Open-source software tools in quantum science

Quantum (circuit) simulation with QuTiP and quantum error mitigation with Mitiq

Galileo Galilei Institute – Summer School on Quantum Computing and Sensing – June 29th 2021

Nathan Shammah
nathan@unitary.fund

Unitary Fund

Creating a quantum technology ecosystem faster, better, and to benefit everyone.

Visiting:

UNIVERSITÀ DEGLI STUDI DI MILANO

RIKEN

SQMS

sqms.fnal.gov

Open-source software tools in quantum science: Part 1

Make your code count: Leveraging open-source tools in quantum technology

Galileo Galilei Institute – Summer School on Quantum Computing and Sensing – June 22nd, 2021

Nathan Shammah
nathan@unitary.fund

Creating a quantum technology ecosystem faster, better, and to benefit everyone.

Visiting:

UNIVERSITÀ DEGLI STUDI DI MILANO

sqms.fnal.gov

Open-source software tools in quantum science: Seminar 1

Make your code count

A Guide to building your open-source scientific computing project in Python.

SUPPORTED BY UNITARY FUND

A cheatsheet to develop a scientific open-source library from scratch.

Shahnawaz Ahmed
Chalmers, Sweden
(RIKEN, Japan)

0: Open
1: Code
2: Develop

3: Test
4: Pack
5: Document

6: Distribute
7: Publish
8: Host

Open, Code, Develop

python git GitLab GitHub

Test, Pack, Document

Travis CI SPHINX Python Documentation Generator Read the Docs

Distribute, Publish, Host

python Package Index CONDA zenodo binder

https://github.com/nathanshammah/make-your-code-count

Outline

- About Unitary Fund
  - Community activities
  - In-house research
- QuTiP
  - QuTiP overview
  - New features & updates
- Mitiq
  - About quantum error mitigation
  - Mitiq overview

Unitary Fund

Unitary Fund

Non-profit helping create a quantum technology ecosystem faster, better, and to benefit everyone.

Our supporters

Our partners

https://unitary.fund/

Unitary Fund Team

Team

Will Zeng, PhD
President. Head of Quantum Research at Goldman Sachs. Fmr. product/sw lead at Rigetti. Oxford quantum algorithms PhD.

Sarah Kaiser, PhD
Community Manager. Co-founder of Q# Community, Microsoft MVP, activist for open source and diversity in quantum. UWaterloo PhD in quantum computing.

Ryan LaRose
NASA Fellowship PhD student at University of Michigan. Fmr at Alphabet X. Wrote first paper benchmarking quantum software packages.

Nathan Shammah, PhD
CTO. Lead developer at QuTiP. Visiting scientist at RIKEN. PhD in quantum physics from Univ. of Southampton.

Andrea Mari, PhD
> 40 peer reviewed scientific publications. Contributor to Pennylane. Fmr. researcher at Xanadu. Univ. of Postdam PhD in quantum information.

Dan Strano
Full stack web engineer. Lead developer on the qrack quantum simulator.

Vincent Russo, PhD
Post-quantum security developer at ISARA. Lead dev on toqito quantum info package. PhD from UWaterloo.

Advisory Board

15 volunteer experts in quantum systems & software from:

IBM, XANADU, Microsoft, University of Waterloo, Oak Ridge National Laboratory, ZAPATA, qosf, Harvard, USC

Community

60+ grantees

21 countries

Full-time Part-time

Unitary Fund Programs

Microgrant Program

Small microgrants => big impact

Unitary Labs

- Open source research team
- Targeting impactful and open niches that aren't monetizable by companies

47 grants in 21 countries;
10+ publications;
30+ libraries with 1167 stars, 50+ contributors and ~6k commits
Helped 12 folks get into quantum tech full time

2 new startups, 1 new non-profit

>$100k

Granted to date

Mitiq: Error mitigating quantum compiler

```
from qiskit import QuantumCircuit
from mitiq import mitigate_executor
qskt_noisy_sim = mitigate_executor(qs)
circ = QuantumCircuit(1, 1)
for __ in range(120): circ.x(0)
circ.measure(0, 0)

expectation = qskt_noisy_sim(circ)
print(f"Error is {1 - expectation:.{3}}")

Error is 0.0582
```

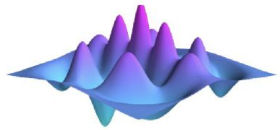
Built on Original Research:

Digital Zero Noise Extrapolation for Quantum Error Mitigation: https://arxiv.org/abs/2005.10921

Mitiq: A software package for error mitigation on noisy quantum computers: https://arxiv.org/abs/2009.04417

QuTiP: Quantum Toolbox in Python

Unitary Fund is setting up governance and fiscal sponsorship for a critical open source quantum technology package



qutip.org/

>30k annual downloads, >2500 citations

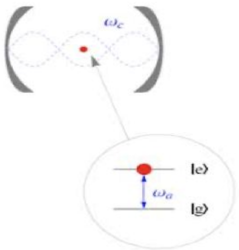
- Sep 2019 downloads 152k total

Unitary Fund

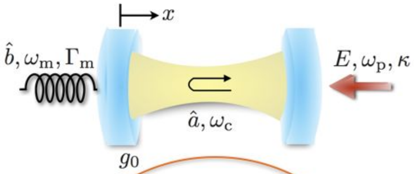
QuTiP: The Quantum *Physics* Simulator

The Quantum Toolbox in Python

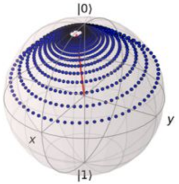
Cavity QED



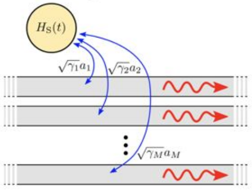
Optomechanics



Quantum Optics, QIP



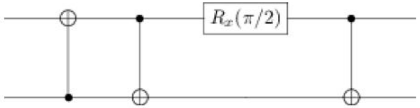
Waveguide QED



QuTiP

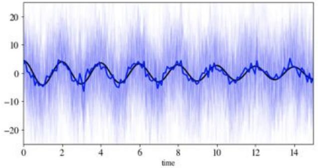
qutip.org

Quantum Circuits

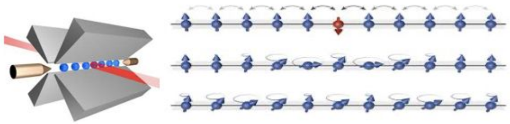


Quantum Optimal Control

Stochastic Dynamics

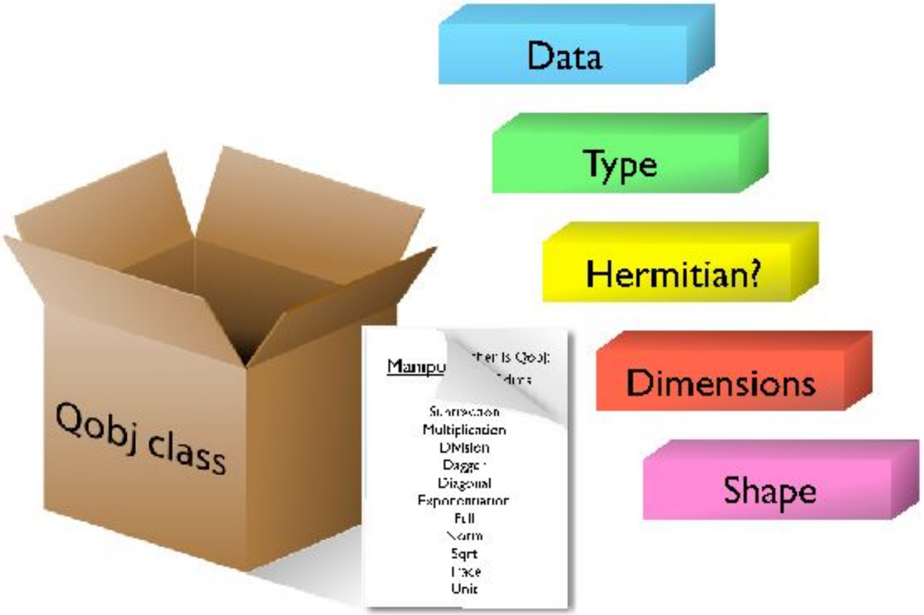


Spin Lattice Dynamics



Unitary Fund

Full Power of Python Classes



The `QObj` class

The **Qu**antum **T**oolbox **in** **P**ython

- State and operators are declared as **`QObj`**

  - Qubit

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

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

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

Take a snapshot

  - Operator

$$a, a^\dagger$$

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

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

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

States, operators, tensor spaces, density matrix

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

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

```
>> from qutip import *
>> H = sigmaz()/2
```

$$a, a^\dagger$$

```
>> a = destroy(2)
```

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

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

$$\sigma_z \otimes \sigma_x$$

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

$$\rho$$

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

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

What does a Schrödinger Cat looks like

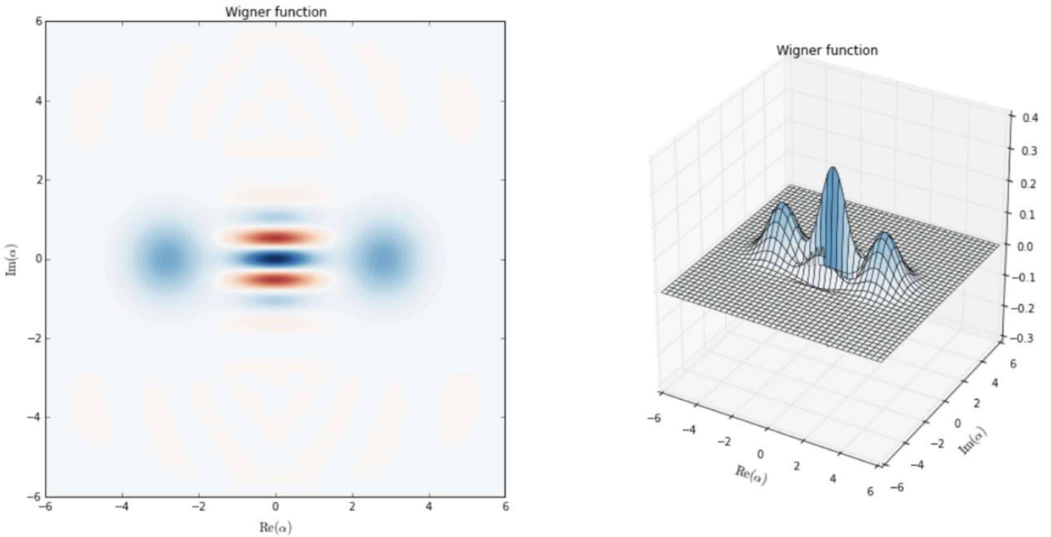
The **Qu**antum **T**oolbox **i**n **P**ython

qutip.org/tutorials

Superposition of coherent states

In [9]:
```
psi = (coherent(N, -2.0) + coherent(N, 2.0)) / np.sqrt(2)
plot_wigner_2d_3d(psi)
```

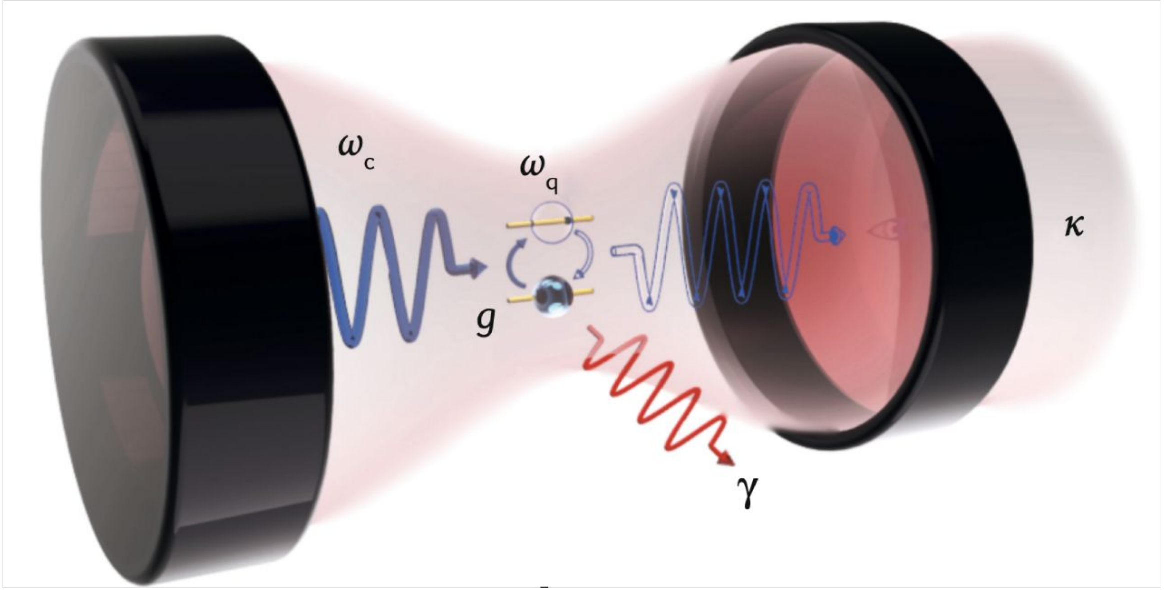
Out[9]:



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

Cavity Quantum Electrodynamics (Cavity QED)

Using the **Qu**antum **T**oolbox **i**n **P**ython to study physics



A. F. Kockum, A. Miranowicz, S. De Liberato, S. Savasta & Franco Nori, Nature Reviews Physics, **1** 19 (2019)

Numerical Solvers at a glance

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

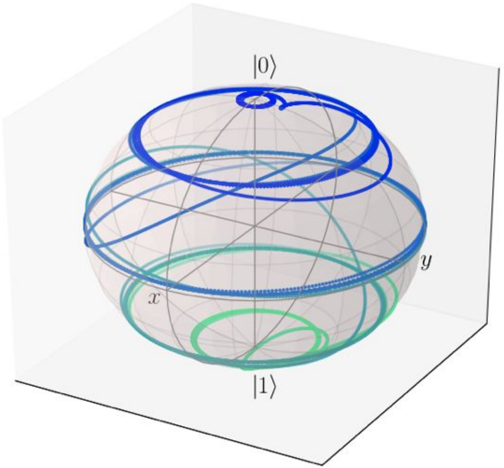
The **`Result`** class stores the expectation values of the operators passed to the solver.

```
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 trajectory
- **`Floquet_modes`**: Floquet theory
- **`bloch_redfield:`** Bloch-Redfield master equation
- **`ssesolve`**: Stochastic Schrödinger equation
- **`smesolve`:** Stochastic master equations



Monte Carlo Quantum Trajectories

Photon counting statistics with `mcsolve`

"quantum jump"

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

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

Stochastic solvers

Stochastic master equation: Continuous weak measurements with: `smesolve`

- continuous weak measurements

Heterodyne Detection

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

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

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

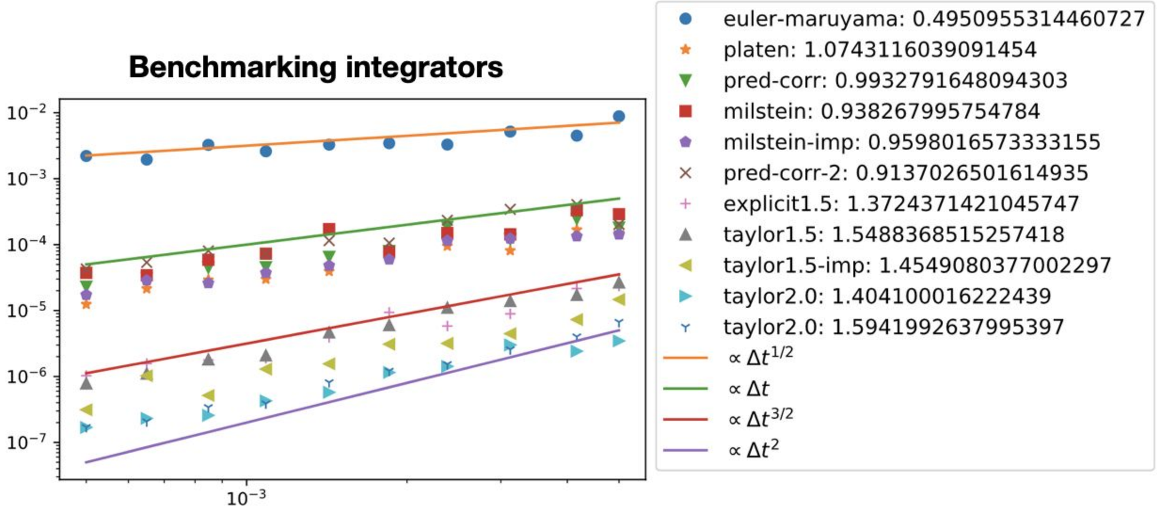
```
result = smesolve(H, rho0, times, [], c_ops, e_ops, method='heterodyne')
```

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

QuTiP: Numerical Stochastic Solvers

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

Method: extracted scaling

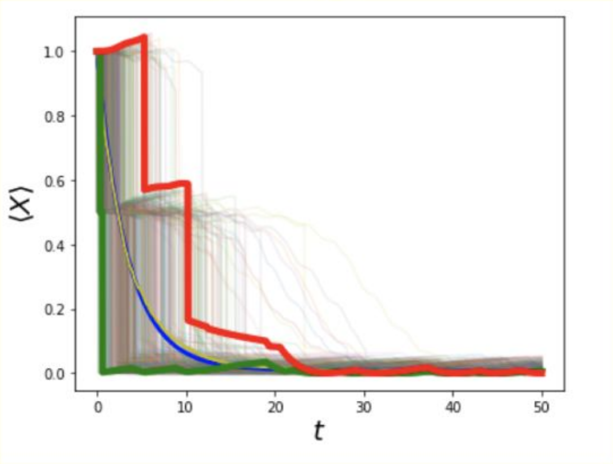



```
solution = ssesolve([H, c_ops = c_ops , fock(N), tlist, sc_op_td, e_op, **kwargs)
```

- `ssesolve`: Stochastic Schrödinger equation `result = ssesolve(H, rho0, times, [], c_ops, e_ops)`

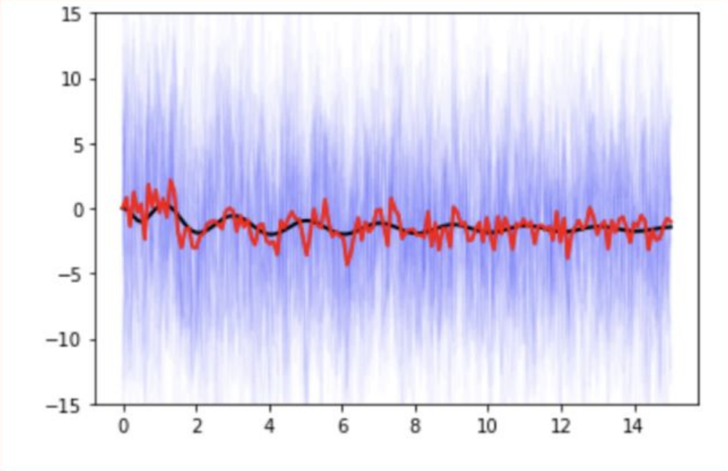
- `smesolve`: Stochastic master equations `result = smesolve(H, rho0, times, [], c_ops, e_ops)`

Monte Carlo Quantum Trajectories
`mcsolve`



50x faster
Time-dependent jumps

Stochastic Master Equations
`smesolve`

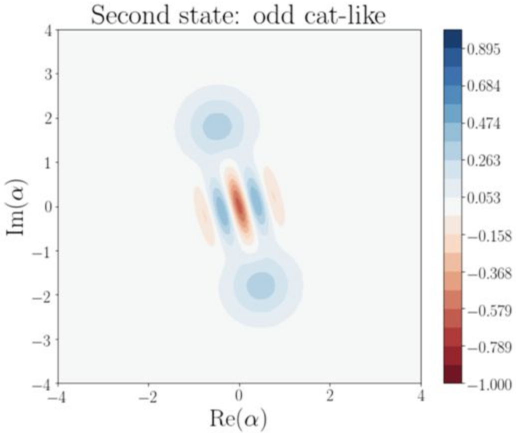


More general
More integration methods
Benchmarked solvers

Contributing to QuTiP – Tutorials

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

Dr. Fabrizio Minganti (RIKEN)→ **EPFL**



qutip.org/tutorials

Recent addition:
Cat vs coherent states in a Kerr resonator and the role of measurement

Homodyne

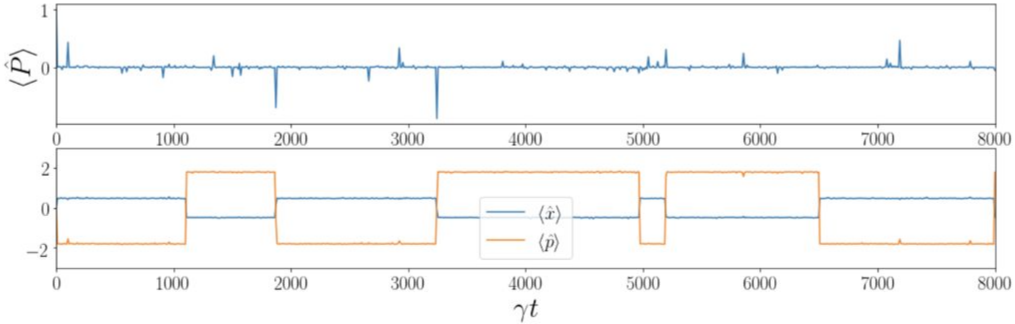
Another possible way to monitor a quantum-optical system is through homodyne detection, a widely-used experimental technique which allows to access the field quadratures [5-6]. To implement this kind of measurement, the cavity output field is mixed to the coherent field of a reference laser through a beam splitter (here assumed of perfect transmittance). Then, the mixed fields are probed via (perfect) photodectors, whose measures are described by new jump operators. We stress that both the coherent and the cavity fields are measured simultaneously.

In the ideal limit $\beta_{1,2} \to \infty$, the system evolves diffusively according to a homodyne stochastic Schr\"odinger equation. Using the ssesolve function with option "method='homodyne'", one can simulate the trajectory.

```
tlist=np.linspace(0,8000,800)


sol_hom=ssesolve(H, fock(20,0), tlist, c_ops, [a.dag()*a, (a+a.dag())/2, -1.j*(a-a.dag())/2, parit
```

