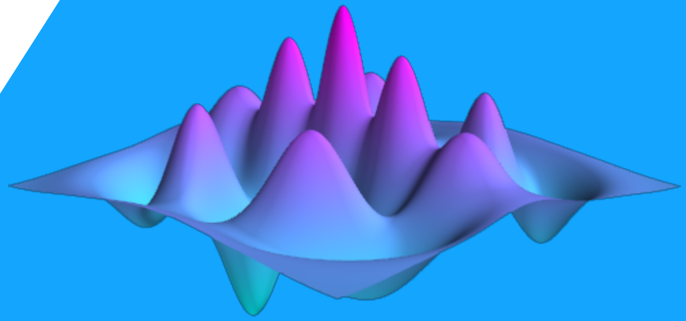
Make your code count

Quantum simulations and collaborative code

Shahnawaz Ahmed



QuTiP: The Quantum Toolbox in Python

About me

sahmed.in

2019

PhD @Applied Quantum Physics,
Chalmers University of Technology, Sweden
Anton Fisk Kockum, Prof. Göran Johansson

Quantum simulations,
Machine Learning

2018

Master’s thesis @Theoretical Quantum Physics Lab
Riken, Japan
Mauro Cirio, Neill Lambert, Prof. Franco Nori

Spin-boson model
Ultrastrong coupling

2017

Intern @Theoretical Quantum Physics Lab
Riken, Japan
Nathan Shammah, Clemens Gneiting, Prof. Franco Nori

Deep Learning

Collective effects in
large spin systems

2016

Intern @Google Summer of Code
Ariel Rokem, Eric Peterson, Rafael Henriques

Diffusion Imaging

Do you guys put “Quantum” in everything?

- Ant-man and the Wasp (2018)

- Quantum Machine Learning
- Quantum Big Data
- Quantum Neural Networks
- Quantum Cryptography
- Quantum Sensing (GPS)
- Quantum Internet

