laser_numeric_simulation_pop

May 11, 2017

```
In [1]: import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns

        from qutip import *
        import laser

        %matplotlib inline
        %reload_ext autoreload
        %autoreload 1
        %aimport laser
```

1 Equation of Motion for the Density Matrix of the Cavity Field

1.1 Introduction

For ρ_{nm}, we have

$$\begin{aligned}\dot{\rho}_{nm} = &-\frac{M_{nm}A}{1+N_{nm}B/A}\rho_{nm} + \frac{\sqrt{nm}A}{1+N_{n-1,m-1}B/A}\rho_{n-1,m-1}\\ &-\frac{C}{2}(n+m)\rho_{nm} + C\sqrt{(n+1)(m+1)}\rho_{n+1,m+1}\end{aligned}$$

where

$$\begin{aligned}A &= \frac{2r_a g^2}{\gamma^2},\\ B &= \frac{4g^2}{\gamma^2}A,\\ M_{nm} &= \frac{1}{2}(n+m+2) + (n-m)^2\frac{B}{8A},\\ N_{nm} &= \frac{1}{2}(n+m+2) + (n-m)^2\frac{B}{16A}.\end{aligned}$$

In particular, for diagonal elements, we have

$$\begin{aligned}\dot{p}(n) = &-\frac{(n+1)A}{1+(n+1)B/A}p(n) + \frac{nA}{1+nB/A}p(n-1)\\ &-Cnp(n) + C(n+1)p(n+1)\end{aligned}$$

1.2 Linear approximation ($B = 0 $)

In the steady state ($\dot{p}(n) = 0$), the equation of motion reduces to

$$-A(n+1)p(n) + Anp(n-1) - Cnp(n) + C(n+1)p(n+1) = 0$$

The detailed balance condition implies that

$$Anp(n-1) - Cnp(n) = 0$$
$$A(n+1)p(n) - C(n+1)p(n+1) = 0$$

The solution is clearly

$$p(n) = \left(1 - \frac{A}{C}\right)\left(\frac{A}{C}\right)^n, \quad A < C.$$

By defining an effective temperature T by

$$\exp\left(-\frac{\hbar\nu}{k_B T}\right) = \frac{A}{C},$$

we could obtain

$$p(n) = \left[1 - \exp\left(-\frac{\hbar\nu}{k_B T}\right)\right]\exp\left(-\frac{n\hbar\nu}{k_B T}\right).$$

So below the threshold, the steady-state solution is essentially that of a black-body cavity.

A numerical example

As $B = \frac{4g^2}{\gamma^2}A \to 0$, the interaction strenght must be much smaller than the atom decay rate, i.e. $g \ll \gamma$. Another condition should be satisfied is $A < C$. Let's choose the following parameters.

- Cavity frequency: $\omega_c = 2\pi$
- Atom frequency: $\omega_a = 2\pi$
- Coupling strength: $g = 0.0001 \times 2\pi$
- atom dissipation rate: $\gamma = 0.01$
- cavity dissipation rate: $C = 0.01$
- atom pump rate: $r_a = 0.50$
- average thermal photon: $n_{\text{th}} = 0$

```
In [40]: # parameters
         w_c = 2.0 * np.pi
         w_a = 2.0 * np.pi
         g = 0.0001 * 2 * np.pi

         gamma = 0.01
         kappa = 0.01
         ra = 0.9

         # initial cavity state
         N_max = 25
         n_list = np.arange(N_max)
         init_psi = fock(N_max, 0)
```

```
# list of time for ode
t_list = 0.2 * 2 * np.pi / w_c * np.arange(10000)
```

```
In [41]: laser_below = laser.LaserOneMode(w_c, w_a, g, ra, gamma, kappa)
```

```
In [42]: laser_below.get_atom_cavity_args()
```

```
Out[42]: {'g': 0.0006283185307179586,
          'gamma': 0.01,
          'kapa': 0.01,
          'ra': 0.9,
          'w_a': 6.283185307179586,
          'w_c': 6.283185307179586}
```

