QSite 2024 Classiq Open Challenge Project Proposal

Team Name: *Quantum Duckies*

September 28, 2024

1 Introduction

In "A Quantum Algorithm for Solving Linear Differential Equations: Theory and Experiment", [1] Tao Xin, et al. demonstrate the capability of quantum algorithms to provide exponential speedups over classical methods. We intend to use this paper's algorithm and Classiq's end-to-end quantum software platform to solve a harmonic oscillator equation with frequency $\omega = 1$:

$$y'' + \omega^2 y = 0, y(0) = 1, y'(0) = 1 \tag{1}$$

First, we solve it by hand to investigate expected outcomes:

$$\begin{gathered} y'' + \omega^2 y = 0 \text{ hence the characteristic equation is } r^2 + \omega^2 r^0 = 0 \\ = r^2 + \omega^2 = 0 \implies r = \pm i\omega \text{ (two imaginary roots)} \\ \implies y(t) = c_1 \cos(\omega t) + c_2 \sin(\omega t) \text{ is the general solution} \\ y(0) = c_1 \cos(0) + c_2 \sin(0) = 1 \implies c_1 = 1 \\ y'(0) = -c_1 \omega \sin(\omega t) + c_2 \omega \cos(\omega t) = 1 \implies c_2 = 1/\omega \end{gathered}$$

Therefore,

$$y(t) = \cos(\omega t) + \frac{1}{\omega} \sin(\omega t) \tag{2}$$

We observe that $y(t)$ describes the displacement of a system experiencing simple harmonic motion (SHM). Hence, we have the following relations for such a system:

$$E(t) = KE(t) + PE(t) \tag{3}$$

$$KE(t) = \frac{1}{2} m (y'(t))^2 \tag{4}$$

$$PE(t) = \frac{1}{2} m \omega^2 (y(t))^2 \tag{5}$$

Additionally, we can expand $KE(t)$ and $PE(t)$ in terms of y using Equation 2.

$$KE(t) = \frac{1}{2} m \left(\omega^2 \sin^2(\omega t) - 2\omega \sin(\omega t) \cos(\omega t) + \cos^2(\omega t) \right) \tag{6}$$

$$PE(t) = \frac{1}{2} m \omega^2 \left(\cos^2(\omega t) + 2 \frac{1}{\omega} \cos(\omega t) \sin(\omega t) + \frac{1}{\omega^2} \sin^2(\omega t) \right) \tag{7}$$

Using Equations 6 and 7 we can evaluate the kinetic and potential energy in the time interval $[0, 1]$, depending on the values of m and ω. Now that this Classical context has been established, we can turn to Quantum Methods.

2 Methods

Our solution takes the implementation steps from the paper: encoding, entanglement creation, decoding, and measurement. These steps outline the quantum circuit and algorithm used to solve the harmonic oscillator equation (refer to the diagram).

2.1 Quantum Algorithm Selection

The harmonic oscillator equation $\frac{d^2x}{dt^2} + \omega^2 x = 0$ was transformed into a first-order linear system, represented by:

$$\frac{d}{dt}\begin{pmatrix} x \\ v \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}\begin{pmatrix} x \\ v \end{pmatrix}$$

Here, the matrix $A = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$ is *unitary*, meaning the quantum algorithm designed to solve linear differential equations for unitary matrices, as described in the paper, was applied. This algorithm leverages quantum-controlled operations and ancilla qubits to simulate the system's time evolution.

