Total: 4.00/4.00
- Notebook: 4/4 Clear and readable. Could be more detailed.
- Data Presentation: 4/4 Key featured frames displayed. Described how particles are identified with great details. Calibration process clearly explained but without the frame of scaler. Captions can be much more informative.
- Data Handling: 4/4 Fitting processes shown with detailed plots. Key items included.
- Uncertainties: 4/4 Uncertainty calculated but with some possible mistakes. Good estimation on direct measurements.
- Results: 4/4 Results well organized. Clear presentation of the whole calculation procedure. Made a mistake in comparing expected value to itself.
- Conclusions: 4/4 Thorough discussion on the result with possible ways of improvement. Clear conclusions are made.

1 Introduction

1.1 Overview

Brownian motion is the motion that describes the random movement of particles within a medium. Following the theory given in the wiki as well as general literature, we can use the equipartition theorem to show the following:

$$\frac{\partial}{\partial t}(\langle x^2 \rangle) = \frac{2k_b T}{\mu} \tag{1}$$

$$\mu = 2\pi\eta a \tag{2}$$

We can then use Stoke's law for frictional force to describe the viscosity of a fluid, as seen in equation 2. Integrating equation 1 with respect to time, and substituting equation 2 for μ, we can write the following relation.

$$\langle x^2 \rangle = \frac{2k_b T}{\mu} t \tag{3}$$

We can then describe a diffusion coefficient as

$$\langle x^2 \rangle = \frac{2k_b T}{2\pi\eta a} t = 2Dt \tag{4}$$

Assuming that the distribution of motion in some direction x is ~~random,~~ then we can assume that the probability of a certain displacement can be described as a Gaussian distribution.

$$C(x) = \frac{N}{\sigma_y \sqrt{2\pi}} e^{(-\frac{(x-x_0)^2}{2\sigma_y^2})} \tag{5}$$

From equation 4, we know that we would like to know the mean squared displacement of the particle, equate it to 2*D*t, then solve for D. With a Gaussian distribution, this mean square displacement is equal to the standard deviation squared, or $\langle x^2 \rangle = \sigma^2$. So the general plan for the experiment is to measure the translational displacement in x and y, create appropriate gaussian fits for the displacement from their starting position, and divide σ^2 from this fit by 2*t to find the diffusion coefficient. This is shown in equation 6.

$$\sigma^2 = 2Dt \to D = \frac{\sigma^2}{2t} \tag{6}$$

We can then compare with the expected value given by dividing the second expression in the equation 4 by 2*t, such that

$$D = \frac{2k_b T}{6\pi\eta a 2t} t = \frac{k_b T}{6\pi\eta a} = \frac{k_b T}{6\pi\eta}\frac{1}{a} \tag{7}$$

We will end up with 2 different diffusion coefficients, D_e (expected diffusion coefficient) and D_m (measured diffusion coefficient). Afterwards, we can fit an inverse equation to our D_m data points like the one in equation 8. Then, we can compare this A value to the expected factor by multiplying equation 7 by a.

$$D(r) = \frac{A}{r} \tag{8}$$

1.2 Uncertainty

There are many different uncertainties that apply when creating the gaussian fit to the displacement in equation 5. A couple uncertainties that apply are the uncertainties in x displacement, y displacement, conversion from pixels to length (dc_p), the measured particle's radius (da), and the temperature (dT). To quantify these uncertainties, we assume that all are independent variables from one another and are gaussian distributed, and use the general root mean square deviation for each uncertainty expected that can be quantified, with a half accounting for the average of x and y.

$$d\sigma = \sqrt{(d\sigma_x)^2 + (d\sigma_y)^2 + (\frac{dc_p}{c_p})^2 + (\frac{da}{a})^2 + (\frac{dT}{T})^2/2} \tag{9}$$

The uncertainty of the displacement in each direction is calculated from the gaussian fits, σ_x and σ_y. The uncertainty in conversion, dc_p, from pixels to length was completed by taking many measurements of a micrometer, and taking the standard deviation of these measurements. The uncertainty in the particle's radius is taken as a certain percentage of the actual radius. We chose 2 percent of the radius ~~because of how small it is,~~ because these materials are lab produced and should have a small error in production. We attempted to find a definitive uncertainty from the labels, but were unable to. The room temperature can affect how fast the particles may move. Since the slide was only on the microscope for a couple of minutes, we assume that the temperature change was as small as 2 Kelvin. Rewriting our uncertainty as a function of the particle size, and uncertainty in displacements:

$$d\sigma = \sqrt{(d\sigma_x)^2 + (d\sigma_y)^2 + (\frac{2.7}{60.9})^2 + (0.02)^2 + (\frac{2}{297})^2/2} \tag{10}$$

