

Physics 360-0: Brownian Motion

Jun Sung, Maura Lally
(Dated: 06-08-21)

The purpose of this lab is to observe the Brownian motion of silica beads (of various diameters) suspended in water and measure the diffusion and Boltzmann constant. To achieve this, we view these beads under a microscope and use particle-tracking software to measure the mean-squared displacement of these beads over a period of time. We then use a simple model from statistical mechanics that gives us a relation between the diffusion constant and the mean-squared displacement. From this relation, we can extract both the diffusion and Boltzmann constants. Furthermore, we analyze whether changes to the bead density, frame rate, and track length have any noticeable effects on calculating the two constants. We find that we are generally successful in recovering the expected Boltzmann constant k_B : The majority of our results were within 2σ of the established value of k_B . Generally, our results were closest to the expected value when using a $1\mu\text{m}$ sample at any dilution, with a frame rate of 10 and a track length of 15 or shorter.

I. INTRODUCTION

Consider a particle is suspended in a fluid (either liquid or gas). If we place ourselves in the shoes of this particle, then we see that we would be constantly bombarded from all sides by the constituents of the fluid. This bombardment results in the random motion of the particle, and we refer to such motion as Brownian motion.

Some might question why such a motion is important enough to be named, to which we would reply that it provided some of the first evidence of atoms. Furthermore, if one looks close enough, then they would find Brownian motion to exist in nearly every aspect of our lives (although it does not affect the macro world much, if at all).

In terms of how Brownian motion could be explained, Einstein published a paper in 1905 doing exactly that by showing that Brownian motion could be explained by assuming that fluids were composed of atoms moving randomly with an average kinetic energy.

For this lab, we will observe the Brownian motion of silica beads suspended in water and use particle-tracking software to measure the mean-squared displacement of these beads over a period of time. We then use a simple model from statistical mechanics to look at how diffusion scales with particle size.

A. Brief Overview of Methods

This lab will consist of one main experiment along with a few extensions. Starting with the main experiment, we will use the microscope and camera to record a 300-frame video of a slide with a drop of silica beads suspended in water. We then analyze the bead motion it captured via `TrackPy` and calculate the diffusion coefficient (as well as k_B). This process will be done for several particle sizes.

As for the extensions, we will look into how particle density (the concentration of the water drop), video frame rate (how many frames per second are captured by the camera), and track length (the minimum length that particle tracks must be in order to be included in the fi-

nal analysis) affects the diffusion constant and the way that we analyze our data (e.g. we might adjust other parameters in `TrackPy` differently in order to obtain a result).

B. Applications

One area where Brownian motion can be applied (that might be a surprise to some) would be in finance. The stock market (and other markets in general) is assumed to follow Brownian motion [1]. In fact, the models used to describe Brownian motion are used to model financial asset and derivative pricing, which can further be used for risk analysis.

Brownian motion is also applied in nanotechnology. As stated earlier, Brownian motion has little to no effect in the macro world. However, when we are dealing with the micro world, the effects of Brownian motion can no longer be ignored. In fact, it turns out that Brownian motion is one of the main controlling factors in the the realm of nanotechnology [2]. For example, the fluid can exert forces on a nano-machine that could potentially cause it's parts to stick together. Furthermore, the Brownian motion that a nano-machine undergoes could make it hard to control it's movement, which could impede the machine from doing what it was designed to do.

II. THEORY

As noted in the lab manual, the theory of Brownian motion of spherical particles can be derived from statistical mechanics. In particular, the velocities of small particles suspended in a fluid due to collisions with molecules obey a Maxwellian velocity distribution. We will follow the steps outlined in [3] to derive the diffusion constant D in terms of the fundamental parameters of the particles and liquid. Because Brownian trajectories are continuous, random, and irregular, it would be hard for us to work with the data as-is. However, we can work instead with the mean-squared displacement for each trajectory.

