OrbitaMX: Predict the orbit of the James Webb space telescope with a quantum algorithm Herman Kolden

Carlos Aráoz Alvarado
carlosaraoz@ciencias.unam.mx
Rodrigo Miqueo
rmiqueo@ciencias.unam.mx

Womanium Quantum Hackathon 2022

24 de agosto de 2022

WOMANIUM
STEM+Entrepreneurship

Task 1: The fast algorithm

Q1) How can you encode the position (x,y) of the telescope on a qubit?

As we are considering the position of the telescope, the components will be real, so remembering that the Euclidean norm for a vector $\mathbf{x} = (x_1, \ldots, x_N)$ is defined as $||\mathbf{x}|| = \sqrt{\sum_{i=1}^{N} x_i^2}$. Therefore, the squared norm of $\mathbf{x}$ should be equal to 1 as the probability of measuring a quantum state is equal to 1.

We can encode the position $\mathbf{x} = (x, y)$ of the telescope on a qubit by considering the norm $||\mathbf{x}|| = \sqrt{\sum_{i=1}^{N} x_i^2} = \sqrt{x^2 + y^2}$, this means to take the normalized version of the position $\mathbf{x}$ that is $\hat{x} = \mathbf{x}/||\mathbf{x}|| = (x/||\mathbf{x}||, y/||\mathbf{x}||)$. So that when we perform a measurment of this normalized state the probability of measuring it is $|\hat{x}| = x^2/||\mathbf{x}||^2 + y^2/||\mathbf{x}||^2 = (x^2 + y^2)/(\sqrt{x^2 + y^2})^2 = (x^2 + y^2)/(x^2 + y^2) = 1$.

Task 2: Input encoding

Q1) How can you encode the state of satellites on qubits, i.e. $\mathbf{x} = (x_1, y_1, x_2, y_2 \ldots, x_N, y_N)$. How many qubits do you need?

As we are considering the position of statellites, the components will be real, so remembering that the Euclidean norm for a vector $\mathbf{x} = (x_1, \ldots, x_N)$ is defined as $||\mathbf{x}|| = \sqrt{\sum_{i=1}^{N} x_i^2}$. Therefore, the squared norm of $\mathbf{x}$ should be equal to 1 as the probability of measuring a quantum state is equal to 1.

We can encode the position $\mathbf{x} = (x_1, y_1, x_2, y_2 \ldots, x_N, y_N)$ of the satellites on a qubit by considering the norm $||\mathbf{x}|| = \sqrt{\sum_{i=1}^{N} x_i^2} = \sqrt{x_1^2 + y_1^2 + \cdots + x_N^2 + y_N^2}$, this means to take the normalized version of the position $\mathbf{x}$ that is $\hat{x} = \mathbf{x}/||\mathbf{x}|| = \mathbf{x} = (x_1/||\mathbf{x}||, y_1/||\mathbf{x}||, x_2/||\mathbf{x}||, y_2/||\mathbf{x}|| \ldots, x_N/||\mathbf{x}||, y_N/||\mathbf{x}||)$.

Task 2: Input encoding

So that when we perform a measurment of this normalized state the probability of measuring it is $|\hat{x}| = (x_1^2/||\mathbf{x}||^2 + y_1^2/||\mathbf{x}||^2 + \cdots + x_N^2/||\mathbf{x}||^2 + y_N^2/||\mathbf{x}||^2) = (x_1^2 + y_1^2 + \cdots + x_N^2 + y_N^2)/(x_1^2 + y_1^2 + \cdots + x_N^2 + y_N^2) = 1$.

We will need N qubits to encode the state of stellites on qubits.

Task 2: Input encoding

Q2) If a classical computer and a quantum computer both spent 1 seconds on solving an 10x10 linear system, how long would it approximately take them each to solve a $10^{12}x10^{12}$ linear system?

To analyze a program, we begin by grouping inputs according to size. What we choose to call the size of an input can vary from program to program. For a program that solves n linear equations in n unknowns, it is normal to take n to be the size of the problem.[1]

For the classical Computer it takes 1 second on solving an 10x10 linear system with the computation complexity of solving linear systems O(n).Considering $O(n) = C_C * n$ for some $C_C > 0$ which gives $O(10) = C_C * 10 = 1sec$ $C_C = 0.1sec$.

