notebook

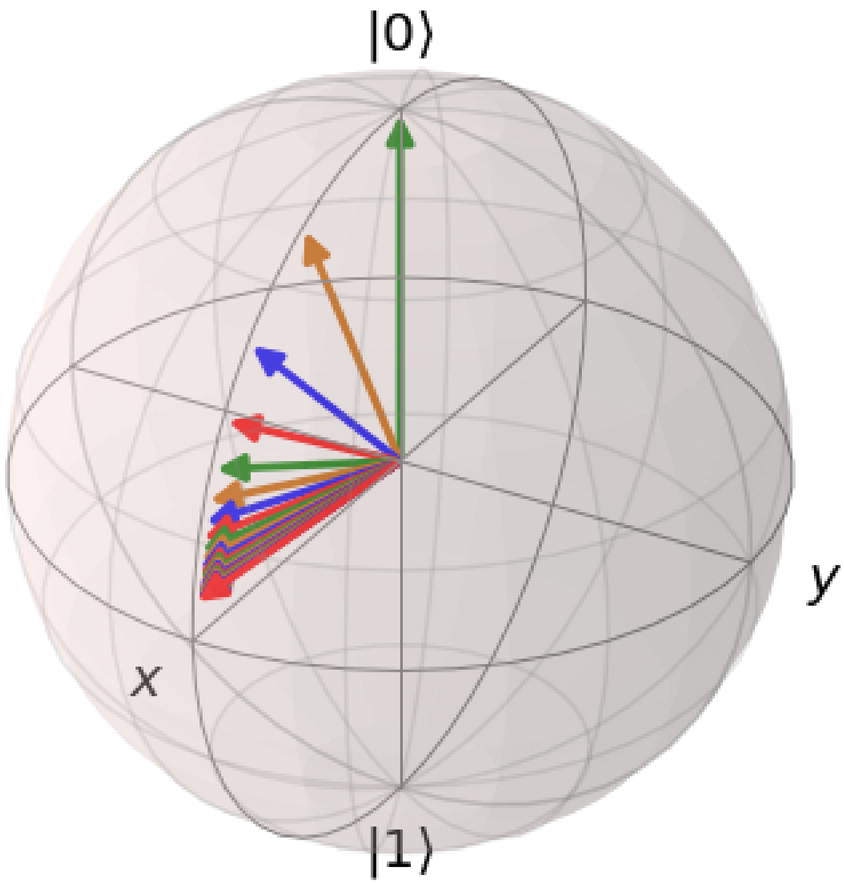
October 17, 2024

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

[2]:
```
# Problem 2
b = qt.Bloch()

for ratio in np.linspace(0, 10, 20):
    # Rewrite formula so that you don't have to divide by (B_x / B_z), in case␣
 ↪it's zero
    positive_eigenvector = qt.basis(2, 0) + qt.basis(2, 1) * ratio / (1 + np.
 ↪sqrt(1 + ratio ** 2))
    b.add_states(positive_eigenvector.unit())

b.show()
```



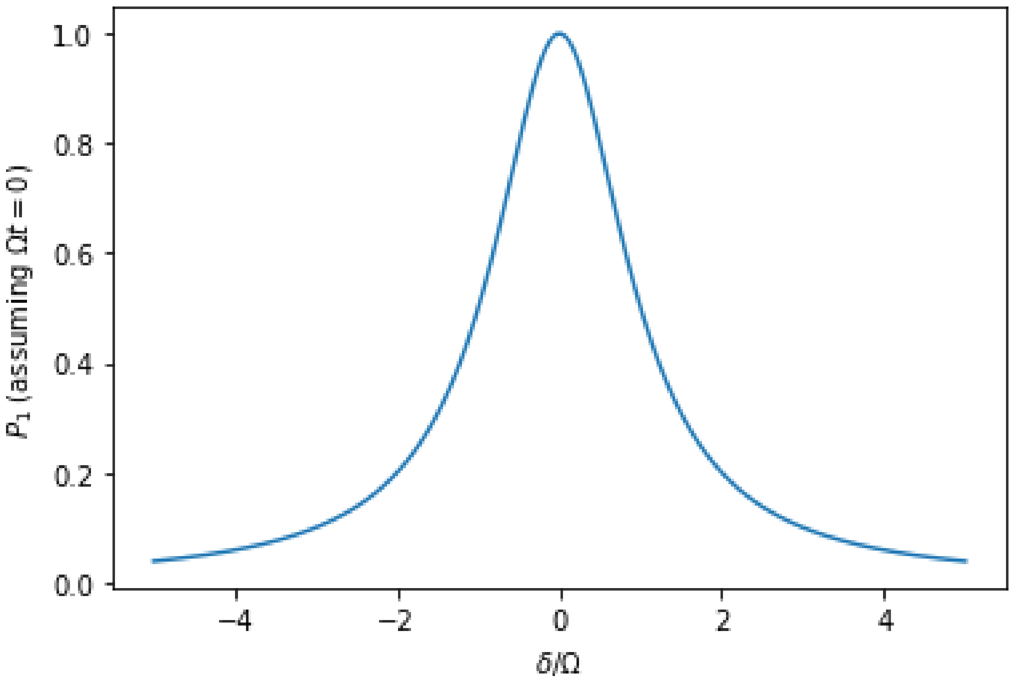
[3]:
```
# Problem 3
print(qt.sigmax().eigenstates(), "\n\n")
print(qt.sigmay().eigenstates(), "\n\n")
print(qt.sigmaz().eigenstates())
```

```
(array([-1.,  1.]), array([Quantum object: dims=[[2], [1]], shape=(2, 1),
type='ket', dtype=Dense
       Qobj data =
       [[-0.70710678]
        [ 0.70710678]]                                                      ,
       Quantum object: dims=[[2], [1]], shape=(2, 1), type='ket', dtype=Dense
       Qobj data =
       [[0.70710678]
        [0.70710678]]                                                       ],
      dtype=object))
```

```
(array([-1.,  1.]), array([Quantum object: dims=[[2], [1]], shape=(2, 1),
type='ket', dtype=Dense
       Qobj data =
       [[-0.70710678+0.j        ]
        [ 0.        +0.70710678j]]                                          ,
       Quantum object: dims=[[2], [1]], shape=(2, 1), type='ket', dtype=Dense
       Qobj data =
       [[-0.70710678+0.j        ]
        [ 0.        -0.70710678j]]                                          ],
      dtype=object))
```

```
(array([-1.,  1.]), array([Quantum object: dims=[[2], [1]], shape=(2, 1),
type='ket', dtype=Dense
       Qobj data =
       [[ 0.]
        [-1.]]                                                              ,
       Quantum object: dims=[[2], [1]], shape=(2, 1), type='ket', dtype=Dense
       Qobj data =
       [[-1.]
        [-0.]]                                                              ],
      dtype=object))
```

[4]:
```
# Problem 5
plt.xlabel("$\delta / \Omega$")
plt.ylabel("$P_1$ (assuming $\Omega t=0$)")
x = np.linspace(-5, 5, 200)
y = 1 / (1 + x**2)
plt.plot(x, y)
plt.show()
```



1.0
0.8
0.6
0.4
0.2
0.0
P_1 (assuming $\Omega t = 0$)
−4
−2
0
2
4
δ/Ω