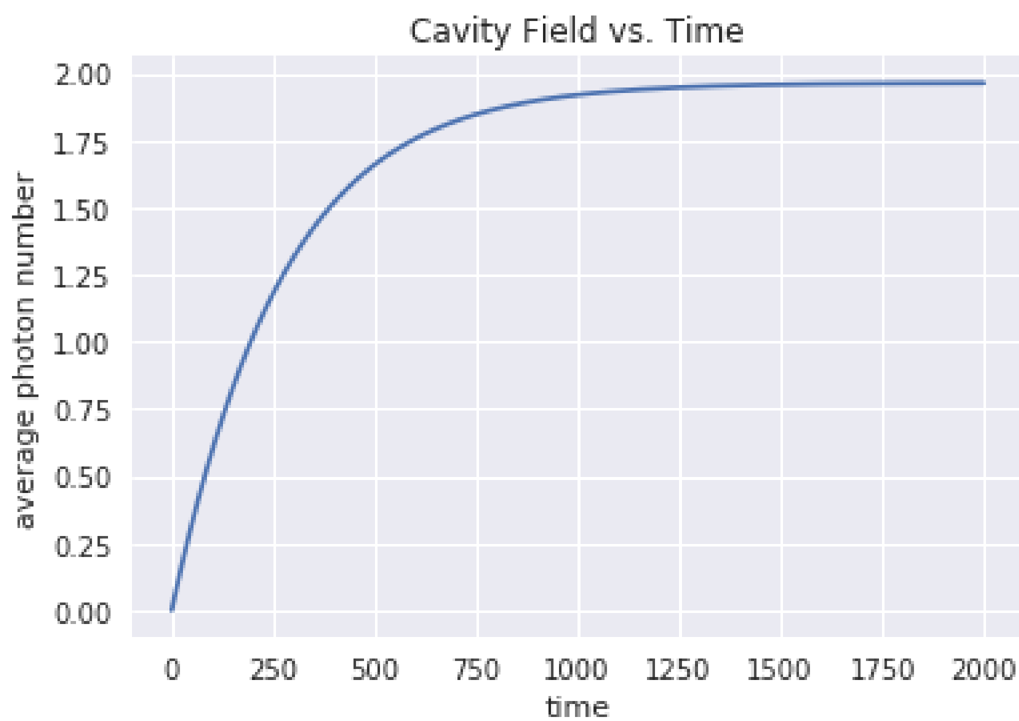
```
In [43]: laser_below.get_abc()
```

```
Out[43]: {'A': 0.007106115168784337, 'B': 0.00011221527287117076, 'C': 0.01}
```

```
In [44]: %time laser_below.pn_evolve(init_psi, N_max, t_list)
```

```
CPU times: user 3.45 s, sys: 4 ms, total: 3.46 s
Wall time: 3.45 s
```

```
In [50]: fig_below, ax_below = laser_below.plot_n_vs_time()
```



1.2.1 11.2.2 Far above threshold ($A \gg C$)

In the steady state ($\dot{p}(n) = 0$), the equation of motion reduces to

$$-\frac{A^2}{B}p(n) + \frac{A^2}{B}p(n-1) - Cnp(n) + C(n+1)p(n+1) = 0.$$

Detailed balance condition is

$$\frac{A^2}{B}p(n-1) - Cnp(n) = 0$$

The normalized solution of these euqaitons is

$$p(n) = e^{-\langle n \rangle}\frac{-\langle n \rangle^n}{n!},$$

with

$$\langle n \rangle = \frac{A^2}{BC}.$$

Thus the photon statistics of the lasre far above threshold are given by a Poisson distribution which is a characteristic of a coherent state.

1.2.2 i. a vacuum state as the initial state

The initial state is a vaccuum $\psi_0 = |0\rangle$.

As $A \gg C$. Let's choose the following parameters.

