Lecture 6: Time evolution and quantum quenches

Hendrik Weimer

Institute for Theoretical Physics, Leibniz University Hannover

Quantum Physics with Python, 6 June 2016

Recap: States and operators in QuTiP

```
psi = alpha*basis(2, 0) + beta*basis(2, 1)

J = 1
H = J*(tensor(sigmax(), sigmax()) + tensor(sigmay(), sigmay()))
print(H)
```

Output:

```
Quantum object: dims = [[2, 2], [2, 2]], shape = [4, 4], type =
Qobj data =
[[ 0.  0.  0.  0.]
 [ 0.  0.  2.  0.]
 [ 0.  2.  0.  0.]
 [ 0.  0.  0.  0.]]
```

Recap: Eigenvalues and eigenvectors

```
print(H.eigenenergies())
print(H.eigenstates(eigvals=1))
```

Output:

```
(array([-2.]), array([ Quantum object: dims = [[2, 2], [1, 1]],
Qobj data =
[[ 0.        ]
 [-0.70710678]
 [ 0.70710678]
 [ 0.        ]]], dtype=object))
```

Outline of the course

1. Introduction to Python
2. SciPy/NumPy packages
3. Plotting and fitting
4. QuTiP: states and operators
5. Ground state problems
6. **Time evolution and quantum quenches**
7. Quantum master equations
8. Generation of squeezed states
9. Quantum computing
10. Grover's algorithm and quantum machine learning
11. Student presentations

The Schrödinger equation in QuTiP

```
sesolve(H, psi0, tlist, e_ops, args={},options=None)
```

`H`	Hamiltonian
`psi0`	Initial state
`tlist`	Desired points in time
`e_ops`	Operators for expectation values (or empty list)

```
import numpy as np
from qutip import basis, sigmax, sesolve

psi0 = basis(2, 0)
H = sigmax()
tlist = np.linspace(0, np.pi, 5)
res = sesolve(H, psi0, tlist, [])
print(res)
```

Output:

```
Result object with sesolve data.
--------------------------------

states = True
num_collapse = 0
```

List of states

```
print(res.states)
```

Output:

```
[Quantum object: dims = [[2], [1]], shape = [2, 1], type = ket
Qobj data =
[[ 1.]
 [ 0.]], Quantum object: dims = [[2], [1]], shape = [2, 1], type
Qobj data =
[[ 0.70710669+0.j         ]
 [ 0.00000000-0.70710687j]], Quantum object: dims = [[2], [1]],
Qobj data =
[[ -1.29572476e-07+0.j]
 [  0.00000000e+00-1.j]], Quantum object: dims = [[2], [1]], sha
Qobj data =
[[-0.70710698+0.j         ]
 [ 0.00000000-0.70710658j]], Quantum object: dims = [[2], [1]],
Qobj data =
[[-1. +0.00000000e+00j]
 [ 0. +2.81856628e-07j]]]
```

Solver options

`atol`	Absolute tolerance
`rtol`	Relative tolerance
`method`	Integrator ('adams'/'bdf')
`nsteps`	Maximum number of steps between two times
...	

```
from qutip import Options
opts = Options()
opts['atol'] = 1e-5
```

A simple two-level system

- Time-independent Hamiltonian

$$H = \Delta\sigma_z + g\sigma_x$$

- Time-dependent Hamiltonian

$$H = \Delta\sigma_z + g\sin(\omega t)\sigma_x$$

Quantum quenches

- Ising model Hamiltonian

$$H = g \sum_{i=0}^{N-1} \sigma_z^{(i)} - \sum_{\langle ij \rangle} \sigma_x^{(i)} \sigma_x^{(j)}$$

- Start in the ferromagnetic ground state for $g = g_0$
- Suddenly change ("quench") the Hamiltonian to $g = g'$
- Observe the time evolution