Quantum Optics with Python: Lecture2:Qutip and Quasi-probability theory

Cai Jiaqi

Huazhong University of Science and Technology

caidish@hust.edu.cn

October 26, 2017

Overview

What is Qutip?

QuTiP
Quantum Toolbox in Python

Qutip

Quantum Toolbox in Python.URL:http://qutip.org/
Cite them with Comp. Phys. Comm. 184, 1234 (2013) or Comp. Phys. Comm. 183, 1760 (2012)!

Why I choose Qutip?

- Free
- Easy
- Partially Fast
- Interactive
- Universal and Scalable

How Many People are Using Qutip?

Many people in the community of Quantum optics and related subjects are now using Qutip. Qutip helps to generate beautiful figure and simulation results of many high-quality papers.
In 2016,the Unique Visitors of qutip.org is larger than 25,473.

Windows

Qutip is first developed on Unix and tested mostly on Linux. But thanks to the portability of Python language, Windows users can also use Qutip as you like.

The recommended installation steps are:

- Install MSVC++ 2015 or VS2015.Notice: The default installation of VS2015 DON'T contain MSVC++ any more! You need to change the default to a manual one.
- Install Anaconda. For CERNET user, I highly recommend you to download the distribution from Tsinghua Opensource Mirror
- Pip install Qutip.
- enjoy it!

Linux and macOs

- Install gcc.
  Note: Ubuntu: sudo wget build-essentials. Mac: brew install gcc.
- Install Anaconda.
- pip install Qutip.
- enjoy it!

Hilbert Space

A computer can only hold a finite representation of Hilbert space. For analytic method, the Fock state can be represented as:

$$\langle m \mid n \rangle = \delta_{mn}, |n\rangle = A\left(a^{\dagger}\right)^{n} |0\rangle$$

However, this Fock state cannot be stored conveniently by a computer. In qutip, we often set an upper bound of Hilbert space, e.g. n . Then, states are $1 \times n$ vectors and operators are $n \times n$ matrix.
To generate many-body Hamiltonian, we should construct a total space from the tensor product(e.g,kronecker product) of two spaces:

$$H^{total} = H^{(1)} \otimes H^{(2)}$$

static function:basis(), create(), destroy(), tensor(),mesolve().....
Qobj class, properties: dims, shape, type, data
some selected function:*,dag(),eigenenergies(),eigenstates,

```python
d = 5
#the dimension of the Hilbert space
g = basis(d,0)
e1 = basis(d,1)
e2 = basis(d,2)
print(g)
print(e1)
print(e2)

a=destroy(d)
a_dag = create(d)
e1 = a_dag*g
e2 = (a_dag * a_dag * g).unit()
print(g)
print(e1)
print(e2)
```

Master Equation Solver

Unitary Evolution

$$\psi(t) = U(t,0)\,\psi(0)$$

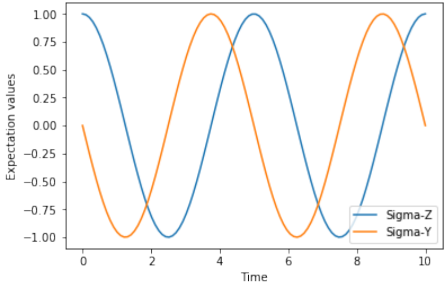
mesolve(H,psi0,times,[],[(exp)])

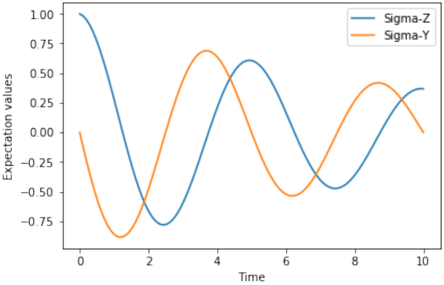
Non-unitary Evolution

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

mesolve(H,psi0,times,[c_ops_list],[(exp)])

The Hamiltonian of a spin in x-direction is read as: $H = \omega_z \sigma_x$. The dynamic evolution of this system can be obtained by qutip.





```python
H = 2 * np.pi * 0.1 * sigmax()
psi0 = basis(2, 0)
times = np.linspace(0.0, 10.0, 20)
result = mesolve(H, psi0, times, [], [sigmaz(),sigmay()])
fig, ax = plt.subplots()
ax.plot(result.times, result.expect[0]);
ax.plot(result.times, result.expect[1]);
ax.set_xlabel('Time');
ax.set_ylabel('Expectation values');
ax.legend(("Sigma-Z", "Sigma-Y"));
plt.show()
```

```
H = 2 * np.pi * 0.1 * sigmax()
psi0 = basis(2, 0)
times = np.linspace(0.0, 10.0, 20)
result = mesolve(H, psi0, times, [np.sqrt
    (0.05)*sigmax()], [sigmaz(),sigmay()])
fig, ax = plt.subplots()
ax.plot(result.times, result.expect[0]);
ax.plot(result.times, result.expect[1]);
ax.set_xlabel('Time');
ax.set_ylabel('Expectation values');
ax.legend(("Sigma-Z", "Sigma-Y"));
plt.show()
```

