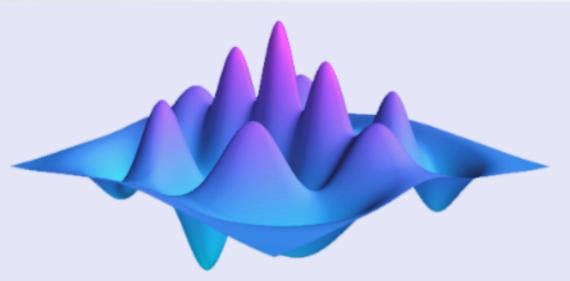


Supyter AND



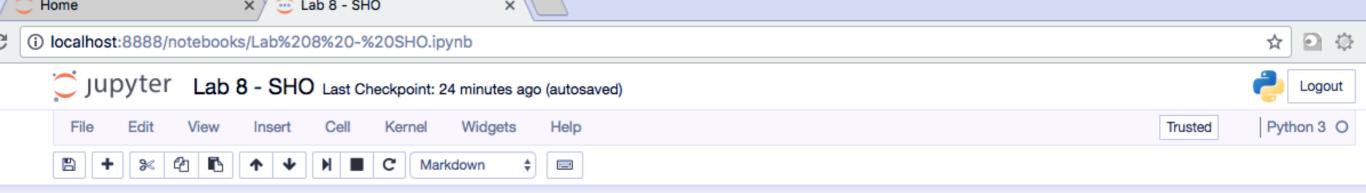
QuTiP Quantum Toolbox in Python



CJUpyter

- interactive computing
- large community
- self-help is built-in (IPython)
- notebook self-documenting