Other uncertainties that may have arisen that could affect our results include any external interferences that could affect how the particles would move through the medium. For example, we may have inadvertently smacked the table causing the particles to move in one direction. Likewise, someone could simply close the door forcefully and make an impact on a particle's path.

2 Data

2.1 Data Handling

2.1.1 Procedure

In order to measure the displacement of the particles in each direction, we would need to record a video of the particles moving through a microscope and use software to measure the actual displacement in each direction. While taking data, we assume that the particles are perfectly circular.

To start, we would need to create a solution with the particles to analyze under the microscope. First, we would clean and rinse a new small plastic tube, the micropipette, the micropipette's tip, microscope slide, and glass coverslip with isopropyl alcohol. Next, we would use a large pipette to transfer doubly deionized water onto the clean plastic tube. Then, we would shake the container with the particles with the vortex genie device and use the micropipette to place either one or two drops of the particles into the plastic tube. We would then shake the tube and use the vortex genie device for about 30 seconds in order to make sure that the solution is well distributed within the tube. Likewise, we would clean the micropipette and its tip again with isopropyl to get rid of all of the particles. After cleaning and shaking, we would use the micropipette to place one drop of the solution onto the microscope slide, and use the cover to confine the particles to move in the liquid. We took special care to not place the drop on the edges of the microscope slide.

After placing the microscope slide with the solution on the microscope, we would use the lowest magnification lens (4x) to find the edge of the drop. Then do it again for the slightly higher magnification (10x), then use the next magnification (x40) to find the drop's edge. From there, we would use the focus controls and xy translation controls to find an area where the frame is focused on many particles of the same size. The camera was on the microscope and connected to the computer in order to live display what was under the microscope. The software completing this was the IC Capture software.

Firstly, we would check on the movement of the particles and make sure that we can notice they are moving, just with our eyes, random movements within the particles. A few times, we would create a dense enough solution that there were many particles, but little movement. If this was the case, we would dilute the solution by pouring some out and replacing it with more doubly deionized water. We would clean the microscope slide and its cover again with isopropyl alcohol, and place a drop of the newly diluted solution onto the clean microscope slide. After completing the task of finding the edge and focusing on a plane of particles, we would now be able to record a test sample.

We would first tweak the device properties to adjust the exposure, gain, and gamma so that the frame would show clearly the solution drop in dark and the background with white. Next we would select the toggle recording info, and make sure we were recording an uncompressed avi file. After creating the file's path and name, we would finally adjust the number of frames to capture, which we chose to use 300 frames for all of our recordings. We chose this because it would amount to about 30 seconds, with a 10 frames per second capture, of data. We believe this would capture particle motion. Finally, we can record the video and save it onto our directory with all of our recordings.

