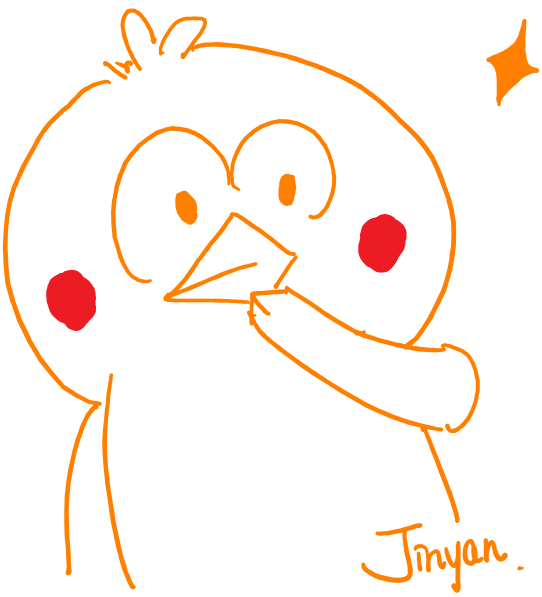
Homework 1

Math 542 - Quantum Optics
Professor Alex Kuzmich



Jinyan Miao

Fall 2023

1

Via Eq. (2.55) from the textbook, we have

$$i\hbar\,\dot{\mathbf{a}}(t) = \frac{\hbar}{2}\begin{bmatrix} -\omega_0 & 2|\Omega_0(t)|\cos(\omega t - \phi(t)) \\ 2|\Omega_0(t)|\cos(\omega t - \phi(t)) & \omega_0 \end{bmatrix}\mathbf{a}(t)\,, \tag{1.1}$$

with the definition

$$\Omega_0(t) = \frac{-(\mu_z)_{21}\,E_0(t)}{\hbar} = |\Omega_0(t)|\,e^{i\,\phi(t)}\,.$$

With the assumption that $\Omega_0 \in \mathbb{R}$ Eq. (1.1) becomes

$$i\hbar\,\dot{\mathbf{a}}(t) = \frac{\hbar}{2}\begin{bmatrix} -\omega_0 & 2\Omega_0\cos(\omega t) \\ 2\Omega_0\cos(\omega t) & \omega_0 \end{bmatrix}\mathbf{a}(t)\,.$$

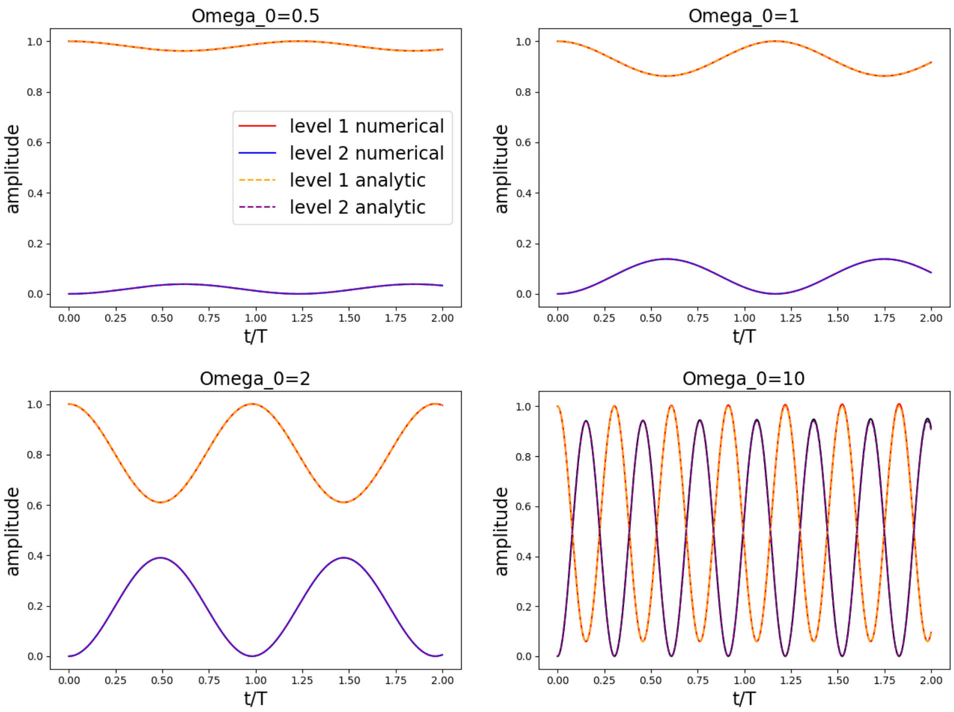
Here we denote $m = i\omega_0\hbar/2$, and $c = -i2\Omega_0\cos(\omega t)$. Then we have

