plot

June 7, 2017

Entropy of Laser with Fixed Average Photon Numbers

- author: Longfei Fan
- created: 05/24/2017
- modified: 05/30/2017

In [32]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns

from scipy.stats import poisson

from qutip import *
import laser, entropy_utils

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

In [2]:
```
from IPython.display import set_matplotlib_formats
set_matplotlib_formats('pdf', 'png')
```

helpers

In [3]:
```
def plot_n_vs_t(filename, xlim, ylim):
    n1_df = pd.read_csv(filename)
    entropy_utils.df_plot(n1_df, xlim=xlim, ylim=ylim, \
                          style = ['-', '-.', ':', '--'], \
                          xlabel=r'$gt$', ylabel=r'$\bar{n}$ (mean photon number)')
    plt.title(r'Evolution of $\bar{n}$ under Different $A/C$', fontsize=14);


def plot_entr_vs_t(filename, xlim, ylim):
    entr1_df = pd.read_csv(filename)
    entropy_utils.df_plot(entr1_df, xlim=xlim, ylim=ylim, \
                          xlabel=r'$gt$', ylabel=r'$S$ (entropy)', \
```

```
                              style = ['-', '-.', ':', '--'], \
                              entr_cohe=ENTR_COHE, entr_thml=False)
    plt.title(r'Evolution of $S$ under Different $A/C$', fontsize=14)
```

In [4]:
```
G = 0.001
KAPPA = 0.0001

NBAR = 200
N_max = 1000
n_list = np.arange(N_max)


# vacuum
vacu = fock(N_max, 0)

# squeezed vacuum
# s = 1
# s_op = squeeze(N_max, s)
# svac = s_op * vacu

# thermal state
n_thml = 20
thml = thermal_dm(N_max, n_thml)


init_psi = vacu
solver = 'pn'
```

In [8]:
```
# fig, axes = plt.subplots(2, 1, figsize=(8, 8), sharex=True)

# plot_fock_distribution(thml, ax=axes[0], unit_y_range=False)
# axes[0].set_xlim(0, 200)
# axes[0].set_title("Thermal State n = 20")

# plot_fock_distribution(vacu, ax=axes[1], unit_y_range=False)
# axes[1].set_xlim(0, 200)
# axes[1].set_title("Squeezed Vacuum r = 2");
```

In [9]:
```
# entropy_vn(vacu), entropy_vn(svac), entropy_vn(thml)
```

In [10]:
```
# nn = create(N_max) * destroy(N_max)
# expect(nn, vacu), expect(nn, svac), expect(nn, thml)
```

The entropy calculated given on the photon statistics of a **coherent state**

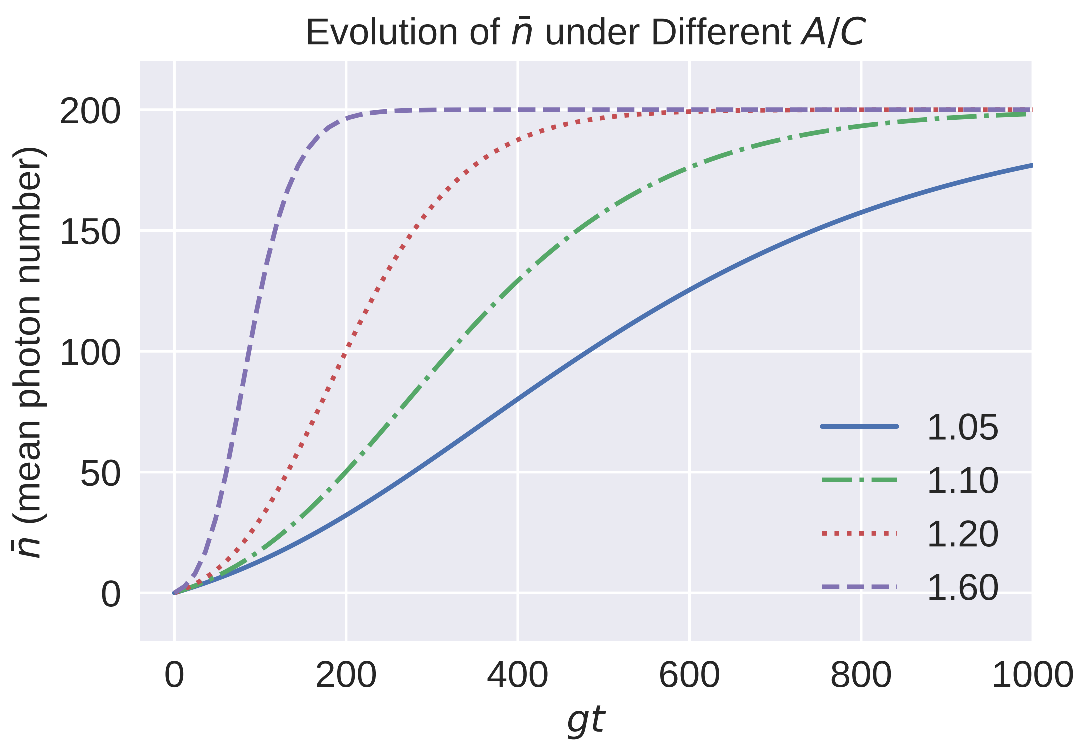
In [5]:
```
pns_cohe = [poisson.pmf(n, NBAR) for n in n_list]
ENTR_COHE = - sum([pn * np.log(pn) for pn in pns_cohe if pn > 0])
print('ENTROPY COHERENT: {:.4f}'.format(ENTR_COHE))
```

```
ENTROPY COHERENT: 4.0677
```

