example-bloch-sphere-animation

June 25, 2014

1 QuTiP example: Bloch sphere animation

J.R. Johansson and P.D. Nation

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

Animation with qutip and matplotlib: decaying qubit visualized in a Bloch sphere. (Animation with matplotlib does not work yet in python3)

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

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

```
In [2]: import matplotlib.animation as animation
        from mpl_toolkits.mplot3d import Axes3D
```

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

```
In [4]: def qubit_integrate(w, theta, gamma1, gamma2, psi0, tlist):
            # operators and the hamiltonian
            sx = sigmax(); sy = sigmay(); sz = sigmaz(); sm = sigmam()
            H = w * (cos(theta) * sz + sin(theta) * sx)
            # collapse operators
            c_op_list = []
            n_th = 0.5 # temperature
            rate = gamma1 * (n_th + 1)
            if rate > 0.0: c_op_list.append(sqrt(rate) * sm)
            rate = gamma1 * n_th
            if rate > 0.0: c_op_list.append(sqrt(rate) * sm.dag())
            rate = gamma2
            if rate > 0.0: c_op_list.append(sqrt(rate) * sz)


            # evolve and calculate expectation values
            output = mesolve(H, psi0, tlist, c_op_list, [sx, sy, sz])
            return output
```

```
In [5]: w     = 1.0 * 2 * pi   # qubit angular frequency
        theta = 0.2 * pi       # qubit angle from sigma_z axis (toward sigma_x axis)
        gamma1 = 0.5           # qubit relaxation rate
        gamma2 = 0.2           # qubit dephasing rate
        # initial state
        a = 1.0
        psi0 = (a* basis(2,0) + (1-a)*basis(2,1))/(sqrt(a**2 + (1-a)**2))
        tlist = linspace(0, 4, 150)
```

```
In [6]: result = qubit_integrate(w, theta, gamma1, gamma2, psi0, tlist)

In [7]: def plot_setup(result):

            fig = figure(figsize=(8,8))
            axes = Axes3D(fig, azim=-40,elev=30)

            return fig, axes

In [8]: sphere = None

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

            global sphere

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

            if not sphere:
                sphere = Bloch(axes=axes)
                sphere.vector_color = ['r']

            sphere.clear()
            sphere.add_vectors([sin(theta),0,cos(theta)])
            sphere.add_points([result.expect[0][:n+1], result.expect[1][:n+1], \
                               result.expect[2][:n+1]], meth='l')
            sphere.make_sphere()

            return fig, axes

In [9]: plot_animation(plot_setup, plot_result, result)

Out[9]: <IPython.core.display.HTML at 0x7fd83786c410>

<matplotlib.figure.Figure at 0x7fd837881650>
```

1.1 Versions

```
In [10]: from qutip.ipynbtools import version_table

         version_table()

Out[10]: <IPython.core.display.HTML at 0x7fd83730fd90>
```