$$\dot{\mathbf{a}} = \mathbf{M}\,\mathbf{a}(t) = \begin{bmatrix} m & c \\ c & -m \end{bmatrix}\mathbf{a}(t)\,.$$

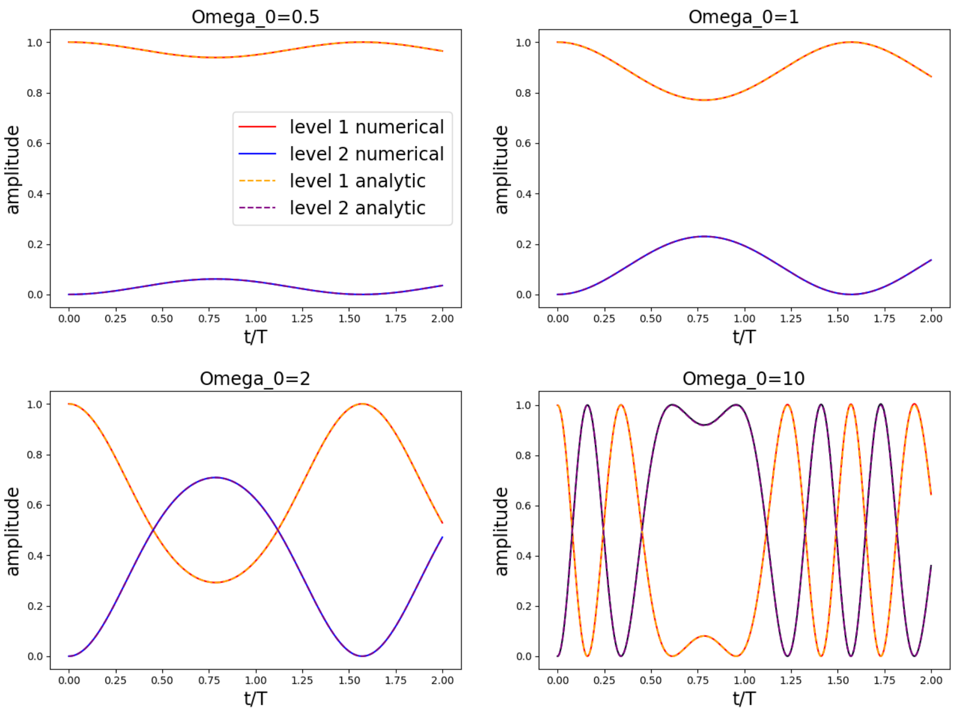
Here we will solve the system as a function of t/T numerically with initial conditions $a_1(0) = 1$ and $a_2(0) = 0$.

* *The amplitude of $a_1(t)$ and $a_2(t)$ are plotted on the next page.*
** *The code is attached at the end of this text.*
*** *For simplicity, the assumption that $\hbar = 1$ has been made.*

Assume $\omega T = 0$, $\Omega_0 T \in \{0.5, 1, 2, 10\}$, $\omega_0 T = 5$.



Assume $\omega T = 2$, $\Omega_0 T \in \{0.5, 1, 2, 10\}$, $\omega_0 T = 0$.



2

Here we consider the system

$$i\hbar\dot{\widetilde{\mathbf{c}}} = \widetilde{\mathbf{H}}\,\widetilde{\mathbf{c}}\,, \tag{2.1}$$

where we have

$$\widetilde{\mathbf{H}} = \frac{\hbar}{2}\begin{bmatrix} -\delta & \Omega_0 \exp(-(t/T)^2) \\ \Omega_0 \exp(-(t/T)^2) & \delta \end{bmatrix}.$$

