Professor feedback: This sounds like a fun project. I look forward to seeing what you come up with.

* a description of the topic you chose
* the results you expected
* the methods you used
* the results you found
* possible explanations for any discrepancies between what you measured and what you expected to measure

Description: We wanted to explore the accuracy of the root mean square equation in particle diffusion. Without accounting for temperature and velocity, we defined a particle’s movement by its free mean path using the equation , where n is the density of particles per unit volume, and is the cross section of a particle. The variables we allow for is the atom used for diffusion, the number of collisions you want to see, the number of atoms in the volume, and the dimensions of the box containing the particles. After several iterations, the mean distance travelled should be close to the approximation where N is the number of collisions.

Results expected: We expect that, after several iterations of simulating the diffusion, we should be able to see that the distance converges to the estimated distance from the equations above.

Our method: We calculated the movement of a particle in a fluid based on the number of collisions, number of particles and volume of the container. Though we didn’t directly use temperature and velocity, those values become less directly important when time is not taken into account. We simply wanted to see how far the particle would travel after a certain number of collisions. To determine a random path, we calculated the approximate mean free path using the equation for above. Then, we inserted some variability into the free path between each collision by adding or subtracting a somewhat random multiplier, which was weighted towards staying close to using an inverse exponential function (so greater random numbers keep the path close to the average, and smaller random numbers correspond with greater variability). Then we assigned random angle diffraction for each collision, both in the x-y plane and in the z direction. By incrementing the motion piece by piece, we found the coordinates of each collision and passed that into the Graphics3D function to plot the path. By using the distance formula on the first and last coordinate, we found the distance traveled over all for each iteration. We iterated this test 10 times for a cubic volume at a reasonable number of collisions and particles, and took the average of the distance travelled and compared it to the theoretical root mean square distance. Then, we conducted the same experiment on a few different container dimensions to see if shape of the container played a major effect. Note that we also did not account for the walls of the container; if the particle randomly makes its way through the container wall, we treat is as though it was still confined in a container in the same way as before.

Results found: Take a look, for example, at the first set of values we used. When we use hydrogen, and place 1 trillion particles in a cube with side length , we estimate the its is going to be equal to about . However, after we run 1000 iterations and find the mean distance traveled, we consistently see about travelled. If we run helium with 100 trillion particles, measuring over 100 collisions in a space , we expect to see a of 1.7 , but observe something closer to . When using 100 trillion atoms of argon, over 500 collisions in a 100 length cube, we expect on average, but find . The seen values vary with each run of 1000 trials, but in every case the error is seen to be about 5% from measured to expected, which is close but consistently low.

Possible discrepancies: In the random integer calculation that determines whether each path length will be larger or shorter than the mean free path, if there is any skew one way over the other, the simulation will give incorrect results. A more likely explanation is that in the equation we used, the mean free path was *estimated* to be equal to , so the actual value might be substantially different to create an error of approximately 5% in every case.