laser_threshold_v2

May 4, 2017

```
In [1]: __author__ = "Longfei Fan"
        __date__ = "05/03/2017"

In [2]: import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns

        from scipy.stats import poisson

        from qutip import *
        import laser

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

        # from matplotlib import rc
        # rc('xtick', labelsize=14)
        # rc('ytick', labelsize=14)
        # np.set_printoptions(threshold='nan', precision=6, suppress=True)

In [3]: from IPython.display import set_matplotlib_formats
        set_matplotlib_formats('png', 'pdf')
```

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

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

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

Parematers for Atom and Cavity

- interaction strength: $g = 0.001000$
- atom decay rate: $\gamma = 0.063246$
- cavity decay rate: $C = 0.000100$

```
In [5]: # parameters
        w_c = 2.0 * np.pi
        w_a = 2.0 * np.pi
        g = 0.001

        gamma = np.sqrt(4000) * g
        kappa = 0.0001
```

```
In [6]: gamma
```

```
Out[6]: 0.063245553203367583
```

Cavity starts from vacuum: $\psi_0 = |0\rangle$
Atom starts from the ground state

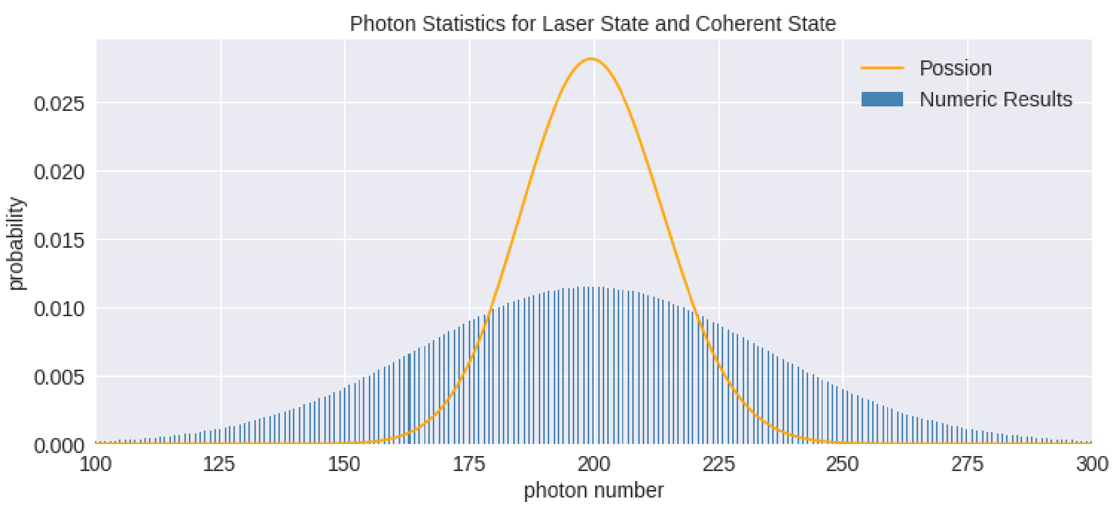
1.1 1. 20% above threshold

- A = 1.2 e-4
- B = 1.2 e-7
- C = 1.0 e-4

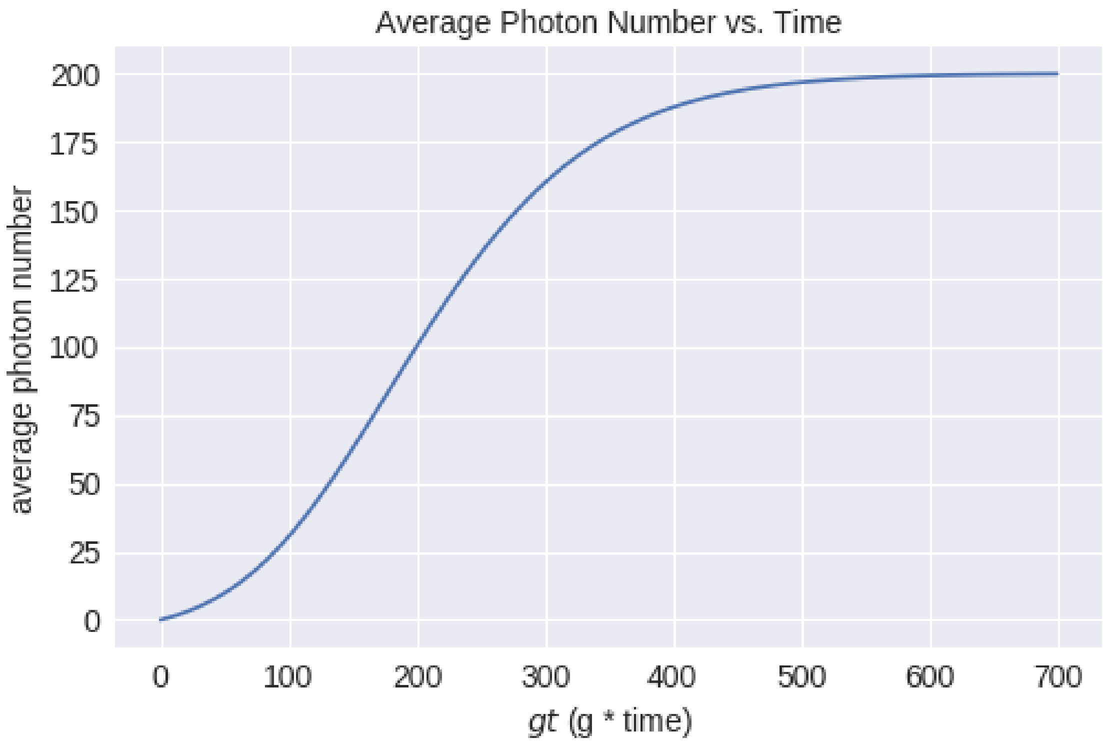
```
In [145]: # initial cavity state
          N_maxp = 1000
          n_listp = np.arange(N_maxp)
          init_psip = fock(N_maxp, 0)
```

```
In [148]: ra = 0.24 # pumping rate
          lp = laser.LaserOneMode(g, ra, gamma, kappa) # create laser object
```

