

# Lab 2 Session 1

## Pre Lab Question 1 - 5

Q1)

microscope calibration factor is 100 nm / px, what dilution of beads needed to desired density in the viewing area? The density of polystyrene is approx. 1.05 g/ml. The depth of field is 5 micro m. Beads are 1 micro meter. Let concentration be 0.5 wt %

$$\Rightarrow s = 100 \text{ nm / px} \quad (\text{calibration})$$

Camera field of view: 1288x964 pixels (Given)

↪ Field Dimensions: in  $\mu\text{m}$ ;  $1288 \times 100 \text{ nm} = 128.8 \mu\text{m}$

$$y; 964 \times 100 \text{ nm} = 96.4 \mu\text{m}$$

$$\begin{aligned} \text{Volume of field is: } V &= x \times y \times z = 128.8 \times 96.4 \times 5 \mu\text{m}^3 \\ &= 6.21 \times 10^4 \mu\text{m}^3 \\ &= 6.21 \times 10^4 \times 10^{-12} \text{ mL} \\ &= 6.21 \times 10^{-8} \text{ mL} \end{aligned}$$

$$\text{concentration for 20 beads, } C_{\text{needed}} = \frac{n}{V} = \frac{20}{6.21 \times 10^{-8}} = 3.22 \times 10^8 \text{ beads/mL}$$

$\Rightarrow$  Stock (0.5% wt) to beads/mL

$$\hookrightarrow 0.5 \text{ wt\%} \approx 0.005 \text{ g beads/mL}$$

$$\text{Mass per bead: } \rho = \frac{m_b}{V_b}; m_b = \rho V_b$$

$$\rightarrow V_b = \frac{4}{3} \pi r^3 = \frac{4}{3} \pi \times (0.5 \mu\text{m})^3 = \frac{4}{3} \pi (5 \times 10^{-5} \text{ cm})^3 \\ = 5.24 \times 10^{-13} \text{ cm}^3 //$$

$$\hookrightarrow C_{\text{stock}} = \frac{0.005}{5.24 \times 10^{-13}} = 9.1 \times 10^9 \text{ beads/mL}$$

$$\text{Dilution} = \frac{C_{\text{stock}}}{C_{\text{needed}}} \approx \frac{9.1 \times 10^9}{3.22 \times 10^8} \approx 28$$

ii

Q2)

root mean square speed  $\langle v^2 \rangle^{1/2}$  for a hydrogen molecule of mass  $3.3 \times 10^{-27} \text{ Kg}$  and for an E. coli bacterium of mass  $\approx 10^{-15} \text{ Kg}$ , at  $T = 300 \text{ K}$ .

Hydrogen

$$\Rightarrow \langle v_{\text{RMS}}^2 \rangle^{1/2} = \sqrt{\frac{3k_b T}{m}} \xrightarrow{\text{Boltzmann}} \sqrt{\frac{1.38 \times 10^{-23} \times 300}{3.3 \times 10^{-27}}} \approx 1.94 \times 10^3 \text{ m/s}$$

For E. coli,

$$v_{\text{RMS}} \approx \sqrt{\frac{3(1.38 \times 10^{-23})(300)}{10^{-15}}} \approx 3.52 \times 10^{-3} \text{ m/s} \approx 3.5 \text{ mm/s}$$

Q3)

All 4-step sequences ( $H = +1, T = -1$ ). Possible net displacements  $r_x$

\* +4: HHHH (1 way)

+2: 3H1T (4 ways)

0: 2H2T (6 ways)

-2: 1H3T (4 ways)

-4: TTTT (1 way)

$$\langle r_x^2 \rangle = \frac{1 \times 16 + 4 \times 4 + 6 \times 0 + 4 \times 4 + 1 \times 16}{16} = \frac{64}{16} = 4$$

$\Rightarrow$  For ten flips, five independent

$$\hookrightarrow \langle r_x^2 \rangle = N b^2 = \underline{\underline{10}}$$

Q4)

Given. Step length per residue,  $b = 0.4 \text{ nm}$

Number of residues,  $N = 375$

$$\Rightarrow \langle r_{\text{ee}}^2 \rangle = N b^2 \Rightarrow \langle r_{\text{ee}}^2 \rangle^{1/2} = b \sqrt{N}$$

$$\hookrightarrow r_{\text{RMS}} = 0.4 \text{ nm} \sqrt{375}$$

$$\approx 0.4 \times 19.36 \approx 7.7 \text{ nm}$$

Q5)

Time for random movement of protein to travel a distance equal diameter of a bacterium of  
 $r = 2 \text{ micro m}$ ,  
diffusion constant,  $D = 10 \mu\text{m}^2/\text{s}$

$$\Rightarrow \langle r^2 \rangle = 6Dt$$

$$t = \frac{r^2}{6D} = \frac{4}{6 \times 10} = \frac{4}{60} \approx 0.0067 \text{ s}$$

for a typical RMS displacement of  $2 \mu\text{m}$  in 3D.

1

## Lab 2 Session 1: Mobility and Mortality (Exploring Microscope and Imaging)

Date: 3rd Feb 2026

Lab Partner: Nathan Unhru

Recorder: Ahilan Kumaresan

### Table of Contents

#### Session 1: Mobility and Mortality (Exploring Microscope and Imaging)

1. Goals
2. Apparatus
3. Variables
4. References
5. Setup
6. Detailed Procedure
7. Analysis
8. Conclusions

#### Post Lab:

9. Second Attempt on Particle Tracking, Trajectory Extraction, and Data Analysis
10. New Conclusions
11. Post Lab Analysis

## I. Goals

1. Learn operation of upright bright-field microscope and achieve proper Kohler illumination
2. Calibrate the 100x oil-immersion objective using stage micrometer ( $\pm 2\%$  uncertainty target)
3. Prepare diluted sample of 1  $\mu\text{m}$  polystyrene spheres with  $\sim 20$  beads in field of view
4. Capture 60-second video of Brownian motion at 30 fps (1800 frames minimum)
5. Extract particle trajectories and compute diffusion coefficient D
6. Compare measured D to Stokes-Einstein prediction within stated uncertainties

## II. Aparatus

Item	Description
Software	Vision Assistant, ImagesJ 1.54f
Immersion oil	$n = 1.518$
Stage micrometer	1 mm / 100 div (10 $\mu\text{m}/\text{div}$ )
Camera	FLIR BlackFly U3-13Y3M
Objective	10 $\times$ , 40 $\times$ & 100 $\times$ oil

### III. Variables

Variable Type	Variable	Description
Dependent	Time	Frame number $\times$ interval
Dependent	$x(t), y(t)$	Bead position
Control	$\Delta x, \Delta y$	Displacement Of Polystyrene
Control	Bead Size	Water
Control	Frame Rate	30 or 60 fps

### IV. References

#### Primary Lab Documents:

1. MM-LabScript-microscopy.pdf (PHYS 332 Microscopy and Cell Motility lab script)
2. "Background: General Aspects of Cell Motion" document
3. Protocol: Microscope Setup
4. Protocol: Acquiring Movies with Vision Assistant
5. Protocol: Tracking Particles (MTrack2)
6. Protocol: Making Sample Chambers.