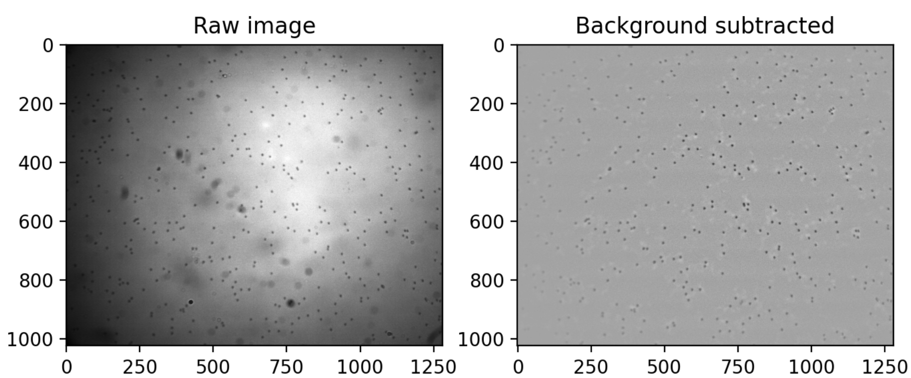

Figure 1: Background Subtraction

2.1.2 Data Calibration

To calibrate our data, we used the micrometer provided in the lab. We placed the micrometer under the microscope and took a short video once we had achieved focus on the tick marks. Then, we would open the video in ImageJ. We would take x and y positions of the tick marks close to the edge of the micrometer to ensure that we were taking measurements of a straight line. Then we took the pythagorean displacement, $d = \sqrt{x^2 + y^2}$, to find the amount of pixels between tick marks. We took 6 data points, each 10 micrometers or 1 tick mark apart, to find 5 pixel measurements for 10 micrometers. Taking the mean of these data, we found the average to be 60.85 pixels per 10 micrometers. For uncertainty in this measurement, we took the standard deviation of these measurements and found 2.73 pixels per 10 micrometers. This is shown in equation 10. Despite having different particles, we used the same setup throughout all measurements. We used the x40 objective lens for all videos. Even if there are small translations up and down, the amount of pixels per micrometer will stay the same because there would need to be some focusing to calibrate. In effect, we assume that this conversion can be used for all measurements.

2.1.3 Measurement with Jupyter Notebook

After we ~~recorded the video and had it saved~~, we had to complete some image processing in order to measure the displacements. To complete this image processing, we used many python packages like pandas, numpy, pims, and trackpy. In general, we complete this by reading in the videos, locating the important features, and linking these features from the video's frame to a particle's trajectory.

First, we read in the data and open the video using pims. We then use pims again to complete some background subtraction. This helps set a uniform background and create frames that we find much easier to identify particles from. This is shown in figure 1 (all subsequent figures in this section will show data for the video recording of the particle Silica with a radius of 0.53 μm).

Afterwards, we locate the relevant particles that are in these frames. First, we use trackpy's locate method in order to store all of the particle's features, which include a particle's position, mass, size, and eccentricity. The mass is estimated as the intensity of the darkness in the frame. The eccentricity is simply the particle's eccentricity, and the size is just the particle's size. However, lots of spurious features pop up that are just fleeting peaks of brightness, so we set a minimum mass when locating a frame's features. ~~In effect, we set a minimum mass that a feature must have to be extracted.~~

Then, we create an interactive window that shows the features located with a red circle around it. This interactive window allows us to tweak the parameters for the specific features we are interested in: mass, size (or diameter), and the space needed between two particles to be considered in our analysis. If two particles are too close to each other, then they may be interacting with each other in a way that would not be described as brownian motion. The green dot in the top left is an indicator for the size that the particle is estimated to with the slider's information.

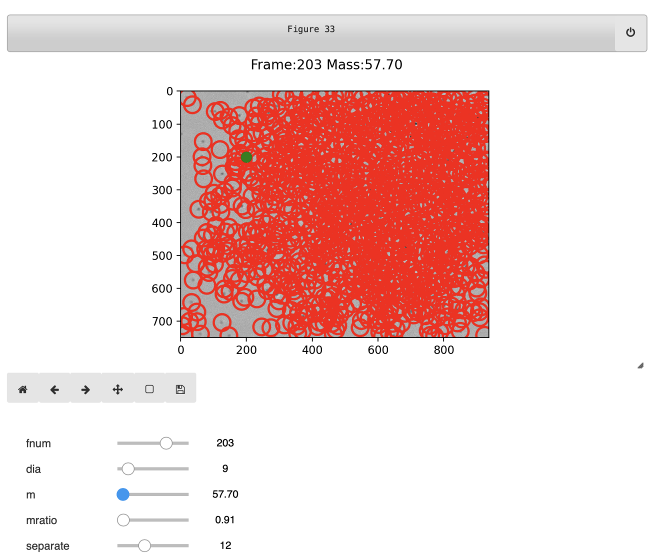

Figure 2: Low Minimum Mass

The reason we create this interactive window is to make sure we include the particles that behave well and are clearly particles. If we were to set the mass as very low, then the minimum threshold to be a particle would extend to potential spurious features again. This is shown in figure 2. On the other hand, if we were to extend the minimum separation to a large amount, we could discount many of the particles that are experiencing Brownian motion, as shown in figure 3. In general, we would tweak the parameters to include most of the particles that are not clumped together, and are clearly particles. An example of this is shown in figure 4. We would then apply the size, minimum mass, and separation onto the data.

