TEACHING QUANTUM MECHANICS WITH PYTHON

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

Pacific University Oregon
RESEARCH CORPORATION for SCIENCE ADVANCEMENT
M.J. MURDOCK CHARITABLE TRUST
NSF

jupyter

AND

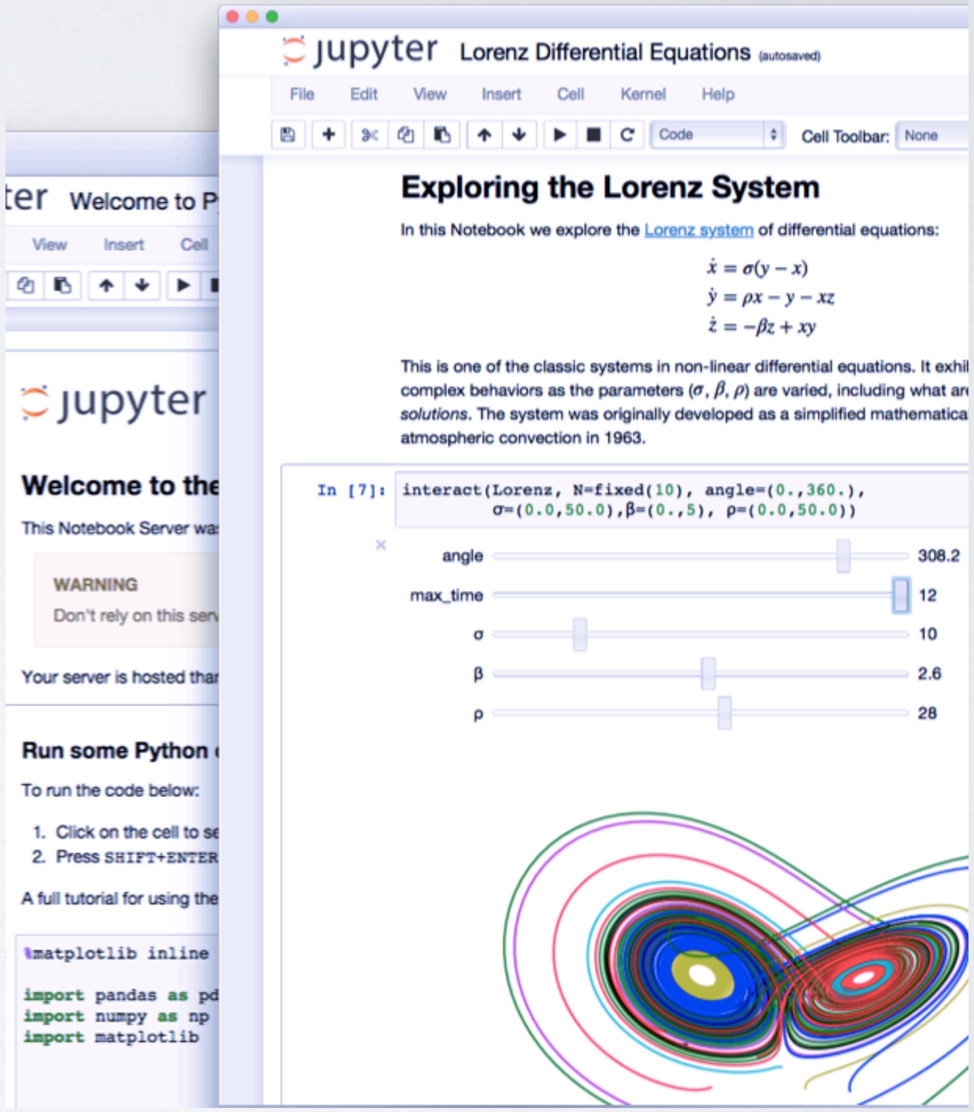
QuTiP

Quantum Toolbox in Python

…and Github

jupyter

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



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
```

