example-JC-model-wigner-function

June 25, 2014

1 QuTiP example: Wigner function animation for the dynamics of the Jaynes-Cumming model

J.R. Johansson and P.D. Nation

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

```
In [1]: %pylab inline
```

```
Populating the interactive namespace from numpy and matplotlib
```

```
In [2]: from mpl_toolkits.mplot3d import Axes3D
        from matplotlib import cm
        import time
```

```
In [3]: from qutip import *
        from qutip.ipynbtools import plot_animation
```

```
In [4]: def jc_integrate(N, wc, wa, g, kappa, gamma, psi0, use_rwa, tlist):

            # Hamiltonian
            idc = qeye(N)
            ida = qeye(2)

            a  = tensor(destroy(N), ida)
            sm = tensor(idc, destroy(2))

            if use_rwa:
                # use the rotating wave approxiation
                H = wc * a.dag() * a + wa * sm.dag() * sm + g * (a.dag() * sm + a * sm.dag())
            else:
                H = wc * a.dag() * a + wa * sm.dag() * sm + g * (a.dag() + a) * (sm + sm.dag())

            # collapse operators
            c_op_list = []

            n_th_a = 0.0 # zero temperature

            rate = kappa * (1 + n_th_a)
            if rate > 0.0:
                c_op_list.append(sqrt(rate) * a)

            rate = kappa * n_th_a
            if rate > 0.0:
                c_op_list.append(sqrt(rate) * a.dag())

            rate = gamma
            if rate > 0.0:
                c_op_list.append(sqrt(rate) * sm)

            # evolve and calculate return state vectors
            result = mesolve(H, psi0, tlist, c_op_list, [])

            return result
```

```
In [5]: # parameters
        wc = 1.0 * 2 * pi   # cavity frequency
        wa = 1.0 * 2 * pi   # atom frequency
        g  = 0.05 * 2 * pi  # coupling strength
        kappa = 0.05        # cavity dissipation rate
        gamma = 0.15        # atom dissipation rate
        N = 10              # number of cavity fock states

        use_rwa = True

        # intial state
        psi0 = tensor(basis(N,0), basis(2,1))    # start with an excited atom
        #psi0 = tensor(coherent(N,1.5), basis(2,0))   # or a coherent state the in cavity
        #psi0 = tensor((coherent(N,2.0)+coherent(N,-2.0)).unit(), basis(2,0))
        # or a superposition of coherent states

        tlist = linspace(0, 30, 150)
```

```
In [6]: result = jc_integrate(N, wc, wa, g, kappa, gamma, psi0, use_rwa, tlist)
```

```
In [7]: xvec = linspace(-5.,5.,100)
        X,Y = meshgrid(xvec, xvec)
```