EXPLOYING WARNING Don't rely on this sen Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Context Con		•	
Comparison		JUPYTET Lorenz Differential Equations	(autosaved)
Exploring the Lorenz SystemView insert CellView insert CellImage: System of CellImage		File Edit View Insert Cell Kernel Help	
View Inset Cell In this Notebook we explore the Lorenz system of differential equations: $\hat{x} = \sigma(y - x)$ $\hat{y} = \rho x - y - xz$ $\hat{z} = -\beta z + xy$ This is one of the classic systems in non-linear differential equations. It is solutions. The system was originally developed as a simplified mathema atmospheric convection in 1963. Welcome to the This Notebook Server was WARNING Don't rely on this sen Your server is hosted thar In Cilick on the cell to st 2. Press SHIFT+ENTER A full tutorial for using the transport pandas as pd import pandas as pd import pandas as pd import numpy as np		E + 3< ℓ2 E + ↓ ► ■ C Code	Cell Toolbar: None
View Insert Cell View Insert Cell In this Notebook we explore the Lorenz system of differential equations: $\hat{x} = \sigma(y - x)$ $\hat{y} = \rho x - y - xz$ $\hat{z} = -\beta z + xy$ This is one of the classic systems in non-linear differential equations. It is omplex behaviors as the parameters (σ , β , ρ) are varied, including what solutions. The system was originally developed as a simplified mathematic atmospheric convection in 1963. Welcome to the This Notebook Server wat WARNING Don't rely on this sen Your server is hosted that Your server is hosted that 1. Click on the cell to st 2. Press SHIFT+ENTER A full tutorial for using the Imatplot11b inline import pandas as pd import numpy as np	OK WALLER	Exploring the Lorenz Sv	stem
View insert Coll $\dot{x} = \sigma(y - x)$ $\hat{y} = \rho x - y - xz$ $\dot{z} = -\beta z + xy$ This is one of the classic systems in non-linear differential equations. It is complex behaviors as the parameters (σ , β , ρ) are varied, including what solutions. The system was originally developed as a simplified mathematic atmospheric convection in 1963. Welcome to the In [7]: This Notebook Server wat In [7]: WARNING Don't rely on this server is hosted that Your server is hosted that σ 10 β β 2.6 ρ 2.8 Run some Python (To run the code below: 1. Click on the cell to set $2.$ Press SHIFT+ENTER A full tutorial for using the Imatplot11b inline Import pandas as pd import numpy as np	21 Welcome to P		
$\dot{y} = \rho x - y - xz$ $\dot{z} = -\beta z + xy$ This is one of the classic systems in non-linear differential equations. It complex behaviors as the parameters (σ , β , ρ) are varied, including what solutions. The system was originally developed as a simplified mathematimus atmospheric convection in 1963. Welcome to the Interact (Lorenz, N=fixed(10), angle=(0.,360.), $\sigma=(0.0,50.0), \beta=(0.0,50.0)$) In [7]: interact (Lorenz, N=fixed(10), angle=(0.,360.), $\sigma=(0.0,50.0), \beta=(0.0,50.0)$) WARNING Don't rely on this sen Your server is hosted that Run some Python of Torun the code below: 1. Click on the cell to se 2. Press SHIFT+ENTER A full tutorial for using the Imatplotlib inline import pandas as pd import numpy as np The second seco	View Insert Cell		
This is one of the classic systems in non-linear differential equations. It complex behaviors as the parameters (σ, β, ρ) are varied, including what solutions. The system was originally developed as a simplified mathematication atmospheric convection in 1963. Welcome to the This Notebook Server wat WARNING Don't rely on this server is hosted that Your server is hosted that To run the code below: 1. Click on the cell to se 2. Press SHIFT+ENTER A full tutorial for using the Imatplotlib inline import pandas as pd import numpy as np			
Complex behaviors as the parameters (σ, β, ρ) are varied, including what solutions. The system was originally developed as a simplified mathematamospheric convection in 1963. Welcome to the This Notebook Server was WARNING Don't rely on this sen Your server is hosted that Your server is hosted that 1. Click on the cell to se 2. Press SHIFT+ENTER A full tutorial for using the Imatplot1lib inline import pandas as pd import numpy as np	_	ż = -	$-\beta z + xy$
This Notebook Server was WARNING Don't rely on this sen Your server is hosted that To run the code below: 1. Click on the cell to st 2. Press SHIFT+ENTER A full tutorial for using the Imatel of the sen Imatel of the sen	🔁 Jupyter	complex behaviors as the parameters (σ , β , ρ) solutions. The system was originally developed) are varied, including what a
WARNING 300 Don't rely on this server is hosted that 12 Your server is hosted that 0 Run some Python (0 To run the code below: 1. Click on the cell to se 2. Press SHIFT+ENTER Afull tutorial for using the Imatplotlib inline Imatplotlib inline Import pandas as pd Import numpy as np	Welcome to the		
WARNING angle 300 Don't rely on this server is hosted that a 12 rour server is hosted that β 2.6 P 28 Run some Python of to run the code below: 1. Click on the cell to se 1. Click on the cell to se 2. Press SHIFT+ENTER A full tutorial for using the import pandas as pd import numpy as np Imate of the cell to se	his Notebook Server was		
Don't rely on this server is hosted that σ 10 Your server is hosted that β 2.6 Run some Python (γ 2.8 To run the code below: 1. Click on the cell to st 2. 1. Click on the cell to st 2. Press SHIFT+ENTER A full tutorial for using the Imatplotlib inline Imatplotlib inline Imatplotlib inline Imatplotlib inline Imatplotlib inline	WARNING	angle	308.2
Your server is hosted that β β 2.6 β Corun the code below: 1. Click on the cell to se 2. Press SHIFT+ENTER A full tutorial for using the Imatplotlib inline Import pandas as pd Import numpy as np		max_time	12
P 28 Run some Python (fo run the code below: 1. Click on the cell to se 2. Press SHIFT+ENTER full tutorial for using the matplotlib inline mport pandas as pd mport numpy as np		0	10
Run some Python of To run the code below: 1. Click on the cell to se 2. Press SHIFT+ENTER A full tutorial for using the Imatplotlib inline Import pandas as pd Import numpy as np	our server is hosted than	β	2.6
To run the code below: 1. Click on the cell to se 2. Press SHIFT+ENTER A full tutorial for using the Imatplotlib inline Import pandas as pd Import numpy as np		ρ	28
1. Click on the cell to se 2. Press SHIFT+ENTER A full tutorial for using the Imatplotlib inline Import pandas as pd Import numpy as np	Run some Python		
2. Press SHIFT+ENTER full tutorial for using the matplotlib inline mport pandas as pd mport numpy as np	o run the code below:		
A full tutorial for using the Imatplotlib inline import pandas as pd import numpy as np			
Imatplotlib inline import pandas as pd import numpy as np			
import pandas as pd import numpy as np	A full tutorial for using the		
import numpy as np	matplotlib inline		
	import pandas as pd		