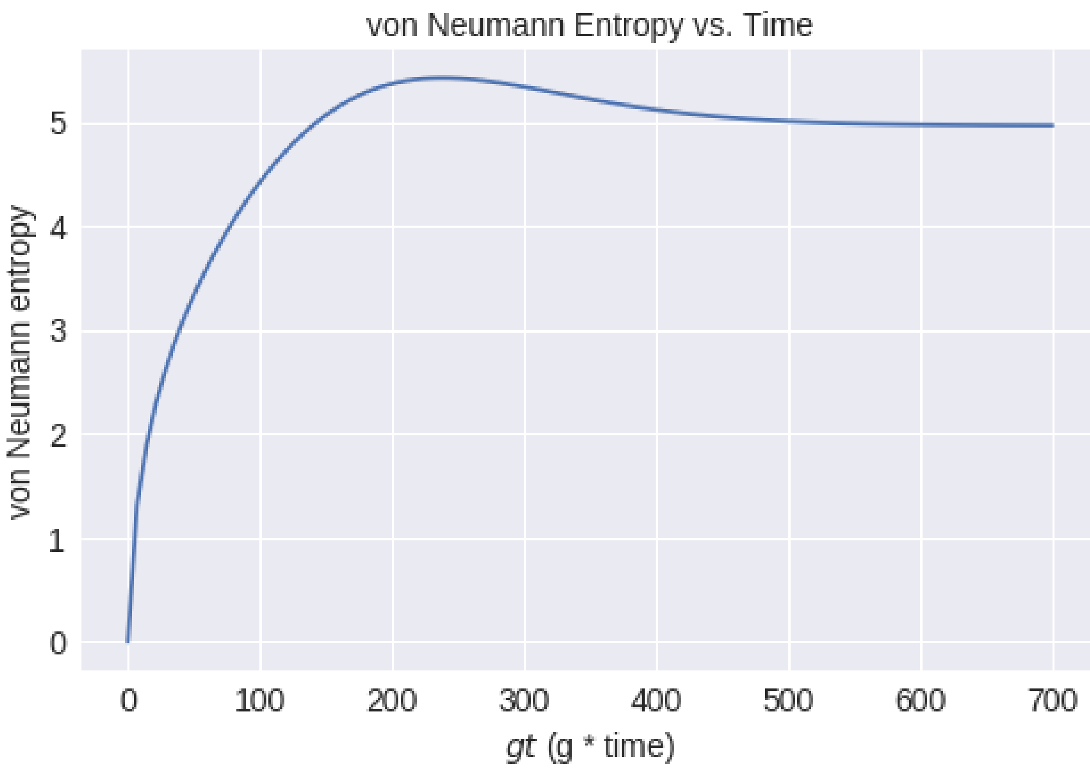
```
In [149]: lp.get_abc() # master equation parameters

Out[149]: {'A': 0.00012000000000000002, 'B': 1.2000000000000002e-07, 'C': 0.0001}

In [150]: # average photon number of the steady state
          lp_avern = lp.steady_n_above()
          lp_avern

Out[150]: 200.00000000000009

In [151]: # list of time for ode
          t_list = np.linspace(0, 700000, 1001)
          # state evolution
          lp.pn_evolve(init_psip, N_maxp, t_listp)

In [162]: # make a comparison between a laser state and a coherent state
          # with the same average photon numbers
          # operated above the threshold by 20%
          fig, ax = plt.subplots(figsize=(12, 5))
          ax.bar(n_listp, lp.get_pns()[-1], width=0.3,
                  color="steelblue", label="Numeric Results")
          ax.plot(n_listp, poisson.pmf(n_listp, lp_avern),
                  color="orange", label="Possion")
          ax.set_title("Photon Statistics for Laser State and Coherent State",
                       fontsize=14)
          ax.set_xlabel("photon number", fontsize=14)
          ax.set_ylabel("probability", fontsize=14)
          ax.legend(fontsize=14)
          ax.set_xlim(100, 300)
          ax.tick_params(axis='both', which='major', labelsize=14);
```



```
In [163]: # plot average photon numbers vs. time
          lp.plot_n_vs_time();
```




```
In [164]: # plot entroy vs. time
          lp.plot_entropy_vs_time();
```

