QuTiP stochastic solvers: `mcsolve`

QuTiP

Quantum Toolbox in Python

Nathan Shammah
Theoretical Quantum Physics Lab
Cluster for Pioneering Research
RIKEN, Saitama, Japan

1st July 2019
CM Seminar
ICTP, Trieste

QuTiP: The Quantum *Physics* Simulator

The **Qu**antum **T**oolbox **i**n **P**ython: A toolbox to study the **open** quantum dynamics of realistic systems.

QuTiP

Interactive Lectures @ ICTP, Leonardo Building

Take a snapshot

Tue 25th June - 11:45am, Seminar Room – Driven-dissipative models in quantum physics
Wed 26th June - 11am, Seminar Room – Quantum Open Source & Introduction to QuTiP
Thur 27th June - 9am, Computer Room – Hands-on session on QuTiP’s main features
Mon 1st July - 9am, Computer Room – QuTiP stochastic solvers
Tue 2nd July - 9am, Computer Room – How to Build your Own Scientific Software Library in Python
(Wed 3rd July - 9am, Computer Room – Extra meeting: SISSA/ICTP projects)

https://github.com/nathanshammah/interactive-notebooks

QuTiP: Interactive Notebooks

The **Qu**antum **T**oolbox **i**n **P**ython

QuTiP

You can find an interactive notebook at

https://github.com/nathanshammah/

Repository: **interactive-notebooks**

You can run the notebook live at

https://mybinder.org/v2/gh/nathanshammah/interactive-notebooks/binder

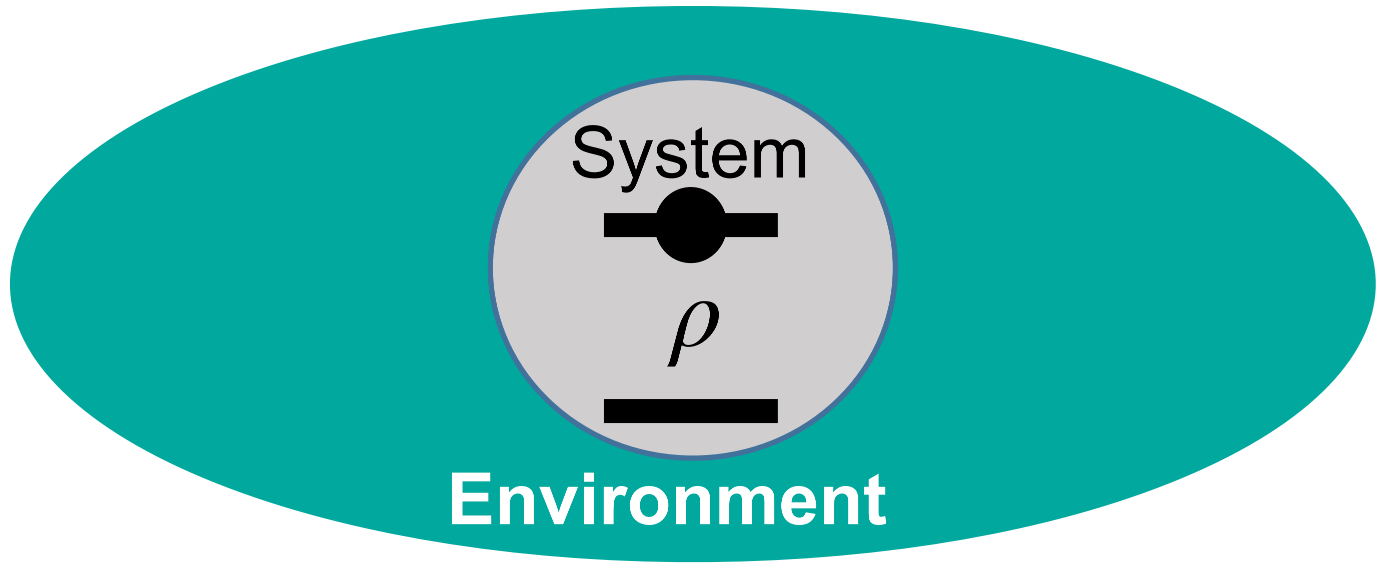
Take a snapshot

QuTiP: The Quantum *Physics* Simulator

The **Qu**antum **T**oolbox **i**n **P**ython

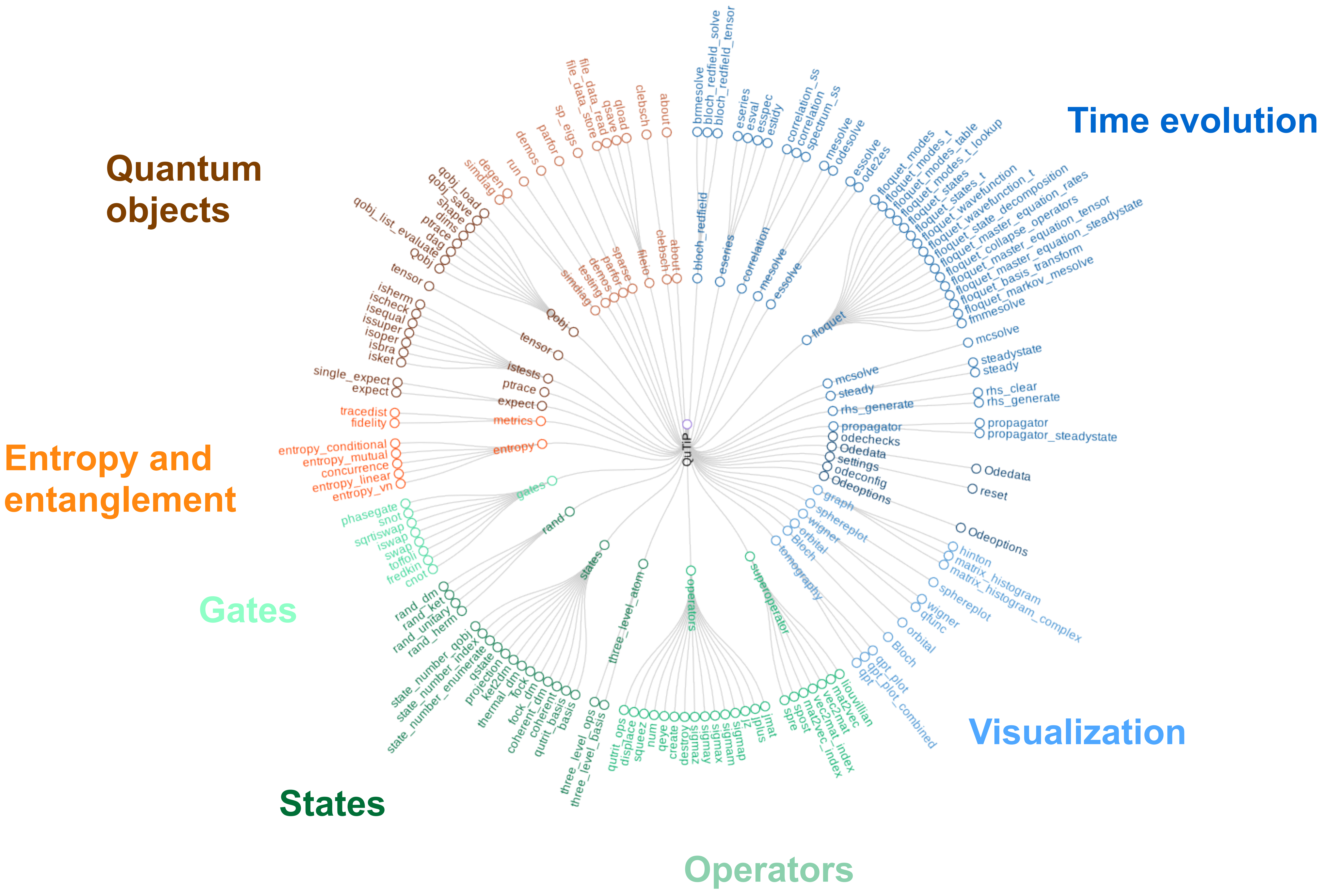
QuTiP

A toolbox to study the **open** quantum dynamics of realistic systems.