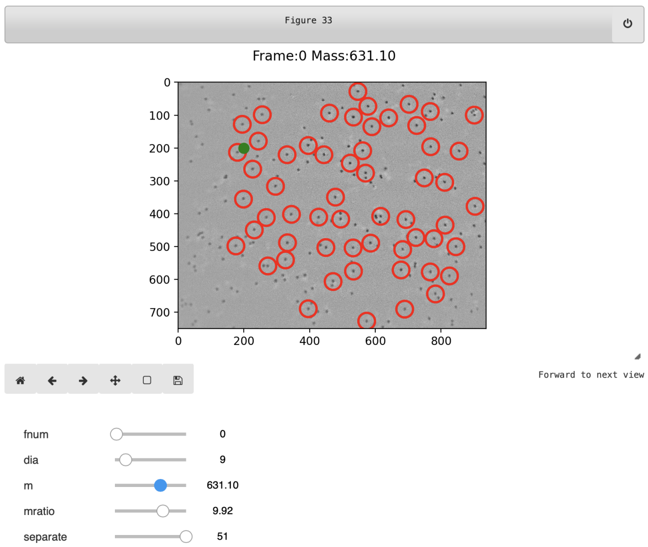

Figure 3: High Minimum Separation

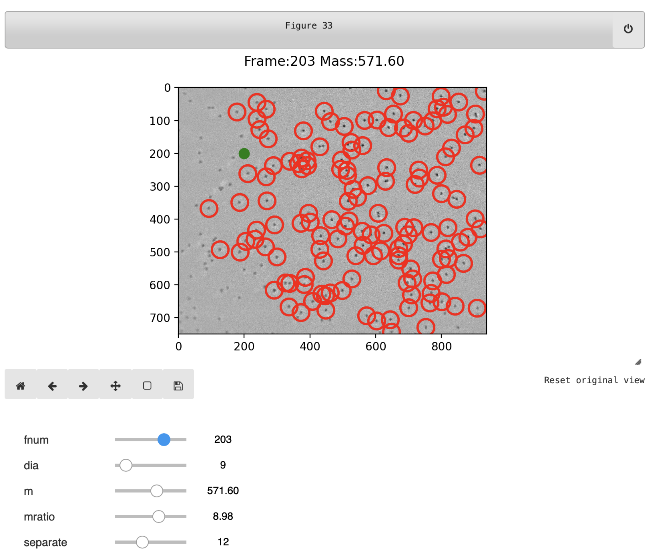

Figure 4: Well Optimized Parameters

Next, we would like to link the features into actual particle trajectories. To do this, we would use trackpy's link method to link each frame's information with each other. This would connect and pick out a particle's trajectory over time. Then, we can apply the condition that a trajectory lasts on the video for a certain amount of frames using trackpy's filter stubs method.

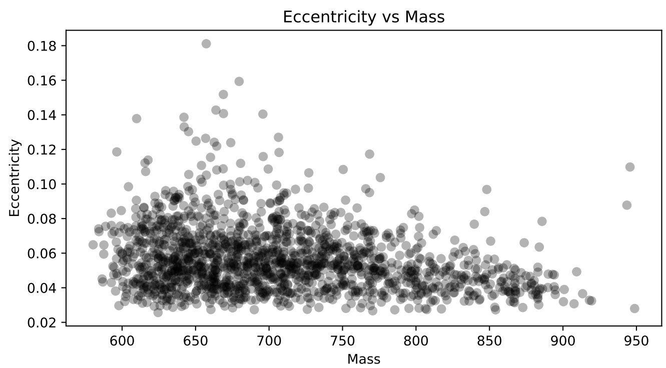

Figure 5: Eccentricity Mass Distribution

Afterwards, we filter the data one more time. We plot the eccentricity vs mass and size against mass, as shown in figures 5 and 6. To get rid of particles that may have been out of focus or those with abnormal characteristics, we only choose a feature's average appearance over the trajectory. To do this, we set another minimum and maximum mass limit, as well as a maximum size and maximum eccentricity in order to account for only the greatest clump of data. We then get rid of these particles and link the trajectories within each other. For the final filter of data, we subtracted any overall drift a trajectory may have had using trackpy's compute drift and subtract drift method.

