BeaMagnetizer Guide

Julie Hinge

August 2023

1 Turning the programme into an exe file

On windows, download the complete reposetory. Then in your command prompt go to the directory of the reposetory. Here, you need to first install python from their website. Then you need to install pyinstaller by running the command "pip install pyinstaller". Then, in the command prompt, run "pip install requirements.txt" (You might need to first install brew to run all these pip commands, there are lots of guides on how to do it online". To finallly convert the programme into an exe file, run the command:
`pyinstaller --onefile --windowed --add-data="gui_figure1.png;." main.py`
A new folder called dist should now pop up. You can then go into that folder and the executable should be there.
To convert the gui to an application that can be used on mac, py2app can be used. https://py2app.readthedocs.io/en/latest/.

2 Be careful of

The optimizer is not working ideally right now. I'm also worried about the consequence of what happens when we add more sections in regards to the formatting of the lists and how the bounds are applied to each parameter. Maybe use a debugger to ensure this is correct.

3 functions explanations

The graphical user interface is designed accordingly with the Model-view-controller (MVC) framework. This design is used in order to easily pass data between different windows using a controller. All user defined data in this programme is passed into the model and are saved in the Pages module, so it is easily changeable and can be accessed at any point.

3.1 Trajectory (functions.get_trajectory.py)

To calculate the trajectory of the particles i use the following functions:

- rot_matrix (functions.get_trajectory). This is simply the rotation matrix that i use
- next_point (functions.get_trajectory). I use this function within the get_trajectory function to calculate the next point along the trajectory of each individual particle. Within this function i use the get_b function to calculate the magnetic field.
- get_trajectory (functions.get_trajectory). This function is the main one within this script. Here i calculate and return the x and y coordinates along each step of the trajectory along with the directional vector of each particle along the trajectory.

3.2 Map magnetic field (functions.map_mag_field.py)

- get_circle_coordinates. This function is simply used to calculate all the positions along the reference radius.
- create_plot. This function sets up the plot itself
- split_vectors. This function splits li into B and G
- calculate_alpha_intervals. Calculates the start and end point of each section and puts them into a list of lists.
- map_magnetic_field. This function "draws" the magnetic field by calculating the magnetic field at every X,Y coordinate
- plot_trajectories. This function takes the alpha list, the B list, the G list, the initial directions, positions and energies of all particles. It then returns the bending radius along the trajectory of all particles.
- display_magnetic_field. This function can either take custom axis or the default ones. The function uses all the functions defined above to draw the field map, the reference radius and the trajectories of the particles. It also sorts the beams by sum of energies.
- trajectory. This function calculates the exit directions of every particle and puts them into a dictionary called exit_directions. It also find the index at every trajectory by looping reversely through the list of particle directions along each particle trajectory, and noting down the index and the direction that occurs when the particle changes directions for the first time, i.e when they exit the magnetic field. The function also returns each particle position xx,yy along the trajectory.

3.3 Optimization (functions.optimization.py)

- beam_diff. This function takes in all the directions of the particles along the trajectory. This is contained in a dictionary where the keys are the beams and the values are a list of lists of all the directions. The function also takes in a list of indicies (this should actually just be a single number, but every item in the list is just that single number so it shouldn't matter). The index is the index that represents the position of which the beam exits the magnetic field. The function then computes the average directional vector of the first and the last beam, and returns the angle in degrees between these two beams.

- beam_disparity. Once again this function takes all the directions of the particles along with the index at which they exit the magnetic field. I then calculates the angle from the outermost and the innermost particle within each beam.

- exit_size. This function calculates the size in meters average particle in each beam to all other particles within that beam. It then returns the largest particle in meters.

- Objective. This function first calls the get_trajectory function to get the x,y, indicies and directions of all particles (refer to the last point in the section above). It then calls the three functions above using these parameters. These paramenters are then used in the merit function as initial parameters, and the user defined goals are used as well to compare. The function then returns the merit function.

- fmin. Finally we have the minimizer function. This function gathers the initial A, B and G that the user defined and makes guesses based on these. It then also defines some bounds to use inside of the objective function. Finally the solution is calculated using scipy's library (maybe it's worth trying other methods within scipy?) and based on this solution we can calculate the beam disparity, exit size and angle between innermost and outermost beam. Finally we return the optimized B, G, A, average beam size, average beam disparity, and beam differences in angle.

4 Modules

The modules used to make this GUI can be broken down into the individual windows that the users see when they click through the GUI. These modules will be explained in this section.

4.1 Pages

This module stores all the user defined values. It is basically the tool used to pass information between windows.

