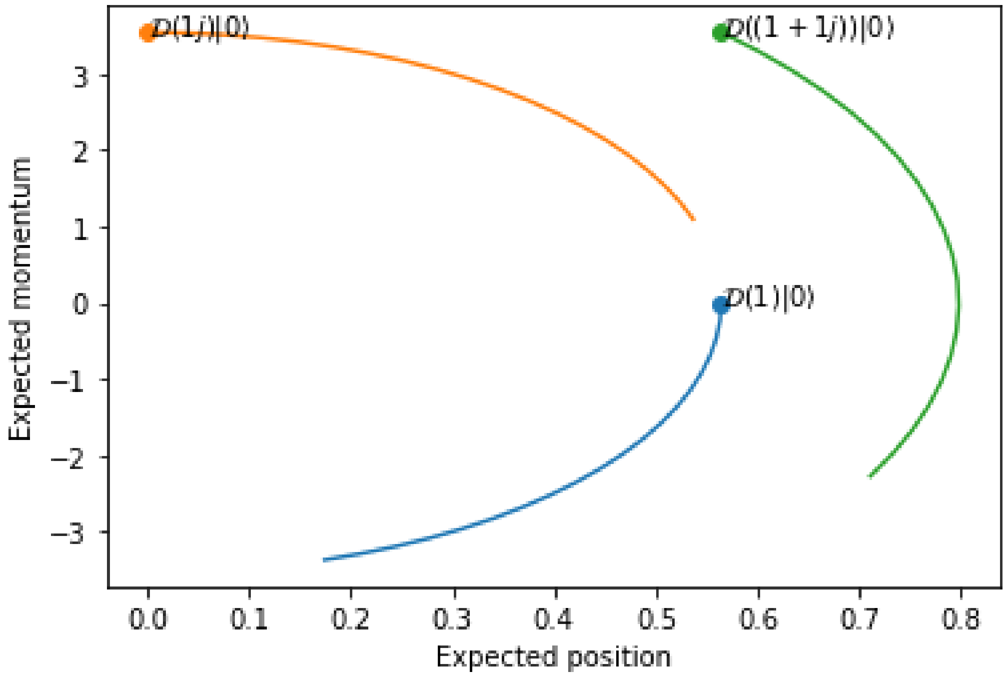
notebook

November 5, 2024

[1]:
```python
import qutip as qt
import matplotlib.pyplot as plt
import numpy as np
```

