Justin James PHYS-265
5/18/2025 Final Project

PlasmaPy

1. Package name and function

PlasmaPy is a python package, intended for use in plasma physics and plasma education. This package allows plasma phycists to easily explore problems in fields such as fusion, with redibly available constants and built in functions related to the field and study of plasma. PlasmaPy intends to be for Plasma what AstroPy is to Astronomy.

2. Reasoning

I selected the PlasmaPy package, because I have been interning for the last 3 years at CMFX here at UMD, CMFX or the Centrifugal Mirror Fusion Experiment, is a novel fusion idea that confines plasma between two strong magnetic fields, with a central electrode with a radiating electric field, the confined plasma also rotates supersonically, improving confinement time a crucial element to achieving net zero gain. Within this experiment I was previously using matlab for some Diamagnetic Loops that I had created and installed, since my introduction to python I have been interested in possibly extending my work to include packages such as PlasmaPy. Also I think Plasma is a very interesting field.

3. Package age

PlasmaPy is around 6 years old, 2024.10.0 this is the version I have installed.

4. Maintenance

PlasmaPy is being actively maintained and is will be updated to every new python version released. PlasmaPy is being maintained by the PlasmaPy Collabaration

5. Ease of interface

PlasmaPy was very easy to install, and use on the website all of the commands and functions are readibly available with demos. I simply used !pip to install it.

6. Installation method

PlasmaPy installs with the basic !pip in Jupyter Notebook.

7. Source code availability

the source code for plasmpy is available on the there webiste plasmapy.org and on github at https://github.com/PlasmaPy/PlasmaPy

8. Usage in other packages

PlasmaPy has a list on its website of affliliated community packages, that use its code or were created in collaboration with the PlasmaPy team. One of these is tofu which is a open-source python library which uses PlasmaPy, built in plasma parameters.

9. Code example I

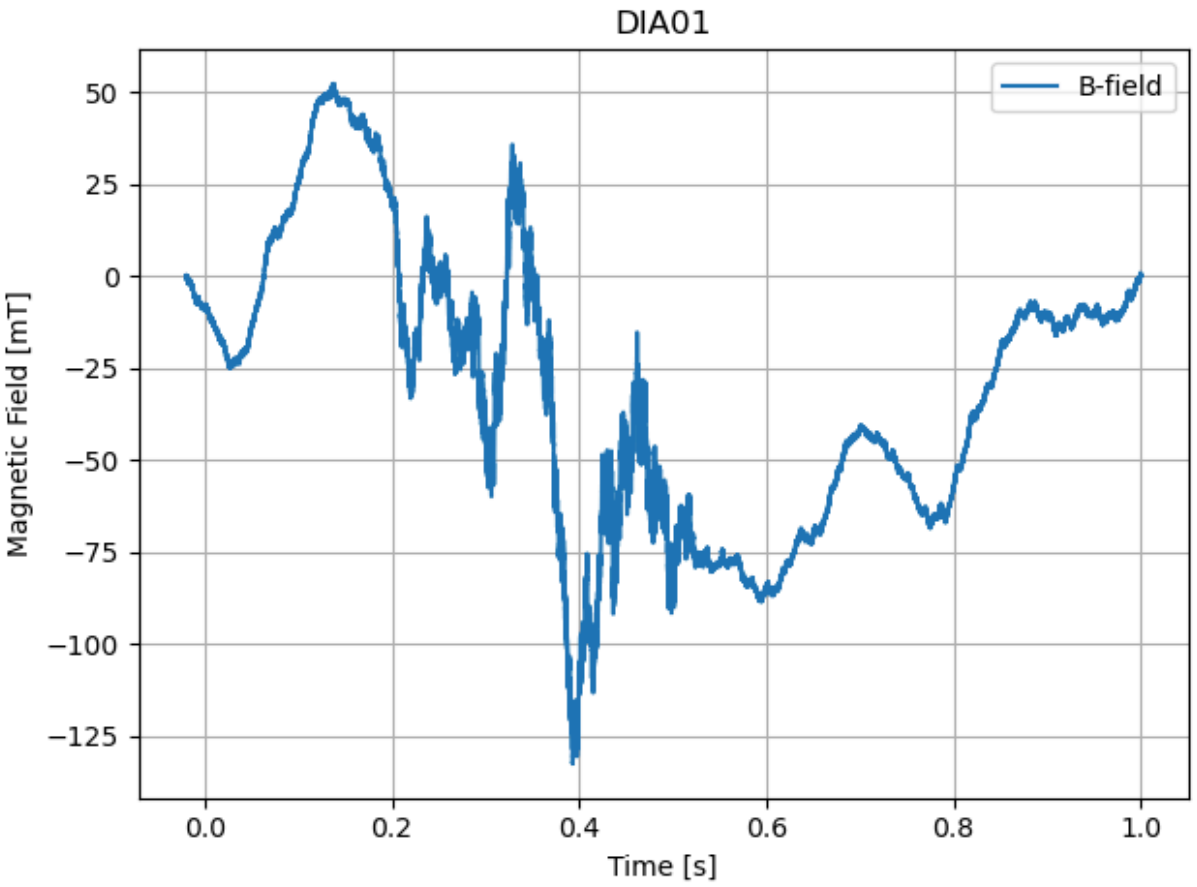
The code is used in a jupyter notebook and in python scripts.