A. Mean-Squared Displacement Derivation

We start by introducing the Langevin equation of a particle which has mass m and velocity v :

$$m \frac{dv}{dt} = -\alpha v + F(t) \quad (1)$$

where α refers to the dampening constant and $F(t)$ is a random fluctuating force due to molecular collisions. Furthermore, if this particle is a sphere, then α is given by Stokes flow for a sphere. Explicitly, we have

$$\alpha = 6\pi\eta a \quad (2)$$

where η is the dynamic viscosity and a is the radius of the particle. From here, we multiply both sides of Equation 1 by x to get

$$mx \frac{dv}{dt} = -\alpha xv + xF(t) \quad (3)$$

where we note that

$$\frac{d(xv)}{dt} = x \frac{dv}{dt} + \frac{dx}{dt} v = x \frac{dv}{dt} + v^2 \iff x \frac{dv}{dt} = \frac{d(xv)}{dt} - v^2. \quad (4)$$

Thus, substituting in Equation 4 into Equation 3 results in

$$m \left[\frac{d(xv)}{dt} - v^2 \right] = -\alpha xv + xF(t). \quad (5)$$

We now take the time average of Equation 5

$$\left\langle m \left[\frac{d(xv)}{dt} - v^2 \right] \right\rangle = m \left[\frac{d\langle xv \rangle}{dt} - \langle v^2 \rangle \right] = -\alpha \langle xv \rangle + \langle xF(t) \rangle. \quad (6)$$

Now because $F(t)$ varies randomly (irrespective of the velocity or position of the particle), we have

$$\langle xF(t) \rangle = \langle x \rangle \langle F(t) \rangle = 0. \quad (7)$$

With this, we have that Equation 6 can be rewritten as

$$m \frac{d\langle xv \rangle}{dt} = m \langle v^2 \rangle - \alpha \langle xv \rangle. \quad (8)$$

The Equipartition Theorem tells us that

$$\frac{1}{2} m \langle v^2 \rangle = \frac{1}{2} k_B T \quad (9)$$

where k_B is Boltzmann's constant and T is the temperature. Substituting in the result we got from Equation 9 into Equation 8, we get

$$m \frac{d\langle xv \rangle}{dt} = k_B T - \alpha \langle xv \rangle. \quad (10)$$

It is from here that we note that

$$\begin{aligned} \langle xv \rangle &= \frac{1}{T} \int_0^T dt \, xv \\ &= \frac{1}{T} \left[x^2 \Big|_0^T - \int_0^T dt \, xv \right]. \end{aligned} \quad (11)$$

Utilizing the Fundamental Theorem of Calculus and combining the integrals together gives us

$$2 \frac{1}{T} \int_0^T dt \, xv = \frac{d\langle x^2 \rangle}{dt} \iff \langle xv \rangle = \frac{1}{2} \frac{d\langle x^2 \rangle}{dt}. \quad (12)$$

Substituting Equation 12 into Equation 10 gives us

$$\frac{m}{2} \frac{d^2 \langle x^2 \rangle}{dt^2} + \frac{\alpha}{2} \frac{d\langle x^2 \rangle}{dt} = k_B T. \quad (13)$$

We now define the following quantity:

$$w \equiv \frac{d\langle x^2 \rangle}{dt} \quad (14)$$

which results in Equation 13 to be

$$\frac{dw}{dt} + \frac{\alpha}{m} w = \frac{2k_B T}{m}. \quad (15)$$

If we further define $\gamma \equiv \frac{\alpha}{m}$, then we have Equation 15 to further simplify into

$$\frac{dw}{dt} + \gamma w = \frac{2k_B T}{m}. \quad (16)$$

This is a simple first-order differential equation whose solution can easily found to be

$$w = \frac{2k_B T}{\alpha} + C e^{-\gamma t}. \quad (17)$$

It is from here that we note that m is small and that $t \gg m$ in our case. Thus we have that the dominating behavior in Equation 17 is that of the first term. Hence, we have that

$$w \approx \frac{2k_B T}{\alpha}. \quad (18)$$

Once we have this, we can use Equations 14 and 18 to obtain an expression for the mean-square displacement:

$$\frac{d\langle x^2 \rangle}{dt} = \frac{2k_B T}{\alpha} \iff \langle x^2 \rangle = \frac{2k_B T}{\alpha} t = 2Dt \quad (19)$$

where D is defined as the diffusion coefficient and is given by

$$D \equiv \frac{k_B T}{\alpha} = \frac{k_B T}{6\pi\eta a}. \quad (20)$$

B. Measuring the Diffusion Coefficient

There are two methods that we can utilize to calculate the diffusion coefficient once we know the mean-squared displacement. We describe these below in Sections II B 1 and II B 2.

