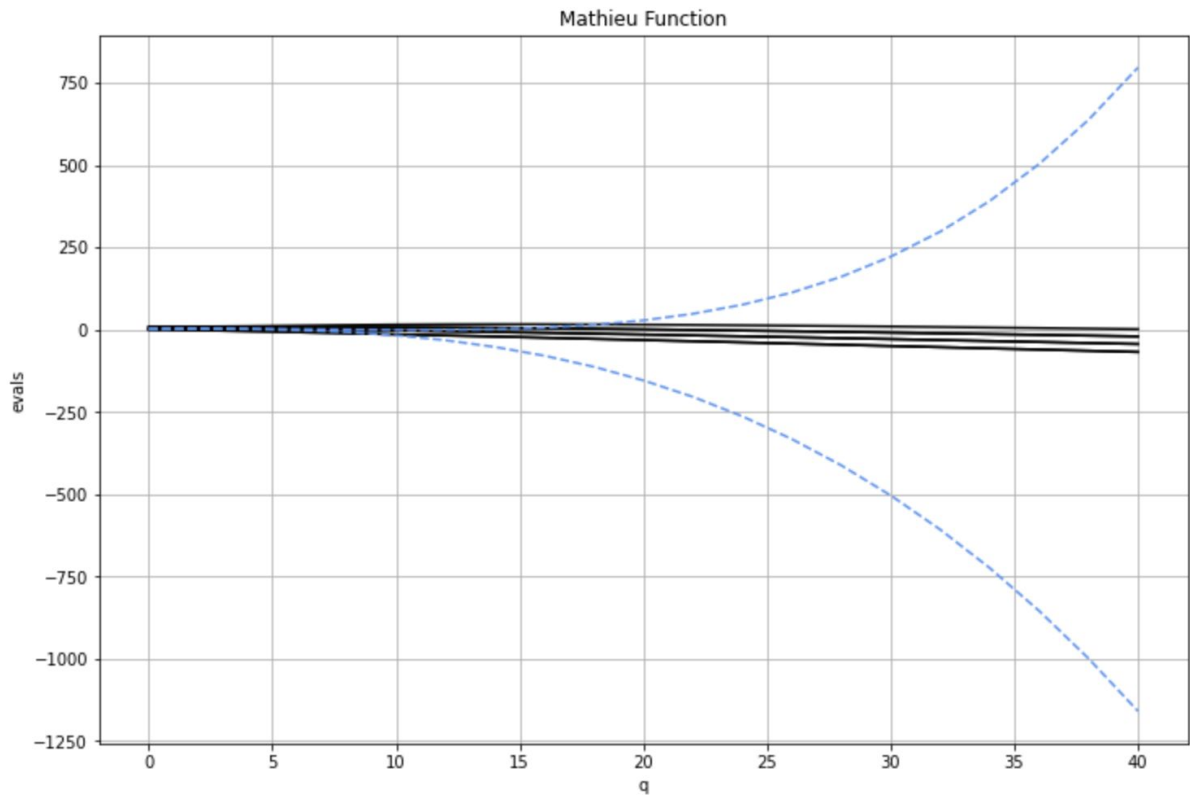


Homework 5



The main idea of this project was to show how to run fortran code with inputs from multiple sources by automating it through python. The python script was structured to change the directory from the current one to one where the fortran code resided and then compile and run the various pieces of code. OS command functions were used to switch into mathieu, run the make file and to return to the parent directory upon exiting the program. Within the main itself, N is first defined to be whatever value the user wants to test. The main is used to define the starting point of the python script and follows the order defined. The rebuild function is called to clean mathieu/ and change dir to the parent folder again. `parsweep_mathieu(N)` which is defined in the .py run code will be called. The range of q is defined and then for each index of q, `run_mathieu(N,q)` is called thus triggering the build function to make the fortran executable and

return to the parent directory again. Next, a `generate_input(N,q)` was fed the values of `N` and `q` through a formatted string (`f'string'`) to write the values into three lines which would then be stored in an `.init` file and a backup of the previous values were saved in a separate file. To achieve this result, `fp` was defined as the open function of which the `.init` file was passed through in write mode (`, 'w'`). After saving the values, return to the parent directory. The executable `mathieu` was run and the function `plot_parsweep(N, nPlot)` was called to store the data generated by `mathieu` into separate Mathieu data files for each `q` resulting in a total of 20 files. These data files are then called upon to be input into the figure as the y-axis for different `q`. The two equations were added and formatted appropriately.

To make adjusting the hard set values of `qRange` easier, there are a few approaches. One way is add a key range parameter to both functions (e.x. Pass "`N, q, qRange`" through function) and place the variable `qRange` to be defined under the main instead. Now `qRange` only has to be changed in one spot where the other parameter "`N`" is also defined.

An interactive python shell such as JupyterNotebook can be used to monitor the generated graphs and other work as you go along. This can be accessed by entering "jupyter notebook" into the command line for the directory the `.py` file is in. From there the code can be worked on more easily. A new notebook can be created and the code can be fed in for each individual function of `mathieuRun.py`.