QUANTUM FLAGSHIP

QUTech

Centre for
Quantum
Technologies

WACQT

NASA

IBM Q

Wallenberg
Center for
Quantum
Technologies

Alibaba.com

XANADU

rigetti

DWave

IonQ

intel

TOSHIBA

quantumcomputingreport.com

Interests @ FOSDEM19

Photonic quantum computing

Standardization and availability of code

Spin systems, quantum annealing

Optimization and control

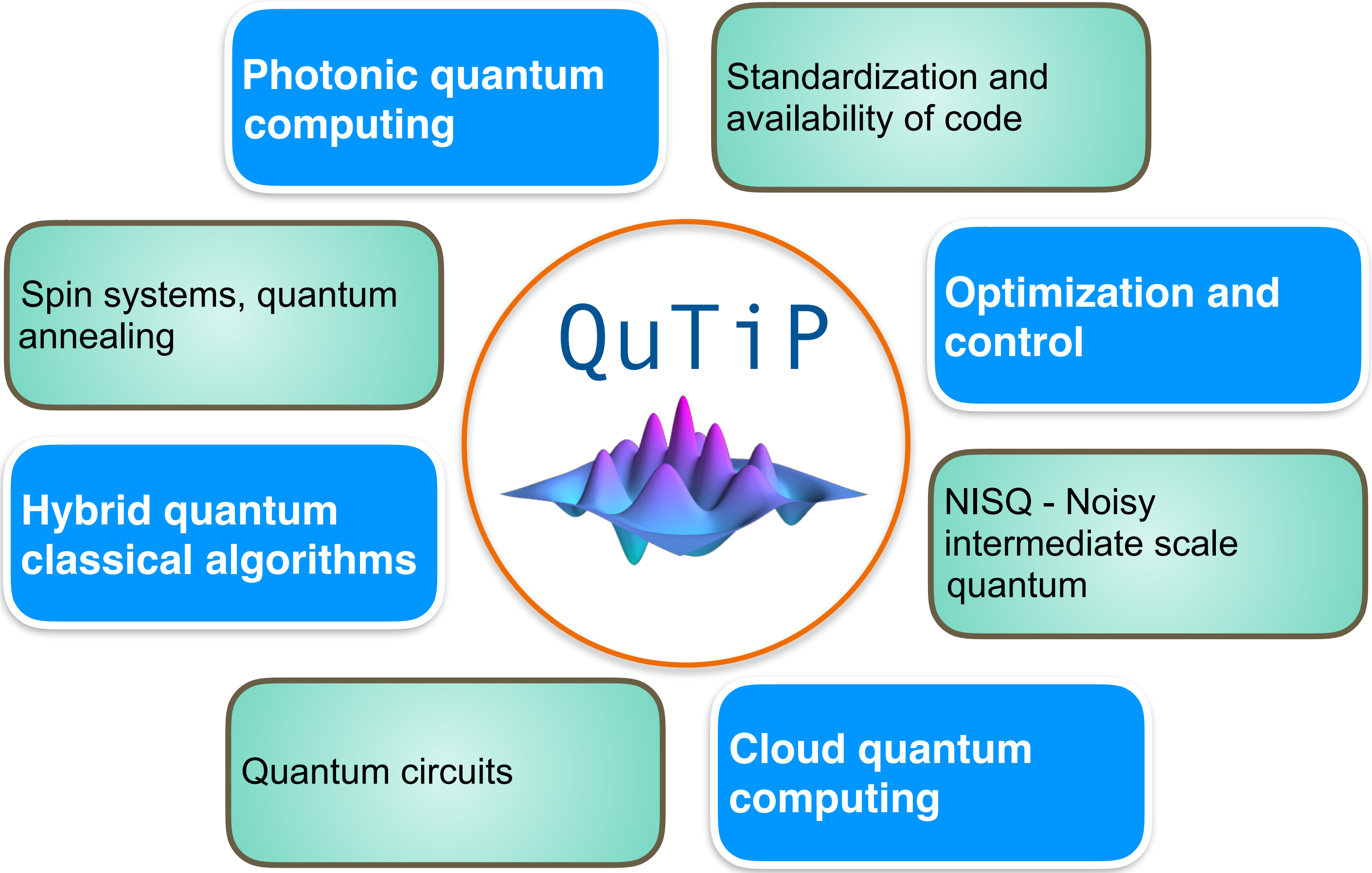
Hybrid quantum classical algorithms

NISQ - Noisy intermediate scale quantum

Quantum circuits

Cloud quantum computing

Quantum physics simulator



Photonic quantum computing

Standardization and availability of code

Spin systems, quantum annealing

QuTiP

Optimization and control

Hybrid quantum classical algorithms

NISQ - Noisy intermediate scale quantum

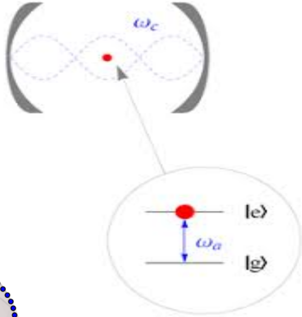
Quantum circuits

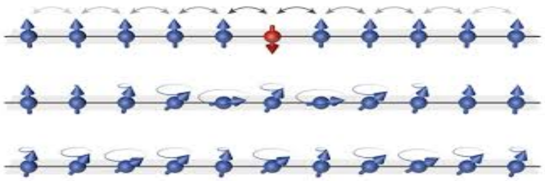
Cloud quantum computing

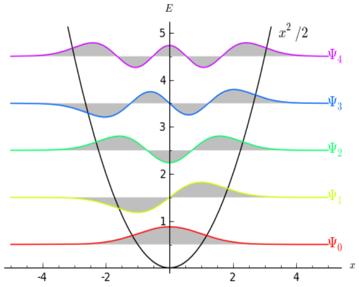
Quantum physics simulator

A collaborative effort over many years by the community

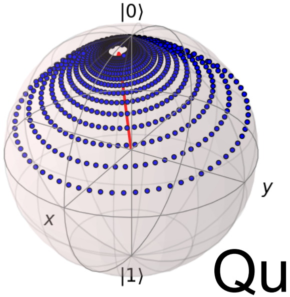
cQED







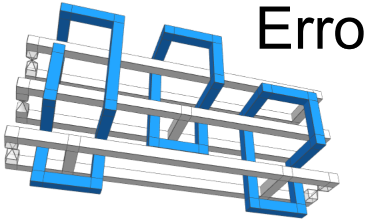
Condensed matter

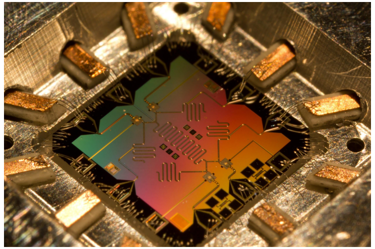


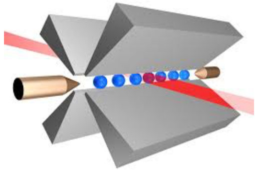
Quantum optics

QuTiP

Error correction

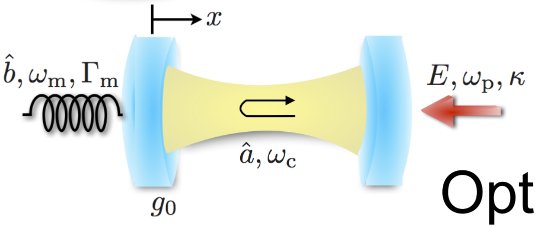






Ion Traps

Superconducting circuits


x
$\hat{b}, \omega_\mathrm{m}, \Gamma_\mathrm{m}$
$E, \omega_\mathrm{p}, \kappa$
$\hat{a}, \omega_\mathrm{c}$
g_0

Optomechanics

and more …

QuTiP: features at a glance

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

Built with Python

Python's **straightforward syntax** allows for constructing, manipulating, and evolving quantum objects using QuTiP with just a few lines of code. QuTiP is the ideal toolbox for research or the classroom.

Custom algorithms

QuTiP can determine if an operator is Hermitian without performing the conjugate transpose. This is just one of many custom algorithms devised to **maximize performance**. Sparse matrices deployment efficiently manipulates large datasets.

Fast

QuTiP is capable of leveraging the multiprocessing power inside every modern computer. QuTiP can take advantage of the Python **multiprocessing** library, OPENMP, SSE3 processor extensions, and Intel MKL.

Built-in solvers

A variety of built-in solvers allow the study of **dynamical simulations** and steady-state analysis. In addition to Lindblad and Monte Carlo solvers, QuTiP offers advanced routines for Bloch-Redfield and Floquet formalism, and non-Markovian systems.

C++ performance

A wide range of time-dependent evolution simulations can be runtime compiled into C++ behind the scenes using Cython. The ease of use of Python is boosted by **compiled code**.

Experimental Data

If you need to construct a function from a data set, QuTiP allows for passing **interpolating functions** as time-dependent arguments to the evolution solvers, also runtime compiling into C++.

Ad-hoc visualization tools

From Bloch spheres to nonlinear colormaps for Wigner functions, QuTiP includes a host of **built-in visualization routines** that help bring data to life, including through animations and 3D graphics.

Independent testing

QuTiP is thoroughly tested, both by its thousands of users, and by a large collection of **built-in test scripts** independently run by Travis CI. Over a thousand such tests help cover nearly all of the built-in functions, continuously running in development.

User friendly

No software should be a black box to the user, especially in science. QuTiP includes hundreds of pages of **documentation**, a multitude of **tutorial** Jupyter notebooks, and a friendly community of users who help answer questions.

Open source quantum (2016 -)

2016	QETLAB	Matlab	University of Waterloo,Canada
2016	Liquil>	F#	Microsoft
2016	Quantum Fog	**Python**	Artiste-qb
2016	Qubiter	**Python**	Artiste-qb
2016	IBM Q Experience	-	IBM
2017	ProjectQ	**Python**	ETH Zurich
2017	Forest (QUIL)	**Python**	Rigetti
2017	QISKit	**Python**	IBM
2017	Quantum Optics.jl	Julia	Universität Innsbruck
2017	PsiQuaSP	C++.	Gegg M, Richter M
2018	Strawberry Fields	**Python**	Xanadu, Canada
2018	Quantum Dev Kit	Q#.	Microsoft
2018	QCGPU	Rust, OpenCl	Adam Kelly
2018	NetKet	C++	The Simons Foundation
2018	OpenFermion	**Python**	Google, Harvard,UMich, ETH ..

https://github.com/markf94/os_quantum_software

QuTiP - IMPACT

Physical Review Letters
moving physics forward

computers

AIP Advances

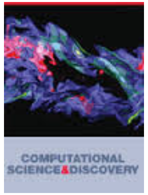
nature communications

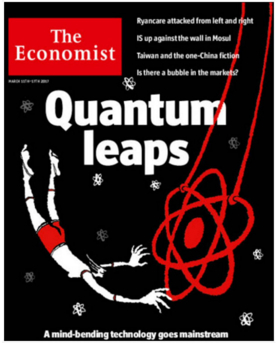
nature physics

The Journal of Physical Chemistry Letters

European Journal of Physics

Journal of Modern Optics


COMPUTATIONAL SCIENCE & DISCOVERY


The Economist

Ryancare attacked from left and right

IS up against the wall in Mosul

Taiwan and the one-China fiction

Is there a bubble in the markets?