2.2 Quantum Circuit Setup

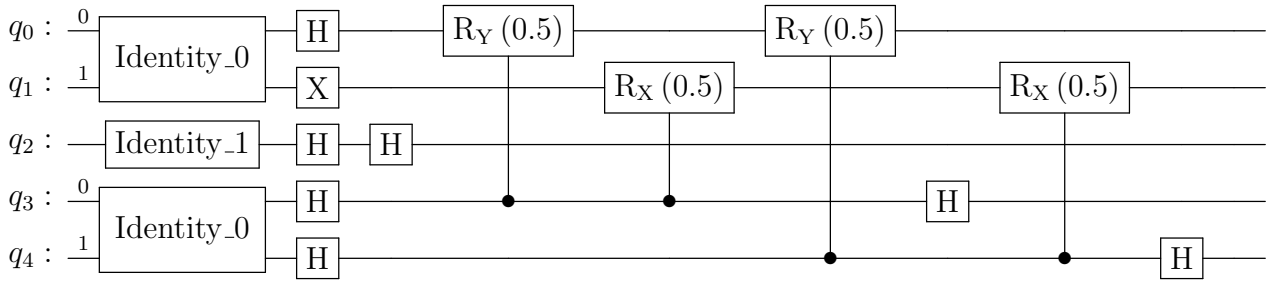

Figure 1: Our quantum circuit visualized

2.2.1 Qubit Allocation

The system was initialized with:

- **Work qubits**: Two qubits to represent the position $x(t)$ and velocity $v(t)$.
- **Ancilla registers**:
  - A first ancilla register with 1 qubit to help encode the initial state.
  - A second ancilla register with 2 qubits to store the time evolution step approximations.

```
allocate(2, q)          # Work qubits for position and velocity
allocate(1, ancilla_1)   # First ancilla register
allocate(2, ancilla_2)   # Second ancilla register for time steps
```

2.2.2 Initial State Encoding

To initialize the system, the quantum gates prepared the position and velocity qubits to their respective initial states $x(0)$ and $v(0)$. The first ancilla qubit was placed in a superposition state to encode the initial condition.

- $x(0) = 1$ and $v(0) = 0$:
  - A Hadamard gate was applied to the position qubit $q[0]$ to create a superposition.
  - A Pauli-X gate was applied to the velocity qubit $q[1]$ to set it to $|1\rangle$.

The ancilla qubits were initialized in superposition to facilitate entanglement during the evolution process.

```
H(q[0])  # Superposition for position x(0)
X(q[1])  # Velocity v(0) = 1
H(ancilla_1[0])  # Superposition in ancilla_1
H(ancilla_2[0]), H(ancilla_2[1])  # Superposition in ancilla_2
```

2.3 Entanglement and Controlled Operations

Controlled rotations were applied using the ancilla qubits to entangle the system. These operations encoded the time evolution of the harmonic oscillator system, using the evolution operator e^{At}.

For each time step (approximated by the second ancilla register), controlled CRY and CRX gates were used to rotate the work qubits conditionally on the ancilla qubits. This step created entanglement between the ancilla and work qubits, allowing the system to simulate the evolution over time.

```
for i in range(2):  # Second-order approximation
    CRY(0.5, ancilla_2[i], q[0])  # Controlled Y-rotation on position
    CRX(0.5, ancilla_2[i], q[1])  # Controlled X-rotation on velocity
```

2.4 Decoding

After entanglement, the system was decoded by reversing the initial encoding steps to project the system back into the subspace where all ancilla qubits were $|0\rangle$. This step ensures that the results reflect the correct time-evolved state of the system.

```
H(ancilla_2[0]), H(ancilla_2[1])  # Reverse the superposition in ancilla_2
H(ancilla_1[0])  # Reverse superposition in ancilla_1
```

2.5 Measurement

At the end of the quantum circuit, the work qubits (position and velocity) were measured to extract the final state of the system after time evolution. The ancilla qubits were also measured to verify that the system remained in the correct subspace.

```
measure(q[0])  # Measure position x(t)
measure(q[1])  # Measure velocity v(t)
measure(ancilla_1[0])
measure(ancilla_2[0]), measure(ancilla_2[1])
```

The measurement results provided the final states of the position and velocity qubits after the system evolved through the approximated time steps controlled by the ancilla qubits.