- Cavity frequency: $\omega_c = 2\pi$
- Atom frequency: $\omega_a = 2\pi$
- Coupling strength: $g = 0.01 \times 2\pi$
- atom dissipation rate: $\gamma = 0.01$
- cavity dissipation rate: $C = 0.01$
- atom pump rate: $r_a = 0.50$
- average thermal photon: $n_{\text{th}} = 0$

Set parameters

In [46]:
```
# parameters
w_c = 2.0 * np.pi
w_a = 2.0 * np.pi
g = 0.01 * 2 * np.pi

gamma = 0.01
kappa = 0.01
ra = 0.4

# initial cavity state
N_max = 80
n_list = np.arange(N_max)
init_psi = fock(N_max, 0)
```

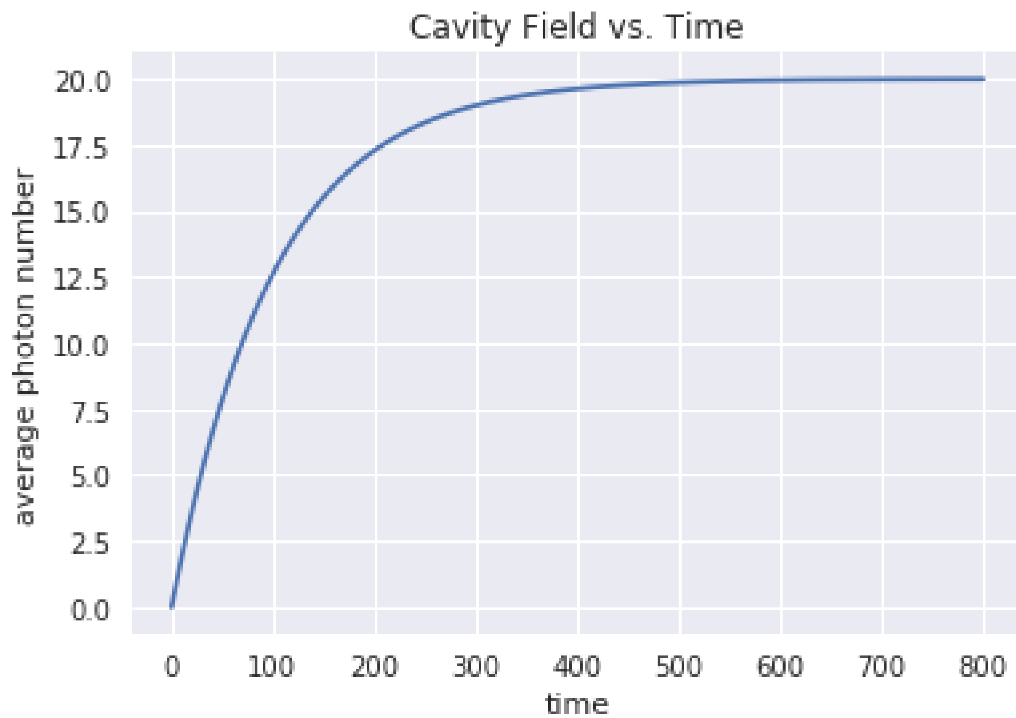
```
# list of time for ode
t_list = 0.02 * 2 * np.pi / w_c * np.arange(40000)
```

Create laser and solve the evolution equations

```
In [47]: laser_above = laser.LaserOneMode(w_c, w_a, g, ra, gamma, kappa)

In [48]: %time laser_above.pn_evolve(init_psi, N_max, t_list)

CPU times: user 16 s, sys: 128 ms, total: 16.2 s
Wall time: 15.2 s
```