Quantum leaps

A mind-bending technology goes mainstream

Google

NORTHROP GRUMMAN

IBM

Alibaba.com

intel

Microsoft

amazon.com

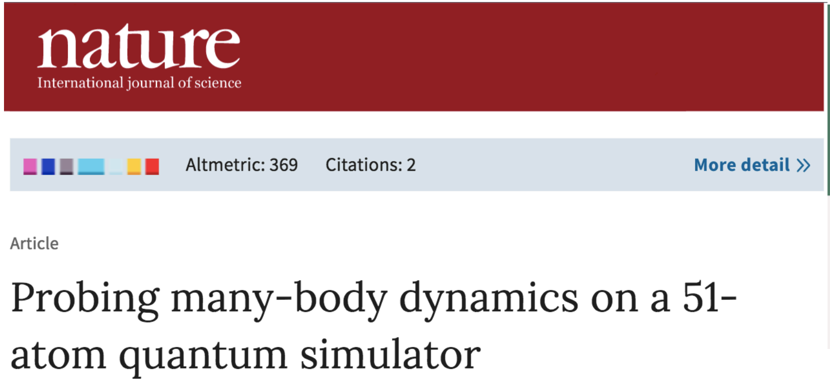
Honeywell

NASA

IONQ

rigetti

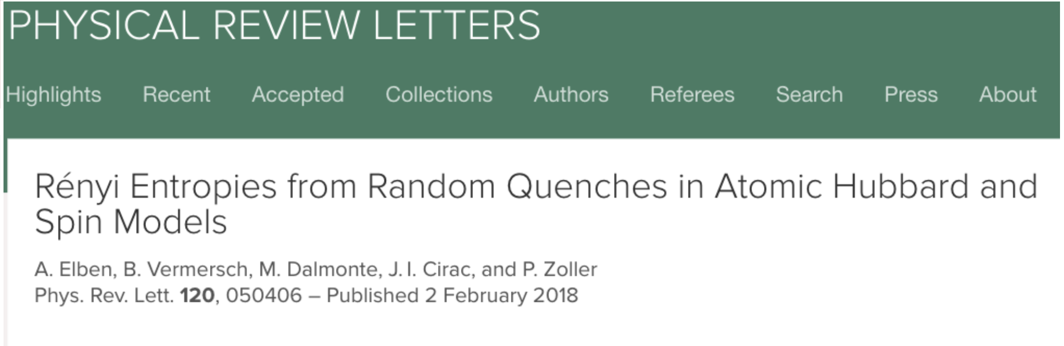
- RESEARCH
- EDUCATION
- INDUSTRY


nature
International journal of science

Altmetric: 369 Citations: 2 More detail »

Article

Probing many-body dynamics on a 51-atom quantum simulator


PHYSICAL REVIEW LETTERS

Highlights Recent Accepted Collections Authors Referees Search Press About

Rényi Entropies from Random Quenches in Atomic Hubbard and Spin Models

A. Elben, B. Vermersch, M. Dalmonte, J. I. Cirac, and P. Zoller
Phys. Rev. Lett. **120**, 050406 – Published 2 February 2018

Open source

Impact

Easier to understand and develop an idea with good code and implementation, wider visibility and impact. (eg., PIQS)

Reproducibility

Faster reproduction of results and application to new problems, data and ideas. (eg., SciNet, Neural ODE, QGAN)

Collaborations and feedback

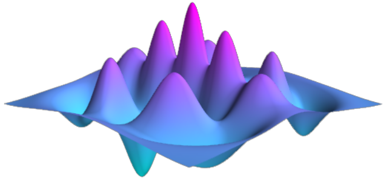
Combine efforts and expertise of a wide range of people without barriers. Get feedback, bug reports, suggestions from users.

Paper to production

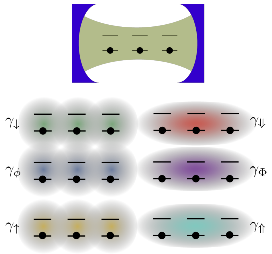
Stable software implementations can be converted to applications faster. (eg., Tensor Flow, PyTorch, Scikit-learn)

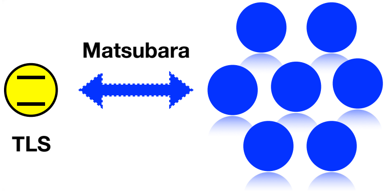
Talk outline

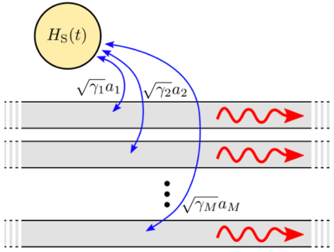
QuTiP

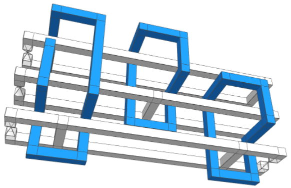


Open source









- Quantum
- QuTiP intro
- Whats new?

GitHub

SPHINX Python Documentation Generator

Travis CI

Read the Docs

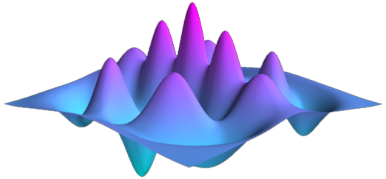
CONDA

zenodo

python™ Package Index

- Open source?
- GSoC 2019

Quantum physics
A brief introduction



Superposition

Why?

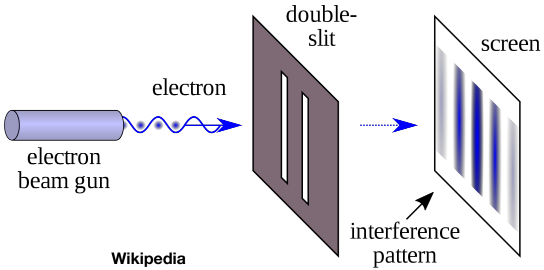
Quantum mechanics describes realities in terms of probability wave functions.

$$0 \rightarrow \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

Associate wave-like nature to electrons.

Probability wave function to describe states.

Double slit (electrons)



Wikipedia

Waves interfere

Superposition

Why?

$$0 \rightarrow \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

Quantum mechanics describes realities in terms of probability wave functions.

Dr. Dan Russell, Grad. Prog. Acoustics, Penn State

Wave amplitudes add up

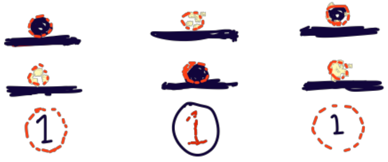
One of the most successful theories out there.

The experiments!

Shut up and calculate

Exponential power of quantum superpositions!(?)

Three qubits can be a superposition of all eight possibilities



0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

One shot application of a function to all possible data.
Seemingly massive parallelization!

Explore all possibilities simultaneously.

$$N \sim 2^N$$

$$\frac{|000\rangle + |001\rangle + \ldots + |111\rangle}{\sqrt{8}}$$

$$\swarrow$$

$$|010\rangle$$

But we see only 1s and 0s when we look!

The measurement problem.

Measurement and reality

Measurement can change a quantum state, collapse it to one of the possibilities.