[1] http://infolab.stanford.edu/ ullman/focs/ch03.pdf

Task 2: Input encoding

So for the $10^{12}x10^{12}$ linear system we have $T_{FC} = O(1X10^{12}) = 0.1sec * 10^{12} = 1x10^{11}sec$.

Now for the quantum computer it takes 1 second on solving an 10x10 linear system with the computation complexity of solving linear systems O(log(n)). (I will consider log(x)=ln(x) for this case) Considering $O(log(n)) = C_Q * n$ for some $C_Q > 0$ which gives $O(log(10)) = C_Q * log(10) = 1sec$ $Q_Q = 0.434sec$.

So for the $10^{12}x10^{12}$ linear system we have $T_{FQ} = O(log(1X10^{12})) = 0.434sec * log(10^{12}) = 12sec$.

Task 2: Input encoding

Q3) Consider your state preparation circuit and how many steps it requires. Can you give a lower bound on its computational complexity in terms of N? What does this mean for the future prospects of quantum differential equation solvers?

For our state preparation we have the following code:

Task 2: Input encoding

```
#Function to prepare the quantum state of N satellites

quantum_satellites=[] #this will be our quantum state for the N satellites

N=10 #Number of satellites

for i in range(N):
    picked_quantum_state=random_quantum_state() #call function that randomly pick a 2 dimentional state to simulate position of satellite
    n_state = normal_quantum_state(picked_quantum_state)  # Function that normalizes a 2 dimentional state to simulate quantum state of the position of satellite
    print(n_state,"this is randomly picked quantum state for satellite",i)
    print("Is it valid?",check_random_quantum_state(n_state)) #check if it is a valid quantum state
    if check_random_quantum_state(n_state)==True: #if it is a quantum valid quantum state then save it to our quantum state for N satellites
        x= n_state[0]  #define x component
        y = n_state[1]  #define y component
        quantum_satellites.append(x)  #save x position of satellite on our quantum state
        quantum_satellites.append(y)   #save y position of satellite on our quantum state
    print() # print an empty line
print("The imput quantum state for the",N,"satellites is given by the following state")
print()
print("x=",quantum_satellites)  #print the quantum state for the N satellites
```

Task 2: Input encoding

Frequently, we find a block of simple statements that are executed consecustatements tively. If the running time of each of these statements is O(1), then the entire block takes O(1) time, by the summation rule. That is, any constant number of O(1)'s sums to O(1).

Lines 4), 6), 7), 8), 9), 10), 11), 12), 13), 14) take O(1) time because they are assignments with no function calls. Lines 2),3),5) take O(1) time because of the summation rule. So for lines 4) to 14) because of the summation rule it takes O(1) time to runn it. We go around the loop N times. Thus, the running time of lines (1) to (14) is the product of N and O(1), which is O(N).

Task 3: Teleportation

Q1) Based on the algorithm described in https://qiskit.org/textbook/ch-algorithms/teleportation.html, describe a practical step-by-step guide for how to teleport the measurement quantum state down to Earth.

The general idea behind the practical step-by-step guide for how to teleport the measurement quantum state down to Earth is the following:

As the satellite scans every satellite orbiting earth and prepares a quantum state that efficiently encodes their positions, we assume that it is an N qubit state and that it is normalized.

For this purpose the satellite (that we will call "SS") needs to share a 2N entangled Generalized Bell state (G-states for simplicity) with our quantum computer in Earth (that we will call "EE").

Task 3: Teleportation

Then SS makes a 2N joint Generalized Bell measurement (G-measurement) with the N qubit to be teleported and half of the shared G-state and sends a 2N bit classical message to EE informing the measurement outcome.By G-measurement we mean that SS makes a joint measurement in the N qubits she wants to teleport plus the N qubits from the G-state shared with EE.

EE finishes the protocol performing at most 2N single qubit gates to obtain the teleported state. The number and type of unitary operations EE needs to apply in his N qubits is conditioned on SS' outcome.

Task 3: Teleportation

The N qubit teleportation protocol can be rigorously constructed with the results form Rigolin [2] as follows:

1) Generate the seed Generalized Bell state (G-states) $|s_0\rangle = (2^{-N/2}) \sum_{j=0}^{M} |x_j\rangle_{SS} |x_j\rangle_{EE}$, where SS and EE is written to emphasize which qubits are with Satellite(SS) and Earth (EE), where $M = 2^N - 1$ and x_j is the binary representation of the number j. Note that zeros must be added to make all x_j with the same amount of bits (N bits).