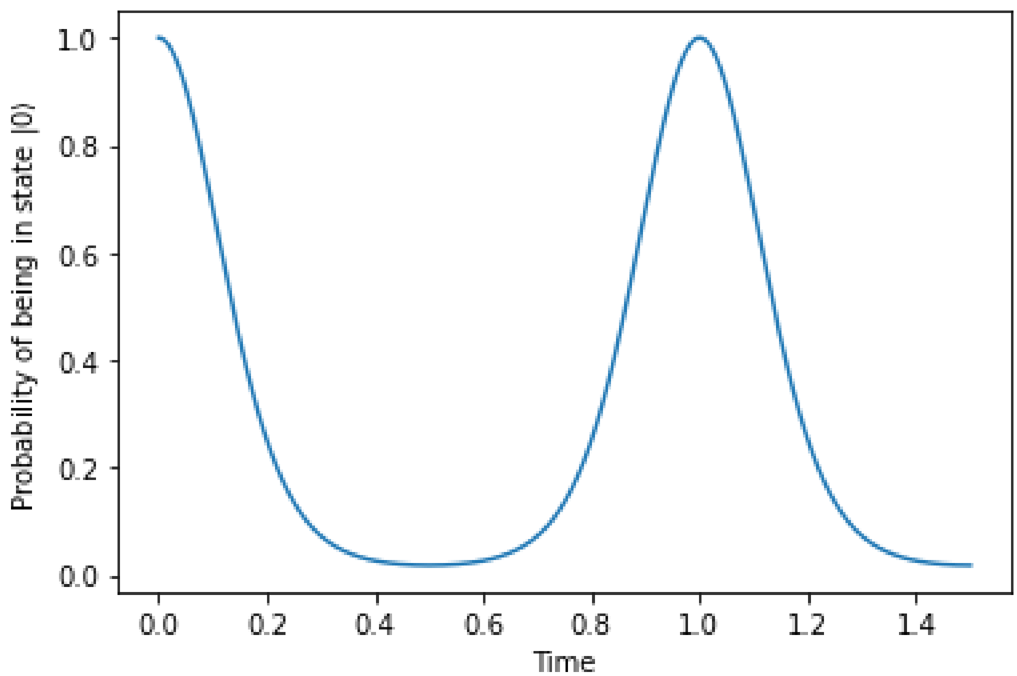
[2]:
```python
# Problem 2
N = 10
m = 1
omega = 2 * np.pi
hbar = 1

a = qt.destroy(N)
x = np.sqrt(hbar / 2 / m / omega) * (a + a.dag())
p = (-1j) * np.sqrt(hbar * m * omega / 2) * (a - a.dag())

for alpha in [1, 1j, 1+1j]:
    Psi = qt.coherent(N, alpha)
    expected_x = qt.expect(x, Psi)
    expected_p = qt.expect(p, Psi)
    plt.scatter(expected_x, expected_p)
    plt.annotate(f"$\\mathcal {{ D }} ({alpha})| 0 \\rangle$", [expected_x,␣
 ↪expected_p])
# Don't call plt.show() yet, because we want to see how these states change in␣
 ↪time

def U(t):
    return (-1j * omega * t * (qt.num(N) + qt.qeye(N) / 2)).expm()

for alpha in [1, 1j, 1+1j]:
    expected_x = []
    expected_p = []
    for t in np.linspace(0, 0.2, 20):
        Psi = U(t) * qt.coherent(N, alpha)
        expected_x.append(qt.expect(x, Psi))
        expected_p.append(qt.expect(p, Psi))
    plt.plot(expected_x, expected_p)
plt.xlabel("Expected position")
plt.ylabel("Expected momentum")
```