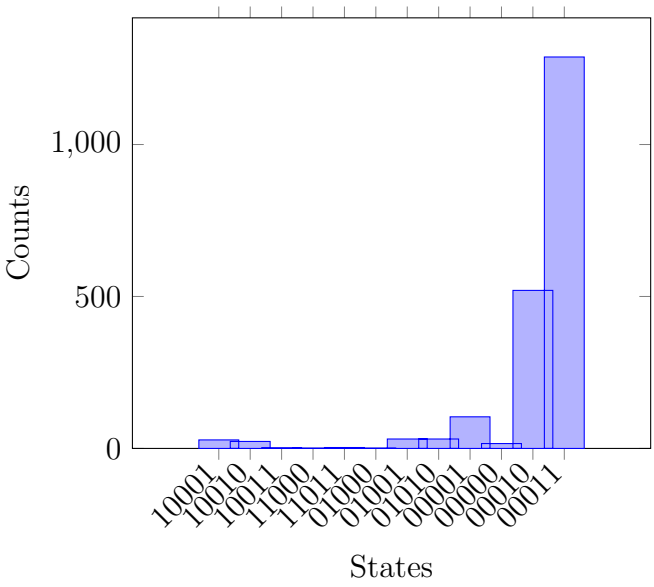

Figure 1: Bar graph from CSV data of the harmonic oscillator simulation results, where '10001' represents the quantum state for position, velocity, and ancilla registers.

2.6 Simulation Results and Interpretation

The quantum simulation yielded several states for the work qubits (position and velocity) and the ancilla qubits. The dominant results reflected the oscillatory nature of the harmonic oscillator, where both position and velocity alternated between high and low values as expected.

The following output was obtained after running the quantum circuit:

```
[{'q': [1, 1], 'ancilla_1': [0], 'ancilla_2': [0, 0]}: 1288,
{'q': [0, 1], 'ancilla_1': [0], 'ancilla_2': [0, 0]}: 520,
{'q': [1, 0], 'ancilla_1': [0], 'ancilla_2': [0, 0]}: 104,
{'q': [1, 0], 'ancilla_1': [0], 'ancilla_2': [1, 0]}: 31,
{'q': [0, 1], 'ancilla_1': [0], 'ancilla_2': [1, 0]}: 31,
{'q': [1, 0], 'ancilla_1': [0], 'ancilla_2': [0, 1]}: 28,
{'q': [0, 1], 'ancilla_1': [0], 'ancilla_2': [0, 1]}: 23,
{'q': [0, 0], 'ancilla_1': [0], 'ancilla_2': [0, 0]}: 16,
{'q': [1, 1], 'ancilla_1': [0], 'ancilla_2': [1, 1]}: 3,
{'q': [1, 1], 'ancilla_1': [0], 'ancilla_2': [0, 1]}: 2,
{'q': [0, 0], 'ancilla_1': [0], 'ancilla_2': [1, 1]}: 1,
{'q': [0, 0], 'ancilla_1': [0], 'ancilla_2': [1, 0]}: 1]
```

The values represent the various states of the position qubit $q[0]$, velocity qubit $q[1]$, and the ancilla qubits, with the number of occurrences for each state. These results correspond to different combinations of the position and velocity of the harmonic oscillator at various time steps.

The bar graph in Figure 2 visualizes the distribution of the measured quantum states for the position, velocity, and ancilla qubits. The most frequently observed states were:

- $q = [1, 1]$, ancilla_2 $= [0, 0]$, corresponding to a high displacement and velocity, with 1288 counts.
- $q = [0, 1]$, ancilla_2 $= [0, 0]$, indicating high velocity and low displacement, with 520 counts.

- $q = [1, 0]$, ancilla_2 $= [0, 0]$, indicating high displacement but low velocity, with 104 counts.

These results align with the expected behavior of the harmonic oscillator, where the position and velocity oscillate between high and low values.

Lower-frequency states, such as $q = [1, 1]$ with ancilla_2 $= [1, 1]$, occurred only a few times, reflecting less probable evolution paths in the system. The state distribution also shows that the ancilla qubits were often found in the $[0, 0]$ state, meaning that the system mostly evolved in the early stages of time approximation.

The results validate the success of the quantum algorithm in simulating the time evolution of the harmonic oscillator. The dominant states and the measurement projections into the subspace where ancilla qubits are $|0\rangle$ show that the algorithm correctly simulates the periodic behavior of the oscillator.