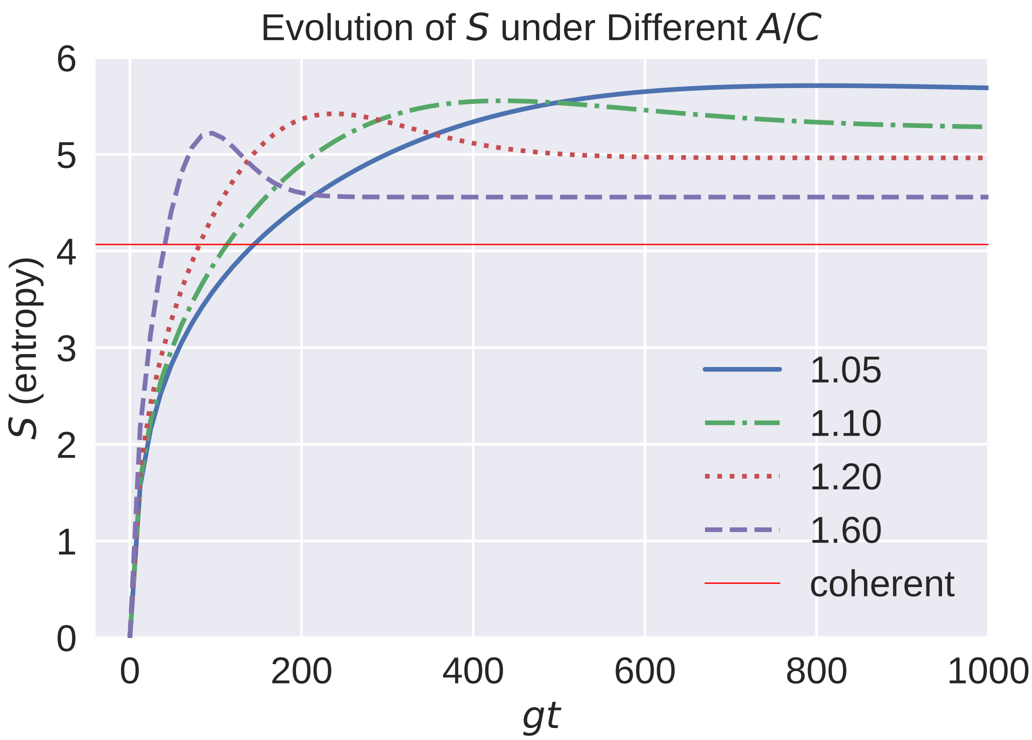
```
In [6]: # fig, ax = plt.subplots(figsize=(12, 3))
        # ax.bar(n_list, pns_cohe, width=0.5)
        # ax.set_xlim(100, 300)
        # ax.set_title('Possion Distribution with an Average of 200', fontsize=14);
```

0.1 Small Ratios

```
In [7]: plot_n_vs_t('./data/vacu_200/200_vacu_n1_df.csv', xlim=(-40, 1000), ylim=(-20, 220))
```

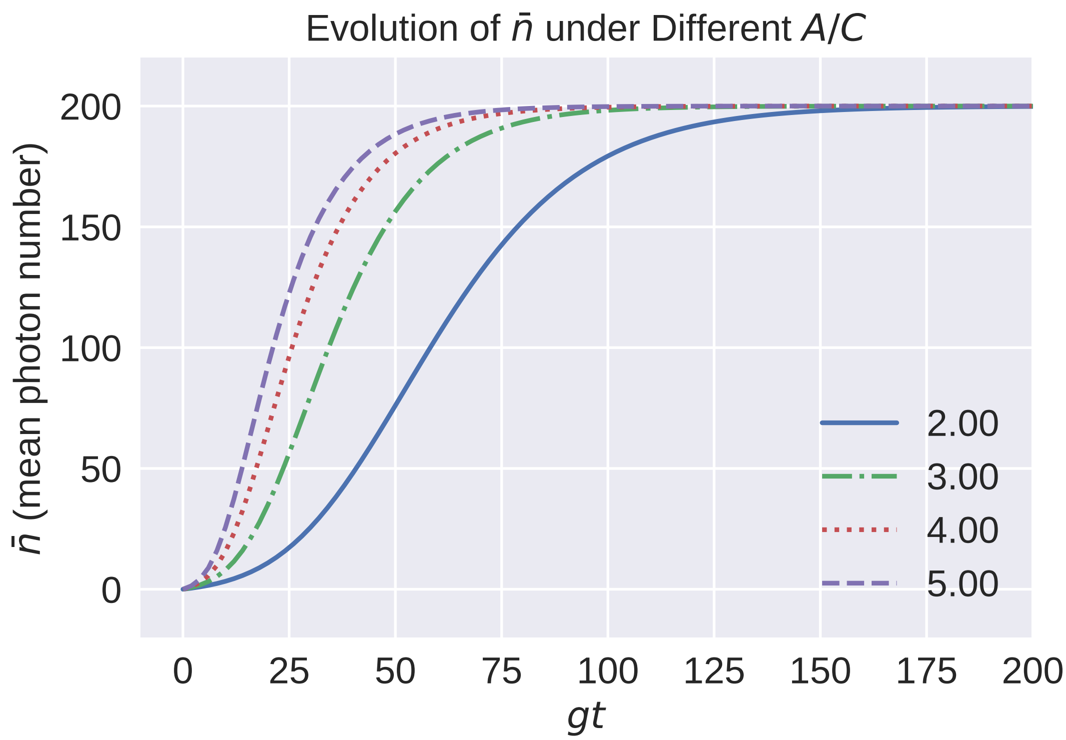



```
In [8]: plot_entr_vs_t('./data/vacu_200/200_vacu_entr1_df.csv', xlim=(-40, 1000), ylim=(0, 6))
```

