Physics 245 Homework #4

Nathan Solomon

October 31, 2024

Problem 0.1.

For parts (a) and (b), see the jupyter notebook. For part (c), note that the value of Ω I found ($6638152s^{-1}$) times the pulse duration $500ns$ is 3.319, so this is approximately a π-pulse, meaning we have almost perfectly flipped the original state from $|0\rangle$ to $|1\rangle$. If we suppose that this was a pulse in the y direction, the final state is

$$R_y(3.319)\,|0\rangle = \begin{bmatrix} -0.0886253 \\ 0.99606504 \end{bmatrix}.$$

Problem 0.2.

(a)

$$\begin{aligned} F_x &= ma = m\ddot{x} \\ F_x &= -kx \\ \ddot{x} &= -\frac{k}{m}x \end{aligned}$$

If we let $\omega = \sqrt{k/m}$, then $\ddot{x} = -\omega^2 x$.

(b) The equation is satisfied by any x of the form

$$x(t) = A\sin(\omega t + \phi).$$

This has two degrees of freedom (A and ϕ), so it is the most general solution to the second-order differential equation.

(c) Kinetic energy at time t is

$$\frac{m\dot{x}^2}{2} = \frac{mA^2\omega^2}{2}\cos^2(\omega t + \phi).$$

(d) The force on the spring is $-kx$ and the potential energy when $x = 0$ is zero, so the potential energy is is

$$\frac{kx^2}{2} = \frac{(m\omega^2)x^2}{2} = \frac{mA^2\omega^2}{2}\sin^2(\omega t + \phi).$$

(e) The sum of the kinetic and potential energies is $mA^2\omega^2/2$.

(f) For parts (f) through (i), see the notebook at the end of this document.

Problem 0.3.

Recall that

$$x = \sqrt{\frac{\hbar}{2m\omega}}\left(a + a^\dagger\right)$$
$$p = -i\sqrt{\frac{\hbar m\omega}{2}}\left(a - a^\dagger\right)$$

(a)

$$\begin{aligned}
\langle x\rangle &= \langle n|\, x\, |n\rangle \\
&= \sqrt{\frac{\hbar}{2m\omega}}\, \langle n| \left(a + a^\dagger\right) |n\rangle \\
&\propto \langle n|\, a\, |n\rangle + \langle n|\, a^\dagger\, |n\rangle \\
&= \sqrt{n}\, \langle n|\, n-1\rangle + \sqrt{n+1}\, \langle n|\, n+1\rangle \\
&= 0 \\
\langle p\rangle &= \langle n|\, p\, |n\rangle \\
&= -i\sqrt{\frac{\hbar m\omega}{2}}\, \langle n| \left(a - a^\dagger\right) |n\rangle \\
&\propto \langle n|\, a\, |n\rangle - \langle n|\, a^\dagger\, |n\rangle \\
&= \sqrt{n}\, \langle n|\, n-1\rangle - \sqrt{n+1}\, \langle n|\, n+1\rangle \\
&= 0
\end{aligned}$$

(b)

$$\begin{aligned}
\sigma_x^2 &= \left\langle x^2\right\rangle - \langle x\rangle^2 \\
&= \langle n|\, x^2\, |n\rangle \\
&= \frac{\hbar}{2m\omega}\, \langle n| \left(aa + aa^\dagger + a^\dagger a + a^\dagger a^\dagger\right) |n\rangle \\
&= \frac{\hbar}{2m\omega}\, \langle n| \left(aa^\dagger + a^\dagger a\right) |n\rangle \\
&= \frac{\hbar}{2m\omega}\, \langle n| \left(2a^\dagger a + [a, a^\dagger]\right) |n\rangle \\
&= \frac{\hbar}{2m\omega}\, (2n+1) \\
\sigma_x &= \sqrt{\frac{(2n+1)\hbar}{2m\omega}} \\
\sigma_p^2 &= \left\langle p^2\right\rangle - \langle p\rangle^2 \\
&= \langle n|\, p^2\, |n\rangle \\
&= -\frac{\hbar m\omega}{2}\, \langle n| \left(aa - aa^\dagger - a^\dagger a + a^\dagger a^\dagger\right) |n\rangle \\
&= \frac{\hbar m\omega}{2}\, \langle n| \left(aa^\dagger + a^\dagger a\right) |n\rangle \\
\sigma_x &= \sqrt{\frac{(2n+1)\hbar m\omega}{2}} \\
\sigma_x\sigma_p &= \frac{\hbar}{2}(2n+1) \geq \frac{\hbar}{2}
\end{aligned}$$

(c) See jupyter notebook. Or don't bother, because I just plotted the expected values of x and p. I considered adding error bars, but they all overlapped, so that graph wasn't interesting either.