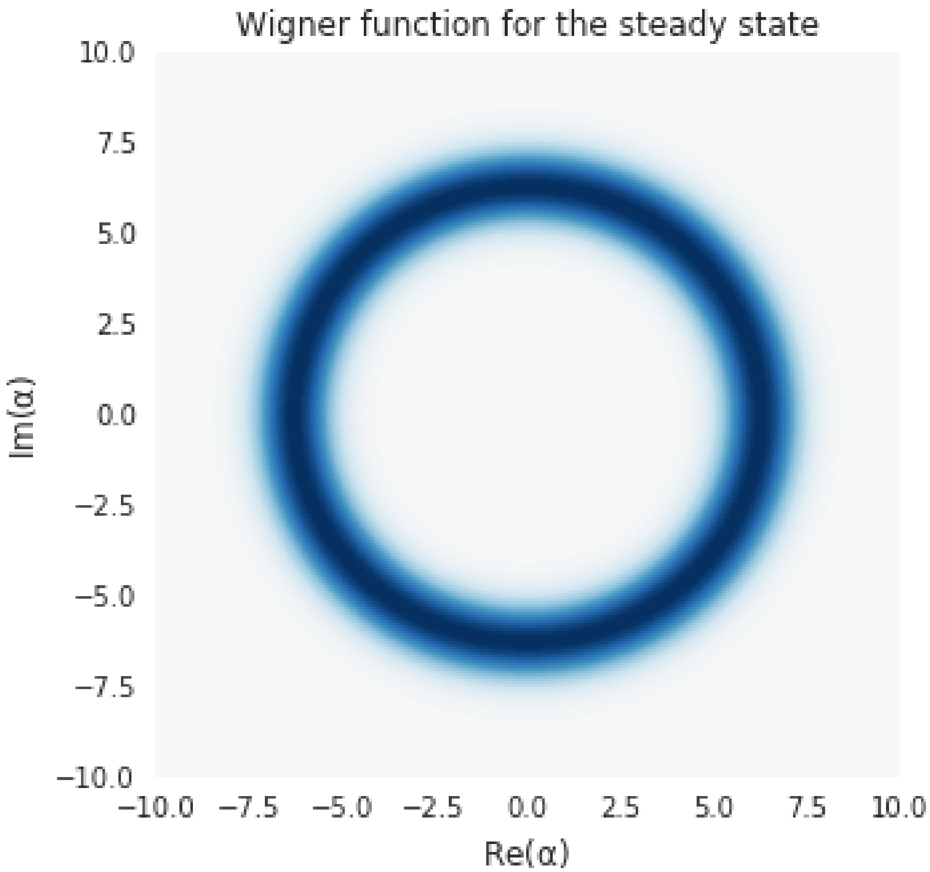
Plot average photon numbers vs. time

```
In [49]: fig_above, ax_above = laser_above.plot_n_vs_time();
```



Properties for the steady state

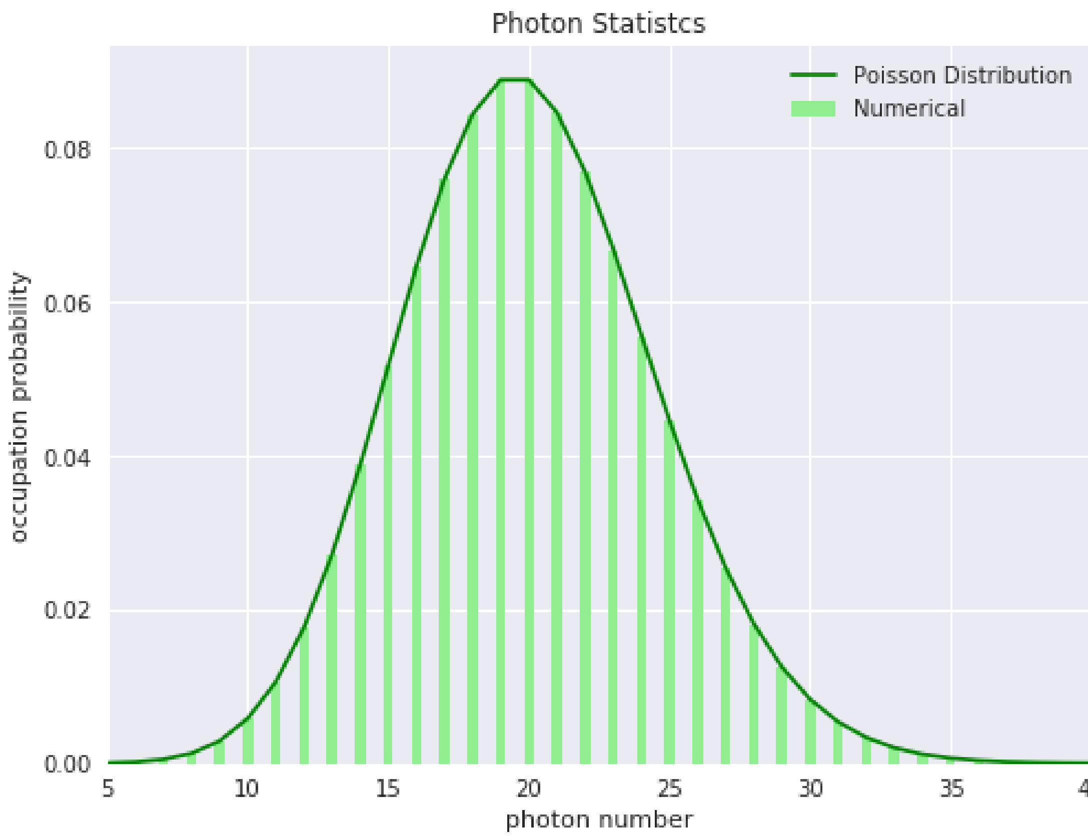
```
In [52]: l2_pn, l2_rho = laser_above.get_steady_state()

In [60]: plot_wigner(l2_rho, alpha_max=10, figsize=(5,5))
         plt.title("Wigner function for the steady state");
```



