example-pseudo-probability-functions

June 25, 2014

1 QuTiP example: Pseudo-probability functions

J.R. Johansson and P.D. Nation

For more information about QuTiP see http://qutip.org

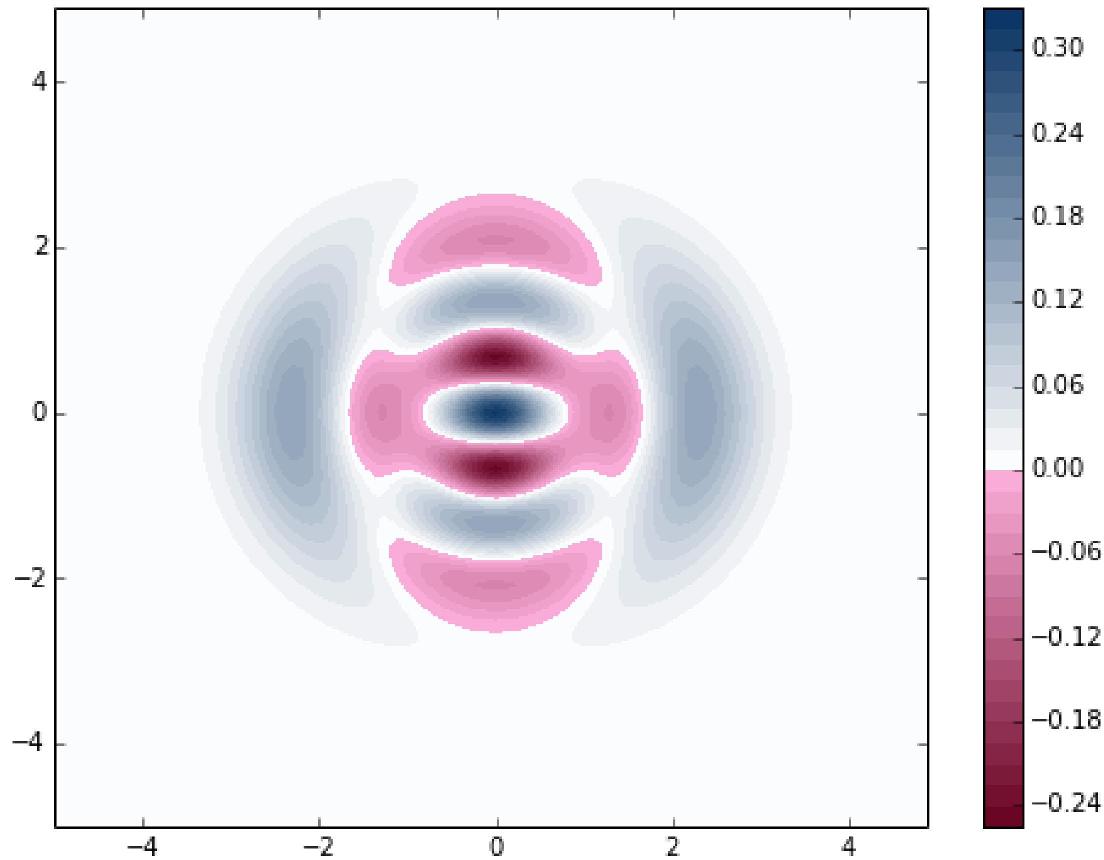
In [1]:
```
%matplotlib inline
import matplotlib.pylab as plt
import matplotlib as mpl
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm
```