Lab 8 - Simple Harmonic Oscillator states

```
Problems from Chapter 12
```

In [1]: from numpy import sqrt
from qutip import *

Define the standard operators

```
In [2]: N = 10 # pick a size for our state-space
a = destroy(N)
```

n = a.dag()*a

Problem 12.1:

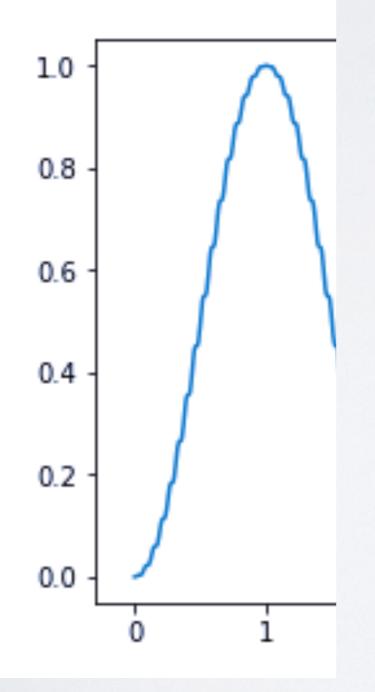
In [3]: a*a.dag() - a.dag()*a

Out [3]: Quantum object: dims = [[10], [10]], shape = (10, 10), type = oper, isherm = True 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.000 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.000 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.000 0.0 0.0 0.0 0.0 -9.00.0 0.0 0.0 0.0 0.0 0.0 0.0

Inline Graphics

In [51]: plt.plot(result

Out[51]: [<matplotlib.li



Markdown

Title Body

Title

Body

GitHub/Gist

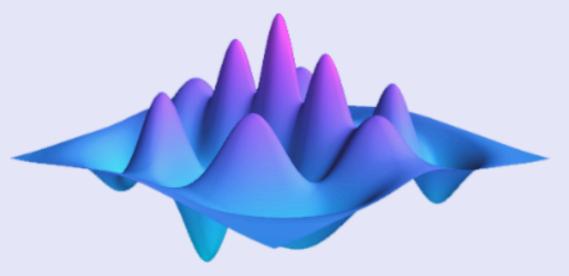


amcdawes / Chapter 10 - Positi Momentum_blank.ipynb

Created a year ago

Chapter 10 - F

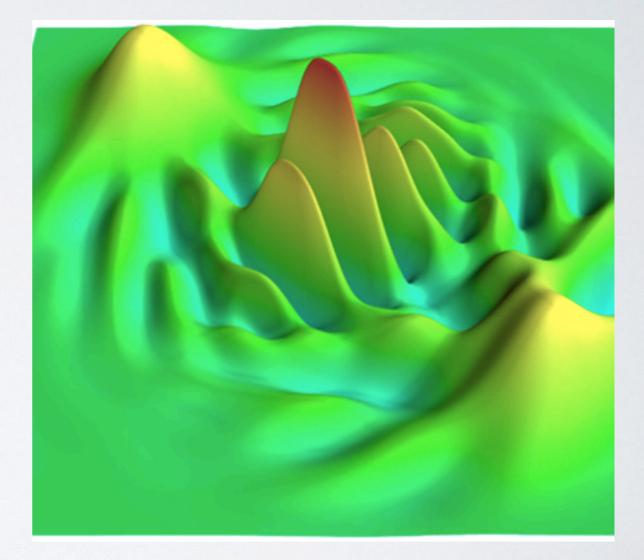
We can start using sympy t



QuTiP Quantum Toolbox in Python

QUTIP

- Not a toy Students learn in a full-strength computing framework
- Convenient object definitions
- Many existing examples



STANDARD OBJECTS

Analogous to:

from numpy import pi
from scipy.constants import speed_of_light

- QuTip defines standard quantum objects
- "objects" in the programming sense, not the physical sense
- Same QM objects we see in the textbook

