Task3

August 24, 2022

Task 3: Teleportation

[1]:
```
# Do the necessary imports
import numpy as np
from qiskit import QuantumCircuit, QuantumRegister, ClassicalRegister
from qiskit import IBMQ, Aer, transpile, assemble
from qiskit.visualization import plot_histogram, plot_bloch_multivector,␣
 ↪array_to_latex
from qiskit.extensions import Initialize
#from qiskit.ignis.verification import marginal_counts
from qiskit.result import marginal_counts
from qiskit.quantum_info import random_statevector
from qiskit.algorithms.linear_solvers.numpy_linear_solver import␣
 ↪NumPyLinearSolver
from qiskit.algorithms.linear_solvers.hhl import HHL
from qiskit.quantum_info import Statevector
from qiskit import transpile
```

```
<frozen importlib._bootstrap>:219: RuntimeWarning:
scipy._lib.messagestream.MessageStream size changed, may indicate binary
incompatibility. Expected 56 from C header, got 64 from PyObject
```

Suppose we want to send the qubit state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ to a friend.

The following code has been taken from https://qiskit.org/textbook/ch-algorithms/teleportation.html and modified to solve our problem

Step 1 We define a function that creates an entangled pair of qubits.

The pair it creates is a special pair called a Bell pair. In quantum circuit language, the way to create a Bell pair between two qubits is to first transfer one of them to the X-basis ($|+\rangle$ and $|-\rangle$) using a Hadamard gate, and then to apply a CNOT gate onto the other qubit controlled by the one in the X-basis.

[2]:
```
def create_bell_pair(qc, a, b):
    """Creates a bell pair in qc using qubits a & b"""
    qc.h(a) # Put qubit a into state |+>
    qc.cx(a,b) # CNOT with a as control and b as target
```

Let's say we keep q_1 and we give q_2 to our friend after they part ways.

Step 2 We define a function that applies a CNOT gate to q_1, controlled by $|\psi\rangle$ (the qubit we are trying to send to our friend). Then it applies a Hadamard gate to $|\psi\rangle$. In our quantum circuit, the qubit ($|\psi\rangle$) that we are trying to send to our friend is q_0:

[3]:
```
def alice_gates(qc, psi, a):
    qc.cx(psi, a)
    qc.h(psi)
```

Step 3 Next, we define a function that applies a measurement to both qubits that we own, q_1 and $|\psi\rangle$, and stores this result in two classical bits. Then we send these two bits to our friend.

[4]:
```
def measure_and_send(qc, a, b):
    """Measures qubits a & b and 'sends' the results to Bob"""
    qc.barrier()
    qc.measure(a,0)
    qc.measure(b,1)
```

Step 4 Our friend, who already has the qubit q_2, then applies the following gates depending on the state of the classical bits:

00 $\rightarrow$ Do nothing

01 $\rightarrow$ Apply X gate

10 $\rightarrow$ Apply Z gate

11 $\rightarrow$ Apply ZX gate

(*Note that this transfer of information is purely classical.*)

We define a function to encode this instructions

[5]:
```
# This function takes a QuantumCircuit (qc), integer (qubit)
# and ClassicalRegisters (crz & crx) to decide which gates to apply
def bob_gates(qc, qubit, crz, crx):
    # Here we use c_if to control our gates with a classical
    # bit instead of a qubit
    qc.x(qubit).c_if(crx, 1) # Apply gates if the registers
    qc.z(qubit).c_if(crz, 1) # are in the state '1'
```

0.1 3. Simulating the Teleportation Protocol

From the Task 1 we recover after each iteration a vector of the form $

$$y = \begin{bmatrix} 0 \\ \vec{x} \end{bmatrix} \tag{1}$$

$ that then transforms itself to a imput vector of the form $

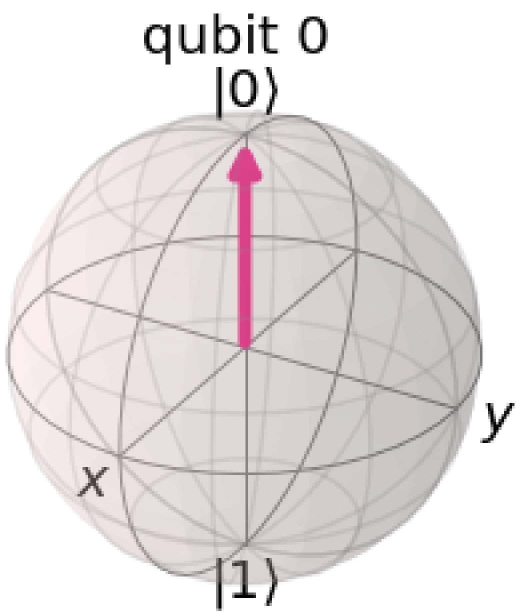
$$\begin{bmatrix} \vec{b} \\ 0 \end{bmatrix} \tag{2}$$

$, in each case we have a 4 entry vector with 2 entryes equal to 0, which means we can reduce the problem back to a 2 components vector and apply the algortihm to perform the teleportation of a single qubit. In this notebook, we will initialize our qubit in the quantum state $|\psi\rangle = (1, 0)$.

[6]:
```
# Create random 1-qubit state
#psi = random_statevector(2)
#We consider only the entryies of vectort \vec{b}
psi  = np.array([1, 0])  #create our quantum state
# Display it nicely
display(array_to_latex(psi, prefix="|\\psi\\rangle ="))
# Show it on a Bloch sphere
plot_bloch_multivector(psi)
```

$$|\psi\rangle = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

[6]:



Let's create our initialization instruction to create $|\psi\rangle$ from the state $|0\rangle$:

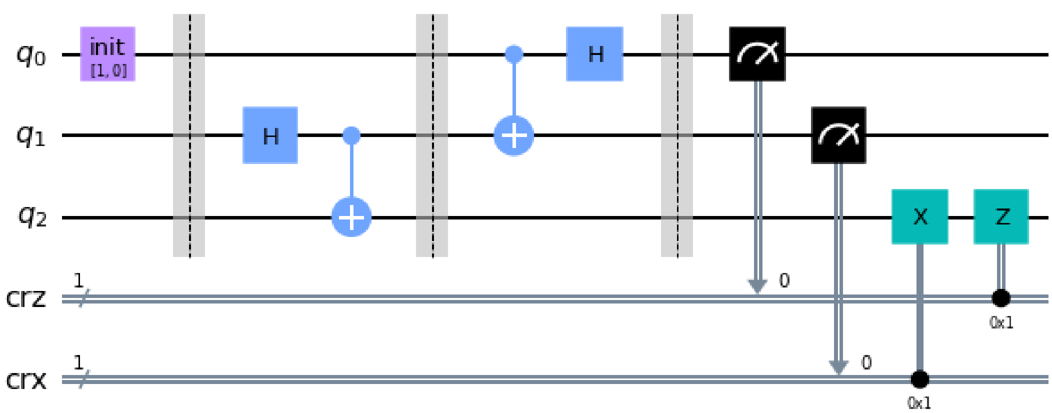
[7]:
```
init_gate = Initialize(psi)
init_gate.label = "init"
```

we can use the Aer simulator to verify our qubit has been teleported.

[8]:
```
## SETUP
qr = QuantumRegister(3, name="q")    # Protocol uses 3 qubits
crz = ClassicalRegister(1, name="crz") # and 2 classical registers
crx = ClassicalRegister(1, name="crx")
```

```
qc = QuantumCircuit(qr, crz, crx)

## STEP 0
# First, let's initialize Alice's q0
qc.append(init_gate, [0])
qc.barrier()

## STEP 1
# Now begins the teleportation protocol
create_bell_pair(qc, 1, 2)
qc.barrier()

## STEP 2
# Send q1 to Alice and q2 to Bob
alice_gates(qc, 0, 1)

## STEP 3
# Alice then sends her classical bits to Bob
measure_and_send(qc, 0, 1)

## STEP 4
# Bob decodes qubits
bob_gates(qc, 2, crz, crx)

# Display the circuit
qc.draw()
```

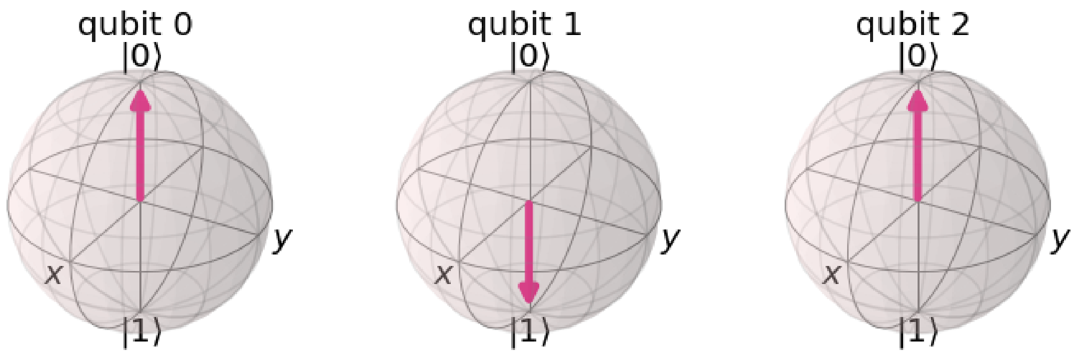
[8]:



We can see below, using the statevector obtained from the aer simulator, that the state of $|q_2\rangle$ is the same as the state $|\psi\rangle$ we created above, while the states of $|q_0\rangle$ and $|q_1\rangle$ have been collapsed to either $|0\rangle$ or $|1\rangle$. The state $|\psi\rangle$ has been teleported from qubit 0 to qubit 2.

