\psi\right\rangle = H\left|\psi\right\rangle \tag{2}$$

the Hamiltonian determines how any system will evolve in time, by describing a differential equation for its wave function $|\psi\rangle$. The Hamiltonian for the QHO has only two tunable parameters: m, the mass of the object in this system, and ω, the angular frequency in phase space. For this paper, the object is chosen to be an electron ($m = 9.109 \times 10^{-31} kg$), and the angular *frequency* is chosen to be $1\mu s^{-1}$, that is, $\omega = 2\pi\mu s^{-1}$.

2.2 State parameters

Now that the desired physical system is fully determined, one must choose the initial state and the final time to measure at. A coherent state is chosen as a convenient initial state, which can be created by displacing the ground state from equilibrium.[5]

The ground state of the QHO is given by

$$\langle x|\psi\rangle = \psi(x) = \left(\frac{m\omega}{\pi\hbar}\right)^{\frac{1}{4}} e^{-m\omega x^2/2\hbar}. \tag{3}$$

The initial displacement of the system is set such that the probability at the origin is 1/20th of the maximum probability, that is, $\psi(0)/\psi(d) = \sqrt{1/20}$, where d is the distance displaced from the origin. Using the ground-state wave function (3), this distance is given by $d = -\sqrt{\hbar \ln(20)/m\omega}$, where the negative signifies that the displacement is to the left. The system is measured after one-half oscillation, that is, at $t_f = \pi/\omega = 500ns$.

2.3 Theoretical Results

Given all this information, one can predict what the final result should be theoretically, a useful result that can be compared to the final output of my simulation. Because the system is measured after one-half oscillation, and the initial displacement is to the left, the theoretical result will be to the right. For a more rigorous derivation of this result, consider that the initial state is the coherent state $|\psi\rangle = e^{-|\alpha|^2/2} \sum_{n=0}^{\infty} \frac{\alpha^n}{n!} |n\rangle_E$, where $\alpha = \sqrt{m\omega/2\hbar}d$, and the ket in the sum is in the energy basis, not the position basis. The time evolution for coherent states is $|\alpha(t)\rangle = e^{-i\omega t/2} \left|\alpha e^{-i\omega t}\right\rangle$[5]. Since $\omega t_f = \pi$, note that this final state is $i\,|-\alpha\rangle$, which is the same (up to an overall phase) as the initial state if one takes $d \rightarrow -d$. The important result here is that $\langle x\rangle_f = -d$, as this serves as a comparison point for the output of the simulation.

3 The Simulation Quantum Circuit

3.1 Circuit parameters

Now for designing the quantum circuit that will simulate the quantum algorithm. For this paper, $n = 5$ qubits are used to represent the wave function in position space at each timestep. Using 5 qubits, the representation is capable of representing $N = 2^n = 32$ different positions. Position space provides a suitable basis to represent the system in, as the Schrödinger equation in position space is not overly complicated, and it is easy to measure the final position, so one can compare the simulation to the theoretical model. Two adjacent basis vectors in the computational basis correspond to adjacent positions in position space - as for the spacing between these positions, for the sake of clarity, $\Delta x = d/8$.

Lastly, the incremental timestep Δt is set to $100ns$, giving 5 timesteps from the start to the end of the simulation. in practice, many more timesteps would be used, and Δt would be much smaller; this number suffices for the sake of demonstration.

3.2 Initial state representation

One requirement of quantum simulation to be of any use is that one can efficiently prepare its initial state. For this specific case, since the initial state was already determined, calculating the digital representation takes $O(N)$ steps. In position space, the initial wavefunction is simply the ground state displaced from the origin by d:

$$\psi(x) = \left(\frac{m\omega}{\pi\hbar}\right)^{\frac{1}{4}} e^{-m\omega(x-d)^2/2\hbar}. \tag{4}$$

The 32 positions represented are centered at the origin, therefore the ith position is $x_i = (i - 15.5) * \Delta x$, assuming the position representation is 0-indexed. The corresponding amplitudes are then:

$$\tilde{\psi}_j = \left(\frac{m\omega}{\pi\hbar}\right)^{\frac{1}{4}} e^{-m\omega(((j-15.5)\Delta x)-d)^2/2\hbar}. \tag{5}$$

The digital representation of the initial state is given in Figure 1, where the computational basis is interpreted as a binary number.