1. Method 1

As priorly mentioned in this report, Brownian trajectories are continuous, random, and irregular. As a result of this (namely, the random displacement of the particle), we have that the probability for any particular displacement (say, x) in a given time interval t is given by the Gaussian distribution:

$$P(x) = \sqrt{\frac{1}{2\pi\sigma^2}} e^{-x^2/2\sigma^2} \quad (21)$$

where the width of the distribution is related to the diffusion coefficient by the following:

$$\sigma^2 = \langle x^2 \rangle = 2Dt. \quad (22)$$

It is through the relation given in Equation 22 that allows for us to extract the diffusion coefficient experimentally. To expand on this, we are able to measure the displacements of several particles along a certain dimension (e.g. along the x - or y -axes), which motion we are able to plot on a normalized histogram and fit to Equation 21. From the fit, we are then able to extract σ^2 , which then gives us the diffusion coefficient.

2. Method 2

With the use of **TrackPy**, we are able to compute the ensemble mean-squared displacement (EMSD) of all particles. Furthermore, we are able to graph out the EMSD over time, which we can then perform a linear fit on. The slope of this linear fit is equivalent to $2D$ (refer to Equation 19), which immediately gives us the diffusion coefficient. It should be noted that we perform the linear fit on the EMSD vs. time graph for *one* dimension (in our case, either along the x - or y -axes) – not both.

III. APPARATUS

A. Colloidal Samples

The particles that we suspend in water are polystyrene spheres of approximately uniform size. There are a total of five samples that we could choose from that each have a different diameter – these being $0.52 \mu\text{m}$, $0.76 \mu\text{m}$, $1 \mu\text{m}$, $2.12 \mu\text{m}$, and $2.86 \mu\text{m}$.

B. Sample Preparation Tools

The tools we used to prepare the colloidal samples are as follows:

- Microscope slides;
- Square, glass coverslips;

- Double-sided tape;
- Exacto/surgical knife;
- Tweezers;
- Capillary tubes and tubes with caps;
- Pipettes; and
- A shaking/vortex device

C. Microscope

To view and record the particles, we used a compound binocular microscope. As noted in the lab manual, this microscope has the following characteristics that are most important for this lab:

- A specimen stage and holder that translates along both the x - and y -axes;
- Course and fine focus controls;
- A rotating turret with three objectives: 10x, 20x, and 40x;
- A camera port that allows us to couple the CMOS camera into the optical path of the microscope;
- A variable-intensity light source; and
- An Abbe condenser with an adjustable iris.

The microscope and its several components are illustrated in Figure 1. It should be noted that the condenser is used to focus the light coming from the light source into a cone. If properly aligned, then the cone narrows to a point at the sample, and then expands back into a cone of light that enters the objective an angle (i.e. the cone angle) that matches numerical aperture (or acceptance angle) for said objective. Furthermore, adjusting the position of the condenser, the diameter of the iris, and brightness of the light source are all vital towards getting the best image possible.

D. Camera and Programs

For video recording, we used a CMOS camera (model FLIR Blackfly S BFS-U3-32S4M). As discussed in the microscope section, the camera is coupled to the microscope and is also connected to the computer via USB. As for actually seeing the output of the camera, the **Spinview** application is needed, and the application allows for us to take both images and videos. Some of the key settings that **Spinview** provides are the following:

- Options to adjust the exposure time and frame rate;
- Options to adjust the gamma correction to the output of the camera; and

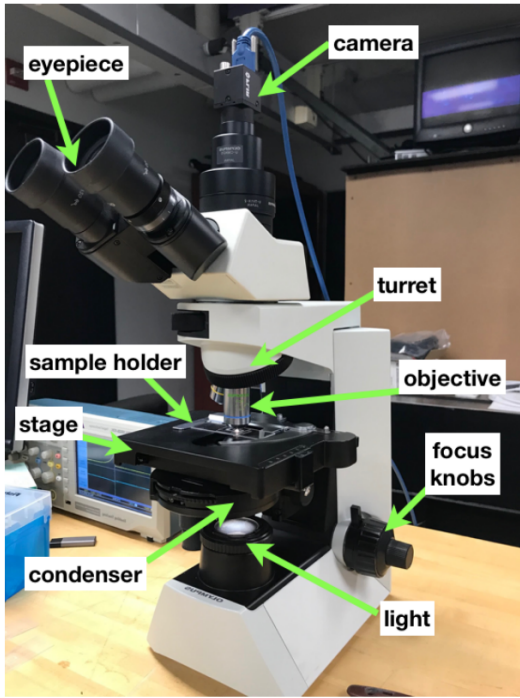


FIG. 1. Image of the microscope with its features labeled

- Ability to record videos as a batch of TIFF images.