Mento Carlo Solver

$$H_{eff} = H_{sys} - \frac{i}{2}\sum_{i} C_n^{\dagger} C_n$$

$$\psi\left(t + dt\right) = C_n \psi\left(t\right) / \sqrt{\left\langle C_n^{\dagger} C_n \right\rangle}$$

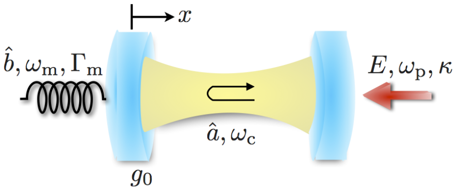
usage:mesolve(H,psi0,times,[c_ops_list],[(exp)])

Steady State Solver

$$\frac{d\rho_{ss}}{dt} = L\rho_{ss} = 0$$

usage:steadystates(H,c_ops)

Example:Optomechanical System



The optomechanical Hamiltonian arises from the radiation pressure interaction of light in an optical cavity where one of the cavity mirrors is mechanically compliant:

$$\frac{\hat{H}}{\hbar} = -\Delta\hat{a}^{+}\hat{a} + \omega_m\hat{b}^{+}\hat{b} + g_0(\hat{b}+\hat{b}^{+})\hat{a}^{+}\hat{a} + E\left(\hat{a}+\hat{a}^{+}\right)$$

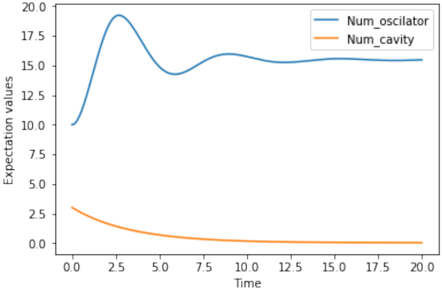
Where $\Delta$is the detuning between pump(ω_p) and cavity(ω_c), ω_m is frequency of the oscillator. g_0 is the single-photon-phonon coupling strength and E is the amplitude of the pump mode.

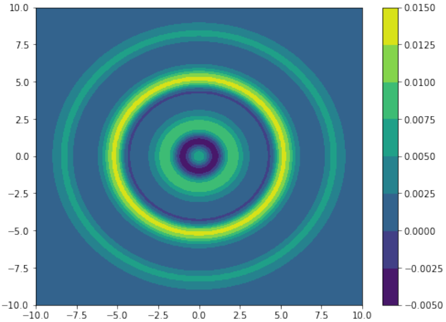
Example:Optomechanical System

To obtain the dynamic evolution of the system , we'll follow the steps below:

- Set system parameter
- Build Hamiltonian and collapse operators
- Solve master equation
- Run steady state Solver
- Visualize the result

Example:Optomechanical System





```python
# System Parameters (in units of wm)
#-----------------------------------
Nc = 10                      # Number of cavity states
Nm = 80                      # Number of mech states
kappa = 0.3                  # Cavity damping rate
E = 0.1                      # Driving Amplitude
g0 = 2.4*kappa               # Coupling strength
Qm = 1e4                     # Mech quality factor
gamma = 1/Qm                 # Mech damping rate
n_th = 1                     # Mech bath temperature
```

```
delta = -0.43                       # Detuning

# Operators
#------------
a = tensor(destroy(Nc), qeye(Nm))
b = tensor(qeye(Nc), destroy(Nm))
num_b = b.dag()*b
num_a = a.dag()*a
psi0=tensor(basis(Nc,5),basis(Nm,10))
ro0 = psi0*psi0.dag()
# Hamiltonian
#--------------
H = -delta*(num_a)+num_b+g0*(b.dag()+b)*num_a
    +E*(a.dag()+a)

# Collapse operators
#---------------------
cc = np.sqrt(kappa)*a
cm = np.sqrt(gamma*(1.0 + n_th))*b
```

```
cp = np.sqrt(gamma*n_th)*b.dag()
c_ops = [cc,cm,cp]

t_list = np.linspace(0,20,1000)
result = mesolve(H,ro0,times,c_ops,[num_b,
    num_a])
```

Motivation

Classical version of fluctuating of complex E(t)

$$\langle E^*(r_1,t)\,E(r_2,t)\rangle = \int E^*(r_1,t)\,E(r_2,t)\,P(E,E^*,t)\,d^2E$$

Q: Can we construct a similar description for quantum field fluctuations?
A:The quasi-probability theory in coherent state representation!