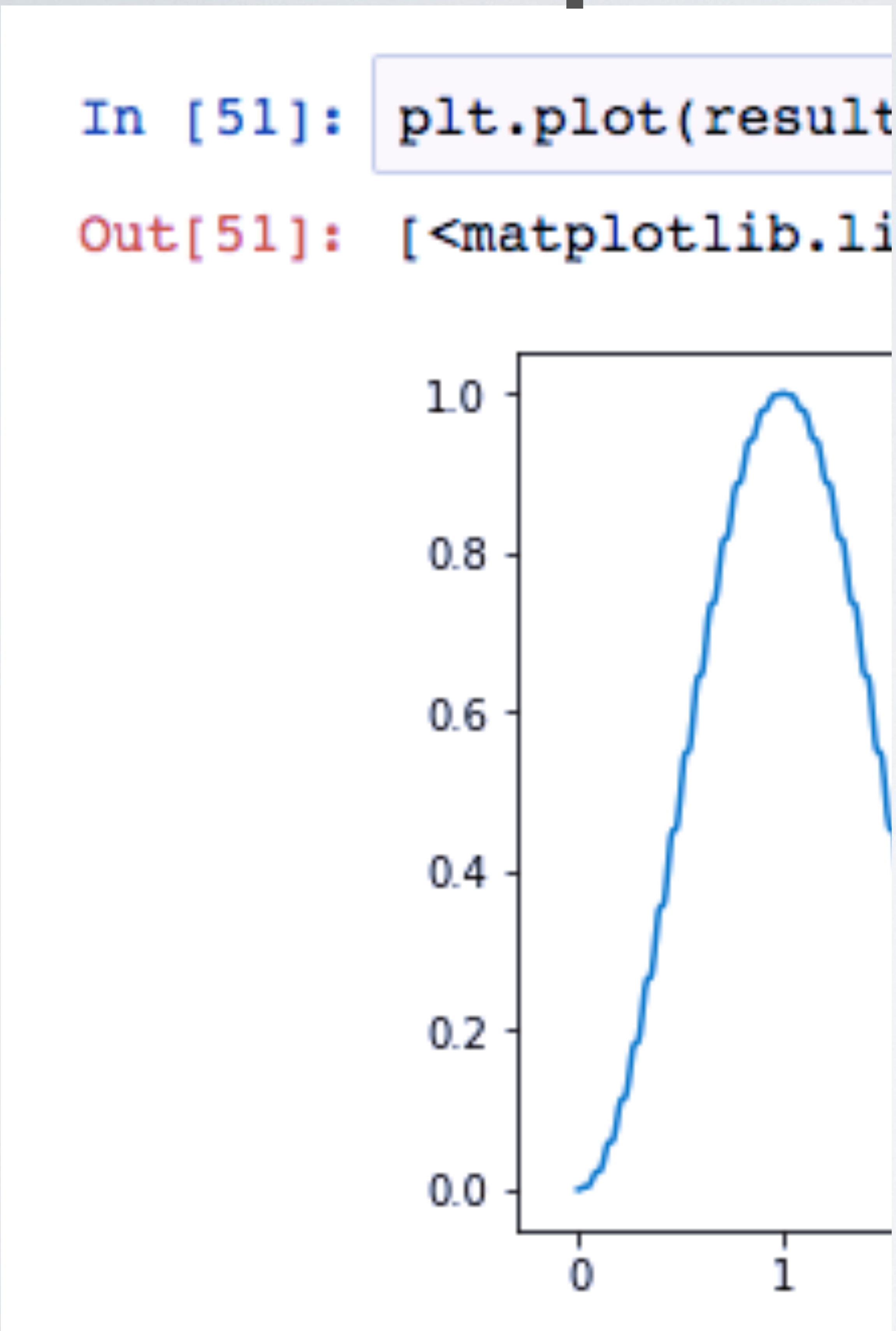
Problem 12.1:

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

Out[3]: Quantum object: dims = [[10], [10]], shape = (10, 10), type = oper, isherm = True

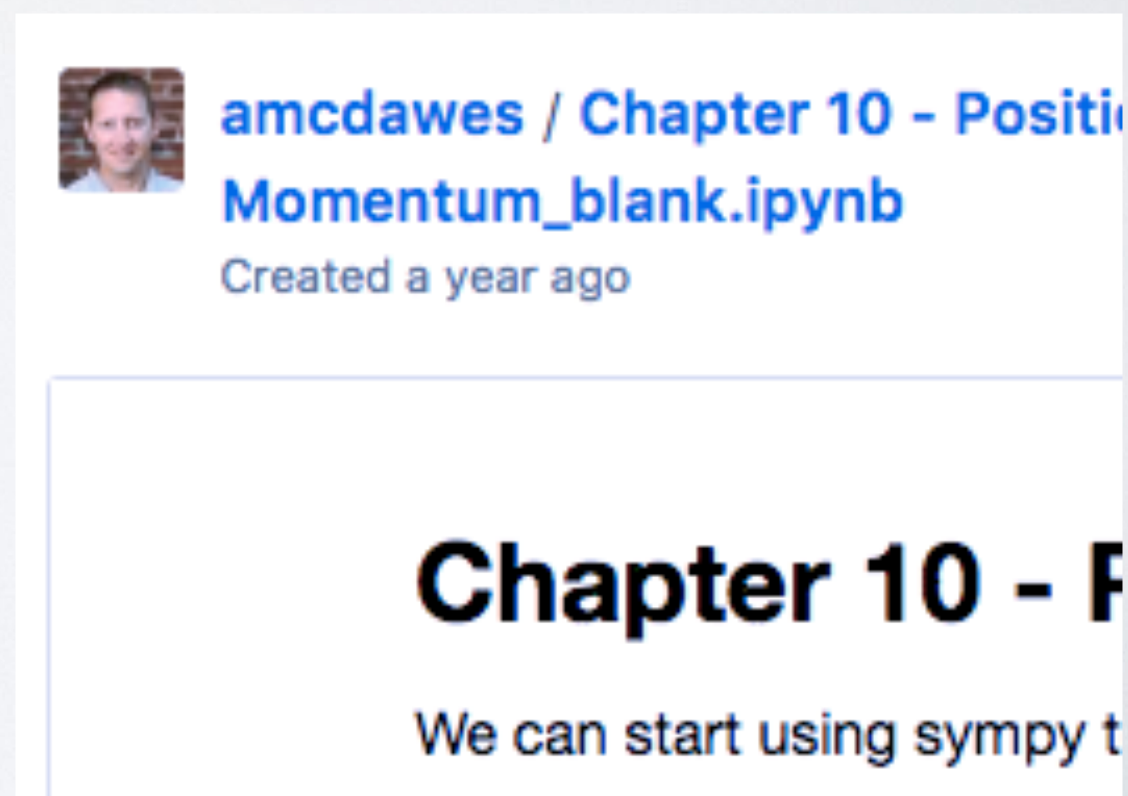
1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	1.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	1.000	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	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	−9.0