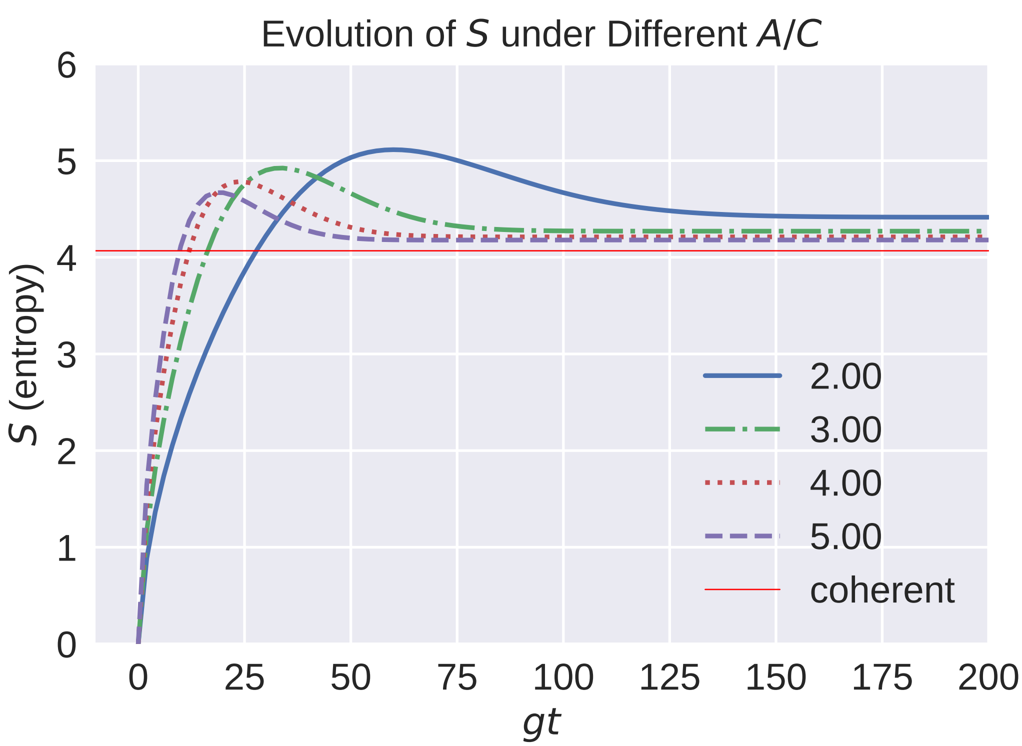


0.1.1 Medium Ratios

```
In [9]: plot_n_vs_t('./data/vacu_200/200_vacu_n2_df.csv', xlim=(-10, 200), ylim=(-20, 220))
```

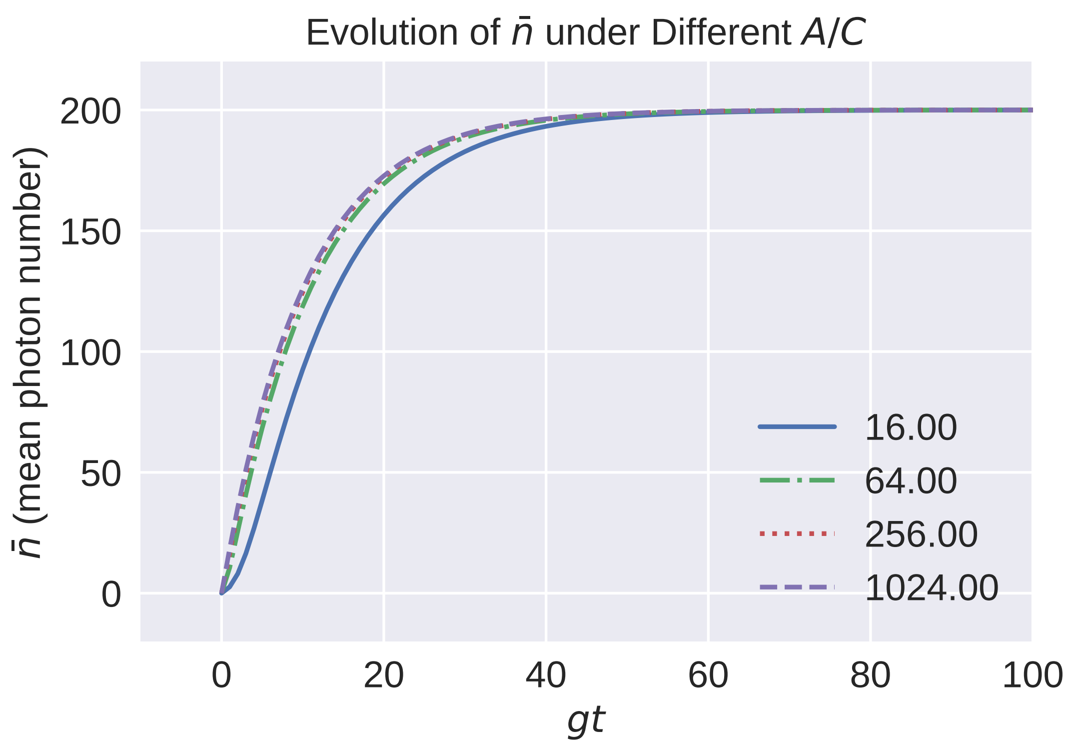



```
In [10]: plot_entr_vs_t('./data/vacu_200/200_vacu_entr2_df.csv', xlim=(-10, 200), ylim=(0, 6))
```