Copenhagen interpretation

$$\frac{|000\rangle + |001\rangle + \ldots + |111\rangle}{\sqrt{8}}$$

$$|010\rangle$$

- Determining an unknown quantum state is tricky. Measurement collapses wave function. **Only a probable answer to the computation.**

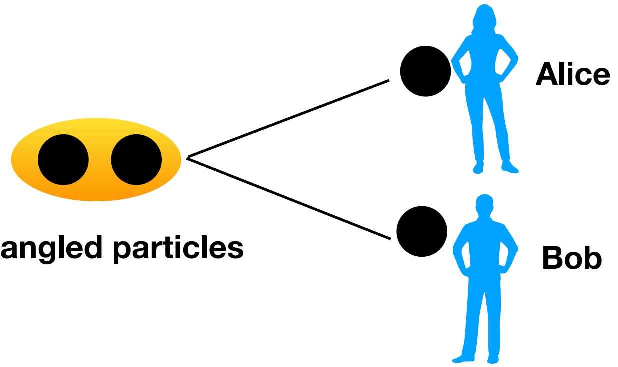
- **Repeating measurement by making copies is not possible** due to no-cloning theorem. Repeat experiment on identically prepared qubits and perform multiple measurements in the end to get the result.

Nature only reveals quantum nature through statistics.

RETHINK

Noise
Error correction
Verification

Entanglement



Entangled particles

Correlations between parts

$$|01\rangle + |10\rangle$$

Measurement Basis

z : Up/Down ↓

x: Left/Right →

If they measure the same property (basis), they get correlated result. Otherwise, random. Measuring parts, collapses the results.

Spooky action at a distance - EPR paradox, Bell inequalities

Spooky! and faster than light communication? (NOT)

Entanglement is a resource: Dense coding, teleportation, quantum key sharing.

What can we do with all that quantum?

- Core idea (Feynman): Simulate quantum physics, atomic and molecular interactions.
- Speed-ups for some problems in Computer Science and Mathematics
    - **Integer factorisation (Shor). Break RSA.**
    - **Grover's search (Brute force search in unstructured databases).**
    - **Random sampling of quantum circuits, Boson sampling …**
- Optimization and Machine Learning.
- Quantum chemistry, drug design and studying complex biological datasets, protein folding.
- Quantum pattern recognition.

Long way to go. But we can still do interesting things with our computers.

QuTiP

Code

QuTiP speaks quantum

Quantum state vectors

$$|\psi\rangle = \frac{1}{\sqrt{(2)}}(|0\rangle + |1\rangle)$$

$$H = \frac{\sigma_z}{2}$$

Operators and Hamiltonians

$$a, a^\dagger$$

Tensors

$$\sigma_z \bigotimes \sigma_x$$

Density matrices

$$\rho$$

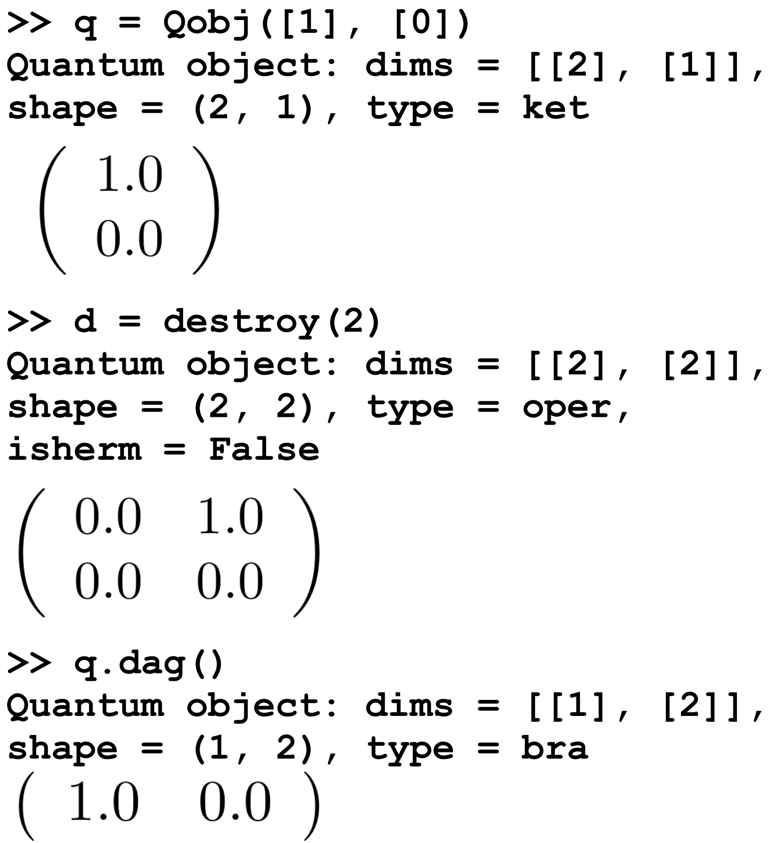
```
>> from qutip import *

>> psi1 = basis(2, 0)
>> psi2 = basis(2, 1)
>> psi = (psi1 + psi2)/1.414

>> H = sigmaz()/2

>> a = destroy(2)


>> tensor(rho1, rho2)
>> tensor(sigmaz(), sigmax())

>> rho = ket2dm(psi)
>> op = vector_to_operator(rho)
```

The Qobj class

- State and operators are declared as Qobj

- Algebra (bosonic)

- Sparse CSR matrices which interact with specialized Cython enhanced code.



```
>> 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
```

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

Operators

Operators	Command (# means optional)	Inputs
Charge operator	`charge(N, M=-N)`	Diagonal operator with entries from M … 0 … N.
Commutator	`commutator(A, B, kind)`	kind = ‘normal’ or ‘anti’ commutator.
Diagonals operator	`qdiags(N)`	Quantum object created from arrays of diagonals at given offsets.
Higher spin operator	`jmat(j,#s)`	j = integer or half-integer representing spin, s = ‘x’, ‘y’, ‘z’, ‘+’, or ‘-’
Identity	`qeye(N)`	N = number of levels in Hilbert space.
Destruction operator	`destroy(N)`	same as above
Momentum operator	`momentum(N)`	same as above
Number operator	`num(N)`	same as above

Methods on Qobj

Command	Description
`Q.check_herm()`	Check if quantum object is Hermitian
`Q.conj()`	Conjugate of quantum object.
`Q.dag()`	Returns adjoint (dagger) of object.
`Q.diag()`	Returns the diagonal elements.
`Q.eigenenergies()`	Eigenenergies (values) of operator.
`Q.eigenstates()`	Returns eigenvalues and eigenvectors.
`Q.groundstate()`	Eigenval & eigket of Qobj groundstate.
`Q.matrix_element(bra,ket)`	Matrix element <bra|Q|ket>
`Q.norm()`	Returns L2 norm for states, trace norm for operators.

