2D Particle Simulation: Testing Performance Using OpenMP and Measuring MSD of Particles

Georges Karagozian [1]

[1]Faculty of Science , University of Ontario Institute of Technology , Oshawa, ON

November 2022

1 Introduction

A particle simulation was developed in Fortran90 based on Langevin dynamics, which is a mathematical model for molecular dynamics described by equation 1 originating from the theory of Brownian motion [1].

$$m\frac{d^2x}{dt^2} = -6\pi\mu a\frac{dx}{dt} + X \tag{1}$$

The simulation will account for particles interaction with each other via the Lennard-Jones potential. The calculations required to predict particle interactions come with a heavy computational cost. To alleviate this cost, OpenMP library will be used to run the code in parallel by creating multiple threads to spread the tasks amongst them. As part of this, the area simulated will be divided into sectors.

The simulation will be used to run with an increasing number of particles as well as different combination of thread numbers using OpenMP parallel library, and number of sectors. The aim of these simulations will be to observe the trends in the wall time of the simulation with varying parameters, as well as to calculate the mean square displacement of particles and analyze their behaviour.

2 Methods

Velocity Verlet is a form of the Verlet algorithm ([2]) which is used to numerically solve Newton's equations of motion and obtain the trajectories of the particles in a Molecular Dynamics (MD) simulation([3]).

The particles are initialized as a uniformly distributed grid. They experience an external force due to their mutual interactions defined by the Lennard-JOnes potential. The forces on the particles can be represented by equation 2

$$\ddot{x} = -\zeta\dot{x} + F_{ext} + \sqrt{\frac{2k_BT}{\zeta}}\xi(t), \tag{2}$$

where $\zeta = \frac{\gamma}{m}$, k_B is the Boltzmann constant, $\xi(t)$ is a random fluctuation force , and T is the absolute temperature; Then equation 2 is rearranged and discretised to be used in the iterative algorithm in the following order:

$$\ddot{x}(t) = -\zeta\dot{x}(t) + F_{ext} + \sqrt{\frac{2k_BT}{\zeta}}\xi(t), \tag{3}$$

$$\dot{x}(t + \frac{1}{2}dt) = \dot{x}(t) + \frac{1}{2}\ddot{x}(t)dt, \tag{4}$$

$$x(t + dt) = x(t) + \dot{x}(t + \frac{1}{2}dt)dt, \tag{5}$$

$$\ddot{x}(t + dt) = -\zeta\dot{x}(t + \frac{1}{2}dt) + F_{ext} + \sqrt{\frac{2k_BT}{\zeta}}\xi(t), \tag{6}$$

$$\dot{x}(t + dt) = \dot{x}(t + \frac{1}{2}dt) + \frac{1}{2}\ddot{x}(t + dt)dt. \tag{7}$$

The mean square displacement (MSD) of particles, as well as the wall time for running the code, were calculated at the end of the simulation and written to a file.

3 Results and Analysis

MSD data was plotted and is seen in figure 1. The trend shows an initial linear behaviour followed by an exponential growth as expected. This result is reproduced several times with 1000 particles using various combinations of number of sectors and threads. Although, a longer simulation with more particles would be better to gain insight into the behaviour after 10 seconds where the plot seems to be reaching a plateau.

The simulations parameters and walltime are represented on a 3D plot as shown in figure 2. Some trends observed were that increasing the number of threads to a value above the number of physical cores present in the machine (8 here) increased the wall time. And contrary to expectations, increasing the number of sectors increased the wall time as seen by the yellow points using 16x16 grid having higher wall time than simulations with equal number of particles. As well as at 5000 particles, 4x4 grid performed better than an 8x8 grid. This defies the theoretical expectation of walltime scaling, so there must be a bug somewhere in the code causing this to happen.