```
Total run time: 636.04s
```



Contributing to QuTiP – Tutorials

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

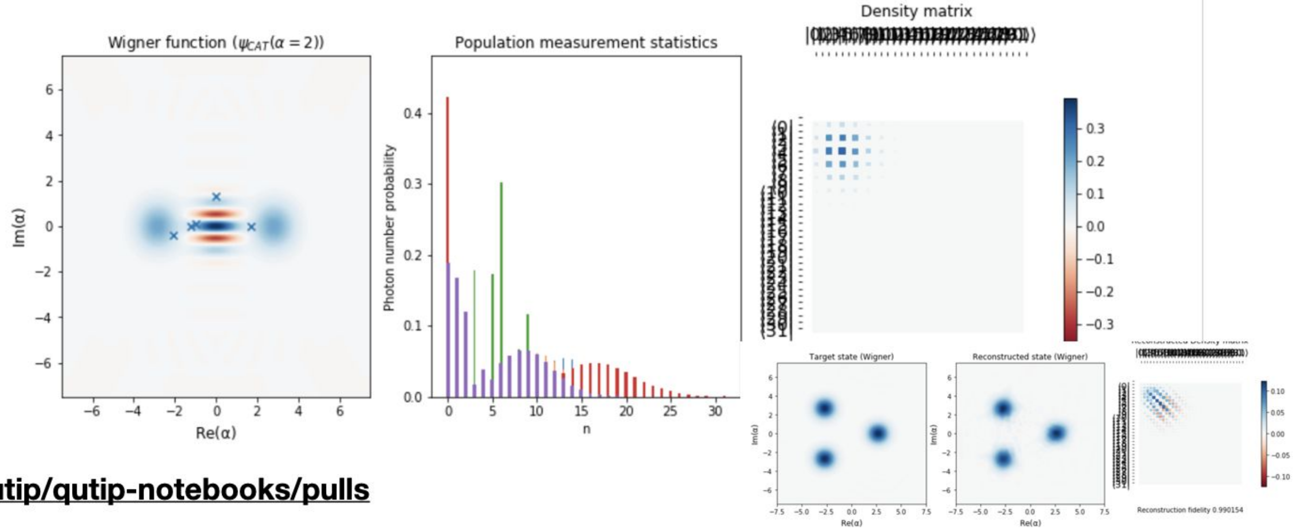
Pull Request: Optical Quantum State tomography with MaxLikelihood

Shahnawaz Ahmed (Chalmers)

Counting photons: measuring the photon number operator $|n\rangle\langle n|$

The expectation value of the photon number operator can be obtained easily from a state written in the fock basis - it is simply the diagonal elements of the density matrix, $|p_n|^2$. Here, we displace the initial state with various values of the displacement, β_i and then we can measure the photon number statistics by simply reading the diagonal elements of the displaced density matrix $\rho\,'$.

$$\rho\,' = D(\beta_i)\rho D^{\dagger}(\beta_i)$$



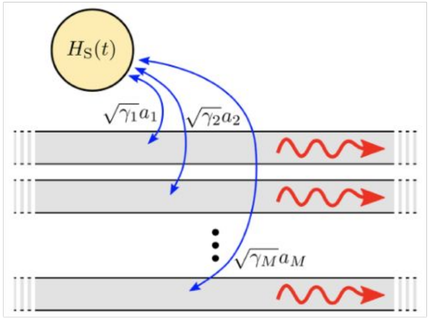
github.com/qutip/qutip-notebooks/pulls

QuTiP recent additions (2018-2019)

qutip.scattering

Model nonlinear photon scattering in multiple waveguides

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



K.A. Fischer, *et.al.* (2017), Quantum **2**, 69 (2018)

Code: Ben Bartlett. Github: bencbartlett

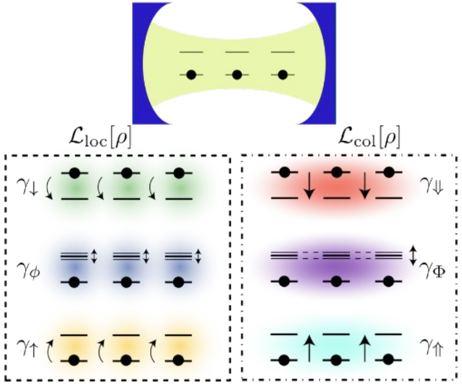
qutip.piqs

Efficiently model local dissipation

Collective dissipation

vs. Local dissipation

vs. Coherent coupling



N. Shammah *et al.*, Phys Rev A **98**, 063815 (2018)

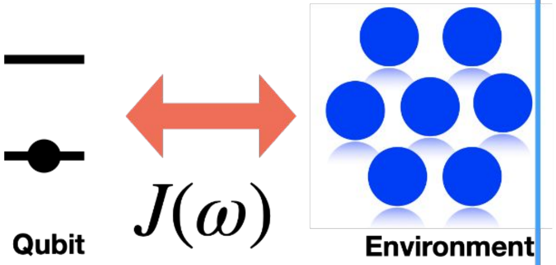
Code: Nathan Shammah and Shahnawaz Ahmed

qutip.nonmarkov

The environment has a memory

Non-Markovian dynamics

Hierarchical Equations of Motion (HEOM).



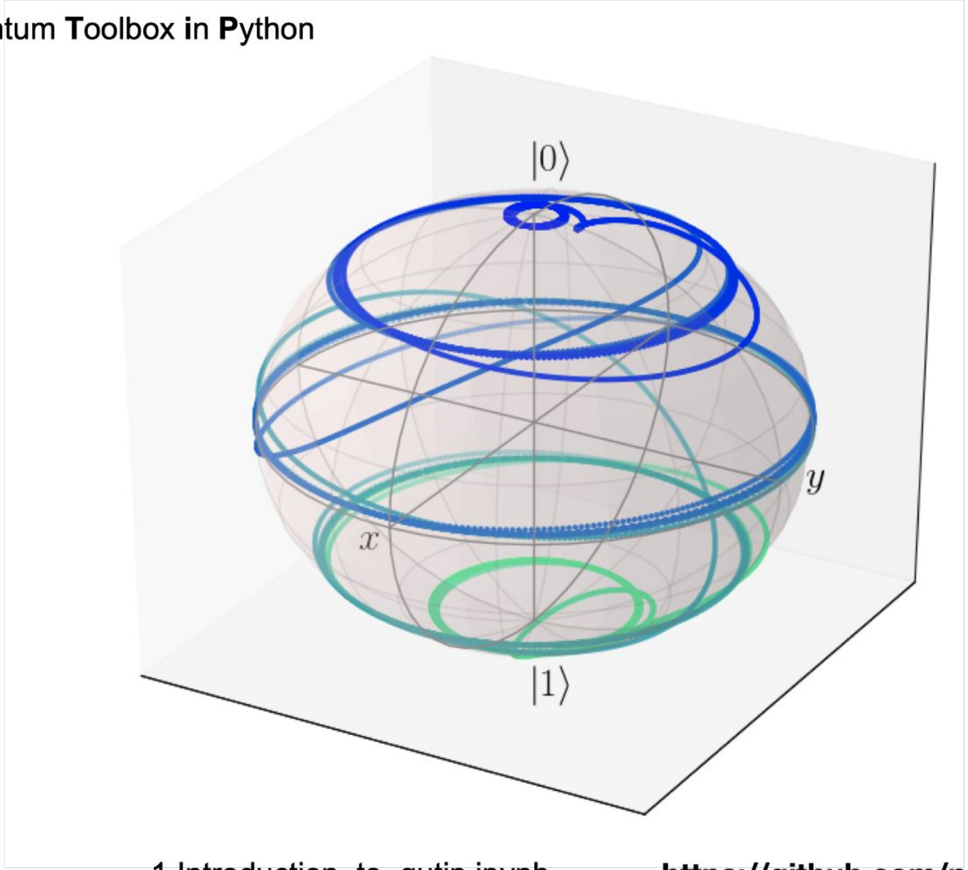
A. Fruchtman, *et al.*, Sci. Rep. **6**, 28204 (2016)

N. Lambert *et al.*, Nature Comm. **10**, 3721 (2019)

Code: Shahnawaz Ahmed and Neill Lambert

QuTiP: Visualize a qubit

The **Qu**an**t**um **T**oolbox **i**n **P**ython



1.Introduction_to_qutip.ipynb

qutip.org/tutorials

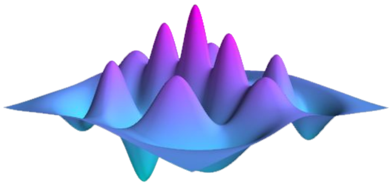
github.com/qutip

launch binder

Take a snapshot

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

QuTiP: The Quantum Toolbox in Python



About QuTiP

QuTiP is a free and open source library for efficient simulation of a wide variety of quantum systems.

Field-specific intuitive framework

The mathematics of quantum mechanics

- Complex number matrices
- Non-commutative algebras
- Operators and superoperators
- Intuitive Python classes: `QObj`
  `Results`

Advanced mathematical techniques

- **Noisy Dynamics**: Master equation solvers
- Correlation functions (quantum regression formula)
- **Modularity**: Permutational invariant quantum solver
  Hierarchical equations of motion
  Waveguide photon scattering
  Topological quantum circuits

Efficient numerical calculations

- *Fast complex-complex matrix-vector multiplication* with SSE3 intrinsics.
- **Multiprocessing**: Enhanced parallel performance with OPENMP
- **Sparse Matrices**: Fast CSR (Compressed Sparse Row) matrix class.
- *Up to 100x improvement:* in CSR adjoint & transpose
  in Hermitian verification, Krönecker product
  in partial trace calculation

More info at http://qutip.org/

QuTiP: Project Governance

Admin Team

Alex Pitchford
Aberystwyth University
Main focus: Quantum optimal control, solvers.

Eric Giguère
Université de Sherbrooke
Main focus: Stochastic solvers, coc optimization.

Neill Lambert
RIKEN
Main focus: Non-Markovian dynamics, counting statistics.

Boxi Li
Forschungszentrum Juelich
Main focus: Quantum control, Quantum device simulation.

Nathan Shammah
RIKEN
Main focus: Symmetrical models, education and outreach.

Shahnawaz Ahmed
Chalmers
Main focus: Machine learning, Nor Markovian dynamics.

Jake Lishman
Imperial College London
Main focus: Core data model and linear algebra.

Simon Cross
RIKEN
Main focus: Quantum control, machine learning, general maintenance.

Supporting Organizations

QuTiP is currently supported by these organizations:

RIKEN

PRIFYSGOL ABERYSTWYTH UNIVERSITY

INSTITUT QUANTIQUE UNIVERSITÉ DE SHERBROOKE

PRESTO SAKIGAKE

QuTiP is proud to be affiliated to:

NUMFOCUS OPEN CODE = BETTER SCIENCE

Unitary Fund

DEPARTMENT OF ENERGY UNITED STATES OF AMERICA

The development of QuTiP was partially supported by the following organizations:

고려대학교 KOREA UNIVERSITY

日本学術振興会

Board: Daniel Burgarth, Anton Frisk Kockum, Robert Johansson, Franco Nori, Will Zeng

https://github.com/qutip/governance

QuTiP: Empowering scientific research

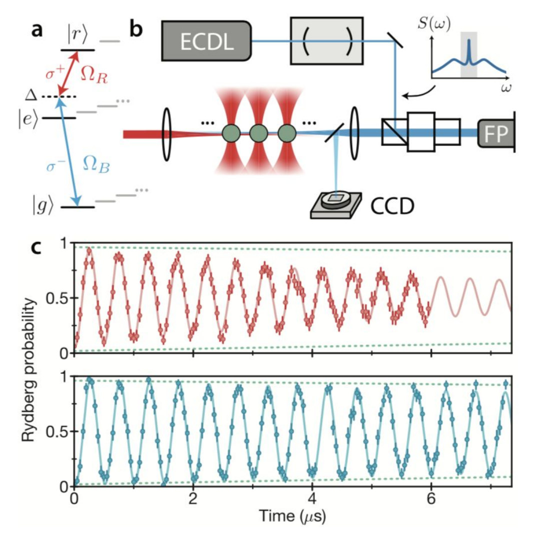
The **Qu**antum **T**oolbox **i**n **P**ython