#### Textbooks:

Hughes, I. and Hase, T. Measurements and their Uncertainties, Oxford (referenced for Ch. 2.9, 5—8)

#### Online Resources:

1. Nikon Microscopy U ([microscopyu.com](http://microscopyu.com)) — Kohler illumination tutorial
2. ImageJ documentation (for MTrack2 plugin)

3. Added a new folder on GitHub:

Home Folder: Lab2-Mobility-and-Motility

<https://github.com/Ahilan-Bucket/phys332W-sfu/tree/main/Lab2-Microscopy-and-Motility>

Marker  
Look Here ~

4

## V. Setup

Image 1: Important elements in this lab

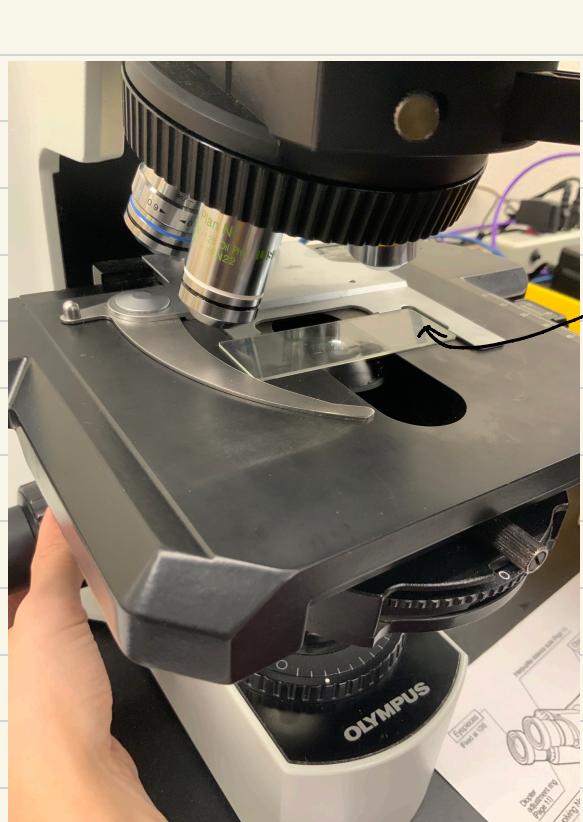


Image 2: Close up of Translation Stage

## VI. Detailed Procedure

### a) Kohler Illumination Setup Procedure:

(Following Protocol: Microscope Setup)

1. Turned on illumination lamp and allowed 5 min warm-up
2. Placed blank slide on stage, focused with 10 × objective
3. Closed field diaphragm until edges visible in FOV
4. Adjusted condenser height until field diaphragm edges sharp
5. Centered condenser using centering screws
6. Opened field diaphragm to just outside FOV
7. Adjusted aperture diaphragm to 70% of objective NA for optimal contrast

Note: Field diaphragm controls illumination area; aperture diaphragm controls NA (and thus resolution/contrast tradeoff).

## B. Camera Calibration with Stage Micrometer

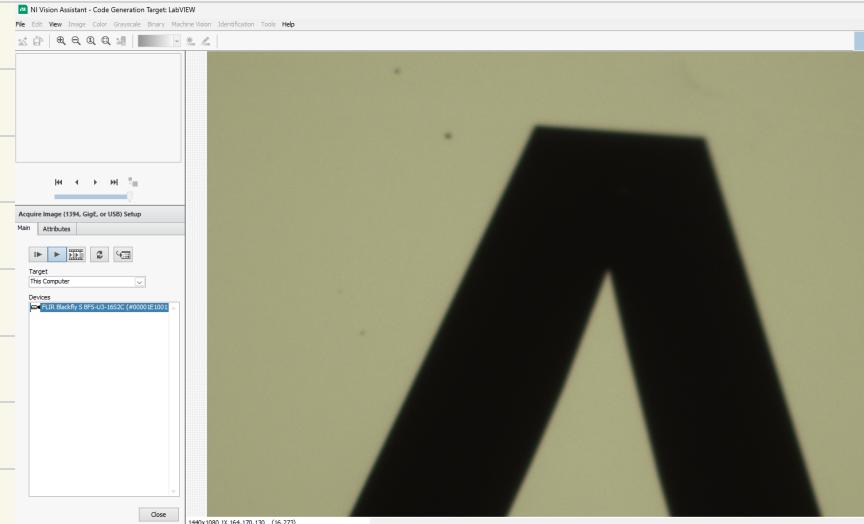
### Procedure:

- 2:30 PM
- Placed stage micrometer into the sample holder, translated a bit with xy knob
  - Focused and adjusted illumination lamp
  - Applied the given oil on the stage
  - Rotated 100x oil objective into position (parfocal - minimal refocus needed)
  - Launched Vision Assistant, LabVision > Image Acquire > Second Acquire.

Error! Our Lab Accusion is not detecting the Microscope, the TA mentions this is a recurring issue for this station. So we will document this troubleshooting.

Solution: Just restart and now it works. That was underwhelming, we were not able to understand why the error occurred.

Image3:  
Screenshot of  
our Working  
Window



Because it was hard to find the grating, we first focused on the letter refer image 3! And then were able to find the ruling

7

## 6. Captured image of micrometer scale bars

Calibration Data:

File: Home/Data/2026-02-03/imagecalibration.tif

## 7. Opened image in ImageJ for calibration measurement.

We selected 8 lines in 100x objective. Edge to Edge. This was the widest possible to reduce the uncertainty.

Physical image length per pixel calibration:  
8 lines has 1168.706 pixels

That is, 73.044 pixels per line

From calibration slide: 0.01mm per line

$$\text{Mm/px} = 6.845177 \times 10^5 \text{ mm} \times 10^3$$

$$= 0.06845177 \mu\text{m / px}$$

$$= 68.45177 \text{ nm/px}$$

Camera model BFS-U3-16S2C-CS

Calculated dilution factor (using prelab method)

20 beads per  $1440 \times 1080$  pixels  $\times 0.0685 \mu\text{m}$  per pixel

= 20 beads per  $98.6 \times 73.98 \mu\text{m} \times 5 \mu\text{m}$

Volume of image:  $36,482 \mu\text{m}^3$  vs  $77,760 \mu\text{m}^3 = 0.46916x$   
 (compared to prelab)  
 Density Prelab:  $2.572 \times 10^8 \text{ beads/ml}$

$$\text{No. of Beads} = \text{Vol} \times \text{Conc.}$$

$$20 \text{ beads in } V = 36,482 \mu\text{m}^3 = 3.65 \times 10^{-8} \text{ ml}$$

$$n = \frac{N}{V} = \frac{20}{3.65 \times 10^{-8} \text{ mL}} = 5.48 \times 10^8 \text{ beads/ml}$$

→ Required:  $5.48 \times 10^8 \text{ beads/ml}$

$$\text{Mass conc: } 5.50 \times 10^{13} \text{ g/bead} \times 5.48 \times 10^8 \text{ beads/} \\ \text{mL} = 3.01 \times 10^{-4} \text{ g/mL}$$

Stock: 0.005 g/mL

Initial: 0.005 g/mL

$$\text{Dilution} = \frac{\text{initial}}{\text{final}} = \frac{0.005}{3.01 \times 10^{-4}} = 16.6x \approx 1:17$$

## C. Sample Preparation

~~2:44 PM~~  
3.12 PM

### Materials:

Stock concentration: 0.5 wt% (5 mg/mL)

Water

Glass microscope slides (Fisher Scientific)

1 coverslips (22x22 mm, 0.13-0.17 mm thick)

Parafilm spacers (100 µm thick)

### Dilution procedure:

Based on pre-lab calculation, target dilution ~1:30

1. Vortexed stock solution for 30 seconds to disperse aggregates

2. Pipetted 1 µL of stock into microcentrifuge tube

3. Added 30 µL of water (1:30 dilution)

4. Gently mixed by pipetting up and down 10 times, just to get things more uniform

Make sure you wear gloves and not touch the outer side.

#### D. Chamber preparation:

(Following Protocol: Making Sample Chambers - thick chamber method)

1. Cut two strips of Parafilm (~3 mm wide, ~20 mm long)
2. Placed strips parallel on glass slide, ~15 mm apart
4. Pipetted 10  $\mu\text{L}$  diluted bead solution between Parafilm strips
5. Gently lowered coverslip onto Parafilm strips
6. Sealed edges with nail polish (quick-dry, 2 min)

#### Quick check:

Initial observation with 100x objective showed good bead distribution

Estimated 25-30 beads in FOV - slightly more than target but acceptable

No obvious aggregates or debris visible

Chamber depth: 100  $\mu\text{m}$  (Parafilm thickness)

CONCLUSION: Sample prepared with 1:30 dilution.