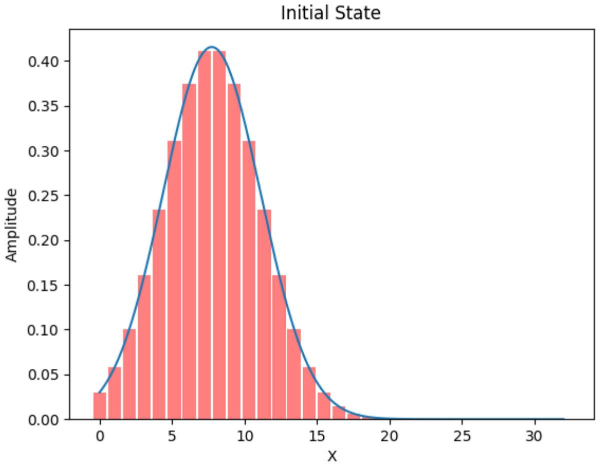

Figure 1: The amplitudes of each term in the superposition of the initial state.

Algorithms already exist for preparing a Gaussian initial state.[1]

3.3 Hamiltonian circuit

The desired unitary transformation to simulate is:

$$e^{iHt/\hbar} = e^{i(K(x)+V(x))t/\hbar} \tag{6}$$

where $K(x)$ represents the kinetic component of the energy and $V(x)$ represents the potential component. To do this, one can take advantage of the relation

$$\lim_{n\longrightarrow\infty} \left(e^{iK(x)t/\hbar n} e^{iV(x)t/\hbar n}\right)^n = e^{i(K(x)+V(x))t/\hbar}. \tag{7}$$

So provided one can efficiently simulate $K(x)$ and $V(x)$, one can chain them one after another in small steps to simulate the full Hamiltonian. The tradeoff is that equality only holds in the infinite limit; in the small-n limit, one gets the approximation:[2]

$$\begin{aligned} e^{i(A+B)\Delta t} &= e^{iA\Delta t/2} e^{iB\Delta t} e^{iA\Delta t/2} + O(\Delta t^3) && (8) \\ &\approx e^{iA\Delta t/2} e^{iB\Delta t} e^{iA\Delta t/2}. && (9) \end{aligned}$$

And some error is introduced here, but this is not an issue. Given the nature of quantum computing, there will always be some level of error. Indeed, the nature of physical simulation itself will also always produce errors, since all measurements have some error associated with them. Crucially, this error is bounded and doesn't grow too quickly, so the simulation is still useful in the end.

All that remains is to find a circuit for these Hamiltonians. The potential energy is diagonal in position space, making the operator simple to construct as one can compute each of its diagonal entries explicitly. This can be done on a classical computer in $O(N)$ time, as each entry is constant in time:

$$V_k = \gamma(k - 15.5)^2 \tag{10}$$

where we have let

$$\gamma = \frac{1}{2} m\omega^2 \Delta x^2 \Delta t/\hbar. \tag{11}$$

As for the kinetic energy operator, position space does not yield a convenient representation. However, position and momentum are a conjugate pair, so one can use a Fourier transform to switch to momentum space (the eigenbasis of the kinetic energy operator), apply the operator $e^{iK\Delta t/\hbar}$, and switch back to position space. Using a different γ, the computation in momentum space is of the same form as it was for position space.

3.4 Circuit schematic

Figure 2 demonstrates the step subroutine within the estimation algorithm. Efficient quantum circuits exist for creating the diagonal K and V operators.[3] A schematic of this simulating quantum circuit is shown in Figure 3. Note that this circuit is specific to the case where $t/\Delta t = 5$, and in general more gates are needed.