Simplifying (2.1) we obtain

$$\dot{\widetilde{\mathbf{c}}} = -i\begin{bmatrix} -\delta/2 & \Omega_0 \exp(-(t/T)^2)/2 \\ \Omega_0 \exp(-(t/T)^2)/2 & \delta/2 \end{bmatrix}\widetilde{\mathbf{c}}\,. \tag{2.2}$$

Here Eq. (2.2) agrees with the form of Eq. (2.151) from the textbook,

$$\dot{\widetilde{\mathbf{c}}}(t) = -i\,\mathbf{A}\,\widetilde{\mathbf{c}}\,.$$

It is not hard to check that the matrix $\mathbf{A}$ has eigenvalues

$$\Lambda_{1,2}(t) = \pm\sqrt{\delta^2 + \Omega_0^2 \exp(-2(t/T)^2)}/2 = \pm\Omega(t)/2\,.$$

Here we set

$$\widetilde{\mathbf{c}} = \mathbf{T}\,\mathbf{x}\,, \tag{2.3}$$

where $\mathbf{T}$ diagonalizes $\mathbf{A}$, then via Eq. (2.154) from the text, we have

$$\dot{\mathbf{x}} = -i\boldsymbol{\Lambda}\,\mathbf{x} - \mathbf{T}^\dagger\dot{\mathbf{T}}\,\mathbf{x}\,,$$

where $\boldsymbol{\Lambda} = \mathrm{diag}(\Lambda_1, \Lambda_2)$. We note further that time-dependent terms in $\mathbf{T}$ can only be in the form of $\sim e^{t^2/T^2}$, thus all nonzero entries in $\dot{\mathbf{T}}$ involves factor of $1/T^2$. We conclude that entries in $\mathbf{T}^\dagger\dot{\mathbf{T}}$, which have order $1/T^2$, are much less than $|\Lambda_1 - \Lambda_2| \sim \Omega(t)$ because $|\Omega(t)\,T| \gg 1$ by assumption. According to Section 2.7.2 from the textbook, we proceed with the approximation

$$\dot{\mathbf{x}} = -i\boldsymbol{\Lambda}\,\mathbf{x}\,. \tag{2.4}$$

Expanding system (2.4), we write

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = -i\begin{bmatrix} \sqrt{\delta^2 + \Omega_0^2 \exp(-2(t/T)^2)}/2 & 0 \\ 0 & -\sqrt{\delta^2 + \Omega_0^2 \exp(-2(t/T)^2)}/2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$

That is,

$$\dot{x}_1 = -i\frac{\sqrt{\delta^2 + \Omega_0^2 \exp(-2(t/T)^2)}}{2}\, x_1\,, \qquad \dot{x}_2 = i\frac{\sqrt{\delta^2 + \Omega_0^2 \exp(-2(t/T)^2)}}{2}\, x_2\,,$$

Solving we obtain

$$x_1(t) = x_1(t_0)\ \exp\left(-i\int_{t_0}^{t} \frac{\sqrt{\delta^2 + \Omega_0^2 \exp(-2(t'/T)^2)}}{2}\, dt'\right)\,, \tag{2.5}$$

$$x_2(t) = x_2(t_0)\ \exp\left(i\int_{t_0}^{t} \frac{\sqrt{\delta^2 + \Omega_0^2 \exp(-2(t'/T)^2)}}{2}\, dt'\right)\,. \tag{2.6}$$

A computation through Mathematica finds that we have

$$\mathbf{T} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

where we have

$$\sin(\theta) = \sqrt{\frac{1}{2}\left(1 - \frac{\delta}{\Omega}\right)}\,, \qquad \cos(\theta) = \sqrt{\frac{1}{2}\left(1 + \frac{\delta}{\Omega}\right)}\,, \qquad \text{with } \Omega = \sqrt{\delta^2 + \Omega_0^2\ \exp(-2(t/T)^2)}\,.$$

Recall from (2.3), we obtain the result

$$\widetilde{c}_1 = x_1\ \cos(\theta) - x_2\ \sin(\theta)\,, \qquad \widetilde{c}_2 = x_1\ \sin(\theta) + x_2\ \cos(\theta)\,.$$

In the limit that $\delta(t) \to$ positive constant as $t \to -\infty$, we can utilize the initial condition $\widetilde{c}_1(t_0 = -\infty) = 1$ and $\widetilde{c}_2(t_0 = -\infty) = 0$, with that $\Omega(-\infty) = |\delta|$, thus $\sin(\theta)|_{t=-\infty} = 0$, $\cos(\theta)|_{t=-\infty} = 1$ we obtain