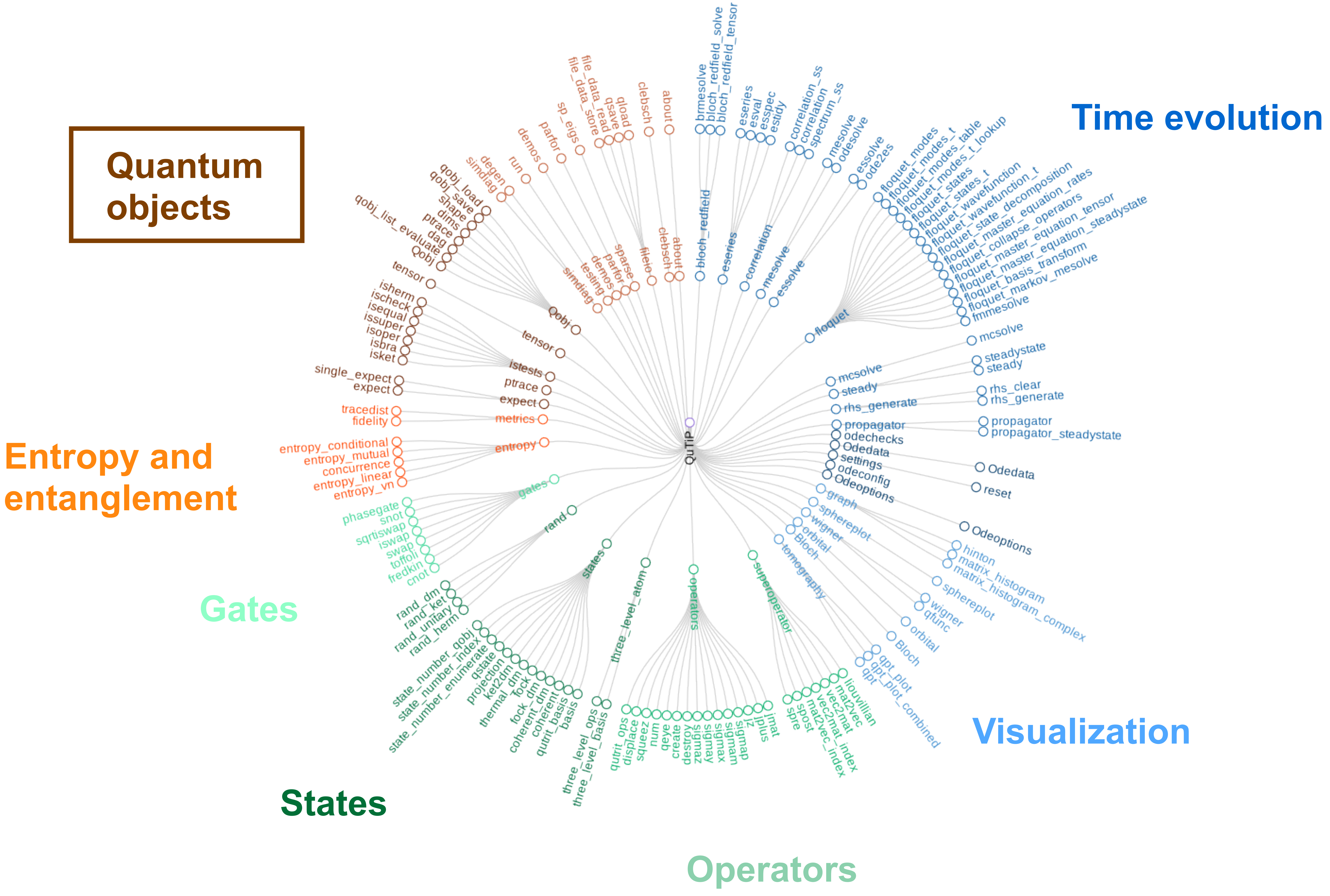
QuTiP: Map of code

The **Qu**antum **T**oolbox **i**n **P**ython



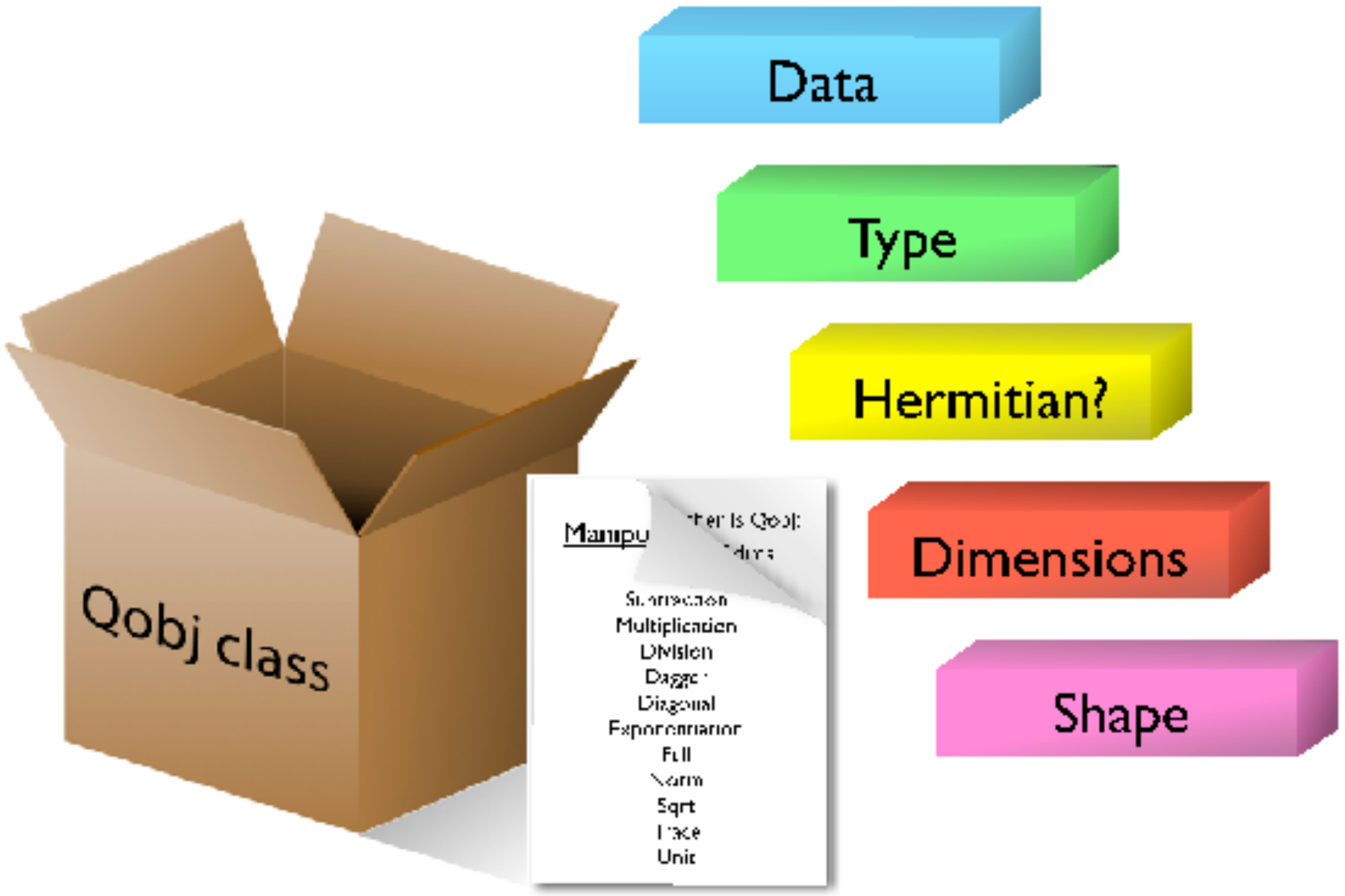
QuTiP: Map of code

The **Qu**antum **T**oolbox **i**n **P**ython



QuTiP: The `QObj` class

The **Qu**antum **T**oolbox **i**n **P**ython



J. R. Johansson, P. D. Nation, and F. Nori, Comp. Phys. Comm. **183**, 1760–1772 (2012)
QuTiP: An open-source Python framework for the dynamics of open quantum systems

QuTiP: The `QObj` class

The **Qu**antum **T**oolbox **i**n **P**ython

- State and operators are declared as `QObj`

- **Generate states** and operators