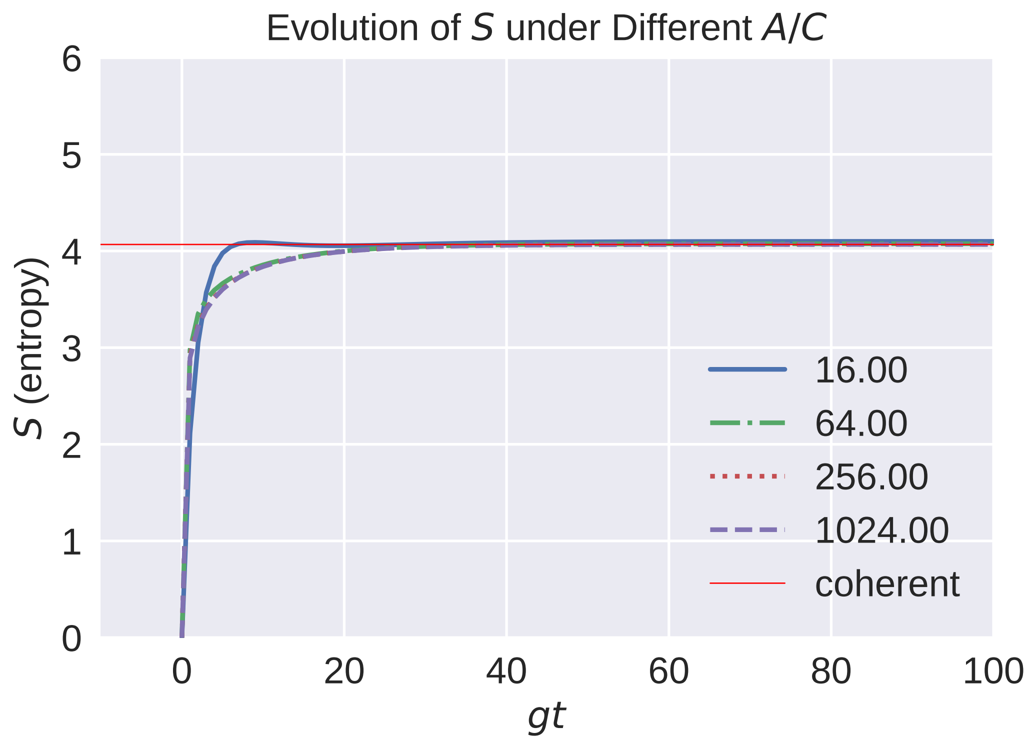


0.1.2 Large Ratios

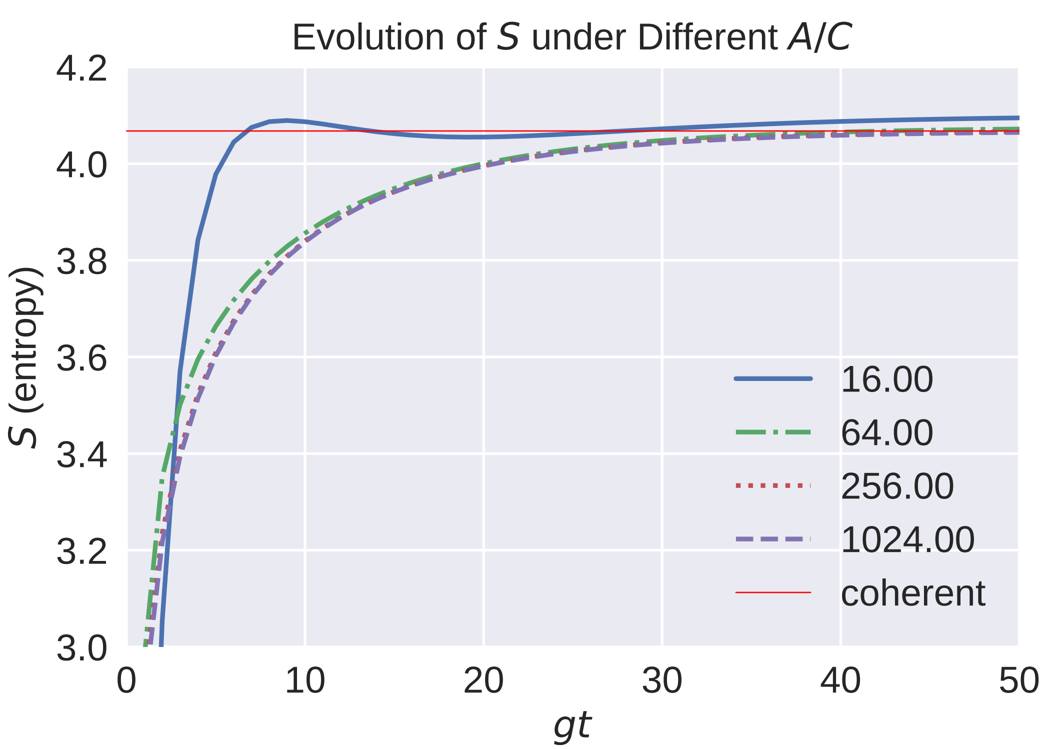
```
In [11]: plot_n_vs_t('./data/vacu_200/200_vacu_n3_df.csv', xlim=(-10, 100), ylim=(-20, 220))
```




```
In [13]: plot_entr_vs_t('./data/vacu_200/200_vacu_entr3_df.csv', xlim=(-10, 100), ylim=(0, 6))
```




```
In [12]: plot_entr_vs_t('./data/vacu_200/200_vacu_entr3_df.csv', xlim=(0, 50), ylim=(3, 4.2))
```