In [2]:
```
from qutip import *
#from qutip.visualization import wigner_cmap
```

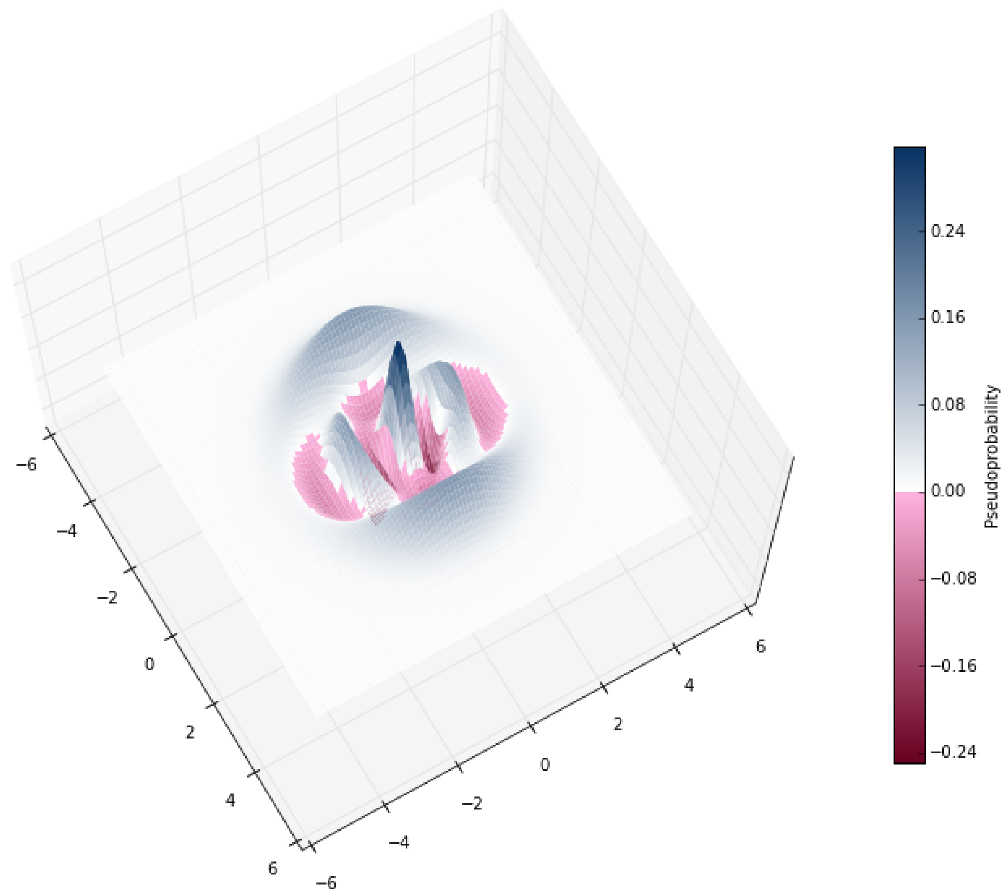
1.1 Wigner function for superposition of fock states

In [3]:
```
x = 1.0 / sqrt(2) * (basis(10, 4) + basis(10, 2))
xvec = arange(-5, 5, 10.0 / 100)
yvec = xvec
W = wigner(x, xvec, yvec)
cmap = wigner_cmap(W)
X, Y = meshgrid(xvec, yvec)
```

In [4]:
```
fig = plt.figure(figsize=(8,6))
plt.contourf(X, Y, W, 50, cmap=cmap)
plt.colorbar();
```




```
In [5]: fig = plt.figure(figsize=(10,8))
        ax = Axes3D(fig, azim=-30, elev=73)
        ax.plot_surface(X, Y, W, cmap=cmap, rstride=1, cstride=1, alpha=1, linewidth=0)
        ax.set_zlim3d(-0.25, 0.25)
        for a in ax.w_zaxis.get_ticklines() + ax.w_zaxis.get_ticklabels():
            a.set_visible(False)
        nrm = mpl.colors.Normalize(W.min(), W.max())
        cax, kw = mpl.colorbar.make_axes(ax, shrink=.66, pad=.02)
        cb1 = mpl.colorbar.ColorbarBase(cax, cmap=cmap, norm=nrm)
        cb1.set_label('Pseudoprobability')
```