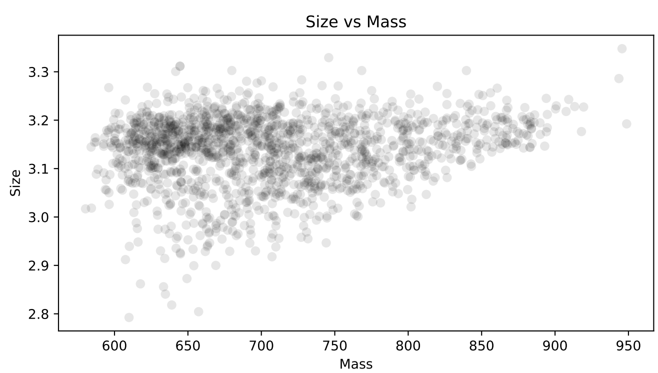

Figure 6: Size Mass Distribution

Now that we have appropriately filtered out all of the data, we now can look at each trajectory and extract the displacement in x, displacement in y, and the particle id for each frame.

2.1.4 Gaussian Fits

Having gone through the processing to complete our measurement of displacements, we are left with some displacements in each direction that we can then fit to the gaussian in equation 5. We first get the displacement in a certain direction and create a histogram. Assuming this is a gaussian distribution, we can have the uncertainty in these displacements be the square root of the data, with a floor of 1.4 in order to agree with Poisson uncertainty with N=0. These gaussian fits are shown in figures 7 and 8.