- Algebra (bosonic)

$$AB - BA \neq 0$$

$$\mathscr{E}(\rho) = A\rho B^\dagger \longrightarrow \mathscr{D} = B^* \otimes A$$

$$\dot{\rho} = \mathscr{D}\rho$$

```
>> q = Qobj([1], [0])
Quantum object: dims = [[2], [1]],
shape = (2, 1), type = ket
```

$$\begin{pmatrix} 1.0 \\ 0.0 \end{pmatrix}$$

```
>> d = destroy(2)
Quantum object: dims = [[2], [2]],
shape = (2, 2), type = oper,
isherm = False
```

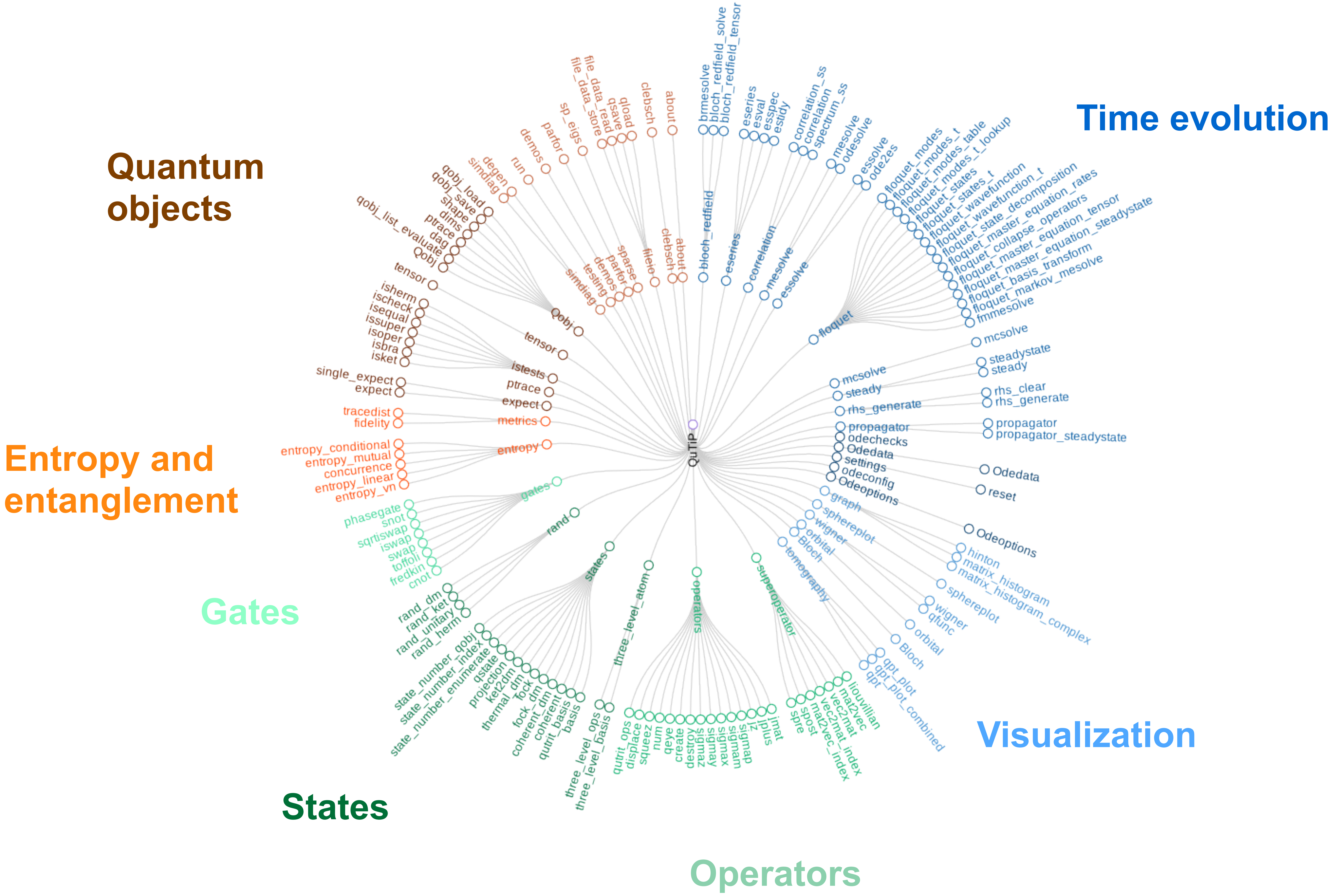
$$\begin{pmatrix} 0.0 & 1.0 \\ 0.0 & 0.0 \end{pmatrix}$$

```
>> q.dag()
Quantum object: dims = [[1], [2]],
shape = (1, 2), type = bra
```

