**Assignment 4 Report-Aaron Berryman**

Link to repository: <https://github.com/aberrypicker/PH510>.

New files created are ‘poisson\_green\_class.py’ and ‘assignment\_4\_code.py’. The former contains a new class created to contain the functions required for the Poisson grid, the random walks, and the Green’s function calculations. The other file imports the class and uses its’ functions to deliver the desired outputs for the tasks. Despite best efforts the Monte Carlo class from assignment 3 did have to be modified to accommodate this assignment, with changes focusing on keeping outputs as arrays, needed for this.

Worked with: Eamonn M., Ben W., Finn D., Nikki W, Natalia K, Jack M. A large amount of discussion over how to approach this task and then helping each other on how to update the classes to work with the Monte Carlo and parallelise, and work on grid.

Pylint score should be 10 across new code, suitably licensed and version controlled.

Task 5 Table for deterministic vs stochastic results:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Boundary Conditions | Position(cm) | Green’s Function (V) | | | | Overrelaxation method (V) |
| 0 Charge | Uniform 10C Charge | Uniform Gradient | Exp. Decay |
| All edges +1V | (5.0, 5.0) | 0.9643 | 0.9928 | 0.9657 | 0.9643 | 1.0000 |
| (2.5, 2.5) | 0.9643 | 0.9820 | 0.9654 | 0.9643 | 1.0000 |
| (0.1, 2.5) | 0.9643 | 0.9695 | 0.9647 | 0.9643 | 1.0000 |
| (0.1, 0.1) | 0.9643 | 0.9662 | 0.9644 | 0.9643 | 1.0000 |
|  | | | | | | |
| Top & Bottom +1V, Left & Right -1V | (5.0, 5.0) | 0.0001 | 0.0260 | 0.0068 | -0.0017 | 0.0000 |
| (2.5, 2.5) | 0.0014 | 0.0156 | -0.0004 | 0.0031 | 0.0000 |
| (0.1, 2.5) | 0.7126 | 0.7190 | 0.7180 | 0.7143 | 0.7415 |
| (0.1, 0.1) | 0.0012 | 0.0039 | 0.0010 | -0.0008 | 0.0000 |
|  | | | | | | |
| Top & Left +2V, Bottom 0V, Right -4V | (5.0, 5.0) | 0.0001 | 0.0346 | 0.0102 | 0.0045 | 0.0000 |
| (2.5, 2.5) | 0.0014 | 0.7277 | 0.7054 | 0.6963 | 0.7012 |
| (0.1, 2.5) | 0.7126 | 0.1926 | 0.1844 | 0.1853 | 0.2014 |
| (0.1, 0.1) | 0.0012 | 0.9551 | 0.9530 | 0.9533 | 0.9710 |

Results are from n=21 x n=21, ran 100000 times. Perhaps due to issues with code but rather inefficient leading to large runtimes for anything beyond 21 x 21, chosen for ability to place 3a and 3b in direct diagonal point, but size of grid means c and d are not in correct spot for 0.1cm, as this would require 100 x 100 grid which is just beyond limitations. Best is done regardless of this. Numbers agree with overrelaxation to degree which is understandably slightly out, due to relatively low number of Monte Carlo runs, but functionality is there. Plots for both desired cases on following page.

Task 3a

A graph of a black square with a black square with a black square with a black square with a black square with a black square with a black square with a yellow and purple square with white lines

AI-generated content may be incorrect.A chart of a square with a yellow and purple center

AI-generated content may be incorrect.

Task 3b

A graph of a black square with a black square with a black square with a black square with a black square with a black square with a black square with a yellow and purple square with a black

AI-generated content may be incorrect.A graph of a charge potential

AI-generated content may be incorrect.

Task 3c

A graph of a black square with a black square with a black square with a black square with a black square with a black square with a yellow and purple square with white numbers and numbers

AI-generated content may be incorrect.A graph of a charge potential

AI-generated content may be incorrect.

Task 3d

A graph of a function

AI-generated content may be incorrect.

A graph of a charge potential

AI-generated content may be incorrect.