The other programs that we used were **ImageJ** and **TrackPy**; these two programs allowed for us to analyze the data and calculate the diffusion constant (as well as k_B).

IV. MEASUREMENTS AND DATA ANALYSIS

A. Data Collection

We note that the following procedure was done for all five colloidal samples.

1. Preparing the Colloidal Sample

In order to view the colloidal sample under the microscope, we first had to dilute it. We decided to dilute each sample 15 to 1 water to solution (where we used a pipette to get the measurements close to exact), and made sure to vortex the colloidal sample well before mixing it in with water in a tube (that could be closed to prevent the water in the sample from evaporating too much). We then vortexed the contents of the tube to form a diluted homogeneous mixture of the colloidal sample. From here, we prepared this mixture onto the microscope slide, and there were two ways that we did this.

The first method involved taping a small strip of double-sided tape on a microscope slide. We then used a knife to cut a small rectangular hole in the tape –

small enough so that the coverslip covers the entire hole. We then pipetted a small amount of our diluted mixture (about $10\ \mu\text{L}$) into the rectangular hole making sure to not overfill it. We then used tweezers to place a coverslip over the hole and gently pressed the coverslip down to ensure a tight seal with the double-sided tape.

The second method was much simpler; it consisted of dipping a capillary tube into the tube that contained our diluted mixture. Once enough of the mixture filled the tube, we then used a wipe to remove any residue on the outside of the tube. Finally, we taped the capillary tube onto the microscope slide.

It should be noted that we got the capillary tubes about a week and a half into the lab, so we prepared the microscope slides using both methods. We re-prepared the slides from our prepared mixtures each day if needed, because the slides from the previous day had dried between lab days. Also, we remade the diluted mixtures if they were a few days old in case the dilution ratio changed at all from the water evaporating.

2. Using *Spinview* to Capture Videos

Once the colloidal sample was prepared, we viewed it under a microscope using the 40x objective. Again, the program that we used to see the output of the camera was **Spinview**. We made sure to adjust the focus, the gamma correction, the exposure time, the position of the condenser, the diameter of the iris, and the brightness of the light source so that the image is as clear as it can get; having a clear image is vital for **ImageJ** and **TrackPy** to work correctly. Once we were satisfied with the image quality, we then set the frame rate of the camera output to be 10 frames per second and recorded a 300 frame video (more accurately, we captured 300 TIFF image files).

B. Data Analysis

1. Using *ImageJ* and *TrackPy* to Calculate the Diffusion Coefficient

We can now hand the video over to **TrackPy**, which will help us calculate the diffusion constant. We followed the procedure outlined in the walkthrough section of the **TrackPy** documentation, and the full details of using the program can be found there [4]. In short, though, we use **TrackPy** to identify the particles in the video. On top of that, **TrackPy** tracks and records the x and y displacement of each particle that it identified. However, these displacements are given in units of pixels, and we need to convert it into microns.

This is where **ImageJ** is helpful. We are able to convert from pixels to microns using a stage micrometer that has rulings inscribed on it in increments of 0.1 mm and 0.01 mm. Since we only used the 40x objective, it suffices

that we find the micron-pixel conversion rate under the 40x objective. Figure 2 shows the image that we captured. We then exported this image to **ImageJ** and used

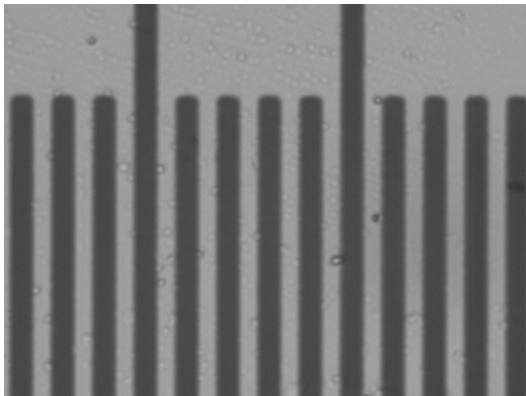


FIG. 2. Image of the stage micrometer under 40x objective

the program to find how many pixels were needed to get a length of 0.1 mm. In our case, we got a conversion of 0.0880 microns per pixel.

We also note that we used **ImageJ** to find the approximate diameter (in pixels) of a particle in our diluted mixture. This information was used to tell **TrackPy** to identify particles of approximately this diameter.