[9]:
```python
from qiskit import QuantumRegister, ClassicalRegister, QuantumCircuit, execute,␣
↪Aer
sim =Aer.get_backend('aer_simulator')
qc.save_statevector()
out_vector = sim.run(qc).result().get_statevector()
plot_bloch_multivector(out_vector)
```

[9]:



0.1.1 3.3 Using the Simulated Counts

We can demonstrate that the gates perform the correct transformations on a single quantum chip. Here we again use the aer simulator to simulate how we might test our protocol.

On a real quantum computer, we would not be able to sample the statevector, so if we wanted to check our teleportation circuit is working, we need to do things slightly differently. The `Initialize` instruction first performs a reset, setting our qubit to the state $|0\rangle$. It then applies gates to turn our $|0\rangle$ qubit into the state $|\psi\rangle$:

$$|0\rangle \xrightarrow{\text{Initialize gates}} |\psi\rangle$$

Since all quantum gates are reversible, we can find the inverse of these gates using:

[10]:
```python
inverse_init_gate = init_gate.gates_to_uncompute()
```

This operation has the property:

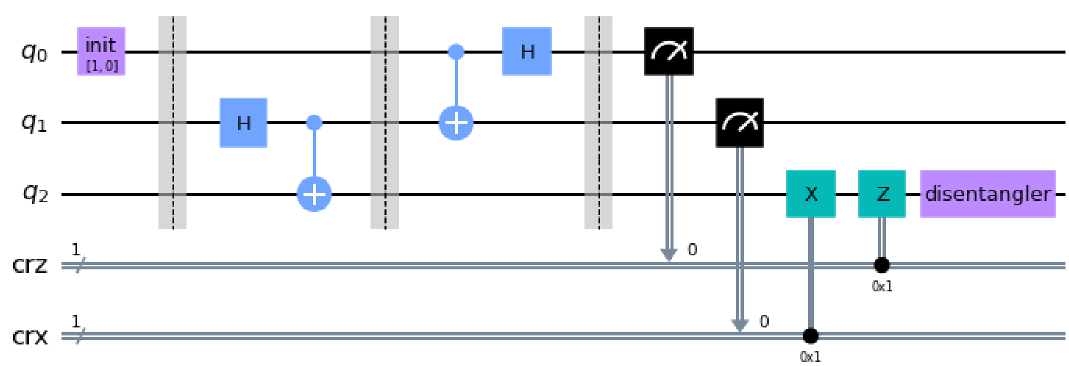
$$|\psi\rangle \xrightarrow{\text{Inverse Initialize gates}} |0\rangle$$

To prove the qubit $|q_0\rangle$ has been teleported to $|q_2\rangle$, if we do this inverse initialization on $|q_2\rangle$, we expect to measure $|0\rangle$ with certainty. We do this in the circuit below:

[11]:
```python
## SETUP
qr = QuantumRegister(3, name="q")    # Protocol uses 3 qubits
crz = ClassicalRegister(1, name="crz") # and 2 classical registers
crx = ClassicalRegister(1, name="crx")
```

```
qc = QuantumCircuit(qr, crz, crx)

## STEP 0
# First, let's initialize Alice's q0
qc.append(init_gate, [0])
qc.barrier()

## STEP 1
# Now begins the teleportation protocol
create_bell_pair(qc, 1, 2)
qc.barrier()

## STEP 2
# Send q1 to Alice and q2 to Bob
alice_gates(qc, 0, 1)

## STEP 3
# Alice then sends her classical bits to Bob
measure_and_send(qc, 0, 1)

## STEP 4
# Bob decodes qubits
bob_gates(qc, 2, crz, crx)

## STEP 5
# reverse the initialization process
qc.append(inverse_init_gate, [2])

# Display the circuit
qc.draw()
```

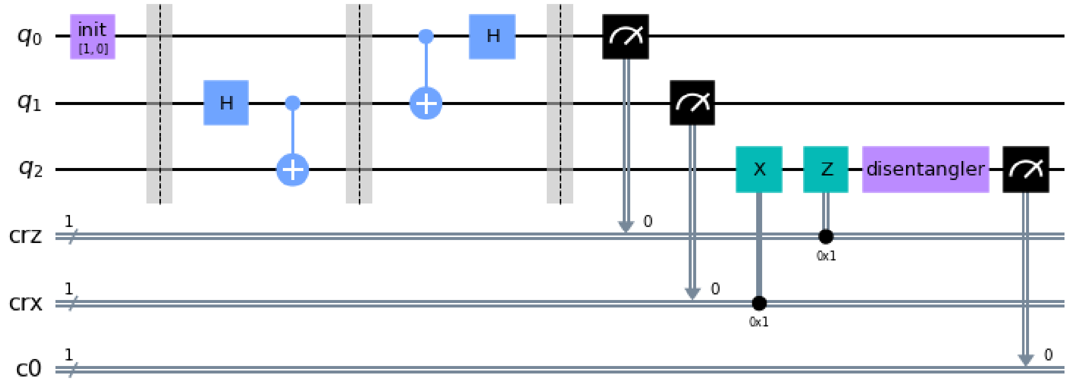
[11]:



We can see the `inverse_init_gate` appearing, labelled 'disentangler' on the circuit diagram. Finally, we measure the third qubit and store the result in the third classical bit:

[12]:
```
# Need to add a new ClassicalRegister
# to see the result
cr_result = ClassicalRegister(1)
qc.add_register(cr_result)
qc.measure(2,2)
qc.draw()
```

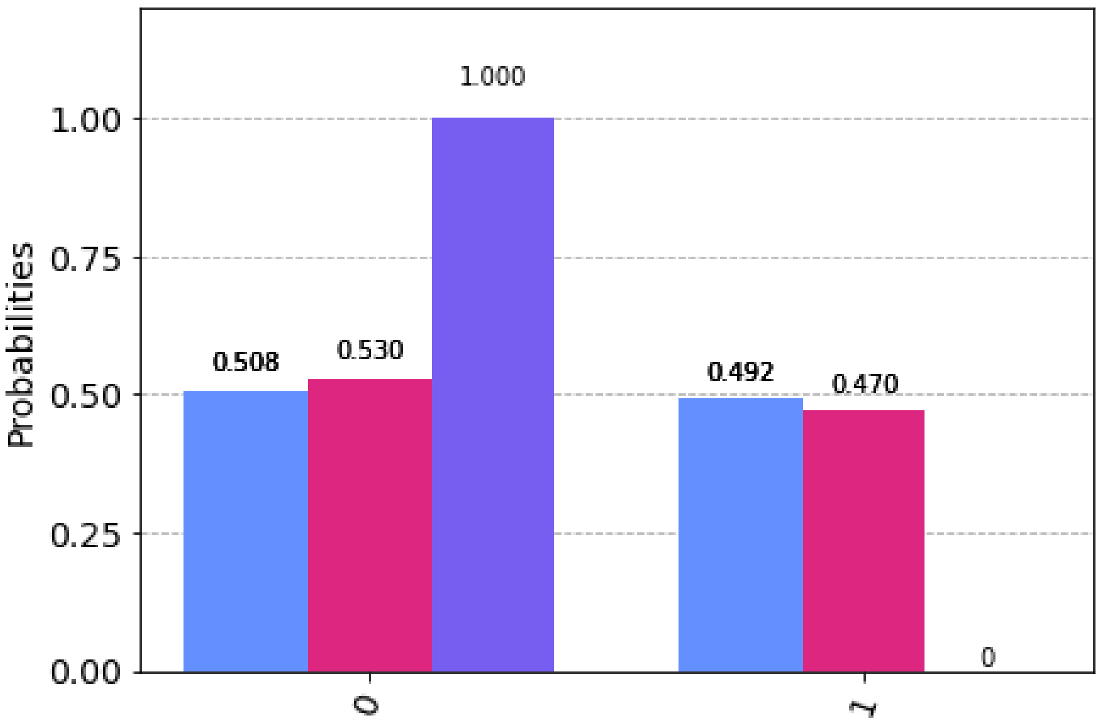
[12]:



and we run our experiment:

[13]:
```
t_qc = transpile(qc, sim)
t_qc.save_statevector()
counts = sim.run(t_qc).result().get_counts()
qubit_counts = [marginal_counts(counts, [qubit]) for qubit in range(3)]
plot_histogram(qubit_counts)
```

[13]:



We can see we have a 100% chance of measuring q_2 (the purple bar in the histogram) in the state $|0\rangle$. This is the expected result, and indicates the teleportation protocol has worked properly.

