QCQT-QE8 Quantum Control

Prof. Paspalakis Emmanuel
Final Project - Topalidis Dimitrios (60687)

1 Preface

This project is done in the context of the course ”QE8 - Quantum Control” of the ”MSC in Quantum Computing and Quantum Technologies - DUTH” with the professor Mr. Emmanuel Paspalakis.
It is based on the following three tutorials for the QuTip python library:

- **Schrödinger Equation Solver: Larmor precession** `https://nbviewer.org/urls/qutip.org/qutip-tutorials/tutorials-v4/time-evolution/002_larmor-precession.ipynb`
- **Master Equation Solver: Single-Qubit Dynamics** `https://nbviewer.org/urls/qutip.org/qutip-tutorials/tutorials-v4/time-evolution/003_qubit-dynamics.ipynb`
- **Floquet Solvers** `https://nbviewer.org/urls/qutip.org/qutip-tutorials/tutorials-v4/time-evolution/011_floquet_solver.ipynb`

The deliverable is to document the basic theory for each concept and demonstrate the use of QuTip for the respective examples.
You can find all of the code written for this project in the following repository: `https://github.com/topalidisd-qcqt-duth/qe8-final-project`
There is also a shared jupyter notebook, where you can run the code directly: `https://colab.research.google.com/drive/1xPYatGO86dwSthS65inqtcSFOStjnRPh?usp=sharing`
The Qutip configuration used to develop and run the code of this project is shown below:

```
QuTiP: Quantum Toolbox in Python
================================
Copyright (c) QuTiP team 2011 and later.
Current admin team: Alexander Pitchford, Nathan Shammah, Shahnawaz Ahmed, Neill Lambert, Eric
Giguere, Boxi Li, Simon Cross, Asier Galicia, Paul Menczel, and Patrick Hopf.
Board members: Daniel Burgarth, Robert Johansson, Anton F. Kockum, Franco Nori and Will Zeng.
Original developers: R. J. Johansson & P. D. Nation.
Previous lead developers: Chris Granade & A. Grimsmo.
Currently developed through wide collaboration. See https://github.com/qutip for details.

QuTiP Version:      5.1.1
Numpy Version:      2.2.5
Scipy Version:      1.15.2
Cython Version:     None
Matplotlib Version: 3.10.1
Python Version:     3.11.12
Number of CPUs:     16
BLAS Info:          Generic
INTEL MKL Ext:      None
Platform Info:      Linux (x86_64)
Installation path:  /home/topalidis/.pyenv/versions/3.11.12/lib/python3.11/site-packages/qutip

Installed QuTiP family packages
-------------------------------

No QuTiP family packages installed.

================================================================================
Please cite QuTiP in your publication.
================================================================================
For your convenience a bibtex reference can be easily generated using `qutip.cite()`
```

2 Schrödinger Equation Solver: Larmor precession

2.1 Constant magnetic field

For a quantum system with state $|\psi(t)\rangle$ and Hamiltonian $H(t)$, the Time Dependent Schrödinger Equation reads as:

$$i\hbar\frac{d}{dt}|\psi(t)\rangle = H(t)|\psi(t)\rangle$$

When a spin-$\frac{1}{2}$ particle (e.g. an electron) is placed in a constant magnetic field $\mathbf{B}$ along the z-axis, the interaction Hamiltonian due to the magnetic moment $\mu = \gamma\mathbf{S}$ is:

$$H = -\gamma B_0\frac{\hbar}{2}\sigma_z = \frac{\hbar}{2}\omega_0\sigma_z$$

where $\omega_0 = -\gamma B_0$ is the **Larmor frequency**.
This Hamiltonian results in a precession of the state vector around the z-axis on the Bloch sphere, known as **Larmor precession**.
If we set for simplicity $\frac{\hbar}{2}\omega_0 = 1$ then we can write the Hamiltonian as:

$$H(t) \propto \sigma_z$$

We remind that $\sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
We setup a arbitrary qubit state, which is in a superposition of two qubit states:

$$|\psi(t)\rangle = c_1(t)|0\rangle + c_2(t)|1\rangle$$

Then from the TDSE, we have:

$$i\hbar\frac{d}{dt}\begin{bmatrix} c1 \\ c2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}\begin{bmatrix} c1 \\ c2 \end{bmatrix}$$

which gives us the following two differential equations:

$$\frac{dc_1}{dt} = -\frac{i}{\hbar}c_1(t)$$
$$\frac{dc_2}{dt} = \frac{i}{\hbar}c_2(t)$$

The solutions to these differential equations are given by:

$$c_1 = c_1(0)e^{-it/\hbar}$$
$$c_2 = c_2(0)e^{it/\hbar}$$

To see an example, we assume initial conditions $c_1(0) = \frac{2}{\sqrt{5}}$ and $c_2(0) = \frac{1}{\sqrt{5}}$
Then $c_1(t) = \frac{2}{\sqrt{5}}e^{-it/\hbar}$ and $c_2(t) = \frac{1}{\sqrt{5}}e^{it/\hbar}$
and the state reads as:

$$|\psi(t)\rangle = \frac{2}{\sqrt{5}}e^{-it/\hbar}|0\rangle + \frac{1}{\sqrt{5}}e^{it/\hbar}|1\rangle$$

or in matrix form:

$$|\psi(t)\rangle = \begin{bmatrix} \frac{2}{\sqrt{5}} e^{-it/\hbar} \\ \frac{1}{\sqrt{5}} e^{it/\hbar} \end{bmatrix}$$

We also calculate the complex conjugate of the state:

$$\langle\psi(t)| = \begin{bmatrix} \frac{2}{\sqrt{5}} e^{it/\hbar} & \frac{1}{\sqrt{5}} e^{-it/\hbar} \end{bmatrix}$$

The expectation value of the operator $\hat{\sigma}_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$ (pauli-y) is given by:

$$\begin{aligned}
\langle\psi(t)|\hat{\sigma}_y|\psi(t)\rangle &= \begin{bmatrix} \frac{2}{\sqrt{5}} e^{it/\hbar} & \frac{1}{\sqrt{5}} e^{-it/\hbar} \end{bmatrix} \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \begin{bmatrix} \frac{2}{\sqrt{5}} e^{-it/\hbar} \\ \frac{1}{\sqrt{5}} e^{it/\hbar} \end{bmatrix} \\
&= \begin{bmatrix} \frac{2}{\sqrt{5}} e^{it/\hbar} & \frac{1}{\sqrt{5}} e^{-it/\hbar} \end{bmatrix} \begin{bmatrix} \frac{-i}{\sqrt{5}} e^{it/\hbar} \\ \frac{2i}{\sqrt{5}} e^{-it/\hbar} \end{bmatrix} \\
&= -\frac{2i}{5} e^{2it/\hbar} + \frac{2i}{5} e^{-2it/\hbar} \\
&= -\frac{2i}{5} (e^{2it/\hbar} - e^{-2it/\hbar}) \\
&= \frac{2}{5} (\frac{e^{2it/\hbar} - e^{-2it/\hbar}}{i}) \\
&= \frac{4}{5} \sin(\frac{2t}{\hbar})
\end{aligned}$$

Below we plot the expectation value of $\hat{\sigma}_y$ with respect to time using our analytic result:

```python
import numpy as np
import matplotlib.pyplot as plt

sigma_y = np.array([[0, -1j],
                    [1j,  0]])

hbar = 1.0
t = np.linspace(0, 10, 1000)

def sigma_y_expectation_value(t):
    return (4/5) * np.sin(2*t/hbar)

plt.figure(figsize=(10, 6))
plt.plot(t, sigma_y_expectation_value(t))
plt.title("Expectation Value of $\sigma_y$")
plt.xlabel("Time")
plt.ylabel(r"$\langle \sigma_y \rangle$")
plt.grid(True)
plt.tight_layout()
plt.show()
```

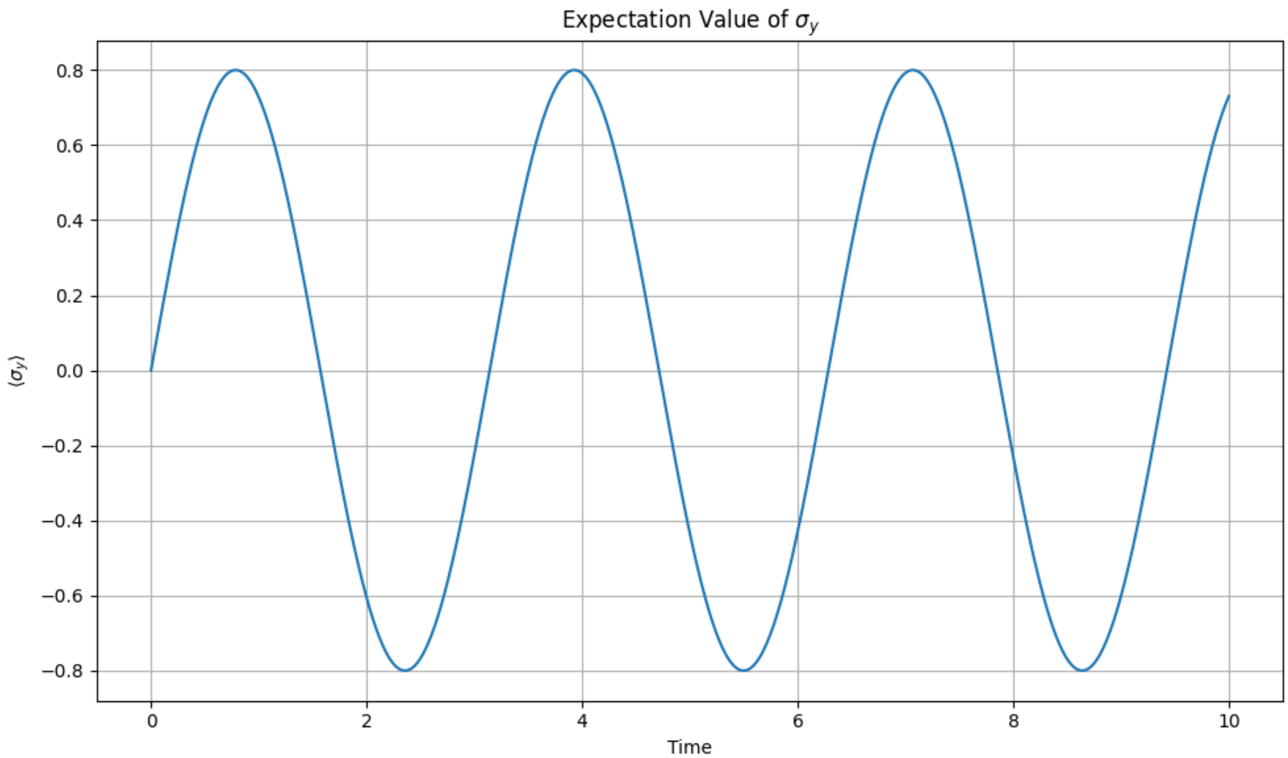

Figure 1: Expectation value of $\hat{\sigma}_y$ as a function of time, using the analytic results.

Now we will use QuTip to achieve the same result.
First we define our state and display it in the Bloch sphere, using QuTip's **basis()** and **Bloch()** functions:

```
import matplotlib.pyplot as plt
import numpy as np
import qutip
from qutip import Bloch, QobjEvo, basis, sesolve, sigmay, sigmaz

psi = (2.0 * basis(2, 0) + basis(2, 1)).unit()
b = Bloch()
b.add_states(psi)
b.show()
```

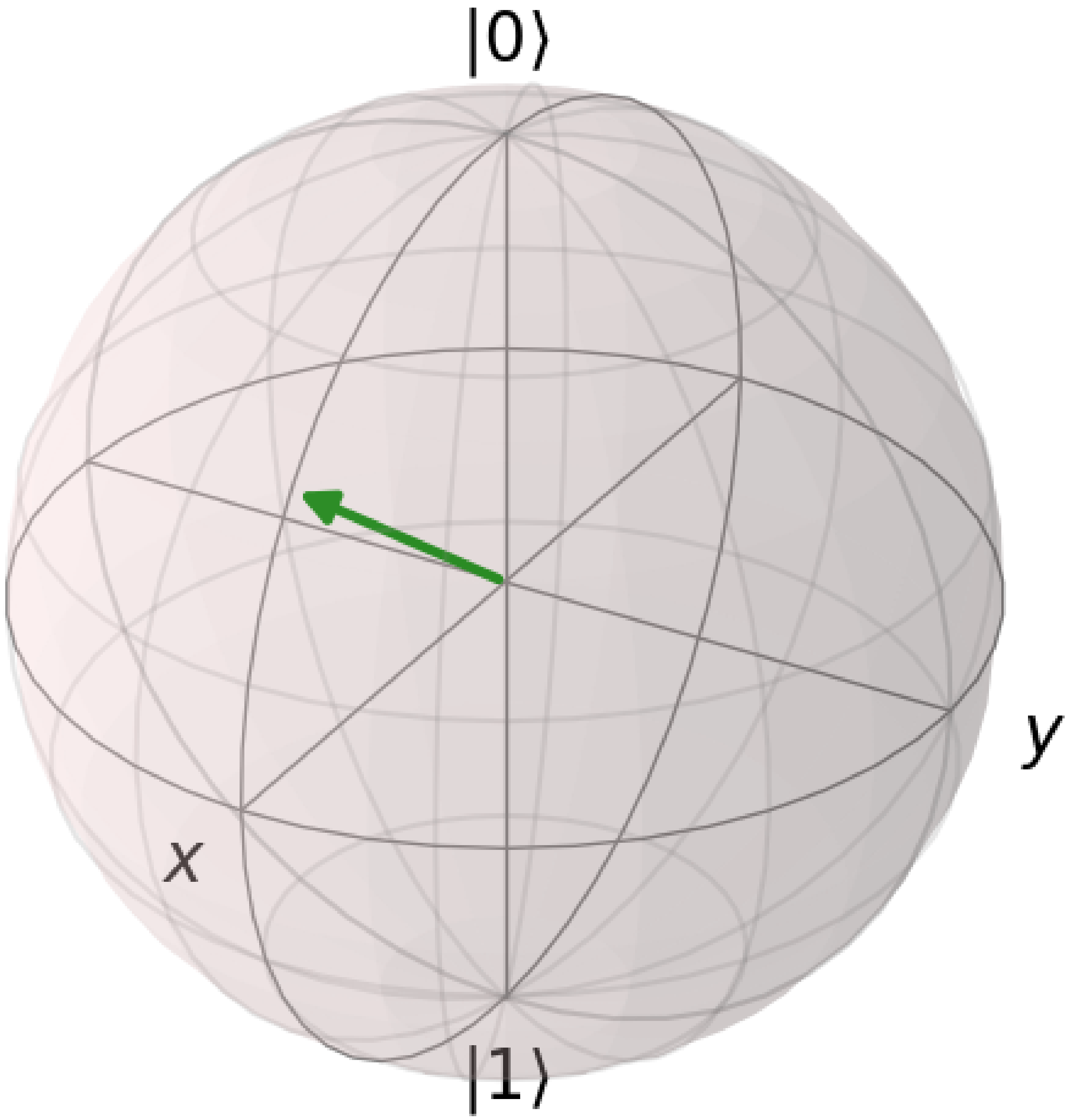

Figure 2: Expectation value of $\hat{\sigma}_y$ as a function of time, using the analytic results.

Then we define our Hamiltonian using QuTip's **sigmaz()** function and find the expectation value of $\hat{\sigma}_y$ using the **sesolve()** and passing **sigmay()** as a parameter.

```
H = sigmaz()
times = np.linspace(0, 10, 1000)
result = sesolve(H, psi, times, [sigmay()])

plt.figure(figsize=(10, 6))
plt.plot(times, result.expect[0])
plt.title("Expectation Value of $\sigma_y$")
plt.xlabel("Time")
plt.ylabel(r"$\langle \sigma_y \rangle$")
plt.grid(True)
plt.tight_layout()
```

```
plt.show()
```

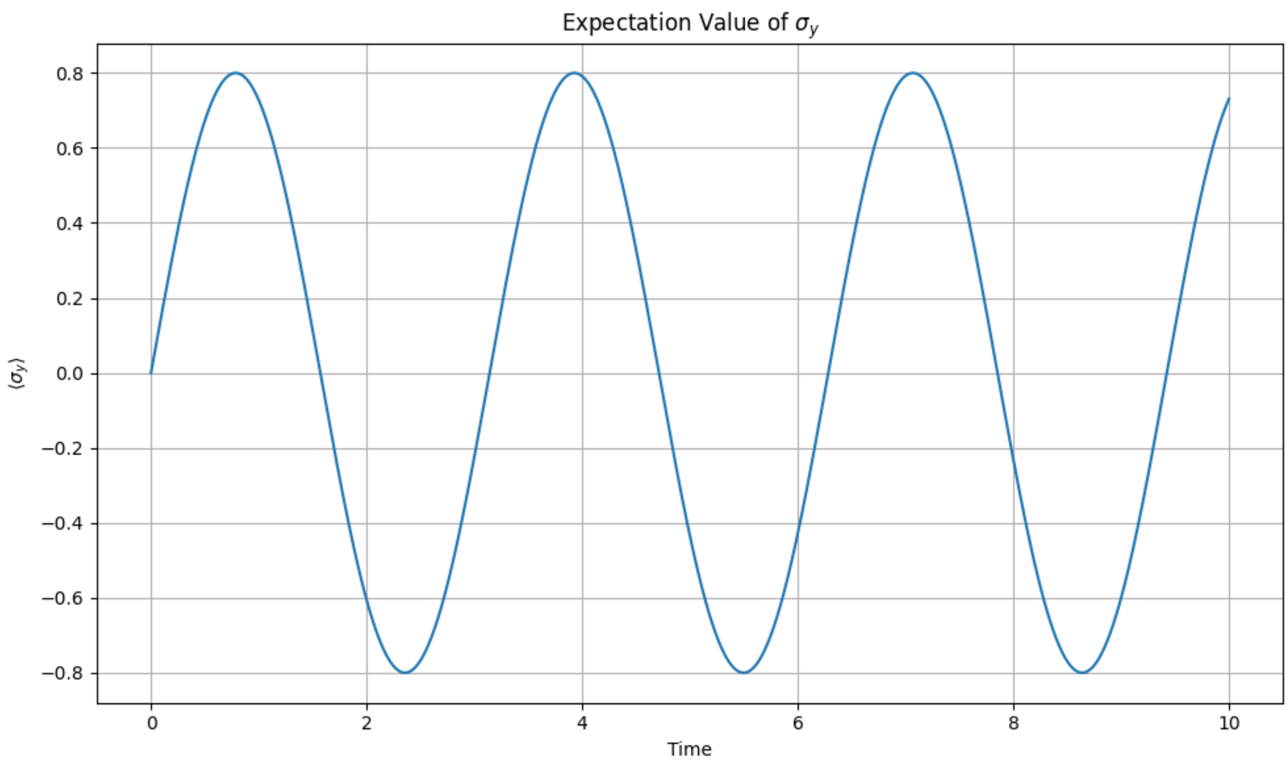

Figure 3: Expectation value of $\hat{\sigma}_y$ as a function of time, using QuTip's **sesolve()**.

We observe that we obtain identical results either using **sesolve()** or the analytic result.

