ME522: High-Performance Scientific Computing - Assignment 2

Dr Gaurav Bhutani

1st May 2024
(Due: 15th May 2024)

Problem: Discrete Element Modelling of dilute systems

We have a few circular particles in a box moving with some velocity, with the assumption that the total area of the particles is small as compared to the area of the 2-D box. In such a case, particle-particle collision can be neglected and only particle-wall collision needs to be modelled and simulated.

Similar to the problem you received in the lab examination in midsem exam, write a Fortran code to simulate the motion of n particles in a 2-D box. Gravity can be neglected. You may initialise all the particles at the centre of the unit box, with velocity magnitudes distributed randomly between 0 and 1. The directions can be distributed uniformly for the n particles. Since we are not worried about particle collisions, particle overlapping is allowed.

As the particles do not interact, this turns out to be an embarrassingly parallel problem. Each particle motion can be handled by a different thread without any need for synchronisation. Use OpenMP to parallalise the problem. Make sure that the code works both with and without the use of the *-fopenmp* flag.

Since the simulation proceeds in time with some Δt value, each particle will have a position and velocity for each timestep value. If Δt is chosen as 0.01 s and we are simulating for a total of 1 s, we will have 100 timesteps. Your code should output a file corresponding to each particle with the file organised as follows: a line will have 5 numbers, with the first corresponding to the timestep number, the next two corresponding to the x and y coordinates and the last two corresponding to the x and y components of the velocity. Line number k will correspond to values for the k-th timestep. Your code should output a file corresponding to each particle with the naming convention as follows: p1.txt, p2.txt, ... all the way to pn.txt.

To visualise these particles, write a python script that takes these n files as input and outputs an animation for the particles. See the attached video for reference.

Allow the user to pass the number of particles as a command line argument, as *./a.out 100*.

You will have to upload a report as a pdf file this time. The report should contain output for different values of n, in the form of particle snapshots. Also, include graphs for strong and weak scaling. Codes — dem.f90 and plot_python.py — will be included in the appendix as text. There is no need to upload code files this time.