Compare numberical/analytical photon statistics for the steady states

```
In [57]: ne = laser_above.A**2/laser_above.kappa/laser_above.B
         pn_analytical = [np.exp(-ne) * ne**n / np.math.factorial(n) for n in n_lis
```

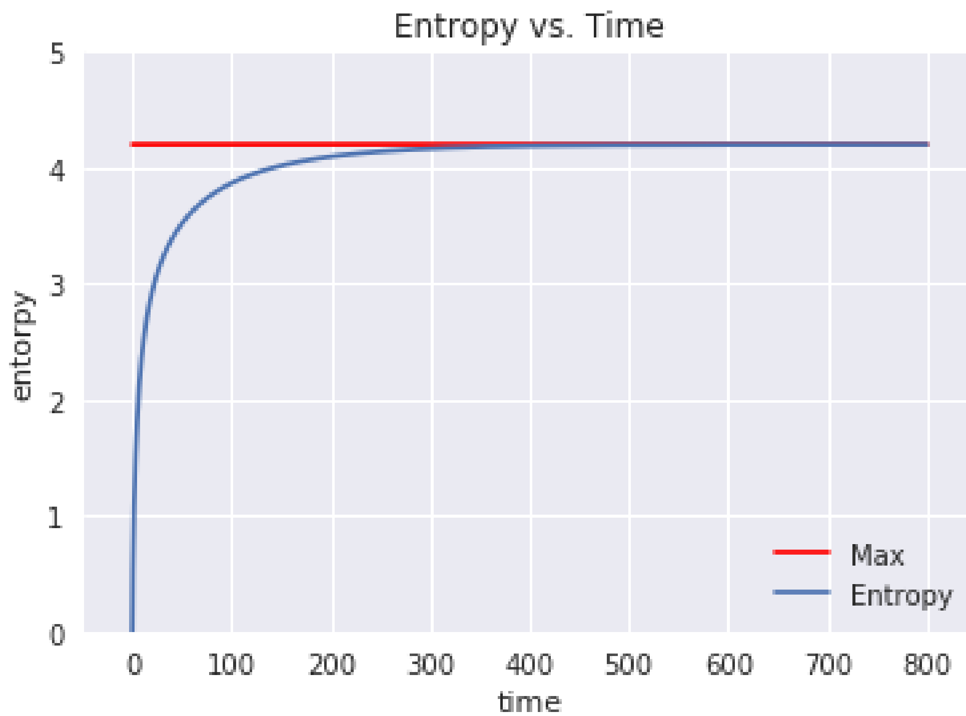
```
In [59]: fig, ax = plt.subplots(figsize=(8,6))
         ax.bar(n_list, l2_pn, color='lightgreen',
                width=0.35, align='center', label="Numerical")
         ax.plot(n_list, pn_analytical, color='green',
                 label="Poisson Distribution")
         plt.xlabel("photon number")
         plt.ylabel("occupation probability")
         plt.title("Photon Statistcs")
         plt.legend()
         plt.xlim(5, 40);
```