2.2 Time dependent magnetic field with linear field strenght

Following a similar methodology, we can calculate the expectation value of $\hat{\sigma}_y$ for a time dependent Hamiltonian that resembles to a magnetic field with with linear field strenth.
To consider a specific example, we assume our Hamiltonian to be:

$$H(t) = 0.3t\sigma_z = \begin{bmatrix} 0.3t & 0 \\ 0 & -0.3t \end{bmatrix}$$

Then from the TDSE, we have:

$$i\hbar \frac{d}{dt} \begin{bmatrix} c1 \\ c2 \end{bmatrix} = \begin{bmatrix} 0.3t & 0 \\ 0 & -0.3t \end{bmatrix} \begin{bmatrix} c1 \\ c2 \end{bmatrix}$$

which gives us the following two differential equations:

$$\frac{dc_1}{dt} = -\frac{i}{\hbar} 0.3tc_1(t)$$
$$\frac{dc_2}{dt} = \frac{i}{\hbar} 0.3tc_2(t)$$

The solutions to these differential equations are given by:

$$c_1 = c_1(0)e^{-\frac{i}{\hbar}0.15t^2}$$
$$c_2 = c_2(0)e^{\frac{i}{\hbar}0.15t^2}$$

To see an example, we assume initial conditions $c_1(0) = \dfrac{2}{\sqrt{5}}$ and $c_2(0) = \dfrac{1}{\sqrt{5}}$

Then $c_1(t) = \dfrac{2}{\sqrt{5}}e^{-\frac{i}{\hbar}0.15t^2}$ and $c_2(t) = \dfrac{1}{\sqrt{5}}e^{\frac{i}{\hbar}0.15t^2}$
and the state reads as:

$$|\psi(t)\rangle = \frac{2}{\sqrt{5}}e^{-\frac{i}{\hbar}0.15t^2}|0\rangle + \frac{1}{\sqrt{5}}e^{\frac{i}{\hbar}0.15t^2}|1\rangle$$

or in matrix form:

$$|\psi(t)\rangle = \begin{bmatrix} \frac{2}{\sqrt{5}}e^{-\frac{i}{\hbar}0.15t^2} \\ \frac{1}{\sqrt{5}}e^{\frac{i}{\hbar}0.15t^2} \end{bmatrix}$$

We also calculate the complex conjugate of the state:

$$\langle\psi(t)| = \begin{bmatrix} \frac{2}{\sqrt{5}}e^{\frac{i}{\hbar}0.15t^2} & \frac{1}{\sqrt{5}}e^{-\frac{i}{\hbar}0.15t^2} \end{bmatrix}$$

The expectation value of the operator $\hat{\sigma}_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$ (pauli-y) is given by:

$$\begin{aligned}
\langle\psi(t)|\hat{\sigma}_y|\psi(t)\rangle &= \begin{bmatrix} \frac{2}{\sqrt{5}}e^{\frac{i}{\hbar}0.15t^2} & \frac{1}{\sqrt{5}}e^{-\frac{i}{\hbar}0.15t^2} \end{bmatrix} \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \begin{bmatrix} \frac{2}{\sqrt{5}}e^{-\frac{i}{\hbar}0.15t^2} \\ \frac{1}{\sqrt{5}}e^{\frac{i}{\hbar}0.15t^2} \end{bmatrix} \\
&= \begin{bmatrix} \frac{2}{\sqrt{5}}e^{\frac{it}{\hbar}} & \frac{1}{\sqrt{5}}e^{-\frac{it}{\hbar}} \end{bmatrix} \begin{bmatrix} \frac{-i}{\sqrt{5}}e^{\frac{i}{\hbar}0.15t^2} \\ \frac{2i}{\sqrt{5}}e^{-\frac{i}{\hbar}0.15t^2} \end{bmatrix} \\
&= -\frac{2i}{5}e^{\frac{i}{\hbar}0.3t^2} + \frac{2i}{5}e^{-\frac{i}{\hbar}0.3t^2} \\
&= -\frac{2i}{5}(e^{\frac{i}{\hbar}0.3t^2} - e^{-\frac{i}{\hbar}0.3t^2}) \\
&= \frac{2}{5}(\frac{e^{\frac{i}{\hbar}0.3t^2} - e^{-\frac{i}{\hbar}0.3t^2}}{i}) \\
&= \frac{4}{5}\sin(\frac{0.3t^2}{\hbar})
\end{aligned}$$

Below we plot the expectation value of $\hat{\sigma}_y$ with respect to time using our analytic result:

```
import numpy as np
import matplotlib.pyplot as plt

sigma_y = np.array([[0, -1j],
                    [1j,  0]])

hbar = 1.0
t = np.linspace(0, 10, 1000)

def sigma_y_expectation_value(t):
    return (4/5) * np.sin((0.3*t**2)/hbar)

plt.figure(figsize=(10, 6))
plt.plot(t, sigma_y_expectation_value(t))
plt.title("Expectation Value of $\sigma_y$")
plt.xlabel("Time")
plt.ylabel(r"$\langle \sigma_y \rangle$")
plt.grid(True)
plt.tight_layout()
plt.show()
```

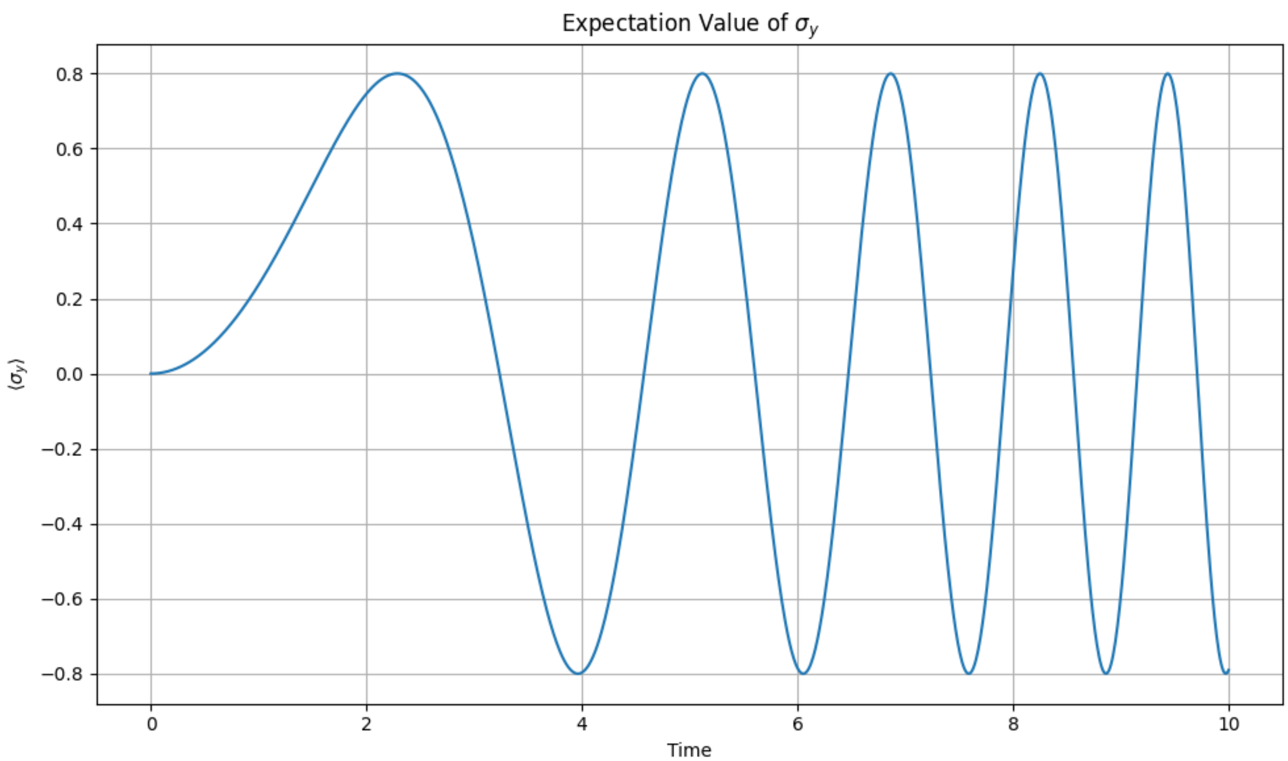

Figure 4: Expectation value of $\hat{\sigma}_y$ as a function of time, using the analytic results.

Now we will use QuTip to achieve the same result.
We define our state and then we define our Hamiltonian using QuTip's **QobjEvo()** class and find the expectation value of $\hat{\sigma}_y$ using the **sesolve()** and passing **sigmay()** as a parameter.

```
import matplotlib.pyplot as plt
import numpy as np
import qutip
from qutip import Bloch, QobjEvo, basis, sesolve, sigmay, sigmaz

times = np.linspace(0, 10, 1000)
psi = (2.0 * basis(2, 0) + basis(2, 1)).unit()
b = Bloch()
b.add_states(psi)
```

```python
def linear(t, args):
    return 0.3 * t

H_lin = QobjEvo([[sigmaz(), linear]], tlist=times)

result_lin = sesolve(H_lin, psi, times, [sigmay()])

plt.figure(figsize=(10, 6))
plt.plot(times, result_lin.expect[0])
plt.title("Expectation Value of $\sigma_y$")
plt.xlabel("Time")
plt.ylabel(r"$\langle \sigma_y \rangle$")
plt.grid(True)
plt.tight_layout()
plt.show()
```

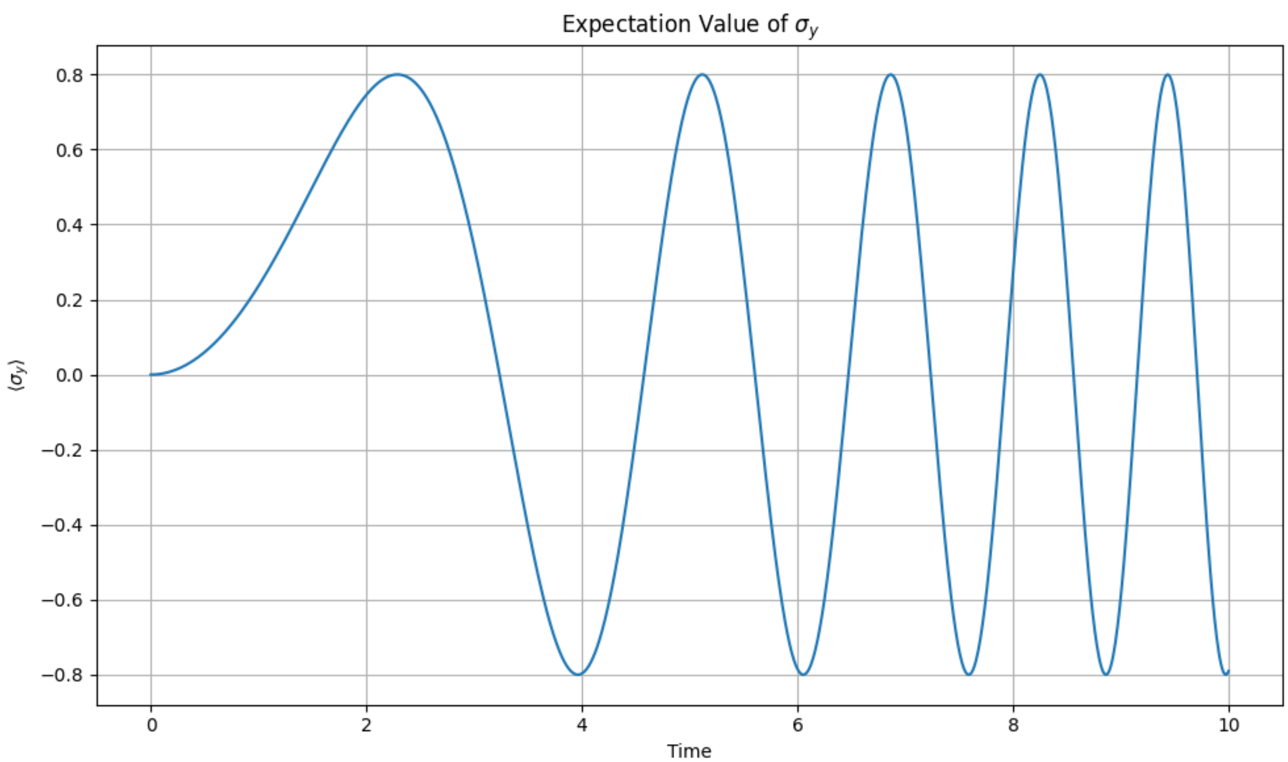

Figure 5: Expectation value of $\hat{\sigma}_y$ as a function of time, using QuTip's **sesolve()**.

We observe that we obtain identical results either using **sesolve()** or the analytic result.

2.3 Time dependent magnetic field with periodic field strenght

Following a similar methodology, we can calculate the expectation value of $\hat{\sigma}_y$ for a time dependent Hamiltonian that resembles to a magnetic field with with periodic field strenth.
To consider a specific example, we assume our Hamiltonian to be:

$$H(t) = \cos(\frac{t}{2})\sigma_z = \begin{bmatrix} \cos(\frac{t}{2}) & 0 \\ 0 & -\cos(\frac{t}{2}) \end{bmatrix}$$

Then from the TDSE, we have:

$$i\hbar\frac{d}{dt}\begin{bmatrix} c1 \\ c2 \end{bmatrix} = \begin{bmatrix} \cos(\frac{t}{2}) & 0 \\ 0 & -\cos(\frac{t}{2}) \end{bmatrix}\begin{bmatrix} c1 \\ c2 \end{bmatrix}$$

which gives us the following two differential equations:

$$\frac{dc_1}{dt} = -\frac{i}{\hbar}\cos(\frac{t}{2})c_1(t)$$
$$\frac{dc_2}{dt} = \frac{i}{\hbar}\cos(\frac{t}{2})c_2(t)$$

The solutions to these differential equations are given by:

$$c_1 = c_1(0)e^{-\frac{2i}{\hbar}\sin(\frac{t}{2})}$$
$$c_2 = c_2(0)e^{\frac{2i}{\hbar}\sin(\frac{t}{2})}$$

To see an example, we assume initial conditions $c_1(0) = \frac{2}{\sqrt{5}}$ and $c_2(0) = \frac{1}{\sqrt{5}}$

Then $c_1(t) = \frac{2}{\sqrt{5}}e^{-\frac{2i}{\hbar}\sin(\frac{t}{2})}$ and $c_2(t) = \frac{1}{\sqrt{5}}e^{\frac{2i}{\hbar}\sin(\frac{t}{2})}$
and the state reads as:

$$|\psi(t)\rangle = \frac{2}{\sqrt{5}}e^{-\frac{2i}{\hbar}\sin(\frac{t}{2})}|0\rangle + \frac{1}{\sqrt{5}}e^{\frac{2i}{\hbar}\sin(\frac{t}{2})}|1\rangle$$

or in matrix form:

$$|\psi(t)\rangle = \begin{bmatrix} \frac{2}{\sqrt{5}}e^{-\frac{2i}{\hbar}\sin(\frac{t}{2})} \\ \frac{1}{\sqrt{5}}e^{\frac{2i}{\hbar}\sin(\frac{t}{2})} \end{bmatrix}$$

We also calculate the complex conjugate of the state:

$$\langle\psi(t)| = \begin{bmatrix} \frac{2}{\sqrt{5}}e^{\frac{2i}{\hbar}\sin(\frac{t}{2})} & \frac{1}{\sqrt{5}}e^{-\frac{2i}{\hbar}\sin(\frac{t}{2})} \end{bmatrix}$$

The expectation value of the operator $\hat{\sigma}_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$ (pauli-y) is given by:

$$
\begin{aligned}
\langle\psi(t)|\hat{\sigma}_y|\psi(t)\rangle &= \begin{bmatrix} \frac{2}{\sqrt{5}} e^{\frac{2i}{\hbar}\sin(\frac{t}{2})} & \frac{1}{\sqrt{5}} e^{-\frac{2i}{\hbar}\sin(\frac{t}{2})} \end{bmatrix} \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \begin{bmatrix} \frac{2}{\sqrt{5}} e^{-\frac{2i}{\hbar}\sin(\frac{t}{2})} \\ \frac{1}{\sqrt{5}} e^{\frac{2i}{\hbar}\sin(\frac{t}{2})} \end{bmatrix} \\
&= \begin{bmatrix} \frac{2}{\sqrt{5}} e^{\frac{2i}{\hbar}\sin(\frac{t}{2})} & \frac{1}{\sqrt{5}} e^{-\frac{2i}{\hbar}\sin(\frac{t}{2})} \end{bmatrix} \begin{bmatrix} \frac{-i}{\sqrt{5}} e^{\frac{2i}{\hbar}\sin(\frac{t}{2})} \\ \frac{2i}{\sqrt{5}} e^{-\frac{2i}{\hbar}\sin(\frac{t}{2})} \end{bmatrix} \\
&= -\frac{2i}{5} e^{\frac{4i}{\hbar}\sin(\frac{t}{2})} + \frac{2i}{5} e^{\frac{-4i}{\hbar}\sin(\frac{t}{2})} \\
&= -\frac{2i}{5}\left(e^{\frac{4i}{\hbar}\sin(\frac{t}{2})} - e^{\frac{-4i}{\hbar}\sin(\frac{t}{2})}\right) \\
&= \frac{2}{5}\left(\frac{e^{\frac{4i}{\hbar}\sin(\frac{t}{2})} - e^{\frac{-4i}{\hbar}\sin(\frac{t}{2})}}{i}\right) \\
&= \frac{4}{5}\sin\left(\frac{4}{\hbar}\sin\left(\frac{t}{2}\right)\right)
\end{aligned}
$$

Below we plot the expectation value of $\hat{\sigma}_y$ with respect to time using our analytic result:

```python
import numpy as np
import matplotlib.pyplot as plt

sigma_y = np.array([[0, -1j],
                    [1j,  0]])

hbar = 1.0
t = np.linspace(0, 10, 1000)

def sigma_y_expectation_value(t):
    return (4/5) * np.sin(4/hbar * np.sin(t/2))

plt.figure(figsize=(10, 6))
plt.plot(t, sigma_y_expectation_value(t))
plt.title("Expectation Value of $\sigma_y$")
plt.xlabel("Time")
plt.ylabel(r"$\langle \sigma_y \rangle$")
plt.grid(True)
plt.tight_layout()
plt.show()
```

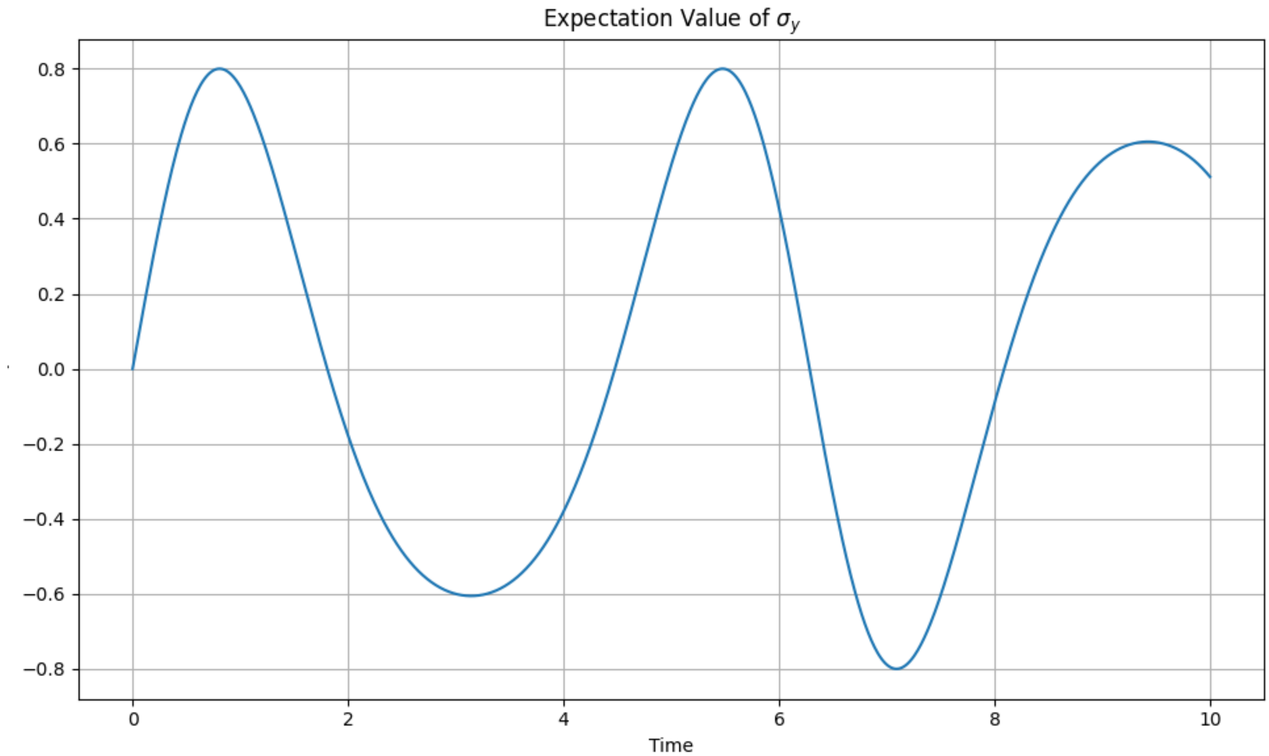

Figure 6: Expectation value of $\hat{\sigma}_y$ as a function of time, using the analytic results.

Now we will use QuTip to achieve the same result.
We define our state and then we define our Hamiltonian using QuTip's **QobjEvo()** class and find the expectation value of $\hat{\sigma}_y$ using the **sesolve()** and passing **sigmay()** as a parameter.

```
import matplotlib.pyplot as plt
import numpy as np
import qutip
from qutip import Bloch, QobjEvo, basis, sesolve, sigmay, sigmaz

times = np.linspace(0, 10, 1000)
psi = (2.0 * basis(2, 0) + basis(2, 1)).unit()

def periodic(t, args):
    return np.cos(0.5 * t)

H_per = QobjEvo([[sigmaz(), periodic]], tlist=times)

result_per = sesolve(H_per, psi, times, [sigmay()])

plt.figure(figsize=(10, 6))
plt.plot(times, result_per.expect[0])
plt.title("Expectation Value of $\sigma_y$")
plt.xlabel("Time")
plt.ylabel(r"$\langle \sigma_y \rangle$")
plt.grid(True)
plt.tight_layout()
plt.show()
```

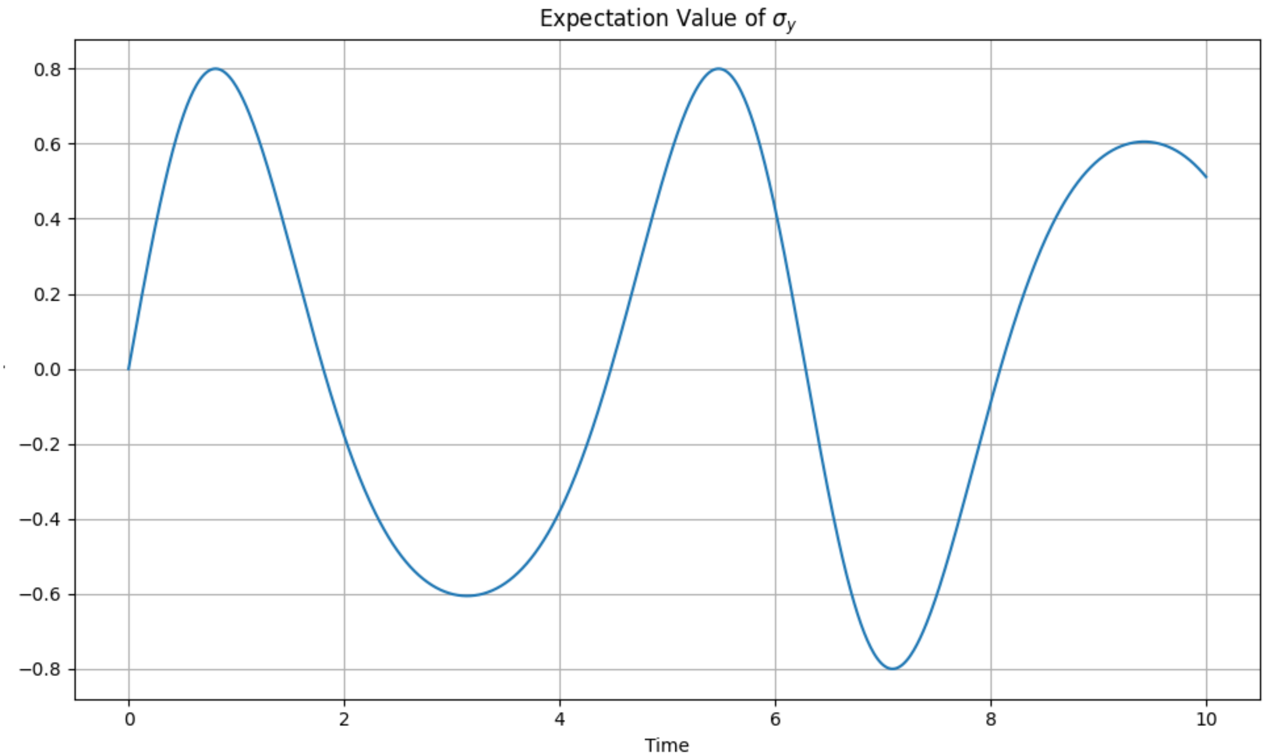

Figure 7: Expectation value of $\hat{\sigma}_y$ as a function of time, using QuTip's **sesolve()**.

