Exploration of Squeezed Spin States in Python

Kalina Peneva

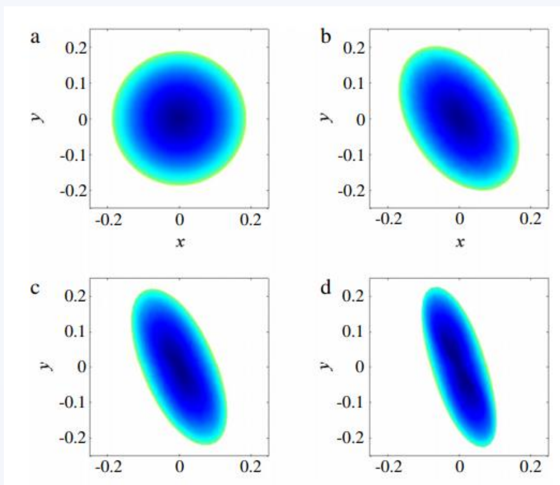
Quantum Science and Technology Workshop at Bellaire High School

SCAN ME

GitHub

INTRODUCTION

- An exploration of visualizing unsqueezed and squeezed spin states in python and comparing them.
- The image below encapsulates the process of squeezing a spin state [1]. The two axes in the graphs below show the values of the spin projection of a spin state.
- The graph represents the range of measurement results for the projection spins



- Motivation: squeezed states can help develop better measurement methods through reducing measurement uncertainty

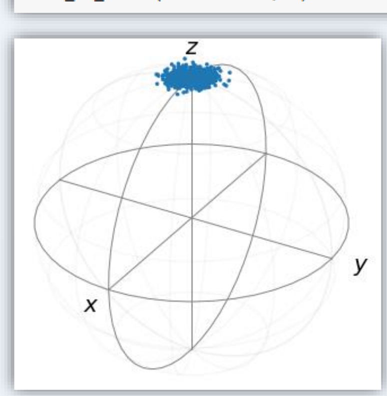
OBJECTIVES

- Explore the visualization of spin state realizations and rotations on a Bloch sphere in Python
- Utilize One Axis Twisting (OAT) and Two Axis Twisting (TAT) to produce squeezed spin states [2]
- Measure and visualize the utility of squeezed spin states relative to unsqueezed spin states
- Goal: to produce a squeezed spin state with the thinnest center; essentially stretching the spin state as much as possible to achieve a narrower distribution

METHODS

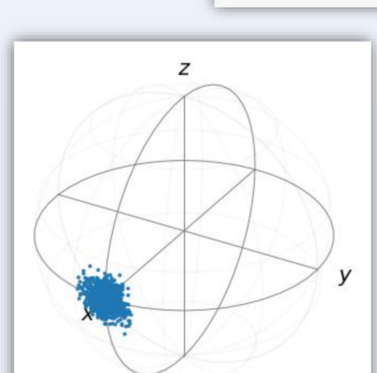
Visualizing an Unsqueezed Spin State

```
#Realization of spin up function on Bloch Sphere
N = 200 #number of atoms
allSresults = [all_S(spinup(N)) for i in range(1000)] #1000 is number of realizations for the set of 200 atoms
show_on_bloch(allSresults, N) #visualization of spin up realization
```



- Establishment of spin up state
- Foundation for all more complex spin states

```
#Rotation along the Y Axis
N = 200
allSresults = [all_S(spinup(N)) for i in range(1000)]
allSresults_X = [single_spin_rotation_along_y(i, -np.pi/2) for i in allSresults]
show_on_bloch(allSresults_X, N)
```



- Rotation along Y Axis
- Easier visualization of the circular shape when rotated
- Easier comparison to squeezed spin state along this axis