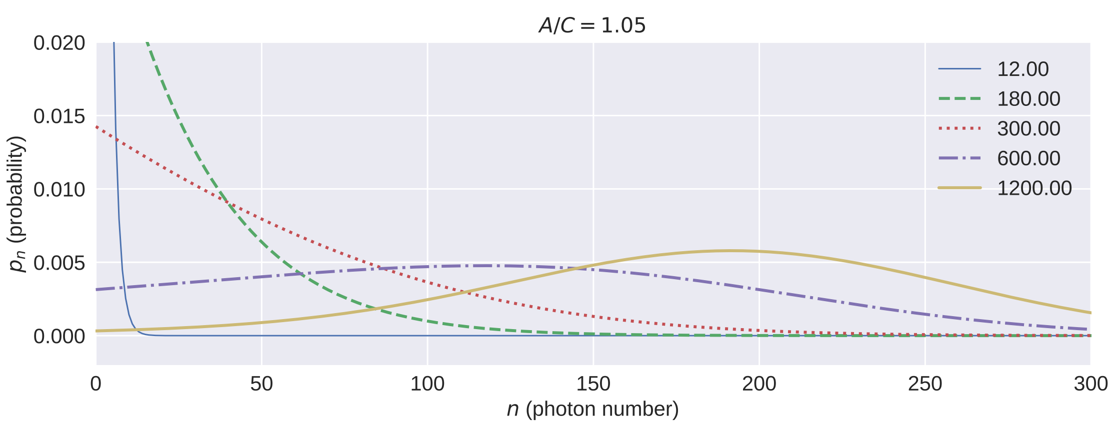
0.1.3 Evolution of Photon Statistics

```
In [147]: def plot_photon_statistics_evolution(l, gts, title, x1, x2, y1, y2):
              lstyle = ['-', '--', ':', '-.', '-', '--']
              lwidth = [1, 2, 2, 2, 2, 2]
              fig, ax = plt.subplots(sharex=True, figsize=(12, 4))
              t_list = l.t_list
              pns_all = l.get_pns()
          #     gts = (1, 4, 8, 15, 100)
              for i in range(len(gts)):
                  pns = pns_all[gts[i]]
                  ax.plot(np.arange(N_max), pns, \
                          linestyle=lstyle[i], linewidth=lwidth[i], \
                          label='{:4.2f}'.format(t_list[gts[i]] * G))
              ax.set_xlim(x1, x2)
              ax.set_ylim(y1, y2)
              ax.set_xlabel(r'$n$ (photon number)', fontsize=14)
              ax.set_ylabel(r'$p_n$ (probability)', fontsize=14)
              ax.tick_params(labelsize=14)
              ax.legend(fontsize=14)
              plt.title(title, fontsize=14);
```

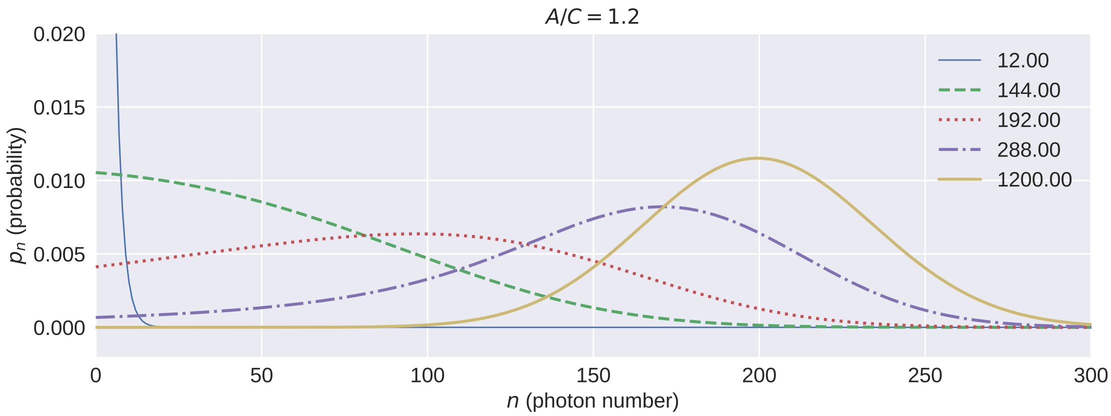
0.1.4 $A/C = 1.05, 1.2$

```
In [148]: laser_s = np.load('./data/vacu_200/200_vacu_l1.npz')
          ls1 = laser_s['lasers'].flatten()[0]['1.05']
          ls2 = laser_s['lasers'].flatten()[0]['1.20']
```

```
In [160]: plot_photon_statistics_evolution(ls1, (1, 15, 25, 50, 100), \
                                             r'$A/C=1.05$', 0, 300, -0.002, 0.02)
```