Observed 25-30 beads in FOV at 100x. No aggregation visible.

## E. Data Collection - Brownian Motion Video

Environmental conditions:

Room temperature:  $22.3 \pm 0.5^\circ\text{C}$

Acquisition procedure:

1. Switched to 100x oil objective, applied immersion oil
2. Focused on mid-plane of chamber
3. Adjusted illumination for good contrast (beads appear as dark circles)
4. Started Vision Assistant recording
5. Recorded for 120 frames

*I'm not too  
sure...*

Saving two versions of the files to explore how frame rate affects us ( 5 frame and 1 frame Rate )

Save: home/Data/2026-02-03/1um-1frame.avi  
home/Data/2026-02-03/1um-5frame.avi

Errors notes:

**Issue 1:** First recording showed drift (all beads moving in same direction)

Cause: Coverslip not fully sealed, causing flow

**Solution:** Applied more nail polish to edges, waited 5 min

Issue 2: Some beads going in/out of focus

This is expected - chamber is much deeper, even labscript tells us this

Solution: Track only beads that stay in focus throughout recording

CONCLUSION: Successfully recorded 60 s video at 1 fps (60 frames).

## F. Particle Tracking and Trajectory Extraction

Method: ImageJ MTrack2 Plugin  
(Following Protocol: Tracking Particles)

Procedure:

1. Opened AVI file in ImageJ (File > Import > AVI)
3. Applied background subtraction (rolling ball radius = 50 pixels)
4. Ran Plugins > MTrack2
5. Exported trajectories as CSV

Save: home/Data/2026-02-03/1um-1frame.avi  
home/Data/2026-02-03/1um-5frame.avi

Results:

Successfully tracked beads for 120 frames

## VII . Analysis

GOAL: With the Movie in Place, we now collect sufficient data next try make histogram of the beads displacement.

Explored Protocol: Tracking Particles to do this.  
Please refer for Instructions.

We try to alter the "Minimum Object Size" and "Minimum track length," and after MTrack2 is run, use MTrack2Loader.ipynb on our Local VSCode and Online Szygy.

Changing the second code block to locate our file.

I am getting an error, seems like our file delimiter is : ";"

After another Attempt, there are more errors.

This could be most like because:

1. Examining our result file, and using Gemini AI, it was clear that this file has multiple chunks of tracks. I do not think this current code works for this.
2. Also, we also get a AttributeError, "deprecated Warning." This was because of using np.int instead of int.

The fixed code can be found in  
Home\Analysis\MTrack2Loader-fixed.ipynb

Intuition: My expectation for the plots is that both the 1D Particle Tracks and 2D tracks would be random motion, showing the Brownian motion of the particles.

figure 1: 1D Plot of 1 micrometer Beads.

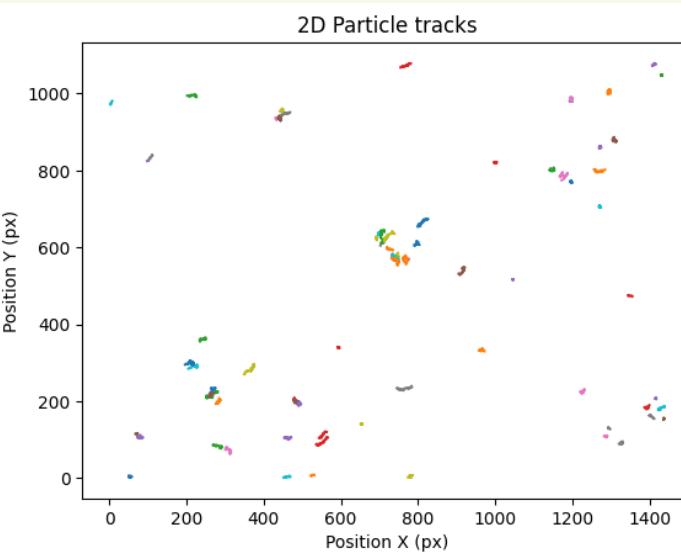
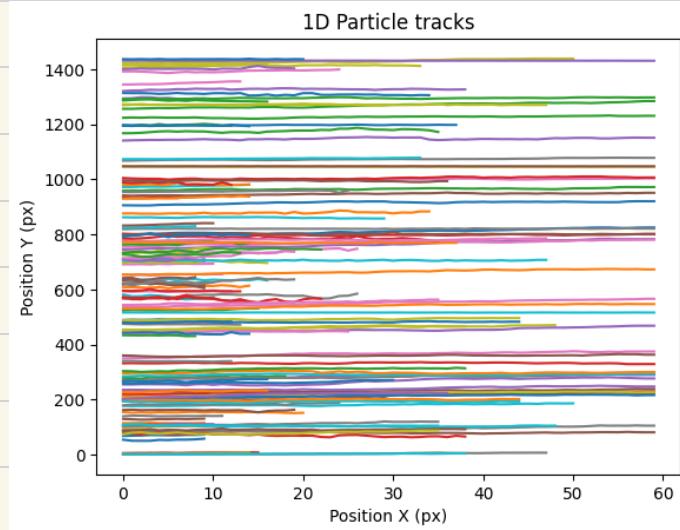


figure 2: 2D Plot of 1 micrometer Beads.

### Observation:

1. There are over 20 particles, way over
2. The Flow on 1D is not random, there clearly is a one directional flowing
3. In 2D there are a lot of small local motion.  
And it is not clear whether they are brownian or just linear.

## Conclusion:

On the Bright side, we have sucessfully created the pipeline to convert the Data from the Accusation to the Plots.

### Errors

1. Our Concentration might not be right, There looks like we have way more than 20 beads
2. The MTrack2 Plug in parameters must be be chnaged. Minimum Object size, Clealry even the Plugin is not filtering the number of beads
3. The Parameter "Minimum track length" is also not working, must adapt this as there are very shortlived beads here.
4. The Sampling duration also might not be long enough. Not a lot of long term trends visible

### Reason for error.

What is happening is, during the Lab we collected the results.txt from ImageJ. Due to time constrainsts, we could not fully explore the parameters. So, we just took any data we could.

## Post Lab

Ahilan  
DK

## IX. Second Attempt on Particle Tracking, Trajectory Extraction, and Data Analysis

Goal: 1. Download Image J on Local Desktop and To explore parameters on the MTrack2 plugin.

2. Create a better TrackResult that tracks fewer long-lived particles

Refer to the previous sections F and G for the method and references.

Downloaded ImageJ from : <https://imagej.net/ij/download.html>

Protocol: Tracking Particle says to use this link to download the Plugin MTrack2

[http://valelab.ucsf.edu/~nico/  
IJplugins/MTrack2.html](http://valelab.ucsf.edu/~nico/IJplugins/MTrack2.html)

Issues: However, this is not working. Alternative links do not work too.

<http://valelab.ucsf.edu/~nico/IJplugins/MTrack2.html>  
On further research I found out that Fiji is a similar app to ImageJ, based on the documentation, it seems like it is highly likely to contain some version of MTrack2 plugin.

Downloaded Fiji: <https://imagej.net/software/fiji/>

I was right!, MTrack2 exists, but under the folder Trackers.

I am going to copy this MTrack2 -2.0.1.jar into ImageJ

Success: On Image J, I am able to open this plug-in but its now. Plugins/ Tracking/MTrack 2.

Continue to Follow instructions from protocol:  
 Tracking Particles to open the .avi file. and  
 Threshold

Error: "AVI File Error: 'ÿþ' @ 0x10f6c2c0 has invalid length. File damaged/  
 truncated?"