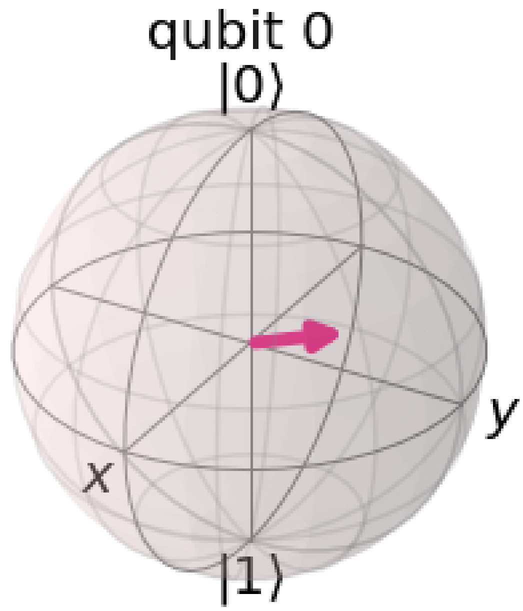
Teleportation algorithm + imput vector

[14]:
```
#
#We consider only the entries of vectort \vec{b} and take for example the␣
 ↪following vector imput
psi = [np.sqrt(0.3),-np.sqrt(0.7)]
# Display it nicely
display(array_to_latex(psi, prefix="|\\psi\\rangle ="))
# Show it on a Bloch sphere
sign1 = np.sign(psi[0]) #keep sign of the first entry of the array psi
sign2 = np.sign(psi[1])  #keep sign of the second entry of the array psi
print('sign of our array:', 'first entry =',sign1,'second entry =',sign2)
plot_bloch_multivector(psi)
```

$$|\psi\rangle = \begin{bmatrix} 0.54772 & -0.83666 \end{bmatrix}$$

```
sign of our array: first entry = 1.0 second entry = -1.0
```

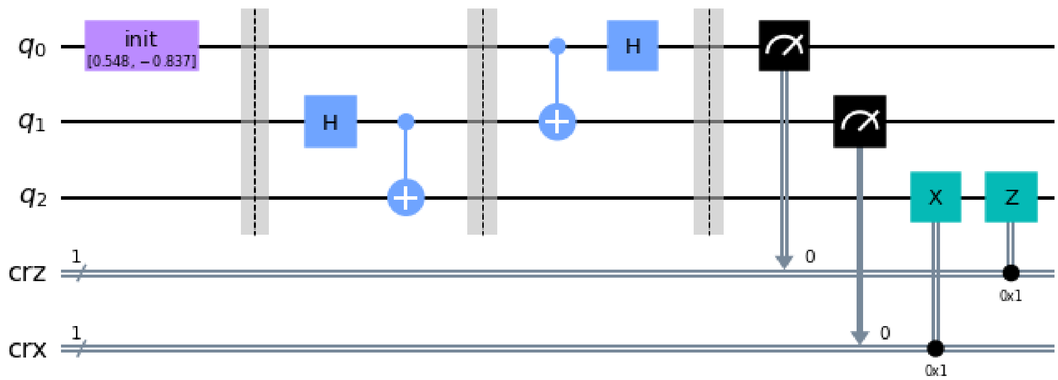
[14]:



[15]:
```
#instruction to create $|\psi\rangle$ from the state $|0\rangle$

init_gate = Initialize(psi)
init_gate.label = "init"
qr = QuantumRegister(3, name="q")    # Protocol uses 3 qubits
crz = ClassicalRegister(1, name="crz") # and 2 classical registers
crx = ClassicalRegister(1, name="crx")
qc = QuantumCircuit(qr, crz, crx)
qr2 =(1,"qrn")

## STEP 0
# First, let's initialize our's q0
qc.append(init_gate, [0])
qc.barrier()
st0 = Statevector.from_instruction(qc)

## STEP 1
# Now begins the teleportation protocol
create_bell_pair(qc, 1, 2)
qc.barrier()

## STEP 2
# Send q1 to Alice and q2 to Bob
alice_gates(qc, 0, 1)
```

```
## STEP 3
# Alice then sends her classical bits to Bob
measure_and_send(qc, 0, 1)

## STEP 4
# Bob decodes qubits
bob_gates(qc, 2, crz, crx)



# Display the circuit
qc.draw()
            #
```

[15]:



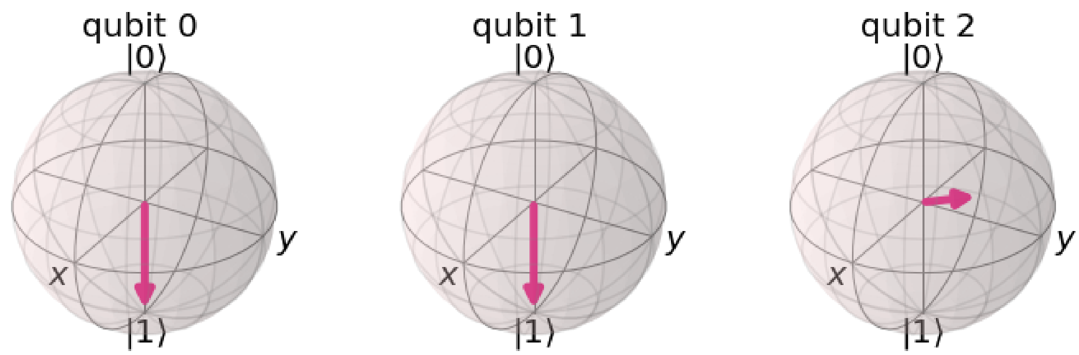
[16]:
```
sim = Aer.get_backend('aer_simulator')
qc.save_statevector()
out_vector = sim.run(qc).result().get_statevector()

#Visualization of the Statevector.
out_vector.draw('latex')
```

[16]:

$$\frac{\sqrt{30}}{10}|011\rangle - \frac{\sqrt{70}}{10}|111\rangle$$

[17]:
```
#Visualization on Bloch Spheres.
plot_bloch_multivector(out_vector)
```

[17]:



[18]:
```
#Obtain probabilities of Statevector
probs = out_vector.probabilities()
print( 'probabilities of our system of qubits: {}'.format(probs))
print()


# Probabilities for measuring only qubit-2
probs_qubit_0 = out_vector.probabilities([2])
print('Qubit-2 probabilities: {}'.format(probs_qubit_0))
print('probabilities qubit 2: zeroth component =' , probs_qubit_0[0],'first␣
 ↪component =' , probs_qubit_0[1])
print()

#create a new teleported imput vector
vector2=np.array([sign1*np.sqrt(probs_qubit_0[0]),sign2*np.
 ↪sqrt(probs_qubit_0[1])])
print('new imput teleported vector', vector2)
print()

# Display the output state vector
print(out_vector)

#References for this part
#https://qiskit.org/documentation/stable/0.24/stubs/qiskit.quantum_info.
 ↪Statevector.html
```

```
probabilities of our system of qubits: [0.  0.  0.  0.3 0.  0.  0.  0.7]

Qubit-2 probabilities: [0.3 0.7]
probabilities qubit 2: zeroth component = 0.2999999999999997 first component =
0.7000000000000001

new imput teleported vector [ 0.54772256 -0.83666003]
```

```
Statevector([-0.        +0.00000000e+00j,  0.        +0.00000000e+00j,
              0.        +0.00000000e+00j,  0.54772256+0.00000000e+00j,
             -0.        +0.00000000e+00j, -0.        +0.00000000e+00j,
              0.        -0.00000000e+00j, -0.83666003+1.02461302e-16j],
            dims=(2, 2, 2))
```

Here we present our code of quantum teleportation + our code to simulate the orbit of the James Webb telescope