Experimental Research

arXiv:1806.04682v1 [quant-ph] 12 Jun 2018

High-fidelity control and entanglement of Rydberg atom qubits

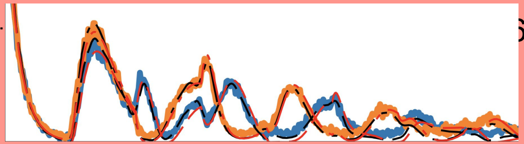
H. L [...] (2018)

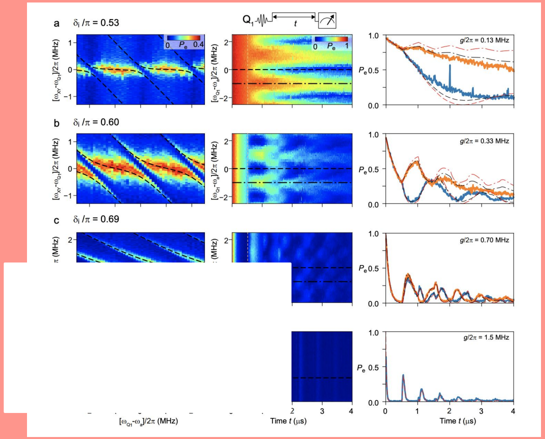


arXiv:1903.05672 [quant-ph] 13 Mar 2019

Phonon-mediated quantum state transfer and remote qubit entanglement

A. Bienfait, [...] (2019)





QuTiP: Impact and Recent Updates

- Over **13,000 downloads per month** from **PyPI** the Python Index Package only (`pip install qutip`) and **more than 350,000 overall** from the Conda Forge channel by Anaconda (`conda install qutip`).

- As of June 2021, **7076 commits**, in over **13 version releases** by over **96 contributors**.

- The two QuTiP **papers** have over **1500 citations** (Google Scholar).
-
- The **QuTiP admin team** is now formed of **eight maintainers meeting monthly**.
- Three ongoing Google Summer of Code projects:
    - 1. QuTiP-**CuPy**: **GPU** integration (Felipe).
    - 2. QuTiP-**Tensorflow** data back-end (Asier).
    - 3. **Universal decomposition** of quantum gates (Purva).

Unitary Fund

Cavity Quantum Electrodynamics (Cavity QED)

Using the **Qu**antum **T**oolbox **in** **P**ython to study physics. Things are scaling up

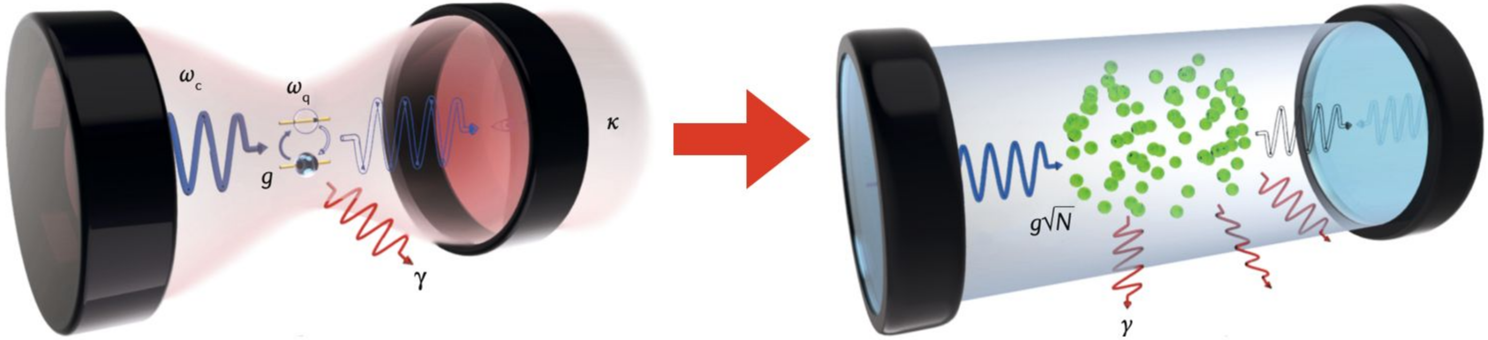


Image Credit: A. F. Kockum, A. Miranowicz, S. De Liberato, S. Savasta & Franco Nori, Nature Reviews Physics, **1** 19 (2019)

Solving a many-qubit dynamics

Closed vs. Open dynamics

Schrödinger Equation

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

Master Equation

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

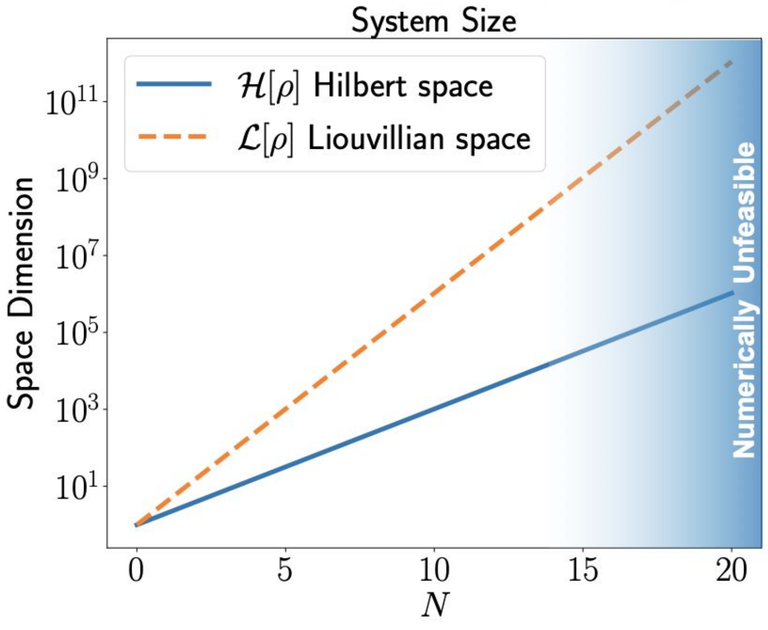
	Schrödinger Equation	Lindblad Master Equation
Probability	Conserved	Conserved
Evolution	Unitary, $H = H^\dagger$	Non-Unitary
Time reversal	Preserved	Broken
Features	Closed system	**Open system**: noise and dissipation
Scaling with N qubits	2^N	

Exploiting permutational symmetry

Numerically solving the dynamics of *many* qubits

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

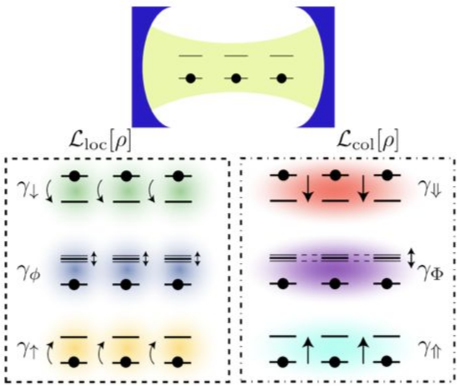
The Liouvillian space grows exponentially as 4^N

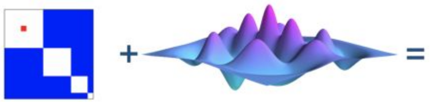


Qubits Number

Efficiently studying out-of-equilibrium systems

What's going on





import qutip.piqs

piqs.readthedocs.io

qutip.org/tutorials#piqs

Efficiently model local dissipation

Collective dissipation
vs. Local dissipation
vs. Coherent coupling

N. Shammah *et al.*,
***Phys Rev A* 98**, 063815 (2018)

Code: Nathan Shammah and Shahnawaz Ahmed

piqs
QuTiP library
Now a qutip module

krotov
QuTiP-based quantum
optimal control library

matsubara
A qutip plugin for
non-Markovian dynamics

QuTiP ecosystem:
Like AstroPy,
but for Quantum

Engineering Quantum Optimal Control

What's going on

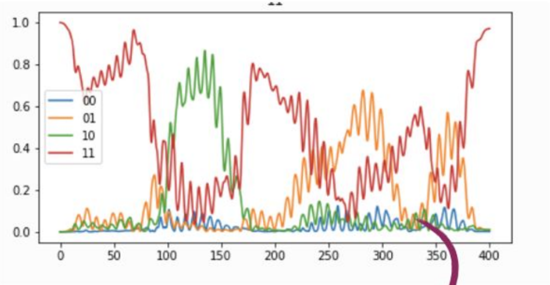
krotov.readthedocs.io

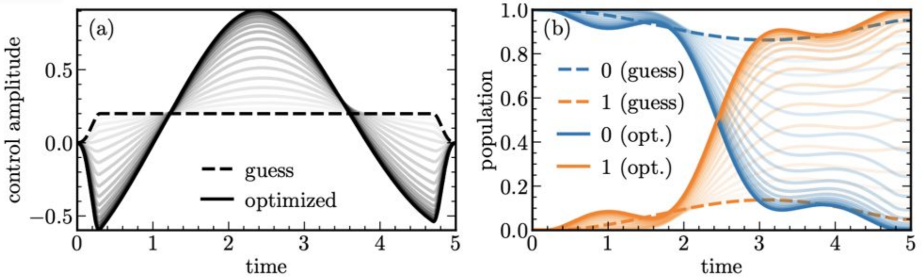
Examples

render on nbviewer | launch binder

- Optimization of a State-to-State Transfer in a Two-Level-System
- Optimization of a state-to-state transfer in a lambda system with RWA
- Optimization of a dissipative state-to-state transfer in a Lambda system
- Optimization of Dissipative Qubit Reset
- Optimization of an X-Gate for a Transmon Qubit
- Optimization of a Dissipative Quantum Gate
- Optimization towards a Perfect Entangler

Population evolutions for a gate (iSWAP)





Krotov: A Python implementation of Krotov's method for quantum optimal control
M. H. Goerz *et al.*, ***SciPost Phys.* 7**, 080 (2019)

piqs
QuTiP library
Now a qutip module

krotov
QuTiP-based quantum optimal control library

matsubara
A qutip plugin for non-Markovian dynamics

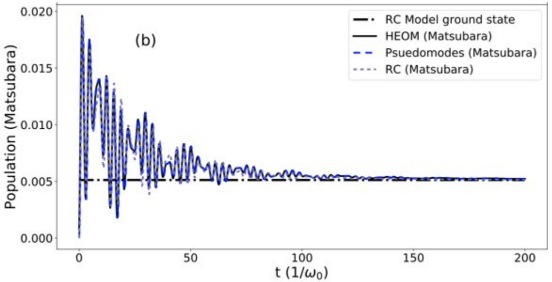
QuTiP ecosystem: Like AstroPy, but for Quantum

QuTiP plugin: The environment is watching

What's going on

matsubara.readthedocs.io

Comparison of open system models



Virtual excitations
in the ultra-strongly-coupled spin-boson model:
physical results from unphysical modes
N. Lambert, S. Ahmed, M. Cirio, and F. Nori
***Nature Communications* 10**, 3721 (2019)

piqs
QuTiP library
Now a qutip module

krotov
QuTiP-based quantum optimal control library

matsubara
A qutip plugin for non-Markovian dynamics

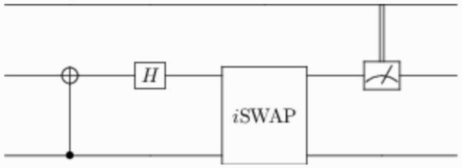
QuTiP ecosystem:
Like AstroPy,
but for Quantum

Quantum circuit simulation with `qutip-qip`

Installation: **`pip install qutip-qip`**

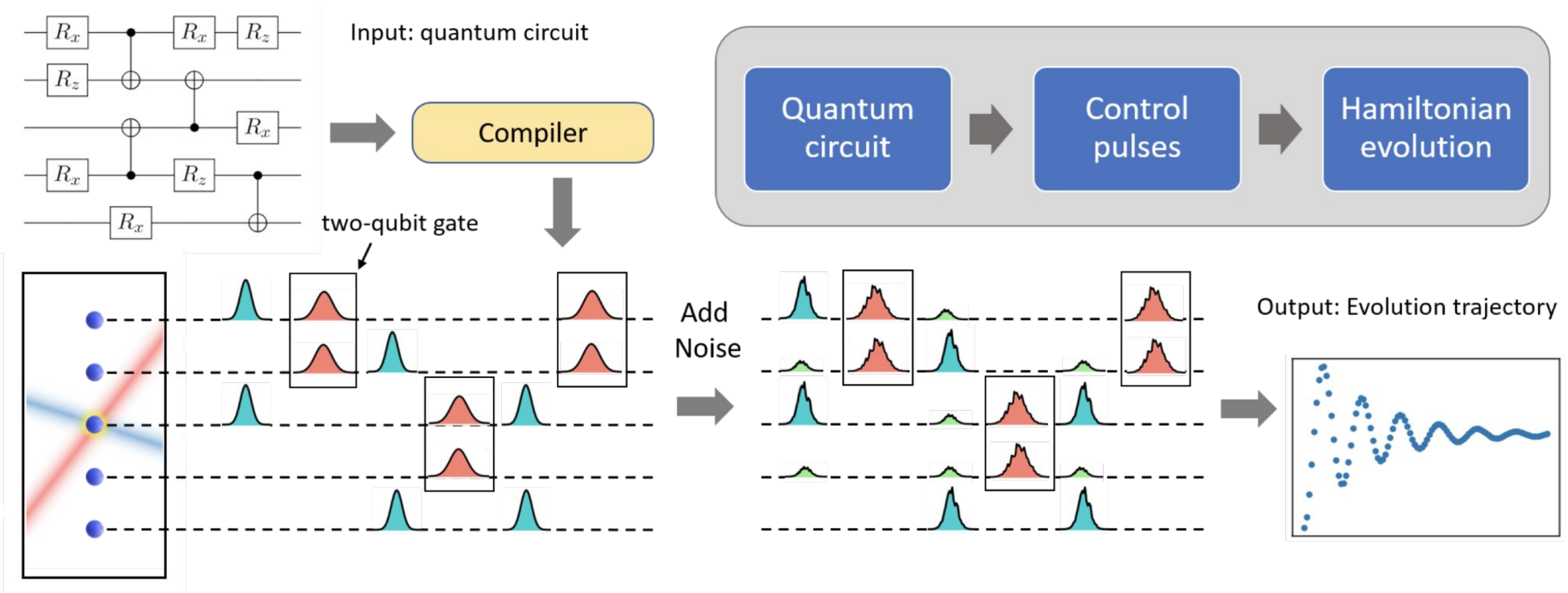
Documentation: **`qutip-qip.readthedocs.io`**