We observe that we obtain identical results either using **sesolve()** or the analytic result.

3 Master Equation Solver: Single-Qubit Dynamics

Previously we described the state of a **closed** quantum system with a state vector. The evolution of the state vector in closed quantum systems is deterministic. When we have to deal with an **open** quantum system, the evolution becomes stochastic as the interaction with the environment can have effects of dissipation and decoherence. For this reason, we describe the state of an open quantum system using the **density matrix** formalism, which gives us a framework for representing both pure and mixed states.
Pure states are states defined by a single state vector and represent maximum knowledge about a quantum system and a coherent superposition of basis states.
Mixed states are states that cannot be defined by a single state vector. They represent statistical uncertainty over several pure states and capture a probabilistic mixture rather than a coherent superposition.

The density matrix for pure states is given by:

$$\hat{\rho}(t) = |\psi(t)\rangle\langle\psi(t)|$$

and the density matrix elements:

$$\rho_{mn}(t) = \langle\psi_m|\hat{\rho}(t)|\psi_n\rangle = c_m(t)c_n^*(t)$$

In matrix form we have:

$$\hat{\rho}(t) = \begin{bmatrix} \rho_{11}(t) & \rho_{12}(t) & \dots \\ \rho_{21}(t) & \rho_{22}(t) & \dots \\ \dots & \dots & \ddots \end{bmatrix}$$

The diagonal elements show the population of state $|\psi_n\rangle$, with $n = 1, 2, 3, \ldots$ while the off-diagonal elements are called coherence and represent the probability amplitudes from coherent superpositions of the two respective states.

The density matrix for mixed states is given by:

$$\hat{\rho} = \sum_j P_j |\psi_j\rangle\langle\psi_j| = \sum_j P_j \hat{\rho}_j$$

where $|\psi_j\rangle$ are pure states and P_j the probabilities which they are prepared with.
The density matrix has the following properties:

- $\hat{\rho}^2 = \hat{\rho}$ (holds only for pure states)
- $\hat{\rho}^\dagger = \hat{\rho}$ (holds for both pure and mixed states)
- $Tr[\hat{\rho}] = 1$ (holds for both pure and mixed states)
- $\hat{\rho} \geq 0$ (holds for both pure and mixed states)

Another useful relationship is that we can calculate the expectation value of an observable Q as:

$$\langle Q\rangle = Tr(\hat{\rho}\hat{Q})$$

The equivalent to the Schrödinger equation but in the density matrix formalism, for either pure or mixed states, is the **von Neumann** or **Liouville-von Neumann** equation:

$$\frac{d\hat{\rho}}{dt} = -\frac{i}{\hbar}[\hat{H}(t)\,, \hat{\rho}(t)]$$

This differential equations give us the equations of motion for a system interacting with its environment by expanding the scope of the system to include the environment. Here the Hamiltonian refers to the total Hamiltonian that includes the initial system's Hamiltonian, the Hamiltonian for the environment and a term for the interaction between the two:

$$H(t) = H_{sys}(t) + H_{env}(t) + H_{int}(t)$$

If we are interested only in the dynamics of the system as an open system interacting with its environment, we can trace out the degrees of freedom associated with the environment assuming that the system interacts weakly with it. The result is a reduced, open-system dynamics equation for the system alone, which is called **the Lindblad master equation**:

$$\frac{d\hat{\rho}}{dt} = -\frac{i}{\hbar}[\hat{H}(t)\,, \hat{\rho}(t)] + \sum_k \gamma_k \big(\hat{L}_k \hat{\rho}(t) \hat{L}_k^\dagger - \frac{1}{2}\{\hat{L}_k^\dagger \hat{L}_k, \hat{\rho}\}\big)$$

where:

- $\hat{\rho}(t)$ is the system's density matrix
- $\hat{H}(t)$ is the system's Hamiltonian
- $\hat{L}_k$ are the operators through which the environment couples to the system in H_{int}, named **Lindblad operators** or **collapse operators**. They represent some non-unitary process like decay or dephasing.
- γ_k are the corresponding rates at which the non-unitary process decribed by the Lindblad operators occur. It is important that these rates are smaller than the minimum energy splitting in the system Hamiltonian.
- $\{\hat{L}_k^\dagger \hat{L}_k, \hat{\rho}\} = \hat{L}_k^\dagger \hat{L}_k \hat{\rho}(t) + \hat{\rho}(t) \hat{L}_k^\dagger \hat{L}_k$ an anticommutator.

The QuTip python library includes the function **mesolve()** which is used for evolution both according to the Schrödinger equation and to the master equation. It automatically determines if it is sufficient to use the Schrödinger equation (if no collapse operators were given) or if it has to use the master equation. A list of collapse operators is passed as the fourth argument (c_ops) to the mesolve() function in order to define the dissipation processes in the master equation. When the c_ops isn't empty, mesolve() will use the master equation instead of the unitary Schrödinger equation.

3.1 Example: Simple use of mesolve() with a collapse operator

We consider a system described by a qubit state initially in the ground state:

$$|\psi(t)\rangle = c_1|0\rangle + c_2|1\rangle = \begin{bmatrix} c1 \\ c2 \end{bmatrix}$$

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

We have the following Hamiltonian:

$$H = \frac{\Delta}{2}\sigma_x$$

or in matrix form:

$$H = \begin{bmatrix} 0 & \frac{\Delta}{2} \\ \frac{\Delta}{2} & 0 \end{bmatrix}$$

The TDSE reads as:

$$\begin{aligned} i\hbar\frac{d}{dt}|\psi(t)\rangle &= H(t)|\psi(t)\rangle \\ \implies \frac{d}{dt}\begin{bmatrix} c1 \\ c2 \end{bmatrix} &= -\frac{i}{\hbar}\begin{bmatrix} 0 & \frac{\Delta}{2} \\ \frac{\Delta}{2} & 0 \end{bmatrix}\begin{bmatrix} c1 \\ c2 \end{bmatrix} \end{aligned}$$

which gives us the following two differential equations:

$$\frac{dc_1}{dt} = -\frac{i}{\hbar}\frac{\Delta}{2}c_2(t) \tag{1}$$

$$\frac{dc_2}{dt} = -\frac{i}{\hbar}\frac{\Delta}{2}c_1(t) \tag{2}$$

We differentiate (1) with respect to time and substitute from (2) and get:

$$(1) \implies \frac{d^2c_1}{dt^2} = -\frac{i}{\hbar}\frac{\Delta}{2}\frac{dc_2}{dt} \tag{3}$$

$$\overset{(2)}{\implies} \frac{d^2c_1}{dt^2} = -\frac{i}{\hbar}\frac{\Delta}{2}\big[-\frac{i}{\hbar}\frac{\Delta}{2}c_1(t)\big] \tag{4}$$

$$\implies \frac{d^2c_1}{dt^2} = -\frac{\Delta^2}{4\hbar^2}c_1(t) \tag{5}$$

The solution to this differential equation is known as:

$$c_1(t) = A\cos(\frac{\Delta}{2\hbar}t) + B\sin(\frac{\Delta}{2\hbar}t) \tag{6}$$

Then we have:

$$\frac{dc_1}{dt} = -\frac{\Delta}{2\hbar}A\sin(\frac{\Delta}{2\hbar}t) + \frac{\Delta}{2\hbar}B\cos(\frac{\Delta}{2\hbar}t) \tag{7}$$

We rearrange (1) and substitue from (7) and get:

$$
\begin{aligned}
(1) \implies c_2(t) &= \frac{2i\hbar}{\Delta}\frac{dc_1}{dt} && (8)\\
\overset{(7)}{\implies} c_2 &= -iA\sin(\frac{\Delta}{2\hbar}t) + iB\cos(\frac{\Delta}{2\hbar}t) && (9)
\end{aligned}
$$

From the initial conditions (ground state), we know that:

$$c_1(0) = 1 \implies A = 1 \tag{10}$$

and

$$c_2(0) = 0 \implies B = 0 \tag{11}$$

So eventually we get:

$$c_1(t) = \cos(\frac{\Delta}{2\hbar}t) \tag{12}$$

and

$$c_2(t) = -i\sin(\frac{\Delta}{2\hbar}t) \tag{13}$$

Then the state is written as:

$$|\psi(t)\rangle = \cos(\frac{\Delta}{2\hbar}t)|0\rangle - i\sin(\frac{\Delta}{2\hbar}t)|1\rangle \tag{14}$$

or in matrix form:

$$|\psi(t)\rangle = \begin{bmatrix} \cos(\frac{\Delta}{2\hbar}t) \\ -i\sin(\frac{\Delta}{2\hbar}t) \end{bmatrix} \tag{15}$$

We also calculate its comple conjugate:

$$\langle\psi(t)| = \begin{bmatrix} \cos(\frac{\Delta}{2\hbar}t) & i\sin(\frac{\Delta}{2\hbar}t) \end{bmatrix} \tag{16}$$

The previous calculations were for the closed quantum system.
If we consider the system to interact with it's external environment though, we need to switch to the density matrix formalism and utilize the Lindblad master equation.
We calculate the commutator:

$$
\begin{aligned}
[H(t), \rho(t)] &= H(t)\rho(t) - \rho(t)H(t) && (17)\\
&= \begin{bmatrix} 0 & \frac{\Delta}{2} \\ \frac{\Delta}{2} & 0 \end{bmatrix} \begin{bmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{bmatrix} - \begin{bmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{bmatrix} \begin{bmatrix} 0 & \frac{\Delta}{2} \\ \frac{\Delta}{2} & 0 \end{bmatrix} && (18)\\
&= \begin{bmatrix} \frac{\Delta}{2}\rho_{21} & \frac{\Delta}{2}\rho_{22} \\ \frac{\Delta}{2}\rho_{11} & \frac{\Delta}{2}\rho_{12} \end{bmatrix} - \begin{bmatrix} \frac{\Delta}{2}\rho_{12} & \frac{\Delta}{2}\rho_{11} \\ \frac{\Delta}{2}\rho_{22} & \frac{\Delta}{2}\rho_{21} \end{bmatrix} && (19)\\
&= \begin{bmatrix} \frac{\Delta}{2}\rho_{21} - \frac{\Delta}{2}\rho_{12} & \frac{\Delta}{2}\rho_{22} - \frac{\Delta}{2}\rho_{11} \\ \frac{\Delta}{2}\rho_{11} - \frac{\Delta}{2}\rho_{22} & \frac{\Delta}{2}\rho_{12} - \frac{\Delta}{2}\rho_{21} \end{bmatrix} && (20)
\end{aligned}
$$

Next we define a collapse operator:

$$\hat{C} = \sqrt{\gamma}\sigma_z = \begin{bmatrix} \sqrt{\gamma} & 0 \\ 0 & -\sqrt{\gamma} \end{bmatrix} \tag{21}$$

and

$$\hat{C}^\dagger = \begin{bmatrix} \sqrt{\gamma} & 0 \\ 0 & -\sqrt{\gamma} \end{bmatrix} \tag{22}$$

We have:

$$\begin{aligned}
\hat{C}\rho(t)\hat{C}^\dagger &= \begin{bmatrix} \sqrt{\gamma} & 0 \\ 0 & -\sqrt{\gamma} \end{bmatrix} \begin{bmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{bmatrix} \begin{bmatrix} \sqrt{\gamma} & 0 \\ 0 & -\sqrt{\gamma} \end{bmatrix} && (23) \\
&= \gamma \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} && (24) \\
&= \gamma \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \rho_{11} & -\rho_{12} \\ \rho_{21} & -\rho_{22} \end{bmatrix} && (25) \\
&= \begin{bmatrix} \gamma\rho_{11} & -\gamma\rho_{12} \\ -\gamma\rho_{21} & \gamma\rho_{22} \end{bmatrix} && (26)
\end{aligned}$$

It is:

$$\hat{C}^\dagger\hat{C} = \begin{bmatrix} \sqrt{\gamma} & 0 \\ 0 & -\sqrt{\gamma} \end{bmatrix} \begin{bmatrix} \sqrt{\gamma} & 0 \\ 0 & -\sqrt{\gamma} \end{bmatrix} = \gamma\mathbb{I} \tag{27}$$

and the anticommutator:

$$\{\hat{C}^\dagger\hat{C}\,, \rho(t)\} = \gamma\mathbb{I} \begin{bmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{bmatrix} + \begin{bmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{bmatrix} \gamma\mathbb{I} = 2\gamma \begin{bmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{bmatrix} = \begin{bmatrix} 2\gamma\rho_{11} & 2\gamma\rho_{12} \\ 2\gamma\rho_{21} & 2\gamma\rho_{22} \end{bmatrix} \tag{28}$$

Then the Lindblad master equation reads as:

$$\begin{aligned}
&\frac{d\rho}{dt} = -\frac{i}{\hbar}[H(t), \rho(t)] + (\hat{C}\rho(t)\hat{C}^\dagger - \frac{1}{2}\{\hat{C}^\dagger\hat{C}\,, \rho(t)\}) && (29) \\
&\implies \frac{d\rho}{dt} = -\frac{i}{\hbar} \begin{bmatrix} \frac{\Delta}{2}\rho_{21} - \frac{\Delta}{2}\rho_{12} & \frac{\Delta}{2}\rho_{22} - \frac{\Delta}{2}\rho_{11} \\ \frac{\Delta}{2}\rho_{11} - \frac{\Delta}{2}\rho_{22} & \frac{\Delta}{2}\rho_{12} - \frac{\Delta}{2}\rho_{21} \end{bmatrix} + \Big(\begin{bmatrix} \gamma\rho_{11} & -\gamma\rho_{12} \\ -\gamma\rho_{21} & \gamma\rho_{22} \end{bmatrix} - \frac{1}{2} \begin{bmatrix} 2\gamma\rho_{11} & 2\gamma\rho_{12} \\ 2\gamma\rho_{21} & 2\gamma\rho_{22} \end{bmatrix} \Big) && (30) \\
&\implies \frac{d\rho}{dt} = -\frac{i}{\hbar} \begin{bmatrix} \frac{\Delta}{2}\rho_{21} - \frac{\Delta}{2}\rho_{12} & \frac{\Delta}{2}\rho_{22} - \frac{\Delta}{2}\rho_{11} \\ \frac{\Delta}{2}\rho_{11} - \frac{\Delta}{2}\rho_{22} & \frac{\Delta}{2}\rho_{12} - \frac{\Delta}{2}\rho_{21} \end{bmatrix} + \Big(\begin{bmatrix} \gamma\rho_{11} & -\gamma\rho_{12} \\ -\gamma\rho_{21} & \gamma\rho_{22} \end{bmatrix} - \begin{bmatrix} \gamma\rho_{11} & \gamma\rho_{12} \\ \gamma\rho_{21} & \gamma\rho_{22} \end{bmatrix} \Big) && (31) \\
&\implies \frac{d\rho}{dt} = \begin{bmatrix} -i\frac{\Delta}{2\hbar}\rho_{21} + i\frac{\Delta}{2\hbar}\rho_{12} & -i\frac{\Delta}{2\hbar}\rho_{22} + i\frac{\Delta}{2\hbar}\rho_{11} \\ -i\frac{\Delta}{2\hbar}\rho_{11} + i\frac{\Delta}{2\hbar}\rho_{22} & -i\frac{\Delta}{2\hbar}\rho_{12} + i\frac{\Delta}{2\hbar}\rho_{21} \end{bmatrix} + \begin{bmatrix} 0 & -2\gamma\rho_{12} \\ -2\gamma\rho_{21} & 0 \end{bmatrix} && (32) \\
&\implies \frac{d}{dt} \begin{bmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{bmatrix} = \begin{bmatrix} -i\frac{\Delta}{2\hbar}\rho_{21} + i\frac{\Delta}{2\hbar}\rho_{12} & -i\frac{\Delta}{2\hbar}\rho_{22} + i\frac{\Delta}{2\hbar}\rho_{11} - 2\gamma\rho_{12} \\ -i\frac{\Delta}{2\hbar}\rho_{11} + i\frac{\Delta}{2\hbar}\rho_{22} - 2\gamma\rho_{21} & -i\frac{\Delta}{2\hbar}\rho_{12} + i\frac{\Delta}{2\hbar}\rho_{21} \end{bmatrix} && (33)
\end{aligned}$$

which gives us the following differential equations

$$\frac{d\rho_{11}}{dt} = -i\frac{\Delta}{2\hbar}[\rho_{21}(t) - \rho_{12}(t)] \tag{34}$$

$$\frac{d\rho_{22}}{dt} = i\frac{\Delta}{2\hbar}[\rho_{21}(t) - \rho_{12}(t)] \tag{35}$$

$$\frac{d\rho_{12}}{dt} = -i\frac{\Delta}{2\hbar}[\rho_{22}(t) - \rho_{11}(t)] - 2\gamma\rho_{12} \tag{36}$$

$$\frac{d\rho_{21}}{dt} = i\frac{\Delta}{2\hbar}[\rho_{22}(t) - \rho_{11}(t)] - 2\gamma\rho_{21} \tag{37}$$

We now want to calculate the expectation value of the operator $\hat{\sigma}_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ (pauli-z).
Remember that in the density matrix formalism, it is:

$$\langle\hat{\sigma}_z\rangle = Tr(\hat{\rho}\sigma_z) = Tr(\begin{bmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}) = Tr(\begin{bmatrix} \rho_{11} & 0 \\ 0 & -\rho_{22} \end{bmatrix}) = \rho_{11} - \rho_{22} \tag{38}$$

In order to be able to calculate this, we need to obtain the solutions to the above differential equation system. The task to do this analytically is a tedious one, so we will solve them first numerically and then calculate the expectation value of $\hat{\sigma}_z$ using the formula above.
After that, we will do the same using QuTip's **mesolve()**.
Finally, we will plot both results in order to compare and verify them.

```python
import numpy as np
from scipy.integrate import solve_ivp
import matplotlib.pyplot as plt
from qutip import Bloch, about, basis, mesolve, sigmam, sigmax, sigmay, sigmaz

Delta = 2 * np.pi
hbar = 1.0
gamma = 0.25

t_span = (0, 5)
t_eval = np.linspace(*t_span, 100)

def drho_dt(t, y):
    rho11, rho22, rho12_real, rho12_imag = y
    rho12 = rho12_real + 1j * rho12_imag
    rho21 = np.conj(rho12)

    drho11_dt = -1j * (Delta / (2 * hbar)) * (rho21 - rho12)
    drho22_dt = 1j * (Delta / (2 * hbar)) * (rho21 - rho12)
    drho12_dt = -1j * (Delta / (2 * hbar)) * (rho22 - rho11) - 2 * gamma * rho12

    return [
        drho11_dt.real,
        drho22_dt.real,
        drho12_dt.real,
        drho12_dt.imag
    ]

#----Analytic solution----
rho11_0 = 1.0
rho22_0 = 0.0
rho12_0 = 0.0 + 0.0j

y0 = [rho11_0, rho22_0, rho12_0.real, rho12_0.imag]

analytic_density_matrix_solution = solve_ivp(drho_dt, t_span, y0, t_eval=t_eval)

rho11 = analytic_density_matrix_solution.y[0]
rho22 = analytic_density_matrix_solution.y[1]

sigma_z_analytic = rho11 - rho22

#----QuTip solution----
```

```
    H = Delta / 2.0 * sigmax()
    c_ops = [np.sqrt(gamma) * sigmaz()]
    psi0 = basis(2, 0)
    qutip_density_matrix_solution = mesolve(H, psi0, t_eval, c_ops, [sigmaz()])
    sigma_z_qutip = qutip_density_matrix_solution.expect[0]

    #----Plotting----
    plt.scatter(t_eval, sigma_z_qutip, c="r", marker="x", label="mesolve")
    plt.plot(analytic_density_matrix_solution.t, sigma_z_analytic, label="Analytic")
    plt.xlabel('Time')
    plt.ylabel(r'$\langle \sigma_z \rangle$')
    plt.title(r'Expectation value of $\sigma_z$')
    plt.grid(True)
    plt.legend()
    plt.show()
```