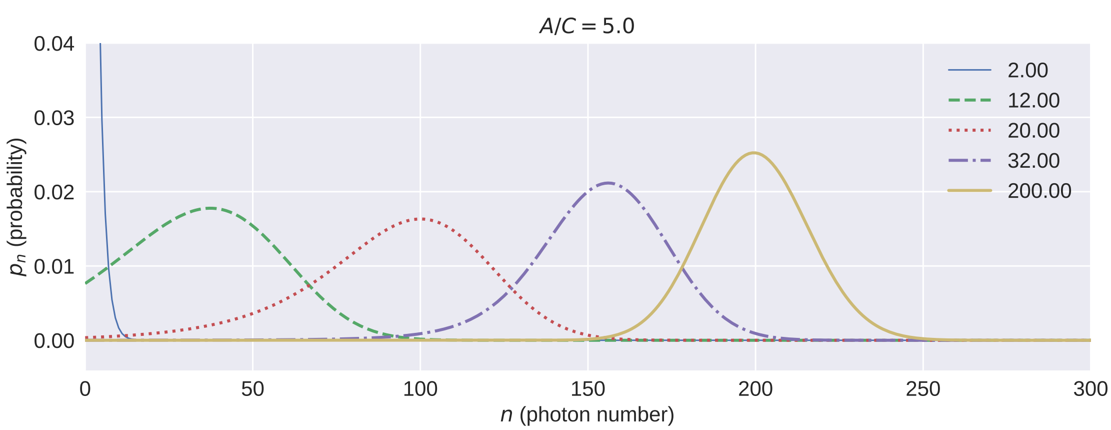
```
In [149]: plot_photon_statistics_evolution(ls2, (1, 12, 16, 24, 100), \
                                           r'$A/C=1.2$', 0, 300, -0.002, 0.02)
```



0.1.5 $A/C = 5$

```
In [134]: laser_m = np.load('./data/vacu_200/200_vacu_l2.npz')
          lm = laser_s['lasers'].flatten()[0]['5.00']
```

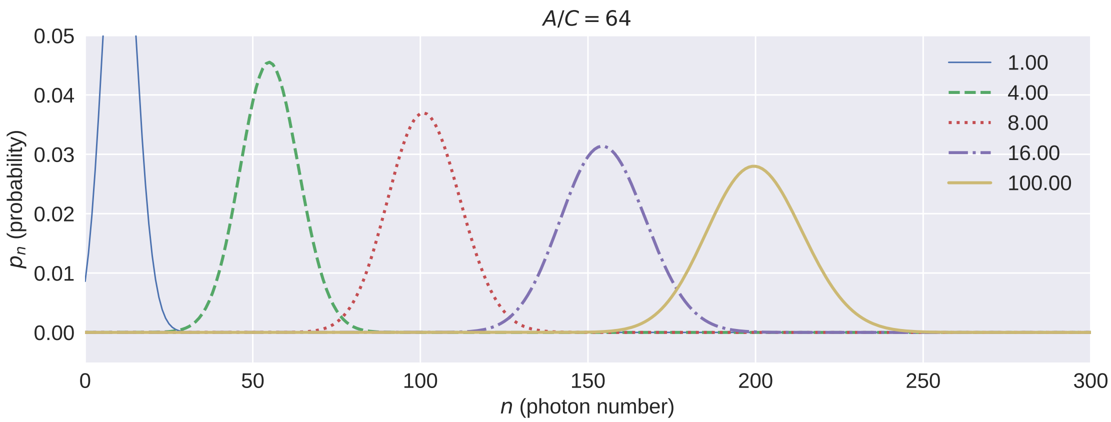
```
In [146]: plot_photon_statistics_evolution(lm, (1, 6, 10, 16, 100), \
                                           r'$A/C=5.0$', 0, 300, -0.004, 0.04)
```



0.1.6 $A/C = 64$

```
In [51]: laser_l = np.load('./data/vacu_200/200_vacu_l3.npz')
         ll = laser_l['lasers'].flatten()[0]['64.00']
```

```
In [25]: plot_photon_statistics_evolution(ll, (1, 4, 8, 16, 100), \
                                          r'$A/C=64$', 0, 300, -0.005, 0.05)
```



0.1.7 Variance and Entropy

```
In [189]: def calc_var_n(l):
              pns_all = l.get_pns()[1:70]
              var_n_all = []
              for pns in pns_all:
                  aver_n = sum([pns[i] * i for i in range(1000)])
                  aver_n2 = sum([pns[i] * i**2 for i in range(1000)])
                  var_n_all.append(aver_n2 - aver_n**2)
              return var_n_all
```

```
In [222]: def plot_varn_entr(l, title, ax):
              varn = calc_var_n(l)
              entr = np.log(np.sqrt(2.0 * np.pi) * np.sqrt(varn)) + 0.5
              t_list = l.get_tlist()[1:70] * l.g

              ln, = ax.plot(t_list, entr, label='entr based on var')
              ax.plot(t_list, l.get_entrs()[1:70], label ='numeric entropy')
              ax.plot(t_list, entr - l.get_entrs()[1:70], label ='difference')
              ax.legend(fontsize=14, loc=4)
          #     ax.set_ylabel("entropy")
              ax.set_title("Entropy for A/C = " + title)

              return ln


          def plot_varn_entr_all(lasers):
              fig, ax_array = plt.subplots(2, 2, figsize=(12, 8), sharey=True)
              ax = np.ravel(ax_array)
              i = 0
              for ratio, l in sorted(lasers.items()):
                  plot_varn_entr(l, ratio, ax[i])
                  i += 1
```