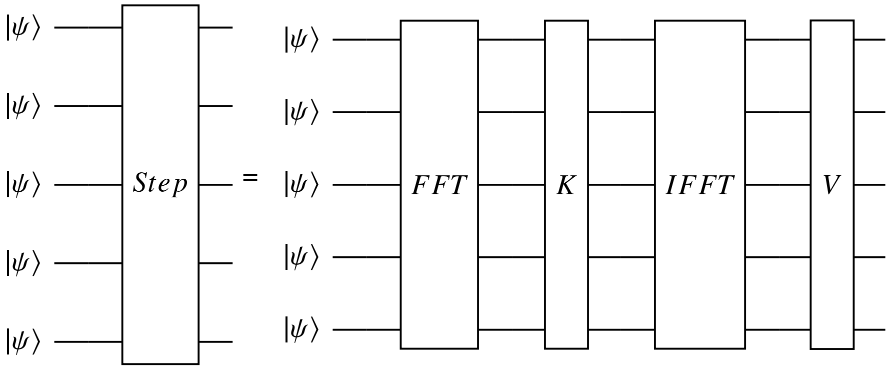

Figure 2: The subroutine for performing a single step. Note that all the operations can be implemented efficiently

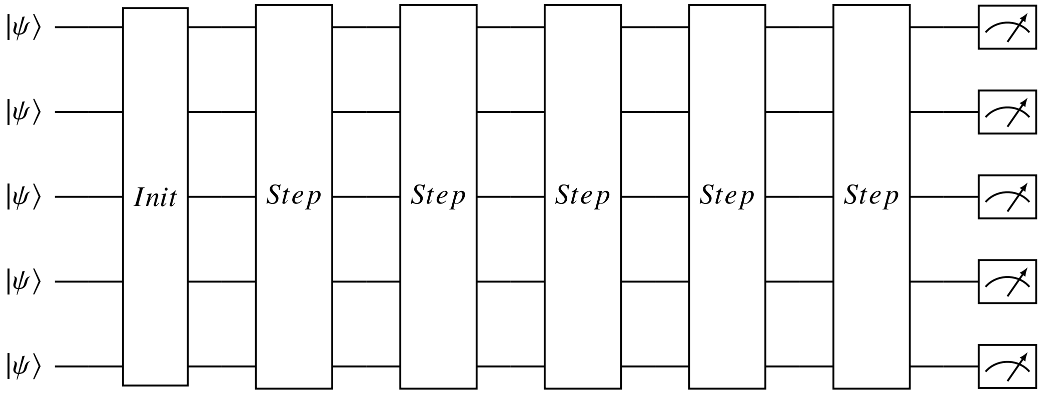

Figure 3: The larger quantum circuit that runs the simulation

4 Running the simulation

4.1 Python implementation

Due to a lack of access to a quantum computer, the quantum circuit described above was simulated using python. However, all the methods as described above would work theoretically on a quantum computer. All the results described below are derived from the results of the python simulations.

4.2 Gate errors

Currently, noisy intermediate-scale quantum (NISQ) computers have noise, that is, every application of a gate to the quantum system introduces some error. Currently available quantum hardware has a CNOT gate fidelity of $F = 0.981$[4]. The overall fidelity of the circuit multiplies with each additional gate, and the readout becomes unreliable when the overall fidelity falls below 50%. Thus, the maximum circuit depth on currently available quantum hardware that gives a useful result is $N = \frac{\log(0.5)}{\log(0.981)} = 36$. The depth of this circuit far exceeds that value, so without any form of error correction, this circuit would not yield useful results on current quantum hardware.

Critically, since the results of this paper are based on the python simulation of the quantum circuit, the results do not include any of these gate errors, which are not negligible. The results are still useful for future quantum hardware where the gate fidelity is high enough for effective error correction.