Now that we have the micron-pixel conversion, we are able to extract/calculate the diffusion coefficient from the displacement data that we gathered. We utilize method 2 as described in the section on measuring the diffusion constant. As already noted in that section, we use **TrackPy** to calculate the EMSD of all the particles and graph this result out over time. It should be noted that **TrackPy** records the net displacement as well as the displacement along the x - and y -axes. We only need the displacement data along the x -axis (or the y -axis, whichever gives us a better linear fit – however, we should expect the linear fit from either axes to give us the same result) since our model only accounts for one dimension. It should be noted that we can easily extend it to two, if we want, by noting that $\langle x^2 \rangle = \langle y^2 \rangle$. Thus:

$$\langle r^2 \rangle = \langle x^2 \rangle + \langle y^2 \rangle = 2\langle x^2 \rangle \implies \langle r^2 \rangle = 4Dt \quad (23)$$

We then perform a linear fit of the EMSD vs. time graph, whose slope gives us $2D$ (where D is the diffusion constant). From this, we can immediately get the diffusion constant, which then immediately gives us k_B . Figure 3 is an example of what **TrackPy**'s particle-tracking outputs, and Figures 4 and 5 are the corresponding plots of the particle's trajectories in position space as well as the EMSD over time. As expected, we do indeed see in Figure 5 that the relationship between EMSD and time is linear.

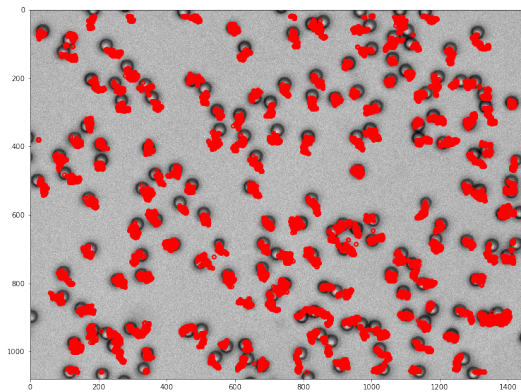


FIG. 3. Recorded trajectories of particles done by **TrackPy**

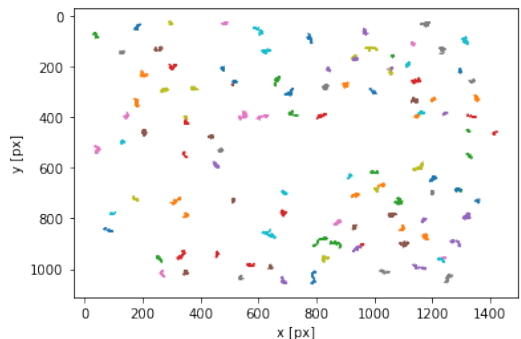


FIG. 4. Plot of recorded trajectories of particles in the xy -plane

2. Extensions

We now perform three extensions to our existing procedure: varying the particle density of our mixture, varying the frame rate of the camera output, and varying the track length used by our python notebook to calculate our final results. For the first two extensions, we work with the colloid sample that contains particles with $1 \mu\text{m}$ diameter, and for the track length extension we include trials with the $2.12 \mu\text{m}$ sample.

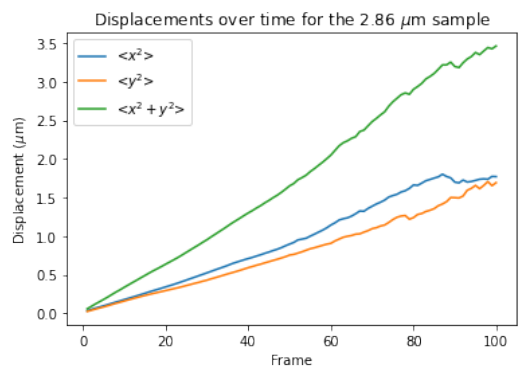


FIG. 5. Plot of EMSD vs. Time

For the extension that involves varying the particle density of our mixture, we repeat the procedure already described, but with the sample now being diluted 5 to 1 as well as 30 to 1 ratios of water to solution.

For the extension that involves changing the frame rate of the camera output, we again repeat the procedure, but with a frame rate of 5 as well as 15.