Pauli matrix

Basis states

- Quantum object: dims = [[2], [2]], shape = (2, 2), ty Out[5]: $\begin{pmatrix} 1.0 & 0.0 \\ 0.0 & -1.0 \end{pmatrix}$
- qutip.basis(2,0) In [7]:

qutip.sigmaz()

In [5]:

- Out[7]: Quantum object: dims = [[2], [1]], shape = (2, 1), ty $\begin{pmatrix} 1.0\\0.0 \end{pmatrix}$
- qutip.thermal_dm(5,2) In [8]:

0.0

Quantum object: dims = [[5], [5]], shape = (5, 5), ty Out[8]: 0.384 0.0 0.0 0.0 0.0 0.0 0.256 0.0 0.0 0.0 0.171 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.114 0.0

0.0

0.0

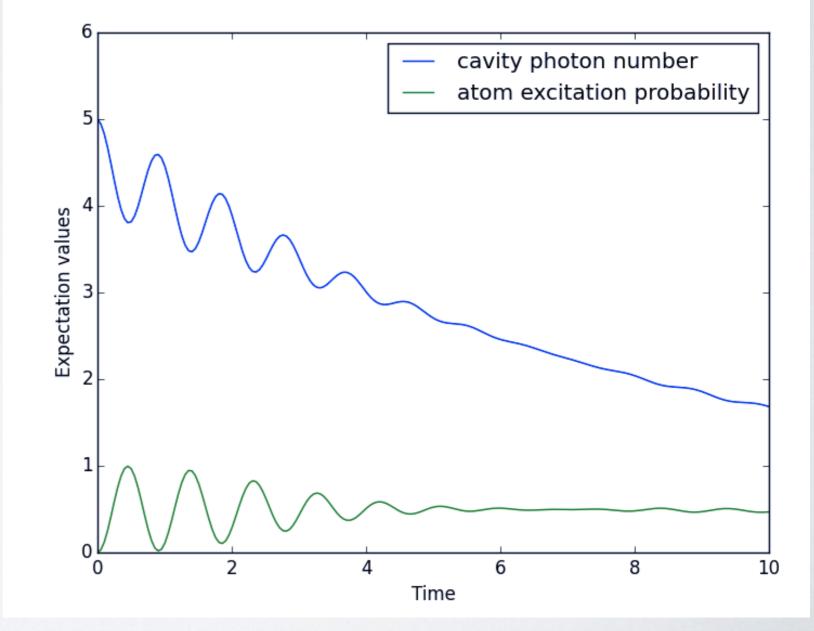
0.076

0.0

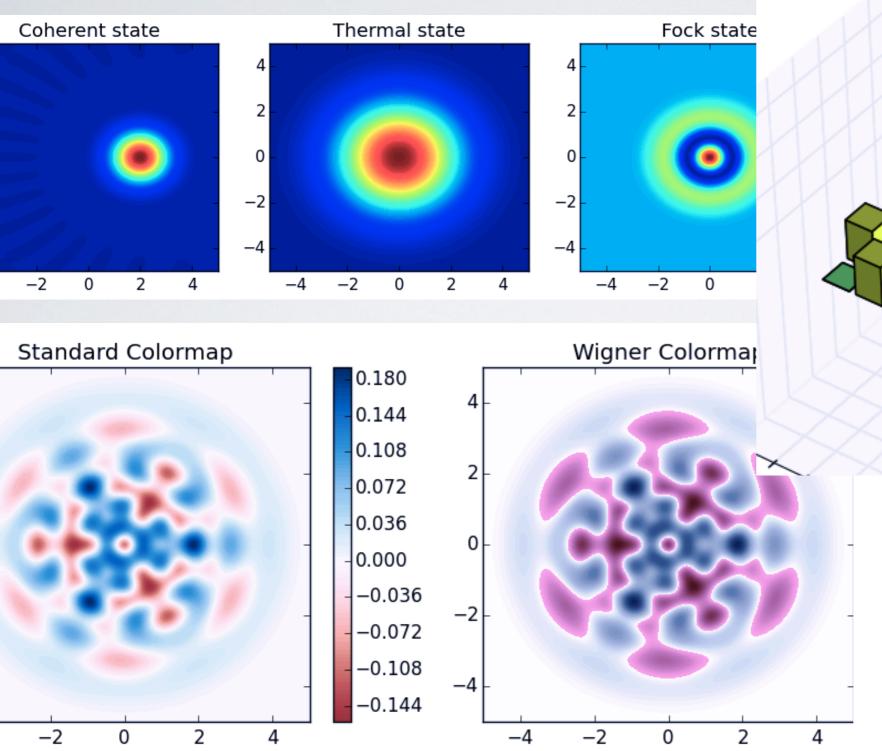
Density matrix

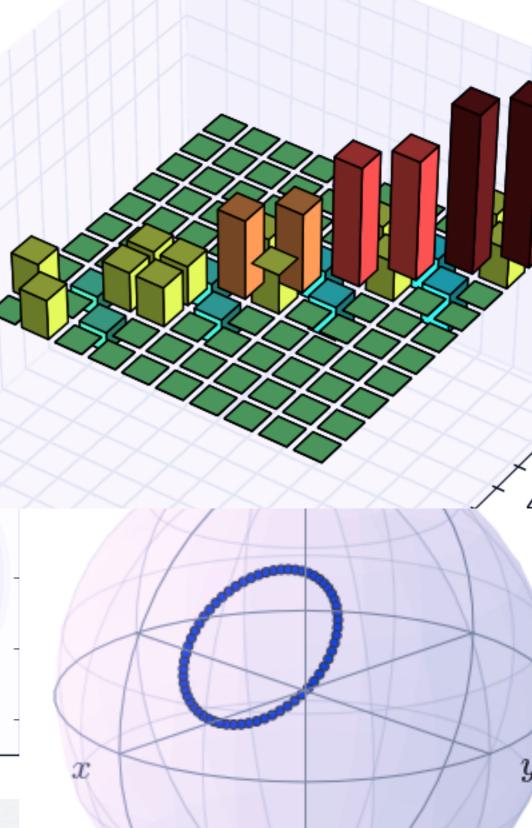
POWERFUL SOLVERS

- Schrödinger
- Master-Equation
- Monte-Carlo



VISUALIZATION TOOLS





COURSE FORMAT

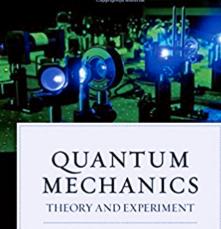
AUDIENCE

