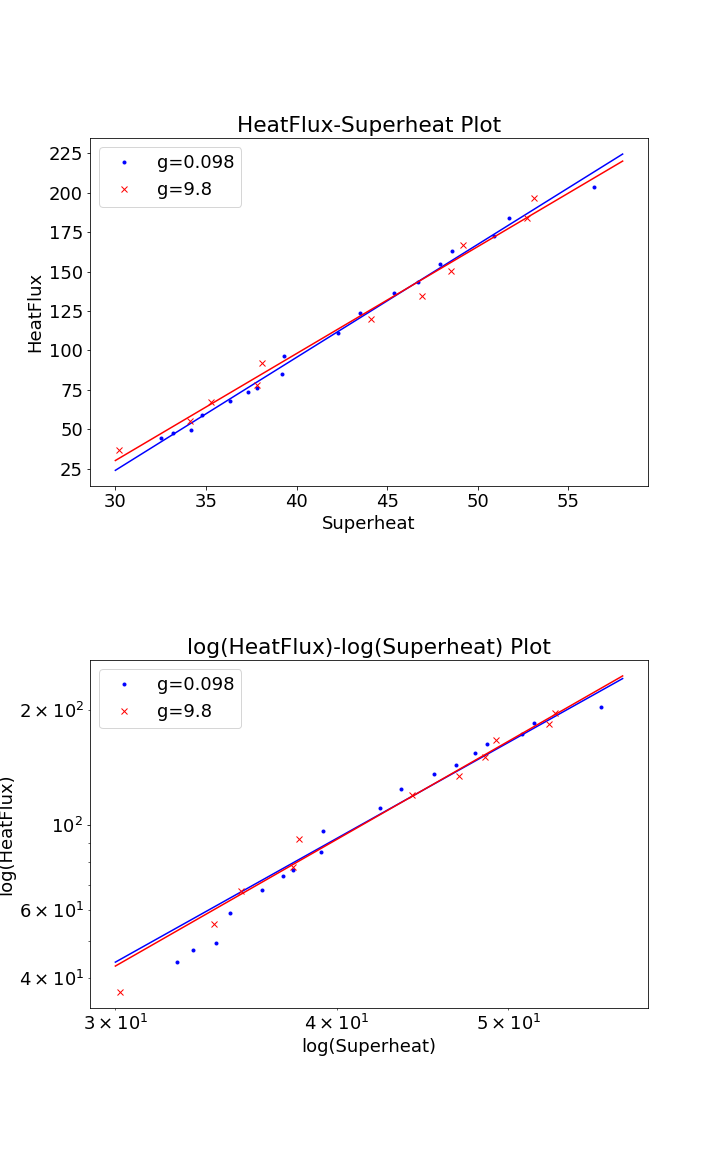
**Report for Project 1**

**Task 1**



The figure above includes a HeatFlux-Superheat plot and a HeatFlux-Superheat log-log plot. Blue dots and lines represent the data and regression under the condition of micro gravity (0.098), while the red ones represent the condition of regular gravity (9.8).

As shown in the figure, heat flux is positively correlated with superheat, and the relationship varied a little in different gravity conditions.

It seems that there could be a linear relationship between heat flux and superheat, so linear regression is applied to the data. (As for the log-log plot, linear relationship should be transferred into exponential relationship according to simple math calculations.) The result is shown in the table below. Mean Square Error (MSE) is used to decide whether the regression is good or not.

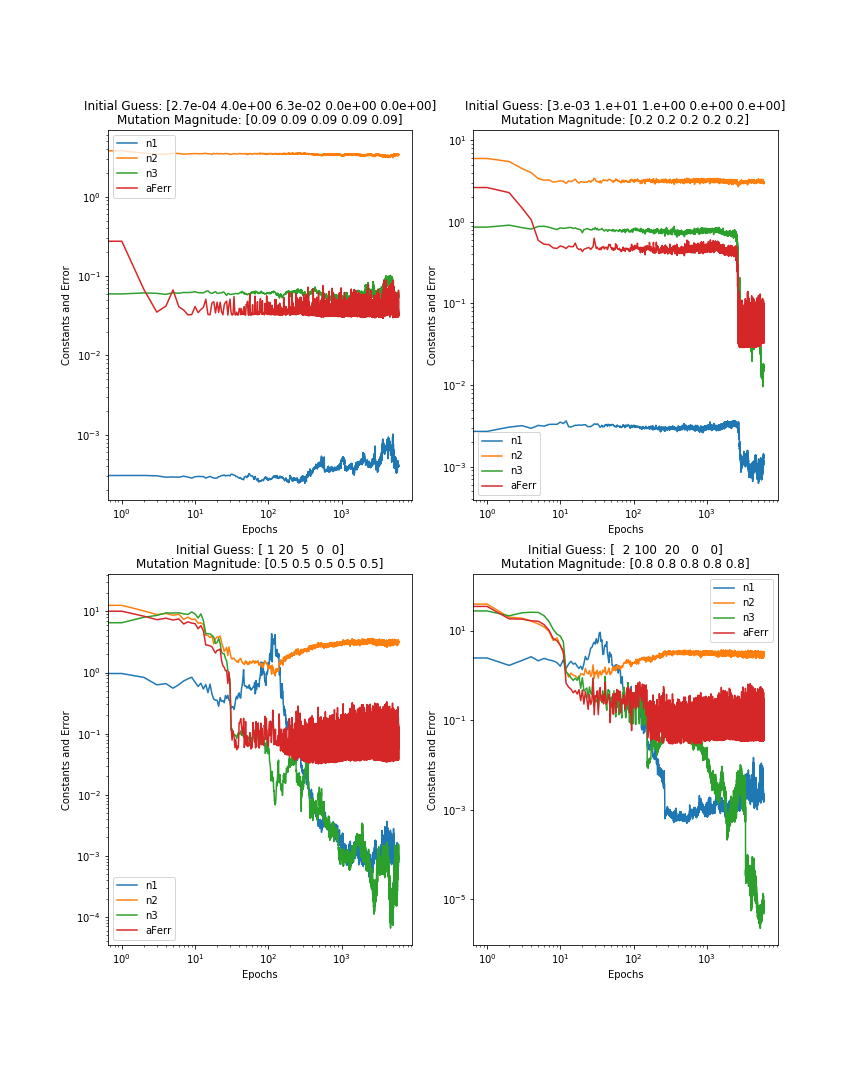
|  |  |  |
| --- | --- | --- |
| g | Lin. Regression | Exp. Regression |
| 0.098 | y = 7.167503 \* x + -191.125966  MSE = 15.158677 | y = 0.006760 \* x ^(2.581364)  MSE = 72.352377 |
| 9.8 | y = 6.785368 \* x + -173.429351  MSE = 38.131234 | y = 0.005384 \* x ^(2.641377)  MSE = 32.514733 |

