Simulation of particles in a box report

Donglin Liu

November 10, 2022

1 Problem Setting

We suppose to construct a program to simulate the moving particles in a confined box. Such a system can be used to simulate e.g. gas in a closed container.

Each particle has a position s, a speed v, and a radius r, then n particles with some initial values of s, v and r will move the distance Δt in each timestep according to the following criterias.

- Decide whether any particle collides with a wall in the box (this means that the distance from the border to the particle is less than the radius). If a collision occurs, the particle should bounce.

- Decide whether two particles collide. If particles i och j collide, use the formula:

  $$v_i = v_i + \frac{(v_j - v_i) \cdot (s_j - s_i)}{|s_j - s_i|^2}(s_j - s_i)$$
  $$v_j = v_j + \frac{(v_i - v_j) \cdot (s_j - s_i)}{|s_j - s_i|^2}(s_j - s_i)$$

  to update their speed. (The operation denotes dot product) Ignore the case where more than two particles collide.

- When all particles have been tested, they move to a new position using the formula:

  $$s = s + \Delta t \cdot v$$

2 Program Structures

I failed to compile the Fortran code to an excutable package, so the project is in general seperated into two parts: Simulation is done by Fortran and visualization is done by Python.

2.1 Fortran Code

The simulation process can be seen in file './particles/main.f90'. We first allocate memory for particle system, then set the initial value for the system. Before update the system, we need check if the particles have collision or hit the wall. These functions are implemented in the following subroutines:

- allocate_particle_system
- init_particle_system
- check_boundary
- check_collision
- update_particle_system
- deallocate_particle_system

The simulation results is saved in file 'particle.state'.

2.2 Python Code

To visualize the simulations, we read the data in 'particle.state' and visualize it by *vedo* package.

3 Run Program

3.1 Parameters setting

In Fortran file './particles/main.f90' you can change the following parameters. The box is 3d container with the lower left corner at (0,0) and the upper right corner at (1,1). We choose $n = 200$ particles with initial positions, velocities and radius are set to uniformly random values in $[0, 1]^3$, $[0.01, 0.09]$ and $[0.005, 0.015]$ respectively. Let the time step $\Delta t = \frac{r_{min}}{3*(v_{min}+v_{max})/2} = \frac{0.005}{3*(0.01+0.09)/2} = \frac{1}{30}$ so that the particles don't pass through the walls.

3.2 Simulation and Visualization

- Step 1: Run Fortran file *'./particles/main.f90'* to simulate the process in Qt Creator.
- Step 2: To change to project directory in Command Prompt, then create conda virtual environment by: *pipenv install*
- Step 3: To visualize the results by the command: *run_python_vedo.sh*