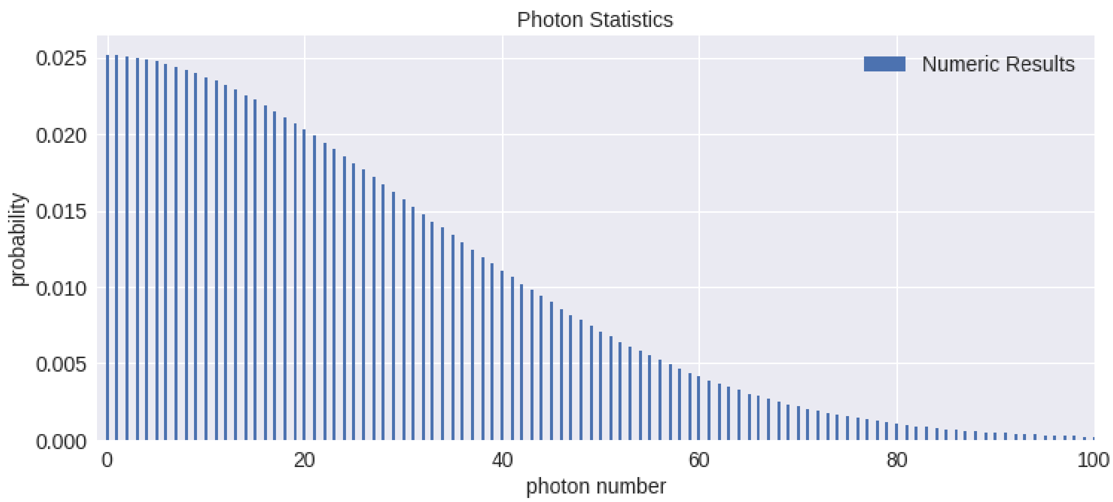


1.2 2. At the threshold

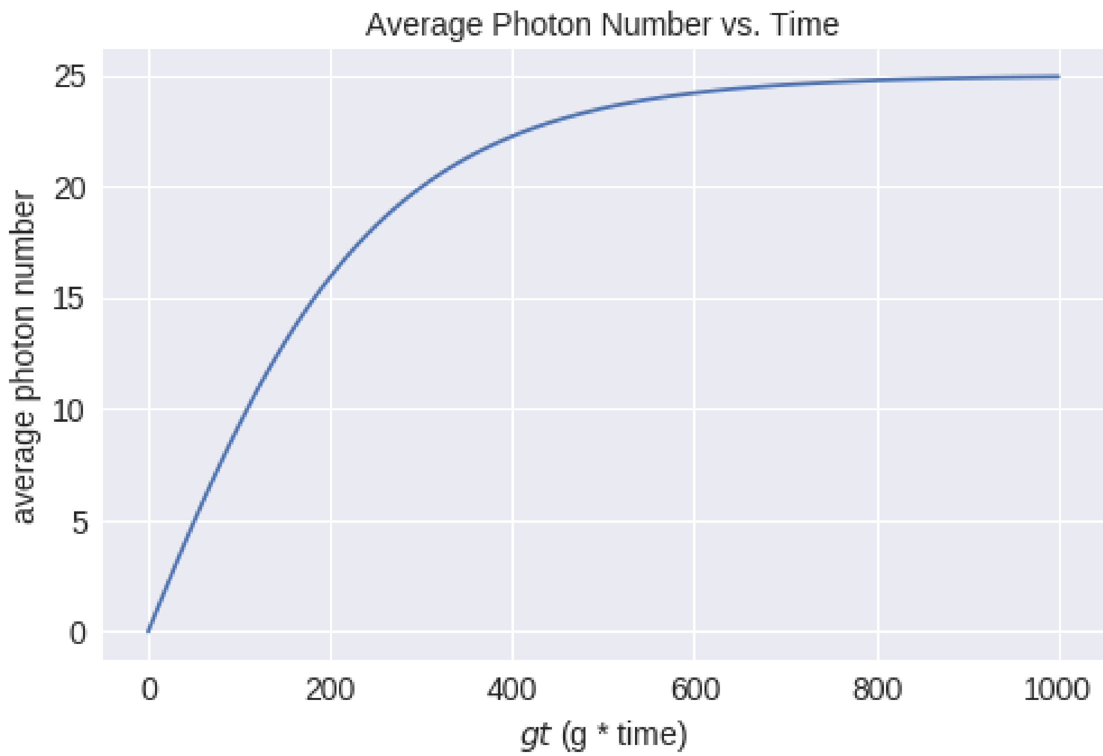
- A = 1.0 e-4
- B = 1.0 e-7
- C = 1.0 e-4

```
In [178]: # initial cavity state
          N_maxa = 150
          n_lista = np.arange(N_maxa)
          init_psia = fock(N_maxa, 0)

In [179]: ra = 0.20
          la = laser.LaserOneMode(g, ra, gamma, kappa)

In [180]: la.get_abc()

Out[180]: {'A': 0.00010000000000000002, 'B': 1.0000000000000004e-07, 'C': 0.0001}

In [181]: # list of time for ode
          t_lista = np.linspace(0, 1000000, 1001)
          # evolution
          la.pn_evolve(init_psia, N_maxa, t_lista)
```