Ignoring the error

Was able to use  
 ImageJ and the  
 MTracker Plugin at  
 Home! Yay! YY

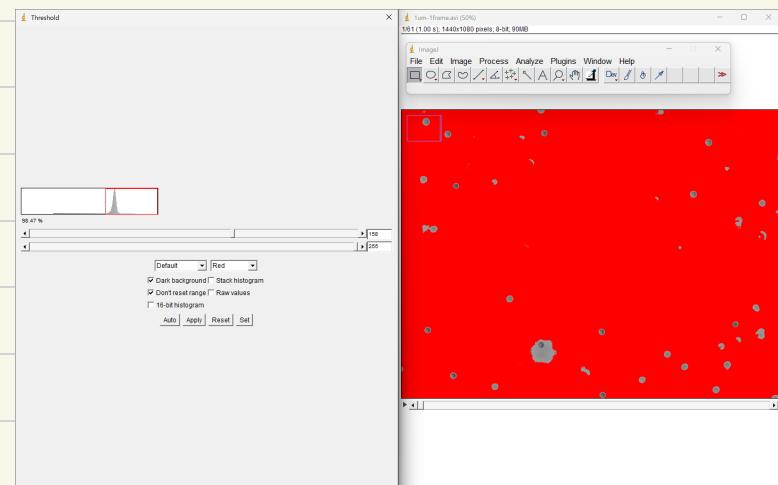


Image 4: Screenshot of ImageJ during Threshold Process

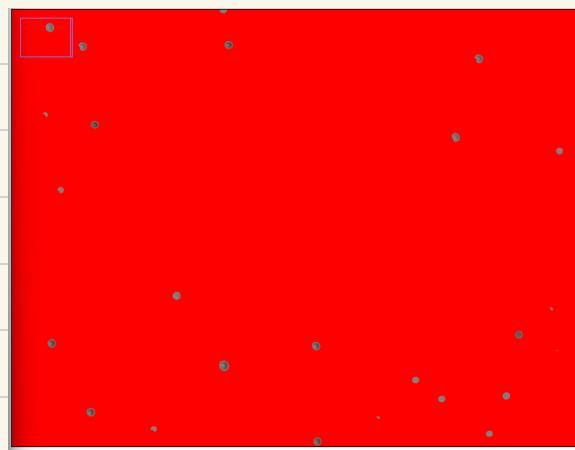


Image 5: Image at threshold values to mask particles

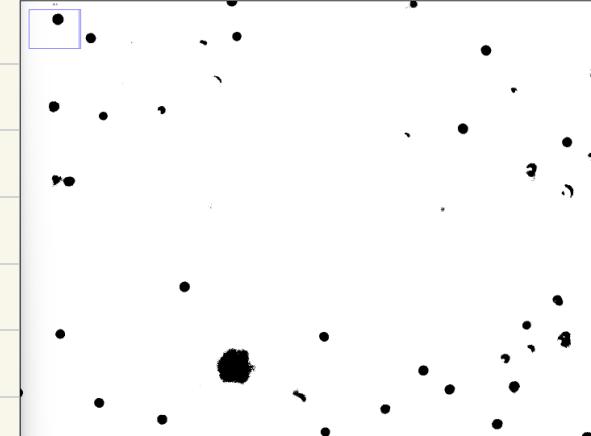


Image 6: Post Conversion into Binary

We had uniform Background, finding threshold was easy.

Threshold Value for masking particles  
 Upper Dial - 138  
 Lower Dial - 255

## Running MTrack2 plugin

Determining Parameters more carefully now.

With the measure tool, i can see than usually particles span: (107-79), (97-72)  
thats, 28 px and 25 px

Let me give this a tolerance value of approx. 7px

So I believe from 23 to 37 would be the range for pixels.

Maximum Velocity, i do not know, I am going to set that to a very large number

Minimum Track Length, our data is 60 frames, so maybe 10 Frames

The above settings were too tight, Gave no Particles.

Gives 12 Particles. Let me try to make it just a bit tighter

Despite Multiple trials, It either oscillates to 0 or 12.

Observation: The Minimum Track Length (frames) seemsnt to be the most inflencial right now.

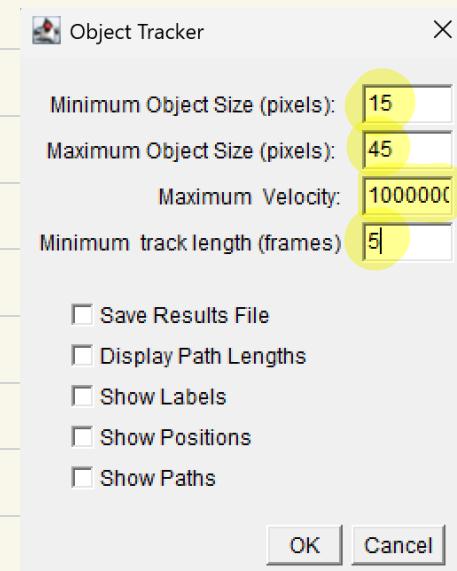


Image 7: Initial Parameters

Saving the track file as: "lum-post-lab-trackresults.txt"

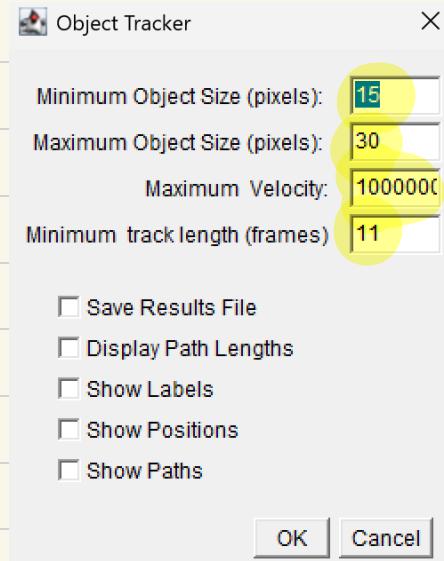


Image 8:  
Parameters  
used to save

I am further experimenting different parameter Combinations and see which one gives the best data

Min Size	Max Size	Max Vel	Min Track	Tracks Found
5px	30px	100px	11 frames	1 particle
10px	30px	100px	11 frames	2 particles
15px	3px	100px	12 frames	ZERO
15px	30px	100px	13 frames	ZERO

*Saved*  
Another File:  
min5px minTrack-  
11-trackResults.txt

Table 1: Summary of explored Parameters and results

**Observation:** I can notice that the most sensitive parameter is the Minimum Track time, followed by the Minimum Size. I say this because a small variation gives dramatic different results

X. New conducting Analysis now  
plotting using the notebook

Sucess:

Figure 3: 2D plot of 1 micrometer bead done post lab

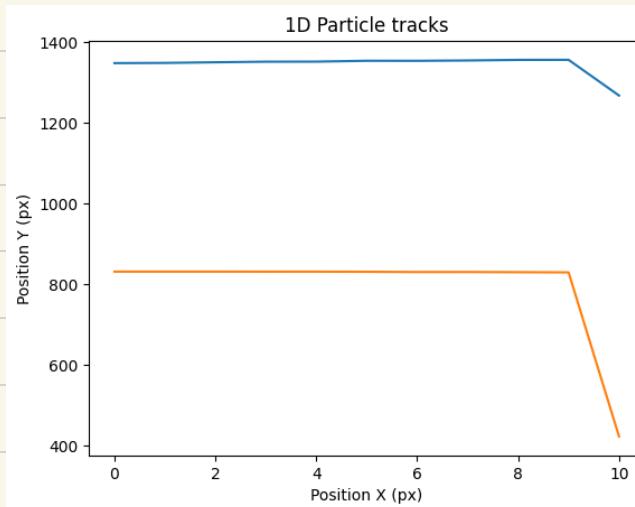
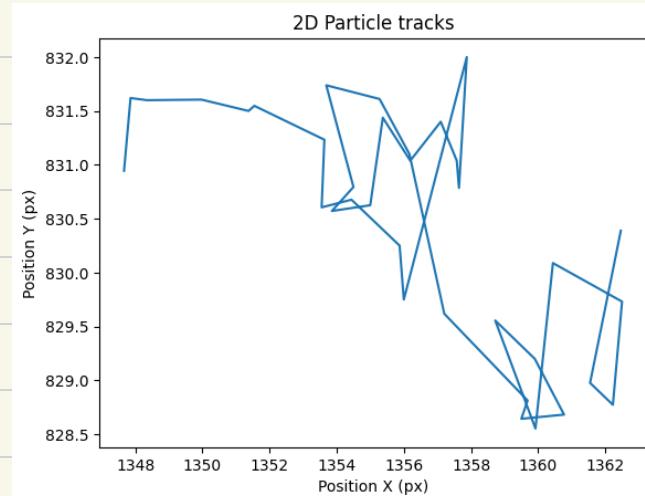


Figure 4: 1D plot of 1 Micrometer bead done post Lab

Success: I think I like the parameters I did post lab better. The 2 D plot especially looks like there is clear Brownian Motion.