[19]:
```
#Code to simulate the orbit of the James Webb telescope
xv1=[1] #initial position for quantum algorithm
yv1=[0]
cxv1 = [1] #initial position for classic algorithm
cyv1 = [0]
t1 = [] #list to keep track of time
vector = np.array([1, 0, 0, 0])  #vector b representing initial position at time␣
 ↪t=0
vector = vector/np.linalg.norm(vector)  #normaized vector b representing initial␣
 ↪position at time t=0
for i in range(181):
    T = 180  #180 days= 6 months (30 days per month)
    dt = 1  #every day we measure
    #Quantum Teleportation
    #We consider only the first two entries of vectort \vec{b} as our imput␣
 ↪state psi
    psi = [vector[0],vector[1]]
    sign1 = np.sign(psi[0]) #keep sign of the first entry of the array psi
    sign2 = np.sign(psi[1])  #keep sign of the second entry of the array psi
    #instruction to create $|\psi\rangle$ from the state $|0\rangle$
    init_gate = Initialize(psi)
    init_gate.label = "init"
    qr = QuantumRegister(3, name="q")   # Protocol uses 3 qubits
    crz = ClassicalRegister(1, name="crz") # and 2 classical registers
    crx = ClassicalRegister(1, name="crx")
    qc = QuantumCircuit(qr, crz, crx)
    qr2 =(1,"qrn")

    ## STEP 0
    # First, let's initialize Alice's q0
    qc.append(init_gate, [0])
    qc.barrier()
    st0 = Statevector.from_instruction(qc)

    ## STEP 1
    # Now begins the teleportation protocol
    create_bell_pair(qc, 1, 2)
    qc.barrier()
```

```
    ## STEP 2
    # Send q1 to Alice and q2 to Bob
    alice_gates(qc, 0, 1)

    ## STEP 3
    # Alice then sends her classical bits to Bob
    measure_and_send(qc, 0, 1)

    ## STEP 4
    # Bob decodes qubits
    bob_gates(qc, 2, crz, crx)

    sim = Aer.get_backend('aer_simulator')
    qc.save_statevector()
    out_vector = sim.run(qc).result().get_statevector()


    #Obtain probabilities of Statevector
    probs = out_vector.probabilities()

    # Probabilities for measuring only qubit-2
    probs_qubit_0 = out_vector.probabilities([2])

    #create a new teleported imput vector adding the sign
    vector2=np.array([sign1*np.sqrt(probs_qubit_0[0]),sign2*np.
↪sqrt(probs_qubit_0[1]),0,0])



    #orbit of the James Webb


    print('On day',i,'the results are') #print day of measurment
    print('new imput teleported vector', vector2)
    #print('normalized vector imput', vector)  #print normalized imput vector
    print('square of the norm of teleported vector imput',np.linalg.
↪norm(vector2)**2) #print square of the norm of vector imput(sum of squared␣
↪components) to cheek if it is a valid quantum state
    matrix = np.array([[0,0,1, (2*np.pi*dt)/T], [0,0,-(2*np.pi*dt)/
↪T,1],[1,-(2*np.pi*dt)/T,0,0],[(2*np.pi*dt)/T,1,0,0] ])  #Hermitian Matrix
    naive_hhl_solution = HHL().solve(matrix, vector2)  #naive_hhl_solution from␣
↪HHL algorithm
    classical_solution = NumPyLinearSolver().solve(matrix, vector2 / np.linalg.
↪norm(vector2)) #classical_solution from classical solution algorithm
    naive_sv = Statevector(naive_hhl_solution.state).data #state vector of␣
↪quantum solution
```

```
    #Extract the right vector components, i.e. those corresponding to the␣
↪ancillary qubit (bottom in the circuits) being 1 and the work qubits (the four␣
↪middle in the circuits) being 0,
    #Extract the right vector components. |1000000> corresponds to the index 64,␣
↪|1000001> corresponds to the index 65,|1000010> corresponds to the index 66␣
↪and |1000011> corresponds to the index 67
    naive_full_vector = np.array([ naive_sv[64], naive_sv[65], naive_sv[66],␣
↪naive_sv[67]])
    naive_full_vector = np.real(naive_full_vector)  #Real part of␣
↪naive_full_vector
    #We are interested in the third and fourth components of the solution␣
↪vector, which correspond to the components of \vec{x}
    #Thus, we are interested in the states |1000010> ,|1000011>, corresponding␣
↪to the third and fourth components of the solution vector respectively.␣
↪|1000010> corresponds to the index 66 and |1000011> corresponds to the index␣
↪67
    naive_partial_vector = np.array([ 0, 0, naive_sv[66], naive_sv[67]])
    naive_partial_vector = np.real(naive_partial_vector)  #Real part of␣
↪naive_full_vector
    x=naive_partial_vector[2]/np.linalg.norm(naive_partial_vector)  #normalized␣
↪x component
    y = naive_partial_vector[3]/np.linalg.norm(naive_partial_vector) ␣
↪#normalized y component
    vector=np.array([x,y, 0, 0])  #normaized vector b representing initial␣
↪position at time t=i*dt
    #print('full naive solution vector:', "y=[",0,"        ",0,"        ␣
↪",naive_hhl_solution.euclidean_norm*naive_full_vector[2]/np.linalg.
↪norm(naive_full_vector),naive_hhl_solution.euclidean_norm*␣
↪naive_full_vector[3]/np.linalg.norm(naive_full_vector),"]")
    #we will divide the vectors by their respective norms to suppress any␣
↪constants coming from the different parts of the circuits. The full solution␣
↪vector can then be recovered by multiplying these normalised vectors by the␣
↪respective Euclidean norms calculated above
    print('full naive solution vector:', "y=[",0,"        ",0,"        ␣
↪",naive_hhl_solution.euclidean_norm*naive_partial_vector[2]/np.linalg.
↪norm(naive_partial_vector),naive_hhl_solution.euclidean_norm*␣
↪naive_partial_vector[3]/np.linalg.norm(naive_partial_vector),"]") #print full␣
↪naive solution vector
    print('classical state:', "y=",classical_solution.state) #print classical␣
↪state
    print()
    xv1.append(naive_hhl_solution.euclidean_norm*naive_partial_vector[2]/np.
↪linalg.norm(naive_partial_vector))  #add the result from "x(t_i)" component␣
↪obtained from HHL algorithm to the list
```

```
    yv1.append(naive_hhl_solution.euclidean_norm*naive_partial_vector[3]/np.
↪linalg.norm(naive_partial_vector))  #add the result from "y(t_i)" component␣
↪obtained from HHL algorithm to the list
    cxv1.append(classical_solution.state[2])  #add the result from "x(t_i)"␣
↪component obtained from classical algorithm to the list
    cyv1.append(classical_solution.state[3])  #add the result from "y(t_i)"␣
↪component obtained from classical algorithm to the list
    t1.append(i) #savind time step
```

```
On day 0 the results are
new imput teleported vector [1. 0. 0. 0.]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          0.9987830131823715
0.03486410418604186 ]
classical state: y= [0.         0.         0.99878301 0.0348641 ]

On day 1 the results are
new imput teleported vector [0.99939132 0.03488534 0.         0.         ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.9969588292189014
0.0696857663000798 ]
classical state: y= [0.         0.         0.99695883 0.06968577]

On day 2 the results are
new imput teleported vector [0.99756603 0.06972821 0.         0.         ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.9939209901395925
0.10442259593729666 ]
classical state: y= [0.         0.         0.99392099 0.1044226 ]

On day 3 the results are
new imput teleported vector [0.99452634 0.10448619 0.         0.         ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.9896731940800472
0.13903230596430174 ]
classical state: y= [0.         0.         0.98967319 0.13903231]

On day 4 the results are
new imput teleported vector [0.99027595 0.13911698 0.         0.         ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.9842206121258348
0.17347276399768854 ]
classical state: y= [0.         0.         0.98422061 0.17347276]

On day 5 the results are
new imput teleported vector [0.98482005 0.17357842 0.         0.         ]
square of the norm of teleported vector imput 1.0
```

```
full naive solution vector: y=[ 0          0          0.9775698820174445
0.2077020436942037 ]
classical state: y= [0.         0.         0.97756988 0.20770204]

On day 6 the results are
new imput teleported vector [0.97816527 0.20782854 0.         0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.9697291000697732
0.2416784757900851 ]
classical state: y= [0.         0.         0.9697291  0.24167848]

On day 7 the results are
new imput teleported vector [0.97031971 0.24182567 0.         0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.9607078113160173
0.27536069882744324 ]
classical state: y= [0.         0.         0.96070781 0.2753607 ]

On day 8 the results are
new imput teleported vector [0.96129293 0.27552841 0.         0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.9505169978879641
0.3087077095059382 ]
classical state: y= [0.         0.         0.950517   0.30870771]

On day 9 the results are
new imput teleported vector [0.95109591 0.30889573 0.         0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          0.939169065646807
0.341678912598443 ]
classical state: y= [0.         0.         0.93916907 0.34167891]

On day 10 the results are
new imput teleported vector [0.93974107 0.34188701 0.         0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          0.9266778290807827
0.37423417036994183 ]
classical state: y= [0.         0.         0.92667783 0.37423417]

On day 11 the results are
new imput teleported vector [0.92724222 0.3744621  0.         0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.9130584944880019
0.40633385143949646 ]
classical state: y= [0.         0.         0.91305849 0.40633385]

On day 12 the results are
new imput teleported vector [0.91361459 0.40658133 0.         0.        ]
```

```
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          0.8983276414649509
0.4379388790257926 ]
classical state: y= [0.         0.         0.89832764 0.43793888]

On day 13 the results are
new imput teleported vector [0.89887477 0.43820561 0.         0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          0.8825032027232029
0.4690107785175508 ]
classical state: y= [0.         0.         0.8825032  0.46901078]

On day 14 the results are
new imput teleported vector [0.88304069 0.46929643 0.         0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.8656044422588927
0.4995117243108732 ]
classical state: y= [0.         0.         0.86560444 0.49951172]

On day 15 the results are
new imput teleported vector [0.86613164 0.49981595 0.         0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.8476519319015551
0.5294045858565174 ]
classical state: y= [0.         0.         0.84765193 0.52940459]

On day 16 the results are
new imput teleported vector [0.84816819 0.52972702 0.         0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.8286675262708592
0.5586529728610454 ]
classical state: y= [0.         0.         0.82866753 0.55865297]

On day 17 the results are
new imput teleported vector [0.82917223 0.55899322 0.         0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.8086743361717242
0.58722127958681 ]
classical state: y= [0.         0.         0.80867434 0.58722128]

On day 18 the results are
new imput teleported vector [0.80916686 0.58757893 0.         0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          0.7876967004602142
0.615074728196861 ]
classical state: y= [0.         0.         0.7876967  0.61507473]

On day 19 the results are
```

```
new imput teleported vector [0.78817645 0.61544934 0.         0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.7657601564144502
0.6421794110919982 ]
classical state: y= [0.         0.         0.76576016 0.64217941]

On day 20 the results are
new imput teleported vector [0.76622654 0.64257053 0.         0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          0.7428914086466266
0.6685023321884372 ]
classical state: y= [0.         0.         0.74289141 0.66850233]

On day 21 the results are
new imput teleported vector [0.74334387 0.66890948 0.         0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          0.719118296593947
0.6940114470858316 ]
classical state: y= [0.         0.         0.7191183  0.69401145]

On day 22 the results are
new imput teleported vector [0.71955628 0.69443413 0.         0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.694469760628085
0.7186757020767567 ]
classical state: y= [0.         0.         0.69446976 0.7186757 ]

On day 23 the results are
new imput teleported vector [0.69489273 0.71911341 0.         0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          0.6689758068244136
0.742465071950171 ]
classical state: y= [0.         0.         0.66897581 0.74246507]

On day 24 the results are
new imput teleported vector [0.66938325 0.74291727 0.         0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          0.6426674704338887
0.7653505965428227 ]
classical state: y= [0.         0.         0.64266747 0.7653506 ]

On day 25 the results are
new imput teleported vector [0.64305889 0.76581673 0.         0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.6155767781020578
0.787304415994117 ]
classical state: y= [0.         0.         0.61557678 0.78730442]
```

```
On day 26 the results are
new imput teleported vector [0.61595169 0.78778392 0.         0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          0.5877367088811913
0.8082998046615216 ]
classical state: y= [0.         0.         0.58773671 0.8082998 ]

On day 27 the results are
new imput teleported vector [0.58809467 0.8087921  0.         0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.5591811540829909
0.8283112036552208 ]
classical state: y= [0.         0.         0.55918115 0.8283112 ]

On day 28 the results are
new imput teleported vector [0.55952172 0.82881569 0.         0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.529944876020756
0.847314251952422 ]
classical state: y= [0.         0.         0.52994488 0.84731425]

On day 29 the results are
new imput teleported vector [0.53026764 0.84783031 0.         0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          0.5000634656912233
0.8652858160534188 ]
classical state: y= [0.         0.         0.50006347 0.86528582]

On day 30 the results are
new imput teleported vector [0.50036803 0.86581282 0.         0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          0.4695732994476094
0.8822040181433269 ]
classical state: y= [0.         0.         0.4695733  0.88220402]

On day 31 the results are
new imput teleported vector [0.46985929 0.88274132 0.         0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          0.4385114947165951
0.8980482627252087 ]
classical state: y= [0.         0.         0.43851149 0.89804826]

On day 32 the results are
new imput teleported vector [0.43877857 0.89859522 0.         0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          0.4069158648131447
0.9127992616921393 ]
classical state: y= [0.         0.         0.40691586 0.91279926]
```

```
On day 33 the results are
new imput teleported vector [0.4071637 0.9133552 0.         0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          0.37482487290820166
0.926439057807741 ]
classical state: y= [0.         0.         0.37482487 0.92643906]

On day 34 the results are
new imput teleported vector [0.37505316 0.9270033  0.         0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.34227758520526064
0.9389510465665395 ]
classical state: y= [0.         0.         0.34227759 0.93895105]

On day 35 the results are
new imput teleported vector [0.34248605 0.93952291 0.         0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.30931362338284035
0.9503199964075894 ]
classical state: y= [0.         0.         0.30931362 0.95032   ]

On day 36 the results are
new imput teleported vector [0.30950201 0.95089879 0.         0.        ]
square of the norm of teleported vector imput 0.9999999999999996
full naive solution vector: y=[ 0          0          0.2759731163607434
0.9605320672567284 ]
classical state: y= [0.         0.         0.27597312 0.96053207]

On day 37 the results are
new imput teleported vector [0.2761412  0.96111708 0.         0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.242296651448822
0.9695748273748834 ]
classical state: y= [0.         0.         0.24229665 0.96957483]

On day 38 the results are
new imput teleported vector [0.24244422 0.97016535 0.         0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          0.20832522493772138
0.9774372684919435 ]
classical state: y= [0.         0.         0.20832522 0.97743727]

On day 39 the results are
new imput teleported vector [0.20845211 0.97803258 0.         0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.17410019219174497
0.9841098192077655 ]
```

```
classical state: y= [0.         0.         0.17410019 0.98410982]

On day 40 the results are
new imput teleported vector [0.17420623 0.98470919 0.         0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          0.13966321730459944
0.989584356643976 ]
classical state: y= [0.         0.         0.13966322 0.98958436]

On day 41 the results are
new imput teleported vector [0.13974828 0.99018706 0.         0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.10505622237930544
0.993854216332421 ]
classical state: y= [0.         0.         0.10505622 0.99385422]

On day 42 the results are