```python
from qutip_qip.circuit import QubitCircuit
from qutip_qip.operations import Gate
# create the quantum circuit
qc = QubitCircuit(2, num_cbits=1)
qc.add_gate("CNOT", controls=0, targets=1)
qc.add_gate("SNOT", targets=1)
qc.add_gate("ISWAP", targets=[0,1])
qc.add_measurement("M0", targets=1, classical_store=0)
# plot the quantum circuit
qc.png
```



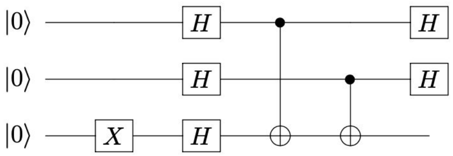
LaTeX rendering powered by **`QCircuit`**

Pulse-level noisy quantum circuits with `qutip-qip`



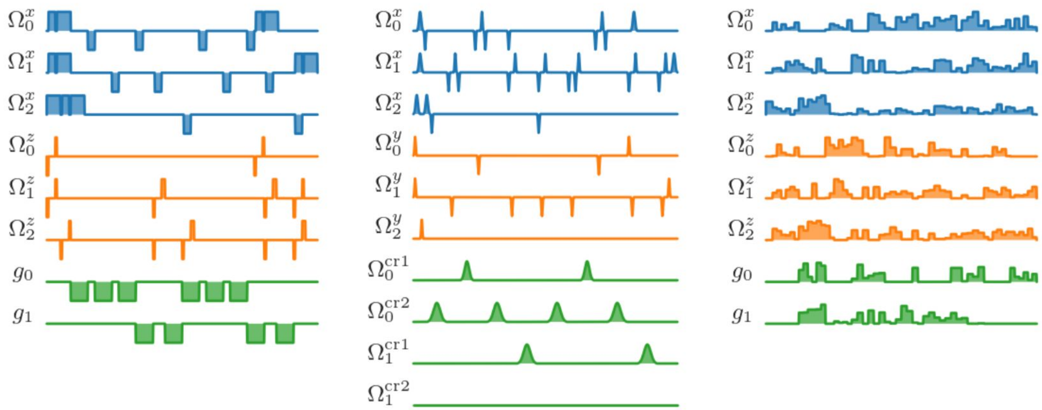
Pulse-level noisy quantum circuits with `qutip-qip`

Pulse-level noisy quantum circuits with `qutip-qip`



(a) an example of quantum circuit

Ω_j^α	Single-qubit rotation around an axis $\alpha = x, y, z$ (color blue and orange)
g_j	Coupling strength (color green)
$\Omega_j^{\mathrm{cr}k}$	The cross-resonance effective interaction (color green)



(b) Spin chain model

(c) Superconducting qubits model

(d) Optimal control model

Realistic and flexible simulation of noisy quantum processors.

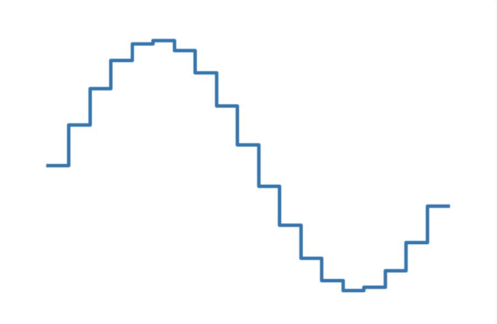
- Reproducing experimental results on crosstalk noise in an ion-based processor.
- Demonstrating the simulation of noisy quantum circuits on different hardware platforms with various native gate sets.
- Multiple types of model processors and noise models and solvers in QuTiP can be simulated.
- Integration with OpenQASM standard.

Pulse-level noisy quantum circuits with QuTiP, B. Li, S. Ahmed, S. Saraogi, N. Lambert, F. Nori, A. Pitchford, N. Shammah, arXiv:2105.09902

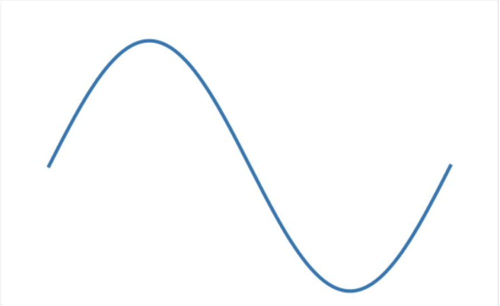
Unitary Fund

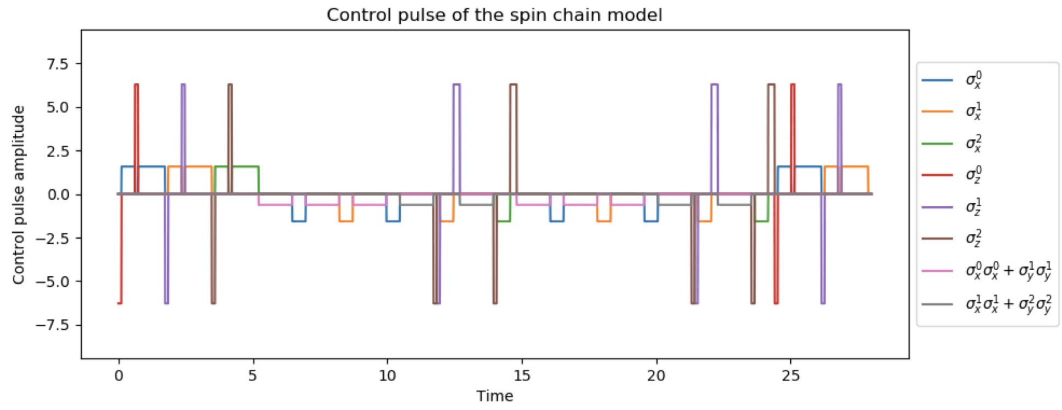
Pulse-level noisy quantum circuits with `qutip-qip`

Discrete Pulses



Continuous Pulses

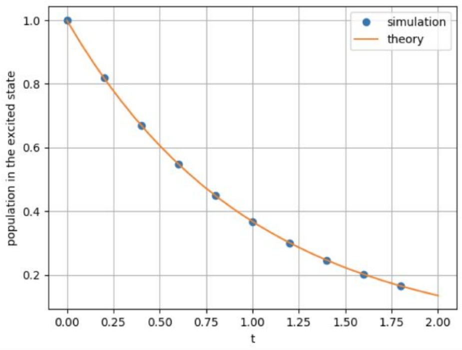




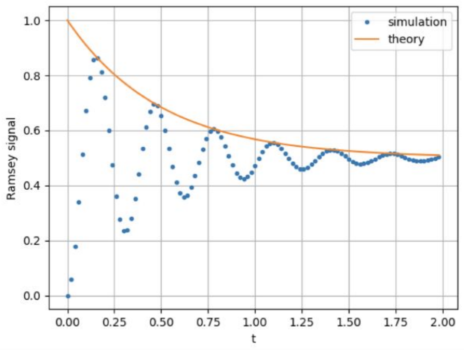
Pulse-level noisy quantum circuits with QuTiP, B. Li, S. Ahmed, S. Saraogi, N. Lambert, F. Nori, A. Pitchford, N. Shammah, arXiv:2105.09902

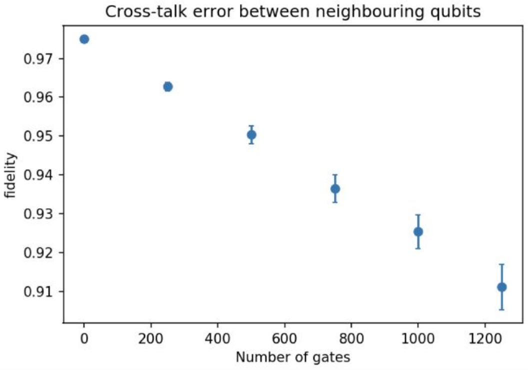
Pulse-level noisy quantum circuits with `qutip-qip`

Decay



Dephasing





Pulse wobbling

Random noise in the pulse intensity

Despite single-qubit decoherence, `Processor` can also simulate coherent control noise. For general types of noise, one can define a noise object and add it to the processor. An example of predefined noise is the random amplitude noise, where random value is added to the pulse every `dt`. `loc` and `scale` are key word arguments for the random number generator `np.random.normal`.

In [15]:
```
from qutip.qip.noise import RandomNoise
processor = Processor(N=1)
processor.add_control(0.5 * sigmaz(), targets=0, label="sigmaz")
processor.add_control(0.5 * sigmay(), targets=0, label="sigmay")
processor.coeffs = np.array([[1., 0., 1.],
                              [0., 1., 0.]])
processor.set_all_tlist(np.array([0., pi/2., 2*pi/2, 3*pi/2]))
processor_white = copy.deepcopy(processor)
processor_white.add_noise(RandomNoise(rand_gen=np.random.normal, dt=0.1, loc=-0.05, scale=0.02))  # gausian white noise
```