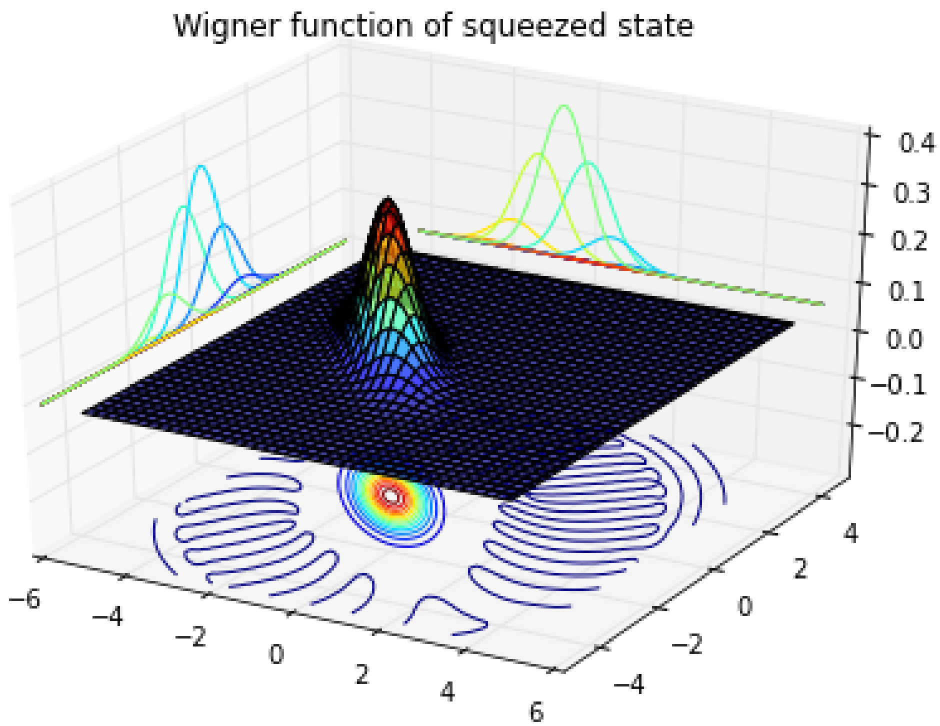
1.2 Winger and Q-function for squeezed states

In [6]:
```
N = 20;
alpha = -1.0;      # Coherent amplitude of field
epsilon = 0.5j;    # Squeezing parameter
a = destroy(N);

D = (alpha*a.dag()-conj(alpha)*a).expm();                    # Displacement
S = (0.5*conj(epsilon)*a*a-0.5*epsilon*a.dag()*a.dag()).expm();  # Squeezing
psi = D*S*basis(N,0); # Apply to vacuum state
g = 2;
```

1.2.1 Wigner function

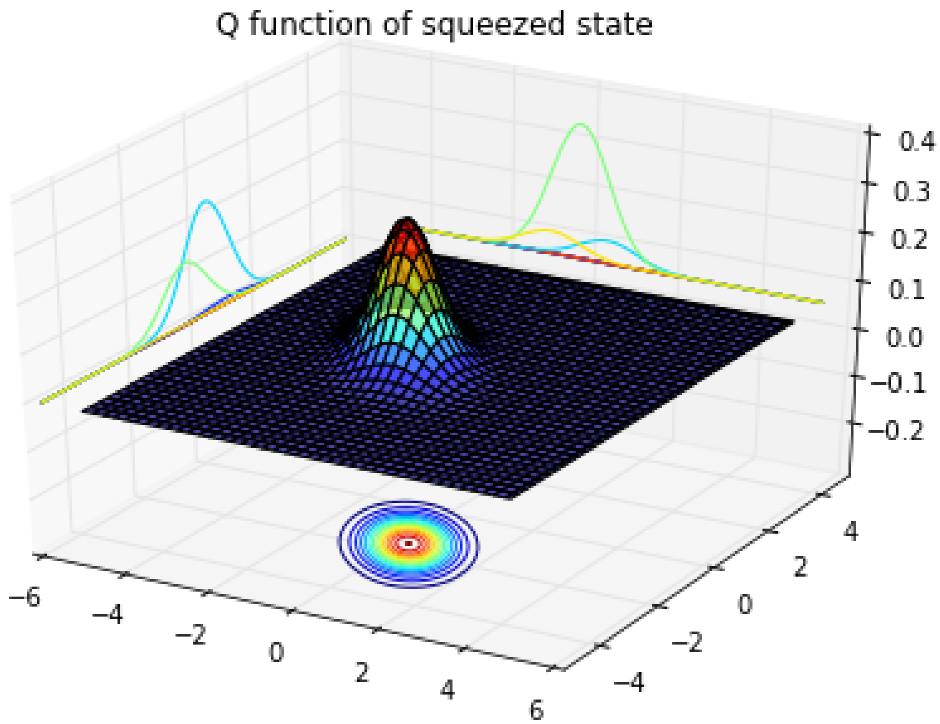
In [7]:
```
xvec = arange(-40.,40.)*5./40
X,Y = meshgrid(xvec, xvec)

W=wigner(psi,xvec,xvec)

fig1 = plt.figure()
ax = Axes3D(fig1)
```

```
ax.plot_surface(X, Y, W, rstride=2, cstride=2, cmap=cm.jet, alpha=0.7)
ax.contour(X, Y, W, 15,zdir='x', offset=-6)
ax.contour(X, Y, W, 15,zdir='y', offset=6)
ax.contour(X, Y, W, 15,zdir='z', offset=-0.3)
ax.set_xlim3d(-6,6)
ax.set_xlim3d(-6,6)
ax.set_zlim3d(-0.3,0.4)
plt.title('Wigner function of squeezed state');
```