Plot entropy vs. time

```
In [61]: l2_rhos = laser_above.get_rhos()

In [62]: %time l2_entropy = np.array([entropy_vn(rho, 2) for rho in l2_rhos])

CPU times: user 1min 13s, sys: 1.12 s, total: 1min 14s
Wall time: 18.7 s

In [64]: entropy_max = - sum(pn_analytical * np.log2(pn_analytical))
         entropy_max

Out[64]: 4.2018873940012158

In [66]: plt.plot(t_list, [entropy_max] * len(t_list), color='red', label="Max")
         plt.plot(t_list, l2_entropy, label="Entropy")
         plt.ylim(0, 5)
         plt.xlim(-50, 850)
         plt.xlabel("time")
         plt.ylabel("entorpy")
         plt.title("Entropy vs. Time")
         plt.legend(loc=4)
```

```
Out[66]: <matplotlib.legend.Legend at 0x7fb5a2d5a710>
```



1.2.3 ii. a squeezed vacuum as the initial state

The initial state is a vaccuum $\psi_0 = S(\xi = 0.1)|0\rangle$.
As $A \gg C$. Let's choose the following parameters.

- Cavity frequency: $\omega_c = 2\pi$
- Atom frequency: $\omega_a = 2\pi$
- Coupling strength: $g = 0.01 \times 2\pi$
- atom dissipation rate: $\gamma = 0.01$
- cavity dissipation rate: $C = 0.01$
- atom pump rate: $r_a = 0.50$
- average thermal photon: $n_{\text{th}} = 0$

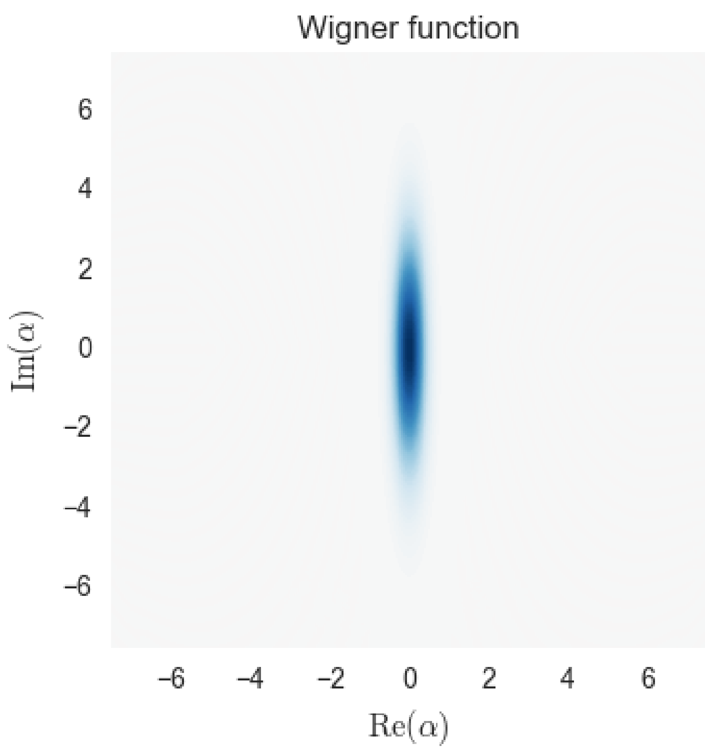
Set parameters

```
In [266]: # parameters
          w_c = 2.0 * np.pi
          w_a = 2.0 * np.pi
          g = 0.01 * 2 * np.pi

          gamma = 0.01
          kappa = 0.01
```

```
ra = 0.4

# initial cavity state
N_max = 80
n_list = np.arange(N_max)

s_op = squeeze(N_max, 1)
vac = fock(N_max, 0)
init_rho = ket2dm(s_op * vac)

# list of time for ode
t_list = 0.2 * 2 * np.pi / w_c * np.arange(4000)
```

Properties of the initial state

```
In [267]: plot_wigner(init_rho, figsize=(4,4), projection='2d');
```