Inline Graphics



Markdown

GitHub/Gist

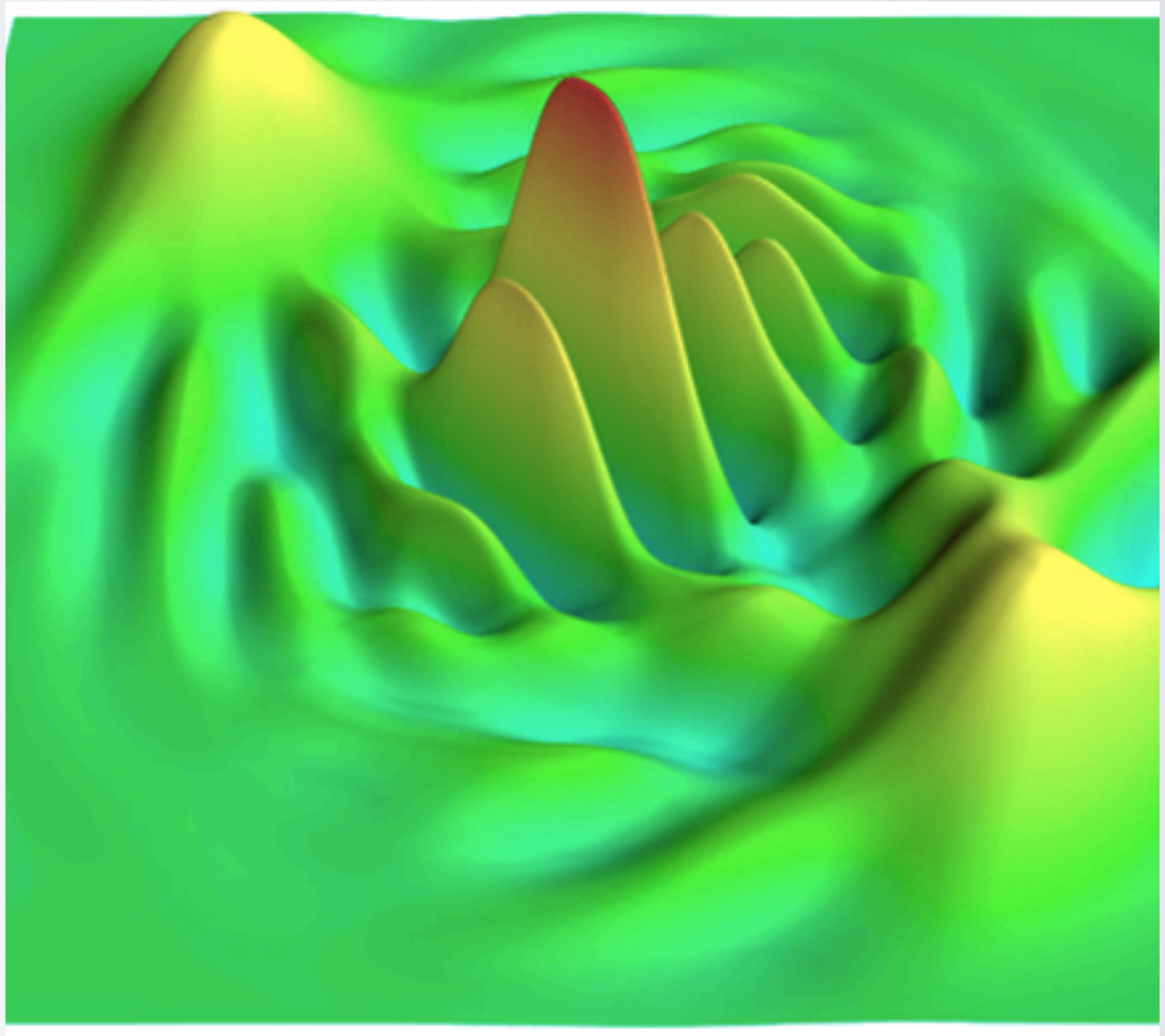


QuTiP

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

```
In [5]: qutip.sigmaz()
```

Out[5]: Quantum object: dims = [[2], [2]], shape = (2, 2), ty

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

Basis states

```
In [7]: qutip.basis(2,0)
```

Out[7]: Quantum object: dims = [[2], [1]], shape = (2, 1), ty