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(t_0) \\ x_2(t_0) \end{bmatrix}$$

and we find that $x_1(t_0) = 1$, $x_2(t_0) = 0$, and thus we have

$$\widetilde{c_2}(t) = x_1(t)\ \sin(\theta(t))\,,$$

with $||x_1(t)|| = 1$, and we conclude

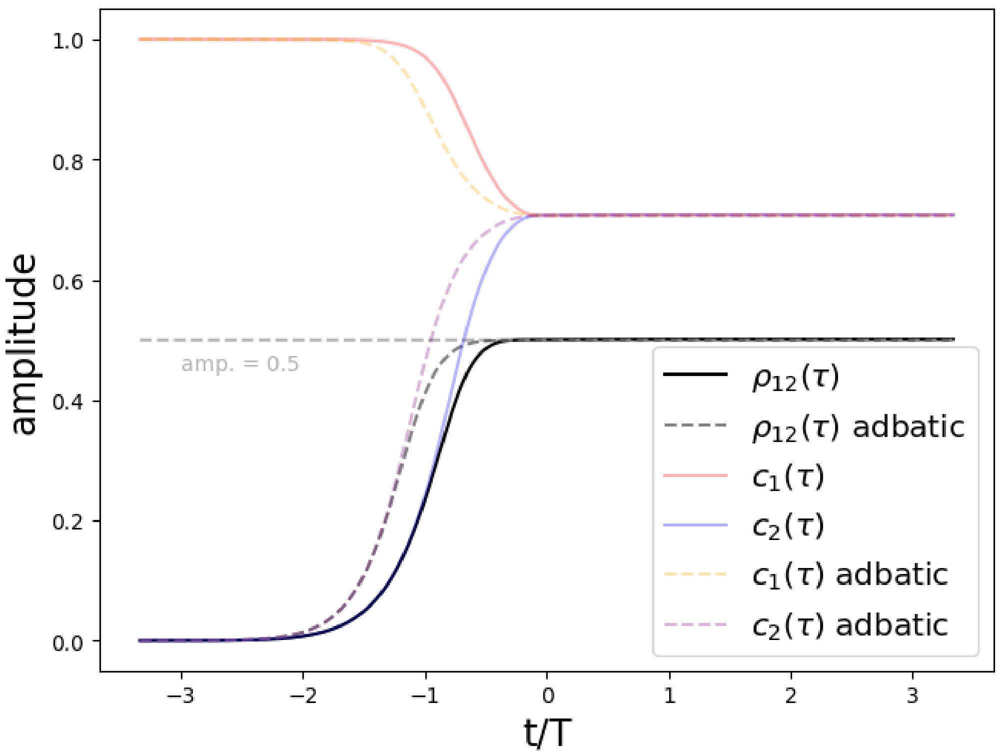
$$||\widetilde{c_2}(t)||^2 = \frac{1}{2}\left(1 - \frac{\delta(t)}{\sqrt{\delta(t)^2 + \Omega_0^2\ \exp(-2(t/T)^2)}}\right)\,, \qquad ||\widetilde{c_1}(t)||^2 = 1 - ||\widetilde{c_2}(t)||^2\,.$$

In the limit $\delta(t) \to$ negative constant as $t \to -\infty$, we can similarly compute

$$||\widetilde{c_1}(t)||^2 = \frac{1}{2}\left(1 - \frac{\delta(t)}{\sqrt{\delta(t)^2 + \Omega_0^2\ \exp(-2(t/T)^2)}}\right)\,, \qquad ||\widetilde{c_2}(t)||^2 = 1 - ||\widetilde{c_1}(t)||^2\,.$$

3

The adiabatic approximation calculation proceed the same as in Problem 2, and is implemented in the program via the use of Eq. (2.3), (2.5), and (2.6) in Problem 2. The numerical results are plotted in solid lines in the following figure, together with the adiabatic approximation plotted in dashed lines. Note here we have set $\hbar = 1$.



From the figure, we clearly see that, both $\rho_{12}(\tau)$ calculated via adiabatic approximation and calculated numerically approaches $1/2$ in the limit $\tau \to \infty$, as expected.