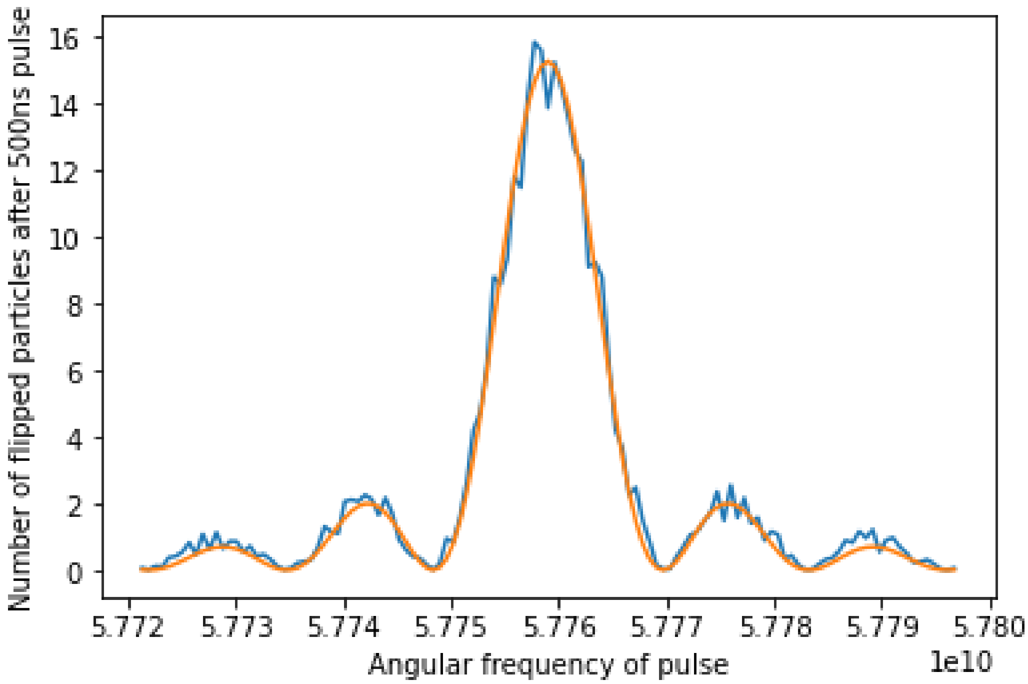
notebook

October 31, 2024

[1]:
```
import qutip as qt
import matplotlib.pyplot as plt
import numpy as np
from scipy.optimize import curve_fit
```

[2]:
```
# Problem 1
data = np.genfromtxt('RabiData.csv', delimiter=',').T
# convert frequency to angular frequency
omega = data[0] * 2 * np.pi
population = data[1]
t = 500e-9

def p(w, amplitude, w_0, Omega):
    return amplitude * (Omega**2 / (Omega**2 + (w-w_0)**2)) * \
        np.sin( np.sqrt(Omega**2 + (w-w_0)**2) * t/2 )**2

p0=[16, 5.776e10, 5e6]
popt, pcov = curve_fit(p, omega, population, p0)
print(f"omega_0 = {popt[1]} +/- {np.sqrt(pcov[1,1])}")
print(f"Omega = {popt[2]} +/- {np.sqrt(pcov[2,2])}")
plt.plot(omega, population)
plt.plot(omega, p(omega, *popt))
plt.xlabel("Angular frequency of pulse")
plt.ylabel("Number of flipped particles after 500ns pulse")
plt.show()

phi = float(popt[2]) * t
print(f"This is a {float(popt[2])} * {t} = {phi} pulse (approximately a␣
↪pi-pulse).")
print(f"The operator corresponding to a rotation around the y axis by {phi}␣
↪radians is")
Rz = (qt.sigmay() * phi * (0-0.5j)).expm()
final_state = Rz * qt.basis(2, 0)
print(Rz)
print(f"So the new state is {final_state}")
```

```
omega_0 = 57759010005.79707 +/- 44234.48845876736
Omega = 6638152.234703342 +/- 111473.39543443453
```



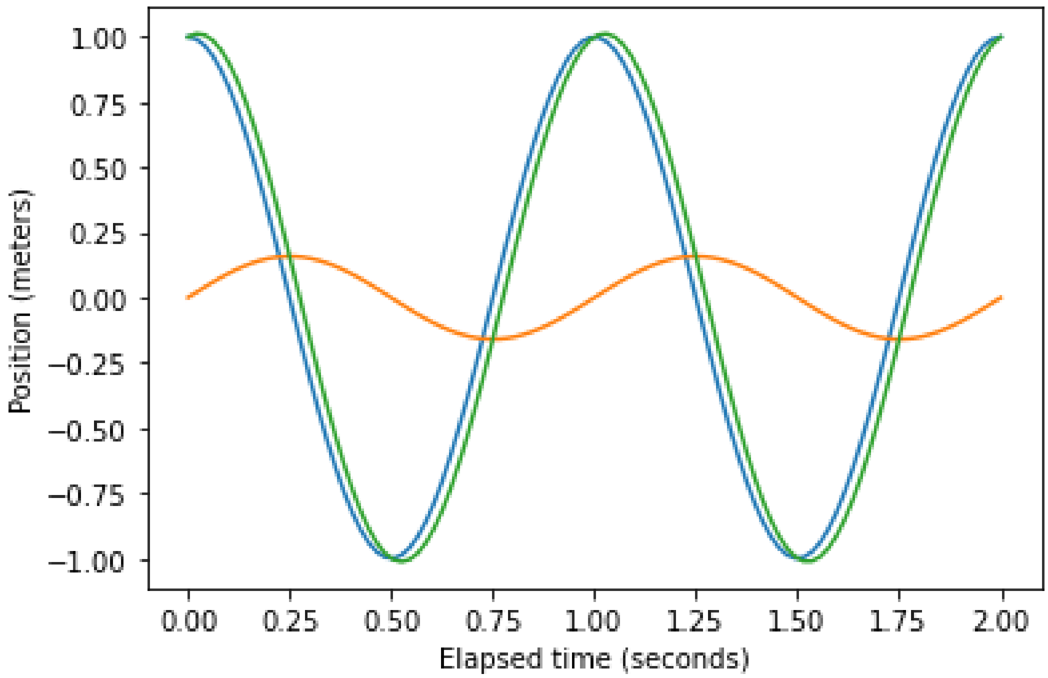

```
This is a 6638152.234703342 * 5e-07 = 3.319076117351671 pulse (approximately a
pi-pulse).
The operator corresponding to a rotation around the y axis by 3.319076117351671
radians is
Quantum object: dims=[[2], [2]], shape=(2, 2), type='oper', dtype=Dense,
isherm=False
Qobj data =
[[-0.0886253  -0.99606504]
 [ 0.99606504 -0.0886253 ]]
So the new state is Quantum object: dims=[[2], [1]], shape=(2, 1), type='ket',
dtype=Dense
Qobj data =
[[-0.0886253 ]
 [ 0.99606504]]
```