10. Code example II

In [1]:
```
# too run this make sure you have the CMFX_02667.csv

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import astropy.units as u
from plasmapy.formulary import magnetic_pressure
df = pd.read_csv('CMFX_02667.csv', usecols=['Discharge Time', 'DIA01 (V)'])
df = df.dropna()
time = df["Discharge Time"].values * u.s
voltage = df["DIA01 (V)"].values * u.V

# Subtract baseline
voltage -= np.mean(voltage)

# Integration to get magnetic flux
dt = (time[1] - time[0]).to(u.s)
flux = np.cumsum(voltage) * dt

# Define loop parameters
area = (0.42 * u.cm**2).to(u.m**2)
num_turns = 10

# Magnetic field from flux
B_field = flux / (area * num_turns)
p_B = magnetic_pressure(B_field)


print(f" magnetic pressure:{p_B}")
# Plot magnetic field
fig, ax = plt.subplots()
ax.plot(time.value, B_field.to(u.millitesla).value, label='B-field')
```

```
ax.set_xlabel("Time [s]")
ax.set_ylabel("Magnetic Field [mT]")
ax.set_title("DIA01")
ax.grid(True)
ax.legend()
fig.tight_layout()
```

```
magnetic pressure:[8.05783895e-04 3.14261283e-06 1.86400789e-03 ... 2.09941909e-02
2.18272485e-03 2.62693913e-23] Pa
```