5 Results

5.1 Accuracy of the simulation

The theoretical physical wave function at the time of measurement should be a mirror of the initial one flipped over the y axis. Upon loading the state into the digital representation, some error was introduced. This error can be reduced by increasing the number of qubits in the circuit. There is also a theoretical ”closest” approximation $\left|\tilde{\psi}\right\rangle$ to the physical wave function that is representable by the digital representation, that is:

$$1-\epsilon = \left\langle \tilde{\psi} \middle| \psi(t_f) \right\rangle = Max_{\tilde{\psi}_i}\left(\left\langle \tilde{\psi}_i \middle| \psi(t_f) \right\rangle\right) \tag{12}$$

where ϵ is the ”error” between the best approximation in the digital space and the physical wave function. In reality it is highly unlikely that one could attain the closest digital approximation $\left|\tilde{\psi}\right\rangle$. To evaluate the algorithm’s accuracy, let the output of the simulating circuit be $|\phi\rangle$. Then the accuracy of the output is $1-\delta = \left\langle \phi \middle| \tilde{\psi} \right\rangle$, where δ represents the error from the inaccuracy of the algorithm. According to equation (8), this error term increases with the cube of Δt, so increasing the number of timesteps decreases δ.

The probability density functions are plotted in Figure 4. This circuit achieved an accuracy of $\delta = 0.001041$, meaning the circuit achieved a very high overlap with $\left|\tilde{\psi}\right\rangle$, the ideal output still in the span of the computational basis.

The expected value of the position according to this simulation was 1.0364 μm, and the theoretical expected value of position was 1.0422 μm.

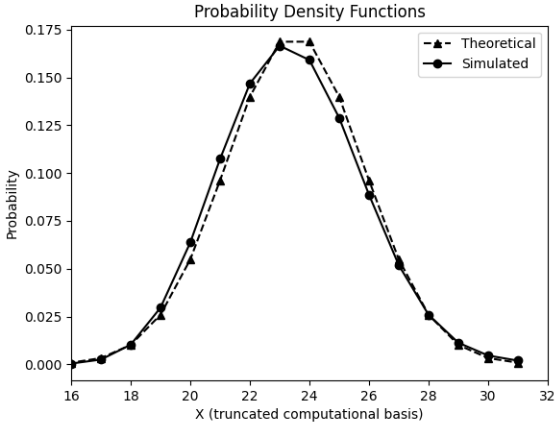

Figure 4: The theoretical and simulated probability density functions overlaid.

5.2 Uncertainties

As with all calculations in quantum computing, one must run the circuit many times to get some certainty about the result. For the final position of the particle, the uncertainty decreases with the square root of the number of iterations. Some values are shown in Table 1.

num iter.	$\sqrt{\Delta x^2}$ (μm)
1	3.106
10	0.9822
100	0.3106

Table 1: Uncertainty of position decreases with increasing number of iterations

Further, with additional measurements, the measurement statistics will approximate the output of this wavefunction. This is, in almost all applications, far more useful than a single number describing the expected value, as one can also gain knowledge about the spread of the object’s position, as well as the object's momentum. However, arbitrary precision is unnecessary with so few qubits representing position space. At some number of iterations, the uncertainty in the position becomes smaller than the step size of the digital representation of position space, i.e. $\sqrt{\Delta x^2} \leq dx$. This point can be calculated:

$$\frac{\sqrt{\Delta x_0^2}}{\sqrt{N}} \leq dx \rightarrow N \geq \frac{\Delta x_0^2}{dx^2} \tag{13}$$

where Δx_0^2 represents the variance of a single measurement. In this particular circuit, this happens at $N \geq 6$.

5.3 Time complexity

As mentioned above, the quantum circuit was simulated and ran in a python environment on a classical computer. This is where quantum advantage, and one day, quantum supremacy, becomes very clear. Figure 5 shows the amount of time taken by the classical computer to run the algorithm as a function of the number of qubits in the input.