$$\begin{pmatrix} 1.0 & 0.0 \end{pmatrix}$$

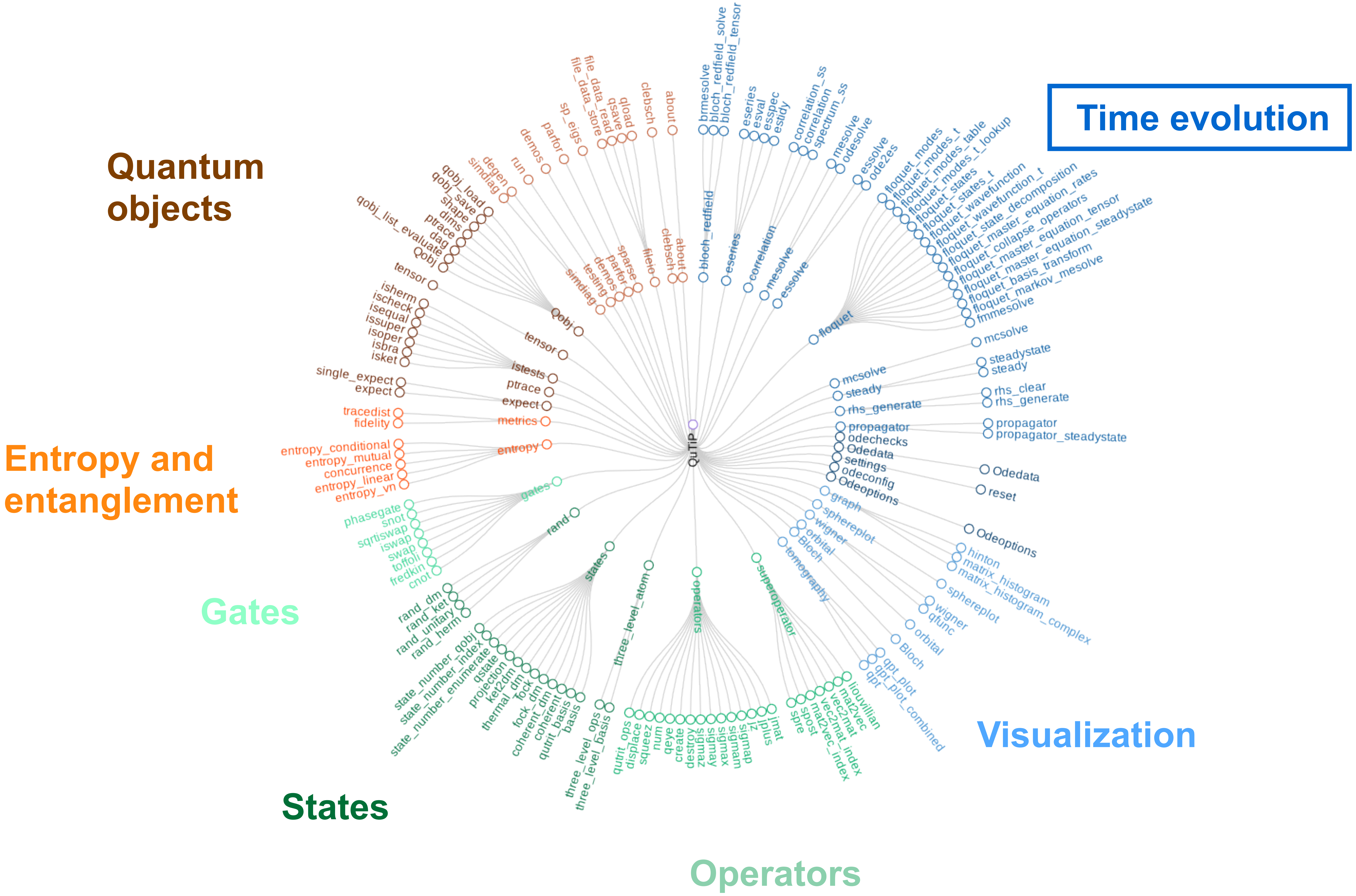
QuTiP: Map of code

The **Qu**antum **T**oolbox **i**n **P**ython



QuTiP: Map of code

The **Qu**antum **T**oolbox **i**n **P**ython



QuTiP: Numerical Solvers

The **Qu**antum **T**oolbox **i**n **P**ython

