Particle Diffusion

Michael Jones and Ian Jackson

Project Description

For this project, we plan to simulate the motion of a particle within a given gas. The simulated gas will have the variable parameters of volume, temperature, and number of particles within the volume. Additionally, beyond the parameters of the gas, there will be the option to adjust number of particles that are displayed on-screen. We will also draw a path traveled by the target particle over the simulated time. Further options will be given to the user to allow the simulation of Brownian motion vs. particle diffusion in a constant gas (so the mass of the target particle could be changed, and the viscosity of the fluid would be a variable in Brownian calculation). We will start with a 2D simulation, and if it comes along too easily we will add a 3D option.

How Work will be Divided

There will be several parts to the project, which may need to be completed sequentially. Despite knowing this, we will attempt to divide the work into a few reasonable sections.

1-Basic motion/foundational equations: Calculation functions

a. Figuring out the equations that lie at the base of computing a particle's path
b. Brownian motion equation, particle diffusion equation, possibly the elastic collision equation, etc.

2-Path-tracking and Graphing: Plotting functions

a. Creating and maintaining an array that tracks the particle(s)' coordinates over successive calculations
b. Graphing a plot that connects each of these locations in a visually clear path
c. Distinguishing two or more particles with differently-colored paths

3-User Interface

a. Making an interface which allows easy manipulation of variables before starting a calculation
b. Showing the motion of the particle in an animation

4-Streamlining and Additions

a. Throughout the coding process, new insights and ideas will occur. Recording and implementing these will be important.
b. Upon completion of one's own section, or upon reviewing another's, we will suggest or make edits to improve the performance of the functions within the project.

Types of Analysis

Given the number of variables at play (and the high number of calculation iterations) in this project, there will be a lot of numerical calculation. This will simplify the computer's job and improve calculation time.

Much of the calculation that we will be doing will use well-known formulas without need for modification; the tricky part will be figuring out how to relate, manipulate and display these calculations on the interface.

There will be sums (due to the large number of particles), manipulation of arrays (for the coordinate and path-drawing system), graphing and plot manipulation, and other calculation principles that will come up as we work further on the project.

Potential Challenges

We expect the animation to take up a lot of computing power. Most of the calculations will have to occur behind the scenes, because the animation will already be performing many calculations based on the sheer number of particles in the container. There are two ways to do this simulation: we could draw each and every particle, and show them bouncing around as they interact with each other, but that animation requires that every particle's position be compared to every other particle to see if a collision happened. The route we will take relies on the modeling of the container using particle diffusion equations. We can still include multiple particles, but the other particles they are interacting with will not be visible. We want the calculations to happen all before the animation shows to prevent lag from interfering with the animation, and this will unfortunately limit the amount of time that the animation can be displayed. However, adding a time variable, as well as a start-position option will help simulate various scenarios.

Another challenge will be how we present the final product in a user-friendly manner. It will probably require a GUI with multiple options, text boxes and check marks, that can be filled by the user to allow for multiple different options. We also only want the user to have to run the program once, and be able to run multiple simulations from the same gui.