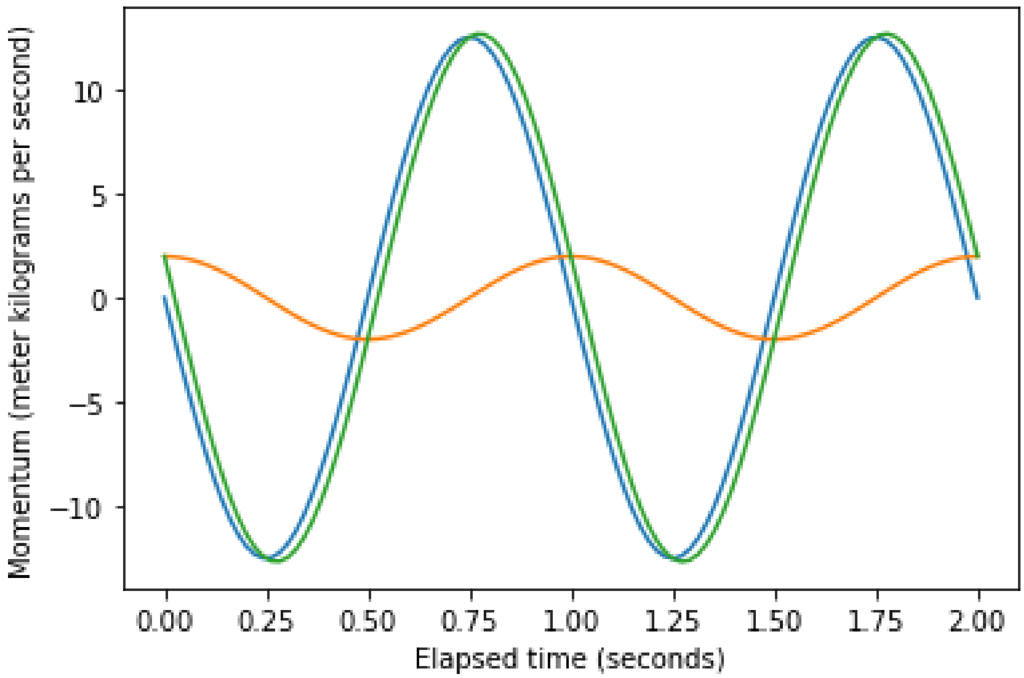
[3]:
```
# Problem 2
omega = 2 * np.pi  # Hertz
m = 2  # kilograms

def plot_data(x_0, v_0):
    # Returns a list of times, positions, and momenta for plotting
    kinetic_energy = m * v_0**2 / 2
    potential_energy = m * omega**2 * x_0**2 / 2
    total_energy = kinetic_energy + potential_energy
```

```python
    amplitude = np.sqrt(2 * total_energy / (m * omega**2))
    phase = np.arctan2(x_0, v_0 / omega)

    t = np.linspace(0, 2, 100)
    x = amplitude * np.sin(omega * t + phase)
    p = m * amplitude * omega * np.cos(omega * t + phase)
    return t, x, p

print("For all of these graphs, case (i) is in blue, case (ii) is in orange,␣
 ↪case (iii) is in green")
t_i  , x_i  , p_i   = plot_data(1, 0)
t_ii , x_ii , p_ii  = plot_data(0, 1)
t_iii, x_iii, p_iii = plot_data(1, 1)

plt.plot(t_i  , x_i  )
plt.plot(t_ii , x_ii )
plt.plot(t_iii, x_iii)
plt.xlabel("Elapsed time (seconds)")
plt.ylabel("Position (meters)")
plt.show()

plt.plot(t_i  , p_i  )
plt.plot(t_ii , p_ii )
plt.plot(t_iii, p_iii)
plt.xlabel("Elapsed time (seconds)")
plt.ylabel("Momentum (meter kilograms per second)")
plt.show()

plt.plot(x_i  , p_i  )
plt.plot(x_ii , p_ii )
plt.plot(x_iii, p_iii)
plt.xlabel("Position (meters)")
plt.ylabel("Momentum (meter kilograms per second)")
plt.show()

x_i   *= np.sqrt(m * omega / 2)
x_ii  *= np.sqrt(m * omega / 2)
x_iii *= np.sqrt(m * omega / 2)
p_i   /= np.sqrt(m * omega * 2)
p_ii  /= np.sqrt(m * omega * 2)
p_iii /= np.sqrt(m * omega * 2)
plt.plot(x_i  , p_i  )
plt.plot(x_ii , p_ii )
plt.plot(x_iii, p_iii)
plt.xlabel("Scaled position")
plt.ylabel("Scaled momentum")
plt.show()
```

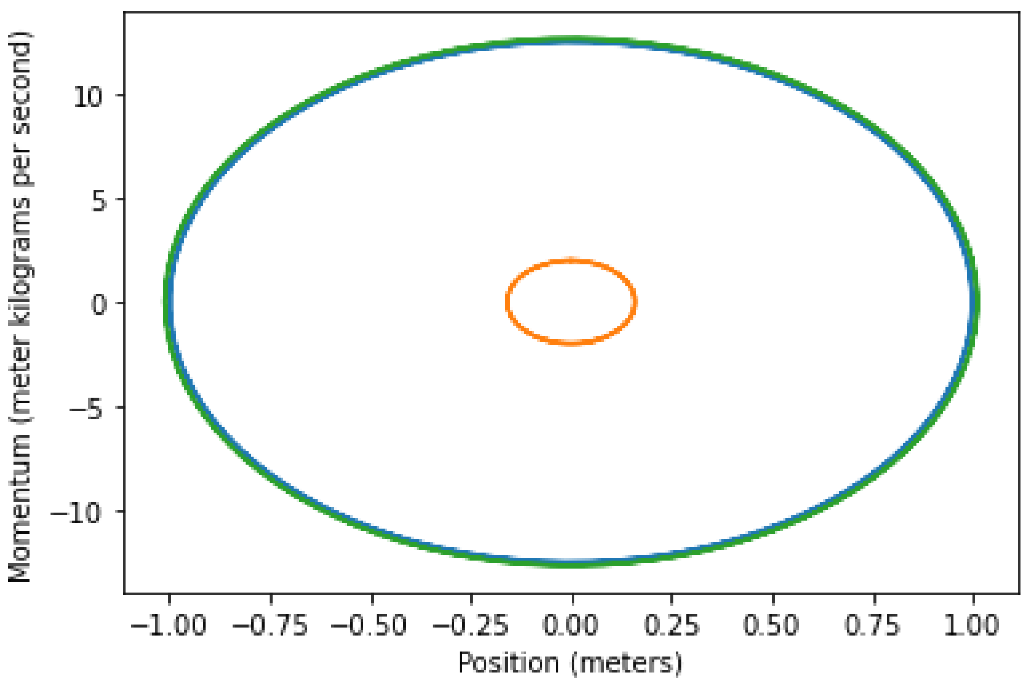
```
plt.plot(x_i  , p_i  )
plt.plot(x_ii , p_ii )
plt.plot(x_iii, p_iii)
plt.xlabel("Re(a)")
plt.ylabel("Im(a)")
plt.show()
```