The 1D plot, to me just says there is a linear flow. I need to find ways to track more beads tho. And image for a longer time.

## XI. Post lab Analysis

\* (For next lab)

Firstly, the Data seems fishy -

I want to recheck the Distillation. Maybe there is an error

We need to do the Calibration error, for that, we must take multiple Data sets of the edge to edge pixel values

I need to verify on which is the true Value of the Camera Pixel Size. The one on Labscript or the one we found online for the Camera we use?

If anything is clear it is that I need more Data. For now let me collect 3 Minutes of Data at least. Just a Heuristic based on how slow these things move.

With this re run data, we run through our Analysis. And then see what our outputs are.

I want to ensure we have a great workflow going on for sample preparation (Dilution, Chamber, Camera). Once a Solid Workflow is set up.

I want to move to the analysis of Larger Beads, Run through the script, debug and then compare the Diffusion Coefficients.

## Lab 2 Session 2: Mobility and Mortality (Exploring Larger Particles and refining workflow)

Date: 5th Feb 2026

Lab Partner: Nathan Unhru

Recordee: Ahilan Kumaresan  
(First 1 Hour)

### Table of Contents

Session 2: Mobility and Mortality (Exploring Larger Particles and refining workflow)

1. Goals
2. Apparatus
3. Variables
4. Procedure (Recordee: Ahilan)
5. Procedure (Recordee: Nathan)
6. Recalibration
7. Calibration Uncertainty
8. Investigating Small and large Particles
9. Diffusion Analysis Code

### Post lab

10. Discussions, reflections and Suggestions

## I. Goals

1. Redo Dillution
2. Take multiple calibration values and calibrate.
3. Verify true Camera Pixel Size
4. Collect more frames and for longer duration.
5. Do Analysis and compare a larger size particle with smaller size ones, qualitatively.
6. Learn a better workflow for this lab from crating sampling and tracking them.
7. Explore Static Live Cells

## 2. APPARATUS

Refer Pg. 2 section II

Additionally

1. Objectives available: 10x, 40x, 100x oil
2. Ethanol and lens paper

Software:

1. ImageJ / Fiji with MTrack2 plugin (for particle tracking)
2. Python (MSD and plotting; Analysis code made by Ahilan, using demo code from Python Demos on Canvas)

## 2. APPARATUS

Samples:

1. 1 μm and 5 μm beads
  2. Glass slides, 1 coverslips, Parafilm (100 μm) or double-sided tape
- (note which objectives available: 10x, 40x, 100x oil (note which s))

## 3. VARIABLES

Refer Pg 2, Section II

— Yeast objective: \_

— HT1080 objective: \_

—

— x

— Bacteria objective: \_

—

— x

Camera: FLIR BlackFly (CMOS, 1/3" diagonal, 1288

ript

. 2.1

Illumination: Bright-field with Köhler illumination (act

## IV. PROCEDURE (Recorder: Ahilan)

Do Initial Check up of Equipment.

## DOING THE PROTOCOL: MICROSCOPE SETUP — OLYMPUS BX51

I verified with the TA that the Labscript is inaccurate about the Field of View for the Camera, From now on.  $1440 \times 1080$  px is the FOV.

I initially thought it would be a good idea to just focus the Calibration slide on  $10x$ , then  $20x$  and then move to  $100x$  with oil.

However, I learnt watching the TA focus, focus progressively from  $10x$  to  $100x$  is way quicker than moving to  $100x$  and not being able to find where the sample is on  $100x$ .

I learnt to clean the Calibration Slide using Ethanol to remove oil. Must use Lens Paper and Dab around.

I learnt from our Professor that the Lens liquid is actually having lot of additives. This made me very conscious about the liquid the entire lab. I avoided direct contact.

Redid Small Beads/ Used .200 ml Solution of 1 μm at 0.5 concentration

Used the Factor. 1 sol : 17 Water.

So collected 200 μm solution and 3400 μm water.

7X

Also simultaneously made the Larger Size Particle

5  $\mu\text{m}$  diameter

$$\begin{aligned} \text{Vol} &= 125 \times 1 \mu\text{m diameter} \\ &= 125 * 5.4978 \times 10^{-13} \text{ g/bead} = 6.8725 \times 10^{-11} \text{ g/bead} \end{aligned}$$

$$\begin{aligned} \text{g/Bead} \times \text{bead/mL} &= 6.8725 \times 10^{-11} \times 5.4894 \times 10^8 \\ &= 0.0377245 \text{ g/mL} \end{aligned}$$

Original: 0.005

Original/new:  $7.5449 \times$  dilution

With factor 1: 6.5

We used 100  $\mu\text{m}$  of solution and 650  $\mu\text{m}$  water

Nathan Arrives, from now on Note Recordee Nathan



## V. Procedure (Recordee: Nathan )

Because our last data from session 1 was analyzed and produced poor results, we first wanted to recalibrate the microscope and retake this data with what we learned last class.

The camera was having technical issues, including stating an error codes or disconnecting after a certain time. This was fixed with help from the TA

28

Error: Again, the camera failed. Same Issue we faced last lab.

Solution: Go to Settings, then Camea, then remove this device.

Unplug the Camera and use a different port.

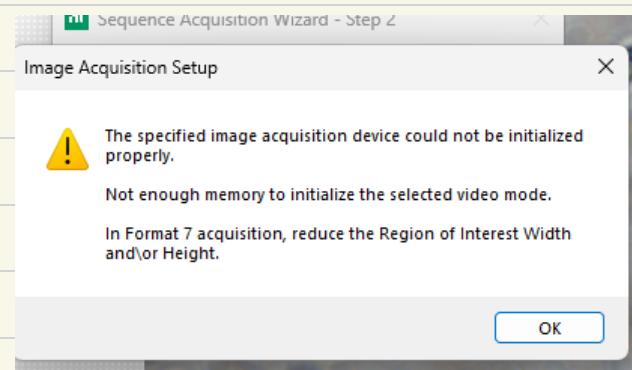
We produced a sample with a 16.57X dilution factor of the original 0.5%w 1um bead solution. This should get us 20 beads per image based on our prelab calculation which we corrected for our true microscope image size.

When placing in the microscope, we noticed the beads count increased rather than decreased. We collected an image of the beads and then tried to correct this.

Error: Screenshot

When we try to record 18000 Frames

Image 1: Error Message



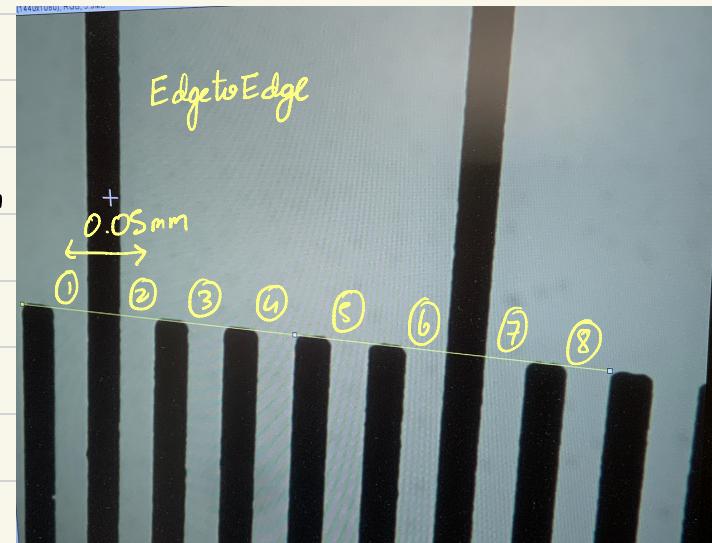
Apparenlty we can only record about 120 Frames.