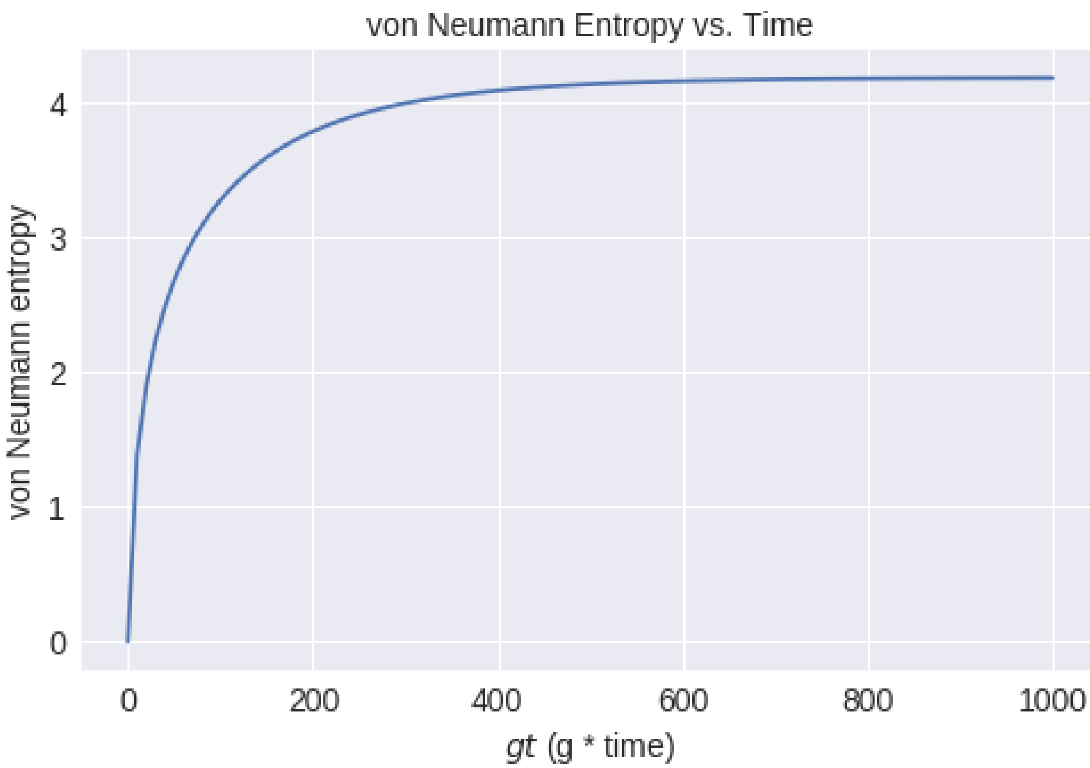
In [184]:
```
fig, ax = plt.subplots(figsize=(12, 5))

w = 0.35
ax.bar(np.arange(N_maxa), la.get_pns()[-1], width=0.3, label="Numeric Results")
# ax.bar(np.arange(N_max) + w, laser.boltzmann(0.8, N_max), width=w,
#         label="Boltzmann Distribution")
ax.legend(fontsize=14)
ax.set_xlim(-1, 100)
ax.set_xlabel("photon number", fontsize=14)
ax.set_ylabel("probability", fontsize=14)
ax.set_title("Photon Statistics", fontsize=14)
ax.tick_params(axis='both', which='major', labelsize=14);
```



In [185]:
```
la.plot_n_vs_time();
```




```
In [186]: la.plot_entropy_vs_time();
```