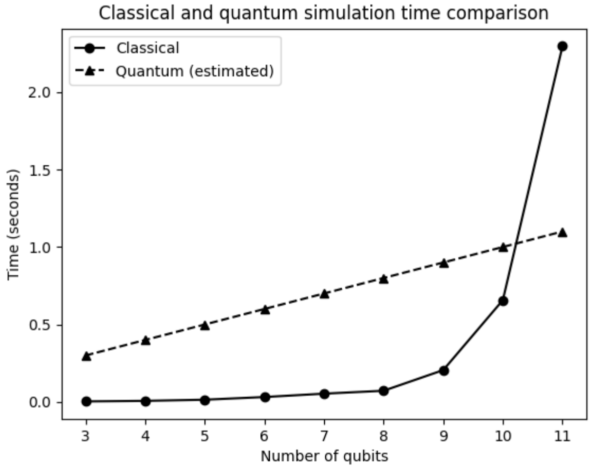

Figure 5: The time taken to complete the simulation (Note: there is no data behind the quantum computer estimates.)

As mentioned before, this circuit was not run on a quantum computer, and thus there are no real measurements of the runtime of the simulation on quantum hardware; the main point of Figure 3 is that the complexity of the simulation algorithm grows linearly on quantum hardware, whereas it grows exponentially on its classical counterparts. The exact numbers would vary wildly based on the specific hardware being used, but at some number of qubits, it would be faster to use a quantum computer than a classical one to simulate the same system.

The superpolynomial speedup is gained from the efficiency of the Hamiltonian circuit. This circuit contains only the quantum Fourier transform, and the application of the $K(x)$ and $V(x)$ operators, both of which are diagonal in the bases their applied in. Clearly, all of these operations are $O(n)$, where n represents the number of qubits present. On the other hand, in the classical case, both the FFT and the application of these (albeit diagonal) operators are linear in the number of entries *classically*, that is, $O(N)$, where $N = 2^n$.

The total runtime of the algorithm on quantum hardware is $O(n\frac{t_f}{\Delta t})$. As mentioned before, for each setup there would be an $O(N)$ step in creating the Hamiltonians H_0 and H_1, but that would only need to run once, whereas the main Hamiltonian circuit is run once per timestep, a large number of times.

Lastly, it is not easy in general to construct initial states - in fact, in general it takes an exponential number of gates to initialize the system into the initial state. In terms of complexity, state initialization as well as the construction of H_0 and H_1 put limits on the complexity. However, for systems for which these preparation steps can be done efficiently, the simulation runtime is exponentially faster than it would be on a classical computer.

6 Conclusions and future work

A quantum harmonic oscillator was simulated for half an oscillation on a simulated quantum computer. The final digital readout was 99.9% similar to the theoretically ideal readout, with minimal error. In addition, some of the advantages of simulating physical systems on quantum computers - in that they are far more space and time efficient in these simulations - were both demonstrated and discussed. In a future work, the physical simulation algorithm would be run on quantum hardware. Running quantum algorithms on quantum hardware over the cloud is already possible through services such as Qiskit[4]. This would both give some measurements as to the runtime of the algorithm and how it scales as the number of qubits grows, as well as better demonstrate the challengs of handling the measurement and control errors present in current quantum technologies.

References

[1] Alexei Kitaev and William A. Webb. Wavefunction preparation and resampling using a quantum computer. *Quantum Physics*, 2008.

[2] Michael Nielsen and Isaac Chuang. *Quantum Computation and Quantum Simulation*. Cambridge University Press, 10th anniversary edition, 2010.

[3] Jonathan Welch *et al.* Efficient quantum circuits for diagonal unitaries without ancillas. *New Journal of Physics*, 16, 2014.

[4] Thomas Alexander *et al.* Qiskit pulse: programming quantum computers through the cloud with pulses. *Quantum Science and Technology*, 5, 2020.

[5] John S. Townsend. *A Modern Approach to Quantum Mechanics*. University Science Books, 2nd edition, 2012.