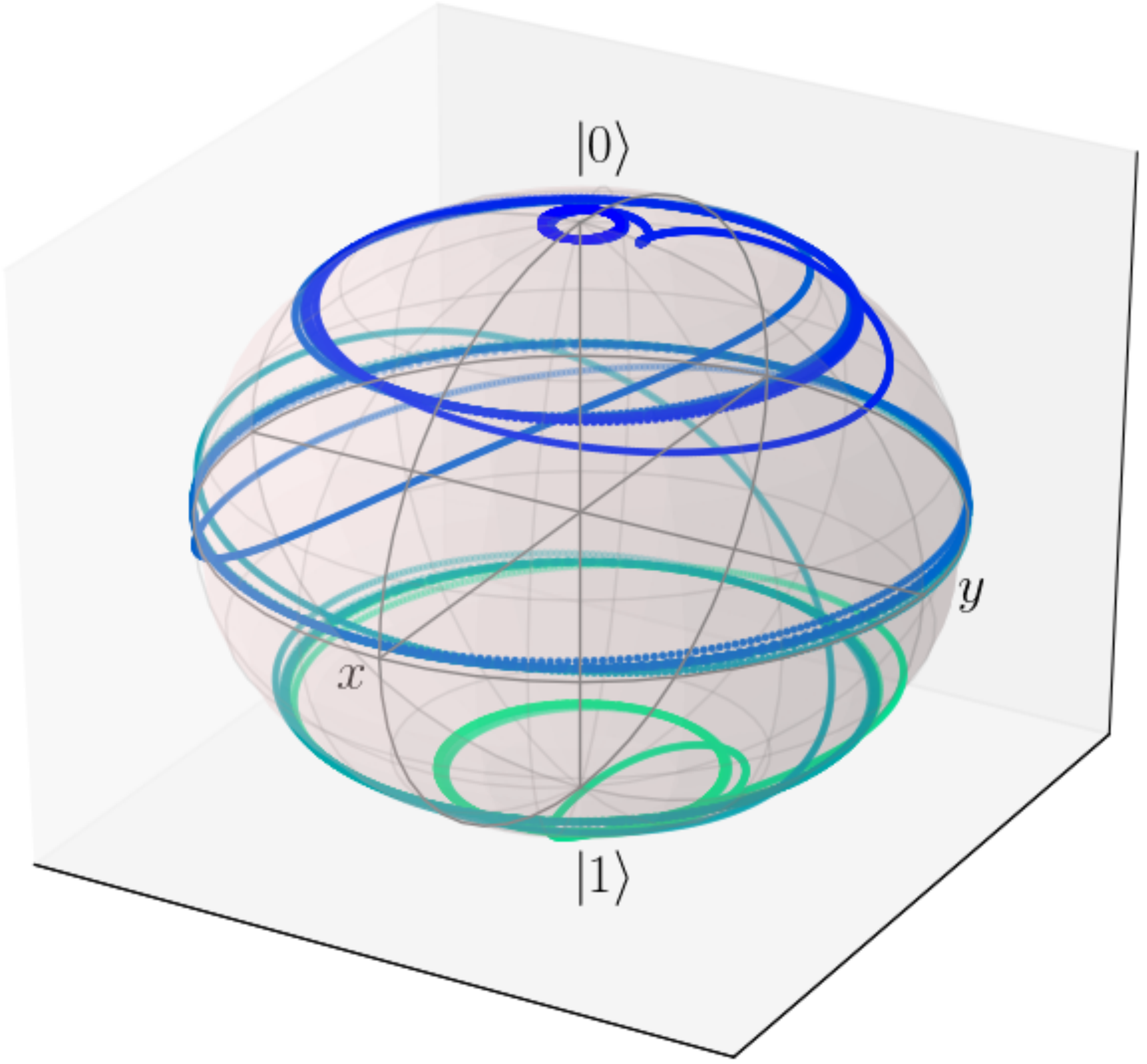
The **`Result`** class stores the expectation values of the operators passed to the solver.

```
result_ref = mesolve(H, rho0, times, c_ops, e_ops)
plot_expectation_values(result_ref, y_labels = "E[a'a], …,)
```

Simple Example: A driven, damped single mode cavity.