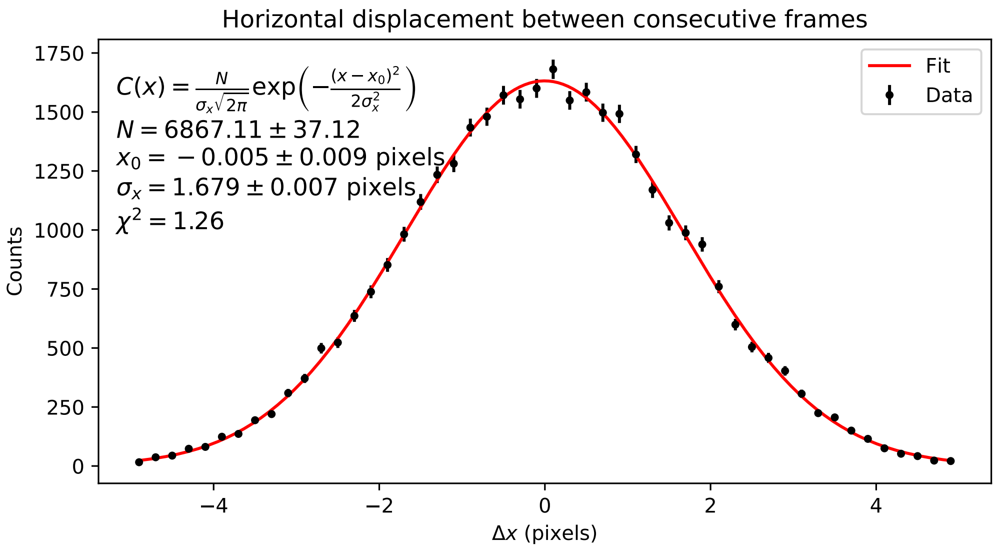

Figure 7: Gaussian Fit to x Horizontal Displacement

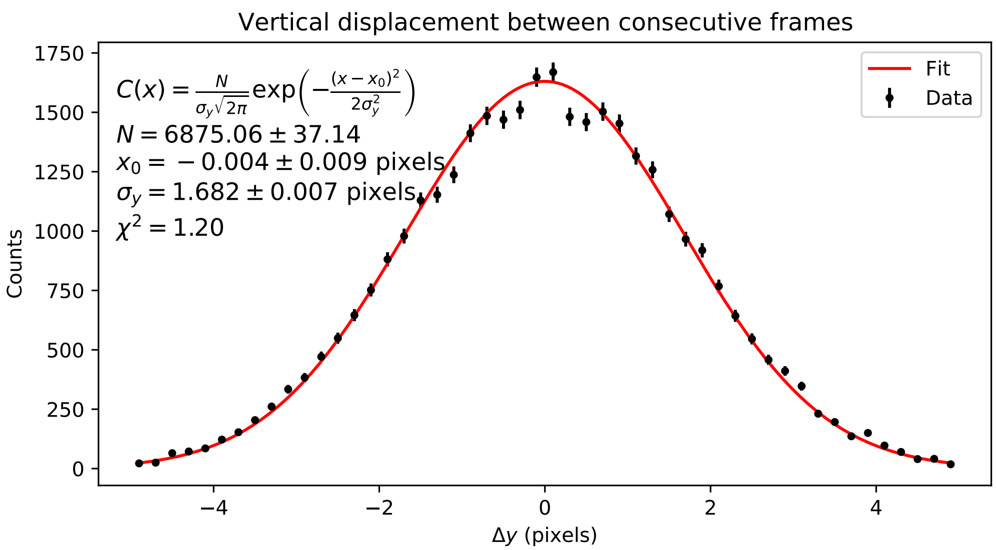

Figure 8: Gaussian Fit to x Vertical Displacement

2.1.5 Conversion from σ to D_m

We then take the average of the step sizes using equation 11 and convert to length by multiplying with the conversion of micrometers per pixel.

$$\sigma = \frac{|\sigma_x| + |\sigma_y|}{2} \tag{11}$$

We can then convert from the variance to diffusion coefficient by equation 12.

$$D_m = \frac{\sigma^2}{2dt} \tag{12}$$

For the change in time, we took the video at a frame rate of 10 frames per second. So we can take the dt in equation 12 as 1/10. For the uncertainty in the diffusion coefficient, we assume that there is no uncertainty in the time, as this is already set by the computer and camera. So the uncertainty in the diffusion coefficient is just the uncertainty in σ, as shown in equation 13. We can then use both equation 11 and equation 9 to simplify.

$$dD_m = D_m \frac{\sqrt{(d\sigma_x)^2 + (d\sigma_y)^2 + (\frac{dc_p}{c_p})^2 + (\frac{da}{a})^2 + (\frac{dT}{T})^2}}{|\sigma_x| + |\sigma_y|} \tag{13}$$

With this, we now have a measured diffusion coefficient and its uncertainty. The expected diffusion coefficient from equation 7, can be translated onto equation 14. The uncertainty in this expected diffusion coefficient would just be the uncertainty in temperature and the particles radius, as detailed in equation 15.

$$D_e = \frac{k_b T}{6\pi\eta}\frac{1}{a} \tag{14}$$

$$dD_e = D_e\sqrt{(dT/T)^2 + (da/a)^2} \tag{15}$$

A table comparing these values can be found in table 2.1.5.

Diameter (μm)	$D_p(10^{-13}m^2s^{-1})$	$D_e(10^{-13}m^2s^{-1})$
1.05	4.65 ± 0.04	3.80 ± 0.05
1.49	3.28 ± 0.03	1.89 ± 0.03
4.19	1.17 ± 0.01	1.17 ± 0.03
1.04	4.70 ± 0.04	3.84 ± 0.07
2.07	2.36 ± 0.02	1.71 ± 0.06

Table 1: Diffusion Coefficients and Uncertainties found.

2.1.6 Inverse Fits

Now that we have the diffusion coefficient for a particular particle's radius, we can now complete a fit to equation 8, which is an inverse relationship between the two quantities. We can complete this fit for both the measured and expected values and compare both against the expected coefficient from equation 7. This fitting can be seen in figure 9. A table describing the quantities can be seen in table 2.1.5. D_m is the diffusion coefficient for the measured quantities, and A_m is the associated coefficient with fitting the D_m values. Likewise, A_e is the coefficient for fitting the D_e, expected diffusion coefficient values, to an inverse relationship. The coefficient to compare against is $A_{lit} = \frac{k_b T}{6\pi\eta}$.

$A_{lit}(10^{-13}ms^{-1})$	$A_m(10^{-13}ms^{-1})$	$A_m(10^{-13}ms^{-1})$
2.44	1.81 ± 0.02	2.45 ± 0.01

Table 2: Diffusion Coefficients and Uncertainties found.