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

Density matrix

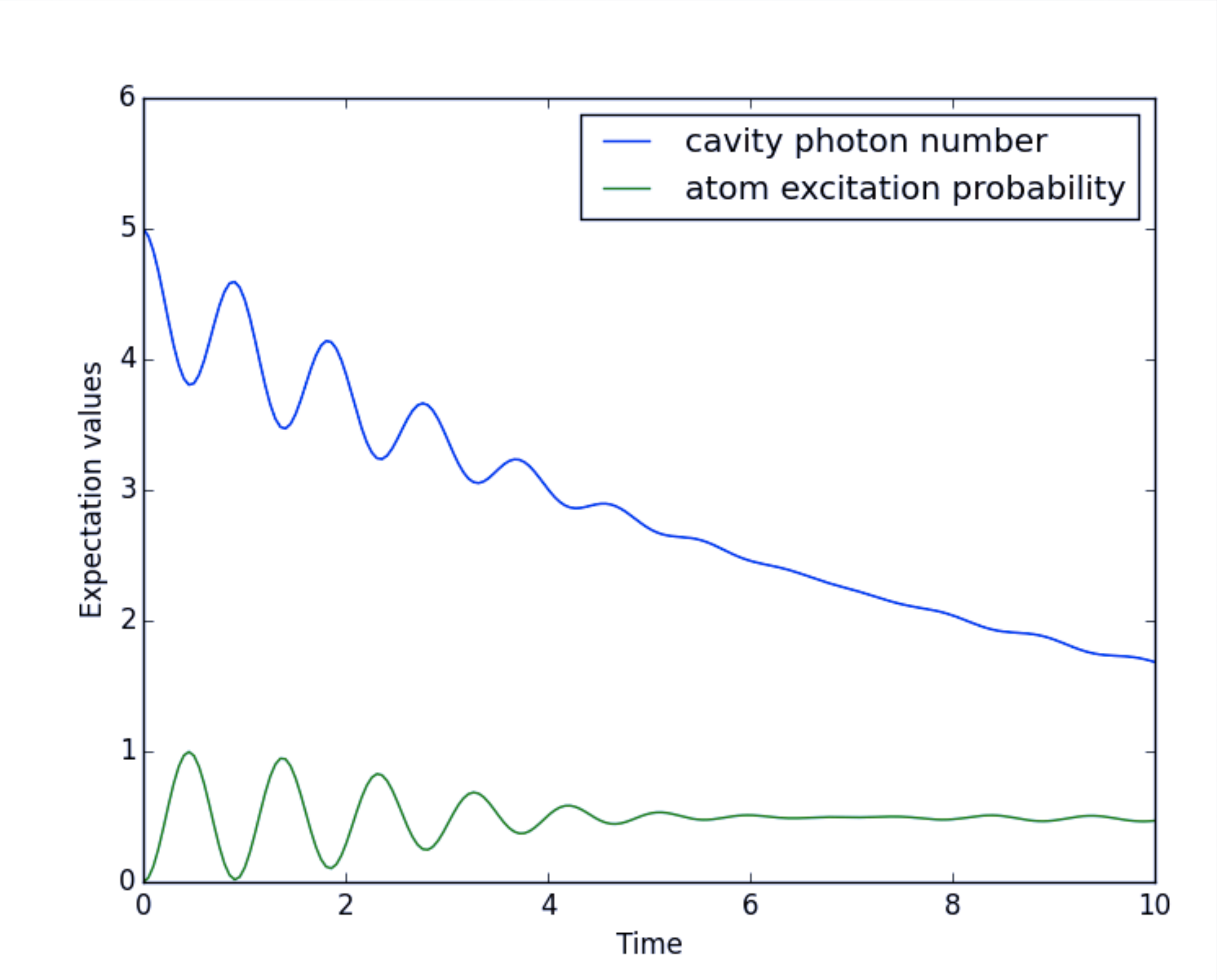
```
In [8]: qutip.thermal_dm(5,2)
```

Out[8]: Quantum object: dims = [[5], [5]], shape = (5, 5), ty

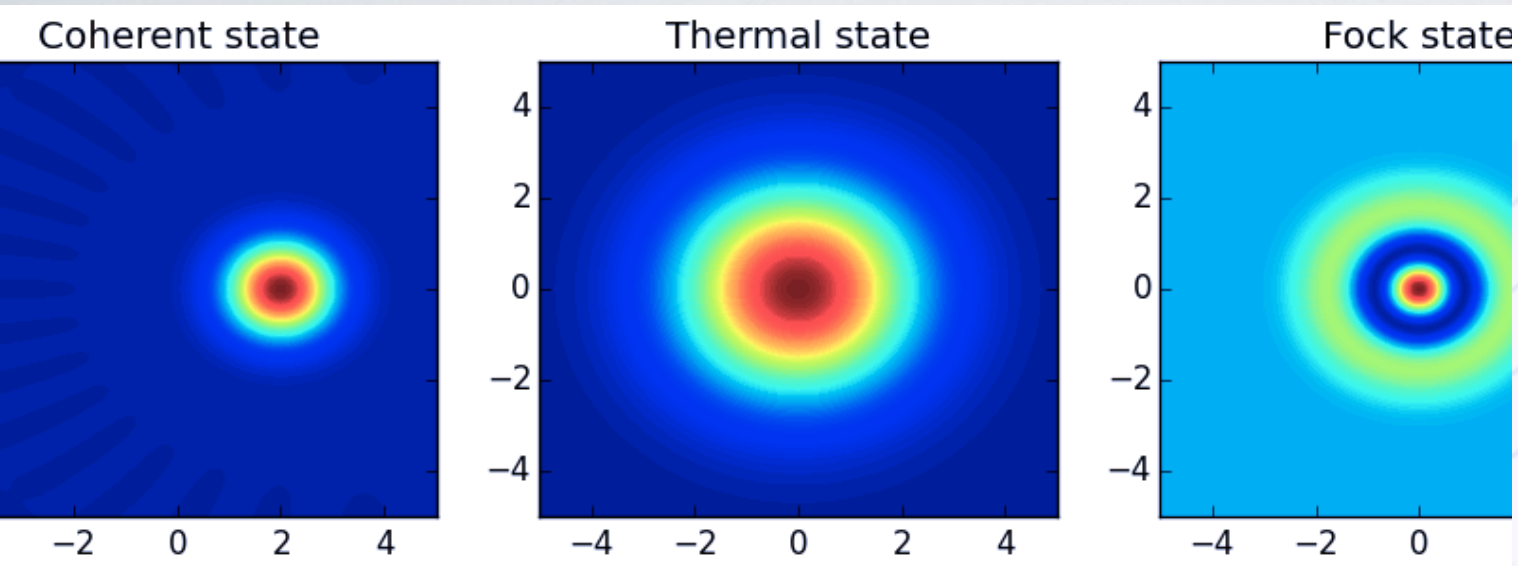
$$\begin{pmatrix} 0.384 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.256 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.171 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.114 