** The code for the program is attached on the next page.*

Script by Jinyan Miao for P1 and P3 on Physics 542 Homework 1

Script for P1

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

delta_t = 0.00005
ts = np.linspace(0,2,int(2/delta_t))

omega = 0
Omega0s = [0.5,1,2,10]
omega0 = 5

red_colors = [(1,0,0), (1,0,0), (1,0,0), (1,0,0)]
blue_colors = [(0,0,1), (0,0,1), (0,0,1), (0,0,1)]

def a1_sqamp_ana(t, omega0, a1i, a2i, Omega0):
    # squared amplitude of a1 analytic function
    omega0 = -omega0
    y = 2*Omega0
    X = ((omega0**2)+(y**2))**(1/2)
    a1 = (np.cos(X*t/2)+1j*omega0/X*np.sin(X*t/2))*a1i-(1j*y/X)*np.sin(X*t/2)*a2i
    return a1*np.conjugate(a1)

def a2_sqamp_ana(t, omega0, a1i, a2i, Omega0):
    # squared amplitude of a2 analytic function
    omega0 = -omega0
    y = 2*Omega0
    X = ((omega0**2)+(y**2))**(1/2)
    a2 = (np.cos(X*t/2)-1j*omega0/X*np.sin(X*t/2))*a2i-(1j*y/X)*np.sin(X*t/2)*a1i
    return a2*np.conjugate(a2)

# def a1_sqamp_ana(t, delta, a1i, a2i, Omega0):
#     #squared amplitude of a1 analytic function
#     theta = Omega0*np.sin(omega*t)/omega
#     a1 = np.cos(theta)*c1i - 1j*np.sin(theta)*c2i
#     return a1*np.conjugate(a1)
#
# def a2_sqamp_ana(t, delta, a1i, a2i, Omega0):
#     #squared amplitude of a1 analytic function
#     theta = Omega0*np.sin(omega*t)/omega
#     a2 = -1j*np.sin(theta)*c1i + np.cos(theta)*c2i
#     return a2*np.conjugate(a2)


# numerical computation
for Omega0 in Omega0s:
    delta = omega0-omega
    c1i = 1
    c2i = 0
    a1 = [c1i]
    a2 = [c2i]
    hbar = 1
    m = 1j*omega0*hbar/2
    c = -1j*Omega0

    for t in ts[:-1]:
```

```
        dot_a1 = m*a1[-1] + c*np.cos(omega*t)*a2[-1]
        dot_a2 = c*np.cos(omega*t)*a1[-1] - m*a2[-1]
        a1.append(a1[-1]+delta_t*dot_a1)
        a2.append(a2[-1]+delta_t*dot_a2)

    a1_sqamp = np.real(np.array([a*np.conjugate(a) for a in a1]))
    a2_sqamp = np.real(np.array([a*np.conjugate(a) for a in a2]))