Pulse-level noisy quantum circuits with QuTiP, B. Li, S. Ahmed, S. Saraogi, N. Lambert, F. Nori, A. Pitchford, N. Shammah, arXiv:2105.09902

Overview of Mitiq

README.md

build passing | codecov 98% | docs passing | pypi package 0.8.0 | arXiv 2009.04417 | Downloads 11k | GitHub

SUPPORTED BY UNITARY FUND

Mitiq

- **Scope**: providing a toolkit to **easily** apply **different** error mitigation techniques
- **Compatible** with many existing quantum software libraries: **Cirq**, **Qiskit**, **PyQuil**, **Braket**.
- Structurally designed to be **backend independent**.

Mitiq uses the best practices of scientific software development

White paper
https://arxiv.org/abs/2009.04417

$ pip install mitiq
https://github.com/unitaryfund/mitiq
(version 0.9.1)

Documentation
https://mitiq.readthedocs.io

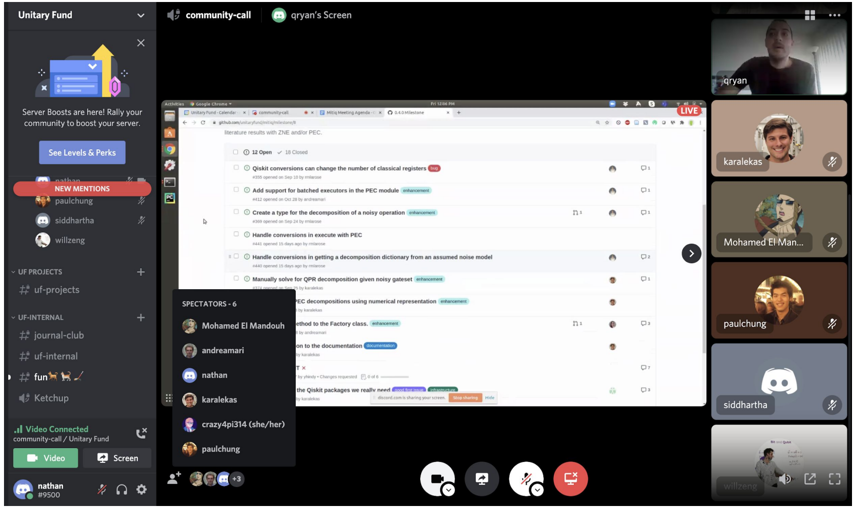
Development	**Testing**	**Continuous Integration**	**Research**	**Documentation**
March-September 2020	98% of lines	Github Actions	T. Giurgica-Tiron *et al.*, arXiv:2005.10921	Tutorials for any back-end
Releasing: GitHub, PyPI	TestPyPI	OS: Linux (ubuntu), Mac, Windows	R. LaRose *et al.*, arXiv:2009.04417	Sphinx.Doctest: >80 tests

Unitary Fund

Mitiq: Open development

Weekly Engineering Meeting: Community Call

Streamed at 12pm-12:40pm EDT at **discord.unitary.fund**



Over **20 external contributors** on GitHub, also thanks to the **UnitaryHACK** hackathon (May 14th – 30th, 2021)

🎉 Accepted Pull Requests 🎉

unitaryfund/mitiq:

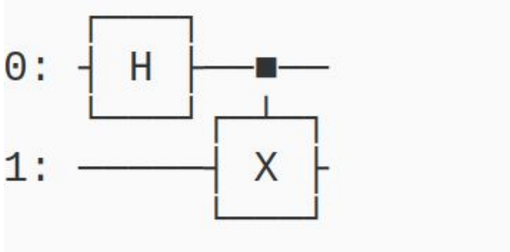
- **andre-a-alves** HACKED [UnitaryHACK] ZNE-PEC uniformity
- **maxtremblay** HACKED [UnitaryHACK] Improve conversion from braket to cirq
- **Yash-10** HACKED Warning for short programs [unitaryHACK]

Over **19 developers** beyond Unitary Fund! More than **10,000 downloads** from the Python Package Index.

Ideal quantum circuits

Ideal Von Neumann master equation: $\partial_t \rho = -i[K(t), \rho]$

noiseless
Hamiltonian



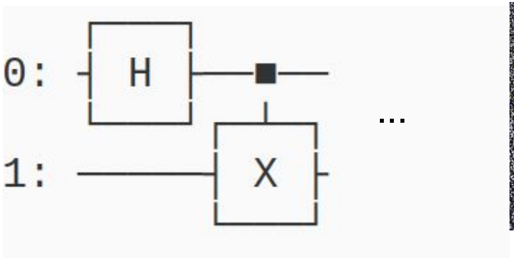
[1] K. Temme, S. Bravyi, and J. M. Gambetta, "Error Mitigation for Short Depth Quantum Circuits," Physical Review Letters,. **119**, 180509,(2017).

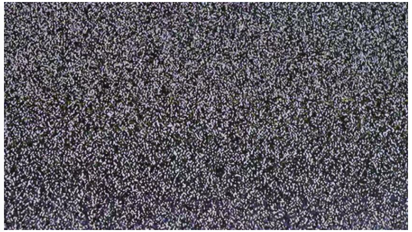
Unitary Fund

Ideal quantum circuits... on noisy computers

Ideal Von Neumann master equation: $\partial_t \rho = -i[K(t), \rho]$

noiseless
Hamiltonian





[1] K. Temme, S. Bravyi, and J. M. Gambetta, "Error Mitigation for Short Depth Quantum Circuits," Physical Review Letters,. **119**, 180509,(2017).

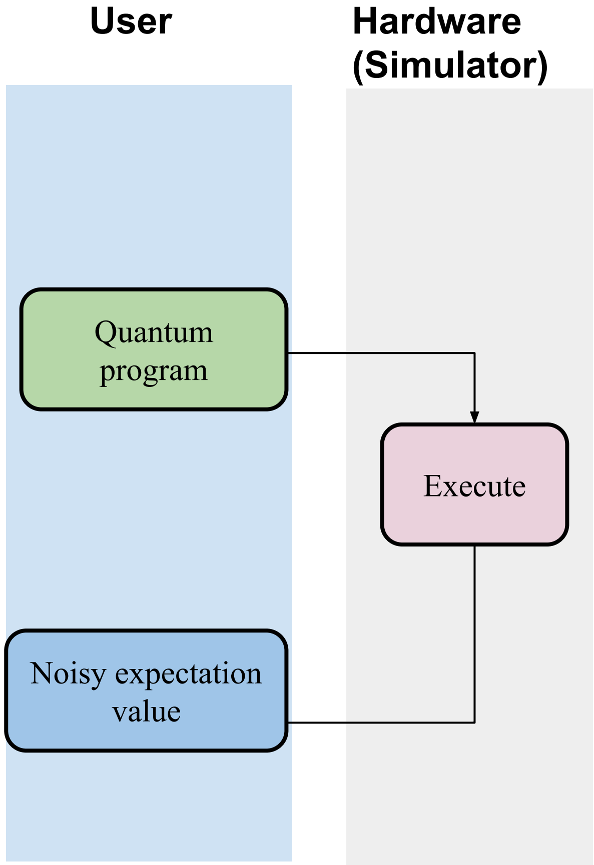
Typical workflow in quantum software

Qiskit

Cirq

rigetti

aws
Amazon Braket



IBM Q

rigetti

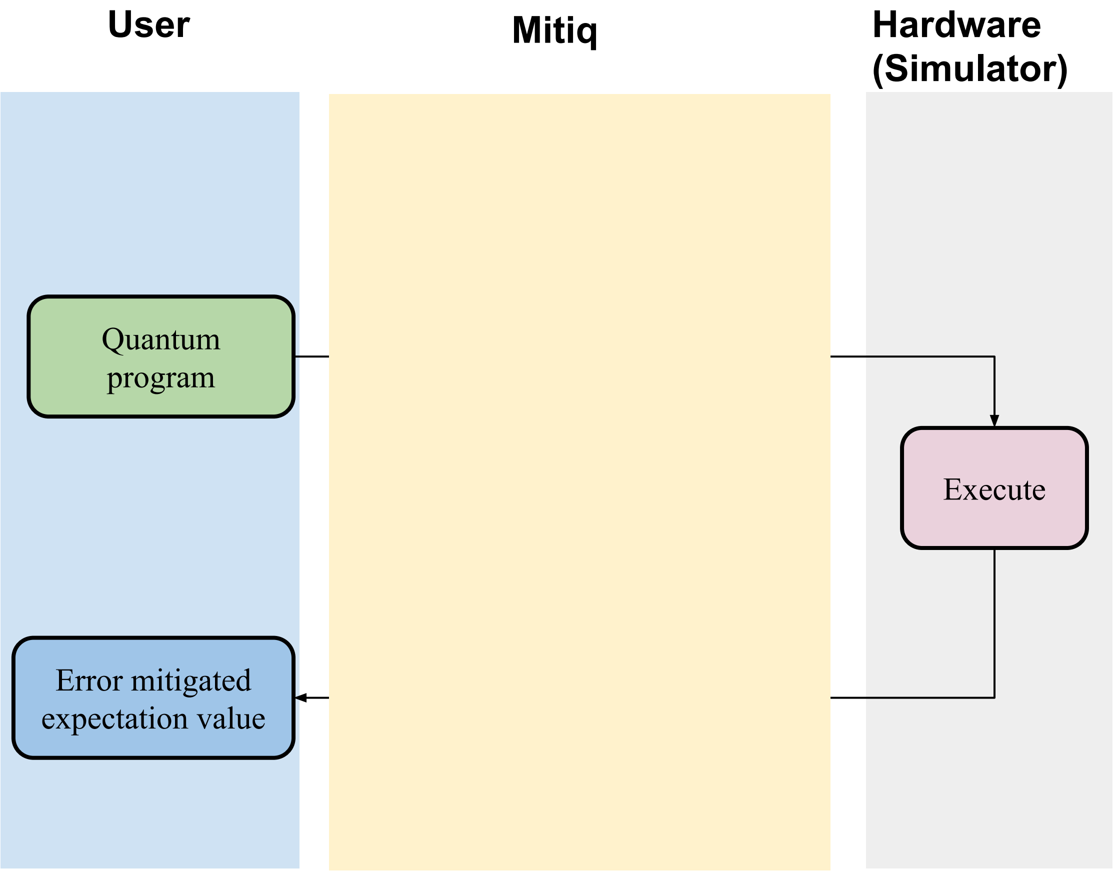
aws
Amazon Braket

QuTiP
Quantum Toolbox in Python

...

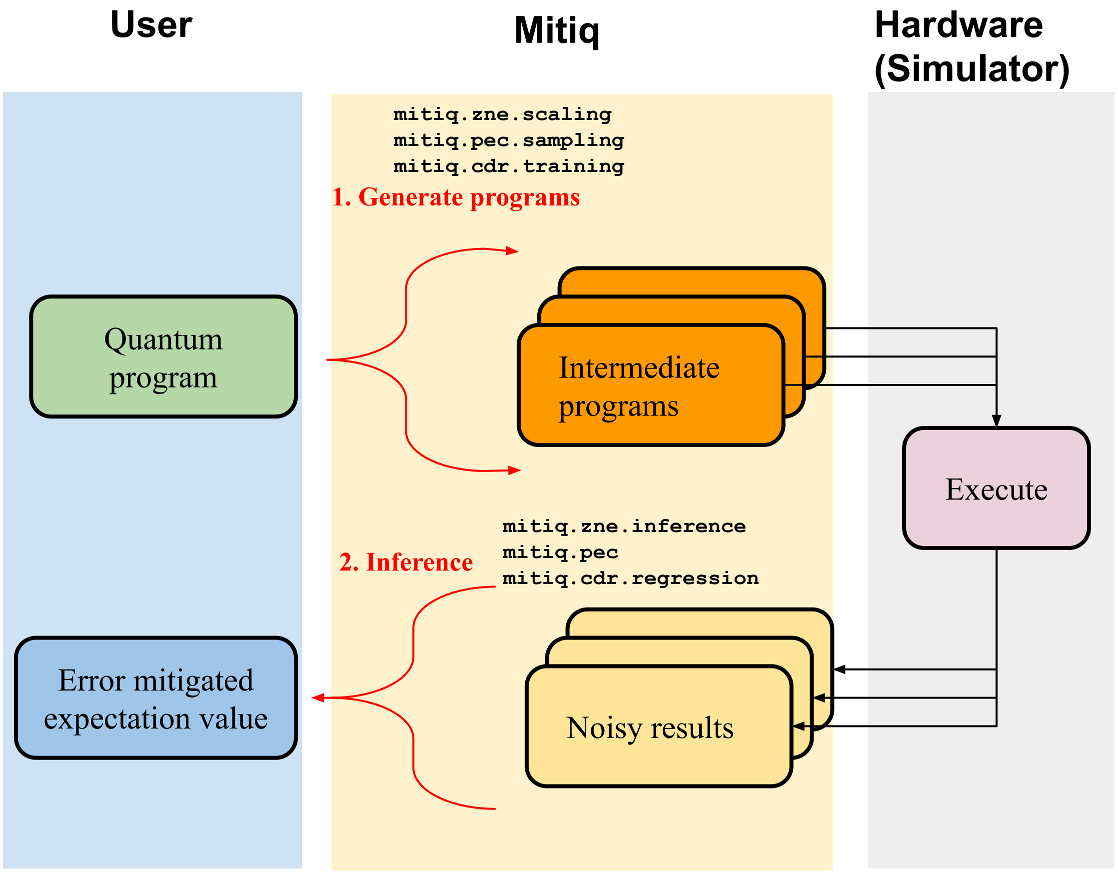
Unitary Fund

Mitiq: a thin layer between the user and the backend

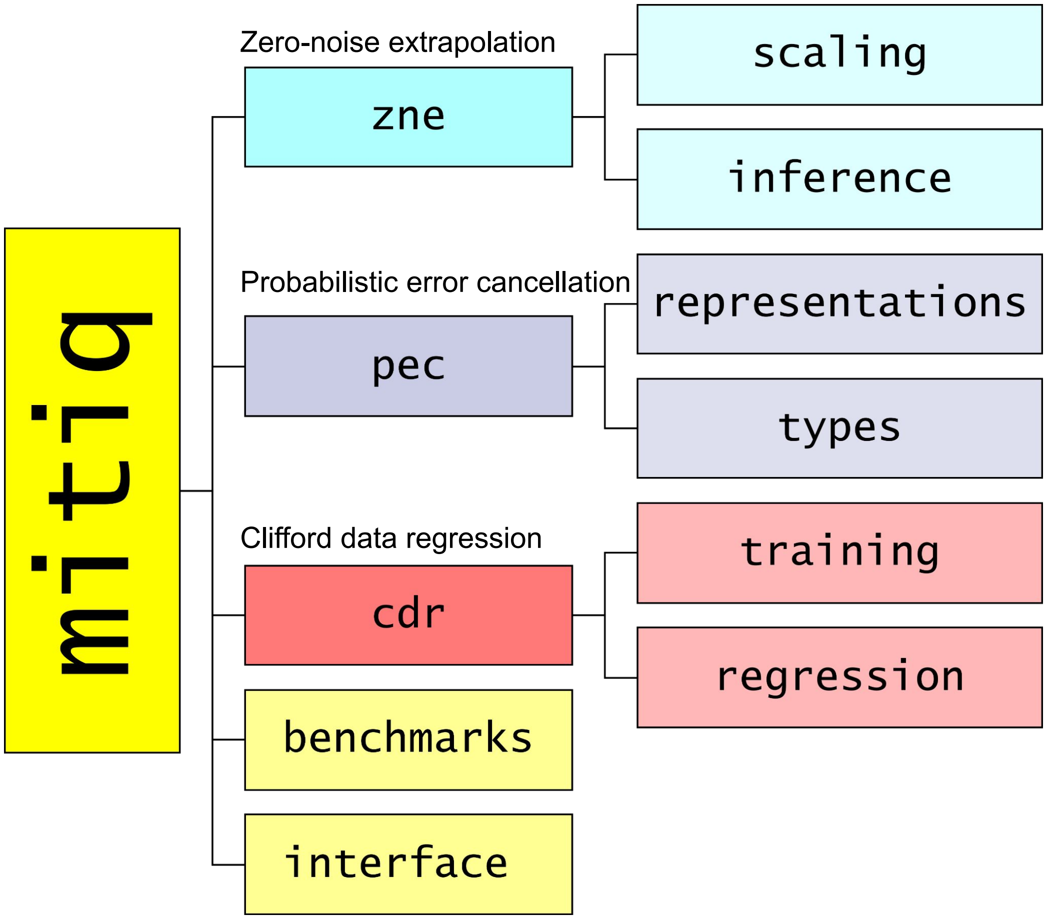


Unitary Fund

Mitiq: a thin layer between the user and the backend

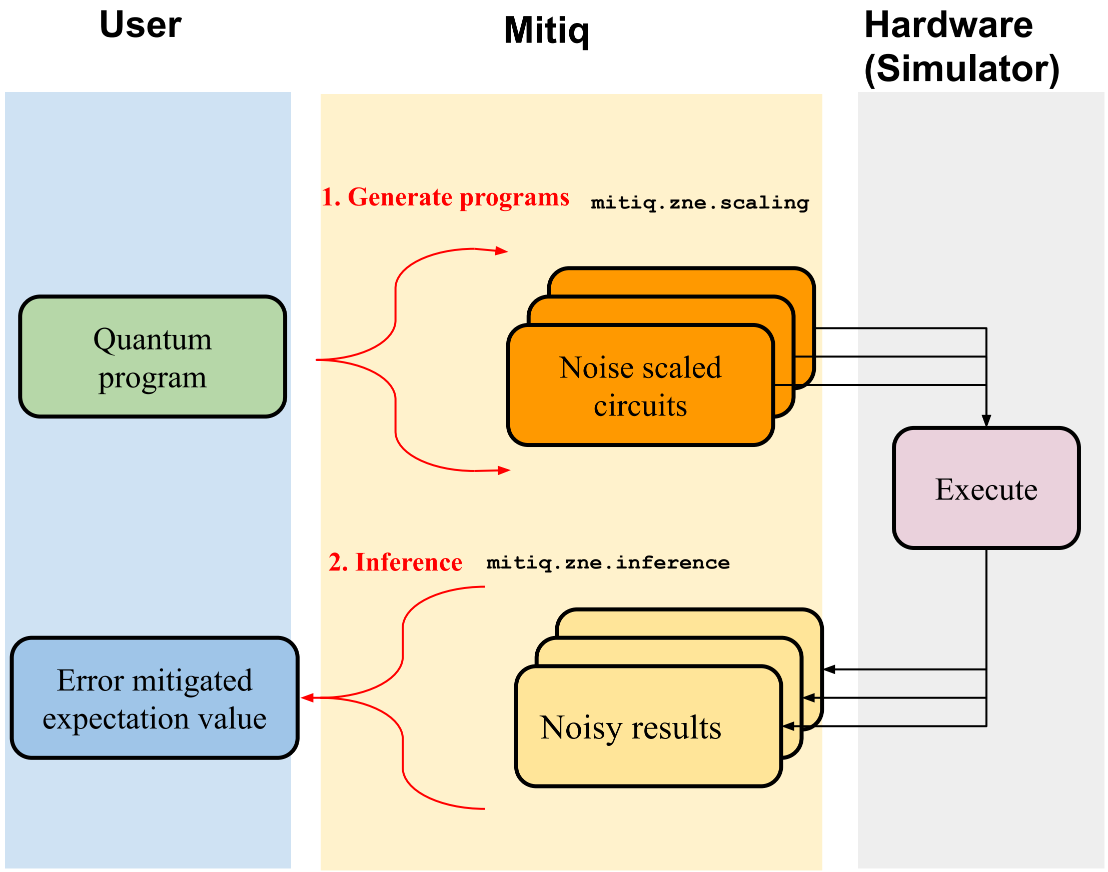


Mitiq code tree (simplified)



Unitary Fund

Zero-noise extrapolation workflow with Mitiq



Unitary Fund

Noisy quantum evolution

noise scale factor
[dimensionless real number]