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.076 \end{pmatrix}$$

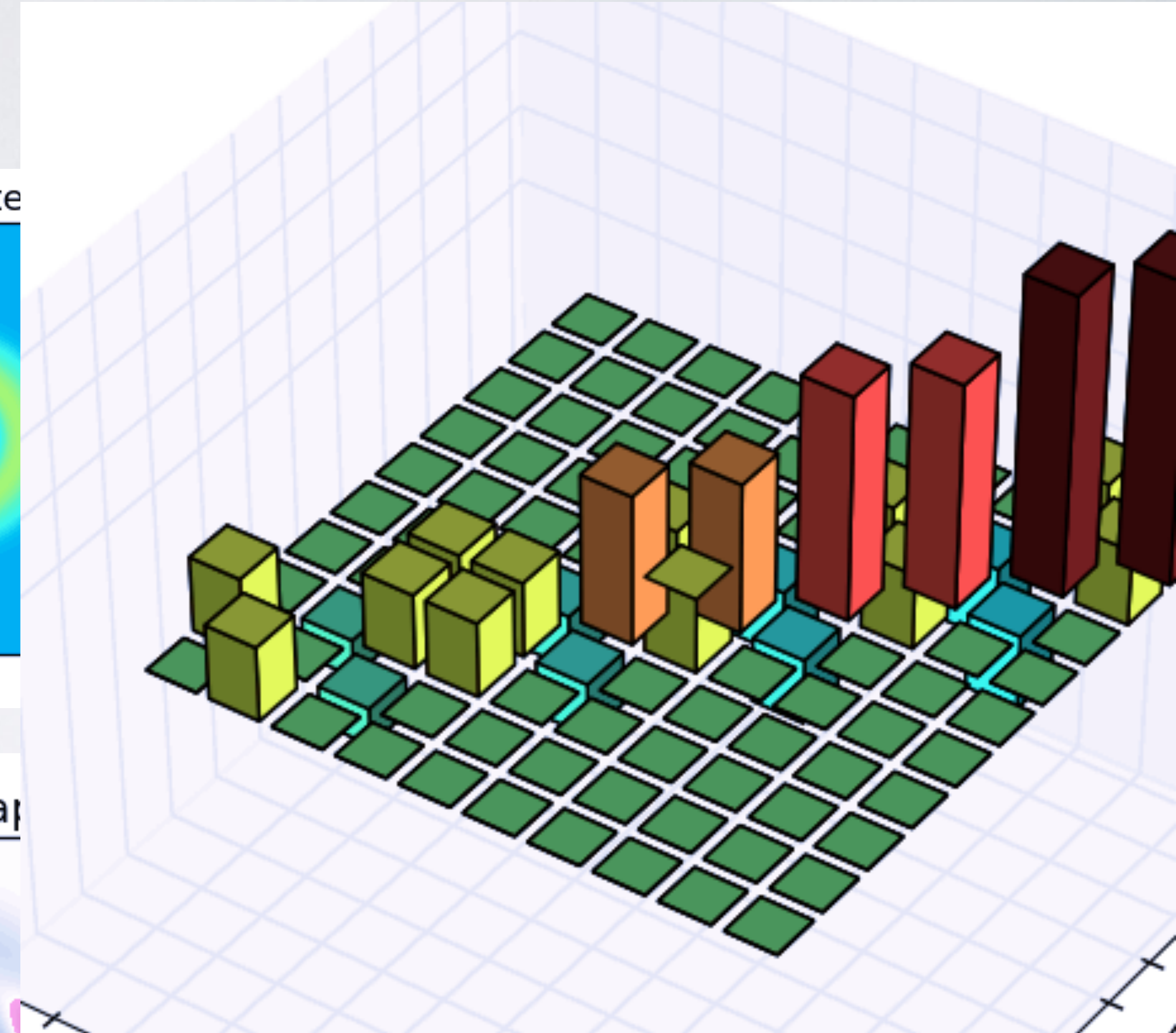
POWERFUL SOLVERS

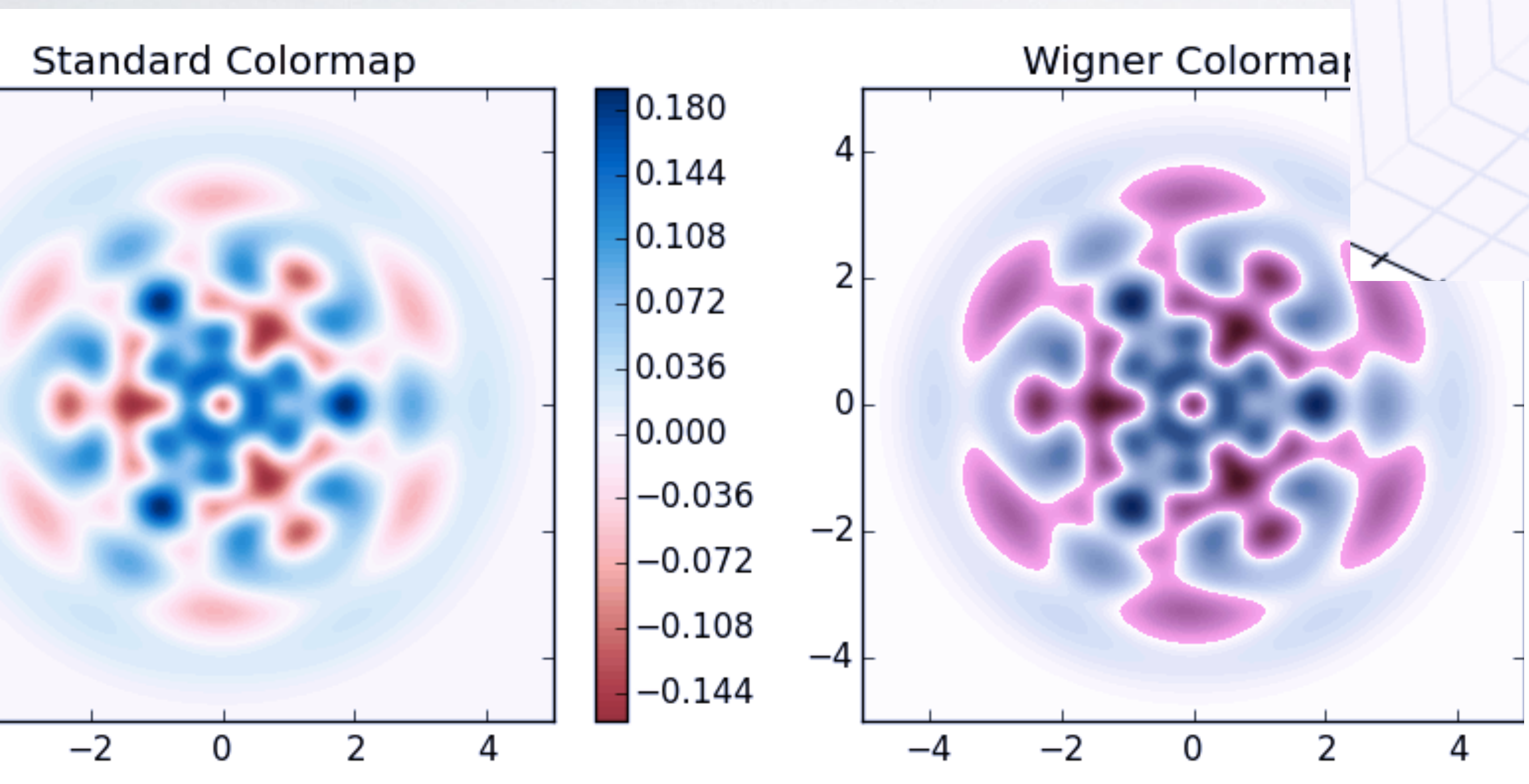
- Schrödinger
- Master-Equation
- Monte-Carlo

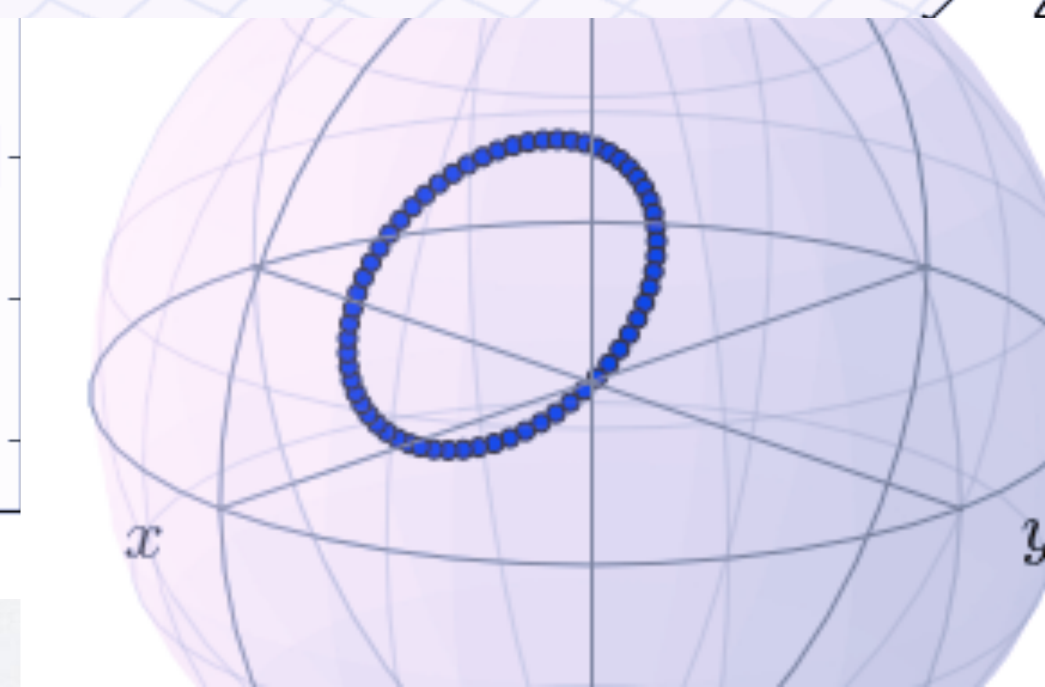


VISUALIZATION TOOLS









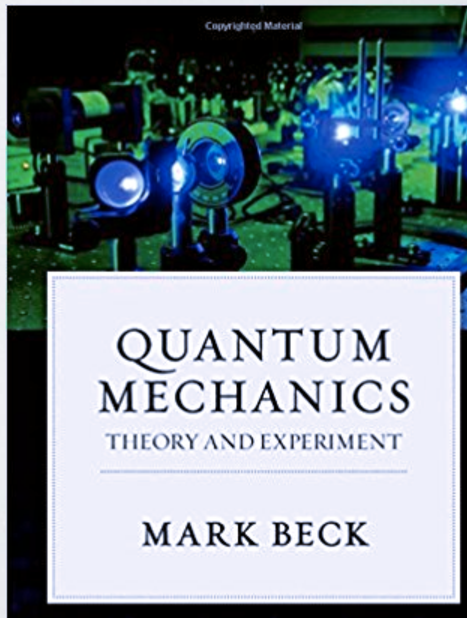
COURSE FORMAT

AUDIENCE

- Junior/Senior Majors
- No CS experience req'd
- 50% had intro-level C++