For all of these graphs, case (i) is in blue, case (ii) is in orange, case (iii) is in green

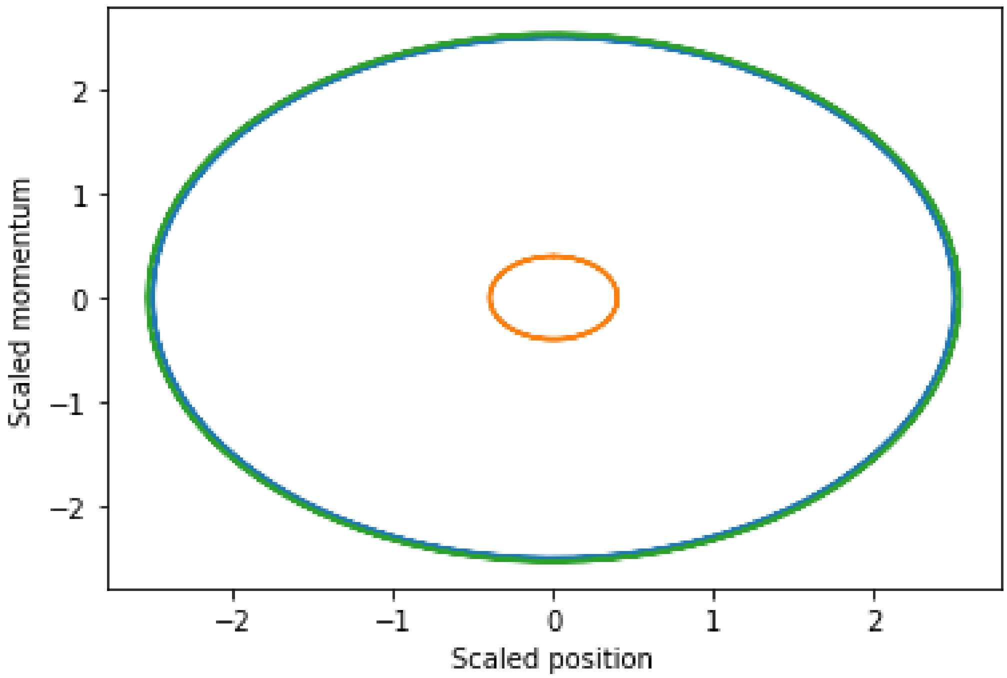


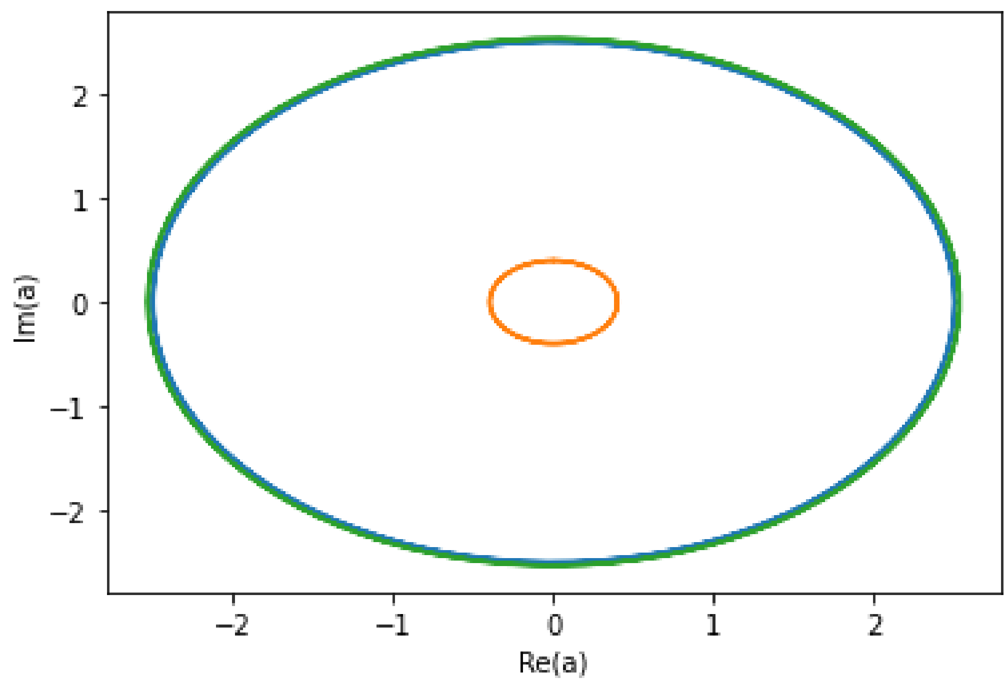


Momentum (meter kilograms per second)
10
5
0
−5
−10
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
Elapsed time (seconds)