1.3 3. 20% below threshold

- A = 0.8 e-4
- B = 0.8 e-7
- C = 1.0 e-4

In [187]:
```
# initial cavity state
N_maxm = 40
n_listm = np.arange(N_maxm)
init_psim = fock(N_maxm, 0)
```

In [189]:
```
ra = 0.16
lm = laser.LaserOneMode(g, ra, gamma, kappa)
```

In [190]:
```
lm.get_abc()
```

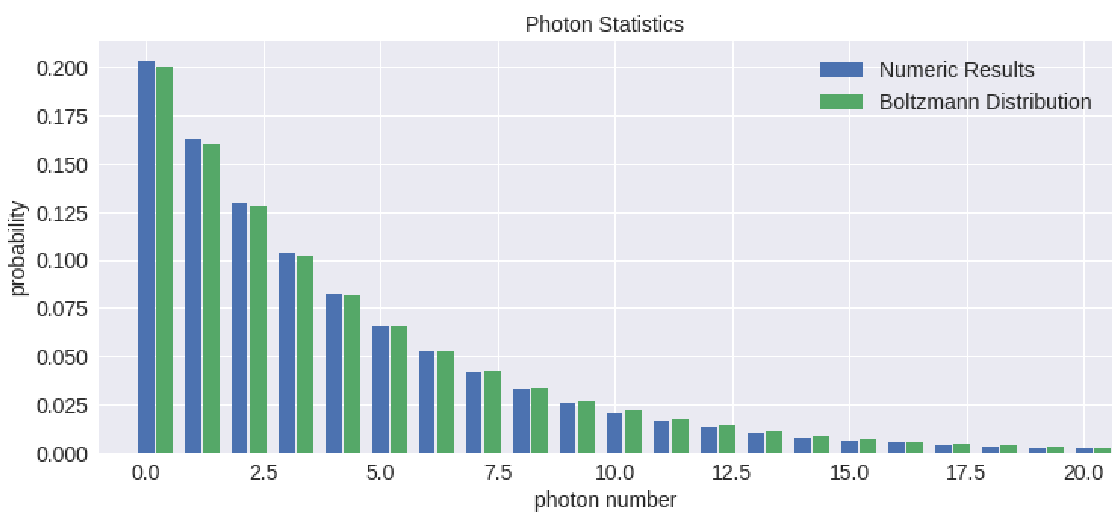
Out[190]:
```
{'A': 8.000000000000002e-05, 'B': 8.0000000000000041e-08, 'C': 0.0001}
```

The average photon number is estimated by

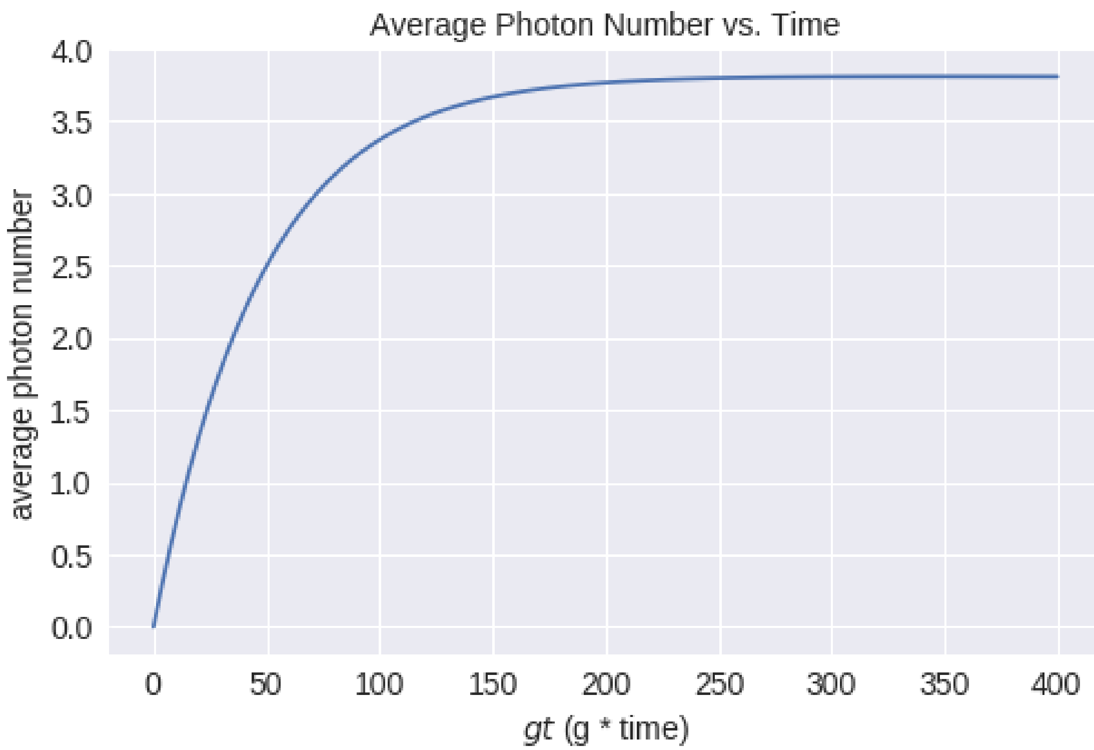
$$\frac{1}{e^{\hbar\nu/kT}-1} = \frac{1}{C/A-1} = 4$$