- 14-18 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"*



TWO-STATE SYSTEMS

- single spin in magnetic field
- hydrogen atom (ground and excited state)
- photon polarization
- represented by 2-element vectors

Out[11]: Quantum object:

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

Out[12]: Quantum object:

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

OPERATOR-AS-MATRIX

Rotation matrix

```
In [12]: Rp(1.3)
```

Out[12]: Quantum object: dims = [[2], [2]],

$$\begin{pmatrix} 0.267 & -0.964 \\ 0.964 & 0.267 \end{pmatrix}$$

Basis change

```
In [10]: ShvLR*Rp(pi/4)*ShvLR.dag()
```

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

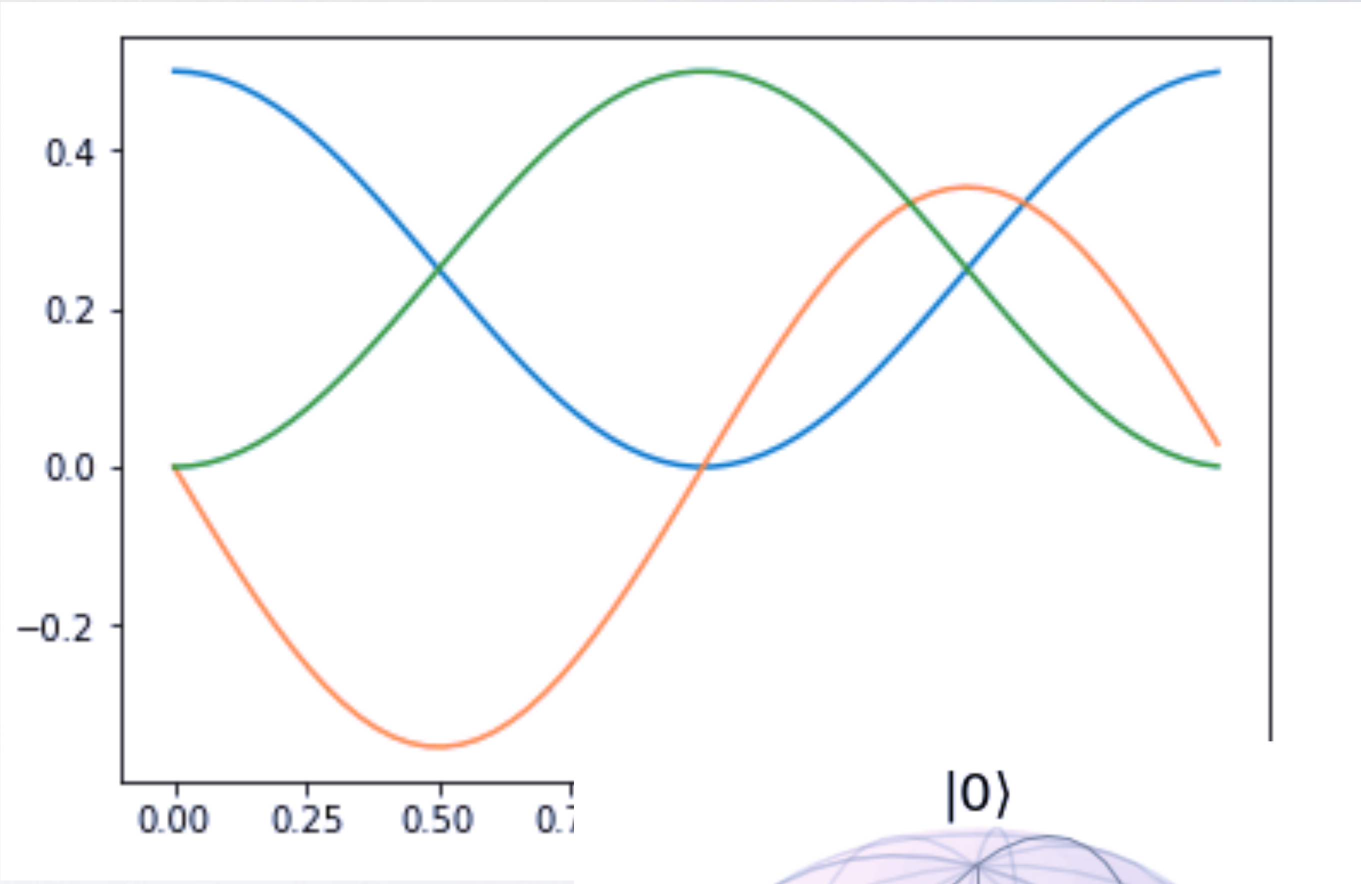
- One notebook per chapter
- Definitions and techniques relevant to that content
- Solved problems, picked from end-of-chapter)
- Re-created examples turn book notation into code