1.2.2 Q-function

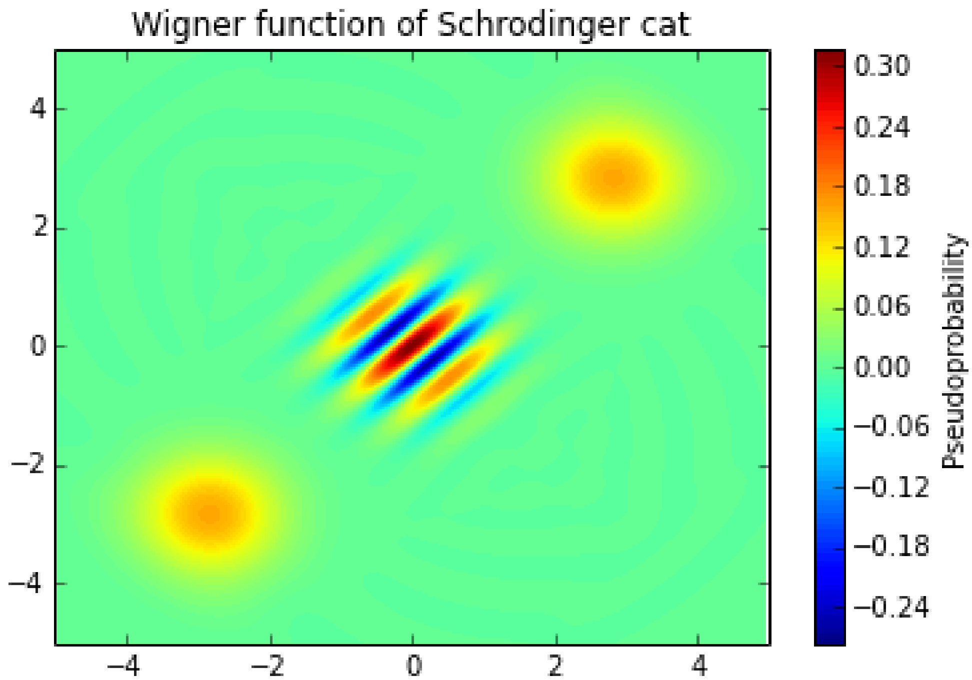
```
In [8]: Q = qfunc(psi,xvec,xvec,g);

        fig2 = plt.figure()
        ax = Axes3D(fig2)
        ax.plot_surface(X, Y, Q, rstride=2, cstride=2, cmap=cm.jet, alpha=0.7)
        ax.contour(X, Y, Q,zdir='x', offset=-6)
        ax.contour(X, Y, Q,zdir='y', offset=6)
        ax.contour(X, Y, Q, 15,zdir='z', offset=-0.4)
        ax.set_xlim3d(-6,6)
        ax.set_xlim3d(-6,6)
        ax.set_zlim3d(-0.3,0.4)
        plt.title('Q function of squeezed state');
```