For the extension that involves changing the track length, we use trials which resulted in basically accurate measurements of k_B (15 to 1 dilutions of 1 μm and 2.12 μm samples) and re-ran our python notebook, now with the track length setting varied to be shorter and longer than the initial setting. This changed the cutoff length of the track of the particles located by **TrackPy** for inclusion in the final calculation of our constants—a higher track length is a more stringent requirement, so fewer points were included in the final calculation, and vice versa.

Again, the main goal of these extensions is to see how varying these three parameters will affect the diffusion constant and result for k_B as well as how it might change the way we analyze our data.

C. Results

Our results of our main experiment are shown in Table I. As described in the caption of that table, these results were obtained from 15:1 water to solution samples, captured with a frame rate of 10. Different track lengths were used for each size, but the setting was consistent for all trials of the same size sphere. The quoted uncertainty is obtained from a standard deviation of our results over several trials. The % Error column is the % difference between our experimental result for k_B , and the established value of $k_B = 1.3806 \times 10^{-23}$.

Sphere Size (μm)	D	k_B	Error %	σ
0.52 μm	0.35 ± 0.01	$(0.56 \pm 0.02) \times 10^{-23}$	6.0	$> 10\sigma$
0.76 μm	0.54 ± 0.04	$(1.25 \pm 0.08) \times 10^{-23}$	1.0	2σ
1 μm	0.43 ± 0.03	$(1.30 \pm 0.10) \times 10^{-23}$	0.6	1σ
2.12 μm	0.23 ± 0.01	$(1.46 \pm 0.05) \times 10^{-23}$	0.5	2σ
2.86 μm	0.25 ± 0.05	$(2.13 \pm 0.41) \times 10^{-23}$	5.5	2σ

TABLE I. Results for diffusion constant D and Boltzmann constant k_B for various polystyrene sphere sizes. Each of these results is a mean value based on several trials. Each was obtained from measurement and analysis of a 15:1 water to solution sample, captured with a frame rate of 10. The track length varied across different sized samples, but was consistent within trials for the same size sample.

1. Particle Density Extension

Results of our first extension, comparing results across different dilutions of our 1 μm sample, are shown in Table II.

Dilution	D	k_B	Error %	σ
5 : 1	0.46 ± 0.02	$(1.39 \pm 0.07) \times 10^{-23}$	0.1	1σ
15 : 1	0.43 ± 0.03	$(1.30 \pm 0.10) \times 10^{-23}$	0.6	1σ
30 : 1	0.45 ± 0.01	$(1.38 \pm 0.03) \times 10^{-23}$	0.0	1σ

TABLE II. Results for diffusion constant D and Boltzmann constant k_B for samples of various dilutions. Each of these results is a mean value based on several trials. Each was obtained from measurement and analysis of a 1 μm sample, captured with a frame rate of 10 and a track length consistent across measurements from the same dilution.

2. Frame Rate Extension

Results of our second extension, comparing results across different frame rates, are shown in Table III.

Dilution	Frame Rate (fps)	D	k_B	Error %	σ
15 : 1	5	0.41 ± 0.03	$(1.25 \pm 0.09) \times 10^{-23}$	1.0	2σ
15 : 1	10	0.43 ± 0.03	$(1.30 \pm 0.10) \times 10^{-23}$	0.6	1σ
15 : 1	15	0.28 ± 0.01	$(0.85 \pm 0.03) \times 10^{-23}$	3.8	$> 10\sigma$
30 : 1	5	0.45 ± 0.01	$(1.38 \pm 0.03) \times 10^{-23}$	0.0	1σ
30 : 1	10	0.45 ± 0.01	$(1.38 \pm 0.03) \times 10^{-23}$	0.0	1σ
30 : 1	15	0.46 ± 0.02	$(1.40 \pm 0.05) \times 10^{-23}$	0.1	1σ

TABLE III. Results for diffusion constant D and Boltzmann constant k_B for different frame rates; conducted on both 15 : 1 and 30 : 1 dilutions of the 1 μm solution. Each of these results is a mean value based on several trials.

3. Track Lengths Extension

Results of our third extension, comparing results across different track lengths, are shown in Table IV.