Markovian master equation: $$\partial_t \rho = -i[K(t), \rho] + \lambda \mathcal{L}[\rho]$$

noiseless
Hamiltonian

noise operator

Noiseless evolution: $\lambda = 0$

Hardware dynamics: $\lambda = 1$

[1] K. Temme, S. Bravyi, and J. M. Gambetta, "Error Mitigation for Short Depth Quantum Circuits," Physical Review Letters,. **119**, 180509,(2017).

Unitary
Fund

Noisy quantum evolution

noise scale factor
[dimensionless real number]

Markovian master equation: $$\partial_t \rho = -i[K(t), \rho] + \lambda \mathcal{L}[\rho]$$

noiseless
Hamiltonian

noise operator

Noiseless evolution: $\lambda = 0$

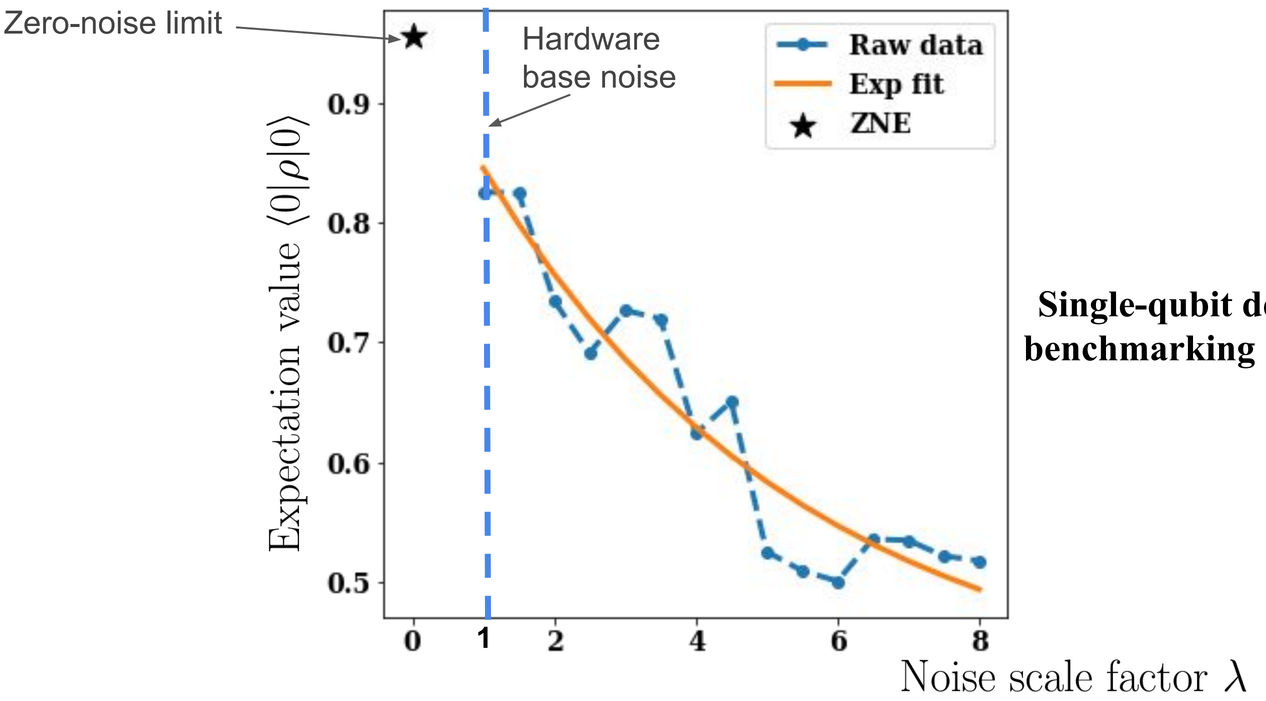
Hardware dynamics: $\lambda = 1$

Increase noise: $\lambda = 2$

[1] K. Temme, S. Bravyi, and J. M. Gambetta, "Error Mitigation for Short Depth Quantum Circuits," Physical Review Letters,. **119**, 180509,(2017).

Zero-noise extrapolation (ZNE): increasing noise… and back

Markovian master equation: $\partial_t \rho = -i[K(t), \rho] + \lambda \mathcal{L}[\rho]$



Single-qubit depth *d* = *200* randomized benchmarking circuit on IBMQ Armonk

Noise scaling [1]

noise scale factor
[dimensionless real number]

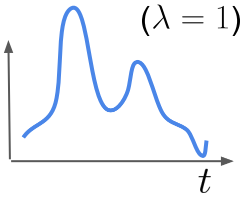
Markovian master equation: $$\partial_t \rho = -i[K(t), \rho] + \lambda \mathcal{L}[\rho]$$

noiseless
Hamiltonian

noise operator

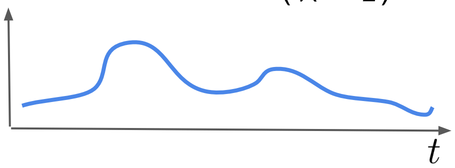
Experimentally:

$(\lambda = 1)$



In practice, all the physical
control pulses must be **stretched** [2]

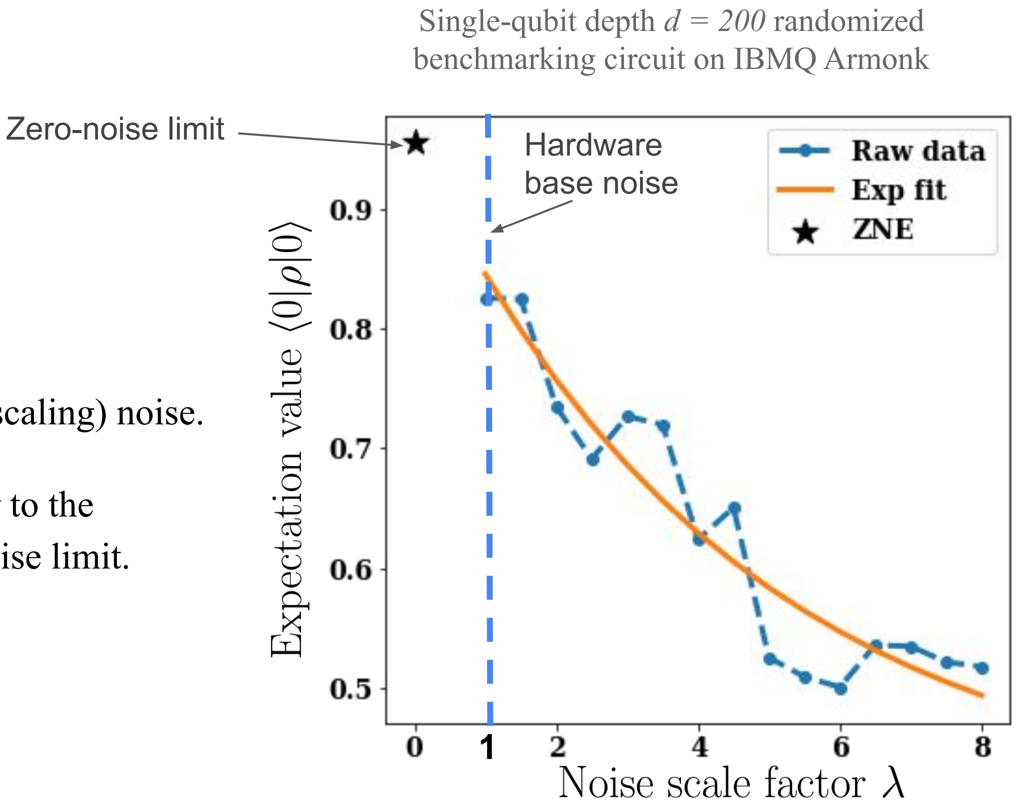
$(\lambda = 2)$



[1] K. Temme, S. Bravyi, and J. M. Gambetta, "Error Mitigation for Short Depth Quantum Circuits," Physical Review Letters,. **119**, 180509,(017).

[2] A. Kandala, K. Temme, A.D. Córcoles, A. Mezzacapo, J.M. Chow, and J.M. Gambetta, "Error mitigation extends the computational reach of a noisy quantum processor" Nature **567**, 491 (2019)

Zero-noise extrapolation



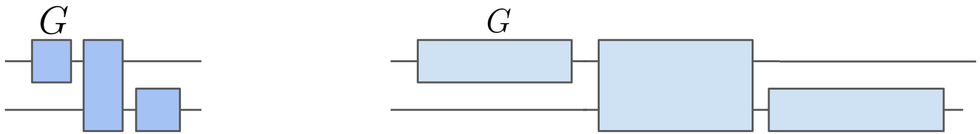
Two main requirements:

- **Noise scaling**: Some means of increasing (scaling) noise.
- **Inference method**: a model that one can *fit* to the measured data to *extrapolate* to the zero-noise limit.

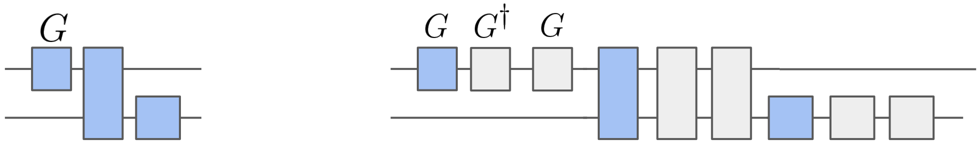
K. Temme, S. Bravyi, and J. M. Gambetta, "Error Mitigation for Short Depth Quantum Circuits," Physical Review Letters, vol. 119, p. 180509, 11 2017.
T. Giurgica-Tiron, Y. Hindy, R. LaRose, A. Mari, W. J. Zeng, *Digital zero noise extrapolation for quantum error mitigation*, 2020 IEEE International Conference on Quantum Computing and Engineering (QCE) (2020).

Analog and *digital* noise scaling methods

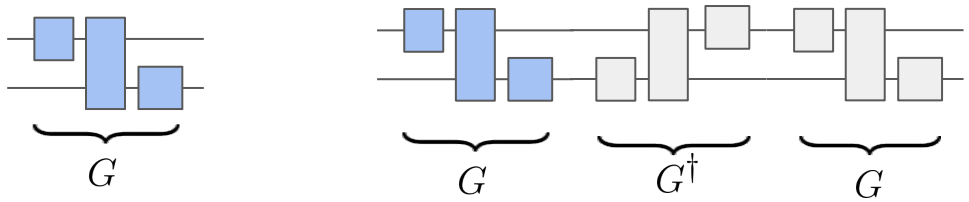
- Pulse stretching



- Local folding



- Global folding



K. Temme, S. Bravyi, and J. M. Gambetta, "Error Mitigation for Short Depth Quantum Circuits," Physical Review Letters, vol. 119, p. 180509, 11 2017.

T. Giurgica-Tiron, Y. Hindy, R. LaRose, A. Mari, W. J. Zeng, *Digital zero noise extrapolation for quantum error mitigation*, 2020 IEEE International Conference on Quantum Computing and Engineering (QCE) (2020).

Example: noise scaling by *global folding* with Mitiq