- Junior/Senior Majors
- No CS experience req'd
- 50% had intro-level C+
- I4-I8 students
- 3x 65-min & a 3-hr lab



TEXTBOOK

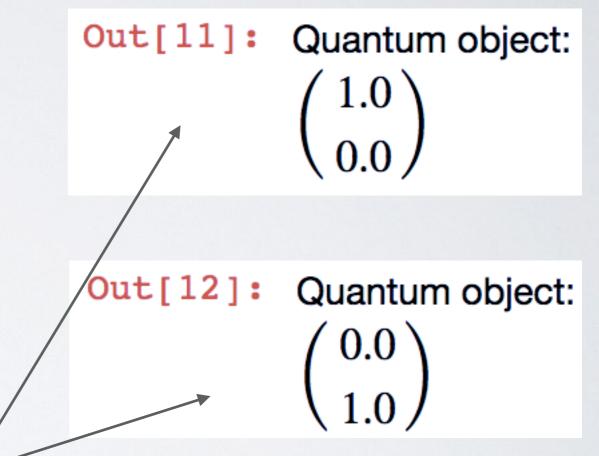
- Mark Beck, Quantum Mechanics: Theory and Experiment
- Matrix-mechanics—an approach to quantum mechanics based on linear algebra aka "Dirac Notation"



MARK BECK

TWO-STATE SYSTEMS

- single spin in magnetic field
- hydrogen atom (ground and excited state)
- photon polarization



represented by 2-element vectors

OPERATOR-AS-MATRIX

Rotation matrix

In [12]: Rp(1.3) Out[12]: Quantum object: dims = [[2], [2]], $\begin{pmatrix} 0.267 & -0.964 \\ 0.964 & 0.267 \end{pmatrix}$

	In [10]:	ShvLR*Rp(pi/4)*ShvLR.dag()	
Basis change	Out[10]:	Quantum object: dims = [[2], [2]], shape = (2, 2), $\begin{pmatrix} (0.707 - 0.707j) & 0.0 \\ 0.0 & (0.707 + 0.707j) \end{pmatrix}$	

