

Visualization with QuTiP

Personal Background

- Name: Yuji TAMAKOSHI
- Location: Japan, Aichi
- School: 4th year (the new academic year starts in April in Japan) undergraduate student at the University of Tokyo, Department of physical engineering
- GitHub: tamakoshi2001, <https://github.com/tamakoshi2001>
- Email: uotstudent2001@g.ecc.u-tokyo.ac.jp
- Phone: +81 70-8367-2723

Abstract

QuTiP has a lot of visualization functions, but they are sometimes not user-friendly in that colors are unchangeable and arguments are not organized. This project aims to improve them.

This project will enhance existing functions by unifying the interface and adding colorblind options for all of them. Additionally, the project will develop an animation function for the time evolution of quantum states.

The deliverable consists of unified interfaces for plot functions that have colorblind support, animation functions for the Result object, and pytests and documents for all of them.

Technical Details

1)

Problem

'visualization.py' includes many functions to visualize the results of quantum simulation. It helps users to visualize results quickly, but the parameters are not organized. This prevents users from using them intuitively. For example, you can set 'figsize' in 'plot_wigner', but not in 'plot_wigner_sphere.' Also, QuTiP has 'colorblind_safe' as an option, but only one function supports this. Here is a table that shows the problem.

functions \ arguments	fig	ax	figsize	colorblind_safe
plot_wigner_sphere	○	○		
hinton		○		○
sphereplot	○	○		
matrix_histogram	○	○		
matrix_histogram_complex	○	○		
plot_energy_levels	○	○	○	
plot_fock_distribution	○	○	○	
plot_wigner	○	○	○	
plot_wigner_fock_distribution	○		○	
plot_expectation_values	○		○	
plot_spin_distribution_2d	○	○	○	
plot_spin_distribution_3d	○	○	○	
plot_qubism	○	○	○	
plot_schmidt	○	○	○	

visualization functions and arguments (○ means the function has the argument.)

Solution

First, I will make a new file, 'color.py.' This file contains the new class, 'Color.' This has 'cmap' as a property and changes it by colorblind_safe in settings. This helps plots and animation functions in many files to change their color easily only by adding 'cmap = Color.cmap'.

Second, I will organize parameters in visualization.py. The parameters in the code will be written like this;

```
def func(
    arg0, arg1, ..., *,
    kwarg0, kwarg1, ...,
    figure, figure_options, axes, axes_options
)
```

'arg' is a positional argument and it is always necessary to use the function. 'kwarg' is a keyword argument and it is an option which depends on the function such as changing objects to plot. Figure and axes are arguments for adding Figure and Axes

objects to the function. This change makes the function easier to use by allowing users to write the required arguments without explicitly specifying them and improves readability by making the optional arguments keyword arguments.

2)

Problem

'Result' is a class for storing simulation results from any dynamics solvers. It is very useful and used everywhere, but it does not have visualization methods. Therefore, users now need to set many parameters to visualize the time evolution of the results. This is not user-friendly.

Solution

Add animation method. Here are the functions that will be added;

```
animate_hinton  
animate_matrix_histogram  
animate_fock_distribution  
animate_wigner  
animate_spin_distribution
```

I will write the code of them in a similar way to plot functions in visualization.py;

```
def animate_func(  
    Result,  
    arg0, arg1, ..., *,  
    kwarg0, kwarg1, ...,  
    figure, figure_options, axes, axes_options  
)
```

There are some useful codes to make an animation from a Result object[1][2]. I will use them as a good starting point. QuTiP aims to provide user-friendly simulation tools. Hence, all functions should be used for specified animations and should work by just adding a Result object (and some required arguments).

Schedule of Deliverables (Medium project 175 hours)

Community Bonding Period

- Set up environment.
- Start to write my blog, <https://qiita.com/tama2001> (I will continue to post during this project)
- Introduce myself to mentors.
- Discuss the project more.