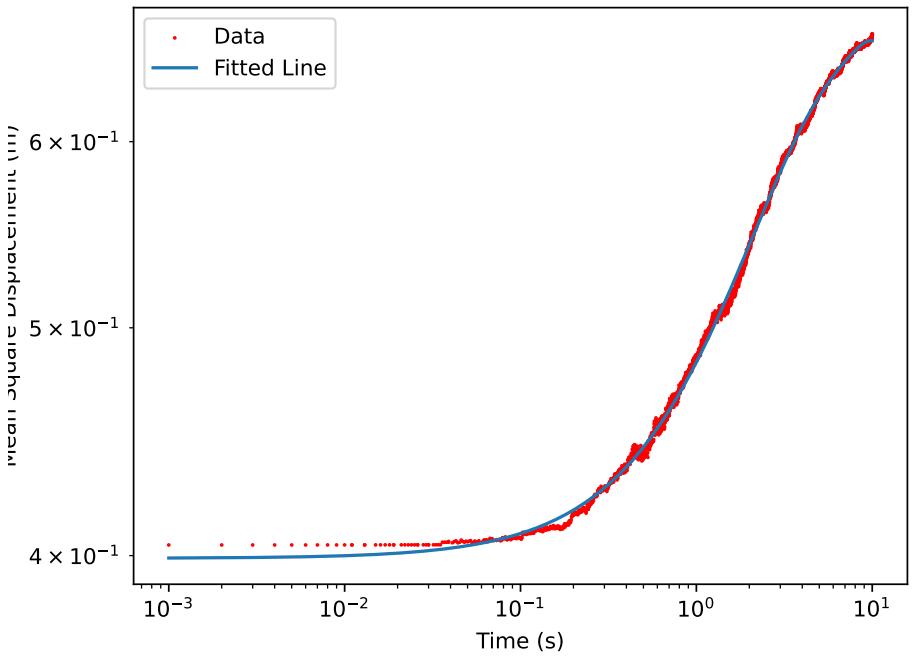

Figure 1: Mean square displacement of 1000 particles using 4x4 sectors and 4 threads and runtime of 10 seconds.

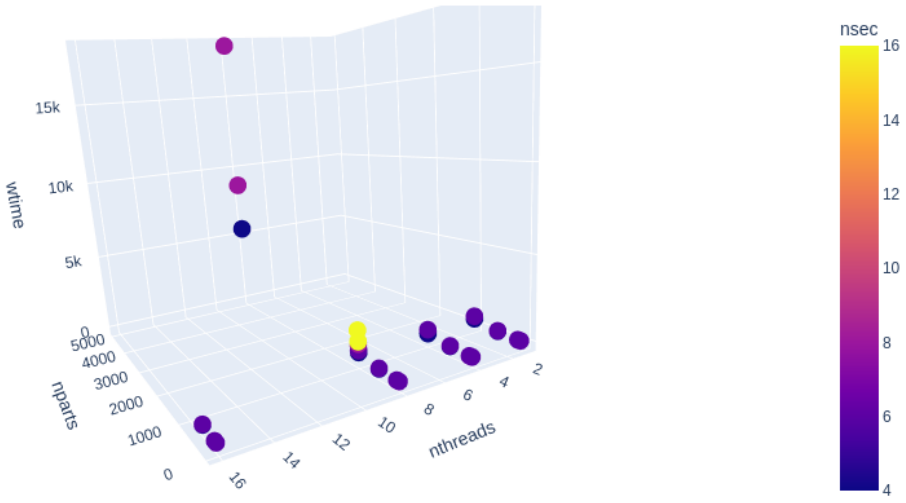

Figure 2: Walltime data as a function of number of particles, number of sectors-per-side, and number of threads.

4 Conclusion

A particle simulation was developed in Fortran90 while accounting for particle interactions with each other. The code was parallelized using OpenMP library and ran using multiple configurations. The MSD of the particles as a function of time was recorded and analyzed As well as the walltime of the program as a function of the various configuration that the code was run in. One of the unusual trends noticed was the increase of walltime in direct proportion to increasing the number of sectors, indicating a bug in the code that needs to be addressed.

5 Acknowledgments

This project was completed in collaboration with classmates (Saadat, Amin, Abe, Darryen) and Prof Lennaert Vanveen.

References

[1] Don S. Lemons and Anthony Gythiel. Paul langevin's 1908 paper "on the theory of brownian motion" ["sur la théorie du mouvement brownien," c. r. acad. sci. (paris) 146, 530–533 (1908)]. *American Journal of Physics*, 65(11):1079–1081, 1997.

[2] Loup Verlet. Computer "experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules. *Phys. Rev.*, 159:98–103, Jul 1967.

[3] Peter H. Berens et al. William C. Swope, Hans C. Andersen. A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters. *J. Chem. Phys.*, 76, 1998.