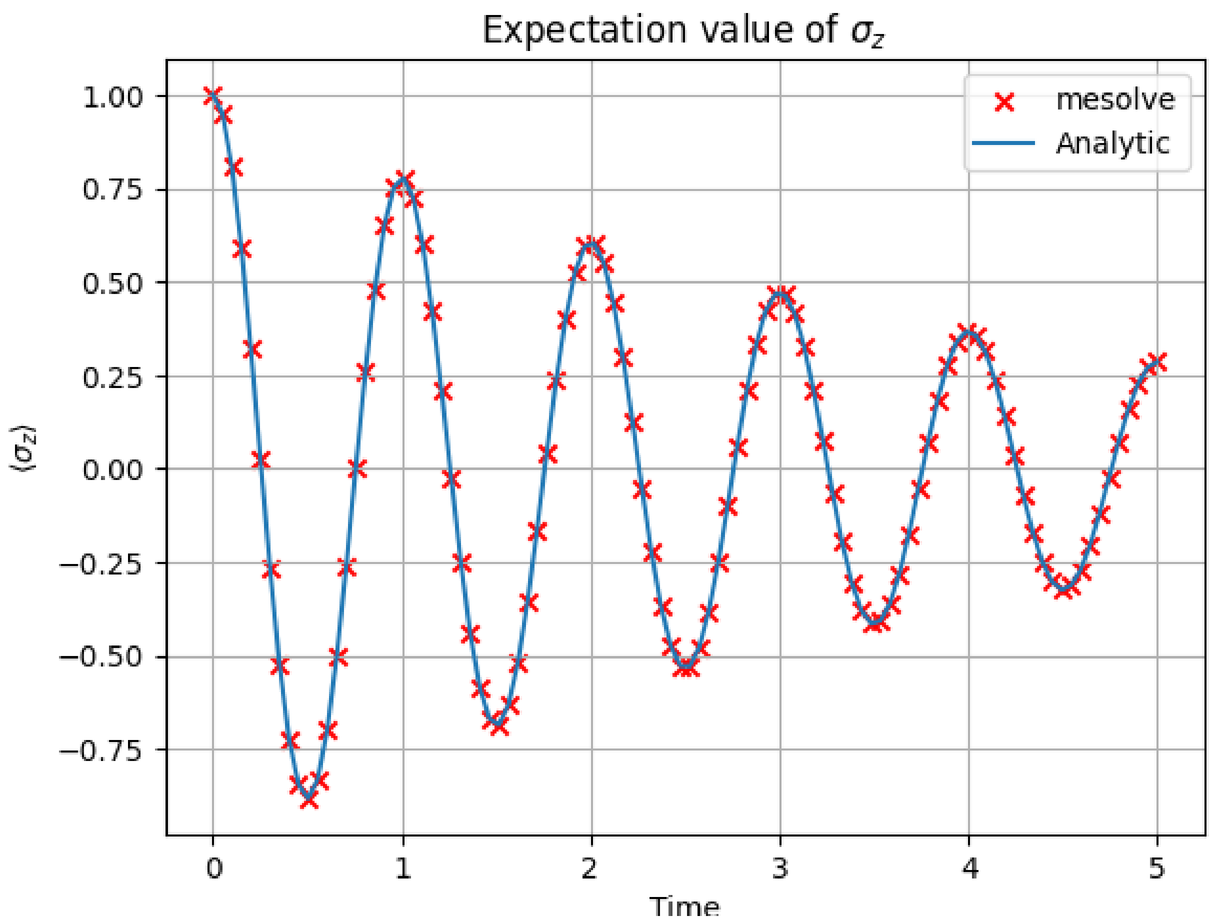

Figure 8: Plot of the expectation value of σ_z, calculated both using the analytic solution (blue line) and QuTip's **mesolve()** (red scattered plot)

We observe that both results are identical.

3.2 Example: Use of mesolve() without collapse operators

We consider a system described by a qubit state initially in the ground state:

$$|\psi(t)\rangle = c_1|0\rangle + c_2|1\rangle = \begin{bmatrix} c1 \\ c2 \end{bmatrix}$$

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

We have the following Hamiltonian:

$$H = \Delta(\cos(\theta)\sigma_z + \sin(\theta)\sigma_x)$$

or in matrix form:

$$H = \Delta(\begin{bmatrix} \cos(\theta) & 0 \\ 0 & -\cos(\theta) \end{bmatrix} + \begin{bmatrix} 0 & \sin(\theta) \\ \sin(\theta) & 0 \end{bmatrix}) = \begin{bmatrix} \Delta\cos(\theta) & \Delta\sin(\theta) \\ \Delta\sin(\theta) & -\Delta\cos(\theta) \end{bmatrix}$$

Here θ defines the angle of the qubit state between the z-axis toward the x-axis.
We will first calculate the expectation values of the pauli matrices σ_x, σ_y, σ_z analytically.
The TDSE reads as:

$$\begin{aligned} & i\hbar\frac{d}{dt}|\psi(t)\rangle = H(t)|\psi(t)\rangle \\ \implies & \frac{d}{dt}\begin{bmatrix} c1 \\ c2 \end{bmatrix} = -\frac{i}{\hbar}\begin{bmatrix} \Delta\cos(\theta) & \Delta\sin(\theta) \\ \Delta\sin(\theta) & -\Delta\cos(\theta) \end{bmatrix}\begin{bmatrix} c1 \\ c2 \end{bmatrix} \end{aligned}$$

which gives us the following two differential equations:

$$\frac{dc_1}{dt} = -\frac{i}{\hbar}\Delta\cos(\theta)c_1(t) - \frac{i}{\hbar}\Delta\sin(\theta)c_2(t) \tag{3.1.1}$$

$$\frac{dc_2}{dt} = -\frac{i}{\hbar}\Delta\sin(\theta)c_1(t) + \frac{i}{\hbar}\Delta\cos(\theta)c_2(t) \tag{3.1.2}$$

From (3.1.1) we get:

$$(3.1.1) \implies c_2(t) = -\frac{1}{\frac{i}{\hbar}\Delta\sin(\theta)}(\frac{dc_1}{dt} + \frac{i}{\hbar}\Delta\cos(\theta)c_1(t)) \tag{3.1.3}$$

We differentiate (3.1.1) with respect to time and substitute from (3.1.2) and (3.1.3) and get:

$$\begin{aligned} & \frac{d^2c_1}{dt^2} = -\frac{i}{\hbar}\Delta\cos(\theta)\frac{dc_1}{dt} - \frac{i}{\hbar}\Delta\sin(\theta)\frac{dc_2}{dt} \\ \overset{(3.1.2)}{\implies} & \frac{d^2c_1}{dt^2} = -\frac{i}{\hbar}\Delta\cos(\theta)\frac{dc_1}{dt} - \frac{i}{\hbar}\Delta\sin(\theta)[-\frac{i}{\hbar}\Delta\sin(\theta)c_1(t) + \frac{i}{\hbar}\Delta\cos(\theta)c_2(t)] \\ \implies & \frac{d^2c_1}{dt^2} = -\frac{i}{\hbar}\Delta\cos(\theta)\frac{dc_1}{dt} - \frac{\Delta^2}{\hbar^2}\sin^2(\theta)c_1(t) + \frac{\Delta^2}{\hbar^2}\sin(\theta)\cos(\theta)c_2(t) \\ \overset{(3.1.3)}{\implies} & \frac{d^2c_1}{dt^2} = -\frac{i}{\hbar}\Delta\cos(\theta)\frac{dc_1}{dt} - \frac{\Delta^2}{\hbar^2}\sin^2(\theta)c_1(t) + \frac{\Delta^2}{\hbar^2}\sin(\theta)\cos(\theta)\big[-\frac{1}{\frac{i}{\hbar}\Delta\sin(\theta)}(\frac{dc_1}{dt} + \frac{i}{\hbar}\Delta\cos(\theta)c_1(t))\big] \\ \implies & \frac{d^2c_1}{dt^2} = -\frac{i}{\hbar}\Delta\cos(\theta)\frac{dc_1}{dt} - \frac{\Delta^2}{\hbar^2}\sin^2(\theta)c_1(t) + \frac{i}{\hbar}\Delta\cos(\theta)\frac{dc_1}{dt} - \frac{\Delta^2}{\hbar^2}\cos^2(\theta)c_1(t) \\ \implies & \frac{d^2c_1}{dt^2} = -\frac{\Delta^2}{\hbar^2}(\sin^2(\theta) + \cos^2(\theta))c_1(t) \\ \implies & \frac{d^2c_1}{dt^2} = -\frac{\Delta^2}{\hbar^2}c_1(t) \\ \implies & \frac{d^2c_1}{dt^2} + \frac{\Delta^2}{\hbar^2}c_1(t) = 0 \end{aligned}$$

Now this differential equation resembles to a harmonic oscillator and has a known solution:

$$c_1(t) = A\cos(\frac{\Delta t}{\hbar}) + B\sin(\frac{\Delta t}{\hbar}) \tag{3.1.4}$$

Then:

$$\frac{dc1}{dt} = -A\frac{\Delta}{\hbar}\sin(\frac{\Delta t}{\hbar}) + B\frac{\Delta}{\hbar}\cos(\frac{\Delta t}{\hbar}) \tag{3.1.5}$$

Then we substitute (3.1.4) and (3.1.5) into (3.1.3) and get:

$$\begin{aligned}
(3.1.3) \overset{(3.1.4)}{\implies} c_2(t) &= -\frac{1}{\frac{i}{\hbar}\Delta\sin(\theta)}\big[-A\frac{\Delta}{\hbar}\sin(\frac{\Delta t}{\hbar}) + B\frac{\Delta}{\hbar}\cos(\frac{\Delta t}{\hbar}) + \frac{i}{\hbar}\Delta A\cos(\theta)\cos(\frac{\Delta t}{\hbar}) + \frac{i}{\hbar}\Delta\cos(\theta)B\sin(\frac{\Delta t}{\hbar})\big] \\
\implies c_2(t) &= -\frac{1}{\frac{i}{\hbar}\Delta\sin(\theta)}\cdot\frac{\Delta}{\hbar}\big[-A\sin(\frac{\Delta t}{\hbar}) + B\cos(\frac{\Delta t}{\hbar}) + i\cos(\theta)A\cos(\frac{\Delta t}{\hbar}) + iB\cos(\theta)\sin(\frac{\Delta t}{\hbar})\big] \\
\implies c_2(t) &= \frac{i}{\sin(\theta)}\big[-A\sin(\frac{\Delta t}{\hbar}) + B\cos(\frac{\Delta t}{\hbar}) + i\cos(\theta)A\cos(\frac{\Delta t}{\hbar}) + iB\cos(\theta)\sin(\frac{\Delta t}{\hbar})\big]
\end{aligned}$$

We know that:

$$c_1(0) = 1 \implies A = 1$$

and

$$\begin{aligned}
&c_2(0) = 0 \\
&\implies \frac{i}{\sin(\theta)}\big[B + i\cos(\theta)\big] = 0 \\
&\implies B = -i\cos(\theta)
\end{aligned}$$

Eventually we get:

$$\begin{aligned}
c_2(t) &= \frac{i}{\sin(\theta)}\big[-\sin(\frac{\Delta t}{\hbar}) - i\cos(\theta)\cos(\frac{\Delta t}{\hbar}) + i\cos(\theta)\cos(\frac{\Delta t}{\hbar}) + \cos^2(\theta)\sin(\frac{\Delta t}{\hbar})\big] \\
&= \frac{i}{\sin(\theta)}[-\sin(\frac{\Delta t}{\hbar}) + \cos^2(\theta)\sin(\frac{\Delta t}{\hbar})] \\
&= \frac{i}{\sin(\theta)}[(\cos^2(\theta) - 1)\sin(\frac{\Delta t}{\hbar})] \\
&= -\frac{i}{\sin(\theta)}\sin^2(\theta)\sin(\frac{\Delta t}{\hbar}) \\
&= -i\sin(\theta)\sin(\frac{\Delta t}{\hbar})
\end{aligned}$$

and

$$c_1(t) = \cos(\frac{\Delta t}{\hbar}) - i\cos(\theta)\sin(\frac{\Delta t}{\hbar})$$

The state reads as:

$$|\psi(t)\rangle = [\cos(\frac{\Delta t}{\hbar}) - i\cos(\theta)\sin(\frac{\Delta t}{\hbar})]|0\rangle - i\sin(\theta)\sin(\frac{\Delta t}{\hbar})|1\rangle$$

or in matrix form:

$$\begin{bmatrix} \cos(\frac{\Delta t}{\hbar}) - i\cos(\theta)\sin(\frac{\Delta t}{\hbar}) \\ -i\sin(\theta)\sin(\frac{\Delta t}{\hbar}) \end{bmatrix}$$

We also calculate the complex conjugate of the state:

$$\begin{bmatrix} \cos(\frac{\Delta t}{\hbar}) + i\cos(\theta)\sin(\frac{\Delta t}{\hbar}) & i\sin(\theta)\sin(\frac{\Delta t}{\hbar}) \end{bmatrix}$$

The expectation value of the operator $\hat{\sigma}_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ (pauli-x) is given by:

$$\begin{aligned}
\langle\psi(t)|\hat{\sigma}_x|\psi(t)\rangle &= \begin{bmatrix} \cos(\frac{\Delta t}{\hbar}) + i\cos(\theta)\sin(\frac{\Delta t}{\hbar}) & i\sin(\theta)\sin(\frac{\Delta t}{\hbar}) \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \cos(\frac{\Delta t}{\hbar}) - i\cos(\theta)\sin(\frac{\Delta t}{\hbar}) \\ -i\sin(\theta)\sin(\frac{\Delta t}{\hbar}) \end{bmatrix} \\
&= \begin{bmatrix} \cos(\frac{\Delta t}{\hbar}) + i\cos(\theta)\sin(\frac{\Delta t}{\hbar}) & i\sin(\theta)\sin(\frac{\Delta t}{\hbar}) \end{bmatrix} \begin{bmatrix} -i\sin(\theta)\sin(\frac{\Delta t}{\hbar}) \\ \cos(\frac{\Delta t}{\hbar}) - i\cos(\theta)\sin(\frac{\Delta t}{\hbar}) \end{bmatrix} \\
&= -i\sin(\theta)\sin(\frac{\Delta t}{\hbar})\cos(\frac{\Delta t}{\hbar}) + \sin(\theta)\cos(\theta)\sin^2(\frac{\Delta t}{\hbar}) + i\sin(\theta)\sin(\frac{\Delta t}{\hbar})\cos(\frac{\Delta t}{\hbar}) + \sin(\theta)\cos(\theta)\sin^2(\frac{\Delta t}{\hbar}) \\
&= 2\sin(\theta)\cos(\theta)\sin^2(\frac{\Delta t}{\hbar})
\end{aligned}$$

The expectation value of the operator $\hat{\sigma}_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$ (pauli-y) is given by:

$$\begin{aligned}
\langle\psi(t)|\hat{\sigma}_y|\psi(t)\rangle &= \begin{bmatrix} \cos(\frac{\Delta t}{\hbar}) + i\cos(\theta)\sin(\frac{\Delta t}{\hbar}) & i\sin(\theta)\sin(\frac{\Delta t}{\hbar}) \end{bmatrix} \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \begin{bmatrix} \cos(\frac{\Delta t}{\hbar}) - i\cos(\theta)\sin(\frac{\Delta t}{\hbar}) \\ -i\sin(\theta)\sin(\frac{\Delta t}{\hbar}) \end{bmatrix} \\
&= \begin{bmatrix} \cos(\frac{\Delta t}{\hbar}) + i\cos(\theta)\sin(\frac{\Delta t}{\hbar}) & i\sin(\theta)\sin(\frac{\Delta t}{\hbar}) \end{bmatrix} \begin{bmatrix} -\sin(\theta)\sin(\frac{\Delta t}{\hbar}) \\ i\cos(\frac{\Delta t}{\hbar}) + \cos(\theta)\sin(\frac{\Delta t}{\hbar}) \end{bmatrix} \\
&= -\sin(\theta)\sin(\frac{\Delta t}{\hbar})\cos(\frac{\Delta t}{\hbar}) - i\sin(\theta)\cos(\theta)\sin^2(\frac{\Delta t}{\hbar}) - \sin(\theta)\sin(\frac{\Delta t}{\hbar})\cos(\frac{\Delta t}{\hbar}) + i\sin(\theta)\cos(\theta)\sin^2(\frac{\Delta t}{\hbar}) \\
&= -2\sin(\theta)\sin(\frac{\Delta t}{\hbar})\cos(\frac{\Delta t}{\hbar})
\end{aligned}$$

The expectation value of the operator $\hat{\sigma}_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ (pauli-z) is given by:

$$\begin{aligned}
\langle\psi(t)|\hat{\sigma}_z|\psi(t)\rangle &= \begin{bmatrix} \cos(\frac{\Delta t}{\hbar}) + i\cos(\theta)\sin(\frac{\Delta t}{\hbar}) & i\sin(\theta)\sin(\frac{\Delta t}{\hbar}) \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \cos(\frac{\Delta t}{\hbar}) - i\cos(\theta)\sin(\frac{\Delta t}{\hbar}) \\ -i\sin(\theta)\sin(\frac{\Delta t}{\hbar}) \end{bmatrix} \\
&= \begin{bmatrix} \cos(\frac{\Delta t}{\hbar}) + i\cos(\theta)\sin(\frac{\Delta t}{\hbar}) & i\sin(\theta)\sin(\frac{\Delta t}{\hbar}) \end{bmatrix} \begin{bmatrix} \cos(\frac{\Delta t}{\hbar}) - i\cos(\theta)\sin(\frac{\Delta t}{\hbar}) \\ i\sin(\theta)\sin(\frac{\Delta t}{\hbar}) \end{bmatrix} \\
&= \cos^2(\frac{\Delta t}{\hbar}) + \cos^2(\theta)\sin^2(\frac{\Delta t}{\hbar}) - \sin^2(\theta)\sin^2(\frac{\Delta t}{\hbar}) \\
&= \cos^2(\frac{\Delta t}{\hbar}) + (\cos^2(\theta) - \sin^2(\theta))\sin^2(\frac{\Delta t}{\hbar})
\end{aligned}$$

Below we plot the expectation values of the three pauli matrices as functions of time, using our analytic solutions:

```python
import numpy as np
import matplotlib.pyplot as plt

theta = 0.2 * np.pi
delta = 2 * np.pi

hbar = 1.0
times = np.linspace(0, 5, 1000)

def sigma_x_expectation_value(t):
    return 2 * np.cos(theta) * np.sin(theta) * np.sin((delta * t)/hbar)**2

def sigma_y_expectation_value(t):
    return -2 *np.sin(theta) * np.sin((delta * t)/hbar) * np.cos((delta * t)/hbar)

def sigma_z_expectation_value(t):
    return np.cos((delta * t)/hbar)**2  + (np.cos(theta)**2 - np.sin(theta)**2) * np.sin((
delta * t)/hbar)**2

sigma_x_expectation_values = sigma_x_expectation_value(times)
sigma_y_expectation_values = sigma_y_expectation_value(times)
sigma_z_expectation_values = sigma_z_expectation_value(times)

plt.figure(figsize=(10, 6))
plt.plot(times,sigma_x_expectation_values)
plt.title("Expectation Value of $\sigma_x$")
plt.xlabel("Time")
plt.ylabel(r"$\langle \sigma_x \rangle$")
plt.grid(True)
plt.tight_layout()
plt.show()


plt.figure(figsize=(10, 6))
plt.plot(times, sigma_y_expectation_values)
plt.title("Expectation Value of $\sigma_y$")
plt.xlabel("Time")
plt.ylabel(r"$\langle \sigma_y \rangle$")
plt.grid(True)
plt.tight_layout()
plt.show()


plt.figure(figsize=(10, 6))
plt.plot(times, sigma_z_expectation_values)
plt.title("Expectation Value of $\sigma_z$")
plt.xlabel("Time")
plt.ylabel(r"$\langle \sigma_z \rangle$")
plt.grid(True)
plt.tight_layout()
plt.show()
```

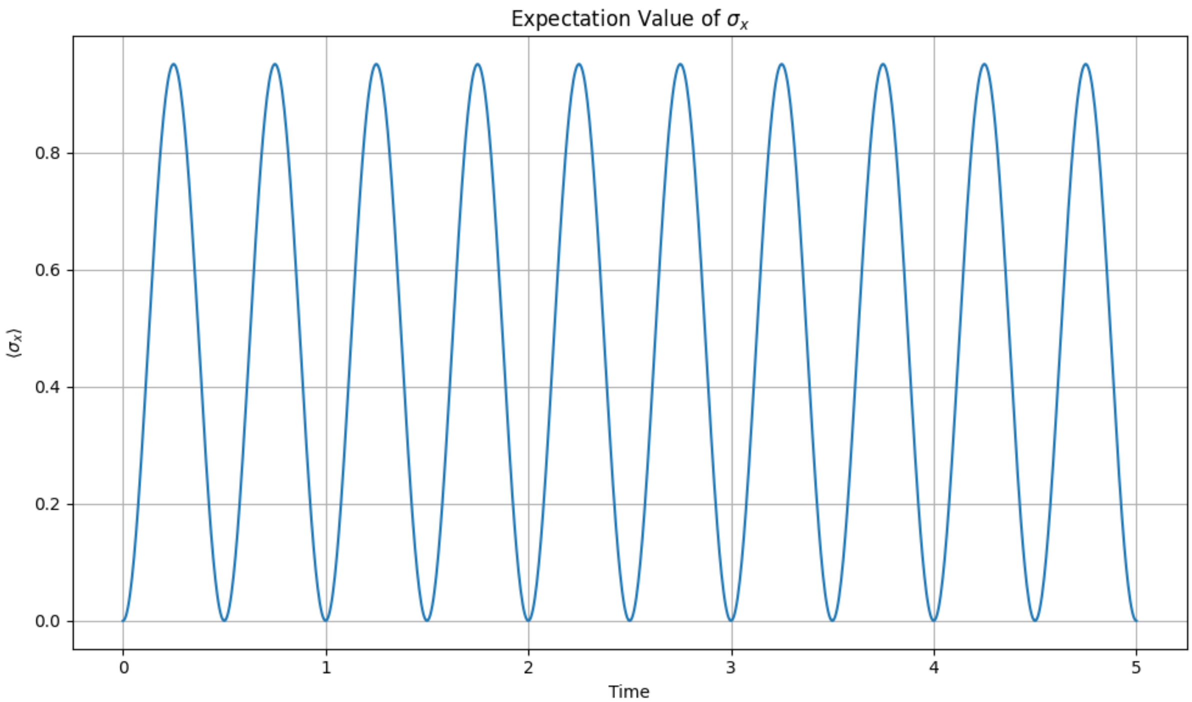

Figure 9: Expectation value of $\hat{\sigma}_x$ as a function of time, using our analytic result.