Superoperators, maps, tensors

- **spre** and **spost** $X\rho X^\dagger$
- Multipartite systems, tensors, two coupled qubits, qubit coupled to an oscillator

```
>> tensor(sigmax(), sigmax())
>> tensor(psi1, psi2)
>> tensor(rho1, rho2)
```

- Composite Hamiltonians, Liouvillians, and super tensors

```
>> H_sys = sigmaz()/2
>> H_cav = destroy(5)
>> H_comp = tensor(H_sys, H_cav)
>> super_tensor(L1, L2)
```

Contractions, exotic maps

```
>> X = sigmax()
>> spre(X)*spost(X.dag())
… type = super, isherm = True
Qobj data =
[[0. 0. 0. 1.]
 [0. 0. 1. 0.]
 [0. 1. 0. 0.]
 [1. 0. 0. 0.]]
```

```
>> H = sigmaz()
>> c_ops = [sigmax(), sigmay()]
>> liouvillian(H, c_ops)
.. type = super, isherm = False
Qobj data =
[[-2.+0.j  0.+0.j  0.+0.j  2.+0.j]
 [ 0.+0.j -2.+2.j  0.+0.j  0.+0.j]
 [ 0.+0.j  0.+0.j -2.-2.j  0.+0.j]
 [ 2.+0.j  0.+0.j  0.+0.j -2.+0.j]]
```

Gates as Quantum Objects

- All valid Qobj methods work directly. Eg: Make a super-operator out of Toffoli

- Quickly move from the picture of circuits and gates to Hamiltonian evolution and use all of the QuTiP machinery

- Noise, quantum control, stochastic evolution

From circuit to physics

```
# Make a super operator out of a gate
>> spre(toffoli())
Quantum object: dims = [[[2, 2, 2], [2,
2, 2]], [[2, 2, 2], [2, 2, 2]]], shape =
(64, 64), type = super, isherm = True
```

$$\begin{pmatrix}
1.0 & 0.0 & 0.0 & 0.0 & 0.0 & \cdots & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 1.0 & 0.0 & 0.0 & 0.0 & \cdots & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 1.0 & 0.0 & 0.0 & \cdots & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 1.0 & 0.0 & \cdots & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 1.0 & \cdots & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & \cdots & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & \cdots & 0.0 & 1.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & \cdots & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & \cdots & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & \cdots & 0.0 & 0.0 & 0.0 & 1.0 & 0.0
\end{pmatrix}$$

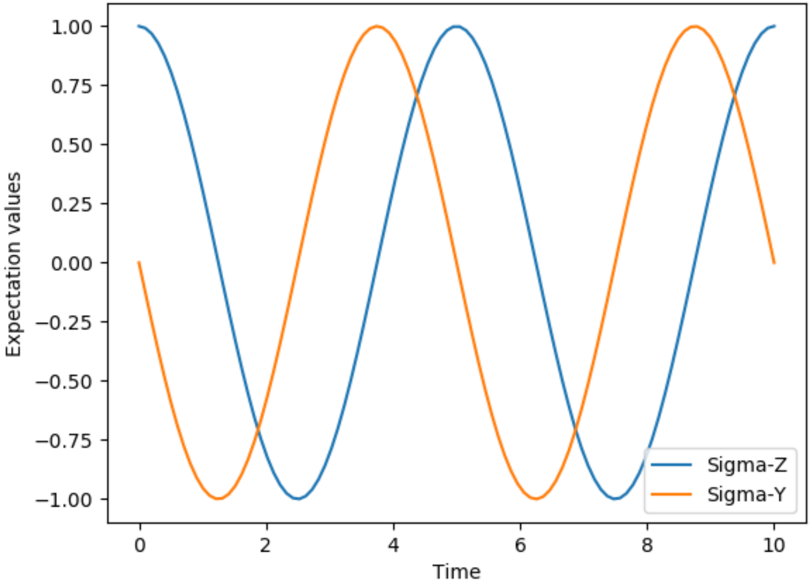
```
>> toffoli_super = spre(toffoli())
>> toffoli_super.iscp
False
```

QuTiP

Quantum dynamics

Lindblad master equation solver

- Typical workflow



$$\dot{\rho} = -i[H,\rho] + \sum_i C_i \rho C_i^\dagger - \frac{1}{2}\left\{C_i C_i^\dagger, \rho\right\}$$

```
>> H = sigmaz()
>> c_ops = [sigmam()]
>> psi0 = basis(2, 0)
>> times = np.linspace(0.0, 10.0, 20.0)
>> result = mesolve(H, psi0, times, c_ops)
```

Visualization

```
# Plot results using Matplotlib
>> plt.plot(result.times,
            result.expect[0])
>> plt.plot(result.times,
            result.expect[1])
```

Time dependent Hamiltonians

- QuTiP handles time-dependent Hamiltonians with ease
- Time-dependent Hamiltonians arise in control problems or driven systems.
- Smart built in solvers such as Floquet formalism: Evolve the wave function "**stroboscopically**", i.e., only evaluating at time multiples of the driving period.

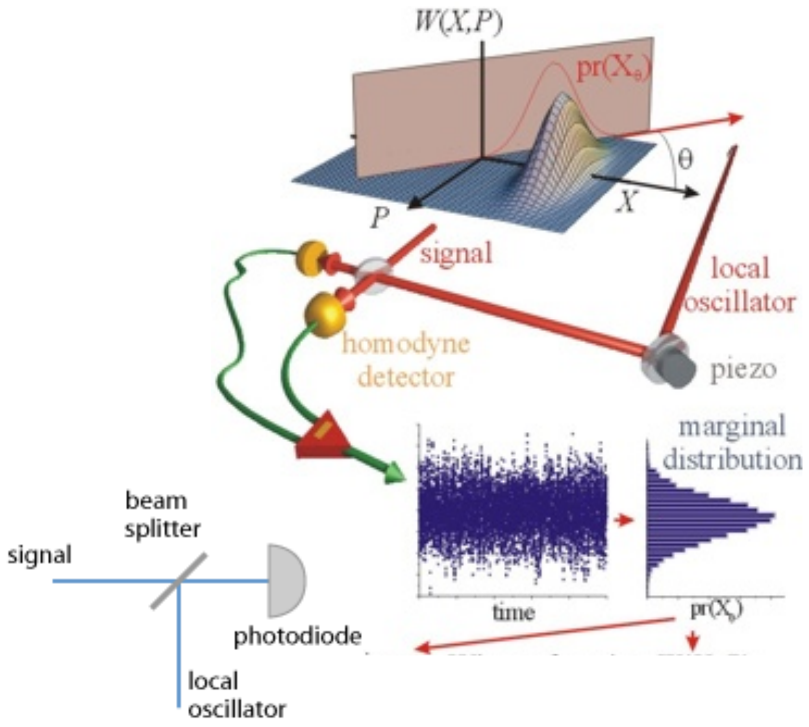
```
>> def f(t, args):
..     return np.exp(-args[0]*t)

>> H0 = sigmax()/2
>> H1 = sigmaz()/2
>> H = [H0,[H1, f]]
```

$$H = H_0 - f(t)H_1$$

```
>> mesolve(H, …)
>> floquet_markov_mesolve
```

Stochastic Master Equations

- **Quantum stochastic calculus** is the non-commutative analogue of Ito’s calculus, developed to study noise in open quantum systems. (Barichielli, 1990)
- Useful tool to model **continuous weak measurements, and implement feedback-control methods**
- **For example:** Weak continuous **Heterodyne** and **Homodyne** measurement techniques (used to extract quadrature information with photon counters)