new imput teleported vector [0.10512021 0.99445952 0.         0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.0703213364940195
0.9969142003282131 ]
classical state: y= [0.         0.         0.07032134 0.9969142 ]

On day 43 the results are
new imput teleported vector [0.07036417 0.99752137 0.         0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.03550084441589212
0.9987605835374663 ]
classical state: y= [0.         0.         0.03550084 0.99876058]

On day 44 the results are
new imput teleported vector [0.03552247 0.99936888 0.         0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          0.0006371351253996882
0.9993911182520959 ]
classical state: y= [0.00000000e+00 0.00000000e+00 6.37135125e-04
9.99391118e-01]

On day 45 the results are
new imput teleported vector [6.37523172e-04 9.99999797e-01 0.00000000e+00
0.00000000e+00]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.034227349786193184
0.998805036886057 ]
classical state: y= [ 0.          0.         -0.03422735  0.99880504]

On day 46 the results are
new imput teleported vector [-0.0342482   0.99941336  0.          0.        ]
```

```
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.06905016778341413
0.9970030529097826 ]
classical state: y= [ 0.          0.         -0.06905017  0.99700305]

On day 47 the results are
new imput teleported vector [-0.06909222  0.99761028  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.10378892705431186
0.9939873599816348 ]
classical state: y= [ 0.          0.         -0.10378893  0.99398736]

On day 48 the results are
new imput teleported vector [-0.10385214  0.99459275  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.1384013381164417
0.9897616292774346 ]
classical state: y= [ 0.          0.         -0.13840134  0.98976163]

On day 49 the results are
new imput teleported vector [-0.13848563  0.99036444  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.17284526529827285
0.9843310050213356 ]
classical state: y= [ 0.          0.         -0.17284527  0.98433101]

On day 50 the results are
new imput teleported vector [-0.17295054  0.98493051  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.20707877803335809
0.9777020982234582 ]
classical state: y= [ 0.          0.         -0.20707878  0.9777021 ]

On day 51 the results are
new imput teleported vector [-0.2072049   0.97829757  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.2410602019048148
0.9698829786319356 ]
classical state: y= [ 0.          0.         -0.2410602   0.96988298]

On day 52 the results are
new imput teleported vector [-0.24120702  0.97047369  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.2747481693779793
0.9608831649091335 ]
classical state: y= [ 0.          0.         -0.27474817  0.96088316]

On day 53 the results are
```

```
new imput teleported vector [-0.2749155   0.96146839  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.3081016701594802
0.9507136130440557 ]
classical state: y= [ 0.          0.         -0.30810167  0.95071361]

On day 54 the results are
new imput teleported vector [-0.30828932  0.95129264  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.34108010112141646
0.9393867030149776 ]
classical state: y= [ 0.         0.        -0.3410801  0.9393867]

On day 55 the results are
new imput teleported vector [-0.34128784  0.93995884  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.3736433157298723
0.9269162237185901 ]
classical state: y= [ 0.          0.         -0.37364332  0.92691622]

On day 56 the results are
new imput teleported vector [-0.37387088  0.92748076  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.4057516729175948
0.9133173561839928 ]
classical state: y= [ 0.          0.         -0.40575167  0.91331736]

On day 57 the results are
new imput teleported vector [-0.4059988   0.91387361  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.43736608534133603
0.8986066550919488 ]
classical state: y= [ 0.          0.         -0.43736609  0.89860666]

On day 58 the results are
new imput teleported vector [-0.43763246  0.89915395  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.4684480669651146
0.8828020286219278 ]
classical state: y= [ 0.          0.         -0.46844807  0.88280203]

On day 59 the results are
new imput teleported vector [-0.46873337  0.8833397   0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.4989597799114776
0.8659227166514546 ]
classical state: y= [ 0.          0.         -0.49895978  0.86592272]
```

```
On day 60 the results are
new imput teleported vector [-0.49926367  0.86645011  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.5288640805237085
0.8479892673342979 ]
classical state: y= [ 0.          0.          -0.52886408  0.84798927]

On day 61 the results are
new imput teleported vector [-0.52918618  0.84850573  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.5581245645829367
0.8290235120860276 ]
classical state: y= [ 0.          0.          -0.55812456  0.82902351]

On day 62 the results are
new imput teleported vector [-0.55846449  0.82952843  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.5867056116250793
0.8090485390073878 ]
classical state: y= [ 0.          0.          -0.58670561  0.80904854]

On day 63 the results are
new imput teleported vector [-0.58706294  0.80954129  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.6145724283036677
0.7880886647778159 ]
classical state: y= [ 0.          0.          -0.61457243  0.78808866]

On day 64 the results are
new imput teleported vector [-0.61494673  0.78856865  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.6416910907457875
0.7661694050533671 ]
classical state: y= [ 0.          0.          -0.64169109  0.76616941]

On day 65 the results are
new imput teleported vector [-0.64208191  0.76663604  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.6680285858495479
0.7433174434050369 ]
classical state: y= [ 0.          0.          -0.66802859  0.74331744]

On day 66 the results are
new imput teleported vector [-0.66843545  0.74377016  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.6935528514728313
0.7195605988353259 ]
classical state: y= [ 0.          0.          -0.69355285  0.7195606 ]
```

```
On day 67 the results are
new imput teleported vector [-0.69397526  0.71999885  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.7182328154643665
0.6949277919125681 ]
classical state: y= [ 0.          0.         -0.71823282  0.69492779]

On day 68 the results are
new imput teleported vector [-0.71867025  0.69535104  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.7420384334896425
0.6694490095642712 ]
classical state: y= [ 0.          0.         -0.74203843  0.66944901]

On day 69 the results are
new imput teleported vector [-0.74249037  0.66985674  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.7649407256056041
0.6431552685723098 ]
classical state: y= [ 0.          0.         -0.76494073  0.64315527]

On day 70 the results are
new imput teleported vector [-0.76540661  0.64354698  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.7869118115395873
0.6160785778144151 ]
classical state: y= [ 0.          0.         -0.78691181  0.61607858]

On day 71 the results are
new imput teleported vector [-0.78739108  0.6164538   0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.8079249446295822
0.5882518992979403 ]
classical state: y= [ 0.          0.         -0.80792494  0.5882519 ]

On day 72 the results are
new imput teleported vector [-0.80841701  0.58861017  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.8279545443844712
0.5597091080333151 ]
classical state: y= [ 0.          0.         -0.82795454  0.55970911]

On day 73 the results are
new imput teleported vector [-0.82845881  0.56005     0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.8469762276246319
0.5304849507960566 ]
```

```
classical state: y= [ 0.          0.         -0.84697623  0.53048495]

On day 74 the results are
new imput teleported vector [-0.84749208  0.53080804  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.8649668381649737
0.500615003827529 ]
classical state: y= [ 0.          0.         -0.86496684  0.500615  ]

On day 75 the results are
new imput teleported vector [-0.86549365  0.5009199   0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.8819044750042914
0.47013562952594146 ]
classical state: y= [ 0.          0.         -0.88190448  0.47013563]

On day 76 the results are
new imput teleported vector [-0.8824416   0.47042197  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.897768518986612
0.43908393218032343 ]
classical state: y= [ 0.          0.         -0.89776852  0.43908393]

On day 77 the results are
new imput teleported vector [-0.8983153   0.43935136  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.9125396579020693
0.40749771280134195 ]
classical state: y= [ 0.          0.         -0.91253966  0.40749771]

On day 78 the results are
new imput teleported vector [-0.91309544  0.4077459   0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.926199909996767
0.37541542310396936 ]
classical state: y= [ 0.          0.         -0.92619991  0.37541542]

On day 79 the results are
new imput teleported vector [-0.92676401  0.37564407  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.9387326458629967
0.34287611869800305 ]
classical state: y= [ 0.          0.         -0.93873265  0.34287612]

On day 80 the results are
new imput teleported vector [-0.93930438  0.34308495  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.9501226086831732
```

```
0.3099194115434381 ]
classical state: y= [ 0.          0.          -0.95012261  0.30991941]

On day 81 the results are
new imput teleported vector [-0.95070128  0.31010817  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.9603559328028276
0.2765854217285531 ]
classical state: y= [ 0.          0.          -0.96035593  0.27658542]

On day 82 the results are
new imput teleported vector [-0.96094084  0.27675388  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.9694201606100616
0.24291472862943037 ]
classical state: y= [ 0.          0.          -0.96942016  0.24291473]

On day 83 the results are
new imput teleported vector [-0.97001059  0.24306268  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.9773042577009244
0.20894832151035758 ]
classical state: y= [ 0.          0.          -0.97730426  0.20894832]

On day 84 the results are
new imput teleported vector [-0.97789948  0.20907558  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.9839986263122056
0.1747275496252431 ]
classical state: y= [ 0.          0.          -0.98399863  0.17472755]

On day 85 the results are
new imput teleported vector [-0.98459793  0.17483397  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.9894951170053623
0.14029407188080434 ]
classical state: y= [ 0.          0.          -0.98949512  0.14029407]

On day 86 the results are
new imput teleported vector [-0.99009777  0.14037952  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.9937870385872816
0.10568980612278837 ]
classical state: y= [ 0.          0.          -0.99378704  0.10568981]

On day 87 the results are
new imput teleported vector [-0.9943923   0.10575418  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
```

```
full naive solution vector: y=[ 0          0          -0.9968691662558641
0.07095687810698043 ]
classical state: y= [ 0.          0.         -0.99686917  0.07095688]

On day 88 the results are
new imput teleported vector [-0.99747631  0.07100009  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.9987377479604723
0.03613757021710491 ]
classical state: y= [ 0.          0.         -0.99873775  0.03613757]

On day 89 the results are
new imput teleported vector [-0.99934603  0.03615958  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.9993905089695122
0.0012742699920570081 ]
classical state: y= [ 0.          0.         -0.99939051  0.00127427]

On day 90 the results are
new imput teleported vector [-0.99999919  0.00127505  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.9988266546395977
-0.033590581474877756 ]
classical state: y= [ 0.          0.         -0.99882665 -0.03359058]

On day 91 the results are
new imput teleported vector [-0.99943499 -0.03361104  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.9970468713829208
-0.06841454120199728 ]
classical state: y= [ 0.          0.         -0.99704687 -0.06841454]

On day 92 the results are
new imput teleported vector [-0.99765412 -0.06845621  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.9940533258316335
-0.10315521598745821 ]
classical state: y= [ 0.          0.         -0.99405333 -0.10315522]

On day 93 the results are
new imput teleported vector [-0.99465875 -0.10321804  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.9898496622002873
-0.13777031401695047 ]
classical state: y= [ 0.          0.         -0.98984966 -0.13777031]

On day 94 the results are
new imput teleported vector [-0.99045253 -0.13785422  0.          0.        ]
```

```
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.9844409978495199
-0.17221769634794382 ]
classical state: y= [ 0.         0.        -0.984441  -0.1722177]

On day 95 the results are
new imput teleported vector [-0.98504057 -0.17232259  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.9778339170564022
-0.20645542820783977 ]
classical state: y= [ 0.          0.          -0.97783392 -0.20645543]

On day 96 the results are
new imput teleported vector [-0.97842947 -0.20658117  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.9700364629990199
-0.24044183004357322 ]
classical state: y= [ 0.          0.          -0.97003646 -0.24044183]

On day 97 the results are
new imput teleported vector [-0.97062726 -0.24058827  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.9610581279650446
-0.27413552826052 ]
classical state: y= [ 0.          0.          -0.96105813 -0.27413553]

On day 98 the results are
new imput teleported vector [-0.96164346 -0.27430249  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.9509098417962392
-0.30749550558894545 ]
classical state: y= [ 0.          0.          -0.95090984 -0.30749551]

On day 99 the results are
new imput teleported vector [-0.95148899 -0.30768279  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999996
full naive solution vector: y=[ 0          0          -0.9396039585829274
-0.34048115101667675 ]
classical state: y= [ 0.          0.          -0.93960396 -0.34048115]

On day 100 the results are
new imput teleported vector [-0.94017622 -0.34068852  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.9271542416246565
-0.37305230922721727 ]
classical state: y= [ 0.          0.          -0.92715424 -0.37305231]

On day 101 the results are
```

```
new imput teleported vector [-0.92771892 -0.37327952  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.9135758466753356
-0.4051693294831105 ]
classical state: y= [ 0.          0.         -0.91357585 -0.40516933]

On day 102 the results are
new imput teleported vector [-0.91413226 -0.4054161   0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.8988853034932807
-0.4367931138950591 ]
classical state: y= [ 0.          0.         -0.8988853  -0.43679311]

On day 103 the results are
new imput teleported vector [-0.89943277 -0.43705914  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.8831004957185826
-0.46788516501802524 ]
classical state: y= [ 0.          0.         -0.8831005  -0.46788517]

On day 104 the results are
new imput teleported vector [-0.88363835 -0.46817013  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.8662406391023276
-0.49840763271637245 ]
classical state: y= [ 0.          0.         -0.86624064 -0.49840763]

On day 105 the results are
new imput teleported vector [-0.86676822 -0.49871119  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.848326258114171
-0.528323360241014 ]
classical state: y= [ 0.          0.         -0.84832626 -0.52832336]

On day 106 the results are
new imput teleported vector [-0.84884293 -0.52864514  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.8293791609567137
-0.5575959294624435 ]
classical state: y= [ 0.          0.         -0.82937916 -0.55759593]

On day 107 the results are
new imput teleported vector [-0.82988429 -0.55793553  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.8094224130171318
-0.5861897052046141 ]
classical state: y= [ 0.          0.         -0.80942241 -0.58618971]
```

```
On day 108 the results are
new imput teleported vector [-0.80991539 -0.58654672  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999996
full naive solution vector: y=[ 0          0          -0.7884803087883606
-0.6140698786256823 ]
classical state: y= [ 0.          0.         -0.78848031 -0.61406988]

On day 109 the results are
new imput teleported vector [-0.78896053 -0.61444388  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.7665783422940126
-0.6412025095928059 ]
classical state: y= [ 0.          0.         -0.76657834 -0.64120251]

On day 110 the results are