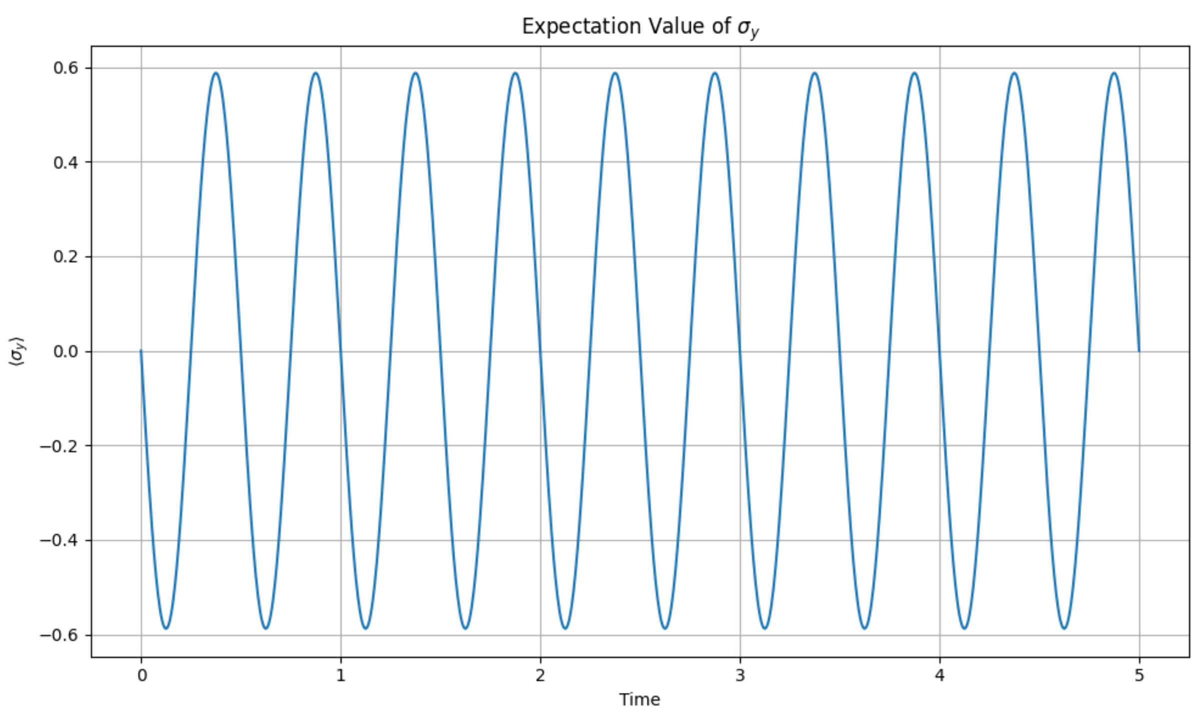

Figure 10: Expectation value of $\hat{\sigma}_y$ as a function of time, using our analytic result.

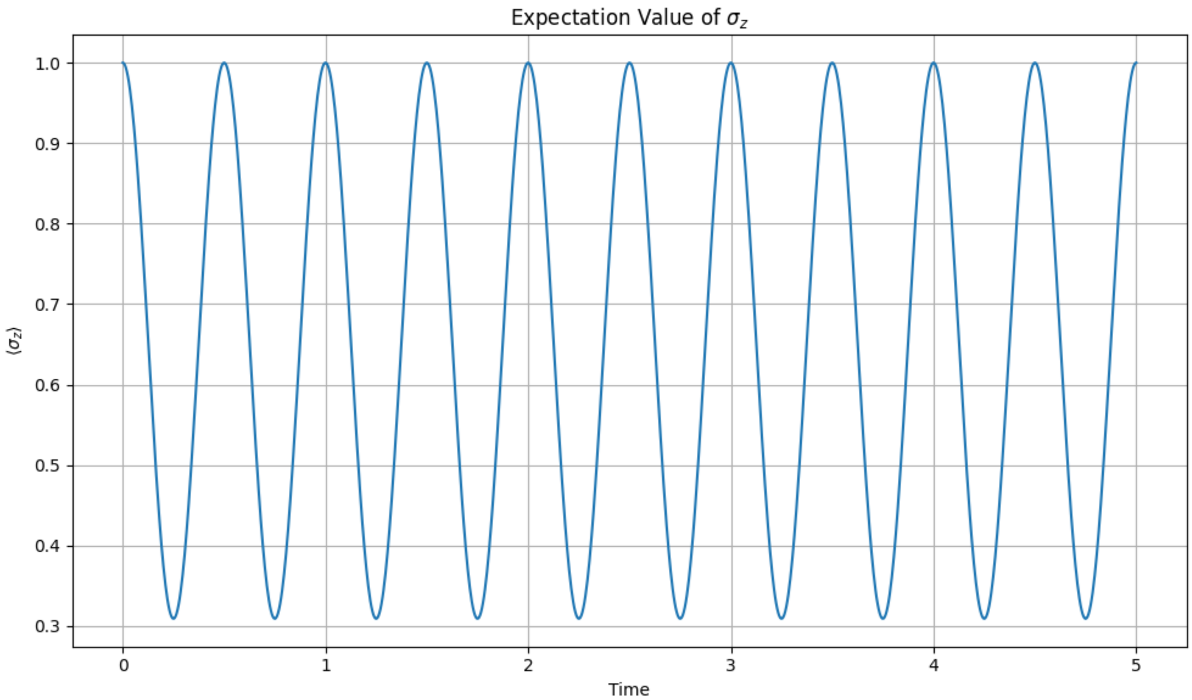

Figure 11: Expectation value of $\hat{\sigma}_z$ as a function of time, using our analytic result.

And then we plot again using QuTip's **mesolve()**

```
import matplotlib.pyplot as plt
import numpy as np
from qutip import Bloch, about, basis, mesolve, sigmam, sigmax, sigmay, sigmaz

theta = 0.2 * np.pi
delta = 2 * np.pi

psi0 = basis(2, 0)

tlist = np.linspace(0, 5, 1000)

H = delta * (np.cos(theta) * sigmaz() + np.sin(theta) * sigmax())

result = mesolve(H, psi0, tlist, [], [sigmax(), sigmay(), sigmaz()])
exp_sx_circ, exp_sy_circ, exp_sz_circ = result.expect

plt.figure(figsize=(10, 6))
plt.plot(tlist, exp_sx_circ)
plt.title("Expectation Value of $\sigma_x$")
plt.xlabel("Time")
plt.ylabel(r"$\langle \sigma_x \rangle$")
plt.grid(True)
plt.tight_layout()
plt.show()

plt.figure(figsize=(10, 6))
plt.plot(tlist, exp_sy_circ)
plt.title("Expectation Value of $\sigma_y$")
plt.xlabel("Time")
plt.ylabel(r"$\langle \sigma_y \rangle$")
plt.grid(True)
plt.tight_layout()
plt.show()

plt.figure(figsize=(10, 6))
plt.plot(tlist, exp_sz_circ)
plt.title("Expectation Value of $\sigma_z$")
```

```
    plt.xlabel("Time")
    plt.ylabel(r"$\langle \sigma_z \rangle$")
    plt.grid(True)
    plt.tight_layout()
    plt.show()
```

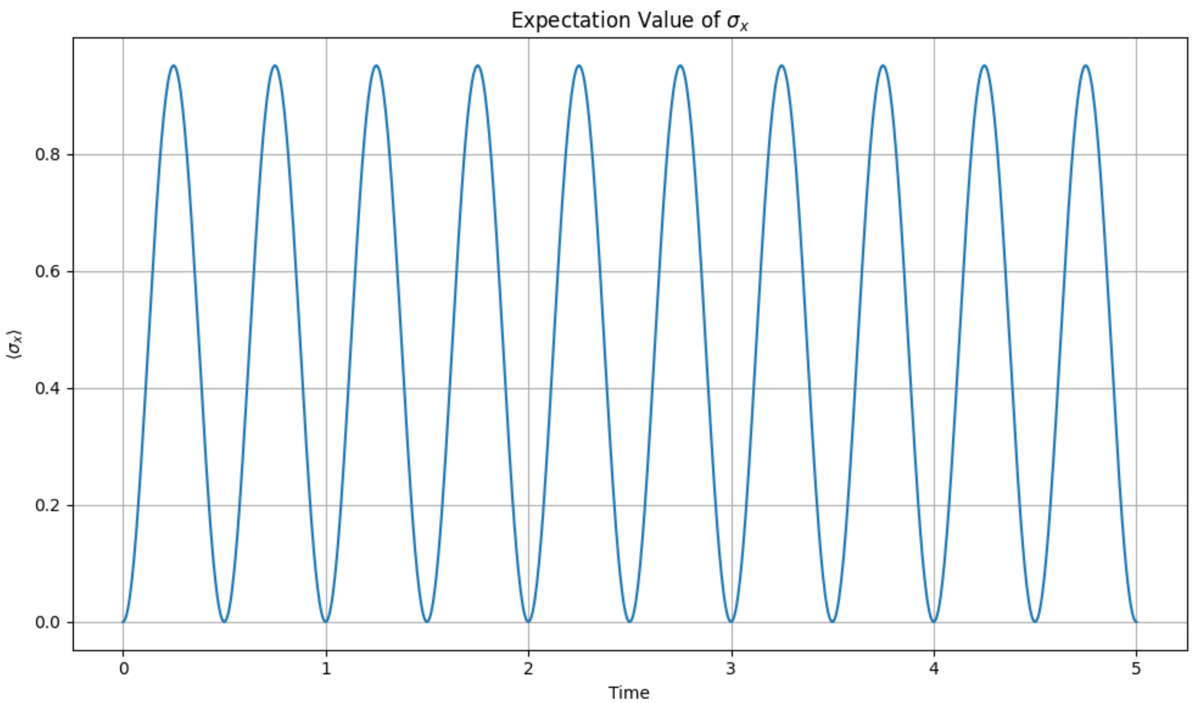

Figure 12: Expectation value of $\hat{\sigma}_x$ as a function of time, using QuTip's **mesolve()**.

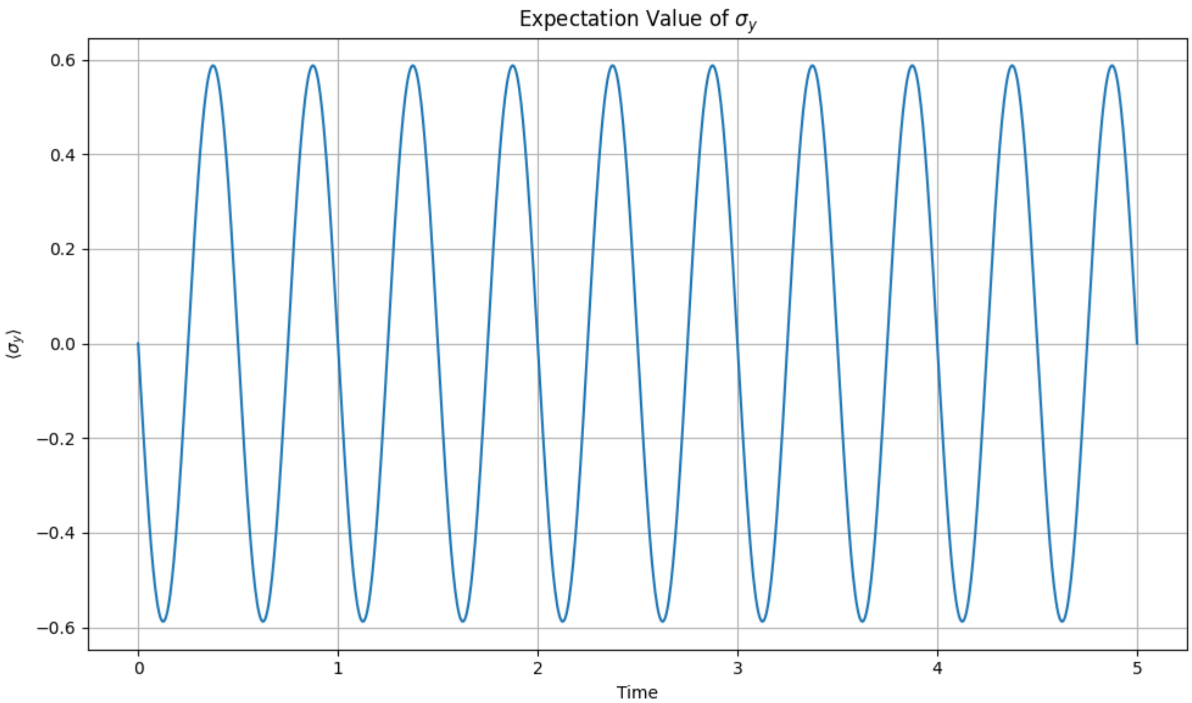

Figure 13: Expectation value of $\hat{\sigma}_y$ as a function of time, using QuTip's **mesolve()**.

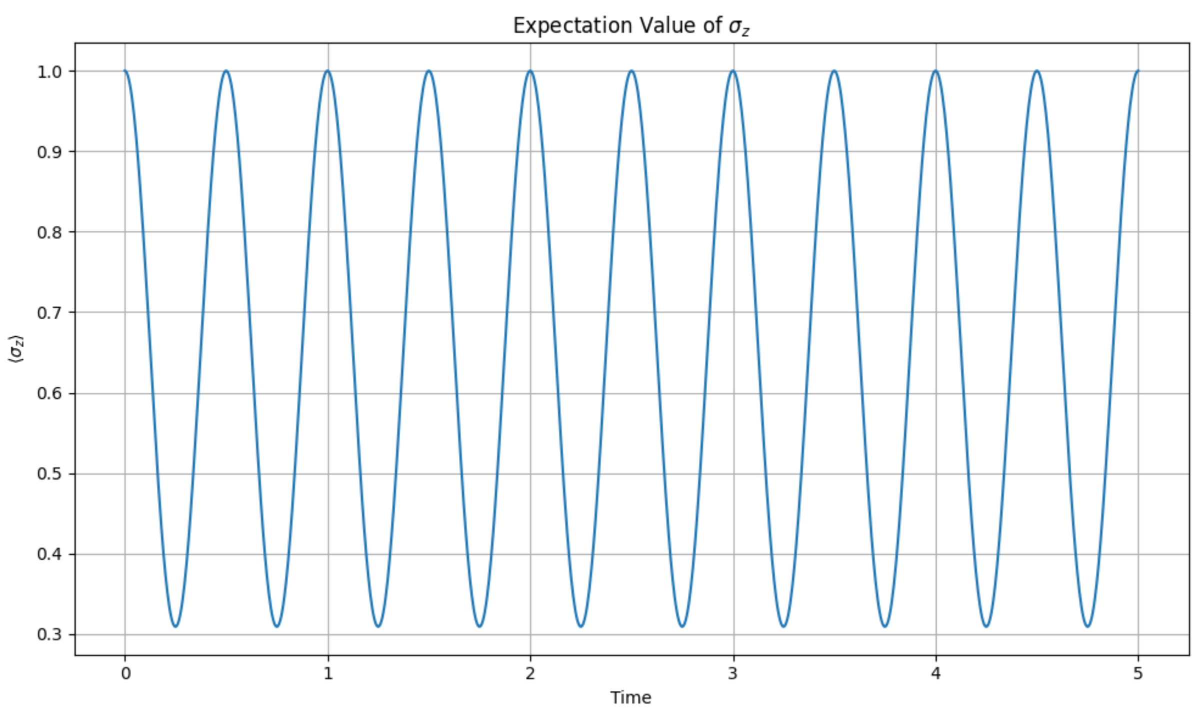

Figure 14: Expectation value of $\hat{\sigma}_z$ as a function of time, using QuTip's **mesolve()**.

We observe that we obtain identical results either using **mesolve()** or the analytic result.

3.3 Example: Use of mesolve() with collapse operators and a more complex Hamiltonian and dephasing

We consider the same system as before, with the same Hamiltonian, but now we assume it interacts with its environment.
This interaction is captured by the collapse operator:

$$\hat{C} = \sqrt{\gamma_p}\sigma_z = \begin{bmatrix} \sqrt{\gamma_p} & 0 \\ 0 & -\sqrt{\gamma_p} \end{bmatrix} \tag{39}$$

which represents the dephasing of the qubit.
It is:

$$\hat{C}^\dagger = \begin{bmatrix} \sqrt{\gamma_p} & 0 \\ 0 & -\sqrt{\gamma_p} \end{bmatrix} \tag{40}$$

We now need to switch to the density matrix formalism and utilize the Lindblad master equation.
We calculate the commutator:

$$\begin{align}
[H(t), \rho(t)] &= H(t)\rho(t) - \rho(t)H(t) \tag{41}\\
&= \begin{bmatrix} \Delta\cos(\theta) & \Delta\sin(\theta) \\ \Delta\sin(\theta) & -\Delta\cos(\theta) \end{bmatrix} \begin{bmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{bmatrix} - \begin{bmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{bmatrix} \begin{bmatrix} \Delta\cos(\theta) & \Delta\sin(\theta) \\ \Delta\sin(\theta) & -\Delta\cos(\theta) \end{bmatrix} \tag{42}\\
&= \begin{bmatrix} \Delta\cos(\theta)\rho_{11} + \Delta\sin(\theta)\rho_{21} & \Delta\cos(\theta)\rho_{12} + \Delta\sin(\theta)\rho_{22} \\ \Delta\sin(\theta)\rho_{11} - \Delta\cos(\theta)\rho_{21} & \Delta\sin(\theta)\rho_{12} - \Delta\cos(\theta)\rho_{22} \end{bmatrix} - \begin{bmatrix} \Delta\cos(\theta)\rho_{11} + \Delta\sin(\theta)\rho_{12} & \Delta\sin(\theta)\rho_{11} - \Delta\cos(\theta)\rho_{12} \\ \Delta\cos(\theta)\rho_{21} + \Delta\sin(\theta)\rho_{22} & \Delta\sin(\theta)\rho_{21} - \Delta\cos(\theta)\rho_{22} \end{bmatrix} \tag{43}\\
&= \begin{bmatrix} \Delta\sin(\theta)\rho_{21} - \Delta\sin(\theta)\rho_{12} & 2\Delta\cos(\theta)\rho_{12} + \Delta\sin(\theta)\rho_{22} - \Delta\sin(\theta)\rho_{11} \\ \Delta\sin(\theta)\rho_{11} - \Delta\sin(\theta)\rho_{22} - 2\Delta\cos(\theta)\rho_{21} & \Delta\sin(\theta)\rho_{12} - \Delta\sin(\theta)\rho_{21} \end{bmatrix} \tag{44}
\end{align}$$

We have:

$$\begin{align}
\hat{C}\rho(t)\hat{C}^\dagger &= \begin{bmatrix} \sqrt{\gamma_p} & 0 \\ 0 & -\sqrt{\gamma_p} \end{bmatrix} \begin{bmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{bmatrix} \begin{bmatrix} \sqrt{\gamma_p} & 0 \\ 0 & -\sqrt{\gamma_p} \end{bmatrix} \tag{45}\\
&= \begin{bmatrix} \sqrt{\gamma_p} & 0 \\ 0 & -\sqrt{\gamma_p} \end{bmatrix} \begin{bmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{bmatrix} \begin{bmatrix} \sqrt{\gamma_p} & 0 \\ 0 & -\sqrt{\gamma_p} \end{bmatrix} \tag{46}\\
&= \begin{bmatrix} \sqrt{\gamma_p} & 0 \\ 0 & -\sqrt{\gamma_p} \end{bmatrix} \begin{bmatrix} \sqrt{\gamma_p}\rho_{11} & -\sqrt{\gamma_p}\rho_{12} \\ \sqrt{\gamma_p}\rho_{21} & -\sqrt{\gamma_p}\rho_{22} \end{bmatrix} \tag{47}\\
&= \begin{bmatrix} \gamma_p\rho_{11} & -\gamma_p\rho_{12} \\ -\gamma_p\rho_{21} & \gamma_p\rho_{22} \end{bmatrix} \tag{48}
\end{align}$$

It is:

$$\hat{C}^\dagger\hat{C} = \begin{bmatrix} \sqrt{\gamma_p} & 0 \\ 0 & -\sqrt{\gamma_p} \end{bmatrix} \begin{bmatrix} \sqrt{\gamma_p} & 0 \\ 0 & -\sqrt{\gamma_p} \end{bmatrix} = \gamma_p\mathbb{I} \tag{49}$$

and the anticommutator:

$$\{\hat{C}^\dagger\hat{C}\,, \rho(t)\} = \gamma_p\mathbb{I} \begin{bmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{bmatrix} + \begin{bmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{bmatrix} \gamma_p\mathbb{I} = 2\gamma_p \begin{bmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{bmatrix} = \begin{bmatrix} 2\gamma_p\rho_{11} & 2\gamma_p\rho_{12} \\ 2\gamma_p\rho_{21} & 2\gamma_p\rho_{22} \end{bmatrix} \tag{50}$$