METHODS

Visualizing a Squeezed Spin State

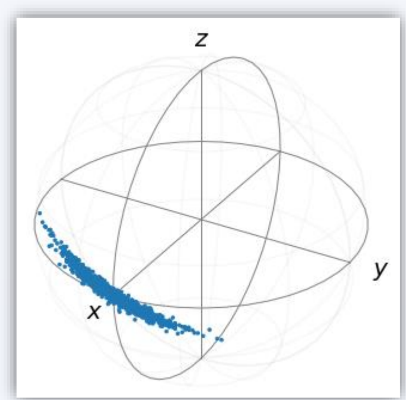
➢ **Writing Evolution Functions**

```
def sz2_evolution(spin_ensemble, t):
  B = all_S(spin_ensemble)[2]
  amount = B * t
  spin_ensemble = [single_spin_rotation_along_z(atom, amount) for atom in spin_ensemble]
  return spin_ensemble
```

Realization of spin ensemble after evolving under Hamiltonian for t

➢ **One Axis Twisting (OAT)**

```
#OAT using sz2_evolution
N = 200
t = 5 / N

final_results = []
all_realizations = []
for realization_index in range(1000):
  spin_ensemble = spinup(N) #establishing initial state of spin_ensemble
  spin_ensemble = [single_spin_rotation_along_y(atom, -np.pi/2) for atom in spin_ensemble] #preparinig spin_ensemble along x-direction
  spin_ensemble = sz2_evolution(spin_ensemble, t) #application of sz2_evolution on the spin_ensemble
  final_results.append(all_S(spin_ensemble)) #calculate all_S (total projection of spin_ensemble)
  all_realizations.append(spin_ensemble) #repeat this process 1000 times by storing all results in a new array

show_on_bloch(final_results, N)
```

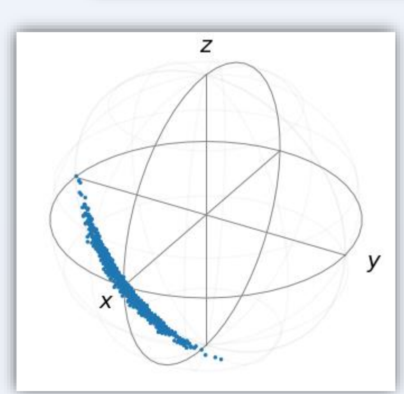


- Sz_2 evolution of spin state
- Other variations with less utility developed in code

➢ **Two Axis Twisting (TAT)**

```
#Defining the perameters of the function TAT_Delta_t(...)
def TAT_Delta_t(spin_ensemble, Delta_t):
  ...
N = 200
t = 5 / N
P = 10
Delta_t = t / (2 * P)
```

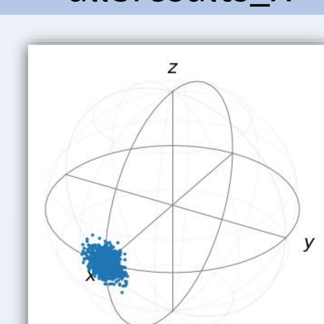
```
# This block is meant to test the funciton TAT_Delta_t(...)
final_results = []
for realization_index in range(1000):
  spin_ensemble = spinup(N)
  spin_ensemble = [single_spin_rotation_along_y(atom, -np.pi/2) for atom in spin_ensemble]
  for _ in range(P):
    spin_ensemble = sz2_evolution(spin_ensemble, Delta_t)
    spin_ensemble = sy2_evolution(spin_ensemble, -Delta_t)
  final_results.append(all_S(spin_ensemble))

show_on_bloch(final_results, N)
```



- Use of two evolution functions
- Twisting along two axes
- Results in spin state rotated 45°