RP Photonics

Other master equation solvers

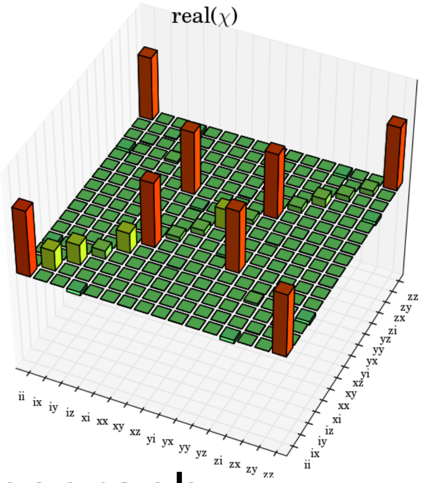
$$\dot{\rho} = -\frac{i}{\hbar}[H, \rho] + \sum_{I=1}^{N^2-1} \gamma_i \left(\mathcal{L}_i \rho \mathcal{L}_i^\dagger - \frac{1}{2} \mathcal{L}_i \mathcal{L}_i^\dagger, \rho \right)$$

```
>> H = sigmaz()
>> c_ops = [sigmam()]
>> psi0 = basis(2, 0)
>> times = np.linspace(0.0, 10.0, 20.0)
>> result = mesolve(H, psi0, times, c_ops)
```

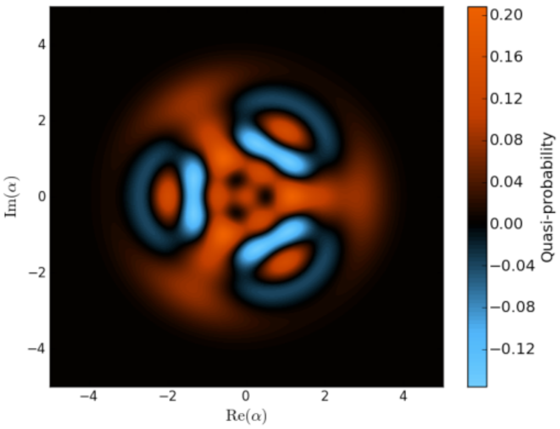
`mcsolve`	Monte Carlo	`HsolverDL`	Hierarchy
`fmmesolve`	Floquet-Markov	`ssesolve`	Stochastic
`rcsolve`	Reaction coordinate	`brmesolve`	Bloch redfield

… and more

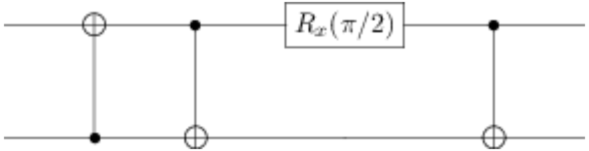
Visualisation



Tomography

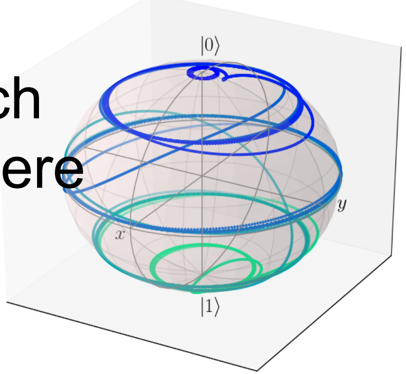


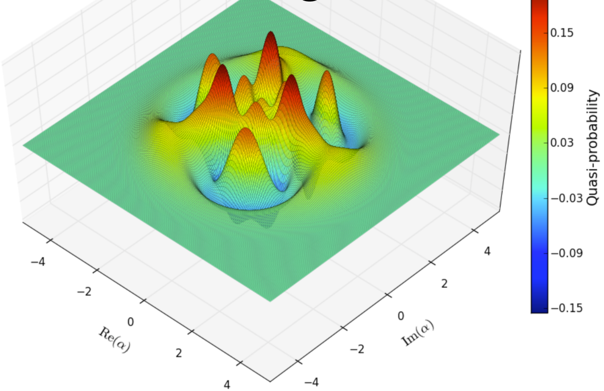
Wigner

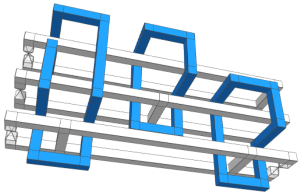


Circuits

Bloch Sphere

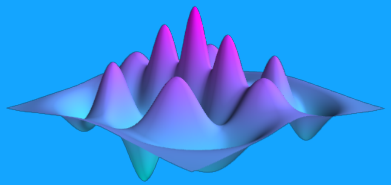




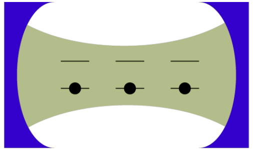


Surface code
Topological Circuits

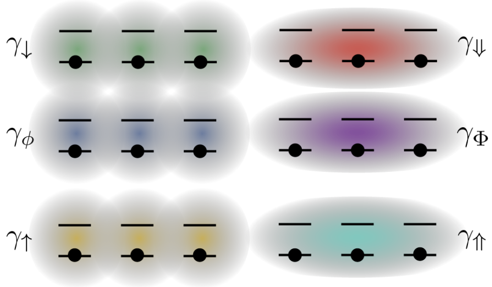
QuTiP: What’s new?



PIQS: simulating qubit ensembles



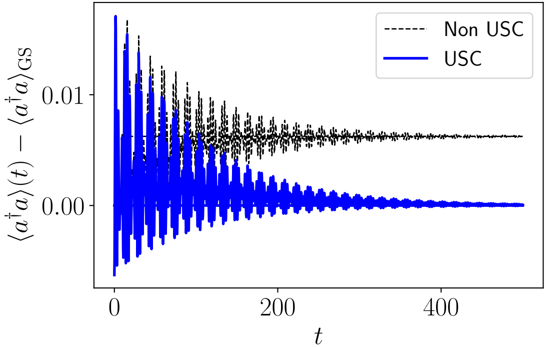
- PIQS (Permutational Invariant Quantum Solver)
  Nathan Shammah, Shahnawaz Ahmed



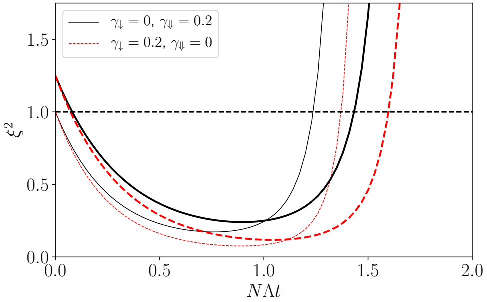
  - Noisy driven dissipative open quantum systems
  - Collective effects in qubit ensembles (100)

Shammah, N., Ahmed, S., Lambert, N., De Liberato, S., & Nori, F. (2018). Open quantum systems with local and collective incoherent processes: Efficient numerical simulation using permutational invariance. arXiv preprint arXiv: 1805.05129.