Then the Lindblad master equation reads as:

$$\frac{d\rho}{dt} = -\frac{i}{\hbar}[H(t), \rho(t)] + (\hat{C}\rho(t)\hat{C}^{\dagger} - \frac{1}{2}\{\hat{C}^{\dagger}\hat{C}\,, \rho(t)\})$$
$$\implies \frac{d\rho}{dt} = -\frac{i}{\hbar}\begin{bmatrix} \Delta\sin(\theta)\rho_{21} - \Delta\sin(\theta)\rho_{12} & 2\Delta\cos(\theta)\rho_{12} + \Delta\sin(\theta)\rho_{22} - \Delta\sin(\theta)\rho_{11} \\ \Delta\sin(\theta)\rho_{11} - \Delta\sin(\theta)\rho_{22} - 2\Delta\cos(\theta)\rho_{21} & \Delta\sin(\theta)\rho_{12} - \Delta\sin(\theta)\rho_{21} \end{bmatrix}$$
$$+ \Big(\begin{bmatrix} \gamma_p\rho_{11} & -\gamma_p\rho_{12} \\ -\gamma_p\rho_{21} & \gamma_p\rho_{22} \end{bmatrix} - \frac{1}{2}\begin{bmatrix} 2\gamma_p\rho_{11} & 2\gamma_p\rho_{12} \\ 2\gamma_p\rho_{21} & 2\gamma_p\rho_{22} \end{bmatrix} \Big)$$
$$\implies \frac{d\rho}{dt} = -\frac{i}{\hbar}\begin{bmatrix} \Delta\sin(\theta)\rho_{21} - \Delta\sin(\theta)\rho_{12} & 2\Delta\cos(\theta)\rho_{12} + \Delta\sin(\theta)\rho_{22} - \Delta\sin(\theta)\rho_{11} \\ \Delta\sin(\theta)\rho_{11} - \Delta\sin(\theta)\rho_{22} - 2\Delta\cos(\theta)\rho_{21} & \Delta\sin(\theta)\rho_{12} - \Delta\sin(\theta)\rho_{21} \end{bmatrix}$$
$$+ \Big(\begin{bmatrix} \gamma_p\rho_{11} & -\gamma_p\rho_{12} \\ -\gamma_p\rho_{21} & \gamma_p\rho_{22} \end{bmatrix} - \begin{bmatrix} \gamma_p\rho_{11} & \gamma_p\rho_{12} \\ \gamma_p\rho_{21} & \gamma_p\rho_{22} \end{bmatrix} \Big)$$
$$\implies \frac{d\rho}{dt} = \begin{bmatrix} -i\frac{\Delta}{\hbar}\sin(\theta)\rho_{21} + i\frac{\Delta}{\hbar}\sin(\theta)\rho_{12} & -2i\frac{\Delta}{\hbar}\cos(\theta)\rho_{12} - i\frac{\Delta}{\hbar}\sin(\theta)\rho_{22} + i\frac{\Delta}{\hbar}\sin(\theta)\rho_{11} \\ -i\frac{\Delta}{\hbar}\sin(\theta)\rho_{11} + i\frac{\Delta}{\hbar}\sin(\theta)\rho_{22} + 2i\frac{\Delta}{\hbar}\Delta\cos(\theta)\rho_{21} & -i\frac{\Delta}{\hbar}\sin(\theta)\rho_{12} + i\frac{\Delta}{\hbar}\sin(\theta)\rho_{21} \end{bmatrix}$$
$$+ \begin{bmatrix} 0 & -2\gamma_p\rho_{12} \\ -2\gamma_p\rho_{21} & 0 \end{bmatrix}$$
$$\implies \frac{d\rho}{dt} = \begin{bmatrix} -i\frac{\Delta}{\hbar}\sin(\theta)\rho_{21} + i\frac{\Delta}{\hbar}\sin(\theta)\rho_{12} & -2(i\frac{\Delta}{\hbar}\cos(\theta) + \gamma_p)\rho_{12} - i\frac{\Delta}{\hbar}\sin(\theta)\rho_{22} + i\frac{\Delta}{\hbar}\sin(\theta)\rho_{11} \\ -i\frac{\Delta}{\hbar}\sin(\theta)\rho_{11} + i\frac{\Delta}{\hbar}\sin(\theta)\rho_{22} + 2(i\frac{\Delta}{\hbar}\cos(\theta) - \gamma_p)\rho_{21} & -i\frac{\Delta}{\hbar}\sin(\theta)\rho_{12} + i\frac{\Delta}{\hbar}\sin(\theta)\rho_{21} \end{bmatrix}$$

which gives us the following differential equations:

$$\frac{d\rho_{11}}{dt} = -i\frac{\Delta}{\hbar}\sin(\theta)[\rho_{21}(t) - \rho_{12}(t)] \tag{51}$$
$$\frac{d\rho_{22}}{dt} = i\frac{\Delta}{\hbar}\sin(\theta)[\rho_{21}(t) - \rho_{12}(t)] \tag{52}$$
$$\frac{d\rho_{12}}{dt} = -i\frac{\Delta}{\hbar}\sin(\theta)[\rho_{22}(t) - \rho_{11}(t)] - 2(i\frac{\Delta}{\hbar}\cos(\theta) + \gamma_p)\rho_{12} \tag{53}$$
$$\frac{d\rho_{21}}{dt} = i\frac{\Delta}{\hbar}\sin(\theta)[\rho_{22}(t) - \rho_{11}(t)] + 2(i\frac{\Delta}{\hbar}\cos(\theta) - \gamma_p)\rho_{21} \tag{54}$$

We now want to calculate the expectation value of the operators:

$$\hat{\sigma}_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \text{ (pauli-x)}$$

$$\hat{\sigma}_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \text{ (pauli-y)}$$

$$\hat{\sigma}_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \text{ (pauli-z)}$$

Remember that in the density matrix formalism, it is:

$$\langle\hat{\sigma}_x\rangle = Tr(\hat{\rho}\sigma_x) = Tr(\begin{bmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{bmatrix}\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}) = Tr(\begin{bmatrix} \rho_{12} & \rho_{11} \\ \rho_{22} & \rho_{21} \end{bmatrix}) = \rho_{12} + \rho_{21} \tag{55}$$

$$\langle\hat{\sigma}_y\rangle = Tr(\hat{\rho}\sigma_x) = Tr(\begin{bmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{bmatrix}\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}) = Tr(\begin{bmatrix} i\rho_{12} & \rho_{11} \\ \rho_{22} & -i\rho_{21} \end{bmatrix}) = i(\rho_{12} - \rho_{21}) \tag{56}$$

$$\langle\hat{\sigma}_z\rangle = Tr(\hat{\rho}\sigma_z) = Tr(\begin{bmatrix}\rho_{11} & \rho_{12}\\ \rho_{21} & \rho_{22}\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}) = Tr(\begin{bmatrix}\rho_{11} & 0\\ 0 & -\rho_{22}\end{bmatrix}) = \rho_{11} - \rho_{22} \tag{57}$$

In order to be able to calculate this, we need to obtain the solutions to the above differential equation system.
The task to do this analytically is a tedious one, so we will solve them first numerically and then calculate the expectation values of $\hat{\sigma}_x$, $\hat{\sigma}_y$, $\hat{\sigma}_z$ using the formulas above.
After that, we will do the same using QuTip's **mesolve()**.
Finally, we will plot both results for each expectation value in order to compare and verify them.

```
import numpy as np
from scipy.integrate import solve_ivp
import matplotlib.pyplot as plt
from qutip import Bloch, about, basis, mesolve, sigmam, sigmax, sigmay, sigmaz

Delta = 2 * np.pi
hbar = 1.0
gamma = 0.5
theta =  0.2 * np.pi

t_span = (0, 5)
t_eval = np.linspace(*t_span, 100)

def drho_dt(t, y):
    rho11, rho22, rho12_real, rho12_imag = y
    rho12 = rho12_real + 1j * rho12_imag
    rho21 = np.conj(rho12)

    drho11_dt = -1j * (Delta /  hbar) * np.sin(theta) * (rho21 - rho12)
    drho22_dt = 1j * (Delta / hbar) * np.sin(theta)   * (rho21 - rho12)
    drho12_dt = -1j * (Delta / hbar) * np.sin(theta)  * (rho22 - rho11) - 2 * (1j * (Delta /
hbar )* np.cos(theta) + gamma) * rho12

    return [
        drho11_dt.real,
        drho22_dt.real,
        drho12_dt.real,
        drho12_dt.imag
    ]


#----Analytic solution----
rho11_0 = 1.0
rho22_0 = 0.0
rho12_0 = 0.0 + 0.0j

y0 = [rho11_0, rho22_0, rho12_0.real, rho12_0.imag]

sol = solve_ivp(drho_dt, t_span, y0, t_eval=t_eval)

analytic_density_matrix_solution = solve_ivp(drho_dt, t_span, y0, t_eval=t_eval)

rho11 = analytic_density_matrix_solution.y[0]
rho22 = analytic_density_matrix_solution.y[1]
rho12 = analytic_density_matrix_solution.y[2] + 1j * analytic_density_matrix_solution.y[3]

sigma_x_analytic = rho12 + np.conj(rho12)
sigma_y_analytic = 1j * (rho12 - np.conj(rho12))
sigma_z_analytic = rho11 - rho22

#----QuTip solution----
psi0 = basis(2, 0)
H = Delta * (np.cos(theta) * sigmaz() + np.sin(theta) * sigmax())
c_ops = [np.sqrt(gamma) * sigmaz()]

result = mesolve(H, psi0, t_eval, c_ops, [sigmax(), sigmay(), sigmaz()])
exp_sx_circ, exp_sy_circ, exp_sz_circ = result.expect

plt.figure(figsize=(10, 6))
```

```
plt.plot(t_eval, exp_sx_circ, label="mesolve")
plt.scatter(analytic_density_matrix_solution.t, sigma_x_analytic, label="Analytic",  c="r",
marker="x")
plt.xlabel('Time')
plt.ylabel(r"$\langle \sigma_x \rangle$")
plt.legend()
plt.grid(True)
plt.show()

plt.figure(figsize=(10, 6))
plt.plot(t_eval, exp_sy_circ, label="mesolve")
plt.scatter(analytic_density_matrix_solution.t, sigma_y_analytic, label="Analytic",  c="r",
marker="x" )
plt.xlabel('Time')
plt.ylabel(r"$\langle \sigma_y \rangle$")
plt.legend()
plt.grid(True)
plt.show()

plt.figure(figsize=(10, 6))
plt.plot(t_eval, exp_sz_circ, label="mesolve")
plt.scatter(analytic_density_matrix_solution.t, sigma_z_analytic, label="Analytic", c="r",
marker="x" )
plt.xlabel('Time')
plt.ylabel(r"$\langle \sigma_z \rangle$")
plt.legend()
plt.grid(True)
plt.show()
```

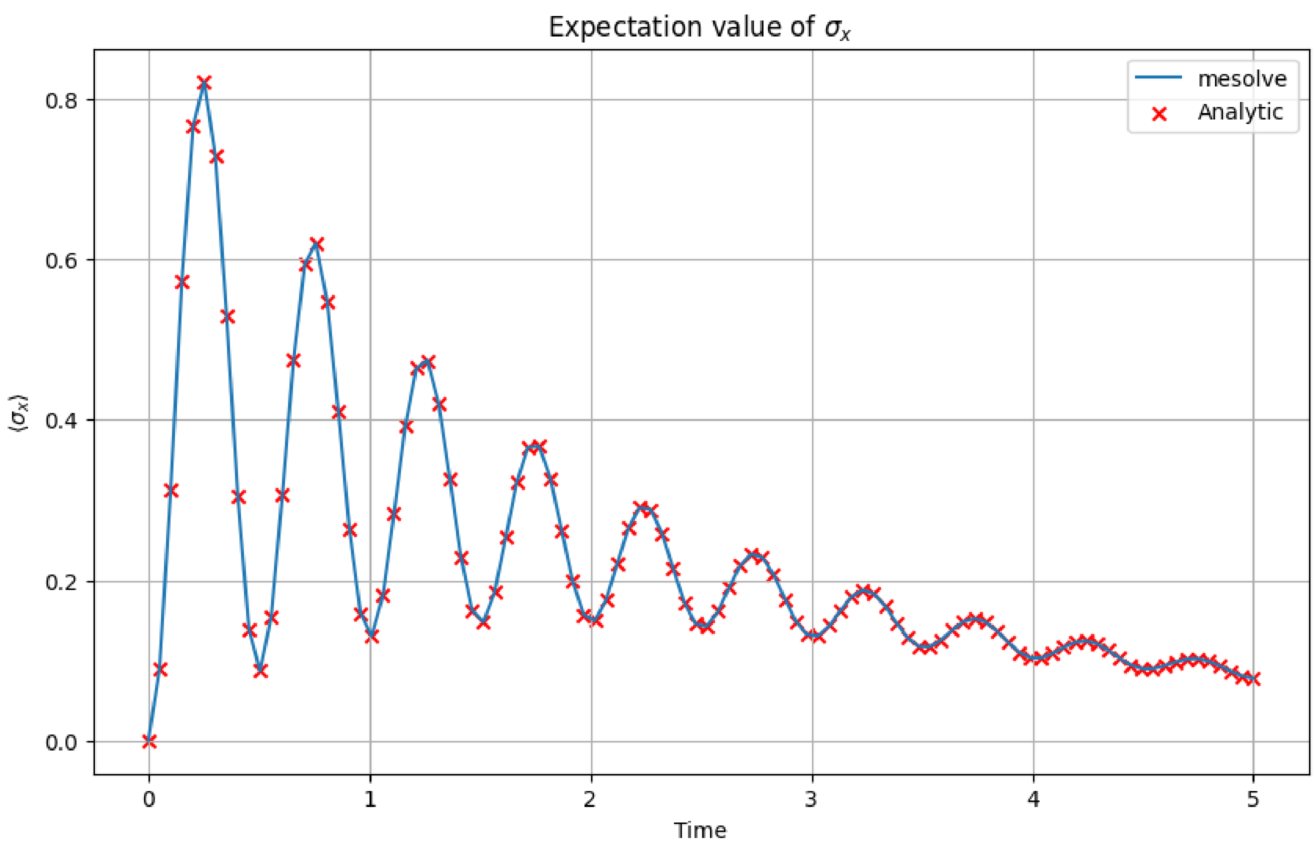

Figure 15: Plot of the expectation value of σ_x, calculated both using the analytic solution (red scattered plot) and QuTip's **mesolve()** (blue line)

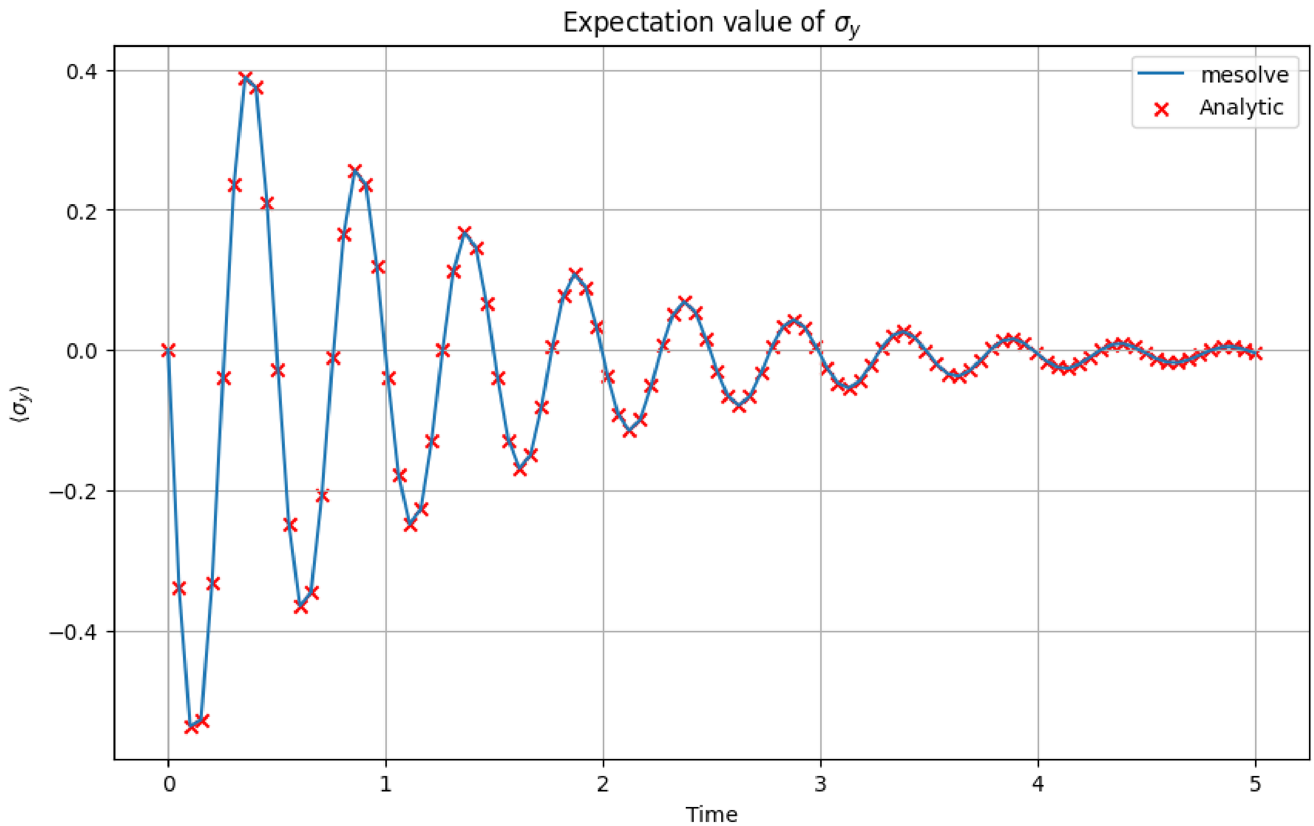

Figure 16: Plot of the expectation value of σ_y, calculated both using the analytic solution (red scattered plot) and QuTip's **mesolve()** (blue line)

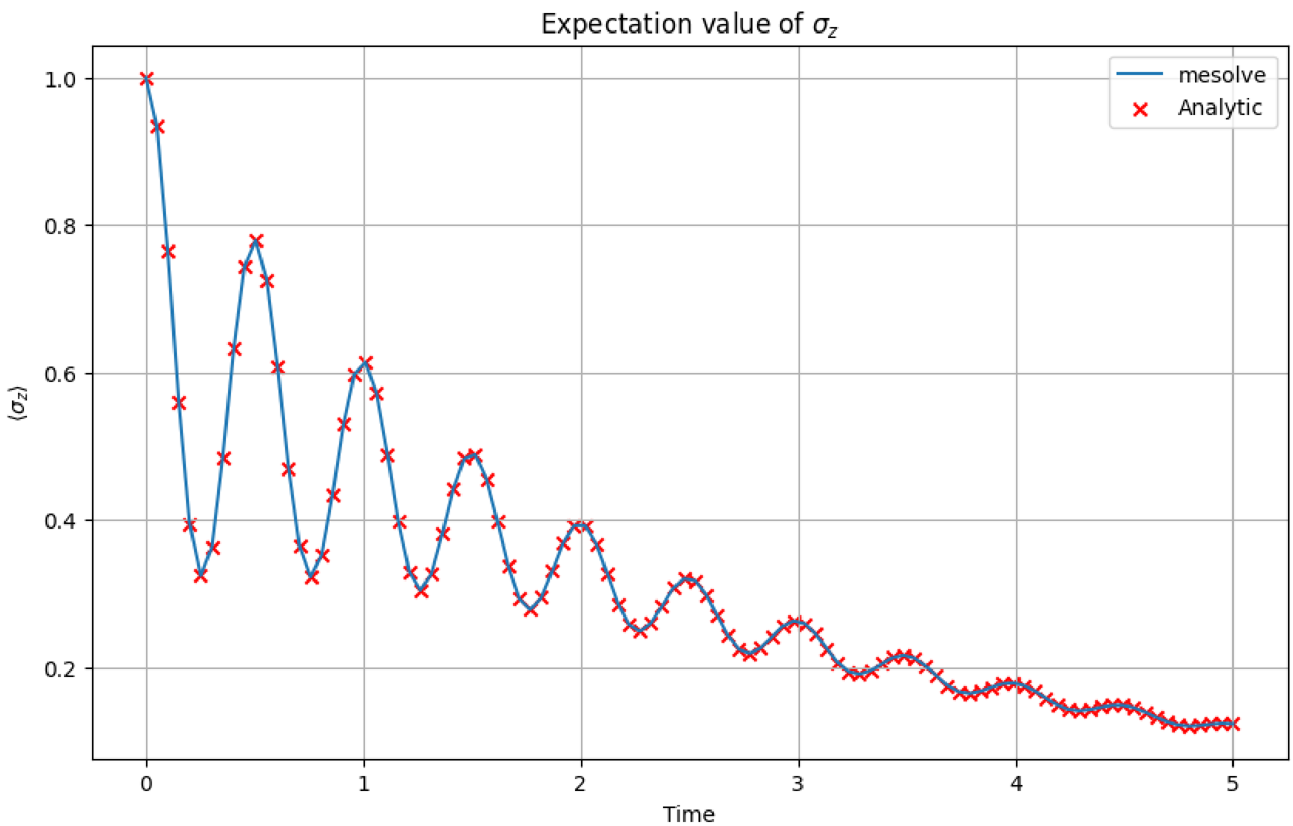

Figure 17: Plot of the expectation value of σ_z, calculated both using the analytic solution (red scattered plot) and QuTip's **mesolve()** (blue line)

We observe that both results are identical for each expectation value.