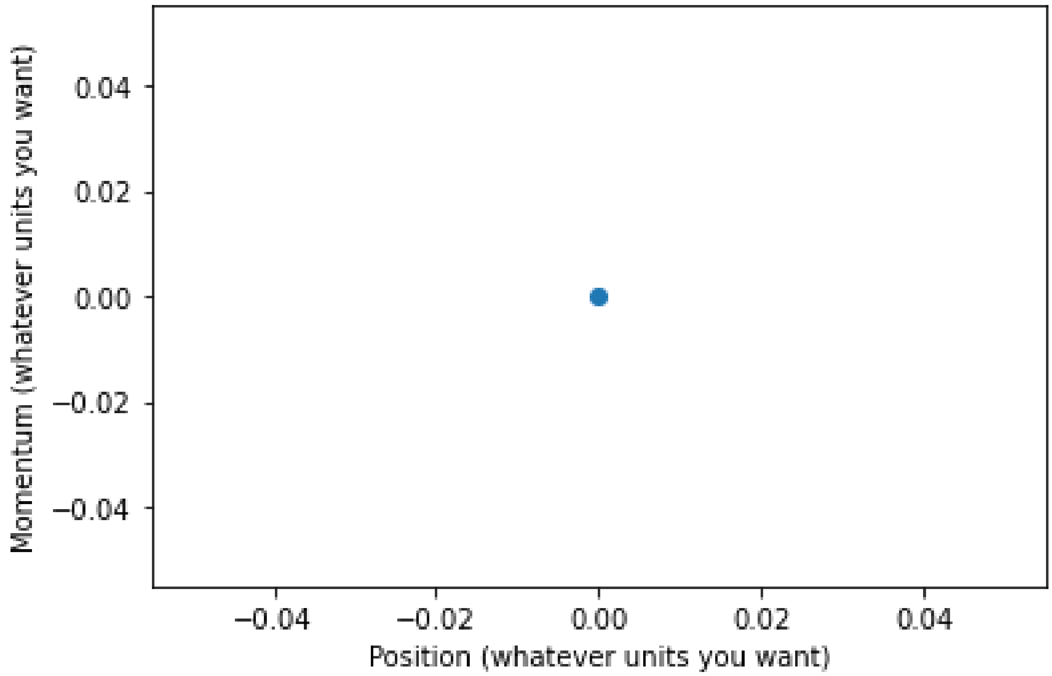
Momentum (meter kilograms per second)
10
5
0
−5
−10
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
Position (meters)





[4]:
```
# Problem 3
plt.xlabel("Position (whatever units you want)")
```

```
plt.ylabel("Momentum (whatever units you want)")
plt.scatter(0, 0)
plt.show()
```



[5]:

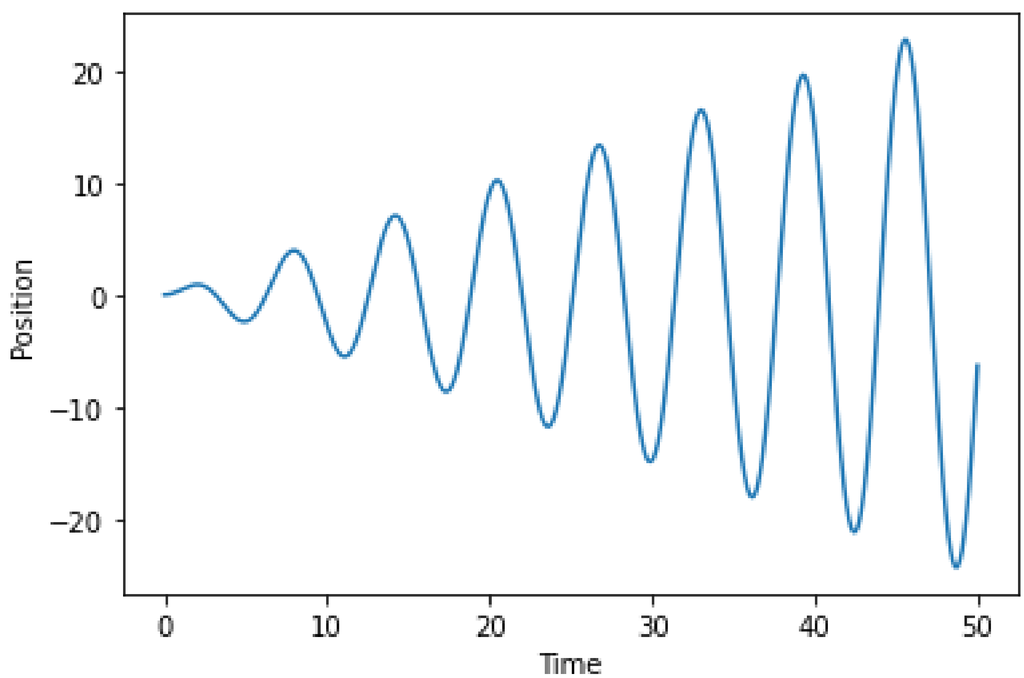
```
# Problem 4
N = 6
print(f"For this problem, ignore |n> if n > {N-1}\n")
a = qt.destroy(N)

print("Part (a):\n")
zero = qt.basis(N, 0)
print(a * zero)
print(a.dag() * zero)

print("\nPart (b):\n")
three = qt.basis(N, 3)
four = qt.basis(N, 4)
print(a * three)
print(a.dag() * four)

print("\nPart (c):\n")
hbar = 1
m = 1
omega = 2 * np.pi
```