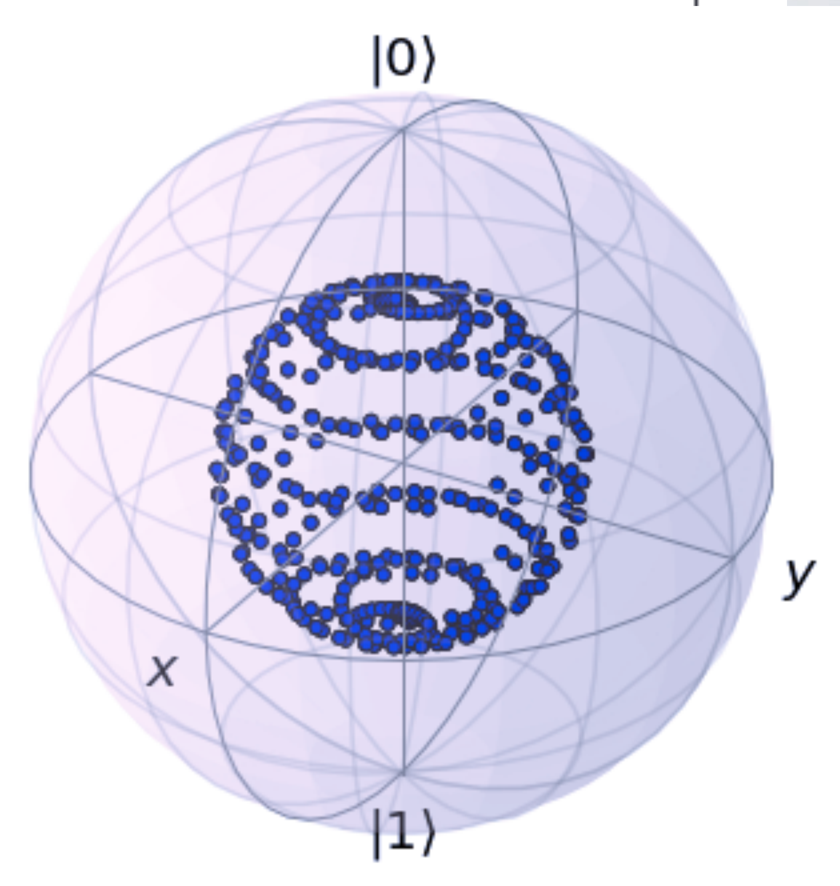
LABS

- 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



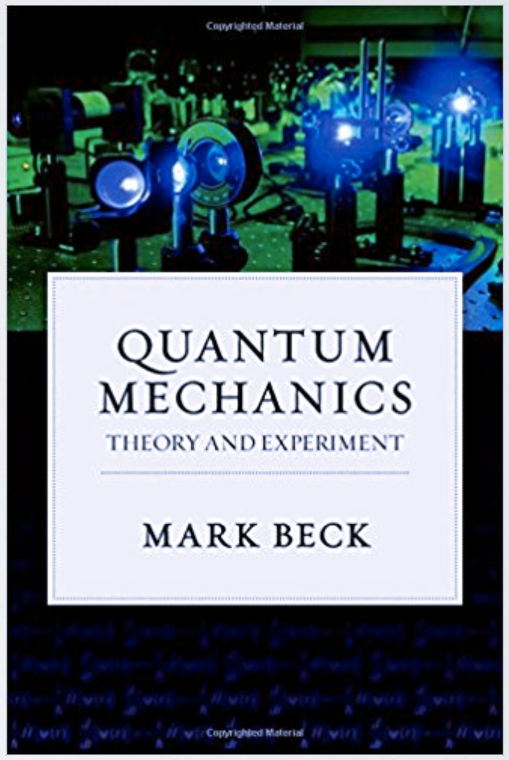


KEY POINTS

- 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



THANK YOU

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

Credits:

- Pacific Univ., Murdock Trust, RCSA, NSF logos used with permission
- Jupyter & QuTiP 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].