new imput teleported vector [-0.76704523 -0.64159303  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.7437431760530441
-0.6675545679994092 ]
classical state: y= [ 0.          0.         -0.74374318 -0.66755457]

On day 111 the results are
new imput teleported vector [-0.74419615 -0.66796114  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.720002608621944
-0.6930939739746302 ]
classical state: y= [ 0.          0.         -0.72000261 -0.69309397]

On day 112 the results are
new imput teleported vector [-0.72044113 -0.6935161   0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.6953855407539513
-0.7177896369359772 ]
classical state: y= [ 0.          0.         -0.69538554 -0.71778964]

On day 113 the results are
new imput teleported vector [-0.69580906 -0.71822681  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.6699219402165228
-0.7416114934376878 ]
classical state: y= [ 0.          0.         -0.66992194 -0.74161149]

On day 114 the results are
new imput teleported vector [-0.67032996 -0.74206317  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.643642805309842
-0.7645305437686744 ]
classical state: y= [ 0.          0.         -0.64364281 -0.76453054]
```

```
On day 115 the results are
new imput teleported vector [-0.64403482 -0.76499618  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.6165801271308166
-0.7865188872555411 ]
classical state: y= [ 0.          0.         -0.61658013 -0.78651889]

On day 116 the results are
new imput teleported vector [-0.61695565 -0.78699792  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.588766850628483
-0.8075497562276667 ]
classical state: y= [ 0.          0.         -0.58876685 -0.80754976]

On day 117 the results are
new imput teleported vector [-0.58912544 -0.80804159  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.5602368344982335
-0.8275975486030316 ]
classical state: y= [ 0.          0.         -0.56023683 -0.82759755]

On day 118 the results are
new imput teleported vector [-0.56057805 -0.8281016   0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.5310248099636794
-0.8466378590550887 ]
classical state: y= [ 0.          0.         -0.53102481 -0.84663786]

On day 119 the results are
new imput teleported vector [-0.53134823 -0.8471535   0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999996
full naive solution vector: y=[ 0          0          -0.5011663384963516
-0.8646475087227781 ]
classical state: y= [ 0.          0.         -0.50116634 -0.86464751]

On day 120 the results are
new imput teleported vector [-0.50147157 -0.86517412  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.47069776852467576
-0.8816045734274733 ]
classical state: y= [ 0.          0.         -0.47069777 -0.88160457]

On day 121 the results are
new imput teleported vector [-0.47098445 -0.88214151  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.43965619118494886
-0.8974884103625442 ]
```

```
classical state: y= [ 0.          0.         -0.43965619 -0.89748841]

On day 122 the results are
new imput teleported vector [-0.43992396 -0.89803503  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.4080793951681771
-0.9122796832230343 ]
classical state: y= [ 0.          0.         -0.4080794  -0.91227968]

On day 123 the results are
new imput teleported vector [-0.40832794 -0.91283531  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.376005820717733
-0.9259603857448382 ]
classical state: y= [ 0.          0.         -0.37600582 -0.92596039]

On day 124 the results are
new imput teleported vector [-0.37623483 -0.92652434  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.3434745128338456
-0.938513863624773 ]
classical state: y= [ 0.          0.         -0.34347451 -0.93851386]

On day 125 the results are
new imput teleported vector [-0.34368371 -0.93908546  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.3105250737418837
-0.9499248347948074 ]
classical state: y= [ 0.          0.         -0.31052507 -0.94992483]

On day 126 the results are
new imput teleported vector [-0.3107142  -0.95050339  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.2771976146822973
-0.9601794080258055 ]
classical state: y= [ 0.          0.         -0.27719761 -0.96017941]

On day 127 the results are
new imput teleported vector [-0.27736644 -0.9607642   0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.24353270708090835
-0.969265099838114 ]
classical state: y= [ 0.          0.         -0.24353271 -0.9692651 ]

On day 128 the results are
new imput teleported vector [-0.24368103 -0.96985543  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.2095713331589837
```

```
-0.977170849698409 ]
classical state: y= [ 0.          0.         -0.20957133 -0.97717085]

On day 129 the results are
new imput teleported vector [-0.20969897 -0.97776599  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          -0.17535483604323399
-0.9838870334843302 ]
classical state: y= [ 0.          0.         -0.17535484 -0.98388703]

On day 130 the results are
new imput teleported vector [-0.17546164 -0.98448627  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.14092486943645444
-0.9894054752004673 ]
classical state: y= [ 0.          0.         -0.14092487 -0.98940548]

On day 131 the results are
new imput teleported vector [-0.1410107  -0.99000807  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.10632334691008406
-0.9937194569314738 ]
classical state: y= [ 0.          0.         -0.10632335 -0.99371946]

On day 132 the results are
new imput teleported vector [-0.1063881  -0.99432468  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.07159239088041496
-0.9968237270201636 ]
classical state: y= [ 0.          0.         -0.07159239 -0.99682373]

On day 133 the results are
new imput teleported vector [-0.07163599 -0.99743084  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.036774281330559655
-0.9987145064606623 ]
classical state: y= [ 0.          0.         -0.03677428 -0.99871451]

On day 134 the results are
new imput teleported vector [-0.03679668 -0.99932277  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          -0.0019114043406051108
-0.9993894934988009 ]
classical state: y= [ 0.          0.         -0.0019114  -0.99938949]

On day 135 the results are
new imput teleported vector [-0.00191257 -0.99999817  0.          0.        ]
square of the norm of teleported vector imput 1.0
```

```
full naive solution vector: y=[ 0          0          0.032953799511389895
-0.998847866434177 ]
classical state: y= [ 0.          0.          0.0329538  -0.99884787]

On day 136 the results are
new imput teleported vector [ 0.03297387 -0.99945621  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          0.06777888681474593
-0.9970902846204536 ]
classical state: y= [ 0.          0.          0.06777889 -0.99709028]

On day 137 the results are
new imput teleported vector [ 0.06782017 -0.99769756  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.10252146299495618
-0.9941188876626967 ]
classical state: y= [ 0.          0.          0.10252146 -0.99411889]

On day 138 the results are
new imput teleported vector [ 0.1025839  -0.99472436  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          0.13713923392303257
-0.9899372928127061 ]
classical state: y= [ 0.          0.          0.13713923 -0.98993729]

On day 139 the results are
new imput teleported vector [ 0.13722276 -0.99054021  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.17159005740257274
-0.9845505905655259 ]
classical state: y= [ 0.          0.          0.17159006 -0.98455059]

On day 140 the results are
new imput teleported vector [ 0.17169456 -0.98515023  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          0.2058319944718709
-0.9779653384625017 ]
classical state: y= [ 0.          0.          0.20583199 -0.97796534]

On day 141 the results are
new imput teleported vector [ 0.20595736 -0.97856097  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.23982336045861932
-0.9701895531084013 ]
classical state: y= [ 0.          0.          0.23982336 -0.97018955]

On day 142 the results are
new imput teleported vector [ 0.23996942 -0.97078045  0.          0.        ]
```

```
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          0.2735227757250488
-0.9612327004123443 ]
classical state: y= [ 0.          0.          0.27352278 -0.9612327 ]

On day 143 the results are
new imput teleported vector [ 0.27368936 -0.96181814  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          0.3068892160417322
-0.9511056840644134 ]
classical state: y= [ 0.          0.          0.30688922 -0.95110568]

On day 144 the results are
new imput teleported vector [ 0.30707613 -0.95168495  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.3398820625287307
-0.9398208322619587 ]
classical state: y= [ 0.          0.          0.33988206 -0.93982083]

On day 145 the results are
new imput teleported vector [ 0.34008907 -0.94039323  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.3724611511032843
-0.9273918827017853 ]
classical state: y= [ 0.          0.          0.37246115 -0.92739188]

On day 146 the results are
new imput teleported vector [ 0.372688   -0.92795671  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.40458682137385105
-0.9138339658564614 ]
classical state: y= [ 0.          0.          0.40458682 -0.91383397]

On day 147 the results are
new imput teleported vector [ 0.40483323 -0.91439054  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.4362199649209751
-0.8991635865551317 ]
classical state: y= [ 0.          0.          0.43621996 -0.89916359]

On day 148 the results are
new imput teleported vector [ 0.43648564 -0.89971122  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.46732207290620614
-0.8833986038912401 ]
classical state: y= [ 0.          0.          0.46732207 -0.8833986 ]

On day 149 the results are
```

```
new imput teleported vector [ 0.4676067  -0.88393664  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.4978552829511154
-0.8665582094816321 ]
classical state: y= [ 0.          0.          0.49785528 -0.86655821]

On day 150 the results are
new imput teleported vector [ 0.4981585  -0.86708599  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.5277824252293384
-0.8486629041034985 ]
classical state: y= [ 0.          0.          0.52778243 -0.8486629 ]

On day 151 the results are
new imput teleported vector [ 0.52810387 -0.84917978  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.5570670677155372
-0.8297344727376095 ]
classical state: y= [ 0.          0.          0.55706707 -0.82973447]

On day 152 the results are
new imput teleported vector [ 0.55740635 -0.83023982  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          0.5856735605361882
-0.8097959580482053 ]
classical state: y= [ 0.          0.          0.58567356 -0.80979596]

On day 153 the results are
new imput teleported vector [ 0.58603026 -0.81028916  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          0.61356707936822
-0.7888716323318379 ]
classical state: y= [ 0.          0.          0.61356708 -0.78887163]

On day 154 the results are
new imput teleported vector [ 0.61394077 -0.78935209  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.640713667832656
-0.7669869679693204 ]
classical state: y= [ 0.          0.          0.64071367 -0.76698697]

On day 155 the results are
new imput teleported vector [ 0.64110389 -0.7674541   0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          0.6670802788316597
-0.7441686064167324 ]
classical state: y= [ 0.          0.          0.66708028 -0.74416861]
```

```
On day 156 the results are
new imput teleported vector [ 0.66748656 -0.74462184  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          0.6926348147786626
-0.7204443257732503 ]
classical state: y= [ 0.          0.          0.69263481 -0.72044433]

On day 157 the results are
new imput teleported vector [ 0.69305666 -0.72088311  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.7173461666725962
-0.6958430069652769 ]
classical state: y= [ 0.          0.          0.71734617 -0.69584301]

On day 158 the results are
new imput teleported vector [ 0.71778307 -0.69626681  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          0.7411842519686571
-0.6703945985880331 ]
classical state: y= [ 0.          0.          0.74118425 -0.6703946 ]

On day 159 the results are
new imput teleported vector [ 0.74163567 -0.6708029   0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          0.764120051199518
-0.644130080447416 ]
classical state: y= [ 0.          0.          0.76412005 -0.64413008]

On day 160 the results are
new imput teleported vector [ 0.76458544 -0.64452239  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          0.7861256433023865
-0.617081425846506 ]
classical state: y= [ 0.          0.          0.78612564 -0.61708143]

On day 161 the results are
new imput teleported vector [ 0.78660443 -0.61745726  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.8071742396089144
-0.5892815626626359 ]
classical state: y= [ 0.          0.          0.80717424 -0.58928156]

On day 162 the results are
new imput teleported vector [ 0.80766585 -0.58964046  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          0.8272402164565795
-0.5607643332623949 ]
classical state: y= [ 0.          0.          0.82724022 -0.56076433]
```

```
On day 163 the results are
new imput teleported vector [ 0.82774405 -0.56110587  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.8462991463818387
-0.5315644533033778 ]
classical state: y= [ 0.          0.          0.84629915 -0.53156445]

On day 164 the results are
new imput teleported vector [ 0.84681458 -0.5318882   0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          0.8643278278570783
-0.501717469472825 ]
classical state: y= [ 0.          0.          0.86432783 -0.50171747]

On day 165 the results are
new imput teleported vector [ 0.86485425 -0.50202304  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999996
full naive solution vector: y=[ 0          0          0.8813043135351608
-0.4712597162146025 ]
classical state: y= [ 0.          0.          0.88130431 -0.47125972]

On day 166 the results are
new imput teleported vector [ 0.88184107 -0.47154674  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.8972079369671967
-0.4402282714972049 ]
classical state: y= [ 0.          0.          0.89720794 -0.44022827]

On day 167 the results are
new imput teleported vector [ 0.89775438 -0.44049639  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999996
full naive solution vector: y=[ 0          0          0.9120193377609864
-0.4086609116766143 ]
classical state: y= [ 0.          0.          0.91201934 -0.40866091]

On day 168 the results are
new imput teleported vector [ 0.9125748  -0.40890981  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.9257204851495425
-0.3765960655089821 ]
classical state: y= [ 0.          0.          0.92572049 -0.37659607]

On day 169 the results are
new imput teleported vector [ 0.92628429 -0.37682543  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          0.9382946999409723
-0.3440727673690978 ]
```

```
classical state: y= [ 0.         0.         0.9382947  -0.34407277]

On day 170 the results are
new imput teleported vector [ 0.93886617 -0.34428232  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0         0         0.9497266748230179
-0.31113060973161083 ]
classical state: y= [ 0.         0.         0.94972667 -0.31113061]

On day 171 the results are
new imput teleported vector [ 0.95030511 -0.3113201   0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0         0         0.9600024929975163
-0.2778096949728382 ]
classical state: y= [ 0.         0.         0.96000249 -0.27780969]

On day 172 the results are
new imput teleported vector [ 0.96058718 -0.27797889  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0         0         0.9691096451221289
-0.2441505865518494 ]
classical state: y= [ 0.         0.         0.96910965 -0.24415059]

On day 173 the results are
new imput teleported vector [ 0.96969988 -0.24429929  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0         0         0.9770370445386638
-0.21019425963023666 ]
classical state: y= [ 0.         0.         0.97703704 -0.21019426]

On day 174 the results are
new imput teleported vector [ 0.97763211 -0.21032228  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0         0         0.9837750407695158
-0.17598205119070806 ]
classical state: y= [ 0.         0.         0.98377504 -0.17598205]

On day 175 the results are
new imput teleported vector [ 0.98437421 -0.17608923  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0         0         0.989315431265735
-0.14155560971520648 ]
classical state: y= [ 0.         0.         0.98931543 -0.14155561]

On day 176 the results are
new imput teleported vector [ 0.98991797 -0.14164182  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0         0         0.9936514713924662
```