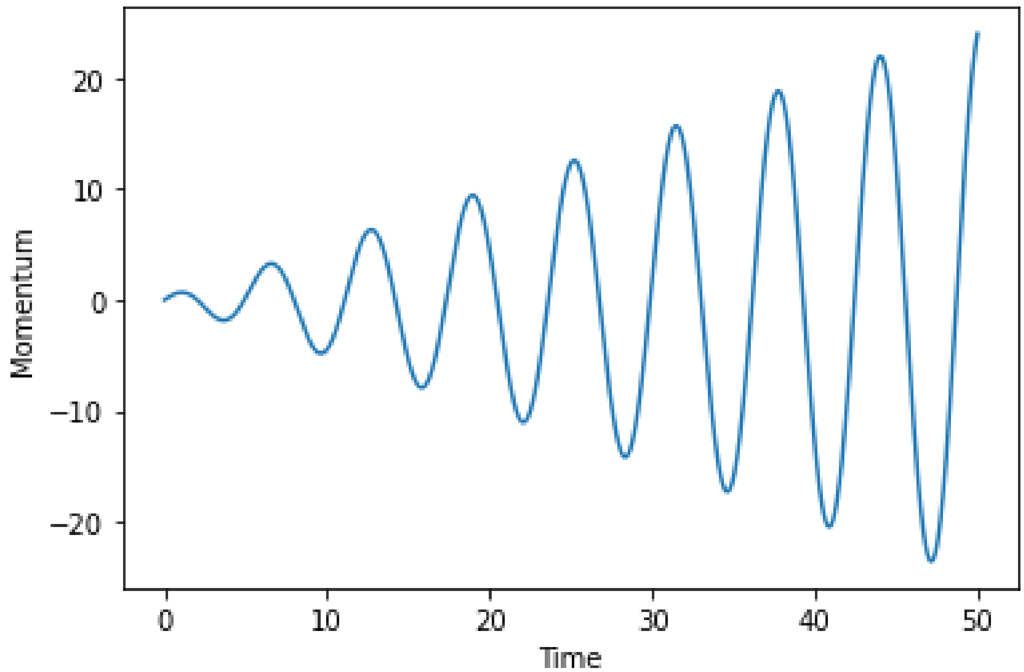
```
x = np.sqrt(hbar / 2 / m / omega) * (a + a.dag())
p = (0-1j) * np.sqrt(hbar * m * omega / 2) * (a - a.dag())
Psi = (zero + three).unit()
print(f"<x> = {qt.expect(x, Psi)}")
print(f"<p> = {qt.expect(p, Psi)}")
```

```
For this problem, ignore |n> if n > 5

Part (a):

Quantum object: dims=[[6], [1]], shape=(6, 1), type='ket', dtype=Dense
Qobj data =
[[0.]
 [0.]
 [0.]
 [0.]
 [0.]
 [0.]]
Quantum object: dims=[[6], [1]], shape=(6, 1), type='ket', dtype=Dense
Qobj data =
[[0.]
 [1.]
 [0.]
 [0.]
 [0.]
 [0.]]

Part (b):

Quantum object: dims=[[6], [1]], shape=(6, 1), type='ket', dtype=Dense
Qobj data =
[[0.        ]
 [0.        ]
 [1.73205081]
 [0.        ]
 [0.        ]
 [0.        ]]
Quantum object: dims=[[6], [1]], shape=(6, 1), type='ket', dtype=Dense
Qobj data =
[[0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [2.23606798]]

Part (c):
```

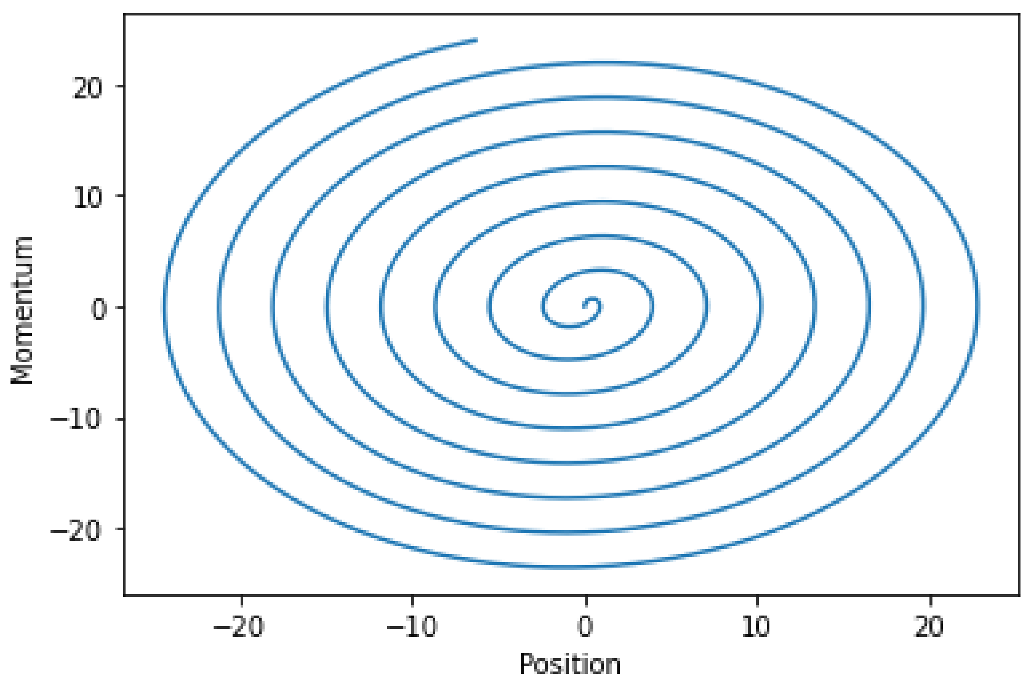
```
<x> = 0.0
<p> = 0.0
```

[60]:
```
# Problem 5.a
# Assume F_0, mass, and omega are all 1
dt = 0.01
t = [0]
x = [0]
p = [0]
for i in range(5000):
    t.append(t[-1] + dt)
    p.append(p[-1] + (np.cos(t[-1]) - x[-1]) * dt)
    x.append(x[-1] + p[-1] * dt)
plt.plot(t, x)
plt.xlabel("Time")
plt.ylabel("Position")
plt.show()
plt.plot(t, p)
plt.xlabel("Time")
plt.ylabel("Momentum")
plt.show()
plt.plot(x, p)
plt.xlabel("Position")
plt.ylabel("Momentum")
plt.show()
```