```
>>> import cirq
>>> from mitiq.zne.scaling import fold_global

# Get a circuit to fold
>>> qreg = cirq.LineQubit.range(2)
>>> circ = cirq.Circuit(cirq.ops.H.on(qreg[0]), cirq.ops.CNOT.on(qreg[0], qreg[1]))
>>> print("Original circuit:", circ, sep="\n")
Original circuit:
0: ───H───@───
          │
1: ───────X───

# Fold the circuit
>>> folded = fold_global(circ, scale_factor=3.)
>>> print("Folded circuit:", folded, sep="\n")
Folded circuit:
0: ───H───@───@───H───H───@───
          │   │           │
1: ───────X───X───────────X───
```

G

$\lambda = 3$

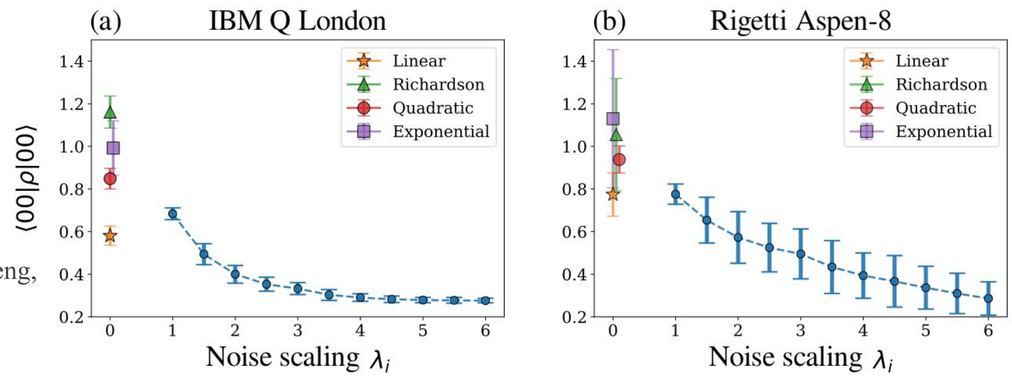
$G \quad G^\dagger \quad G$

Example: using different extrapolation methods in Mitiq with **`Factories`**

`mitiq.zne.inference.`

Class	Extrapolation Method
`LinearFactory`	Extrapolation with a linear fit.
`RichardsonFactory`	Richardson extrapolation.
`PolyFactory`	Extrapolation with a polynomial fit.
`ExpFactory`	Extrapolation with an exponential fit.
`PolyExpFactory`	Similar to `ExpFactory` but the exponent can be a non-linear polynomial.
`AdaExpFactory`	Similar to `ExpFactory` but the noise scale factors are adaptively chosen.

Experiment (2-qubit RB circuit):



R. LaRose, A. Mari, P. J. Karalekas, N. Shammah, W. J. Zeng, “Mitiq: A software package for error mitigation on noisy quantum computers”, arXiv:2009.04417 (2020).

Probabilistic Error Cancellation

Typical step of many quantum algorithms: **estimating expectation values**.

Ideal quantum circuit $\mathcal{U} = \mathcal{G}_t \circ \ldots \mathcal{G}_2 \circ \mathcal{G}_1$ Ideal gates (local unitary operations)

Ideal expectation value Observable Initial state

$$\langle A \rangle_{\text{ideal}} = \text{tr}[A\mathcal{U}(\rho_0)],$$

Unfortunately, we cannot apply an ideal gate with a NISQ device, but… we can **represent** it as a linear combination of noisy gates!

Replacing $\mathcal{G}_i = \sum_{\alpha} \eta_{i,\alpha} \mathcal{O}_{i,\alpha},$ we get: $\langle A \rangle_{\text{ideal}} = \sum_{\vec{\alpha}} \eta_{\vec{\alpha}} \langle A_{\vec{\alpha}} \rangle_{\text{noisy}}$

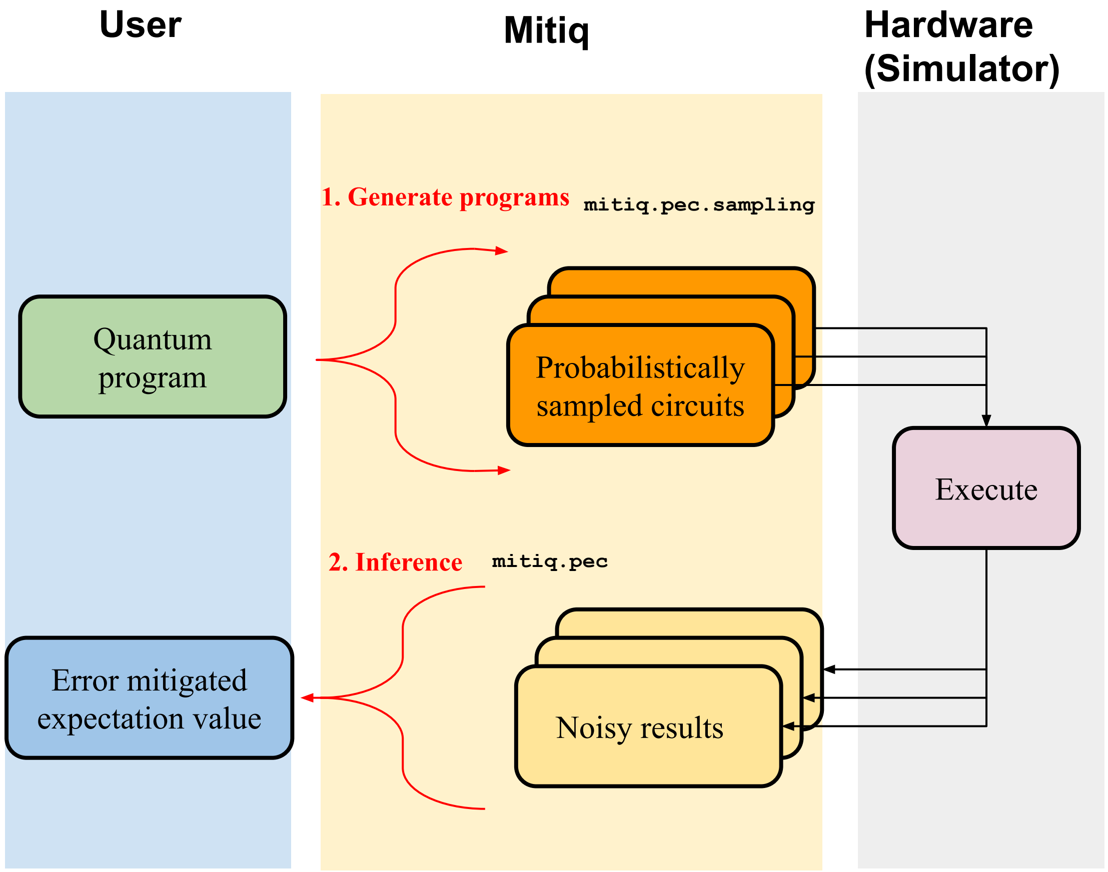
Quasi-probability (can have negative values)

Implementable noisy gates

We can express the **ideal result** as a linear combination of **noisy results**! One estimates the sum by **probabilistically sampling** and measuring only a subset of all the terms.

K. Temme, S. Bravyi, and J. M. Gambetta, "Error Mitigation for Short Depth Quantum Circuits," Physical Review Letters, vol. 119, p. 180509, 11 2017.

Probabilistic Error Cancellation workflow with Mitiq



Probabilistic Error Cancellation with Mitiq

Define the **circuit of interest**

```
from cirq import X, H, CNOT

seed = np.random.RandomState(0)
q0, q1 = LineQubit.range(2)
ideal_circuit = Circuit(X(q0), H(q1), CNOT(q0, q1))
ideal_circuit
```

```
0: ───X───@───
          │
1: ───H───X───
```

Define an **executor** (a black box function to get noisy expectation values)

```
from cirq import DensityMatrixSimulator

SIMULATOR = DensityMatrixSimulator()

def noisy_executor(circ: Circuit) -> float:
    """Simulates a circuit with depolarizing noise and returns the expectation value
    of the projector on the ground state |00...><00...|.
    """
    circuit = circ.with_noise(depolarize(BASE_NOISE))
    rho = SIMULATOR.simulate(circuit).final_density_matrix
    projector = np.zeros(rho.shape)
    projector[0, 0] = 1.0
    expectation = np.real(np.trace(rho @ projector))
    return expectation
```

Unitary Fund

Probabilistic Error Cancellation with Mitiq

Using the `mitiq.pec.representations` sub-module:

```
from cirq import LineQubit, Circuit, X
from mitiq.pec.representations import represent_operation_with_local_depolarizing_noise

BASE_NOISE = 0.1
q = LineQubit(1)

x_rep = represent_operation_with_local_depolarizing_noise(Circuit(X(q)), noise_level=BASE_NOISE)
print(x_rep)
```

```
1: ───X─── = 1.115*1: ───X───-0.038*1: ───X───X───-0.038*1: ───X───Y───-0.038*1: ───X───Z───
```

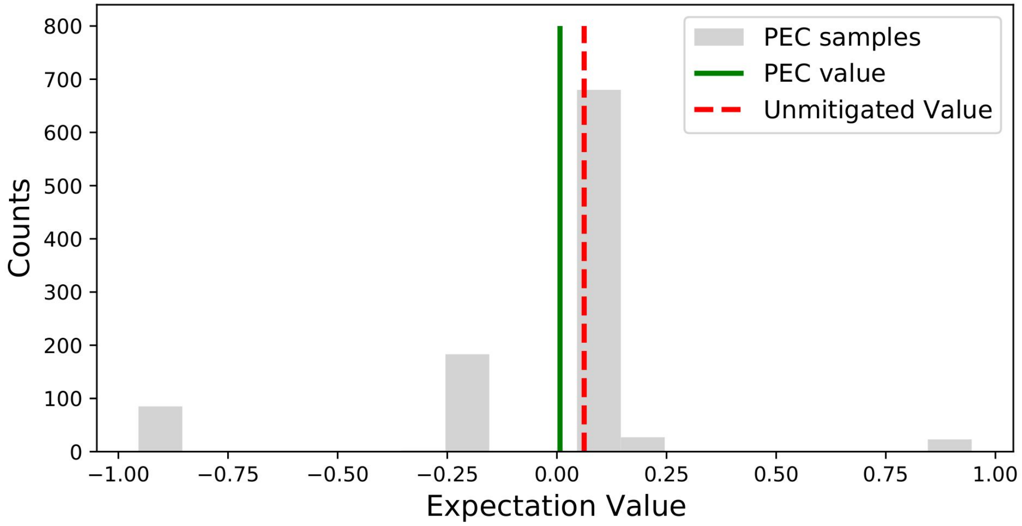
$$\mathcal{G}_i = \sum_{\alpha} \eta_{i,\alpha} \mathcal{O}_{i,\alpha},$$

Probabilistic Error Cancellation with Mitiq

Beyond expectation values: analysis of **raw data**

```
from mitiq.pec import execute_with_pec

pec_value, pec_data = execute_with_pec(ideal_circuit, noisy_executor, representations, full_output=True)
```



Thank you!

discord.unitary.fund

nathan@unitary.fund

Unitary Fund