    # general plots
    plt.plot(ts, a1_sqamp, color=red_colors[n], label='level 1 numerical')
    plt.plot(ts, a2_sqamp, color=blue_colors[n], label='level 2 numerical')
    plt.plot(ts, [a1_sqamp_ana(t, omega0, c1i, c2i, Omega0) for t in ts],
             label='level 1 analytic', linestyle='dashed', color='orange')
    plt.plot(ts, [a2_sqamp_ana(t, omega0, c1i, c2i, Omega0) for t in ts],
             label='level 2 analytic', linestyle='dashed', color='purple')
    plt.ylabel(r'amplitude', fontsize='xx-large')
    plt.xlabel("t/T",fontsize='xx-large')
    plt.legend(fontsize='xx-large')
    plt.title(r'Omega_0='+str(Omega0), fontsize='xx-large')
    plt.tight_layout()
    plt.show()
```

Script for P3

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

## discretize for numerical computation
delta_t = 0.00001
ts = np.linspace(-10,10,int(20/delta_t))
## discretize for analytic plot
delta_t_ana = 0.005
ts_ana = np.linspace(-10,10,int(20/delta_t_ana))

# initial conditions
c1i = 1
c2i = 0
c1 = [c1i]
c2 = [c2i]
# parameters
hbar = 1
Omega0Mod = 30
delta0 = 30

def Omega0Func(t):
    return Omega0Mod*np.exp(-(t**2))

def deltaFunc(t):
    if t<0:
        delta = delta0*((1-np.exp(t))**3)
    else:
        delta = 0
```

```
    return delta

def c1_dressed_ana(ts_truncated,ini0):
    ## dressed c1 analytic form
    Omegas = []
    for t in ts_truncated:
        Omega0 = Omega0Func(t)
        delta = deltaFunc(t)
        Omega = np.sqrt(delta**2 + np.absolute(Omega0)**2)
        Omegas.append(Omega)
    I = np.trapz(Omegas, x=ts_truncated)
    return np.exp(1j*I/2)*ini0

def c2_dressed_ana(ts_truncated,ini0):
    ## dressed c2 analytic form
    Omegas = []
    for t in ts_truncated:
        Omega0 = Omega0Func(t)
        delta = deltaFunc(t)
        Omega = np.sqrt(delta**2 + np.absolute(Omega0)**2)
        Omegas.append(Omega)
    I = np.trapz(Omegas, x=ts_truncated)
    return np.exp(-1j*I/2)*ini0

def c_tilde_ana(t, c1_dressed, c2_dressed):
    ## convert from dressed c to tilde c
    Omega0 = Omega0Func(t)
    delta = deltaFunc(t)
    Omega = np.sqrt(delta**2 + np.absolute(Omega0)**2)
    sin = np.sqrt((1/2)*(1-delta/Omega))
    cos = np.sqrt((1/2)*(1+delta/Omega))
    c1_tilde_ana = cos*c1_dressed+sin*c2_dressed
    c2_tilde_ana = -sin*c1_dressed+cos*c2_dressed
    return c1_tilde_ana, c2_tilde_ana


## numerical method computed as follows
for t in ts[:-1]:
    Omega0 = Omega0Func(t)
    delta = deltaFunc(t)
    c = -1j*np.conjugate(Omega0)*np.exp(-1j*delta*t)/2
    cc = -1j*Omega0*np.exp(1j*delta*t)/2
    dot_c1 = c*c2[-1]
    dot_c2 = cc*c1[-1]
    c1.append(c1[-1]+delta_t*dot_c1)
    c2.append(c2[-1]+delta_t*dot_c2)

c1 = np.array(c1)
c2 = np.array(c2)
rho12 = np.absolute(c1*np.conjugate(c2))

## analytic results computed as follows
# initial condition for dressed c
t_ini = ts_ana[0]
Omega0_ini = Omega0Func(t_ini)
delta_ini = deltaFunc(t_ini)
Omega_ini = np.sqrt(delta_ini**2 + np.absolute(Omega0_ini)**2)
sin_ini = np.sqrt((1/2)*(1-delta_ini/Omega_ini))
cos_ini = np.sqrt((1/2)*(1+delta_ini/Omega_ini))
c1_tilde_ini = np.exp(1j*delta_ini*t_ini/2)*c1i
```

```
c2_tilde_ini = -np.exp(1j*delta_ini*t_ini/2)*c2i
c1_dressed_ini = cos_ini*c1_tilde_ini-sin_ini*c2_tilde_ini
c2_dressed_ini = sin_ini*c1_tilde_ini+cos_ini*c2_tilde_ini

c1_tilde_anas = [c1_tilde_ini]
c2_tilde_anas = [c2_tilde_ini]

# compute the analytic values
for i in range(len(ts_ana)):
    t = ts_ana[i]
    c1_dressed = c1_dressed_ana(ts_ana[:i],c1_dressed_ini)
    c2_dressed = c2_dressed_ana(ts_ana[:i],c2_dressed_ini)
    c1_tilde, c2_tilde = c_tilde_ana(t, c1_dressed, c2_dressed)
    c1_tilde_anas.append(c1_tilde)
    c2_tilde_anas.append(c2_tilde)

# compute analytic magnitude
c1_tilde_mag = np.absolute(np.array(c1_tilde_anas))
c2_tilde_mag = np.absolute(np.array(c2_tilde_anas))
rho12_adbatic = np.absolute(c1_tilde_anas*np.conjugate(c2_tilde_anas))

## truncate t interval for plotting
length = len(ts)
L31 = int(length/3)
L32 = 2*int(length/3)

length = len(ts_ana)
L31ana = int(length/3)
L32ana = 2*int(length/3)

## plot the results
plt.plot(ts[L31:L32], rho12[L31:L32],
         label=r'$\rho_{12}(\tau)$', color='black')
plt.plot(ts_ana[L31ana:L32ana], rho12_adbatic[L31ana:L32ana],
         label=r'$\rho_{12}(\tau)$ adbatic', linestyle='--', color='gray'
    )
plt.plot(ts[L31:L32], np.absolute(c1)[L31:L32],
         label=r'$c_1(\tau)$', alpha=0.3, color='red')
plt.plot(ts[L31:L32], np.absolute(c2)[L31:L32],
         label=r'$c_2(\tau)$', alpha=0.3, color='blue')
plt.plot(ts_ana[L31ana:L32ana], c1_tilde_mag[L31ana:L32ana],
         label=r'$c_1(\tau)$ adbatic', linestyle='--',
         alpha=0.3, color='orange')
plt.plot(ts_ana[L31ana:L32ana], c2_tilde_mag[L31ana:L32ana],
         label=r'$c_2(\tau)$ adbatic', linestyle='--',
         alpha=0.3, color='purple')
plt.plot(ts[L31:L32], ([0.5]*len(ts))[L31:L32],
         linestyle='--', alpha=0.3, color='black')
plt.annotate('amp. = 0.5', (-3,0.45), alpha=0.3)
plt.ylabel(r'amplitude', fontsize='xx-large')
plt.xlabel("t/T",fontsize='xx-large')
plt.legend(fontsize='x-large')
plt.tight_layout()
plt.show()
```

