example-rabi-oscillations

June 25, 2014

1 QuTiP example: Vacuum Rabi oscillations in the Jaynes-Cummings model

J.R. Johansson and P.D. Nation

This ipython notebook demonstrates how to simulate the quantum vacuum rabi oscillations in the Jaynes-Cumming model, using QuTiP: The Quantum Toolbox in Python.

For more information about QuTiP see project web page: http://code.google.com/p/qutip/

In [1]:
```
# import the required python packages

%pylab inline

from qutip import *
```

```
Populating the interactive namespace from numpy and matplotlib
```

2 Introduction

The Jaynes-Cumming model is the simplest possible model of quantum mechanical light-matter interaction, describing a single two-level atom interacting with a single electromagnetic cavity mode. The Hamiltonian for this system is (in dipole interaction form)

$H = \hbar\omega_c a^\dagger a + \frac{1}{2}\hbar\omega_a\sigma_z + \hbar g(a^\dagger + a)(\sigma_- + \sigma_+)$

or with the rotating-wave approximation

$H_{\rm RWA} = \hbar\omega_c a^\dagger a + \frac{1}{2}\hbar\omega_a\sigma_z + \hbar g(a^\dagger\sigma_- + a\sigma_+)$

where ω_c and ω_a are the frequencies of the cavity and atom, respectively, and g is the interaction strength.

2.0.1 Problem parameters

Here we use units where $\hbar = 1$:

In [2]:
```
wc = 1.0  * 2 * pi  # cavity frequency
wa = 1.0  * 2 * pi  # atom frequency
g  = 0.05 * 2 * pi  # coupling strength
kappa = 0.005       # cavity dissipation rate
gamma = 0.05        # atom dissipation rate
N = 15              # number of cavity fock states
n_th_a = 0.0        # temperature in frequency units
use_rwa = True

tlist = linspace(0,25,100)
```

2.0.2 Setup the operators, the Hamiltonian and initial state

In [3]:
```
# intial state
psi0 = tensor(basis(N,0), basis(2,1))    # start with an excited atom

# operators
a  = tensor(destroy(N), qeye(2))
sm = tensor(qeye(N), destroy(2))

# Hamiltonian
if use_rwa:
    H = wc * a.dag() * a + wa * sm.dag() * sm + g * (a.dag() * sm + a * sm.dag())
else:
    H = wc * a.dag() * a + wa * sm.dag() * sm + g * (a.dag() + a) * (sm + sm.dag())
```

2.0.3 Create a list of collapse operators that describe the dissipation

In [4]:
```
c_op_list = []

rate = kappa * (1 + n_th_a)
if rate > 0.0:
    c_op_list.append(sqrt(rate) * a)

rate = kappa * n_th_a
if rate > 0.0:
    c_op_list.append(sqrt(rate) * a.dag())

rate = gamma
if rate > 0.0:
    c_op_list.append(sqrt(rate) * sm)
```

2.0.4 Evolve the system

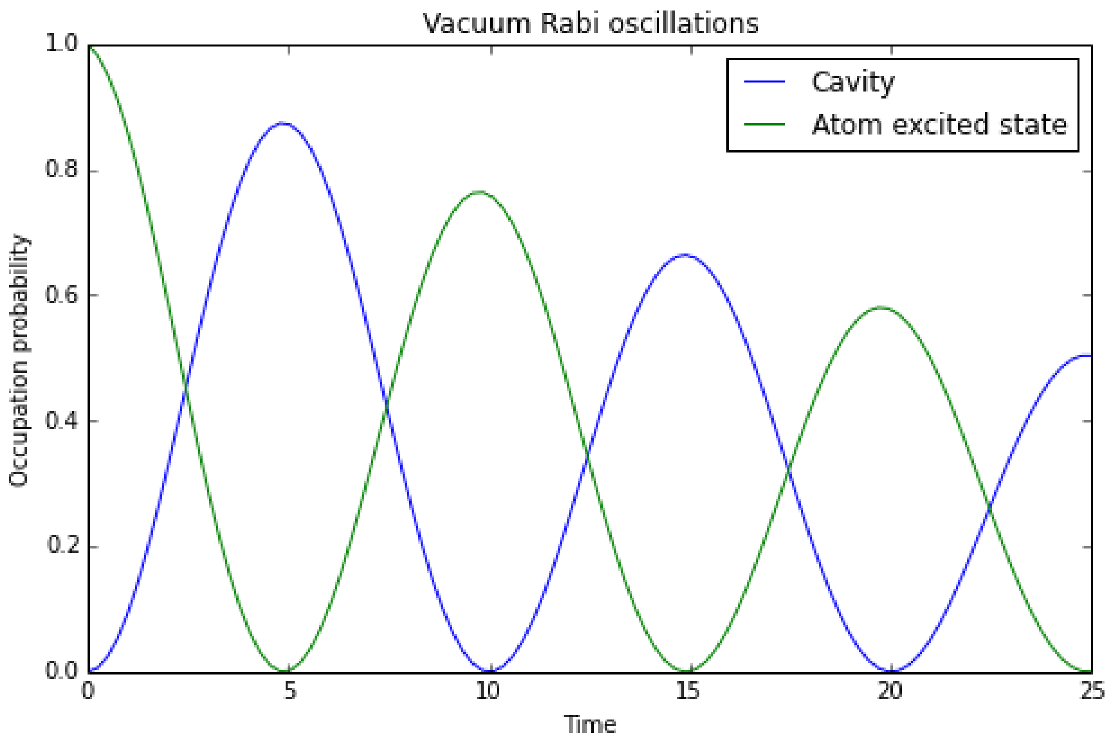
Here we evolve the system with the Lindblad master equation solver, and we request that the expectation values of the operators $a^\dagger a$ and $\sigma_+\sigma_-$ are returned by the solver by passing the list `[a.dag()*a, sm.dag()*sm]` as the fifth argument to the solver.

In [5]:
```
output = mesolve(H, psi0, tlist, c_op_list, [a.dag() * a, sm.dag() * sm])
```

2.1 Visualize the results

Here we plot the excitation probabilities of the cavity and the atom (these expectation values were calculated by the `mesolve` above). We can clearly see how energy is being coherently transferred back and forth between the cavity and the atom.

In [6]:
```
figure(figsize=(8,5))
plot(tlist, output.expect[0], label="Cavity")
plot(tlist, output.expect[1], label="Atom excited state")
legend()
xlabel('Time')
ylabel('Occupation probability')
title('Vacuum Rabi oscillations')
show()
```



2.1.1 Software version:

```
In [7]: from qutip.ipynbtools import version_table

        version_table()
```

```
Out[7]: <IPython.core.display.HTML at 0x7f2358c26ba8>
```