Sphere Size (μm)	Track Length	D	k_B	Error %	σ
1 μm	5	0.42 ± 0.03	$(1.29 \pm 0.10) \times 10^{-23}$	0.6	1σ
1 μm	15	0.43 ± 0.03	$(1.30 \pm 0.10) \times 10^{-23}$	0.6	1σ
1 μm	50	0.20 ± 0.09	$(0.60 \pm 0.26) \times 10^{-23}$	5.7	3σ
2.12 μm	5	0.23 ± 0.01	$(1.48 \pm 0.04) \times 10^{-23}$	0.7	3σ
2.12 μm	25	0.23 ± 0.01	$(1.45 \pm 0.05) \times 10^{-23}$	0.5	2σ
2.12 μm	50	0.22 ± 0.001	$(1.41 \pm 0.01) \times 10^{-23}$	0.2	5σ
2.12 μm	115	0.15 ± 0.04	$(0.99 \pm 0.28) \times 10^{-23}$	2.8	2σ

TABLE IV. Results for diffusion constant D and Boltzmann constant k_B for various track lengths. Each of these results is a mean value based on several trials. Each was obtained from measurement and analysis of a 15:1 water to solution sample, captured with a frame rate of 10.

V. DISCUSSION

The majority of our results are within 2σ of the established k_B value we expect to find from theory. The exception to this from our main experimental results is the result from the $0.52\mu\text{m}$ sample. Of all of our measurements, this is the one with the greatest percent error, but it also has one of the smallest uncertainties. This means that our measurements were consistently yielding small diffusion coefficients (D) and thus small values for k_B .

From our particle density extension, we find that with a $1\mu\text{m}$ sample, we are able to recover results within 1σ to what we expect for k_B for all three particle dilutions. There is no recognizable trend that we can identify regarding how D or k_B scale with particle density in our sample.

From our frame rate extension, we find that the effects of the frame rate also depend on the dilution of the sample. For the 15 : 1 dilution, decreasing the frame rate to 5 pushed our result further from the expected value, and increasing the frame rate to 15 significantly decreased our measured D and caused our k_B value to be $> 10\sigma$ from the expected value. However, varying the frame rate in the 30 : 1 dilution did not have similarly significant results: we were able to recover k_B to within 0.1 % for each frame rate. This may be because the **TrackPy** tracking finds it harder to associate particles in one frame with the same particle in the neighboring frames when there are twice as many particles present in the sample; the 30 : 1 dilution makes it easier to identify and track unique particles, even when greater or fewer frames are captured per second.

From our track lengths extension, we find that our best results come from the middling track lengths that we chose for our initial results. For the $1\mu\text{m}$ sample, shortening the track length did not change our results much, but lengthening it to 50 (where only about 5 particles were being tracked in each trial) nearly tripled our uncertainty, and caused our measured D and k_B to be much lower than expected. For the $2.12\mu\text{m}$ sample, the percent error for track lengths 5 through 50 is not very high, but the uncertainties are likely being underestimated; lowering the track length potentially includes spurious tracks, and increasing it causes our measurement to be based on fewer "trials" (as each particle tracked can count as a "trial" of measuring the mean-squared displacement of a particle under Brownian motion). For the $2.12\mu\text{m}$ sample, a track length of 50 still retained about 52 particles for each trial; the track length of 115 yields the result for k_B that is furthest from expected, as each trial was only retaining the tracking of about 5 particles.

One possible source of error could have come from our sample being exposed to the light source for a long period of time (e.g. 30 minutes). It should be noted that we assumed the temperature of the fluid to be constant when we calculated both the diffusion and Boltzmann constant. However, might not be the case when we did the lab since the light source within the apparatus could

have heated up our samples. We would expect the change in temperature to be negligible since water requires quite a lot of energy to heat it up. However, we note that the high intensity of our light source along with the time that the sample is exposed to it (up to 30 minutes at the most) could result in a non-negligible change in temperature of our sample – especially with the sample being a small amount of water (around $10\mu\text{L}$). Thus, because both the diffusion and Boltzmann constant depends on temperature, a change in temperature would result in the value of the two constants to change as well.

We should also note that we had to estimate what the temperature of each sample would be. As noted in the Data Analysis section, we took each sample to be the temperature of the lab room, which we also had to estimate as well. Thus, us not knowing the exact temperature of the sample to begin with could have contributed to a noticeable error with our calculation of the two constants.

Another possible source of error could be from the the particles undergoing non-random drift. In some runs, we found this non-random drift to be heavily present (e.g. the trajectories of all particles in one run were a straight line down). One possible cause of this drift could be a non-level microscope surface; though the particles are suspended in liquid and very small, we can't rule out the possible effect of gravity on the system. Another possible source for this non-random drift could be from the sample leaking from the rectangular hole. We think that when leakage is present, the particles near the edges of the rectangular hole would want to move with the leakage – that is, the particles would be subject to a non-Brownian force to move in a particular direction.