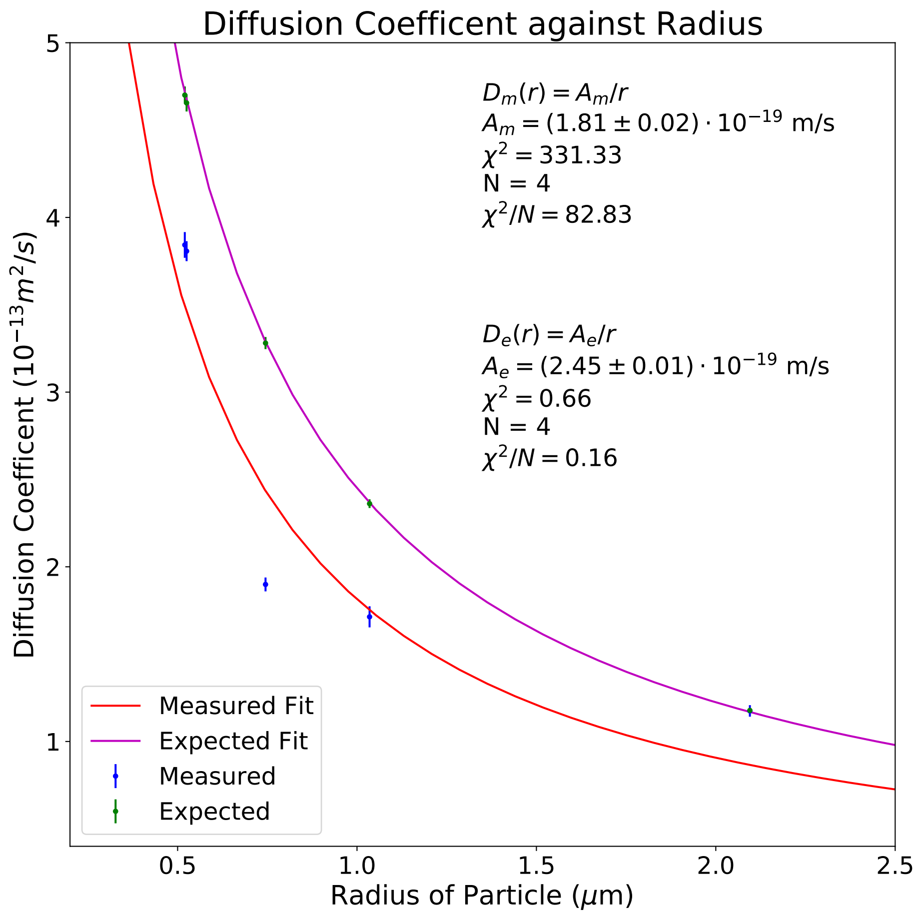

Figure 9: Fit for Coefficient

3 Discussion

Looking at figure 9, we can see that we cannot conclude with certainty that our measured data concludes an inverse relationship. The reduced chi-squared value is far from the optimal value of 1, indicating that this is not a good fit. On the other hand, we can see that there is a relationship for the diffusion coefficient that was expected, and the associated coefficient is within the uncertainty of what the expected value is, as can be seen in table 2.1.6. Something interesting to note is that most of our data are underestimating what the expected value is. Since the diffusion coefficient and the displacements are quadratically related, we may have completed some action in the measurement that underestimates the actual amount of movement made. This could be up to the code written to measure these displacements. However, this is not likely since the difference between the expected and measured diffusion coefficients don't really have a proportional relationship. There is not a clear difference between the two. The largest particle's diffusion coefficient does fall within the expected value's uncertainty. Another interesting property of the plots is the low reduced chi squared for the diffusion coefficient for the expected fit is small. Since it is below 1, this may indicate that our uncertainty analysis was poor and has to be revisited to account for other uncertainties. For example, we could more precisely quantify the uncertainty in temperature by using a thermometer and finding how long the temperature changes over time. For a future experiment trying to replicate or build upon this analysis, more data would have helped tremendously. Since this relationship should be independent of number density, we could take multiple videos for particles of the same size and slowly dilute it over time. We could then take the average of the measured diffusion coefficients. We tested it here, but a future experiment could more robustly support the idea that a particle's material does not factor in. This is because from the data taken, we measured the diffusion coefficient for 3 silica and 2 PolyStyrene. An assumption made pretty early on was that the particles are perfect spheres, but this may not strictly be true. The conclusion is that we cannot say that our data supports the expected inverse relationship from Brownian motion of the medium's diffusion coefficient and the size of the particle.