This G-state is our quantum channel and it is composed of 2N qubits.

2) Using the seed G-state it is possible to obtain all the G-states locally operating on its first N qubits, $|s_j\rangle = \otimes_{k=1}^{N} (\sigma_k^z)^{j_{2k-1}} (\sigma_k^x)^{j_{2k}} |s_0\rangle$.

[2] Rigolin, G. (2005). **Quantum teleportation of an arbitrary two-qubit state and its relation to multipartite entanglement.** Physical Review A, 71(3), 032303.

Task 3: Teleportation

Now j_k represents the k-th bit (from right to left) of the number $0 \leq j \leq 2^{2N} - 1$, which is written in binary notation and again zeros should be added to leave all j's with the same number of bits (2N bits).

The subindex k indicate on which qubit the Pauli matrices σ^x and σ^z should operate

3) SS makes a G-measurement obtaining with equal probabilities one of the 16 G-states.with the N qubits to be teleported and with her N qubits of the shared G-state. She then sends EE a 2N bit classical message informing EE the measurement outcome.

Task 3: Teleportation

4) With this information EE applies the corresponding unitary operation on his N qubits. These operations are given by $U_j = \otimes_{k=1}^{N} (\sigma_k^z)^{j_{2k-1}} (\sigma_k^x)^{j_{2k}}$.

After applying the unitary operation the protocol is finished and SS has succeeded in teleporting its arbitrary N qubit state.

Task 4: Practical application to new problems

Q1) If you had a perfect quantum computer, which simulation problem would you want it to solve?

If we had a perfect quantum computer we would like to simulate our research results about Supersymmetry Quantum Cosmology where we found a Hamiltonian equation and some modified Einstein-Klein-gordon equations.

Task 4: Practical application to new problems

Q2)Which algorithms have promise for solving such problems within the next 20 years? Optimism is encouraged.

There are some interesting advances in this field including the work done by Kapil et. al.[3], where they simulated the Klein Gordon equation on in an IBM quantum computer. Also in the work done by Bargassa et.al. [4] there are advances in a Quantum algorithm for the classification of supersymmetric top quark events.

[3] Kapil, M., Behera, B. K., & Panigrahi, P. K. (2018). **Quantum simulation of Klein Gordon equation and observation of Klein paradox in IBM quantum computer.** arXiv preprint arXiv:1807.00521.

[4] Bargassa, P., Cabos, T., Choi, A. C. O., Hessel, T., & Cavinato, S. (2021). **Quantum algorithm for the classification of supersymmetric top quark events.** Physical Review D, 104(9), 096004.

Task 4: Practical application to new problems

Q3)Which data and in what format would that algorithm need fed as input to perform the simulation?

As the simulations has been done by varying the equation for different values of their components no data input has been proposed, it may be possible to use data from the early universe like cosmological clocks.

Task 4: Practical application to new problems

Q4)How could you collect that data as quantum states or qubits instead of classical bits?

We would need to rewrite the data imput in to valid quantum states, as we have done on this challenge, but we need to be carefull to preserve the quantum conditions. So we would need to transform classical data in to quantum data, which can generate some problems.

Task 4: Practical application to new problems

Q5)Which quantum sensors or collection method would you need to gather the quantum data?

The sensor we would need are telescopes beyond our capability, and some more advances sensors using lasers and mirrors.

Task 4: Practical application to new problems

Q6) How would you transfer the quantum data from the quantum sensor to the quantum computer?

We could use some Teleportation algorithm for N imputs as mencioned by Rigolin. [5]

[5] Rigolin, G. (2005). **Quantum teleportation of an arbitrary two-qubit state and its relation to multipartite entanglement.** Physical Review A, 71(3), 032303.

Task 4: Practical application to new problems

Q7) What would the impact of your described end-to-end application be, and how could it solve real-life problems?

As the Klein Gordon equation was the first attempt at unifying special relativity and quantum mechanics it represents a very important equation for the search of a unified theory of General Relativity and Quantum Mechanics, in our specific problem we are also considering a Supersymmetric background which means we have a set of equations involving a lot of theories of Physics. By performing a simulation of these equations we could discover a new way to simulate these sets of equations and therefore to encounter many a more powerful way to explore physics. Also, in the process of having this ideal quantum computer we would discover the limitations of modern Quantum Computers.