In [192]:
```
# list of time for ode
t_listm = np.linspace(0, 400000, 1001)
# evolution
lm.pn_evolve(init_psim, N_maxm, t_listm)
```

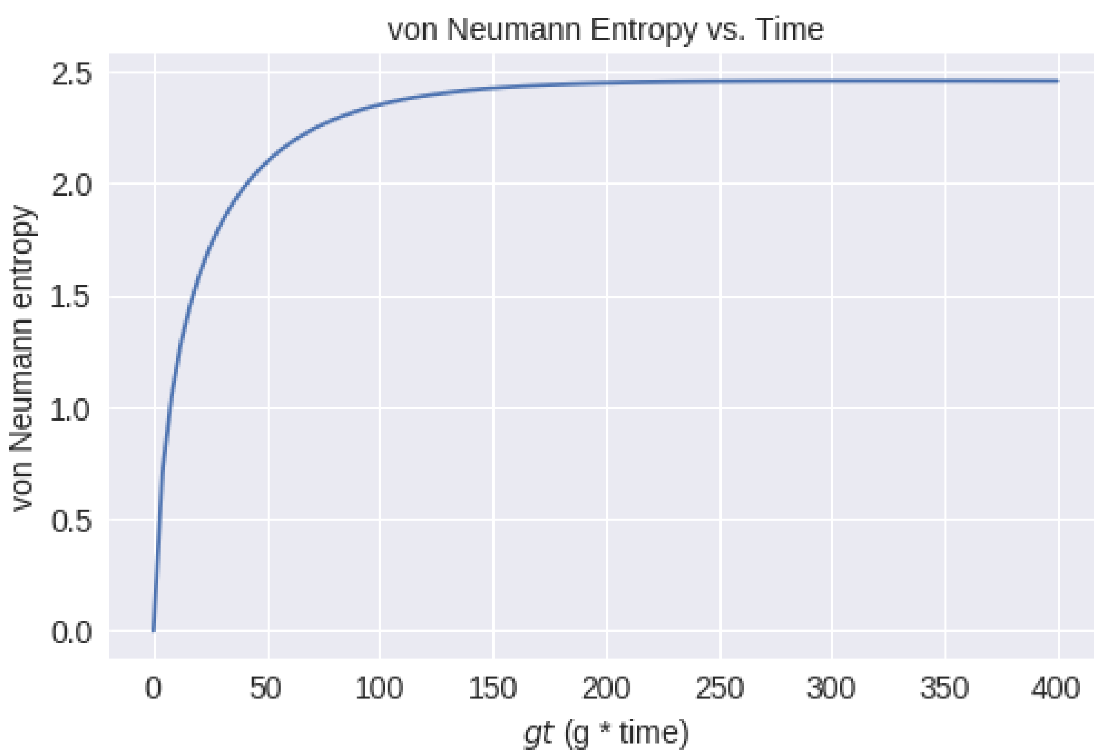
In [193]:
```
fig, ax = plt.subplots(figsize=(12, 5))

w = 0.35
ax.bar(np.arange(N_maxm), lm.get_pns()[-1], width=w,
       label="Numeric Results")
ax.bar(np.arange(N_maxm) + w + 0.05, laser.boltzmann(0.8, N_maxm), width=w,
       label="Boltzmann Distribution")
ax.legend(fontsize=14)
ax.set_xlim(-1, 20.6)
ax.set_xlabel("photon number", fontsize=14)
ax.set_ylabel("probability", fontsize=14)
ax.set_title("Photon Statistics", fontsize=14)
ax.tick_params(axis='both', which='major', labelsize=14);
```



In [194]:
```
lm.plot_n_vs_time();
```




```
In [195]: lm.plot_entropy_vs_time();
```