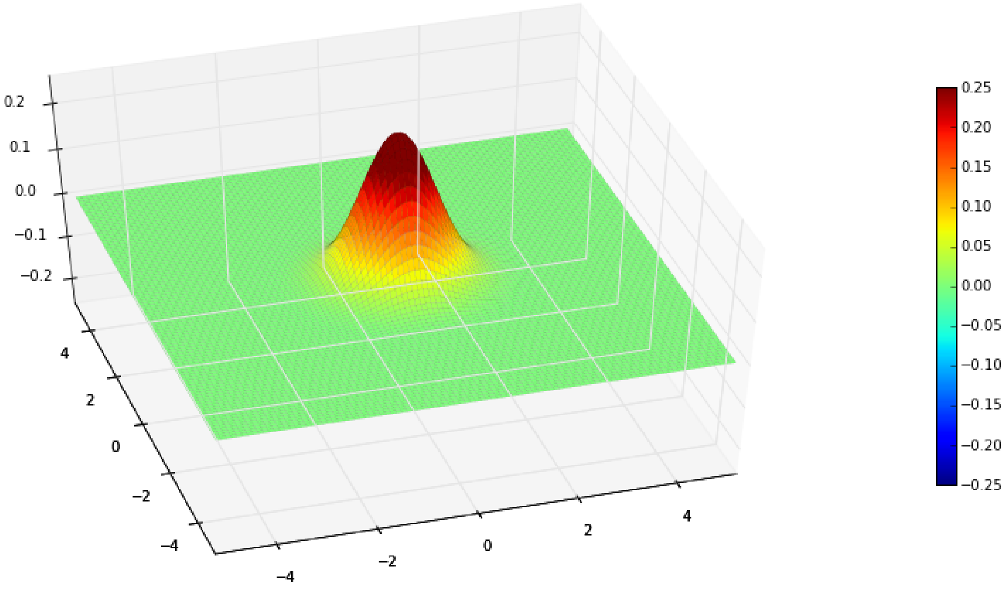
```
In [8]: def plot_setup(result):

            fig = plt.figure(figsize=(12, 6))
            ax = Axes3D(fig, azim=-107, elev=49)

            return fig, ax
```

```
In [9]: cb = None

        def plot_result(result, n, fig=None, axes=None):

            global cb

            if fig is None or axes is None:
                fig, ax = plot_setup(result)

            axes.cla()

            # trace out the atom
            rho_cavity = ptrace(result.states[n], 0)
```

```
    W = wigner(rho_cavity, xvec, xvec)

    surf = axes.plot_surface(X, Y, W, rstride=1, cstride=1, cmap=cm.jet,
                           alpha=1.0, linewidth=0.05, vmax=0.25, vmin=-0.25)
    axes.set_xlim3d(-5, 5)
    axes.set_ylim3d(-5, 5)
    axes.set_zlim3d(-0.25, 0.25)

    if not cb:
        cb = plt.colorbar(surf, shrink=0.65, aspect=20)

    #check_update

    return fig, axes
```

```
In [10]: plot_animation(plot_setup, plot_result, result)
```

```
    ---------------------------------------------------------------------------
UnicodeDecodeError                        Traceback (most recent call last)

    <ipython-input-10-78c818104ba7> in <module>()
----> 1 plot_animation(plot_setup, plot_result, result)


    /usr/local/lib/python3.4/dist-packages/qutip/ipynbtools.py in plot_animation(plot_setup_func, pl
    282     anim = animation.FuncAnimation(fig, update, frames=len(result.times), blit=True)
    283
--> 284     anim.save(name + '.mp4', fps=10, writer="avconv", codec="libx264")
    285
    286     plt.close(fig)


    /usr/lib/python3/dist-packages/matplotlib/animation.py in save(self, filename, writer, fps, dpi
    717                     #TODO: Need to see if turning off blit is really necessary
    718                     anim._draw_next_frame(d, blit=False)
--> 719                 writer.grab_frame(**savefig_kwargs)
    720
    721         # Reconnect signal for first draw if necessary


    /usr/lib/python3/dist-packages/matplotlib/animation.py in grab_frame(self, **savefig_kwargs)
    203             # frame format and dpi.
    204             self.fig.savefig(self._frame_sink(), format=self.frame_format,
--> 205                              dpi=self.dpi, **savefig_kwargs)
    206         except RuntimeError:
    207             out, err = self._proc.communicate()


    /usr/lib/python3/dist-packages/matplotlib/figure.py in savefig(self, *args, **kwargs)
   1420             self.set_frameon(frameon)
   1421
-> 1422         self.canvas.print_figure(*args, **kwargs)
   1423
   1424         if frameon:
    /usr/lib/python3/dist-packages/matplotlib/backend_bases.py in print_figure(self, filename, dpi,
   2223                 restore_bbox()
   2224
-> 2225             self.figure.dpi = origDPI
   2226             self.figure.set_facecolor(origfacecolor)
   2227             self.figure.set_edgecolor(origedgecolor)


    /usr/lib/python3/dist-packages/matplotlib/figure.py in _set_dpi(self, dpi)
    383     def _set_dpi(self, dpi):
    384         self._dpi = dpi
--> 385         self.dpi_scale_trans.clear().scale(dpi, dpi)
    386         self.callbacks.process('dpi_changed', self)
    387     dpi = property(_get_dpi, _set_dpi)


    /usr/lib/python3/dist-packages/matplotlib/transforms.py in clear(self)
   1784         Reset the underlying matrix to the identity transform.
   1785         """
-> 1786         self._mtx = np.identity(3)
   1787         self.invalidate()
   1788         return self


    /usr/lib/python3/dist-packages/numpy/core/numeric.py in identity(n, dtype)
   2051
   2052     """
-> 2053     from numpy import eye
   2054     return eye(n, dtype=dtype)
   2055


    /usr/lib/python3.4/importlib/_bootstrap.py in _handle_fromlist(module, fromlist, import_)


    UnicodeDecodeError: 'utf-8' codec can't decode byte 0xff in position 0: invalid start byte
```



1.1 Versions

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

        version_table()
```