Phase 1

Deliverables

- Functions that have a colorblind option and organized arguments in visualization.py
- Pytests
- Documents

Week 1: 5/29-6/4

- Add class Color and see if visualization.py can import it
- Merge plot_spin_distribution_2d and plot_spin_distribution_3d into plot_spin_distribution

Week 2-5: 6/5-7/2

- Apply colorblind_mode, unify interfaces and add pytests for the following 13 functions. Each pytest should be added at the same time as I write the function that it covers.

plot_wigner_sphere,
hinton,
sphereplot,
matrix_histogram,
matrix_histogram_complex
plot_energy_levels,
plot_fock_distribution,
plot_wigner,
plot_wigner_fock_distribution,
plot_expectation_values,
plot_spin_distribution,
plot_qubism,
plot_schmidt

Week 6: 7/3-7/9

- Write about new features and update related documents.

Week 7: 7/10-7/16

- Update tutorial jupyter notebooks to tell the effective use of new features.
- Add tutorial notebooks about the Result object

Phase 2

Deliverables

- Animation functions for the Result object
- Pytests
- Documents

Week 8-10: 7/17-8/6

- Make animation functions and pytests for the following 5 functions. Each pytest should be added at the same time as I write the function that it covers.

animate_hinton
animate_matrix_histogram
animate_fock
animate_wigner
animate_spin_distribution

Week 11: 8/7-8/13

- Add documents. They should have an explanation about new features.

Week 12: 8/14-8/20

- Add tutorial jupyter notebooks

Final Week

- Prepare for the final review and submit!!!

After the project

- Keep contributing to OSS!
- Tell others about GSoC through the blog

Development Experience

1. I have contributed to two issues [3][4][5][6], talked with others about the solution to an issue [3], and helped another contributor [7]. This shows my problem-solving skills and communication skills to complete this project. Also, I made this proposal with Eric Giguere (eric.giguere@calculquebec.ca) and Simon Cross (hodgestar@gmail.com), mentors of QuTiP. They helped me make my proposal much better.
2. I worked as a software engineer at ROBOT PAYMENT (a fintech company in Japan) remotely from home on March/2022-June/2022. I am accustomed to the conditions of GSoC.

Availability

1. I have studied as an exchange student in Ireland for almost one year and my TOEFL score is 100. I have no difficulty communicating in English.
2. My major is physical engineering. I have enough knowledge to understand QuTiP and have taken coursework to solve problems by numerical computing like LU decomposition.
3. I am participating in an exchange program that will finish before this project starts and I do not have lectures until October. I have plenty of time to spend on this project

Why this project?

I want to do this project for the following three reasons.

First, visualization is important in quantum mechanics. Quantum mechanics is often abstract, and the phenomena that arise are sometimes unintuitive. In addition, quantum states often change periodically as time evolves. Visualizing them can help us understand them well. Also, some people find physical phenomena boring because they are expressed in mathematical formulas. Visualization makes learning fun!

Second, about 4% of the world's population suffers from color blindness. This is not a small number. It is very unfortunate that so many people find it difficult to learn quantum mechanics with QuTiP, one of the main software packages available to learn it. In particular, as I mentioned earlier, visualization plays an important role to understand it. I am convinced that this difficulty should be corrected immediately.

Third, my major is physical engineering and I have studied quantum mechanics. Many of my friends are not good at programming. However, there are few tools to simulate the phenomena of it so I have seen many of them spend a great deal of time programming as we work together on assignments. I believe that they need

tools that help them learn physics without the need to become proficient in programming.

For these reasons, I believe this project should be done now and decided to work on this among others.

Appendix

- [1]<https://nbviewer.org/urls/qutip.org/qutip-tutorials/tutorials-v4/visualization/JC-model-wigner-function.ipynb>
- [2]<https://github.com/qutip/qutip/blob/ab6a0d7d7fb5464f2d53474d49721689215a08e8/qutip/ipynbtools.py#L298>
- [3]<https://github.com/qutip/qutip/issues/2100>
- [4]<https://github.com/qutip/qutip/issues/2109>
- [5]<https://github.com/qutip/qutip/pull/2113>
- [6]<https://github.com/qutip/qutip/pull/2120>
- [7]<https://github.com/qutip/qutip/issues/2098>