Create the laser and solve the evolution equations

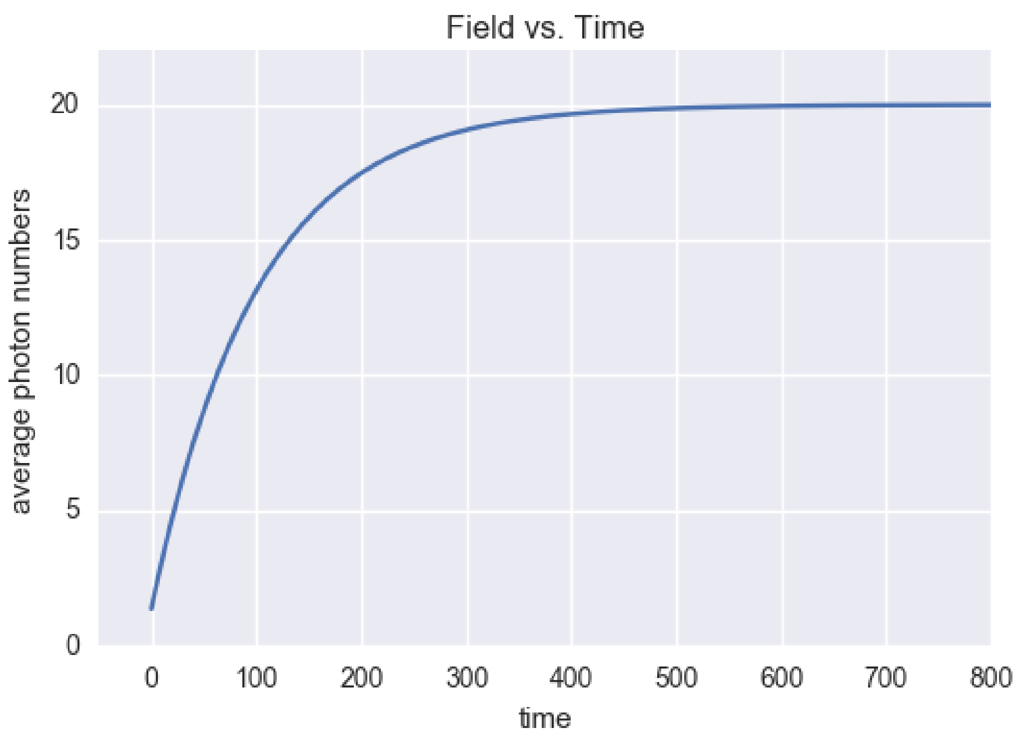
```
In [268]: laser2 = laser.LaserOneMode(w_c, w_a, g, ra, gamma, kappa)

In [269]: %time laser2.rho_evolve(init_rho, N_max, t_list)

Wall time: 14min 33s
```

Field vs. Time

```
In [270]: laser2_pns = laser2.get_pns()
          %time aver_n = [sum(pn * n_list) for pn in laser2_pns]
```

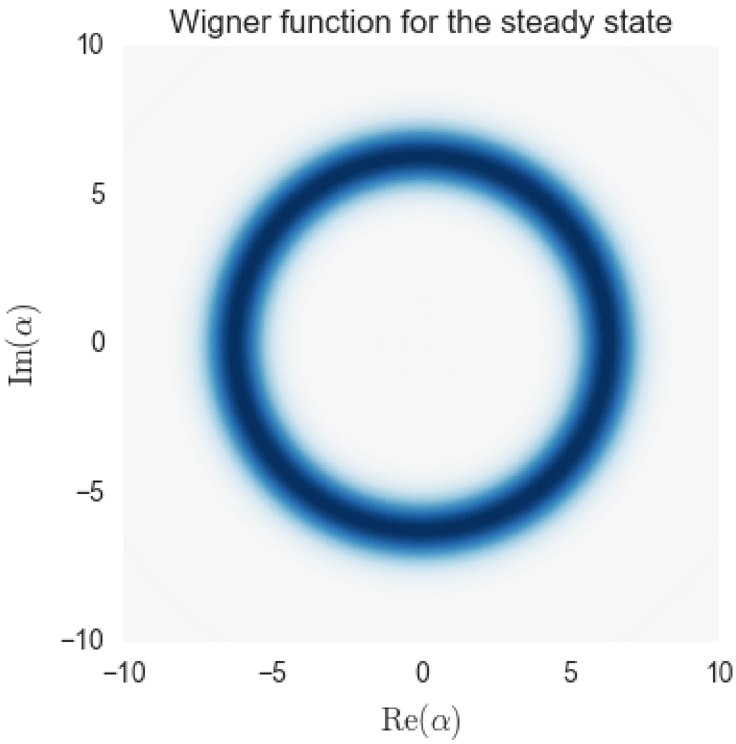
```
Wall time: 90 ms
```

```
In [272]: plt.figure(figsize=(6,4))
          plt.plot(t_list, aver_n)
          plt.xlim(-50, 800)
          plt.ylim(0, 22)
          plt.xlabel("time")
          plt.ylabel("average photon numbers")
          plt.title("Field vs. Time");
```