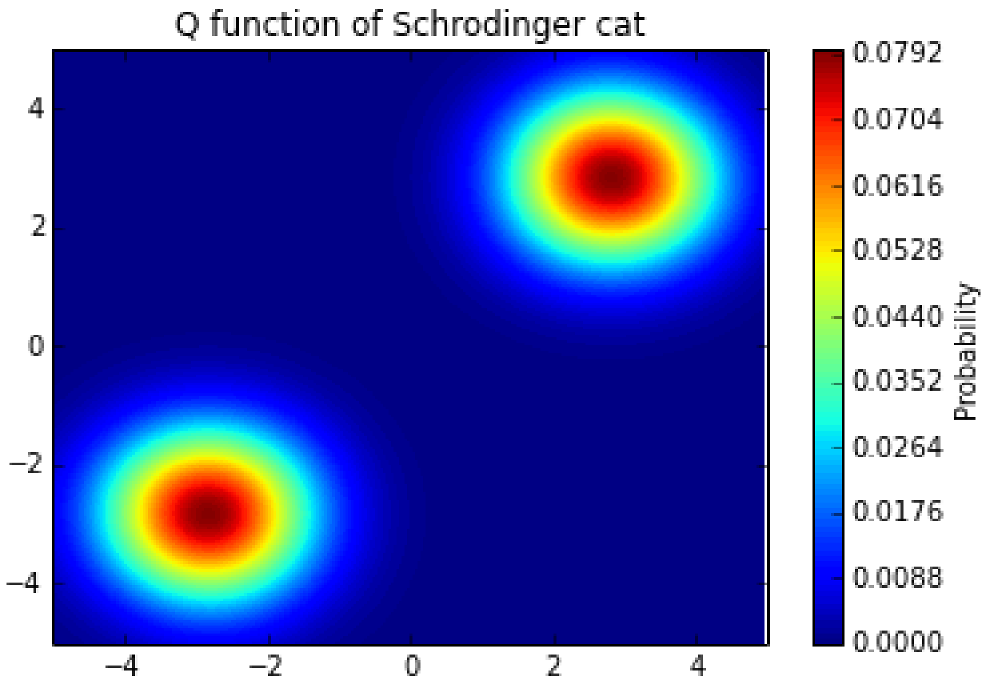
1.3 Schrodinger cat state

```
In [9]: N = 20;
        #amplitudes of coherent states
        alpha1=-2.0-2j
        alpha2=2.0+2j
        #define ladder oeprators
        a = destroy(N);
        #define displacement oeprators
        D1=(alpha1*dag(a)-conj(alpha1)*a).expm()
        D2=(alpha2*dag(a)-conj(alpha2)*a).expm()
        #sum of coherent states
        psi = sqrt(2)**-1*(D1+D2)*basis(N,0); # Apply to vacuum state
```

```
In [10]: #calculate Wigner function
         yvec = xvec = arange(-100.,100.)*5./100
         g=2.
         W = wigner(psi,xvec,yvec)
         c = plt.contourf(xvec,yvec,real(W),100)
         plt.xlim([-5,5])
         plt.ylim([-5,5])
         plt.title('Wigner function of Schrodinger cat')
         cbar = plt.colorbar(c)
         cbar.ax.set_ylabel('Pseudoprobability');
```




```
In [11]: #calculate Q function
         Q = qfunc(psi,xvec,yvec)
         qplt = plt.contourf(xvec,yvec,real(Q),100)
         plt.xlim([-5,5])
         plt.ylim([-5,5])
         plt.title('Q function of Schrodinger cat')
         cbar = plt.colorbar(qplt)
         cbar.ax.set_ylabel('Probability');
```



1.4 Software version:

In [12]:
```
from qutip.ipynbtools import version_table

version_table()
```

Out[12]: <IPython.core.display.HTML at 0x7fba77550d68>