Momentum
Time

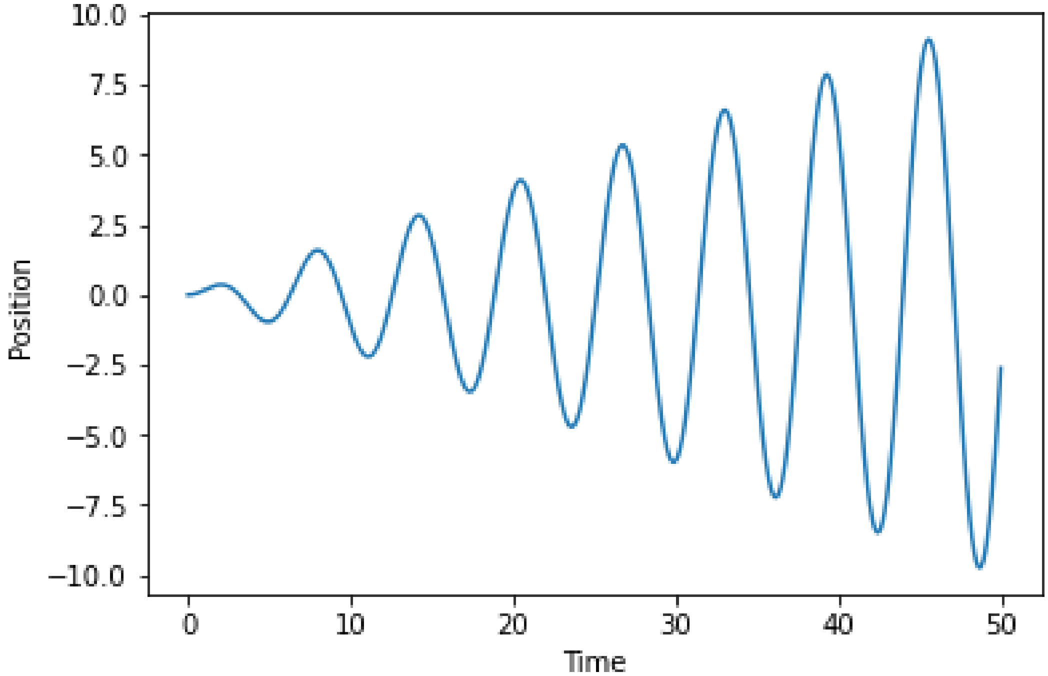


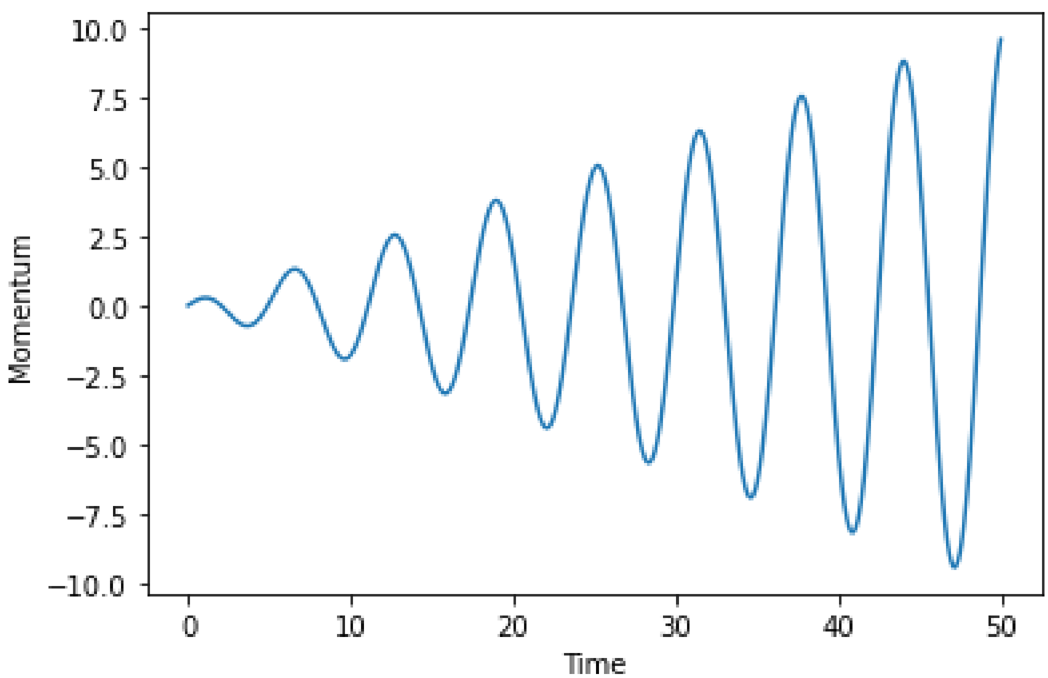
Momentum
Position

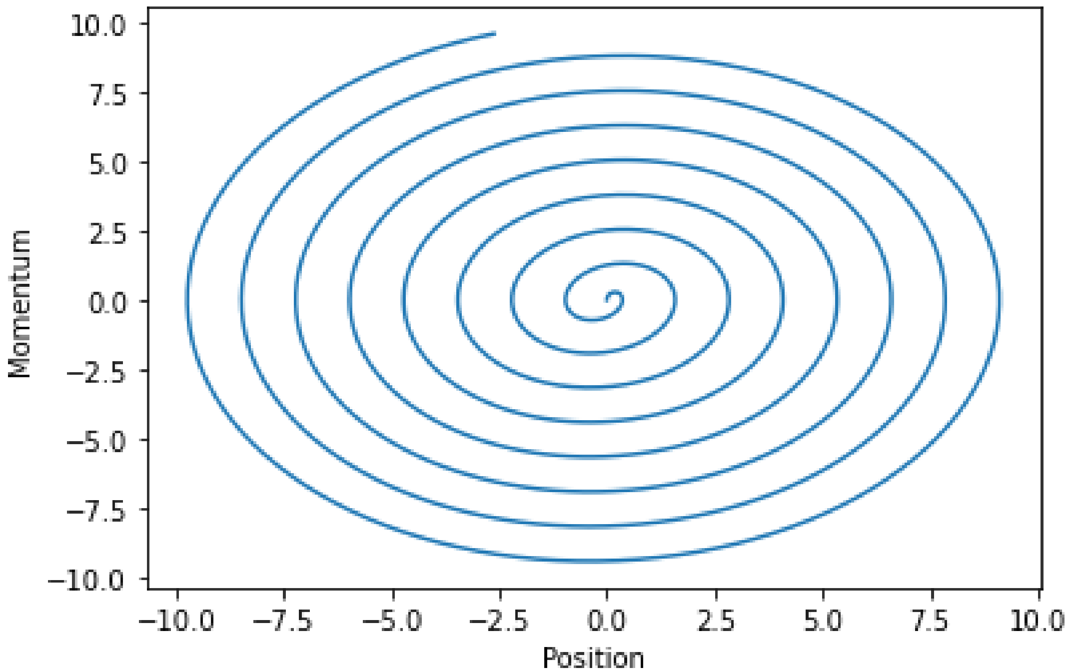
[16]:

```
# Problem 5.b
# The higher N is and the smaller time is, the better the approximation
N = 70
print(f"For this problem, ignore |n> if n > {N-1}\n")
a = qt.destroy(N)
x = a + a.dag()
p = (0-1j) * (a - a.dag())
F_0 = 0.2
def H(t):
    return (qt.num(N) + qt.qeye(N) / 2) - x * np.cos(t) * F_0

initial_state = qt.basis(N, 0)
times = np.linspace(0, 50, 1000)
evolved_states = qt.sesolve(H, initial_state, times, e_ops=[x, p])
positions = evolved_states.expect[0]
momenta = evolved_states.expect[1]
plt.plot(times, positions)
plt.xlabel("Time")
plt.ylabel("Position")
plt.show()
plt.plot(times, momenta)
plt.xlabel("Time")
plt.ylabel("Momentum")
plt.show()
plt.plot(positions, momenta)
plt.xlabel("Position")
plt.ylabel("Momentum")
plt.show()
```

```
For this problem, ignore |n> if n > 69
```