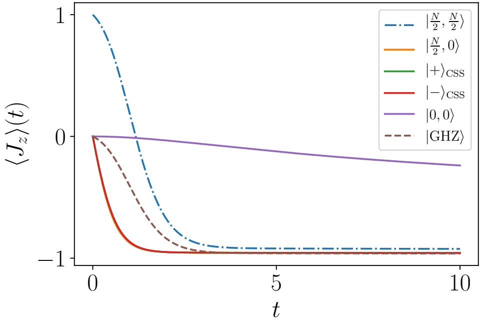
Ultrastrong Coupling



Spin Squeezing

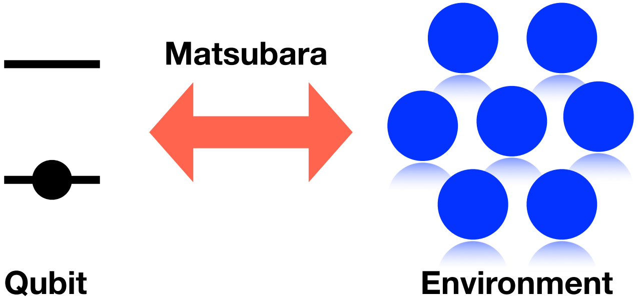


Superradiance



Non-Markovian methods, virtual photons

- A generic Hierarchical Equations of Motion (HEOM) implementation.
  Neill Lambert, Shahnawaz Ahmed
  - Non-Markovian dynamics: **Environment has a memory**
  - Ultrastrong coupling regime of light and matter
  - Bound states and virtual photons



A. Fruchtman, N. Lambert, and E.M. Gauger, Scientific Reports, **6** 28204 (2016).
When do perturbative approaches accurately capture the dynamics of complex quantum systems?

Scattering

qutip.org/tutorials

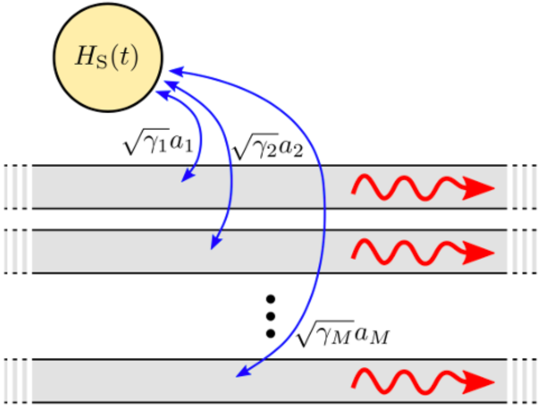
- Photon scattering in Quantum Optical Systems
  Ben Bartlett, P.h.D student, Stanford University

Email: benbartlett@stanford.edu
Github: bencbartlett

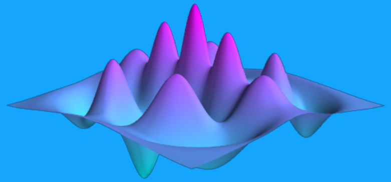
- Photon scattering
- Multiple waveguides: SPDC, Photon emission

K.A. Fischer, et.al. (2017), "Scattering of Coherent Pulses from Quantum-Optical Systems" (arXiv: 1710.02875)

"How do photons scatter into the waveguide when the system is driven with some excitation field?"



Open source and open science



How QuTiP uses open source

Code, Testing

GitHub

Travis CI

Documentation

SPHINX
Python Documentation Generator

Read the Docs

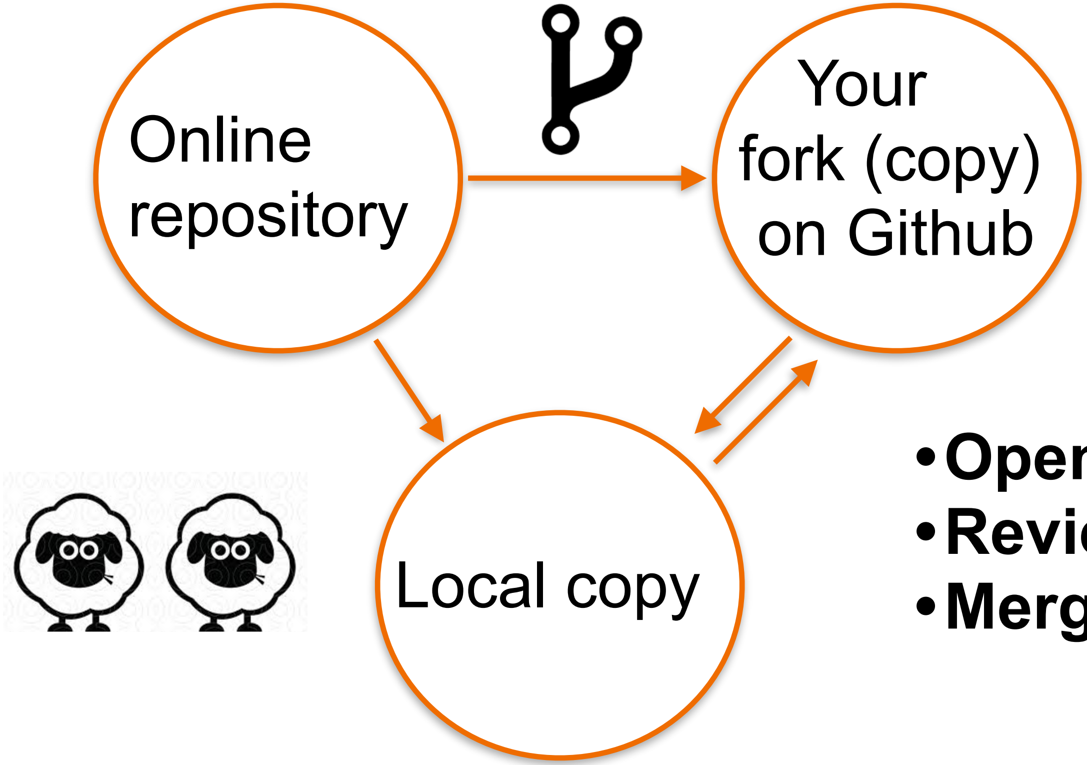
Publish

CONDA

python™
Package
Index

zenodo

Github



Online repository

Your fork (copy) on Github

Local copy

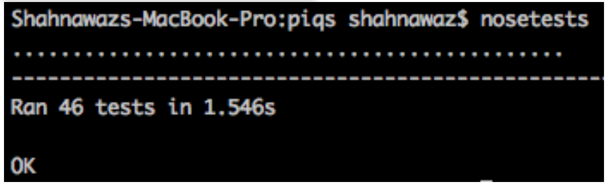
- **Open a Pull request**
- **Review by others**
- **Merge**