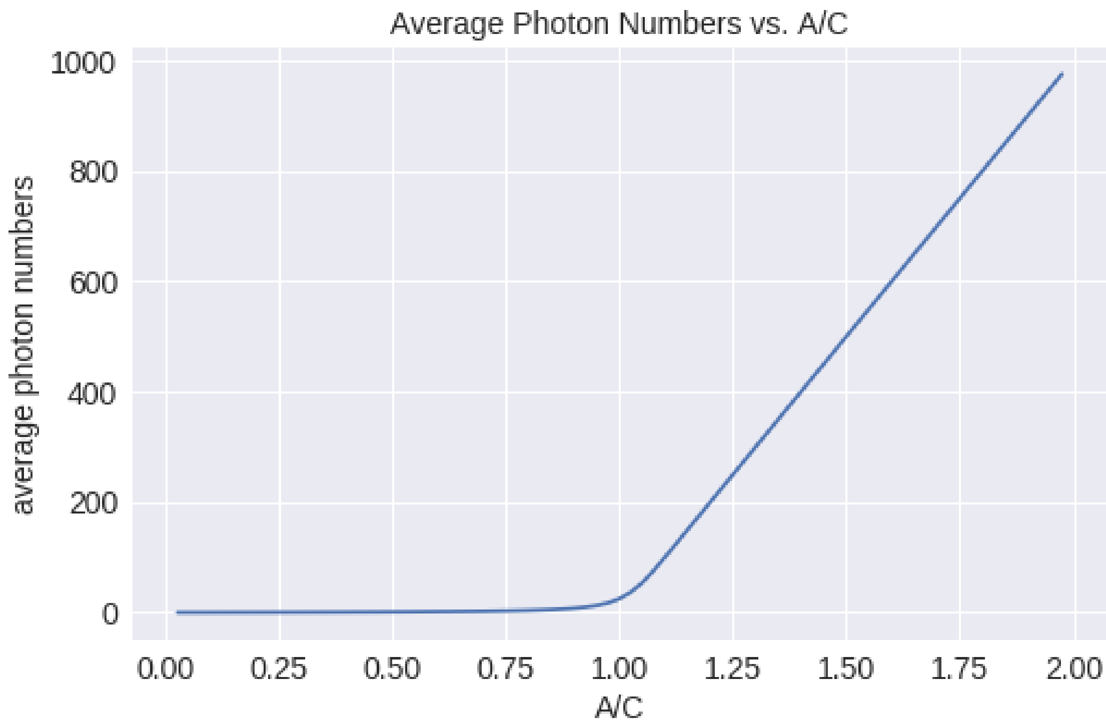


1.4 4. Steady values around the threshold

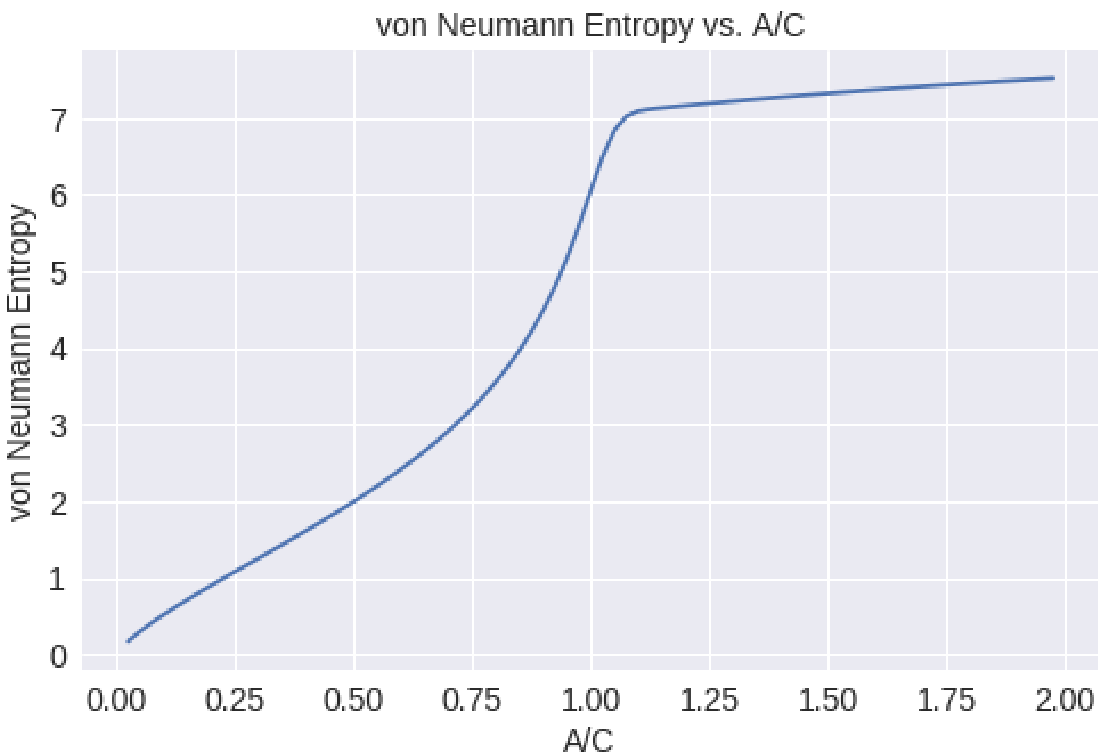
```
In [196]: # parameters
          g = 0.001
          gamma = np.sqrt(4000) * g
          kappa = 0.0001

          # initial cavity state
          N_max = 2000
          n_list = np.arange(N_max)
          init_psi = fock(N_max, 0)
```

```
In [213]: ratios, ns, entrs, g2s = [], [], [], []
          a = destroy(N_max)

          for ra in np.arange(0.005, 0.4, 0.005):
              l = laser.LaserOneMode(g, ra, gamma, kappa) # create laser object
              l.set_N_max(N_max)

              pn, n, entr = l.solve_steady_state()
              rho = Qobj(np.diag(pn))
              g2 = expect(a.dag() * a.dag() * a * a, rho) / n**2

              ratios.append(2 * ra * g**2 / gamma**2 / kappa)
              ns.append(n)
              entrs.append(entr)
              g2s.append(g2)
```