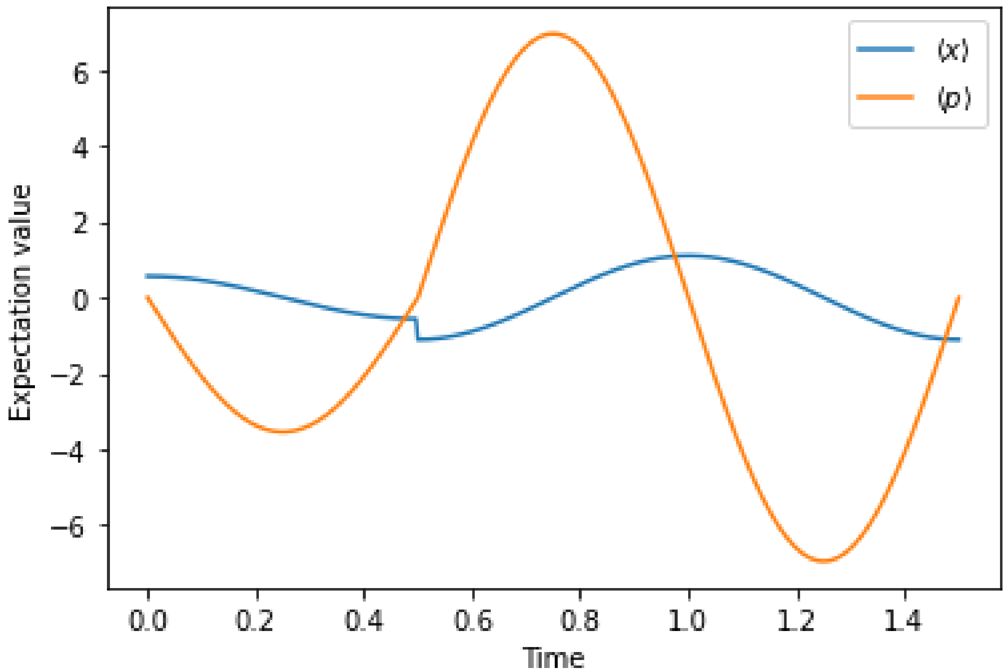
```
plt.show()
```



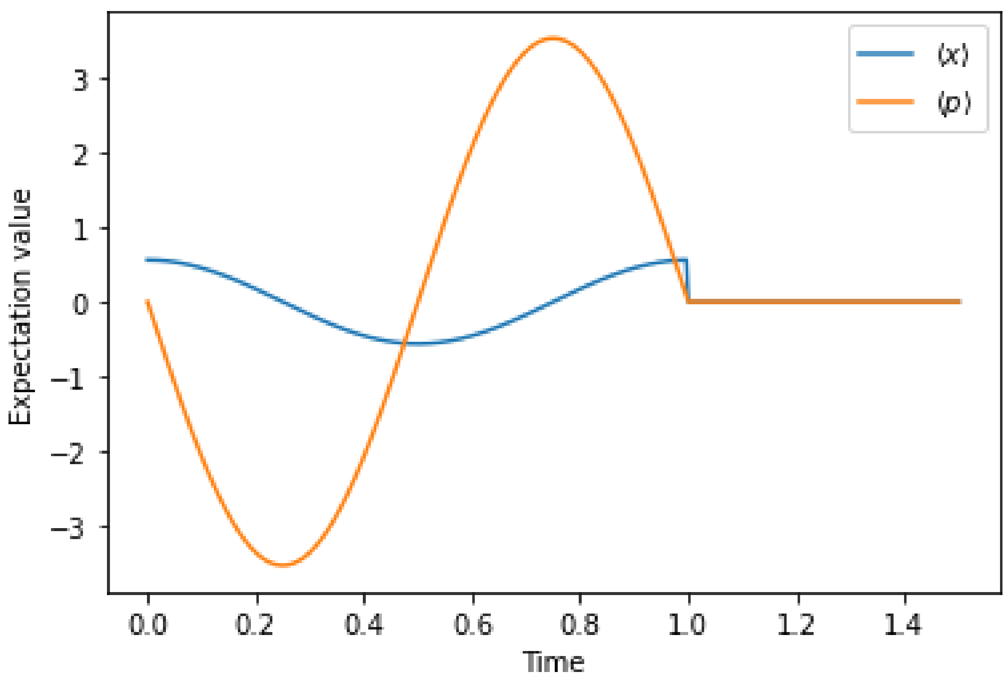

```
[3]: # Problem 3
N = 10
m = 1
omega = 2 * np.pi
hbar = 1

a = qt.destroy(N)
x = np.sqrt(hbar / 2 / m / omega) * (a + a.dag())
p = -1j * np.sqrt(hbar * m * omega / 2) * (a - a.dag())
H = hbar * omega * (qt.num(N) + qt.qeye(N) / 2)

Psi = qt.coherent(N, 1)

def U(t, t_w):
    if t < t_w:
        return (-1j*H*t/hbar).expm()
    return (-1j*H*(t-t_w)/hbar).expm() * qt.displace(N,-1) * (-1j*H*t_w/hbar).
 ↪expm()

times = np.linspace(0, 1.5, 400)
probabilities = [qt.expect(qt.basis(N, 0).proj(), U(t_w, t_w) * Psi) for t_w in␣
 ↪times]
```

```
plt.plot(times, probabilities)
plt.xlabel("Time")
plt.ylabel("Probability of being in state $|0\\rangle$")
plt.show()

def plot(t_w):
    states = [U(t, t_w) * Psi for t in times]
    positions = [qt.expect(x, state) for state in states]
    momenta = [qt.expect(p, state) for state in states]
    plt.xlabel("Time")
    plt.ylabel("Expectation value")
    plt.plot(times, positions, label = "$\\langle x \\rangle$")
    plt.plot(times, momenta, label = "$\\langle p \\rangle$")
    plt.legend()
    plt.show()

plot(0.5)
plot(1)
```





Expectation value
Time
$\langle x \rangle$
$\langle p \rangle$



Expectation value
Time
$\langle x \rangle$
$\langle p \rangle$