Therefore, linear regression has a better performance when gravity is low, while exponential regression works better under a regular-gravity condition. However, the conclusion isn’t accurate enough because of inadequacy of data.

Code Modified:

* Save the data as a csv file, so that the reading process will look concise.
* The data type is changed to pandas.DataFrame, also to make the code concise.
* Linear and exponential regression added.
* Plot the scatter data and the regression graph and save it.

**Task 2**



The figure above indicates how constants and errors change (within totally 6000 generations/epochs) under different initial guesses and mutation magnitude. Mutation magnitude are adjusted to make the genetic algorithm come to ideal results. With a more dramatic initial guess (farther from the optimal parameters), the mutation magnitude has to be adjusted larger for a broader range of trial. Parameters and some of the results are listed below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Experiment | Parameters | | | |
| n1 | n2 | n3 | Mutation  Magnitude |
| 1 (Original) | 2.7e-4 | 4 | 6.3e-2 | 9% |
| 1 (Optimal) | 7.640e-4 | 3.194 | 6.064e-2 |
| 2 (Original) | 3.0e-3 | 10 | 1 | 20% |
| 2 (Optimal) | 1.290e-3 | 3.055 | 5.000e-2 |
| 3 (Original) | 1 | 20 | 5 | 50% |
| 3 (Optimal) | 3.682e-3 | 2.755 | 1.098e-2 |
| 4 (Original) | 2 | 100 | 20 | 80% |
| 4 (Optimal) | 9.000e-4 | 3.150 | 4.700e-2 |

|  |  |  |
| --- | --- | --- |
| Experiment | Results | |
| Relative Absolute Error | Best Epoch |
| 1 | 2.958% | 3679 |
| 2 | 2.872% | 3233 |
| 3 | 3.186% | 485 |
| 4 | 2.935% | 682 |

Some more details can be found out from the tables. Though different solutions are reached when initial guesses vary, all of the solutions have the same order of magnitudes, which means these solutions are closed to the exact optimal solution. All of the relative absolute errors are near 3%, and it also proves genetic algorithm works well in this case.

As this problem can also be considered as a linear regression problem, therefore we could create a baseline with linear regression algorithm (See Task2\_LR.py). Listed below is the result of linear regression.

|  |  |  |  |
| --- | --- | --- | --- |
| Parameters | | | Results |
| n1 | n2 | n3 | Relative Absolute Error |
| 3.933e-3 | 2.746 | 4.943e-2 | 2.995% |

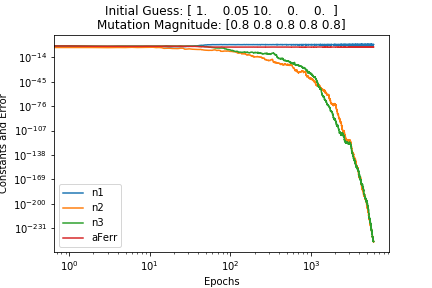
It is amazing that the solutions from Experiment 1, 2, 4 perform better than linear regression when the loss function is Relative Absolute Error (no doubt linear regression has the minimum mean square loss). All of the results above prove genetic algorithm to be potential when the model is non-linear or the loss function is partly non-differentiable.

The result of task 2 also aligns with that of task 1. One of the conclusions from task 1 is that the relationship between heat flow and superheat varied a little under different conditions of gravity. According to task 2, there are two orders of magnitude difference between n2 and n3, correspondingly represents the weight of superheat and gravity. It is a quantitative proof for the conclusion of task 1.

However, adjusting the mutation magnitude doesn’t always promise good solutions. Furthermore, if initial values are extremely badly chosen, the algorithm will hardly approach the optimal solution, no matter how mutation magnitude is chosen. Here’s an example.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Experiment | Parameters | | | |
| n1 | n2 | n3 | Mutation  Magnitude |
| 5 (Original) | 1 | 0.05 | 10 | 80% |
| 5 (Optimal) | 96.70 | 1.7e-5 | 2.072e-2 |

|  |  |  |
| --- | --- | --- |
| Experiment | Results | |
| Relative Absolute Error | Best Epoch |
| 5 | 9.576% | 61 |



As shown in the graph, n2 and n3 rapidly slopes to zero. However, it’s not the single case. Similar trends could also be observed in experiment 2, 3, 4 (the n3/green line). The problem may exist in how mutation is generated. Let m be the magnitude of mutation, the mutation