Took two data values, 5mu-0\_5p.avi  
initial-1mu-0\_5p.avi

## VI. ReCalibration

Objective 100x  
8 lines  
Calibration slide, 1 div = 0.05mm

Figure 1: Calibration lines from the calibration slide, 100X magnification



$$\text{For Calibration: } \frac{8 \text{ lines} \times 0.05}{1159.3 \text{ px}} \rightarrow \text{Physical length} \\ \approx 3.45 \times 10^{-4} \text{ mm/px}$$

$$1 \text{ px} \approx 0.345 \mu\text{m}$$

## VII. Calibration Uncertainty

Measurement no.	Left Pixel (x,y)	Right Pixel (x,y)	Distance = $\sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$
1	(52, 550)	(1208, 637)	1159.27
2	(52, 554)	(1206, 632)	1156.63
3	(50, 552)	(1208, 636)	1161.05
4	(50, 552)	(1210, 633)	1162.83
5	(70, 549)	(1208, 633)	1141.10

Table 1: Values for multiple Calibration Data

Mean Pixel length:  $\bar{L} = \frac{1}{N} \sum_{i=1}^N L_i$

$$= \frac{1159.27 + 1156.63 + 1161.05 + 1162.83 + 1141.10}{5}$$

$$\bar{L} = \underline{\underline{1156.98 \text{ px}}}$$

Sample Standard Deviation

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N-1} (L_i - \bar{L})^2}$$

$$= \sqrt{\frac{(2.29)^2 + (-0.35)^2 + (4.07)^2 + (5.85)^2 + (-15.88)^2}{4}}$$

$$\sigma = \underline{\underline{8.40 \text{ px}}}$$

Standard error of mean:  $\sigma_{\bar{L}} = \frac{\sigma}{\sqrt{N}} = \underline{\underline{3.76 \text{ px}}}$

## VIII. Investigating small and large particles

Hi, From here, Ahilan is editing the Notebook.

We make samples of 5  $\mu\text{m}$  and 1  $\mu\text{m}$  particles. Following the Same procedure as before, we make the sample, record the movie on Image J. (Refer to pg 9 - 13 of Lab Notebook)

We noticed we had to dilute the 1  $\mu\text{M}$  solutions such that we use, 1 part solution and 224 parts water. 1: 224. To get 10-20 particles in our FOV.

This is approximately 17.5x off our calculation.

Sources of error: 1. Either that the sample was not prepared correctly.  
2. Students drew from the sample liquid surface without mixing it, which would increase the concentration.

File Saved as: 5mu-21C-1isto224w-0\_5p.avi

Data folder: Data/06-02-2026

1:6.5 dilution of 5-micrometer particle of 0.5 %, which were confirmed by previous calculations.

Saved as : 5mu-21C-1isto6\_5w-0.5p.avi

We could then collect good quality data with these new samples prepared and injected into a glass slide. See our data results below:

We tracked the data using the software and found better results by tracking the darker black parts of the small beads rather than the bright white centers, as some beads didn't have centers in our plane of focus for the 1 micrometer beads. The 5 micrometer beads were much easier to focus and get good results.

Found those Calibration parameters the Best

Figure 2: Parameters for Tracking 1 micrometer particles

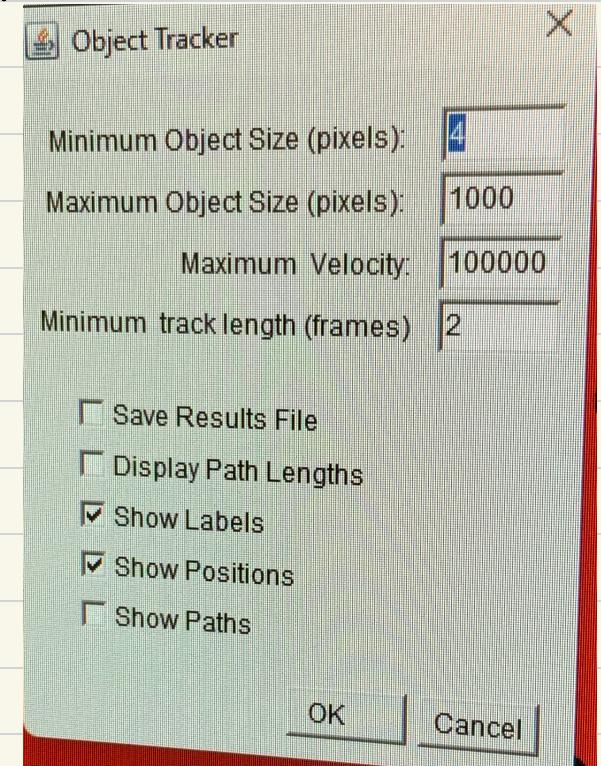
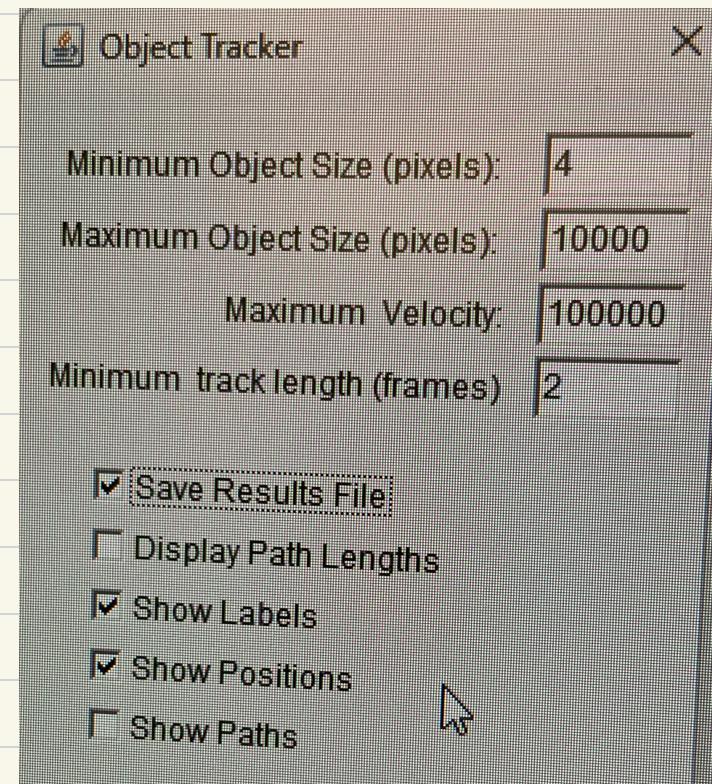


Figure 3: Tracking parameters for 5 micrometer particles



## 9. Diffusion\_Analysis Code

February 6, 2026

### 1 Brownian Motion Diffusion Analysis

Ahilan Kumaresan

This code has been Built with ideas inspired by the Reference code the Prof has in Canvas, along with using an LLM.