Properties for the steady state

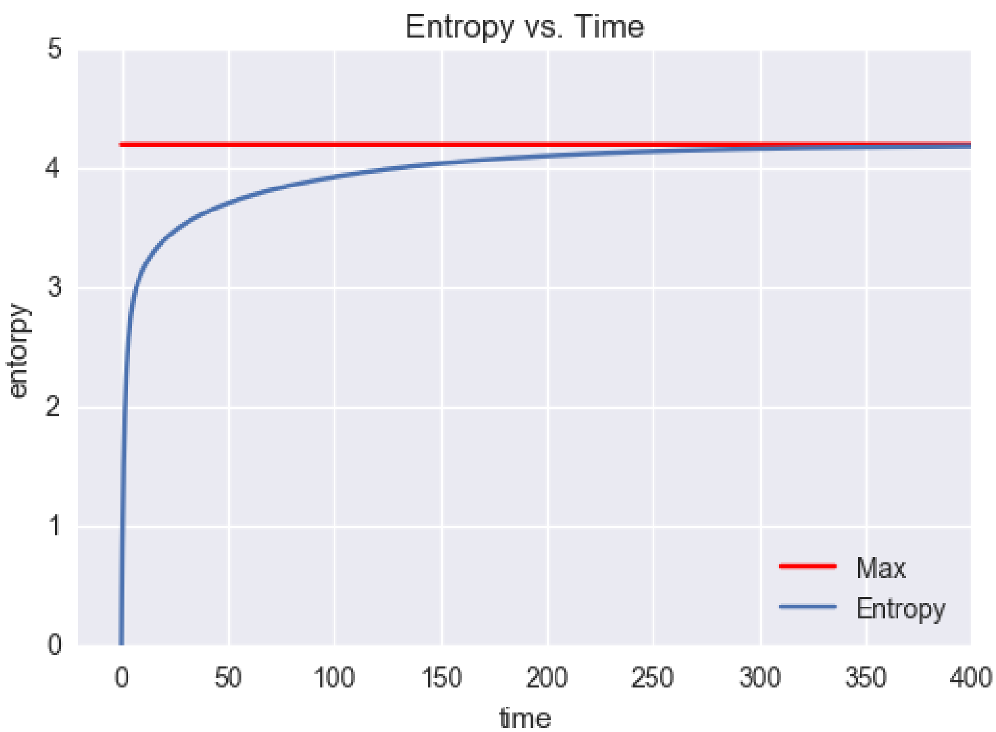
```
In [274]: laser2_s_pn, laser2_s_rho = laser2.get_steady_state()
          plot_wigner(laser2_s_rho, alpha_max=10, figsize=(4,4))
          plt.title("Wigner function for the steady state");
```



Plot entropy vs. time

```
In [275]: laser2_rhos = laser2.get_rhos()

          %time laser2_entropy = np.array([entropy_vn(rho, 2) for rho in laser2_rho

Wall time: 5.58 s
```

```
In [278]: plt.plot(t_list, [entropy_max] * len(t_list), color='red', label="Max")
          plt.plot(t_list, laser2_entropy, label="Entropy")
          plt.ylim(0, 5)
          plt.xlim(-20, 400)
          plt.xlabel("time")
          plt.ylabel("entorpy")
          plt.title("Entropy vs. Time")
          plt.legend(loc=4)

Out[278]: <matplotlib.legend.Legend at 0x280b7ef0>
```


Entropy vs. Time
entorpy
time
Max
Entropy
0
1
2
3
4
5
0
50
100
150
200
250
300
350
400