

1. About our implementation:

- Data structures: We simulate this environment as a 2D square grid contains cells
- Algorithms:
 1. Process each cell. For all particles in one cell, first check if they collide with the wall, and resolve collisions if found. Then if they collide with other particles within this cell or with particles in another cell, and resolve collisions if found.
 2. Process all the cells sequentially using the above implementation.
 3. To parallelize the program, we divide the grid into 4 regions (i.e: all the cells in 1 region would be processed parallelly, but 4 regions are processed sequentially). For examples:

1	2	1	2	1	2	1	2
3	4	3	4	3	4	3	4
1	2	1	2	1	2	1	2
3	4	3	4	3	4	3	4
1	2	1	2	1	2	1	2
3	4	3	4	3	4	3	4
1	2	1	2	1	2	1	2
3	4	3	4	3	4	3	4

Explain: cells in the same region are not adjacent to each other, hence we can process them parallelly without race condition. However, cells in different regions are adjacent together, hence it would modify particles in each other when processing cells, hence race conditions need to process regions sequentially.

Also: we parallelize particles' moving, since we just modify particle itself, not involve the others

- OpenMP construct be used:

<`#pragma omp for`>: to parallelly simulate the moving of each particle, and the process of each region

<`#pragma omp atomic`>: avoid data race

2. This is runtime while we run our code on file: **tests/large/100k_density_0.9_fixed.in**
 Average after run `run_bench`

i7-7700 (8 threads)	xs-4114 (20 threads)	sequential (1 thread)
20.666 +/- 0.196 seconds	15.5591 +/- 0.0257 seconds	~163.07 +/- 0.05 (seconds)

3. Two optimizations we attempted:

- Divide grid into 4 regions to parallelly process
- Each pair of cells and each pair of particles are processed only 1 time, to avoid wasting multiple processes.

4 Appendix:

Command to run test

```
make  
perf stat -- ./sim.perf ./tests/large/100k_density_0.9_fixed.in 8
```

Command to run run_bench

```
./run_bench.sh i7-7700 tests/large/100k_density_0.9_fixed.in 8  
./run_bench.sh xs-4114 tests/large/100k_density_0.9_fixed.in 20
```