10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
Position
0
10
20
30
40
50
Time


10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
Momentum
0
10
20
30
40
50
Time


10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
Momentum
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
Position

[ ]:

Phys 245 Quantum Computation
Homework 4

1. *[40 + 5] Calibrating a new qubit!* On the class website (on the page with Lecture 7) there is a file which has the experimental result of a Rabi spectroscopy experiment. Specifically, the qubit was initially prepared in $|0\rangle$ and the probability of finding the qubit in $|1\rangle$ measured. The Rabi pulse was applied for 500 ns. Like real data, this experimental result is noisy. By fitting the expected lineshape, extract values for:
    a. [20] The qubit frequency ω_o and the Rabi frequency Ω
    b. [20] What are the uncertainties on these extracted parameters?
    c. [Bonus + 5] What is this qubit?
    (Hint for problem 1: Be careful with the factors of 2π. Remember, in all our work we have been using $H/\hbar$, therefore ω_o and Ω are angular frequencies, but the data, as is typical in the lab, is in Hz.)

2. *[50] Harmonic Oscillators are classic!* In this problem, we'll develop intuition about classical harmonic oscillators, which will prepare us for tackling the quantum problem. Suppose we have a mass on a spring. As the mass is moved from its equilibrium position ($x = 0$) the spring provides a restoring force of $F_x = -kx$, where k is the spring constant.
    a. [5] Show that the motion of the mass is given by a differential equation of the form $\frac{d^2x}{dt^2} = -\omega^2 x$ and determine ω.
    b. [5] Solve the differential equation from part (a) and write it as a single sine or cosine with an amplitude and phase to be found from initial conditions.
    c. [5] Use the solution from (b) to calculate the kinetic energy as a function of time.
    d. [5] Use the solution from (b) to calculate the potential energy as a function of time.
    e. [5] Use the solution from (b) to calculate the total energy as a function of time.
    f. [10] For $\omega = 2\pi \times 1$ Hz and m = 2 kg, make a plot of the position x(t) and momentum p(t) as a function of time for the initial conditions:
        i. x = 1, v = 0
        ii. x = 0, v = 1
        iii. x = 1, v = 1
    g. [10] For the same three cases as part (f), make a phase space plot of the evolution. That is, make a parametric plot of (x(t),p(t)) over one cycle of the oscillation.
    h. [5] Define a scaled position and momentum as:

    $$\tilde{x} = \sqrt{\frac{m\omega}{2}}\,x \text{ and } \tilde{p} = \frac{p}{\sqrt{2m\omega}}$$

    And remake the plots of part(g)
    i. [10] Construct a complex variable a as $a = \tilde{x} + \imath\tilde{p}$ and write it in phasor notation. Now make a plot where the real part of a is on the x axis and the imagine part of a is on the y axis.

3. *[25] Quantum Harmonic Oscillators by hand.* For the states $|n\rangle$:

a. [10] Calculate $\langle \hat{x} \rangle$ and $\langle \hat{p} \rangle$
b. [10] Calculate the uncertainties, $\sigma_{\hat{x}}$ and $\sigma_{\hat{p}}$.
c. [5] Sketch these states on a phase space plot (x vs p) for several values of n.

4. [30] Calculate *Quantum Harmonic Oscillators by QuTip.* For QHO (just set $\hbar = m = 1$ and $\omega = 2\pi$), use QuTip to do the following:
   a. [10] Show the result of $\hat{a}|0\rangle$ and $\hat{a}^{\dagger}|0\rangle$.
   b. [10] Show the result of $\hat{a}|3\rangle$ and $\hat{a}^{\dagger}|4\rangle$.
   c. [10] Calculate $\langle \hat{x} \rangle$ and $\langle \hat{p} \rangle$ for $|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |3\rangle)$.

5. *[40] A driven oscillator*. Suppose a harmonic oscillator is driven by a force on resonance, i.e. $F_o \cos \omega t$. Pick your own parameters for the strength of the force and the harmonic oscillator. Use numerical integration to solve the following.
   a. [20] Solve Newton's equation to find the evolution of the classical harmonic oscillator under this driving force. Plot x vs t, p vs t, and make a phase space plot of (x(t),p(t)). (If you prefer to do this one analytically, that's fine too. But still make the plots.)
   b. [20] Solve Schrodinger's equation (e.g. use sesolve in QuTip) to find the evolution of the quantum harmonic oscillator under this driving force. Plot x vs t, p vs t, and make a phase space plot of (x(t),p(t))