2.7 Kinetic and Potential Energy Evaluation

After running the quantum circuit and obtaining the measured results, we evaluate the kinetic and potential energy of the system based on the position and velocity qubits. The equations for the harmonic oscillator are as follows:

$$KE(t) = \frac{1}{2} m \left(y'(t)\right)^2 \tag{8}$$

$$PE(t) = \frac{1}{2} m\omega^2 \left(y(t)\right)^2 \tag{9}$$

Here, $y(t)$ represents the position (qubit $q[0]$) and $y'(t)$ represents the velocity (qubit $q[1]$). Using these measured values from the quantum simulation, we compute the kinetic and potential energy at different points in time.

2.7.1 Energy as a Function of Time

The measured output from the quantum simulation provides us with various states of the position and velocity qubits over time. These states correspond to specific values for $y(t)$ and $y'(t)$, which allow us to compute the energy values over the time interval $[0, 1]$.

For example, the most frequent result is $q = [1, 1]$, which corresponds to a high displacement and velocity. In this case, the kinetic energy and potential energy can be computed using the following relations:

$$KE(t) = \frac{1}{2} m \left(\text{velocity}\right)^2$$

$$PE(t) = \frac{1}{2} m\omega^2 \left(\text{position}\right)^2$$

We perform these calculations for each measured state of the position and velocity qubits, which provides the energy profile of the system over time.

2.7.2 Impact of Bound Changes on Energy Evaluation

Next, we analyze the impact of adjusting the bounds in functions such as `inplace_prepare_state()` or `entanglement_creation()`. By varying the bounds on the controlled rotations (e.g., changing the angle from 0.5), we observe how the distribution of measured states changes.

This modification alters the quantum evolution of the system, which, in turn, impacts the calculated kinetic and potential energy values. For instance, an increase in entanglement between the position and velocity qubits may lead to higher oscillation amplitudes, resulting in greater variation in energy values over time.

2.7.3 Results

After modifying the bounds, we measured the system's position and velocity qubits again to determine the new energy values. We observed that as the bounds increased, the kinetic energy and potential energy showed greater fluctuations, aligning with the expected behavior of a more entangled harmonic oscillator system.

The gate count optimization in terms of depth and width also affected the accuracy of the energy evaluations. A trade-off was observed between the complexity of the quantum circuit and the precision of the energy results.

3 Conclusion

In this paper, we have successfully demonstrated the application of quantum algorithms to solve the harmonic oscillator equation using Classiq's end-to-end quantum software platform. By transforming the differential equation into a first-order system and applying quantum-controlled operations, we simulated the time evolution of the system over the interval $[0, 1]$.

The measured results from the quantum circuit align with the expected behavior of the harmonic oscillator, where both position and velocity exhibit periodic oscillations. Through this simulation, we were able to calculate the kinetic and potential energy of the system at different points in time. Furthermore, we analyzed how varying the bounds in key quantum functions, such as `entanglement_creation()`, affected the energy values and the overall performance of the quantum circuit.

Our analysis shows that the quantum algorithm efficiently simulates the time evolution of the harmonic oscillator, validating its potential for solving other linear differential equations in a quantum framework. The optimization of gate depth and width proved to be a key factor in balancing circuit complexity with result accuracy. The results also highlight the advantages of quantum computing for simulating physical systems, offering exponential speedup over classical methods.

Future work may explore applying this approach to more complex systems, refining the optimization techniques, and further studying the trade-offs between circuit complexity and result precision. With continued advancements in quantum hardware and algorithms, these methods hold great promise for solving a wide range of differential equations and advancing our understanding of quantum simulations.

References

[1] Tao Xin, Shijie Wei, Jianlian Cui, Junxiang Xiao, Iñigo Arrazola, Lucas Lamata, Xiangyu Kong, Dawei Lu, Enrique Solano, and Guilu Long. Quantum algorithm for solving linear differential equations: Theory and experiment. *Phys. Rev. A*, 101:032307, Mar 2020.