3.4 Example: Use of mesolve() with collapse operators and a more complex Hamiltonian and relaxation

Let's see another example.
We consider the same system as before, with the same Hamiltonian but this time we introduce the collapse operator:

$$\hat{C} = \sqrt{\gamma_r}\sigma_- = \begin{bmatrix} 0 & 0 \\ \sqrt{\gamma_r} & 0 \end{bmatrix} \tag{58}$$

which represents the relaxation of the qubit.
(Here $\sigma_- = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$ is the Annihilation operator for Pauli spins)
It is:

$$\hat{C}^\dagger = \begin{bmatrix} 0 & \sqrt{\gamma_r} \\ 0 & 0 \end{bmatrix} \tag{59}$$

We now need to switch to the density matrix formalism and utilize the Lindblad master equation.
We have previously calculated the commutator:

$$[H(t), \rho(t)] = \begin{bmatrix} \Delta\sin(\theta)\rho_{21} - \Delta\sin(\theta)\rho_{12} & 2\Delta\cos(\theta)\rho_{12} + \Delta\sin(\theta)\rho_{22} - \Delta\sin(\theta)\rho_{11} \\ \Delta\sin(\theta)\rho_{11} - \Delta\sin(\theta)\rho_{22} - 2\Delta\cos(\theta)\rho_{21} & \Delta\sin(\theta)\rho_{12} - \Delta\sin(\theta)\rho_{21} \end{bmatrix} \tag{60}$$

We have:

$$\hat{C}\rho(t)\hat{C}^{\dagger} = \begin{bmatrix} 0 & 0 \\ \sqrt{\gamma_r} & 0 \end{bmatrix} \begin{bmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{bmatrix} \begin{bmatrix} 0 & \sqrt{\gamma_r} \\ 0 & 0 \end{bmatrix} \tag{61}$$

$$= \begin{bmatrix} 0 & 0 \\ \sqrt{\gamma_r} & 0 \end{bmatrix} \begin{bmatrix} 0 & \sqrt{\gamma_r}\rho_{11} \\ 0 & \sqrt{\gamma_r}\rho_{21} \end{bmatrix} \tag{62}$$

$$= \begin{bmatrix} 0 & 0 \\ 0 & \gamma_r\rho_{11} \end{bmatrix} \tag{63}$$

It is:

$$\hat{C}^{\dagger}\hat{C} = \begin{bmatrix} 0 & \sqrt{\gamma_r} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ \sqrt{\gamma_r} & 0 \end{bmatrix} = \begin{bmatrix} \gamma_r & 0 \\ 0 & 0 \end{bmatrix} \tag{64}$$

So we have the anticommutator:

$$\{\hat{C}^{\dagger}\hat{C}\,, \rho(t)\} = \hat{C}^{\dagger}\hat{C}\rho(t) + \rho(t)\hat{C}^{\dagger}\hat{C} \tag{65}$$

$$= \begin{bmatrix} \gamma_r & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{bmatrix} + \begin{bmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{bmatrix} \begin{bmatrix} \gamma_r & 0 \\ 0 & 0 \end{bmatrix} \tag{66}$$

$$= \begin{bmatrix} \gamma_r\rho_{11} & \gamma_r\rho_{12} \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} \gamma_r\rho_{11} & 0 \\ \gamma_r\rho_{21} & 0 \end{bmatrix} \tag{67}$$

$$= \begin{bmatrix} 2\gamma_r\rho_{11} & \gamma_r\rho_{12} \\ \gamma_r\rho_{21} & 0 \end{bmatrix} \tag{68}$$

Then the Lindblad master equation reads as:

$$\frac{d\rho}{dt} = -\frac{i}{\hbar}[H(t), \rho(t)] + (\hat{C}\rho(t)\hat{C}^{\dagger} - \frac{1}{2}\{\hat{C}^{\dagger}\hat{C}\,, \rho(t)\})$$

$$\implies \frac{d\rho}{dt} = \begin{bmatrix} -i\frac{\Delta}{\hbar}\sin(\theta)\rho_{21} + i\frac{\Delta}{\hbar}\sin(\theta)\rho_{12} & -2i\frac{\Delta}{\hbar}\cos(\theta)\rho_{12} - i\frac{\Delta}{\hbar}\sin(\theta)\rho_{22} + i\frac{\Delta}{\hbar}\sin(\theta)\rho_{11} \\ -i\frac{\Delta}{\hbar}\sin(\theta)\rho_{11} + i\frac{\Delta}{\hbar}\sin(\theta)\rho_{22} + 2i\frac{\Delta}{\hbar}\cos(\theta)\rho_{21} & -i\frac{\Delta}{\hbar}\sin(\theta)\rho_{12} + i\frac{\Delta}{\hbar}\sin(\theta)\rho_{21} \end{bmatrix}$$

$$+ \Big(\begin{bmatrix} 0 & 0 \\ 0 & \gamma_r\rho_{11} \end{bmatrix} - \frac{1}{2} \begin{bmatrix} 2\gamma_r\rho_{11} & \gamma_r\rho_{12} \\ \gamma_r\rho_{21} & 0 \end{bmatrix} \Big)$$

$$\implies \frac{d\rho}{dt} = \begin{bmatrix} -i\frac{\Delta}{\hbar}\sin(\theta)\rho_{21} + i\frac{\Delta}{\hbar}\sin(\theta)\rho_{12} & -2i\frac{\Delta}{\hbar}\cos(\theta)\rho_{12} - i\frac{\Delta}{\hbar}\sin(\theta)\rho_{22} + i\frac{\Delta}{\hbar}\sin(\theta)\rho_{11} \\ -i\frac{\Delta}{\hbar}\sin(\theta)\rho_{11} + i\frac{\Delta}{\hbar}\sin(\theta)\rho_{22} + 2i\frac{\Delta}{\hbar}\cos(\theta)\rho_{21} & -i\frac{\Delta}{\hbar}\sin(\theta)\rho_{12} + i\frac{\Delta}{\hbar}\sin(\theta)\rho_{21} \end{bmatrix}$$

$$+ \Big(\begin{bmatrix} 0 & 0 \\ 0 & \gamma_r\rho_{11} \end{bmatrix} - \begin{bmatrix} \gamma_r\rho_{11} & \frac{1}{2}\gamma_r\rho_{12} \\ \frac{1}{2}\gamma_r\rho_{21} & 0 \end{bmatrix} \Big)$$

$$\implies \frac{d\rho}{dt} = \begin{bmatrix} -i\frac{\Delta}{\hbar}\sin(\theta)\rho_{21} + i\frac{\Delta}{\hbar}\sin(\theta)\rho_{12} & -2i\frac{\Delta}{\hbar}\cos(\theta)\rho_{12} - i\frac{\Delta}{\hbar}\sin(\theta)\rho_{22} + i\frac{\Delta}{\hbar}\sin(\theta)\rho_{11} \\ -i\frac{\Delta}{\hbar}\sin(\theta)\rho_{11} + i\frac{\Delta}{\hbar}\sin(\theta)\rho_{22} + 2i\frac{\Delta}{\hbar}\cos(\theta)\rho_{21} & -i\frac{\Delta}{\hbar}\sin(\theta)\rho_{12} + i\frac{\Delta}{\hbar}\sin(\theta)\rho_{21} \end{bmatrix}$$

$$+ \begin{bmatrix} -\gamma_r\rho_{11} & -\frac{1}{2}\gamma_r\rho_{12} \\ -\frac{1}{2}\gamma_r\rho_{21} & \gamma_r\rho_{11} \end{bmatrix}$$

$$\implies \frac{d\rho}{dt} =$$

$$\begin{bmatrix} -i\frac{\Delta}{\hbar}\sin(\theta)\rho_{21} + i\frac{\Delta}{\hbar}\sin(\theta)\rho_{12} - \gamma_r\rho_{11} & (-2i\frac{\Delta}{\hbar}\cos(\theta) - \frac{1}{2}\gamma_r)\rho_{12} - i\frac{\Delta}{\hbar}\sin(\theta)\rho_{22} + i\frac{\Delta}{\hbar}\sin(\theta)\rho_{11} \\ -i\frac{\Delta}{\hbar}\sin(\theta)\rho_{11} + i\frac{\Delta}{\hbar}\sin(\theta)\rho_{22} + (2i\frac{\Delta}{\hbar}\cos(\theta) - \frac{1}{2}\gamma_r)\rho_{21} & -i\frac{\Delta}{\hbar}\sin(\theta)\rho_{12} + i\frac{\Delta}{\hbar}\sin(\theta)\rho_{21} + \gamma_r\rho_{11} \end{bmatrix}$$

which gives us the following differential equations:

$$\frac{d\rho_{11}}{dt} = -i\frac{\Delta}{\hbar}\sin(\theta)[\rho_{21}(t) - \rho_{12}(t)] - \gamma_r\rho_{11} \tag{69}$$
$$\frac{d\rho_{22}}{dt} = i\frac{\Delta}{\hbar}\sin(\theta)[\rho_{21}(t) - \rho_{12}(t)] + \gamma_r\rho_{11} \tag{70}$$
$$\frac{d\rho_{12}}{dt} = -i\frac{\Delta}{\hbar}\sin(\theta)[\rho_{22}(t) - \rho_{11}(t)] + (-2i\frac{\Delta}{\hbar}\cos(\theta) - \frac{1}{2}\gamma_r)\rho_{12} \tag{71}$$
$$\frac{d\rho_{21}}{dt} = i\frac{\Delta}{\hbar}\sin(\theta)[\rho_{22}(t) - \rho_{11}(t)] + (2i\frac{\Delta}{\hbar}\cos(\theta) - \frac{1}{2}\gamma_r)\rho_{21} \tag{72}$$

We now want to calculate the expectation value of the operators:

$$\hat{\sigma}_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \text{ (pauli-x)}$$

$$\hat{\sigma}_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \text{ (pauli-y)}$$

$$\hat{\sigma}_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \text{ (pauli-z)}$$

Remember that in the density matrix formalism, it is:

$$\langle\hat{\sigma}_x\rangle = Tr(\hat{\rho}\sigma_x) = Tr(\begin{bmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{bmatrix}\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}) = Tr(\begin{bmatrix} \rho_{12} & \rho_{11} \\ \rho_{22} & \rho_{21} \end{bmatrix}) = \rho_{12} + \rho_{21} \tag{73}$$

$$\langle\hat{\sigma}_y\rangle = Tr(\hat{\rho}\sigma_x) = Tr(\begin{bmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{bmatrix}\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}) = Tr(\begin{bmatrix} i\rho_{12} & \rho_{11} \\ \rho_{22} & -i\rho_{21} \end{bmatrix}) = i(\rho_{12} - \rho_{21}) \tag{74}$$

$$\langle\hat{\sigma}_z\rangle = Tr(\hat{\rho}\sigma_z) = Tr(\begin{bmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}) = Tr(\begin{bmatrix} \rho_{11} & 0 \\ 0 & -\rho_{22} \end{bmatrix}) = \rho_{11} - \rho_{22} \tag{75}$$

In order to be able to calculate this, we need to obtain the solutions to the above differential equation system. The task to do this analytically is a tedious one, so we will solve them first numerically and then calculate the expectation values of $\hat{\sigma}_x$, $\hat{\sigma}_y$, $\hat{\sigma}_z$ using the formulas above.
After that, we will do the same using QuTip's **mesolve()**.
Finally, we will plot both results for each expectation value in order to compare and verify them.

```
import numpy as np
from scipy.integrate import solve_ivp
import matplotlib.pyplot as plt
from qutip import Bloch, about, basis, mesolve, sigmam, sigmax, sigmay, sigmaz

Delta = 2 * np.pi
hbar = 1.0
gamma = 0.5
theta =  0.2 * np.pi

t_span = (0, 5)
t_eval = np.linspace(*t_span, 200)


def drho_dt(t, y):
    rho11, rho22, rho12_real, rho12_imag, rho21_real, rho21_imag = y
    rho12 = rho12_real + 1j * rho12_imag
    rho21 = rho21_real + 1j * rho21_imag
```

```
    drho11_dt = -1j * (Delta /  hbar) * np.sin(theta) * (rho21 - rho12) -  gamma* rho11
    drho22_dt = 1j * (Delta / hbar) * np.sin(theta)   * (rho21 - rho12) + gamma* rho11
    drho12_dt = -1j * (Delta / hbar) * np.sin(theta)  * (rho22 - rho11) + (-2 * 1j * (Delta /
hbar )* np.cos(theta) - gamma/2 ) * rho12
    drho21_dt = 1j * (Delta / hbar) * np.sin(theta)  * (rho22 - rho11) + (2 * 1j * (Delta /
hbar )* np.cos(theta) - gamma/2 ) * rho21

    return [
        drho11_dt.real,
        drho22_dt.real,
        drho12_dt.real,
        drho12_dt.imag,
        drho21_dt.real,
        drho21_dt.imag
    ]


#----Analytic solution----
rho11_0 = 1.0
rho22_0 = 0.0
rho12_0 = 0.0 + 0.0j
rho21_0 = 0.0 + 0.0j

y0 = [rho11_0, rho22_0, rho12_0.real, rho12_0.imag, rho21_0.real, rho21_0.imag]

sol = solve_ivp(drho_dt, t_span, y0, t_eval=t_eval)

analytic_density_matrix_solution = solve_ivp(drho_dt, t_span, y0, t_eval=t_eval)

rho11 = analytic_density_matrix_solution.y[0]
rho22 = analytic_density_matrix_solution.y[1]
rho12 = analytic_density_matrix_solution.y[2] + 1j * analytic_density_matrix_solution.y[3]
rho21 = analytic_density_matrix_solution.y[4] + 1j * analytic_density_matrix_solution.y[5]

sigma_x_analytic = rho12 + rho21
sigma_y_analytic = 1j * (rho12 - rho21)
sigma_z_analytic = rho11 - rho22

#----QuTip solution----
psi0 = basis(2, 0)
H = Delta * (np.cos(theta) * sigmaz() + np.sin(theta) * sigmax())
c_ops = [np.sqrt(gamma) * sigmam()]

result = mesolve(H, psi0, t_eval, c_ops, [sigmax(), sigmay(), sigmaz()])
exp_sx_circ, exp_sy_circ, exp_sz_circ = result.expect

plt.figure(figsize=(10, 6))
plt.plot(t_eval, exp_sx_circ, label="mesolve")
plt.scatter(analytic_density_matrix_solution.t, sigma_x_analytic, label="Analytic",  c="r",
marker="x" )
plt.xlabel('Time')
plt.ylabel(r"$\langle \sigma_x \rangle$")
plt.title(r"Expectation value of $\sigma_x$")
plt.legend()
plt.grid(True)
plt.show()

plt.figure(figsize=(10, 6))
plt.plot(t_eval, exp_sy_circ, label="mesolve")
plt.scatter(analytic_density_matrix_solution.t, sigma_y_analytic, label="Analytic",  c="r",
marker="x" )
plt.xlabel('Time')
plt.ylabel(r"$\langle \sigma_y \rangle$")
plt.title(r"Expectation value of $\sigma_y$")
plt.legend()
plt.grid(True)
plt.show()

plt.figure(figsize=(10, 6))
plt.plot(t_eval, exp_sz_circ, label="mesolve")
```

```
plt.scatter(analytic_density_matrix_solution.t, sigma_z_analytic, label="Analytic",  c="r",
marker="x" )
plt.xlabel('Time')
plt.ylabel(r"$\langle \sigma_z \rangle$")
plt.title(r"Expectation value of $\sigma_z$")
plt.legend()
plt.grid(True)
plt.show()
```

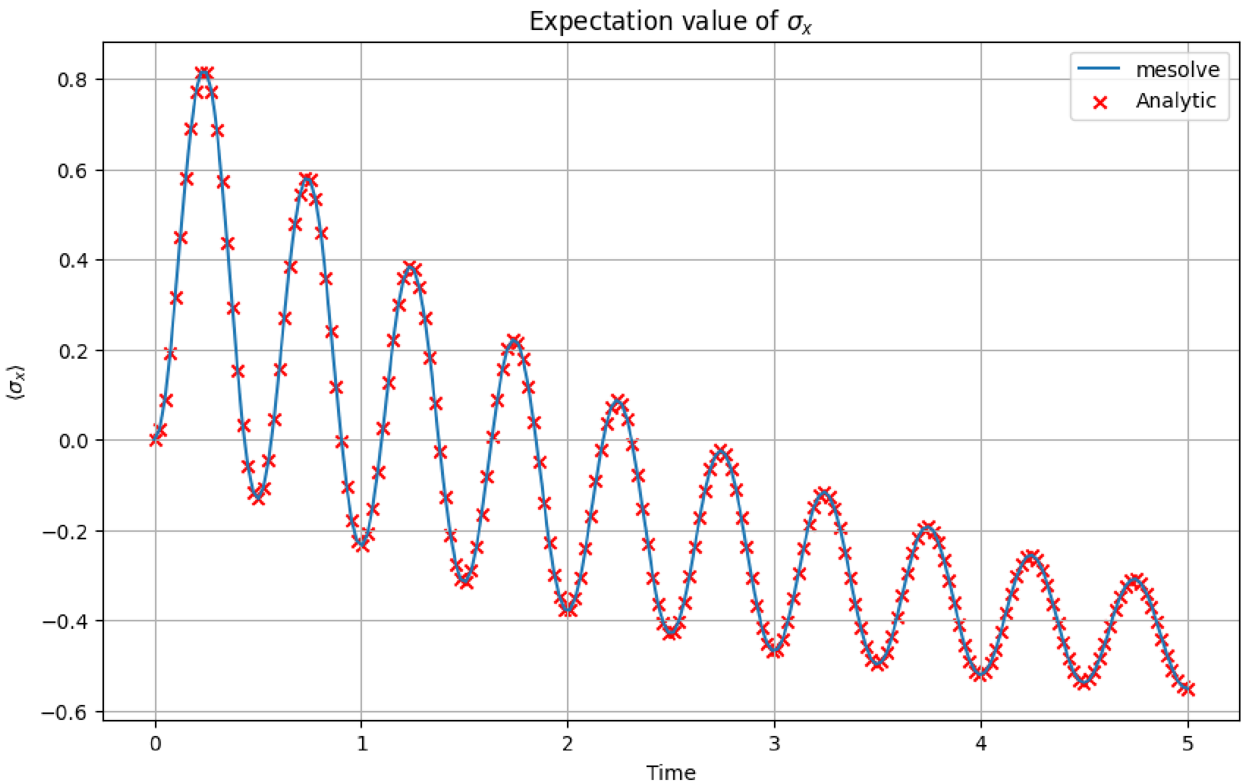

Figure 18: Plot of the expectation value of σ_x, calculated both using the analytic solution (red scattered plot) and QuTip's **mesolve()** (blue line)

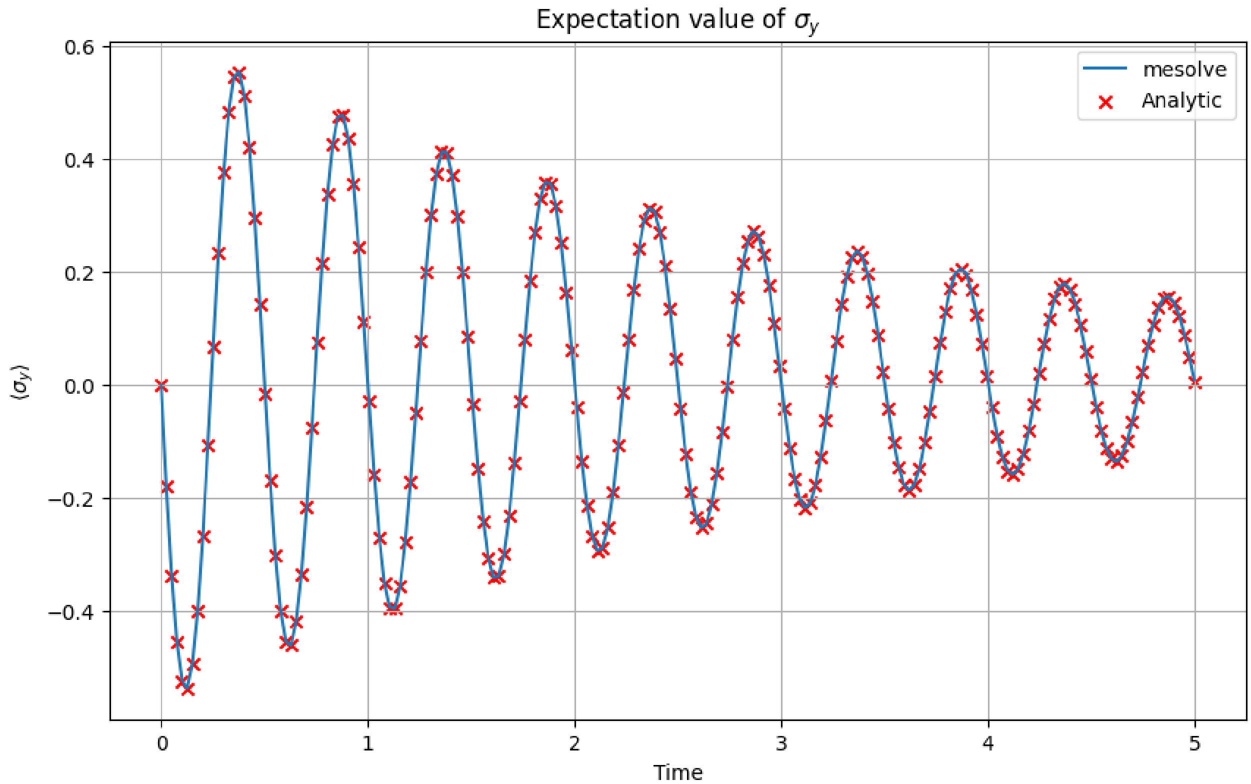

Figure 19: Plot of the expectation value of σ_y, calculated both using the analytic solution (red scattered plot) and QuTip's **mesolve()** (blue line)