Testing

“Untested code is broken code”

```
def test_trace():
     '''
     Tests the calculation of trace
      '''
      …
     calculated = trace(mat)
     assert_(calculated, 1.)
```

```
>> nosetests
```



```
Shahnawazs-MacBook-Pro:piqs shahnawaz$ nosetests
..............................................
----------------------------------------------------------------------
Ran 46 tests in 1.546s

OK
```

```
dist: trusty
language: python
```

✓

Write a .travis.yml file

Travis with Github

Automated testing online

Documentation

Auto-generate

```
>> sphinx-quickstart
```

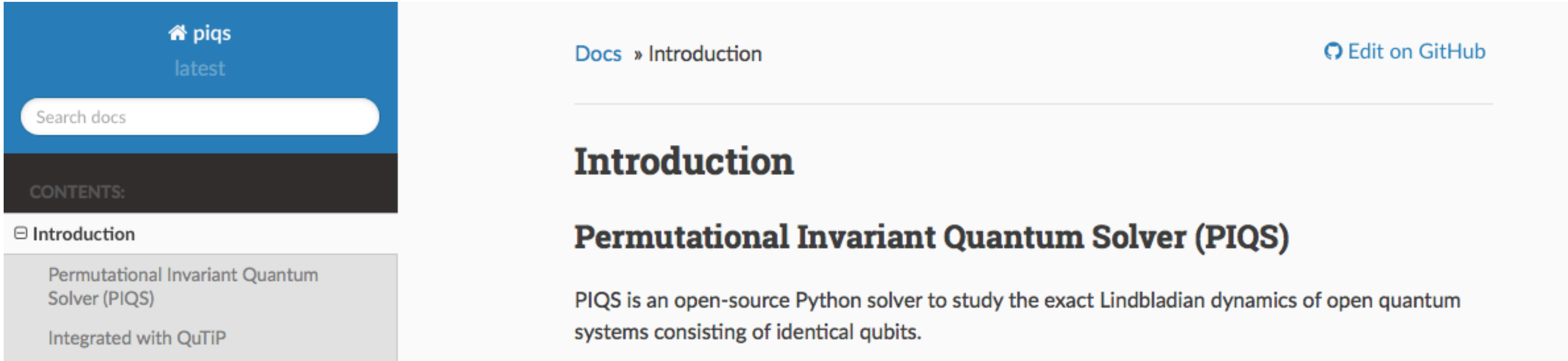
Edit conf

```
doc/source/conf.py

# -- Project information
project = 'piqs'
```

Generate doc

```
>> make html
```

piqs.readthedocs.com



Introduction

Permutational Invariant Quantum Solver (PIQS)

PIQS is an open-source Python solver to study the exact Lindbladian dynamics of open quantum systems consisting of identical qubits.

Publishing and distributing code

setup.py

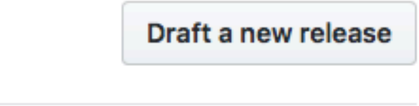
meta.yml

zenodo Code/Data

DOI 10.5281/zenodo.1212802

python™ Package Index

CONDA



GitHub

```
>> pip install qutip
>> conda -c conda-forge install qutip
>> python setup.py install/develop
```

QuTiP: summing up

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

Project Impact

QuTiP
Quantum Toolbox in Python

>**600** citations (Google Scholar)

downloads 43k (conda forge)

More info at http://qutip.org/

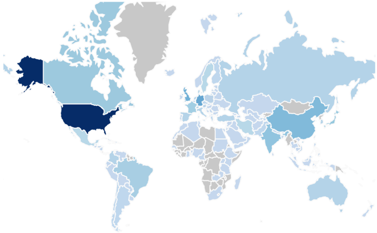
Authors

Comp. Phys. Comm. 183, 1760–1772 (2012); ibid. 184, 1234 (2013).

Code

Users

Distribution of 25k website visitors (2016)



Timeline:
Inspired by the Quantum Toolbox in MatLAB.

2011-2012: **QuTiP** 1.0

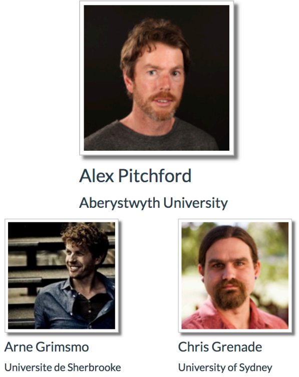
Aug 2015: 100 citations

Aug 2016: 200 citations

Jan 2017: QuTiP 4.0

July 2018: QuTiP 4.3

Lead Developers



Contributing Developers

- Neill Lambert (RIKEN)
- Denis Vasilyev (Leibniz)
- Kevin Fischer (Stanford)
- Jonathan Zoller (Ulm University)
- Ben Criger (RWTH Aachen)
- …
- Eric Giguere
- Shahnawaz Ahmed (Chalmers)
- Nathan Shammah (RIKEN)

- **GitHub**: 44 contributors, 4k commits

License: BSD
Style: PEP8 compliant

Libraries used:
- Scipy
- NumPy
- Cython
- Matplotlib
- SymPy

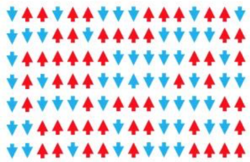
- Jupyter notebooks
- Online documentation
- Independent testing

Google summer of Code 2019

We are applying to Google Summer of Code 2019 with NumFocus!

https://github.com/qutip/qutip/wiki//Google-Summer-of-Code-2019

- Lattice models in QuTiP. Ising, Hubbard, XY, Heisenberg model
  - Clemens Gneiting (Riken, Japan)
  - Eric Giguere (Université de Sherbrooke)



- GPU backend for dynamics with the Hierarchical Eq of Motion
  - Neill Lambert (Riken, Japan)
  - Alex Pitchford (Université de Sherbrooke)
- An overhaul and abstraction of the quantum object class
  - Alex Pitchford (Université de Sherbrooke)
  - Eric Giguère (Université de Sherbrooke, UK)

Google
Summer of Code

NUMFOCUS
OPEN CODE = BETTER SCIENCE

medium.com/quantum-tech

Go to menti.com and use code **54 73 02**

Thank you

sahmed.in

Twitter @quantshah