*Analysis of multiple bead sizes for diffusion coefficient comparison*

This notebook performs complete diffusion analysis including: 1. Loading MTrack2 data for multiple bead sizes 2. Track splitting at large jumps (preserves more data) 3. Displacement histogram and Gaussian fitting 4. Direct calculation of diffusion coefficient from variance 5. Mean-Squared Displacement (MSD) analysis 6. Comparison with Stokes-Einstein theory 7. **Comparison between bead sizes**

```
[ ]: # Import required libraries
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.cm as cm
from scipy.optimize import curve_fit
from numpy import pi, exp, sqrt
import warnings
warnings.filterwarnings("ignore")

# =====
# CONTROL VARIABLES - MODIFY THESE BEFORE RUNNING
# =====

# Data file paths for each bead size
# Format: {bead_diameter_um: file_path}
DATA_FILES = {
    1.0: "../Data/2026-02-05/1mu-21c-1isto224w-0_5p-trackresults.txt", # 1μ
    # micron beads (75 particles)
    5.0: "../Data/2026-02-05/5mu-21c-1isto6_5w-0_5p-trackresults.txt", # 5μ
    # micron beads (17 particles)
}

# Number of best track segments to use (ranked by length)
NUM_BEST_SEGMENTS = 10 # <select top N track segments
```

```

# Minimum track segment length required (frames)
MIN_TRACK_LENGTH = 20 # <-- Segments must have at least this many frames

# Whether to subtract mean displacement (removes drift)
SUBTRACT_DRIFT = True

# Maximum allowed displacement per frame (pixels) - tracks SPLIT at larger jumps
# This splits tracks into valid segments rather than discarding entire tracks
MAX_DISPLACEMENT_PER_FRAME = 20 # pixels - tracks split at jumps larger than
    ↪ this

# === Experimental Parameters ===
FRAME_RATE = 226      # fps (frames per second)
PIXEL_SIZE = 0.345     # microns per pixel (345 nm)
TEMPERATURE = 294.15   # Kelvin (21C)
VISCOSITY = 0.0009764 # Pa.s (water at 21C)

# Physical constants
k_B = 1.381e-23 # Boltzmann constant (J/K)

# =====
print("CONTROL VARIABLES SET")
print(f" Data files: {len(DATA_FILES)} bead sizes")
for d, path in DATA_FILES.items():
    print(f" {d} um: {path.split('/')[-1]}")
print(f" Using top {NUM_BEST_SEGMENTS} track segments")
print(f" Min track length: {MIN_TRACK_LENGTH} frames")
print(f" Track split threshold: {MAX_DISPLACEMENT_PER_FRAME} px/frame")
print(f" Drift correction: {'ON' if SUBTRACT_DRIFT else 'OFF'}")
print(f"\nExperimental parameters:")
print(f" Frame rate: {FRAME_RATE} fps")
print(f" Pixel size: {PIXEL_SIZE*1000:.1f} nm/px")
print(f" Temperature: {TEMPERATURE} K ({TEMPERATURE-273.15:.1f} C)")

```

## CONTROL VARIABLES SET

Data files: 2 bead sizes  
 1.0 um: 1mu-21c-1isto224w-0\_5p-trackresults.txt  
 5.0 um: 5mu-21c-1isto6\_5w-0\_5p-trackresults.txt  
 Using top 10 track segments  
 Min track length: 20 frames  
 Track split threshold: 20 px/frame  
 Drift correction: ON

## Experimental parameters:

Frame rate: 226 fps  
 Pixel size: 345.0 nm/px  
 Temperature: 294.15 K (21.0 C)

## 1.1 Helper Functions

Functions for loading data, splitting tracks at jumps, and performing analysis.

**Source:** Data loading adapted from `LabLevel References/Reference-code/Python demos/MTrack2Loader.ipynb` (Rev11 250919 DAL, Physics SFU). Modified: fixed dtype bug (`np.int -> int`), added `invalid_raise=False`. Track splitting logic added to preserve more data by splitting at large jumps instead of discarding.

```
[1]: def load_mtrack2_data(file_path):
    """
    Load MTrack2 output file and return cleaned data matrix.

    Returns:
        data: numpy array with shape (frames, 2*particles) containing X, Y pairs
    """
    my_data = np.genfromtxt(file_path, delimiter='\t', skip_header=2,
                           skip_footer=1, invalid_raise=False)

    # Remove indexer, summary box and Flags column
    A = np.zeros(my_data.shape[1]//3+1, dtype=int)
    for i in range(my_data.shape[1]//3+1):
        A[i] = 3*i
    new_data = np.delete(my_data, A, axis=1)

    # Sort data - move NaN values to the end
    mask = np.isnan(new_data)
    new_mask = np.zeros(mask.shape)
    for ind, value in enumerate(mask):
        new_mask[ind, :] = ~value * (ind + 1)
    new_mask = new_mask.astype(np.int_)

    for row_index, row in enumerate(new_mask):
        for col_index, item in enumerate(row):
            if item == 0:
                new_mask[row_index][col_index] = (new_mask.shape[0] + 5)

    for i in range(new_mask.shape[1]):
        new_mask[:, i] = np.sort(new_mask[:, i])

    data = np.empty((mask.shape[0], mask.shape[1]))
    data[:, :] = np.nan

    for i in range(new_mask.shape[0]):
        for j in range(new_mask.shape[1]):
            temp = new_mask[i, j]
            if temp < (new_mask.shape[0]):
                data[i, j] = new_data[temp, j]
```

```

    return data

def split_tracks_at_jumps(data, min_length, max_jump_px):
    """
    Extract valid trajectory segments from raw data.
    SPLITS tracks at large jumps (tracking errors) instead of discarding entire
    ↴tracks.

    Parameters:
        data: Raw X,Y data (rows=frames, columns=X1,Y1,X2,Y2,...)
        min_length: Minimum number of points for a valid segment
        max_jump_px: Maximum allowed jump per frame (pixels) - tracks split here

    Returns:
        List of trajectory dicts with 'x', 'y', 'id' keys
    """
    segments = []
    n_particles = data.shape[1] // 2
    segment_id = 0

    for i in range(n_particles):
        x_col = i * 2
        y_col = i * 2 + 1

        # Extract track
        x_raw = data[:, x_col]
        y_raw = data[:, y_col]

        # Remove NaNs (get valid indices)
        valid_mask = ~np.isnan(x_raw) & ~np.isnan(y_raw)
        x_clean = x_raw[valid_mask]
        y_clean = y_raw[valid_mask]

        if len(x_clean) < min_length:
            continue

        # Calculate step sizes
        dx = np.diff(x_clean)
        dy = np.diff(y_clean)
        steps = np.sqrt(dx**2 + dy**2)

        # Find indices where jumps exceed threshold
        bad_jump_indices = np.where(steps > max_jump_px)[0]

        if len(bad_jump_indices) == 0:

```

41

```

# No bad jumps - use entire track as one segment
segment_id += 1
segments.append({
    'x': x_clean,
    'y': y_clean,
    'id': segment_id,
    'original_particle': i + 1,
    'length': len(x_clean)
})
else:
    # Split at bad jump locations
    split_indices = bad_jump_indices + 1
    x_segments = np.split(x_clean, split_indices)
    y_segments = np.split(y_clean, split_indices)

    for x_seg, y_seg in zip(x_segments, y_segments):
        if len(x_seg) >= min_length:
            segment_id += 1
            segments.append({
                'x': x_seg,
                'y': y_seg,
                'id': segment_id,
                'original_particle': i + 1,
                'length': len(x_seg)
            })

# Sort by length (longest first)
segments.sort(key=lambda s: s['length'], reverse=True)

return segments

def gaussian(x, amplitude, mean, std_dev):
    """Gaussian function for curve fitting."""
    return amplitude * np.exp(-(x - mean)**2 / (2 * std_dev**2))

def linear(t, slope, intercept):
    """Linear function for MSD fitting."""
    return slope * t + intercept

```

## 1.2 Main Analysis Function

This function performs the complete diffusion analysis for a single bead size.

```
[ ]: def analyze_bead_size(file_path, bead_diameter, label):
    """
```

Perform complete diffusion analysis for one bead size.