But: **???$a^\dagger$ and a are not commute???**

- Normal ordering $a^\dagger a$——*P-representation*
- Anti-normal ordering $aa^\dagger$——*Q-representation*
- Symmetric ordering $\left(aa^\dagger + a^\dagger a\right)/2$——*Wigner function*

P-representation: Theoretical Case

We want to expand any operator of the light field with probability distribution, Which means:

$$\left\langle O\left(a,a^{\dagger}\right)\right\rangle = \int d^2\alpha P\left(\alpha,\alpha^*\right) O\left(\alpha,\alpha^*\right)$$

Basically, any operator can be expanded in a normal ordering:

$$\delta\left(\alpha^* - a^{\dagger}\right)\delta\left(\alpha - a\right) = \frac{1}{\pi^2}\int d^2\beta\left[e^{-\beta\left(\alpha^* - a^{\dagger}\right)} e^{\beta(\alpha - a)}\right]$$

Then, take the expectation value of the operator:

$$\left\langle O\left(a,a^{\dagger}\right)\right\rangle = Tr\left[\rho O\right] = \sum_n \sum_m c_{nm} Tr\left[\rho\left(a^{\dagger}\right)^n a^m\right]$$

Define an operator:

$$\delta\left(\alpha^* - a^{\dagger}\right)\delta\left(\alpha - a\right) = \frac{1}{\pi^2}\int d^2\beta\left[e^{-\beta\left(\alpha^* - a^{\dagger}\right)} e^{\beta(\alpha - a)}\right]$$

P-representation: Theoretical Case

Then, the expectation can be written as:

$$\langle O\rangle = \int d^2\alpha \sum_n \sum_m c_{nm} Tr\left[\rho\delta\left(\alpha^* - a^\dagger\right)\delta\left(\alpha - a\right)\right]\left(\alpha^*\right)^n \alpha^m$$

So:

$$P\left(\alpha, \alpha^*\right) \equiv Tr\left[\rho\delta\left(\alpha^* - a^\dagger\right)\delta\left(\alpha - a\right)\right]$$

A convenient way to calculate P-function is take the Fourier inverse of anti-diagonal matrix element of ρ (exercise):

$$P\left(\alpha, \alpha^*\right) = \mathfrak{F}_{x_\beta, y_\beta \to x_\alpha, y_\alpha}\left[\langle -\beta|\,\rho\,|\beta\rangle\, e^{|\beta|^2}\right]$$

P-representation:Quantum Fock state

Last time, it is mentioned that Fock state is a quantum state of light. Here comes the reason:

$\rho = |n\rangle \langle n|$ is the density matrix of a Fock state, and

$$\langle -\beta| \rho |\beta\rangle = \exp\left(-|\beta|^2\right) \frac{(-1)^n |\beta|^{2n}}{n!}$$

It then follows that:

$$P(\alpha, \alpha^*) = \frac{e^{|\alpha|^2}}{\pi^2} \frac{(-1)^n}{n!} \int d^2\beta \left[|\beta|^{2n} e^{-\beta\alpha^* + \beta^*\alpha}\right]$$
$$= \frac{e^{|\alpha|^2}}{\pi^2} \frac{(-1)^n}{n!} \frac{\partial^{2n}}{\partial\alpha^n \partial\alpha^{*n}} \int d^2\beta \left[e^{-\beta\alpha^* + \beta^*\alpha}\right]$$
$$= \frac{e^{|\alpha|^2}}{\pi^2} \frac{(-1)^n}{n!} \frac{\partial^{2n}}{\partial\alpha^n \partial\alpha^{*n}} \delta^2(\alpha, \alpha^*)$$

Which turns out to be **negative**. In a classical version of distribution theory, probability is required to be nonnegative! These whose $P(\alpha, \alpha^*)$ is negative for some value are called non-classical.

Q-representation: Theoretical Case

The definition of Q-representation is:

$$Q\left(\alpha,\alpha^*\right)=Tr\left[\rho\delta\left(\alpha-a\right)\delta\left(\alpha^*-a^\dagger\right)\right]$$
$$Q\left(\alpha,\alpha^*\right)=\frac{1}{\pi}\left\langle\alpha\right|\rho\left|\alpha\right\rangle$$

It then follows that(exercise):

$$\left\langle O\left(a,a^\dagger\right)\right\rangle=\int Q\left(\alpha,\alpha^*\right)O\left(\alpha,\alpha^*\right)d^2\alpha$$

And $Q\left(\alpha,\alpha^*\right)$ is nonnegative and bounded(exercise):

$$0\leqslant Q\left(\alpha,\alpha^*\right)\leqslant\frac{1}{\pi}$$

Wigner Distribution: Theoretical Case

Note that $P\&Q$ can be written in terms of **characteristic functions**. So:

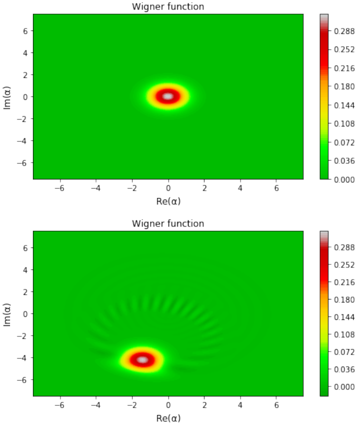
$$P(\alpha,\alpha^*) = \frac{1}{\pi^2}\int d^2\beta e^{-i\beta\alpha^* - i\beta^*\alpha} C^p(\beta,\beta^*)$$
$$C^p(\beta,\beta^*) = Tr\left[e^{i\beta a^\dagger} e^{i\beta^* a}\rho\right]$$
$$Q(\alpha,\alpha^*) = \frac{1}{\pi^2}\int d^2\beta e^{-i\beta\alpha^* - i\beta^*\alpha} C^Q(\beta,\beta^*)$$
$$C^Q(\beta,\beta^*) = Tr\left[e^{i\beta^* a} e^{i\beta a^\dagger}\rho\right]$$

Wigner and Weyl introduced Wigner function, which is defined as:

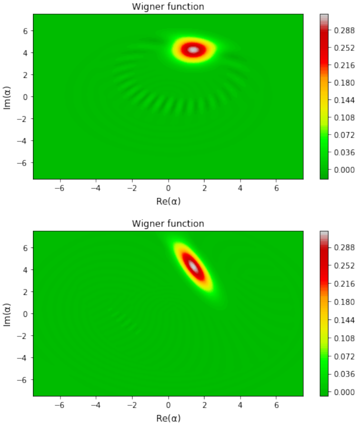
$$W(\alpha,\alpha^*) = \frac{1}{\pi^2}\int d^2\beta e^{-\beta\alpha^* - i\beta^*\alpha} C^W(\beta,\beta^*)$$
$$C^W(\beta,\beta^*) = Tr\left[e^{i\beta^* a + i\beta a^\dagger}\right]$$

Among three types of quasi-distribution, Wigner function is used mostly because it serves as a phase diagram of a quantum state.

Wigner Distribution:Vacuum state, Coherent state and squeezed state



(a) Vacuum state displacing to a cohrent state



(b) Cohrent state squezing to a squezeed state

```
N = 50
vac = basis(N,0)
rho_vac = vac*vac.dag()
rho_coh1 = coherent_dm(N,1+3j)
rho_coh2 = coherent_dm(N,-1-3j)
plot_wigner(rho_vac,colorbar=True,cmap='
    spectral')
plot_wigner(rho_coh1,colorbar=True,cmap='
    spectral')
plot_wigner(rho_coh2,colorbar=True,cmap='
    spectral')
vac = basis(N,0)
d = displace(N,1+3j)
s = squeeze(N,0.25+0.25j)
psi = s*d*vac
rho = psi*psi.dag()
plot_wigner(rho,colorbar=True,cmap='spectral'
    )
```

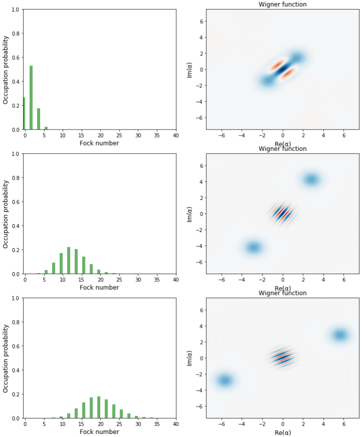
Cat State(Demo)

Consider a so-called Schrodinger-cat state

$$|\psi\rangle = |\alpha\rangle + |-\alpha\rangle$$

where $|\alpha\rangle$ is a coherent state. (a) Find the normalization constant N. (b) What is the photon distribution function and wigner function?

Cat State(Demo)



Kerr Effect to create Cat state(exercise)

The effective Hamiltonian will result in Cat state from an initial coherent state:

$$H = \frac{1}{2}\chi a^{\dagger}a^{\dagger}aa = \frac{1}{2}\chi n\left(n-1\right)$$

Test and verify this by numerical simulations.

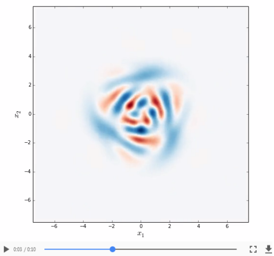


Figure: See:*Kirchmair, G., Vlastakis, B., Leghtas, Z., Nigg, S. E., Paik, H., Ginossar, E., Schoelkopf, R. J. (2012). Observation of quantum state collapse and revival due to the single-photon Kerr effect. Nature, 495(7440), 205209 https://doi.org/10.1038/nature11902* for details