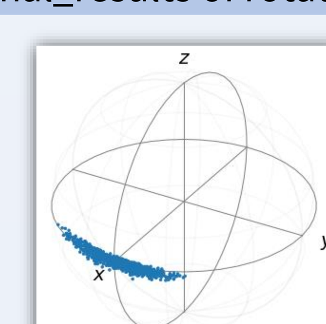
➢ **Setting up Comparison**

allSresults_X



Standard unsqueezed spin state

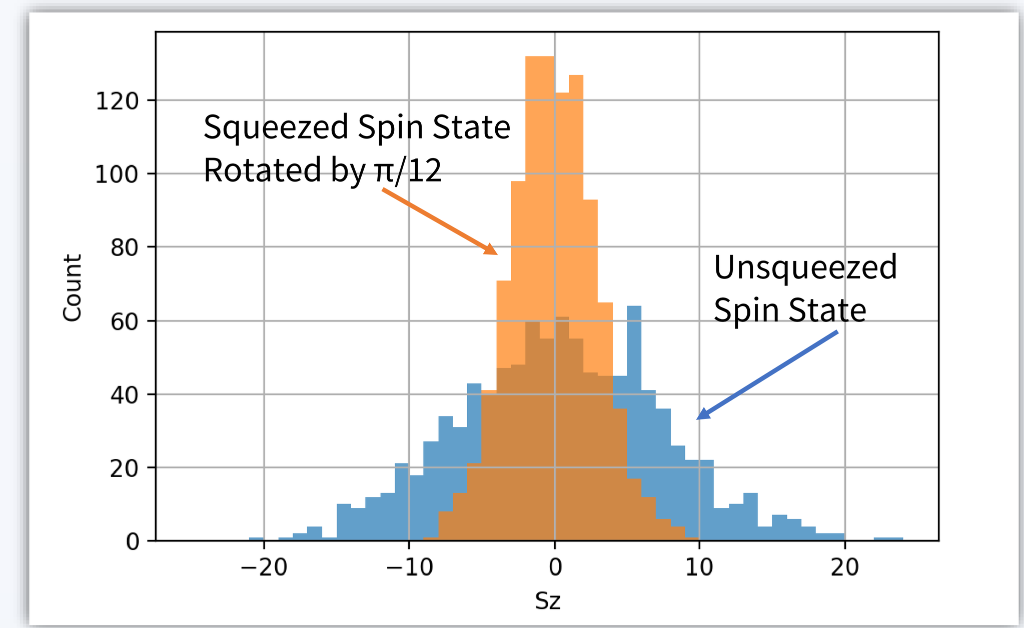
Final_results of rotated OAT



Rotated by π/12 to produce most narrow spin state

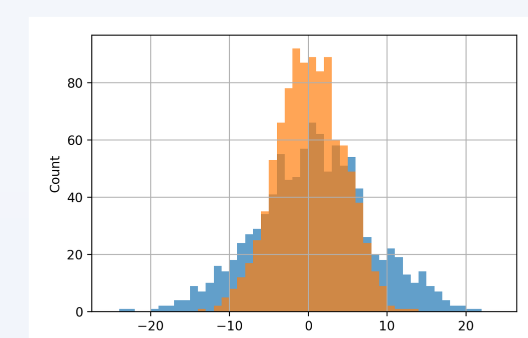
RESULTS

Results of comparison of squeezed vs. unsqueezed state

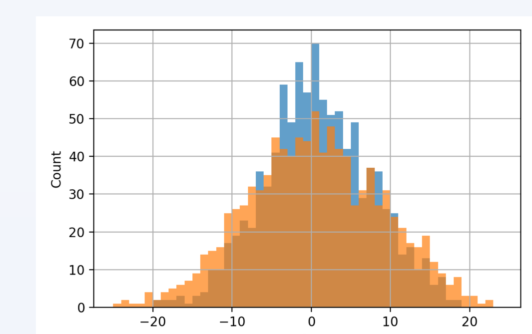


Demonstration of a narrower distribution

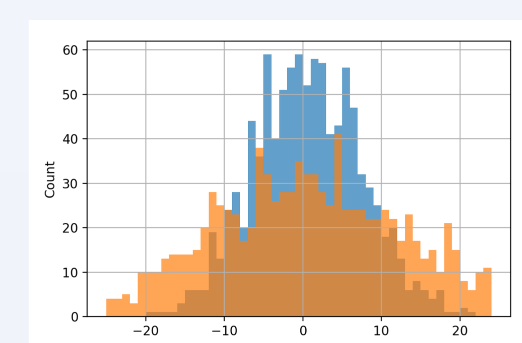
Angle of Rotation Matters



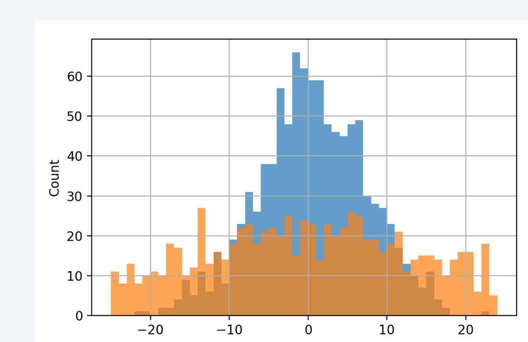
Θ = π /6



Θ = π /4

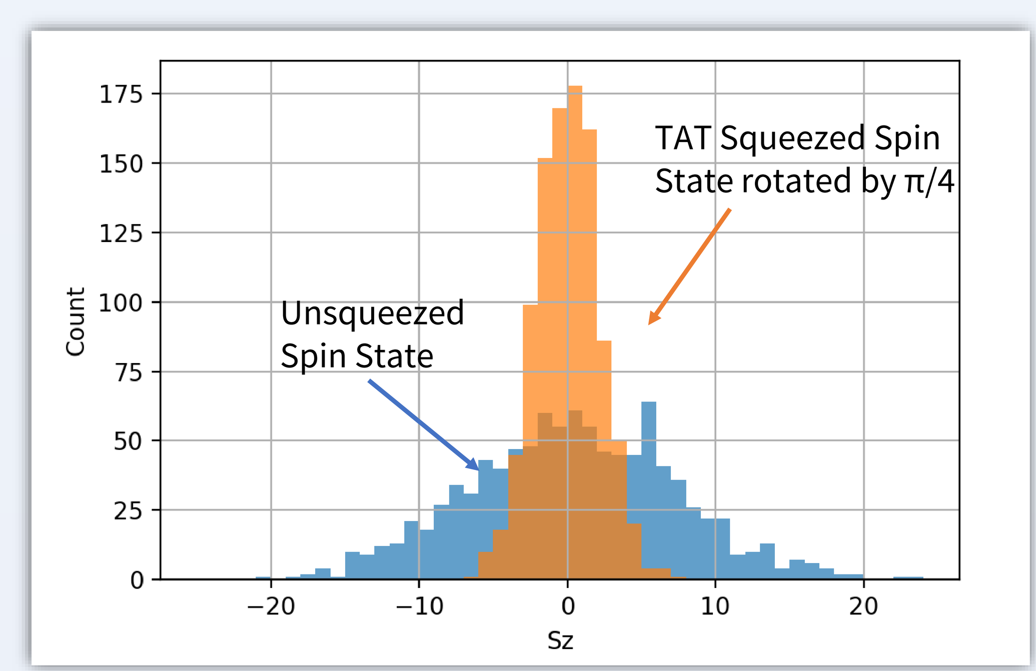


Θ = π /3



Θ = π /2

Testing the TAT Result



Squeezed Spin States produced from TAT result in a narrower distribution

CONCLUSION

- Squeezing spin states can facilitate increased accuracy in measurements as depicted through the narrower distribution seen in the results section
- Not every squeezed spin state will be useful in creating narrower distributions; in the demonstrations done, OAT spin states rotated by a larger theta were the least useful
- Two Axis Twisting (TAT) was shown to be most useful in producing narrower distributions

OUTLOOK

- Can be used as a method to detect quantum entanglement
- Exploring more variations of TAT and comparing their utility to OAT in producing useful squeezed spin states
- Utility of Python in modeling quantum systems, in this case squeezed spin states, facilitating further discussions based on more understandable material

ACKNOWLEDGEMENTS

We thank our mentors Ognjen Markovic and Zeyang Li, as well as Mr. Jimmy Newland, for their help and guidance. This work was completed as part of the Quantum Science & Technology Workshop @ Bellaire High School, supported by MIT CQE-iQuISE (Center for Quantum Engineering, Interdisciplinary Quantum Information Science and Engineering program).

REFERENCES

[1] Ma et al., “Quantum Spin Squeezing”, *Phys. Rep.* **509**, 89 (2011)

[2] Kitagawa, Masahiro, and Masahito Ueda. "Squeezed spin states." *Phys. Rev. A* **47**, 5138 (1993)