4

Here we consider the field of the form

$$\mathbf{E}(t) = E_0\left(\hat{\mathbf{x}}\cos(\omega t) + \hat{\mathbf{y}}\sin(\omega t)\right)\,,$$

with $E_0 \in \mathbb{R}$ being a constant. We define

$$\epsilon_\pm = \mp\frac{\hat{\mathbf{x}} \pm i\hat{\mathbf{y}}}{\sqrt{2}}\,.$$

Thus we can write

$$\begin{aligned}
\frac{E_0}{\sqrt{2}}(-\hat{\epsilon}_+ e^{-i\omega t} + \hat{\epsilon}_- e^{i\omega t}) &= \frac{E_0}{\sqrt{2}}\left(\frac{\hat{\mathbf{x}} + i\hat{\mathbf{y}}}{\sqrt{2}}(\cos(\omega t) + i\sin(\omega t)) + \frac{\hat{\mathbf{x}} - i\hat{\mathbf{y}}}{\sqrt{2}}(\cos(\omega t) + i\sin(\omega t))\right) \\
&= \frac{E_0}{2}\left((\hat{\mathbf{x}} + i\hat{\mathbf{y}})(\cos(\omega t) - i\sin(\omega t)) + (\hat{\mathbf{x}} - i\hat{\mathbf{y}})(\cos(\omega t) + i\sin(\omega t))\right) \\
&= \frac{E_0}{2}\left((\hat{\mathbf{x}} + i\hat{\mathbf{y}})(\cos(\omega t) - i\sin(\omega t)) + (\hat{\mathbf{x}} - i\hat{\mathbf{y}})(\cos(\omega t) + i\sin(\omega t))\right) \\
&= \frac{E_0}{2}\left(2\cos(\omega t)\hat{\mathbf{x}} + 2\sin(\omega t)\hat{\mathbf{y}}\right) \\
&= E_0\left(\cos(\omega t)\hat{\mathbf{x}} + \sin(\omega t)\hat{\mathbf{y}}\right) \\
&= \mathbf{E}(t)\,.
\end{aligned}$$

Defining $\hat{\mathbb{I}} = |1\rangle\langle 1| + |2\rangle\langle 2|$. Now we can compute the interaction Hamiltonian

$$\hat{\mathbb{I}}\hat{H}_I\hat{\mathbb{I}} = -\frac{E_0}{\sqrt{2}}\left(|1\rangle\langle 1| + |2\rangle\langle 2|\right)\left(-\hat{\mu}\cdot\hat{\epsilon}_+ e^{-i\omega t} + \hat{\mu}\cdot\hat{\epsilon}_- e^{i\omega t}\right)\left(|1\rangle\langle 1| + |2\rangle\langle 2|\right)$$