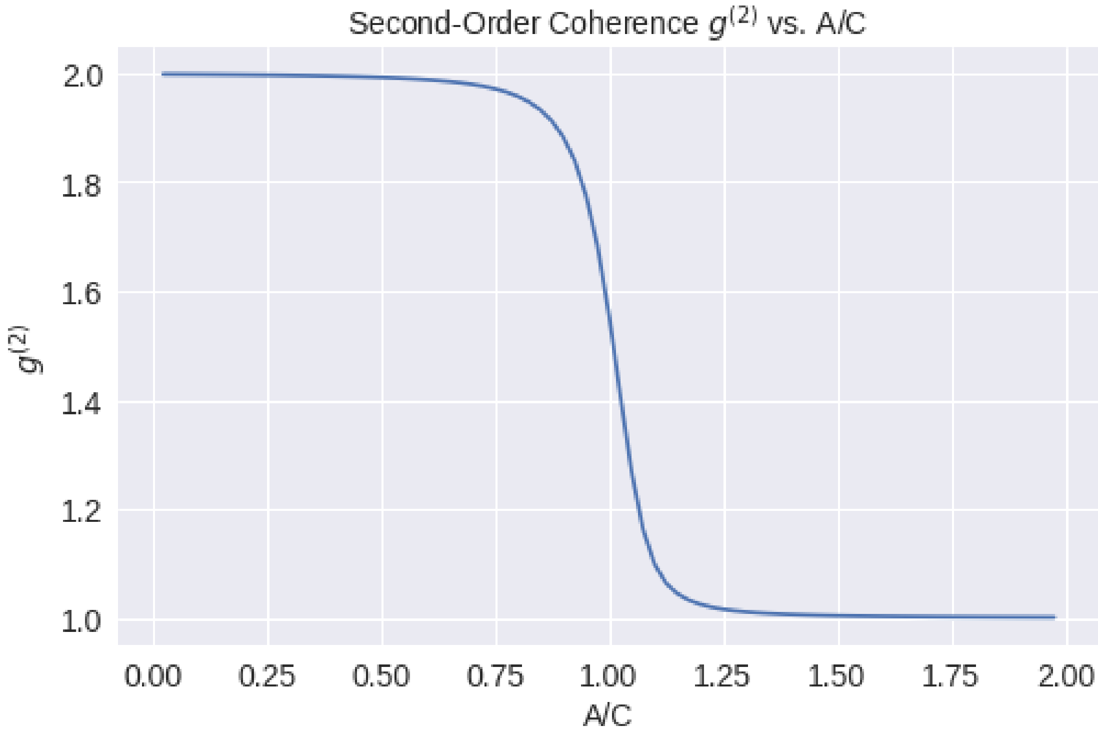
```
In [214]: fig, ax = plt.subplots(figsize=(8, 5))
          ax.plot(ratios, ns)
          ax.set_title("Average Photon Numbers vs. A/C", fontsize=14)
          ax.set_xlabel("A/C", fontsize=14)
          ax.set_ylabel("average photon numbers", fontsize=14)
          ax.tick_params(axis='both', which='major', labelsize=14);
```



In [215]:
```
fig, ax = plt.subplots(figsize=(8, 5))
ax.plot(ratios, entrs)
ax.set_title("von Neumann Entropy vs. A/C", fontsize=14)
ax.set_xlabel("A/C", fontsize=14)
ax.set_ylabel("von Neumann Entropy", fontsize=14)
ax.tick_params(axis='both', which='major', labelsize=14);
```



In [216]:
```
fig, ax = plt.subplots(figsize=(8, 5))
ax.plot(ratios, g2s)
ax.set_title("Second-Order Coherence $g^{(2)}$ vs. A/C", fontsize=14)
ax.set_xlabel("A/C", fontsize=14)
ax.set_ylabel("$g^{(2)}$", fontsize=14)
ax.tick_params(axis='both', which='major', labelsize=14);
```



Second-Order Coherence $g^{(2)}$ vs. A/C
$g^{(2)}$
2.0
1.8
1.6
1.4
1.2
1.0
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
A/C