Easily compare computation to pen & paper

CHAPTER-SPECIFIC

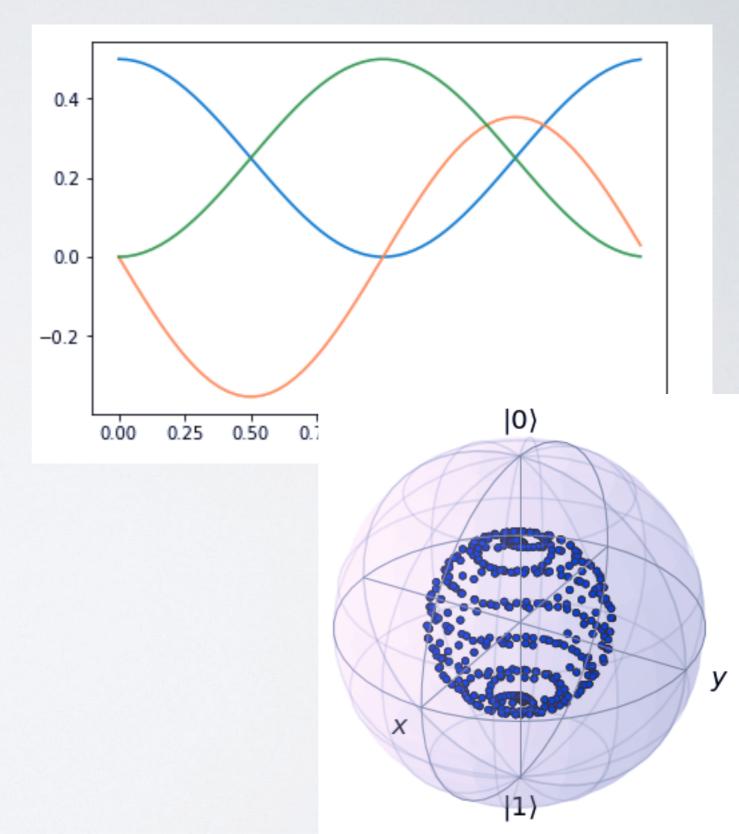
- One notebook per chapter
- Definitions and techniques relevant to that content
- Solved problems, picked from end-of-chapter)
- Re-created examples turn book notation into code



- Larger (multi-hour) exploration of a topic
- Follows chapter content
- include chapter problems
- in addition to single-photon experiments

NUMERICAL EXPERIMENTS

- Use solvers to explore advanced dynamics
- Higher-order problems not tractable by hand
- Demo Lab 7



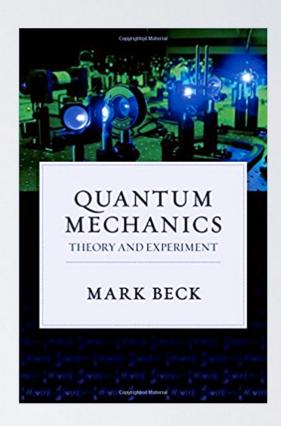
KEY POINTS

- Use a real-world computing framework (many other field-specific examples exist)
- Re-create examples to reinforce what students see in other references
- Don't be afraid to give fully-worked examples
- Encourage tinkering

FOR MORE:



- Aaron Titus: Using Jupyter Notebook for Computational Thinking, Monday 8pm (FB03)
- Mark Beck, Richtmyer Award Lecture, Tuesday 10:30-11:30
- Partnership for Integration of Computation into Undergraduate Physics: picup.org





THANKYOU

Andrew M.C. Dawes @drdawes amcdawes.com https://github.com/amcdawes/QMlabs

Credits:

- Pacific Univ., Murdock Trust, RCSA, NSF logos used with permission
- <u>Jupyter & QuTiP</u> open source projects
- Images and logo from QuTiP documentation, QuTiP is: J. R. Johansson, P. D. Nation, and F. Nori: "QuTiP 2: A Python framework for the dynamics of open quantum systems.", Comp. Phys. Comm. **184**, 1234 (2013) [DOI: 10.1016/j.cpc.2012.11.019].