First we consider the case with $|1\rangle$ representing the level $J = 0$, and $|2\rangle$ representing the level $J = 1$ with $m_J = 1$. Then by assumptions, with coefficients $-2E_0/(\sqrt{2}\hbar)$ absorbed by a constant $k \in \mathbb{C}$, we have

$$\hat{\mathbb{I}}\hat{H}_I\hat{\mathbb{I}} = \frac{\hbar}{2}\left(ke^{i\omega t}|1\rangle\langle 2| + k^* e^{-i\omega t}|2\rangle\langle 1|\right)$$

Thus the full Hamiltonian reads

$$\mathbf{H} = \frac{\hbar}{2}\begin{bmatrix} -\omega_0 & 0 \\ 0 & \omega_0 \end{bmatrix} + \frac{\hbar}{2}\begin{bmatrix} 0 & ke^{i\omega t} \\ k^* e^{-i\omega t} & 0 \end{bmatrix} = \frac{\hbar}{2}\begin{bmatrix} -\omega_0 & ke^{i\omega t} \\ k^* e^{-i\omega t} & \omega_0 \end{bmatrix}$$

Thus the Schrodinger's equation of the system reads

$$i\hbar\,\dot{\mathbf{a}}(t) = \frac{\hbar}{2}\begin{bmatrix} -\omega_0 & ke^{i\omega t} \\ k^* e^{-i\omega t} & \omega_0 \end{bmatrix}\mathbf{a}(t)\,. \tag{4.1}$$

In the interaction representation, $\mathbf{a}(t) = \mathbf{c}(t)\, \exp(-i\,\mathbf{E}\,t/\hbar)$, Eq. (4.1) becomes

$$i\hbar \begin{bmatrix} \dot{c}_1 e^{i\omega_0 t/2} + (i\omega_0 c_1/2) e^{i\omega_0 t/2} \\ \dot{c}_2 e^{-i\omega_0 t/2} - (i\omega_0 c_2/2) e^{-i\omega_0 t/2} \end{bmatrix} = \frac{\hbar}{2} \begin{bmatrix} -\omega_0 c_1 e^{i\omega_0 t/2} + c_2 e^{-i\omega_0 t/2} k e^{i\omega t} \\ c_1 e^{i\omega_0 t/2} k^* e^{i\omega t} + \omega_0 c_1 e^{i\omega_0 t/2} \end{bmatrix},$$

which is equivalent to the system

$$i\hbar\, \dot{\mathbf{c}}(t) = \frac{\hbar}{2} \begin{bmatrix} 0 & k e^{-i\delta t} \\ k^* e^{i\delta t} & 0 \end{bmatrix} \mathbf{c}(t)\,, \tag{4.2}$$

with the definition of detuning $\delta := \omega_0 - \omega$. Comparing Eq. (4.1) and (4.2) to the form of Eq. (2.62) and (2.63) from the textbook, we see that we have arrived to the desired system of equations without using the RWA.

On the other hand, if $|2\rangle$ represents the level $J = 1$ with $m_J = -1$ instead. Then by assumption, the interaction Hamiltonian gives

$$\hat{\mathbb{I}} \hat{H}_I \hat{\mathbb{I}} = \frac{\hbar}{2} \left(-k e^{-i\omega t} |1\rangle \langle 2| - k^* e^{i\omega t} |2\rangle \langle 1| \right) .$$

Similar argument leads to the system

$$i\hbar\, \dot{\mathbf{a}}(t) = \frac{\hbar}{2} \begin{bmatrix} -\omega_0 & -k e^{-i\omega t} \\ -k^* e^{i\omega t} & \omega_0 \end{bmatrix} \mathbf{a}(t)\,,$$

and in the interaction representation,

$$i\hbar\, \dot{\mathbf{c}}(t) = \frac{\hbar}{2} \begin{bmatrix} 0 & -k e^{-i(\omega+\omega_0)t} \\ -k^* e^{i(\omega+\omega_0)t} & 0 \end{bmatrix} \mathbf{c}(t)\,,$$

suggesting that the counterrotating term (with frequency $\omega + \omega_0$) drives the transitions between the two levels.