0.1.8 Time dependency of laser entropy and photon number variance

Longfei Fan
06/07/2017

The Poisson distribution can be appproximates by a Gaussian distribution. For a Gaussain distribution with variance $\sigma^2 = A$, mean 0, and truancated at R, the entropy is given by

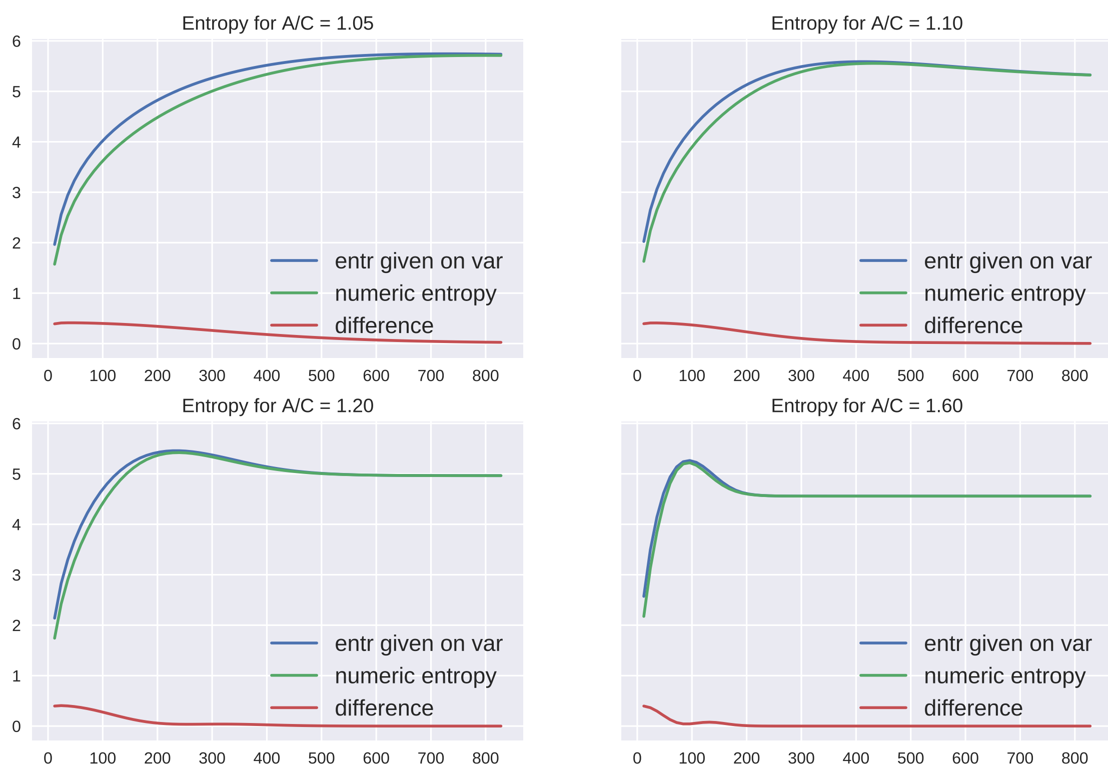
$$S = \log\sqrt{2\pi\sigma^2} + \frac{1}{2} + \log\left(\frac{1}{2}\text{erfc}\frac{R}{\sqrt{2\sigma^2}}\right) + \frac{R\exp(-\frac{R^2}{2\sigma^2})}{\sqrt{2\pi\sigma^2}\,\text{erfc}\frac{R}{\sigma^2}}$$

When the laser is at the steady state, $\sigma^2 \approx \frac{A^2}{BC}$, and $R = \frac{A}{B}$. Before the laser reach the steady state, the truncated position R may not be easy to find. However, fi the laser is operated appreciably above the threshold, the first two leading terms are still good approximation.