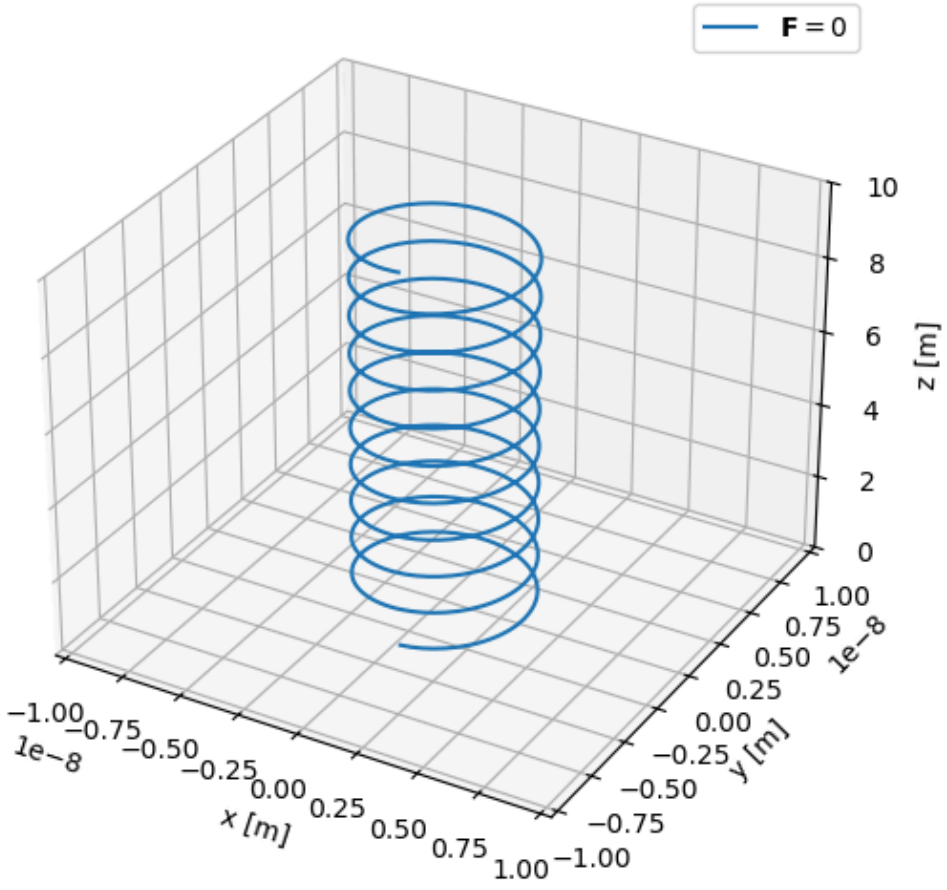


This code is using the Diamagnetic Loop data from DMLs I had installed on CMFX, using PlasmaPy I was able to calculate the Magnetic Pressure, which is a very important parameter, that before I had to actually define an function, and actually do work by hand to calculate. This shows how PlasmaPy's functions can easily be integrated into previous work.

11. Figure generation

PlasmaPy does not prodcue figures on its own, so it mainly uses matplotlib for image generation.

12. Figure example



Here is an example of figure generation within PlasmaPy, this shows a particle driftin in a uniform magnetic field. This type of plot is useful in fusion experiments where a uniform magnetic field confines a plasma, and the trajectories of ions and neutrals within the plasma want to be plotted and studied.

13. Code purity

PlasmaPy is purely python, and depends on python packages, and was developed for python.

14. Input parameters

PlasmaPy can handle many input parameters, but mainly deals with specific plasma related parameters such as particles, plasma properties and real diagnostic data from plasma experiments.

15. Output/yield

PlamsaPy, can output both parameters and datasets, depending on the argument that is inputed to a specific function within PlasmaPy. Basically the same as the inputs.

16. Unit tests

Yes PlasmaPy has unit tests available through pytest, every contribution to PlasmaPy with functionality must have a corresponding test. They even have a guide on there website, at https://docs.plasmapy.org/en/latest/contributing/testing_guide.html

17. Reliability

I am confident that PlasmaPy gives reliable results as the package is used and trusted by plasma physicists around the world, and many of its outputs can be confirmed by doing the physics by hand.

18. Main package dependence

PlasmaPy heavily relies on Numpy. A full list of depedencies can be found in their github source code here is a list of some of the most important. "ipykernel>=5.5.6", "ipywidgets>=6", "lmfit>=1.1", "matplotlib>=3.7", "mpmath>=1.3", "numpy>=1.24", "packaging>=22", "pandas>=2", "requests>=2.28", "scipy>=1.10", "setuptools>=66", "tqdm>=4.65",

19. Documentation

PlasmaPy provides a plethora of documentation on its main website https://docs.plasmapy.org/en/stable/index.html This was more than sufficient for me.

20. Citation requirements

If PlasmaPy code is used in a paper they do show how to reference PlasmaPy on their website.

This research made use of PlasmaPy version 2024.10.0, a community-developed open source Python package for plasma research and education (PlasmaPy Community et al. 2024).

21. References

Matplotlib: https://matplotlib.org/stable/api/index.html

Numpy: https://numpy.org/ Pandas:https://pandas.pydata.org/ Astropy: https://www.astropy.org/ Math: https://docs.python.org/3/library/math.html

22. Package references

I was able to find 2 scientific paper that reference and use PlasmaPy, the first one is written by the creators of PlasmaPy themselves; Published July 20 2024 by Zenodo, the Paper shows the many uses of PlasmaPy. The second was "Machine learning analysis of high-repetition-rate two-dimensional Thomson scattering spectra from laser-produced plasmas". Where they compare the speed of the machine learning algorthim they created to PlasmaPy modules for analyzing Thomson scattering data.

23. New python method requirements

For me the PHYS265 class was more than enough to be able to use PlasmaPy, though I would need to learn more physics in general, specifically plasmaphysics to be able to get the most out of PlasmaPy and use it to its full potential.

24. Final Disclamer

I had never used PlasmaPy in the past, this was my first time using it, I have had some experience with coding relating to plasma through MatLab, but not with python or any python packages. The data CMFX_02667.csv that I used for my coding example I was familiar with however, but I had never tried to determine the magnetic pressure before as I did in my code.