# CSC 581 Homework 3
# Particle Simulation Part 1
## Serial Optimization
**Tyler Hetland, James Speers**

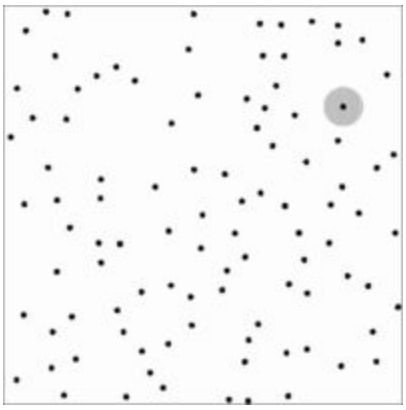
# 1. Introduction

### 1.1 PARTICLE SIMULATION

In the particle simulation being analyzed for this assignment, the particles interact by repelling one another when they come within a specified cutoff distance of each other. The particles in the simulation are bound within a square. If a particle hits one of the walls of the square it will bounce off and continue moving at the same speed at which it hit the wall. The particle density is set low enough so that only N particle interactions would be expected per step in the simulation (Where N is the number of particles).
The short range repulsive force between each particle is defined as:

$$\frac{\left(\frac{1-c}{\sqrt{r}}\right)}{r}MV$$

Where ***c*** is the force cutoff distance, ***r*** is the distance between the two particles, ***M*** is the particle mass and ***V*** is the velocity of the particle.
Below is a visual example of the simulation with the force cutoff distance for an individual particle highlighted in grey:



### 1.2 CORRECTNESS

The minimum distance between 2 particles is tracked throughout the simulation. Correct simulations will have particles stay at greater than 0.4 * c from one another. Values below

0.4 * c indicates that at least one particle in the simulation has not interacted with any other particles.
The average distance between 2 particles is also tracked throughout the simulation. Correct simulations will have an average particle distance of greater than 0.8 * c. Values below 0.8 * c indicates that most particles in the simulation are not interacting.

### 1.3 PROBLEM STATEMENT

The objective of the assignment was to optimize the provided naive serial implementation so that the operational complexity is reduced from $O(n^2)$ to roughly O(n). Given the constants in the provided code, the optimization should be better than or equal to O(n) for simulations with less than 10000 particles.

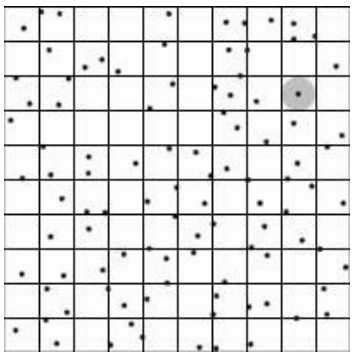
# 2. Serial

### 2.1 NAIVE SERIAL APPROACH

The first approach to optimizing this simulation was a serial approach. This approach would perform all calculations and operations sequentially and had every particle interact with every other particle. As mentioned above 2 particles will only interact if they are within a given cutoff distance due to the nature of the short-range repulsive force implemented. Given our number of particles and particle density that means a significant number of these interactions yields no result. The naive serial approach for a simulation with n particles is an $O(n^2)$ problem.

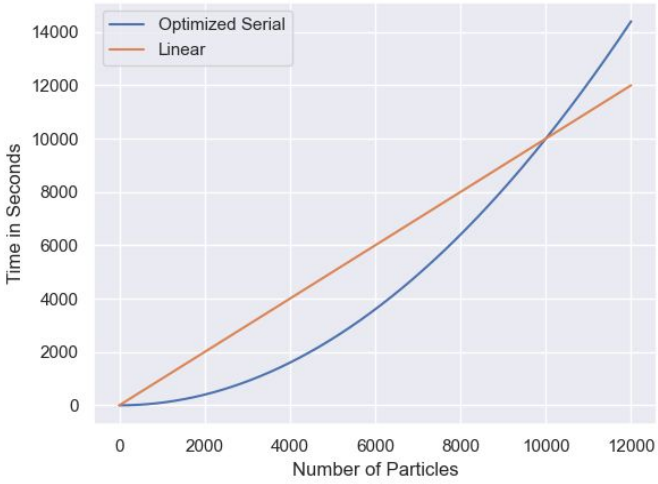
### 2.2 SERIAL OPTIMIZATION THROUGH BLOCKING

While our iterations were still quadratic in nature, the goal was to provide such a sufficiently small coefficient that we would satisfy our upper bound, O(n).
To achieve this we focused on eliminating interactions that yielded no results, i.e. interacting two particles that were outside of the cutoff distance. We did this by logically dividing our simulation space into blocks and then iterating over each block and its neighbors.



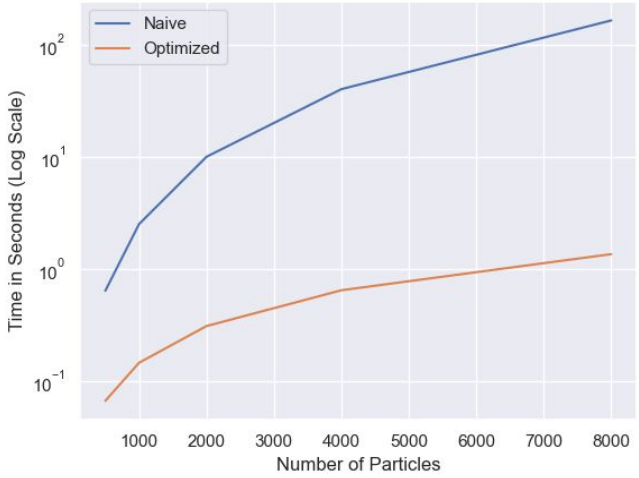
The size of each of these blocks was cutoff * cutoff, or .01 * .01. This effectively means that any iteration we are processing is on average .0001 * ($n^2$). This approach yielded a

significantly lower number of interactions, thus vastly improving the performance. The actual slope achieved was 1.084970, compared to the original slope of 2.003053.




The run time improvements can be seen below. As the number of particles in the simulation increased so did the factor by which the runtime was improved.

| Particles | Optimized (s) | Naive (s) |
|---|---|---|
| 500 | 0.066334 | 0.637175 |
| 1000 | 0.145085 | 2.50661 |
| 2000 | 0.308434 | 9.99832 |
| 4000 | 0.643867 | 40.1501 |
| 8000 | 1.35267 | 164.761 |



### 2.3 CHALLENGES

One of the main challenges of this approach was the need to manage which blocks a given particle belonged to and reassigning those values. Individual particles had their block coordinates updated appropriately, but due to the structure used there needed to an additional step to reassign all particles to their corresponding blocks. One such improvement for this would be to add a pointer to the particle for its respective block which can be easily updated after moving a particle.

# 3. Conclusion

The results of this optimization were successful as long as we stayed within the tuned density and size. While our theoretical best case was to be less than linear, our actual result was just slightly above linear. This is due to the computational overhead added at the end of each step by updating the block identifier for particles that changed blocks during the move() function call. This was still an improvement over the original performance which was quadratic.

# 4. References

Men, Xiaoqian. “Particle Simulation Using Serial, GPU and Distributed Approaches.” *Particle Simulation Using Serial, GPU and Distributed Approaches*, 2015, www.engr.mun.ca/~dpeters/papers/MenMEng.pdf.