This is where we note that we manually made the rectangular hole for each sample. The size of the hole therefore varied while the amount that we put in stayed about the same (again, around $10\mu\text{L}$), and this could result in some leakage to occur in some samples if the hole was made too small.

Furthermore, if the hole is too large, then there would not be enough of the sample to fill the hole; this resulted in air bubbles to be present in the sample. This could be another possible source for the non-random drift. In some cases, the air bubbles were much larger than the particles, and the movement of these air bubbles could displace them such that their trajectories are no longer random.

Finally, one other possible source of error would be from how we used **TrackPy**. To expand on this, we had to play around with a lot of parameters to make sure that **TrackPy** was tracking actual particles (e.g. **TrackPy** might mistake a darker region to be a particle) and that the trajectories for those particles were not too short (e.g. we wanted to omit trajectories that only lasted a couple frames since it could result skew our data). We note that changing parameters would almost certainly change the values that we would get for the diffusion and Boltzmann constant; most of the time, these changes would be small,

but there were instances where the change was drastic. Thus, the values that we calculated from **TrackPy** might not be the most accurate that they could be. However, the more comfortable we got with using **TrackPy** (i.e. understanding what each of the parameters did), we feel that we mostly utilized **TrackPy** correctly and find that the error from our use of **TrackPy** is likely not a significant source of error.

VI. CONCLUSION

In this lab, we were generally successful in experimentally measuring the Boltzmann constant k_B by observing and analyzing the Brownian motion of small polystyrene spheres in water. The majority of our results were within 2σ of the established and expected value of k_B . We found that, generally, our results were closest to the expected value when using a $1\mu\text{m}$ sample at any dilution, with a frame rate of 10 and a track length of 15 or shorter. This finding is due to the limitations of our apparatus and software, which were best attuned to capture and track spheres of that size.

One possible improvement would be to use capillary tubes for all samples. This would almost certainly remove our issues with leakages or air bubbles, which in turn would mitigate any non-random drift.

Another improvement would be taking exact temperatures of the samples prior to capturing our data, so that we could take into account possible shifts in the base sample temperature over the several days of the experiment. Indeed it would be even better to quantify the possible temperature change due to the light source that

may occur while collecting data; this may be done by taking the temperature of the sample after taking our measurements, but this would be made difficult by the small amount of sample on each slide.

Our experimental results could also be improved with some improvements to **TrackPy**, including automating the ideal particle size based on the .tiff images we supplied. We used **ImageJ** to make an educated guess of what particle size we should tell **TrackPy** to track, but some adjustments needed to be made to procure good results in some cases. We largely based our **TrackPy** settings on by-eye evaluations of the plots shown in Figures 3, 4 and 5, but accuracy could be improved by the inclusion of some automated evaluations of these fits in the **TrackPy** program.

We believe that extensions of the lab to allow for more data collection would also help improve our results, especially for the particle sizes for which our results were far from the expected value (namely $0.52\mu\text{m}$ and $2.86\mu\text{m}$). Though our apparatus and analysis software was obviously not ideal to obtain results from these samples, we believe that better results could have been obtained by attempting many different setups—essentially, by performing all of our extensions on all of our samples, instead of just a few. We may have been able to find different dilutions, frame rates, or track lengths which yielded better results for these samples, given more time.

VII. FIGURE REFERENCES

Figure 1 can be found in the Brownian motion handout on our Canvas page.

-
- [1] Angeliki Ermogenous. Brownian motion and its applications in the stock market. *Undergraduate Mathematics Day: Proceedings and Other Materials*, 2006. https://ecommons.udayton.edu/mth_epumd/15/.
 - [2] Jason Holden and Kevin Kelly. Nanotechnology: Content and context: 2.2 brownian motion. <https://cnx.org/contents/ozbioN5l01.4:9ZteFgZF03/Brownian-Motion>.
 - [3] Dongdong Jia, Jonathan Hamilton, Lenu M. Zaman, and Anura Goonewardene. The time, size, viscosity, and temperature dependence of the brownian motion of polystyrene microspheres. *American Journal of Physics*, 75(2):111–115, 2007. <https://doi.org/10.1119/1.2386163>.
 - [4] Trackpy: Walkthrough. <http://soft-matter.github.io/trackpy/v0.5.0/tutorial/walkthrough.html>.