Solvers:

- **`mesolve`**: Lindblad master equations
- **`mcsolve`**: Monte-Carlo quantum trajectory
- **`floquet_modes`**: Floquet theory for strong driving
- **bloch_redfield:** Bloch-Redfield solver
- **`sesolve`, `ssesolve`,`smesolve`**: Stochastic solvers



Solving a many-qubit dynamics

Closed vs. Open dynamics

Schrödinger Equation

$$i\hbar \frac{d}{dt}|\psi\rangle = H|\psi\rangle$$

Master Equation

$$i\hbar \frac{d}{dt}\rho = [H, \rho] + \textbf{Noise}$$

	Schrödinger Equation	Lindblad Master Equation
Probability	Conserved	Conserved
Evolution	Unitary, $H = H^\dagger$	Non-Unitary
Time reversal	Preserved	Broken
Features	Closed system	**Open system**: noise and dissipation
Scaling with N qubits	2^N	

Solving a many-qubit dynamics

Exponential scaling even with Lindblad dynamics

$$\frac{d}{dt}\rho = -i[H,\rho] + \gamma \sum_{i}^{N} \left(L_i \rho L_i^\dagger - \frac{1}{2} L_i^\dagger L_i \rho - \frac{1}{2} \rho L_i^\dagger L_i \right)$$

	Von Neumann Equation	Lindblad Master Equation
Probability	Conserved	Conserved
Evolution	Unitary, $H = H^\dagger$	Non-Unitary
Features	Closed system	**Open system**: noise and dissipation
Scaling with N qubits	2^N	4^N

$$\frac{d}{dt}|\rho\rangle = \mathscr{L}|\rho\rangle$$

Solving a many-qubit dynamics: Quantum Trajectories

Photon counting statistics

Lindblad master equation

$$\frac{d}{dt}\rho = -\frac{i}{\hbar}[H,\rho] + \gamma\left(L\rho L^\dagger - \frac{1}{2}L^\dagger L\rho - \frac{1}{2}\rho L^\dagger L\right)$$

Solving a many-qubit dynamics: Quantum Trajectories

Photon counting statistics

$$\frac{d}{dt}\rho = -\frac{i}{\hbar}[H,\rho] + \gamma\left(\overset{\text{“quantum jump”}}{L\rho L^\dagger} - \underbrace{\frac{1}{2}L^\dagger L\rho - \frac{1}{2}\rho L^\dagger L}_{\text{continuous decay}}\right)$$

Effective Hamiltonian:
shrinking the state

$$H_{\text{eff}} = H_{\text{sys}} - \frac{i\hbar}{2}L^\dagger L$$

$$|\psi(t+\delta t)\rangle = \frac{L|\psi(t)\rangle}{\left\langle \psi(t) \middle| L^+ L \middle| \psi(t) \right\rangle^{1/2}}$$

Solving a many-qubit dynamics: Quantum Trajectories

Photon counting statistics

$$\frac{d}{dt}\rho = -\frac{i}{\hbar}[H,\rho] + \gamma\left(\overbrace{L\rho L^\dagger}^{\text{"quantum jump"}} - \underbrace{\frac{1}{2}L^\dagger L\rho - \frac{1}{2}\rho L^\dagger L}_{\text{continuous decay}}\right)$$

A. **Effective Hamiltonian**:
shrinking the state

$$H_{\text{eff}} = H_{\text{sys}} - \frac{i\hbar}{2}L^\dagger L$$

B. **Quantum jump:**

$$|\psi(t+\delta t)\rangle = \frac{L|\psi(t)\rangle}{\left\langle \psi(t)\middle| L^{+}L \middle| \psi(t)\right\rangle^{1/2}}$$

1. Generale random number, *r*.
2. Is *r* > norm(psi)?
   I. Yes: keep integrating (A)
   II. No: apply jump and renormalize