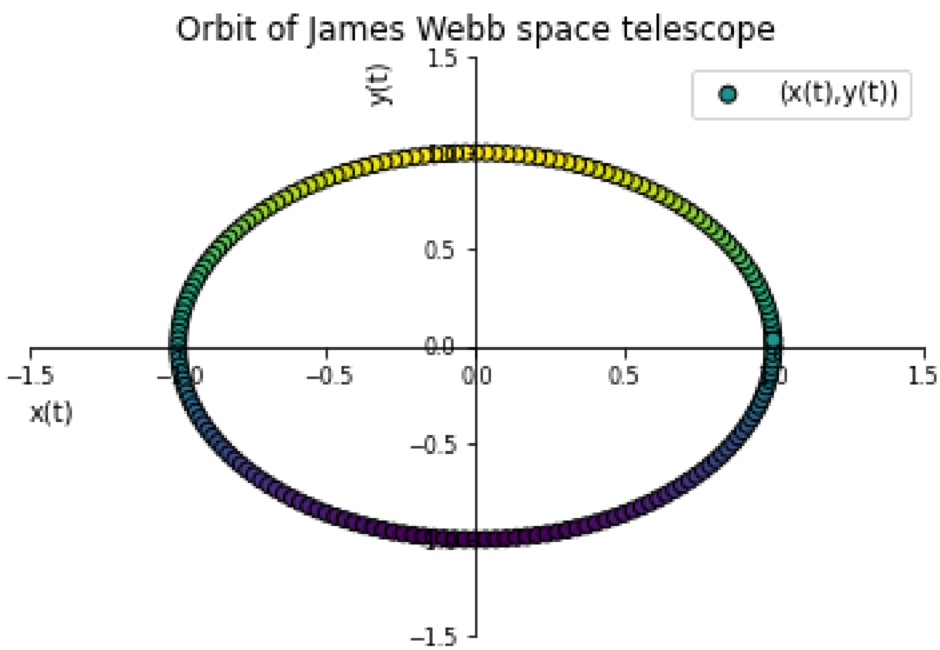
```
-0.10695684448382727 ]
classical state: y= [ 0.          0.          0.99365147 -0.10695684]

On day 177 the results are
new imput teleported vector [ 0.99425665 -0.10702199  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.9967778826395765
-0.07222787455624832 ]
classical state: y= [ 0.          0.          0.99677788 -0.07222787]

On day 178 the results are
new imput teleported vector [ 0.99738497 -0.07227186  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999996
full naive solution vector: y=[ 0          0          0.9986908590474898
-0.03741097749778684 ]
classical state: y= [ 0.          0.          0.99869086 -0.03741098]

On day 179 the results are
new imput teleported vector [ 0.99929911 -0.03743376  0.          0.        ]
square of the norm of teleported vector imput 1.0
full naive solution vector: y=[ 0          0          0.999388071840394
-0.002548537912494583 ]
classical state: y= [ 0.          0.          0.99938807 -0.00254854]

On day 180 the results are
new imput teleported vector [ 0.99999675 -0.00255009  0.          0.        ]
square of the norm of teleported vector imput 0.9999999999999998
full naive solution vector: y=[ 0          0          0.9988686722612031
0.03231700415404895 ]
classical state: y= [0.         0.         0.99886867 0.032317  ]
```