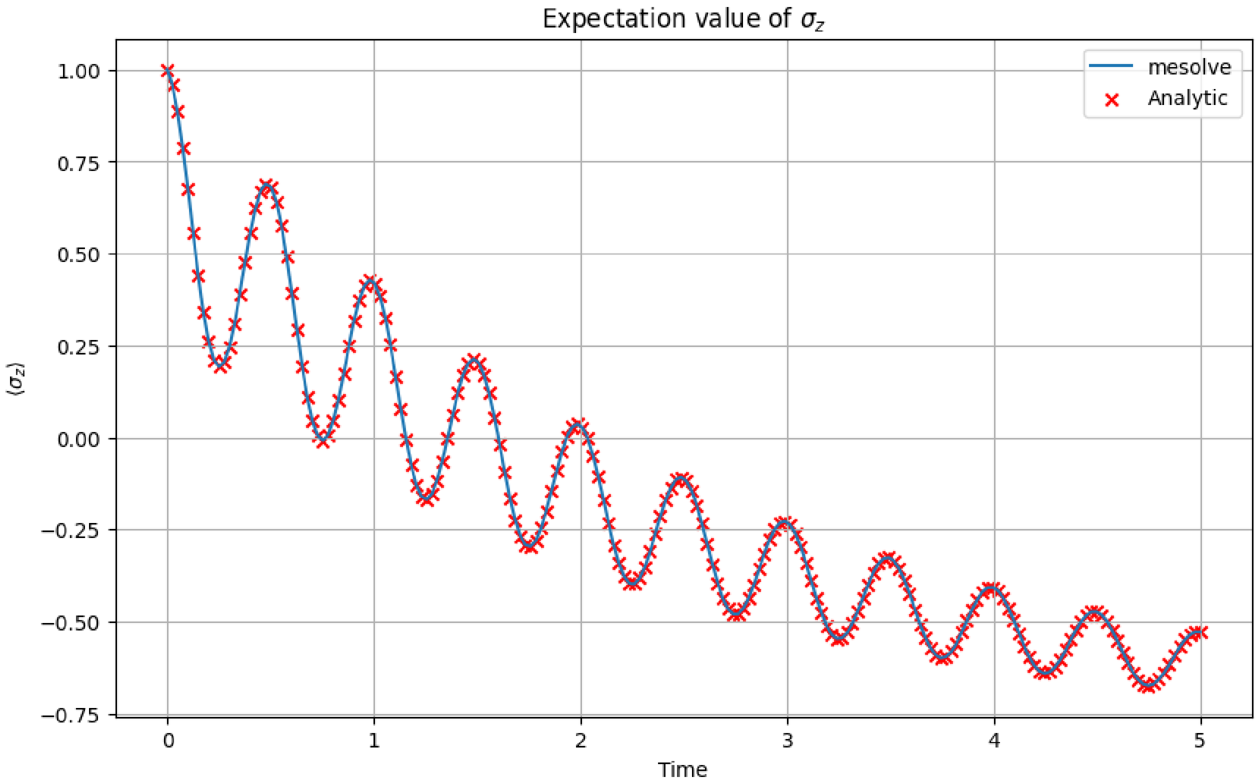

Figure 20: Plot of the expectation value of σ_z, calculated both using the analytic solution (red scattered plot) and QuTip's **mesolve()** (blue line)

We observe that both results are identical for each expectation value.

4 Floquet Solvers

A first order linear differential system is of the form:

$$\frac{dx}{dt} = A(t)x(t) + b(t)$$

where the matrix $A \in \mathbb{C}^{n\times n}$ is called the **coefficient matrix**.
A set of n linearly independent solutions $x_1, \ldots, x_n$ of the above system is called **fundamental system of solutions**.
The matrix:

$$X(t) = (x_1, \ldots, x_n)$$

is called **the fundamental matrix**.
If we have a first order linear differential system:

$$\frac{dx}{dt} = A(t)x(t)$$

and λ_i is an eigenvalue of A and v_i the corresponding eigenvector, then:

$$x_i(t) = v_i e^{\lambda_i t}$$

defines a solution of the above system.
Then again the fundamental matrix is given by $X(t) = (x_1, \dots, x_n)$, provided that $\{v_1, \dots, v_n\}$ are linearly independent.
There is a special category of differential equations: **Those with periodic coefficients**
Periodic coefficients mean that the coefficient matrix is periodic, that is $A(t) = A(t+T)$ for all $t \in \mathbb{R}$ and the period $T > 0$ In this case the fundamental matrix cannot be expressed as $X(t) = (x_1, \dots, x_n)$
Therefore, we need another approach to calculate it.
One such approach is given by Gaston Floquet who introduces a canonical decomposition of the fundamental matrix. This is captured in the following theorem:

Floquet theorem The fundamental matrix $X(t)$ of a first order homogeneous linear differential system has a **Floquet normal form**:

$$X(t) = Q(t)e^{Bt}$$

where $Q \in C^1(\mathbb{R})$ is T-periodic and the matrix $B \in \mathbb{C}^{n \times n}$ satisfies the equation $C = X(T) = e^{Bt}$. It is also $Q(0) = \mathbb{I}$ and $Q(t)$ is an invertible matrix for all t.

The Floquet theorem can be used in quantum mechanics, along with the Schrödinger equation. Together the constitue the **Floquet theory for unitary evolution**.
The TDSE reads as:

$$i\hbar \frac{d}{dt}|\psi\rangle = H(t)|\psi(t)\rangle$$

According to the Floquet theorem, there exist solutions:

$$|\psi_a(t)\rangle = e^{-i\epsilon_a t/\hbar}|\phi_a(t)\rangle$$

where:

- $|\psi_a(t)\rangle$ are the **Floquet states**
- $|\phi_a(t)\rangle = |\phi_a(t+T)\rangle$ are the periodic **Floquet modes**
- ϵ_a are the **quasienergy levels** which are constant in time and uniquely defines up to multiples of $\dfrac{2\pi}{T}$

We know that the whole dynamics of the system are captured by the unitary time evolution operator such that:

$$\hat{U}(t, t_0)|\psi(t_0)\rangle = |\psi(t)\rangle$$

Then for the Floquet states we have:

$$\hat{U}(t+T, t)|\psi_a(t)\rangle = |\psi_a(t+T)\rangle$$

And by inserting the Floquet modes that we obtained using the Floquet theorem, we get:

$$\hat{U}(t+T, t)e^{-i\epsilon_a t/\hbar}|\phi_a(t)\rangle = e^{-i\epsilon_a (t+T)/\hbar}|\phi_a(t+T)\rangle$$

and since $|\phi_a(t)\rangle = |\phi_a(t+T)\rangle$, it is:

$$\begin{aligned} \hat{U}(t+T, t)e^{-i\epsilon_a t/\hbar}|\phi_a(t)\rangle &= e^{-i\epsilon_a (t+T)/\hbar}|\phi_a(t)\rangle \\ &= \eta_a|\phi_a(t)\rangle \end{aligned}$$

This shows that the Floquet modes are eigenstates of the one-period propagator. We can therefore find the Floquet modes and quasienergies by calculating and diagonalizing $\hat{U}(t+T)$
It is particularly useful to calculate $\phi_a(0)$ by diagonalizing $\hat{U}(T,0)$
Then we find the Floquet modes at arbirary time by:

$$\begin{aligned}
&\hat{U}(t,0)|\psi_a(0)\rangle = |\psi_a(t)\rangle \\
&\implies \hat{U}(t,0)|\phi_a(0)\rangle = e^{-i\epsilon_a t/\hbar}|\phi_a(t)\rangle
\end{aligned}$$

Knowing this, we can construct a comple set of solutions for the TDSE, with the form:

$$|\psi(t)\rangle = \sum_a c_a e^{-i\epsilon_a t/\hbar}|\phi_a(t)\rangle$$

where

$$c_a = \langle\phi_a(t)\,|\,\psi(0)\rangle$$

Now we should consider again the case of an open quantum system and a dissipative environment.
A driven system that is interacting with its environment is not always well described by the Lindblad master equation that we have seen before.
Especially for periodically-driven dissipative systems the Floquet theory produces interesting and some times better results.
We assume that at the initial time the environment is in equillibrium and is uncorrelated with the system. We also assume **weak interaction** between the system and the environment at later times.
We start by the Liouville-von Neumann equation

$$\frac{d\rho_T}{dt} = -\frac{i}{\hbar}[H(t), \rho_T(t)]$$

where $H(t)$ the total Hamiltonian and $\rho_T(t)$ the density matrix of the whole system.
Under the Born approximation, the Markov approximation and the Rotating Wave approximation , we trace out the bath and obtain:

$$\frac{d\rho}{dt} = -\frac{i}{\hbar}[H(t), \rho(t)] + \mathcal{D}[\rho(\sqcup)]$$

where $\mathcal{D}$ Is a dissipator term derived from the system-bath interaction and bath correlation function.
Now we expand the density matrix in the Floquet basis:

$$\rho_{ab}(t) = \langle\phi_a(t)|\rho(t)|\phi_b(t)\rangle$$

where $\phi_i(t)$ are the Floquet modes as presented earlier.
The quasienenergy difference between two floquet modes is given by:

$$\Delta_{ab}^{(k)} = \frac{\epsilon_a - \epsilon_b}{\hbar} + k\Omega$$

where ϵ_i are the respective quasienergies and Ω the driving frequency.
So, in the Floquet representation, the dissipative term becomes a convolution involving transition rates between Floquet states. For this reason, given a coupling operator q we define the matrix X, which quantifies how the system couples to the bath and the dissipative transitions:

$$X_{ab}(t) = e^{i(\epsilon_a-\epsilon_b)t/\hbar}\langle\phi_a(t)|q|\psi_b(t)\rangle = \sum_k e^{i\Delta_{abk}t}X_{ab}^{(k)}$$

where

$$X_{ab}^{(k)} = \frac{1}{T}\int_T^0 e^{-ik\Omega t}\langle\phi_a(t)|q|\psi_b(t)\rangle$$

Given the spectral density $J(\omega)$, which tells us how strongly the environment couples to the system at different frequencies, we define the decay rates:

$$\gamma_{ab}^{(k)} = 2\pi\Theta(\Delta_{ab}^{(k)})J(\Delta_{ab}^{(k)})|X_{ab}^{(k)}|^2$$

where $\Theta(\Delta_{ab}^{(k)})$ is the Heaviside function.
Then, **under the Rotating Wave Approximation**, we obtain two decoupled subsets of equations. One subset describes the approach of the diagonal elements towards a steady state (dissipation), while the other is concerned with the decay of the off-diagonal elements (effect of decoherence). Those equations constitue the **Floquet-Markov master equation** and read as:

$$\frac{d\rho_{aa}}{dt} = \sum_l [A_{al}\rho_{ll}(t) - A_{la}\rho_{aa}(t)]$$

$$\frac{d\rho_{ab}}{dt} = -\frac{1}{2}\sum_l (A_{la} + A_{lb})\rho_{ab}(t) \text{ for } a \neq b$$

with

$$A_{ab} = \sum_k \left[\gamma_{ab}^{(k)} + n_{th}(|\Delta_{ab}^{(k)}|)(\gamma_{ab}^{(k)} + \gamma_{ba}^{(-k)})\right]$$

The time-independent coefficients A_{ab} contain the specifics of the potential and driving through the quasienergy differences $\Delta_{ab}^{(k)}$ and Floquet modes via $X_{ab}^{(k)}$
The equation for the nondiagonal elements predicts exponentially vanishing off-diagonal elements at long time. Hence, the reduced density matrix becomes diagonal in the Floquet basis at long times.

4.1 Floquet Schrödinger Equation in Qutip

Qutip supports the Floquet formalism out of the box.
For the Schrödinger equation, it provides the method **fsesolve()** which works similarly to sesolve().
It accepts the following parameters:

- **H : qutip.qobj.Qobj**, the periodic system's time dependent Hamiltonian
- **psi0 : qutip.qobj**, the initial state vector
- **tlist: list / array**, a list of times for t.
- **e_ops : list of qutip.qobj**, a list of operators for which to evaluate expectation values
- **T: float**, the period of the time-dependence of the hamiltonian
- **args: dictionary**, a dictionary with variables required to evaluate H

It retruns an instance of the class **qutip.solver.Result**, which contains either an array of expectation values or an array of state vectors, for the times specified by tlist.
Its use is demonstrated with the following code:

```python
import numpy as np
from qutip import (about, basis, fmmesolve, fsesolve,
                   plot_expectation_values, sigmax, sigmaz)


delta = 0.2 * 2 * np.pi
eps0 = 2 * np.pi
A = 2.5 * 2 * np.pi
omega = 2 * np.pi

H0 = -delta / 2.0 * sigmax() - eps0 / 2.0 * sigmaz()

H1 = [A / 2.0 * sigmaz(), "sin(w*t)"]
args = {"w": omega}

H = [H0, H1]

psi0 = basis(2, 0)

T = 2 * np.pi / omega

tlist = np.linspace(0, 2.5 * T, 101)

result = fsesolve(H, psi0, tlist, T=T, e_ops=[sigmaz()],args=args)

plot_expectation_values([result], ylabels=["<Z>"])
```

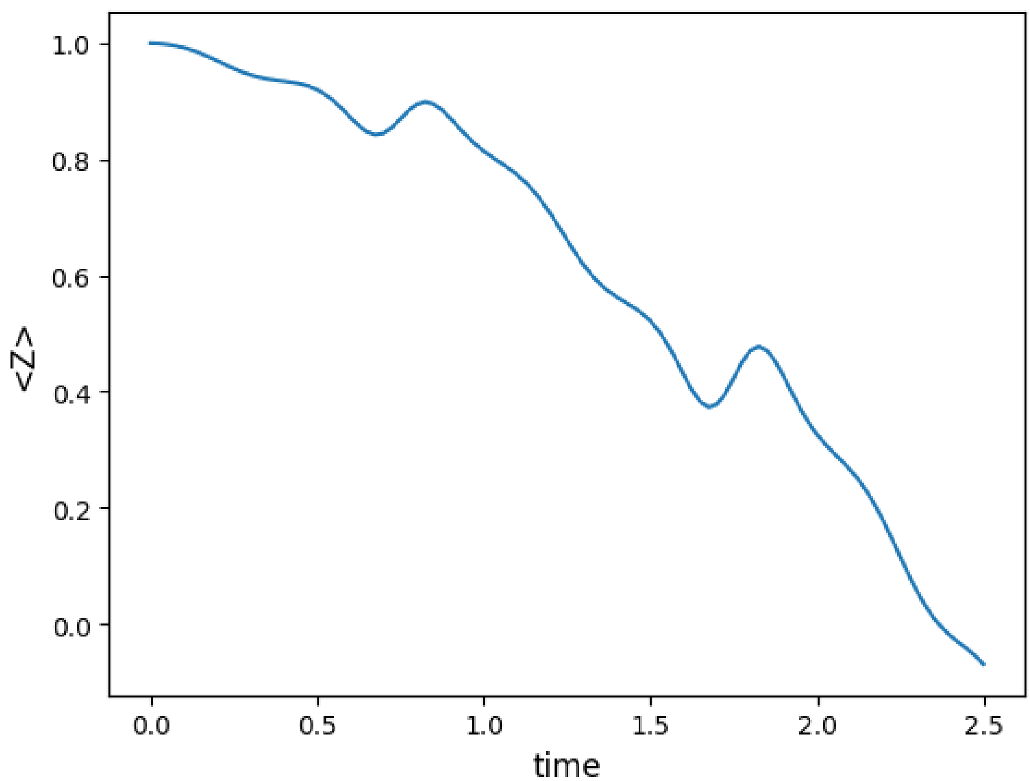

Figure 21: Result of use of **fsesolve()** in Qutip

4.2 Floquet Markov Master Equation in Qutip

Qutip supports the Floquet formalism to solve a master equation for a dissipative quantum systemout of the box. For the Master equation, it provides the method **fmmesolve()** which works similarly to mesolve(). However, the dissipation process is here described as a noise spectral-density function.
It accepts the following parameters:

- **H : qutip.qobj.Qobj**, the periodic system's time dependent Hamiltonian
- **psi0 : qutip.qobj**, the initial state vector
- **tlist: list / array**, a list of times for t.
- **c_ops : list of qutip.qobj**, List of collapse operators. Time dependent collapse operators are not supported. If this argument is not provided the method falls back to fsesolve
- **e_ops : list of qutip.qobj**, alist of operators for which to evaluate expectation values
- **spectra_cb: list of functions**, a list of callback functions that compute the noise power spectrum as a function of frequency for the collapse operators in c_ops
- **T: float**, the period of the time-dependence of the hamiltonian
- **args: dictionary**, a dictionary with variables required to evaluate H

Its use is demonstrated with the following code:

```python
import numpy as np
from qutip import (about, basis, fmmesolve, fsesolve,
                   plot_expectation_values, sigmax, sigmaz)

def noise_spectrum(omega):
    return gamma * omega / (4 * np.pi)

gamma = 0.1
delta = 0.2 * 2 * np.pi
eps0 = 2 * np.pi
A = 2.5 * 2 * np.pi
omega = 2 * np.pi

H0 = -delta / 2.0 * sigmax() - eps0 / 2.0 * sigmaz()

H1 = [A / 2.0 * sigmaz(), "sin(w*t)"]
args = {"w": omega}

H = [H0, H1]

psi0 = basis(2, 0)

c_ops = [sigmax()]

T = 2 * np.pi / omega

tlist = np.linspace(0, 2.5 * T, 101)

fme_result = fmmesolve(
    H,
    psi0,
    tlist,
    c_ops=c_ops,
    spectra_cb=[noise_spectrum],
    e_ops=[sigmaz()],
    T=T,
    args=args
)

plot_expectation_values([fme_result],ylabels=["<Z>"])
```

```
10.0%. Run time:   0.00s. Est. time left: 00:00:00:00
20.0%. Run time:   0.00s. Est. time left: 00:00:00:00
30.0%. Run time:   0.00s. Est. time left: 00:00:00:00
40.0%. Run time:   0.01s. Est. time left: 00:00:00:00
50.0%. Run time:   0.01s. Est. time left: 00:00:00:00
60.0%. Run time:   0.01s. Est. time left: 00:00:00:00
70.0%. Run time:   0.01s. Est. time left: 00:00:00:00
80.0%. Run time:   0.01s. Est. time left: 00:00:00:00
90.0%. Run time:   0.01s. Est. time left: 00:00:00:00
100.0%. Run time:   0.02s. Est. time left: 00:00:00:00
Total run time:   0.02s
```

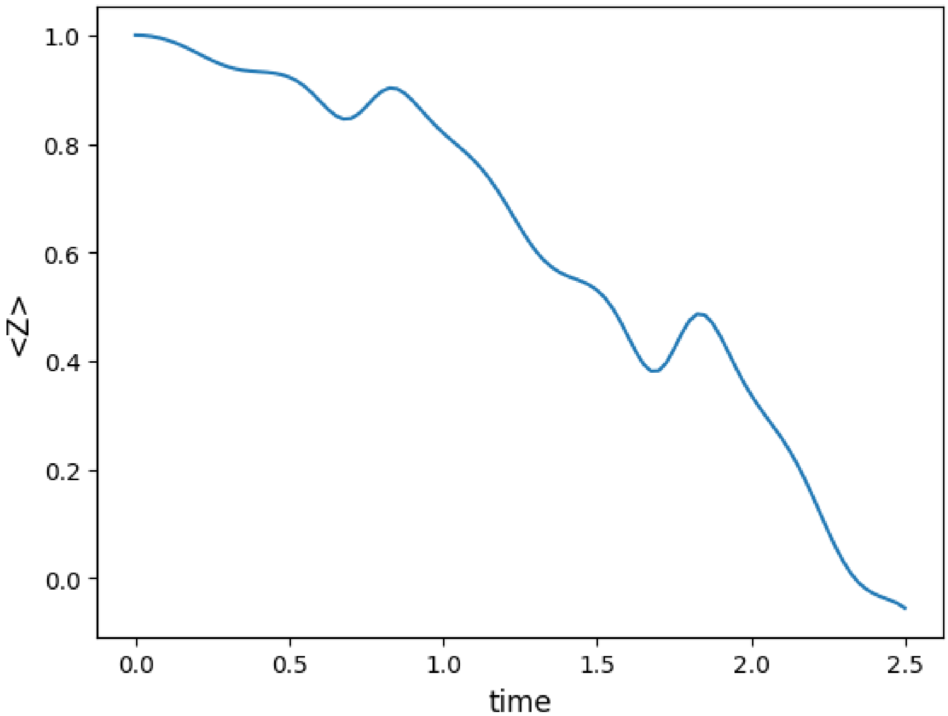

Figure 22: Result of use of **fmmesolve()** in Qutip

References

[1] `https://eclass.duth.gr/modules/document/index.php?course=1031603&openDir=/6442f3c67ilM`

[2] `https://qutip.org/docs/4.0.2/guide/dynamics/dynamics-master.html`

[3] `https://qutip.org/docs/4.7/guide/dynamics/dynamics-floquet.html`

[4] `https://www.12000.org/my_notes/liapunov_floquet_transformation/bMATH_2018_FolkersE.pdf`

[5] `https://www.ggi.infn.it/sft/SFT_2019/LectureNotes/Santoro.pdf`

[6] `https://iopscience.iop.org/article/10.1088/1361-6455/abb127/pdf`

[7] `https://www.physik.uni-augsburg.de/theo1/hanggi/Papers/213.pdf`

[8] `https://arxiv.org/pdf/2203.16358`

[9] `https://qutip.readthedocs.io/en/latest`