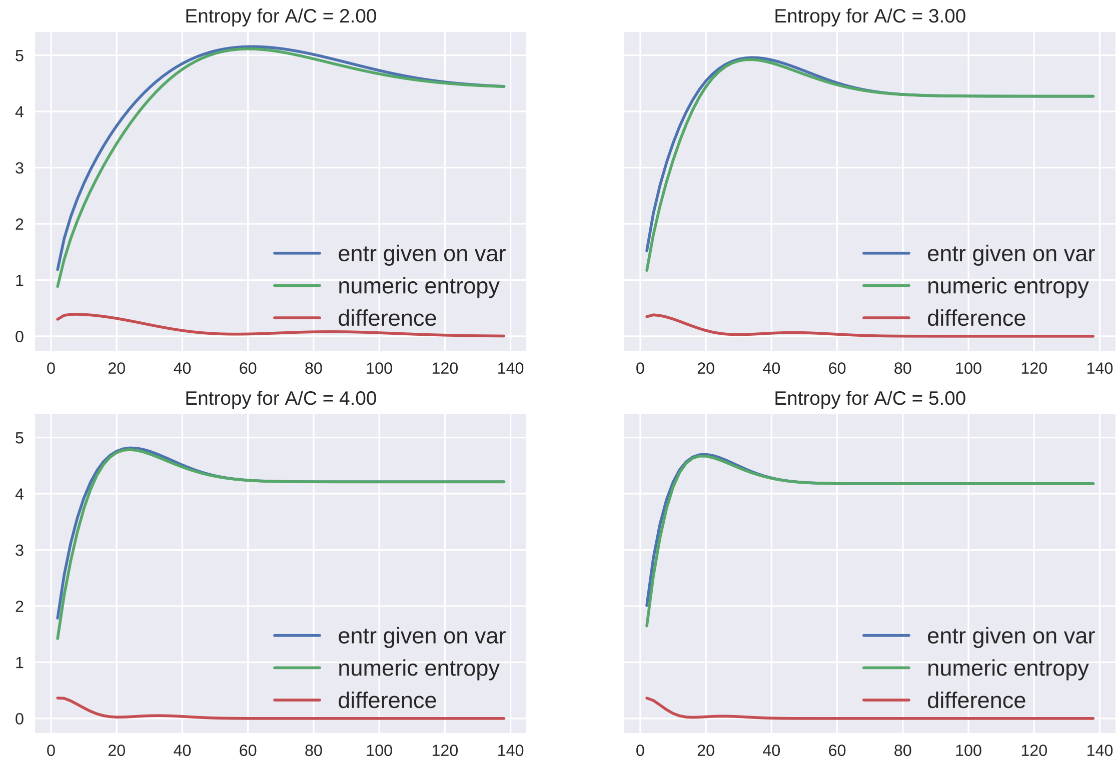
$$S = \log\sqrt{2\pi\sigma^2} + \frac{1}{2}$$

In the following three figures, I show the approximated entropy calculated using the above equation. The variance σ^2 is calculated given on the photon statistics $p(n)$ numerically. The lasers shown are operated at different values of A/C, but all have the same steady average photon number of 200.

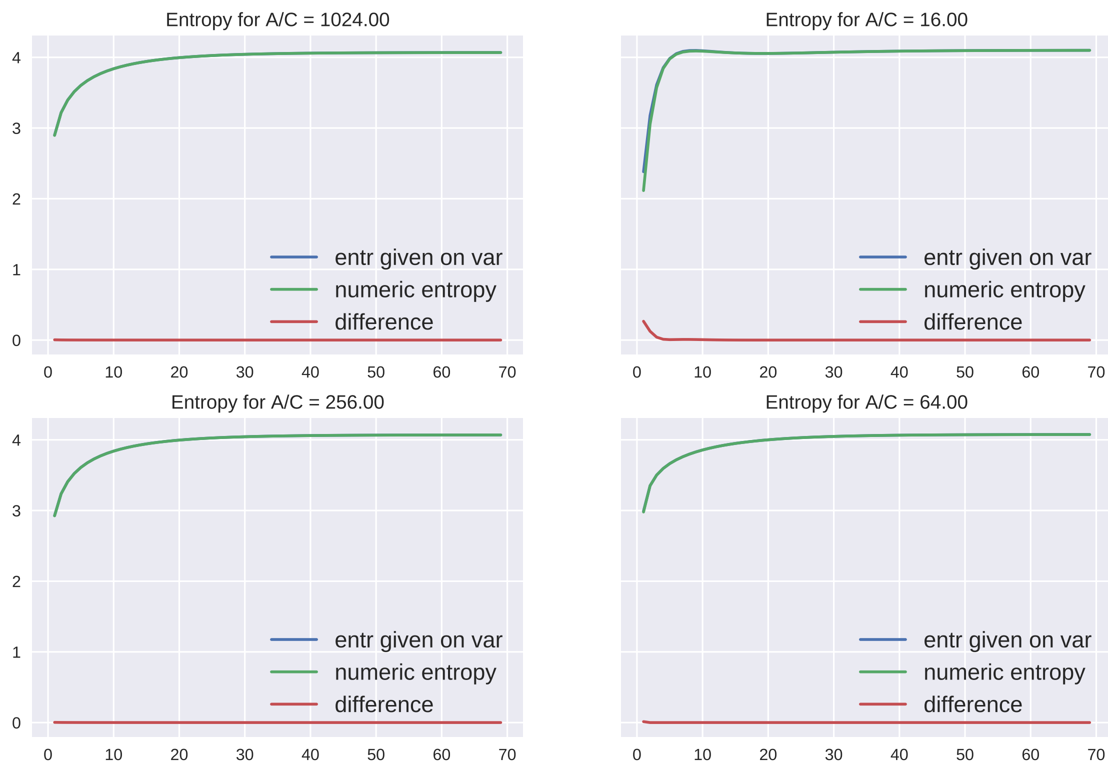
```
In [219]: laser_s = np.load('./data/vacu_200/200_vacu_l1.npz')['lasers'].flatten()[0]
          plot_varn_entr_all(laser_s)
```



```
In [220]: laser_m = np.load('./data/vacu_200/200_vacu_l2.npz')['lasers'].flatten()[0]
          plot_varn_entr_all(laser_m)
```




```
In [221]: laser_l = np.load('./data/vacu_200/200_vacu_l3.npz')['lasers'].flatten()[0]
          plot_varn_entr_all(laser_l)
```


Entropy for A/C = 1024.00
entr given on var
numeric entropy
difference
Entropy for A/C = 16.00
entr given on var
numeric entropy
difference
Entropy for A/C = 256.00
entr given on var
numeric entropy
difference
Entropy for A/C = 64.00
entr given on var
numeric entropy
difference

```
In [ ]:
```