[20]:

```
from matplotlib import pyplot as plt

fig = plt.figure()
ax = fig.add_subplot(111)
ax.spines['left'].set_position('zero')
ax.spines['right'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['top'].set_color('none')
plt.axis([-1.5,1.5,-1.5,1.5])
ax.xaxis.set_ticks_position('bottom')
ax.yaxis.set_ticks_position('left')
ax.set_ylabel('y(t)', loc='top')
ax.set_xlabel('x(t)', loc='left')
plt.xticks(fontsize=8)
plt.yticks(fontsize=8)
```

```
plt.title('Orbit of James Webb space telescope')
#plt.grid()
plt.scatter(xv1, yv1,c=cyv1, ec='k', label='(x(t),y(t))')
ax.legend()
# changing the size of figure to 2X2
```

[20]: <matplotlib.legend.Legend at 0x7f5a9b13cbb0>



What the code does is it measures the position (x,y) of a satellite and it prepares the valid quantum state, telleports it to earth and then it performs the linear differential equation solver for the period of T= 6 months. “Performing” another measurment we have other input state constructed from the position (x_1, y_1) of another satellite.

As we have seen a quantum circuit can simulate the state that comes from the quantum state prepared by the satellite that scans one satellite orbiting earth and that efficiently encodes its positions. Then transmit its quantum measurements to a quantum computer on good old Earth and then perform the the linear differential equation solver. In order to do so, we have to consider valid quantum conditions early mencioned on Task 1.

We have plotted all the data for it’s trayectory over the period of T=180 days= 6 months (30 days per month) performing a measurment every day. We found that the plotts of the $\vec{x} = (x(t), y(t))$ form a unitary circle, which we can consider as the trayectory of this telescope.

For the quantum teleportation algorithm we had to consider only the components we were interested in, an extension of this work can be made by considering the full state preparation that the satellite makes every 24 hours and then sends it to earth. Other extension of this work comes from considering the full naive solution vector components (4 components) as a quantum imput following the results from Rigolin, G. (2005). Quantum teleportation of an arbitrary two-qubit state and its relation to multipartite entanglement. Physical Review A, 71(3), 032303.

```
[ ]:
```