Explore this Physics with Interactive Notebooks

The **Qu**antum **T**oolbox **i**n **P**ython

QuTiP

You can find an interactive notebook at

https://github.com/nathanshammah/

Repository: **interactive-notebooks**

You can run the notebook live at

https://mybinder.org/v2/gh/nathanshammah/interactive-notebooks/binder

Take a snapshot

Thank you

@NathanShammah

GitHub: nathanshammah
LinkedIn: Nathan Shammah

Extra Slides

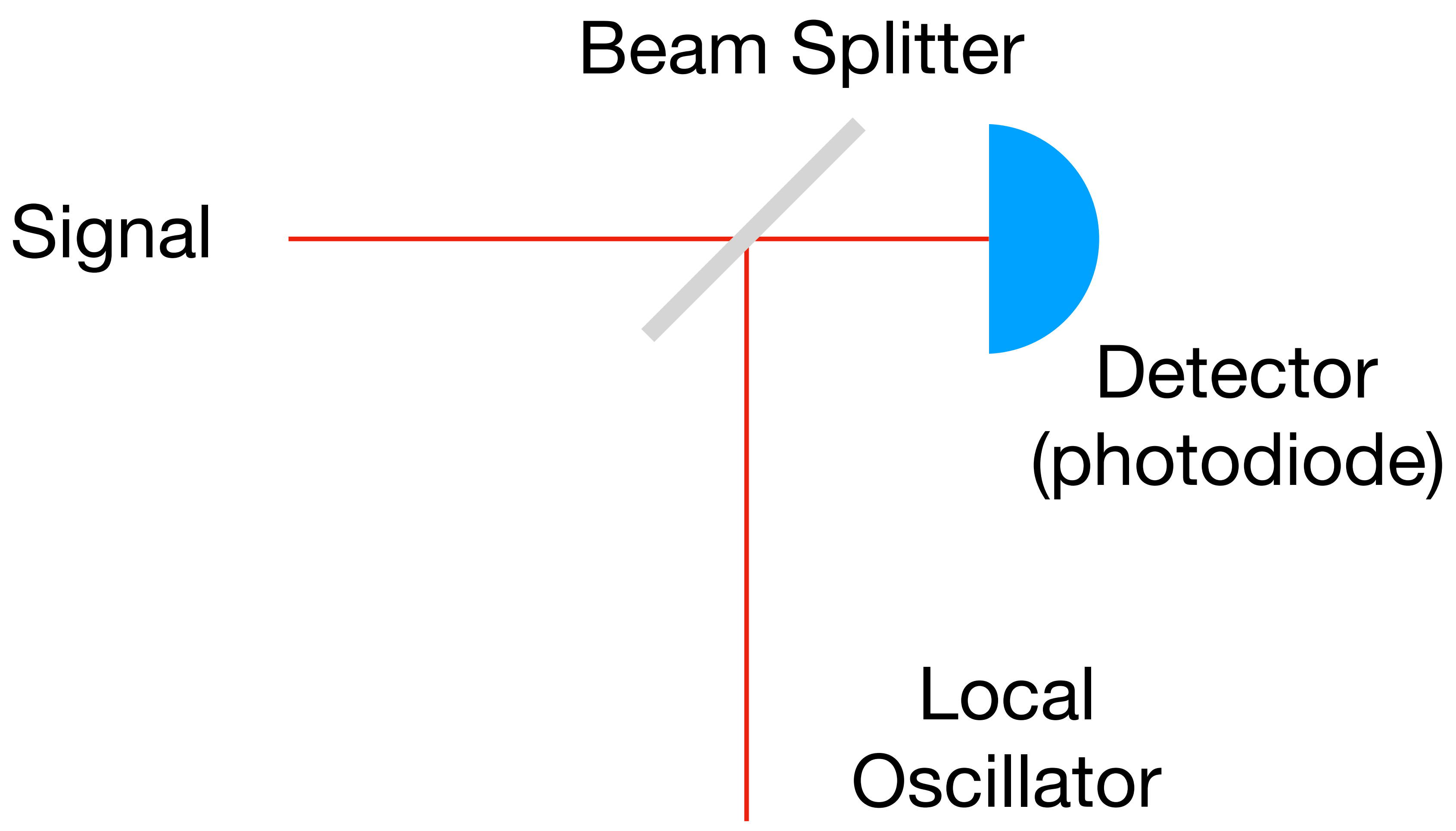
Stochastic solvers

Stochastic master equation: Continuous weak measurements

- continous weak measurements

Heterodyne Detection

$$\omega_{LO} \neq \omega_{signal}$$



Homodyne Detection

$$\omega_{LO} = \omega_{signal}$$

```
result = smesolve(H, rho0, times, [], c_ops, e_ops)
```

Stochastic solvers

Stochastic master equation: Continuous weak measurements

- continous weak measurements

Heterodyne Detection

$$d\rho(t) = -i[H,\rho(t)]dt + \gamma\mathcal{D}[a]\rho(t)dt + \frac{1}{\sqrt{2}}\overbrace{dW_1(t)\sqrt{\gamma}\mathcal{H}[a]\rho(t)}^{J_x(t)} + \frac{1}{\sqrt{2}}\overbrace{dW_2(t)\sqrt{\gamma}\mathcal{H}[-ia]\rho(t)}^{J_y(t)}$$

$$\mathcal{D}[A]\rho = A\rho A^\dagger - \frac{1}{2}\left(\rho A^\dagger A + A^\dagger A\rho\right)$$ **Lindblad term**

$$\mathcal{H}[A]\rho = A\rho + \rho A^\dagger - \mathrm{Tr}\left[A\rho + \rho A^\dagger\right]\rho$$ **Nonlinear term**

```
result = smesolve(H, rho0, times, [], c_ops, e_ops, method='heterodyne')
```

```
method = 'heterodyne'
method = 'homodyne'
method = 'photocurrent'
```