4.2 StartPage

The first page that the user see when they open the GUI is the start page. Here a visualisation of the magnetic field is showed to give the user a brief introduction on what the programme will be. In the start page, there is also a button that takes the user to page one.

4.3 PageOne

Page one is the page where the user inputs the radius of the reference radius that will later be displayed in all plots to give a reference of how the beams of particles are travelling. The programme then assures that the user inputs a real number, and pops up a warning message if it is not a real number. The user will then also be asked if they want to input the parameters of the magnetic field manually or by uploading CSV files. If they choose manually they will be taken to PageTwo, while if they choose CSV upload, they will be taken to PageThree. Once again, if they choose no options or both options a warning message will pop up before and they will not be allowed to click the Ok button.

4.4 PageTwo

This is the window of the manual uploading of parameters. When the window is opened, two entries are displayed: One where the user can input the magnetic field, B and the gradient G, and one entry where the user can input the angle of the first section. The user can then add sections within the magnetic field using the add button, remove sections using the remove button, and remove all except the first section using the clear all button. In each section, there is already inputted random numbers between 0-10 in grey to give the users and example of what they can input. Once the user clicks the entries, these numbers are cleared, and the color of the text in the entry is changed to white if the user has a dark theme, and black if the user has a light theme. It is only the pre-filled (random) numbers that are cleared when the user clicks them, and the numbers that the user input themselves are not cleared to avoid confusion. Once again, a warning text is displayed if the numbers are not real, and when the total number of degrees for all sections exceed 360 degrees. Finally, an Ok button is included which takes users to Options, and a back button is included to take users to PageOne.

4.5 PageThree

PageThree opens if the user chooses to upload their magnet parameters using csv/txt upload. Here an upload button is included which opens the users' finder. The user can then upload a file where the first column consist of the alpha of the sections and the rest of the columns consist of the magnetic field specifications. Once the file is uploaded, the path of the file is displayed to show the user which file they uploaded. Once again, a warning message appears if the user

did upload either a csv or a txt file. When the user clicks Ok, they are taken to PageFour.

4.6 PageFour

PageFour resembles PageTwo in the set up. The biggest difference is that the entries are already filled out according to the file that the user uploaded. The user can still add, remove and edit sections as in PageTwo. On this page, the back button takes the user to PageThree and the Ok button takes the user to Options.

4.7 Options

This is the page that stores the menu. To view the field map, the user is taken to PageSix. To view the zoomed fieldmap the user is taken to PageFive. To view the tracking of the particles, the user is taken to PageEleven. To go to the optimization, the user is also taken to PageEleven. To go back, the user is taken to PageFour if the csv file upload has previously been chosen, and otherwise to PageTwo if the manual upload has been chosen.

4.8 PageSix

PageSix simply view tha magnetic field map.

4.9 PageFive

PageFive allows the user to choose which ranges they want for the X axis and the Y axis to be in the zoomed field map. When ok is chosen, the bounds are sent to the Pages module and the user is taken to PageSeven

4.10 PageSeven

PageSeven displays the zoomed magnetic field map.

4.11 PageEleven

PageEleven is the page where the user uploads their csv's of their beams of particles. There is also an option to delete these files. When ok is clicked, the file data is being sent to a dictionary, file_data, that is stored in the Pages module. If the user has chosen the optimization button they are taken to PageFifteen. If they have chosen the tracking they are taken to PageFourteen. This choice is also stored in the Pages module.

4.12 PageFourteen

PageFourteen contains the vizualisation of the beam trajectories. This visualisation is made by calling the function display_magnetic_fild from functions.map_mag_field. The seperation angle, average beam size and average beam disparities are also shown in this section, which are calculated by calling first calling the trajectory function from functions.map_mag_field, to get the exit directions and positions. They are then calculated using the functions beam_diff, beam_disparity, exit_size, all imported from functions.optimization2.

4.13 PageFifteen

This is the function that is used when the user has chosen the optimisation button, and have already imported their files. Here they can specify the ideal seperation angle in degrees, average beam divergence in radians and average beam size in radians, which will the saved in the Pages module. When they click ok, the Optimisation is run.

4.14 PageSixteen

This is the page that displays the trajectory of each beam once they have been optimized. At the button of the page, the seperation angle, beam divergence and beam size reached by the optimiser are shown, along with the section sizes, magnetic field strengths and gradients. The trajectories and optimisation is done using the fmin function from Pages.optimisation2, and the seperation angle, beam divergence and beam size are calculated using the functions beam_diff, beam_disparity, exit_size, all imported from functions.optimization2.

4.15 PageEight

Here the user can specify what bounds they want in the plot. The optimisation will then be run like in PageSixteen, except this time the bounds of the x and y axis will be different.