*Parameters:*

- file\_path: Path to MTrack2 results file*
- bead\_diameter: Bead diameter in micrometers*
- label: String label for plots*

*Returns:*

- Dictionary with all results*

```
"""
results = {'bead_diameter': bead_diameter, 'label': label}

# Derived parameters
dt = 1 / FRAME_RATE
r = (bead_diameter / 2) * 1e-6 # radius in meters

# Theoretical D (Stokes-Einstein)
D_theory = k_B * TEMPERATURE / (6 * np.pi * VISCOSITY * r)
D_theory_um = D_theory * 1e12 # Convert to um^2/s
results['D_theory'] = D_theory_um

# =====#
# LOAD DATA
# =====#
print("=" * 60)
print(f"ANALYZING {label}")
print("=" * 60)

try:
    data = load_mtrack2_data(file_path)
except Exception as e:
    print(f"ERROR loading file: {e}")
    print(f" File path: {file_path}")
    return None

total_frames = data.shape[0]
total_particles = data.shape[1] // 2

print(f"\nFile: {file_path.split('/')[-1]}")
print(f"Total frames: {total_frames}")
print(f"Total particles detected: {total_particles}")

# =====#
# SPLIT TRACKS AT LARGE JUMPS
# =====#
segments = split_tracks_at_jumps(data, MIN_TRACK_LENGTH,
MAX_DISPLACEMENT_PER_FRAME)
```

```

print(f"\nTrack splitting (threshold: {MAX_DISPLACEMENT_PER_FRAME} px/
frame):")
print(f"  Valid segments extracted: {len(segments)}")
print(f"  (Min segment length: {MIN_TRACK_LENGTH} frames)")

if len(segments) == 0:
    print("WARNING: No valid segments! Try adjusting parameters.")
    return None

# Select top segments
n_to_use = min(NUM_BEST_SEGMENTS, len(segments))
selected_segments = segments[:n_to_use]

print(f"\nUsing top {n_to_use} segments:")
print(f"  {'Seg':<5} {'Orig Part':<10} {'Frames':<8}")
print(f"  {'-'*25}")
for seg in selected_segments[:10]: # Show max 10
    print(f"  {seg['id']:<5} {seg['original_particle']:<10} {seg['length']:<8}")

results['n_segments'] = n_to_use
results['segments'] = selected_segments

# =====
# EXTRACT DISPLACEMENTS
# =====
all_dx = []
all_dy = []

for seg in selected_segments:
    dx_seg = np.diff(seg['x'])
    dy_seg = np.diff(seg['y'])
    all_dx.extend(dx_seg)
    all_dy.extend(dy_seg)

dx = np.array(all_dx)
dy = np.array(all_dy)

# Apply drift correction if enabled
if SUBTRACT_DRIFT and len(dx) > 0:
    mean_dx = np.mean(dx)
    mean_dy = np.mean(dy)
    dx = dx - mean_dx
    dy = dy - mean_dy
    print(f"\nDrift correction applied:")

```

```

        print(f" Mean dx removed: {mean_dx:.4f} px ({mean_dx*PIXEL_SIZE*1000:.
        ↪1f} nm)")
        print(f" Mean dy removed: {mean_dy:.4f} px ({mean_dy*PIXEL_SIZE*1000:.
        ↪1f} nm)")

print(f"\nTotal displacement steps: {len(dx)}")
print(f" dx: std = {np.std(dx):.4f} px ({np.std(dx)*PIXEL_SIZE*1000:.1f} u
↪nm}")
print(f" dy: std = {np.std(dy):.4f} px ({np.std(dy)*PIXEL_SIZE*1000:.1f} u
↪nm}")

results['dx'] = dx
results['dy'] = dy
results['n_steps'] = len(dx)

# =====
# METHOD 1: DIRECT VARIANCE CALCULATION
# =====

var_dx = np.var(dx) # px^2
var_dy = np.var(dy) # px^2
var_dx_um = var_dx * PIXEL_SIZE**2 # um^2
var_dy_um = var_dy * PIXEL_SIZE**2 # um^2

D_x_direct = var_dx_um / (2 * dt)
D_y_direct = var_dy_um / (2 * dt)
D_avg_direct = (D_x_direct + D_y_direct) / 2

print(f"\n" + "-" * 40)
print(f"METHOD 1: Direct Variance")
print("-" * 40)
print(f" D_x = {D_x_direct:.4f} um^2/s")
print(f" D_y = {D_y_direct:.4f} um^2/s")
print(f" D_avg = {D_avg_direct:.4f} um^2/s")
print(f" D_theory = {D_theory_um:.4f} um^2/s")
print(f" Ratio = {D_avg_direct/D_theory_um:.2f}x")

results['D_direct'] = D_avg_direct
results['var_dx'] = var_dx
results['var_dy'] = var_dy

# =====
# METHOD 2: GAUSSIAN FIT TO HISTOGRAM
# =====

nbins = 20

# Fit X
counts_x, bins_x = np.histogram(dx, bins=nbins, density=True)

```



```
bin_centers_x = (bins_x[:-1] + bins_x[1:]) / 2
try:
    p0_x = [max(counts_x), np.mean(dx), np.std(dx)]
    popt_x, pcov_x = curve_fit(gaussian, bin_centers_x, counts_x, p0=p0_x)
    std_x_fit = abs(popt_x[2])
except:
    std_x_fit = np.std(dx)

# Fit Y
counts_y, bins_y = np.histogram(dy, bins=nbins, density=True)
bin_centers_y = (bins_y[:-1] + bins_y[1:]) / 2
try:
    p0_y = [max(counts_y), np.mean(dy), np.std(dy)]
    popt_y, pcov_y = curve_fit(gaussian, bin_centers_y, counts_y, p0=p0_y)
    std_y_fit = abs(popt_y[2])
except:
    std_y_fit = np.std(dy)

var_x_fit_um = (std_x_fit * PIXEL_SIZE)**2
var_y_fit_um = (std_y_fit * PIXEL_SIZE)**2
D_x_fit = var_x_fit_um / (2 * dt)
D_y_fit = var_y_fit_um / (2 * dt)
D_avg_fit = (D_x_fit + D_y_fit) / 2

print(f"\n" + "-" * 40)
print(f"METHOD 2: Gaussian Fit")
print("-" * 40)
print(f"  sigma_x = {std_x_fit:.4f} px")
print(f"  sigma_y = {std_y_fit:.4f} px")
print(f"  D_avg = {D_avg_fit:.4f} um^2/s")
print(f"  Ratio = {D_avg_fit/D_theory_um:.2f}x")

results['D_fit'] = D_avg_fit
results['std_x_fit'] = std_x_fit
results['std_y_fit'] = std_y_fit

# =====
# METHOD 3: MSD ANALYSIS
# =====
# Find minimum track length
min_track = min([seg['length'] for seg in selected_segments])
max_lag = min(min_track // 2, 30) # Cap at 30 for stability

# Calculate MSD for each segment
all_MSDs = []
for seg in selected_segments:
    x = seg['x'].copy()
```

```

y = seg['y'].copy()

if SUBTRACT_DRIFT:
    x = x - np.mean(x)
    y = y - np.mean(y)

MSD_seg = np.zeros(max_lag)
n_frames = len(x)

for lag in range(max_lag):
    if n_frames > lag:
        dx_lag = x[lag:] - x[:-lag] if lag > 0 else np.zeros(n_frames)
        dy_lag = y[lag:] - y[:-lag] if lag > 0 else np.zeros(n_frames)
        r_sq = dx_lag**2 + dy_lag**2
        MSD_seg[lag] = np.mean(r_sq) if len(r_sq) > 0 else 0

all_MSDs.append(MSD_seg)

all_MSDs = np.array(all_MSDs)
MSD = np.mean(all_MSDs, axis=0)
MSD_err = np.std(all_MSDs, axis=0) / np.sqrt(len(selected_segments))

# Convert to physical units
MSD_um = MSD * PIXEL_SIZE**2
MSD_err_um = MSD_err * PIXEL_SIZE**2
lag_times = np.arange(max_lag) * dt

# Fit linear region (exclude lag=0)
n_fit = min(15, max_lag // 2)
fit_times = lag_times[1:n_fit+1]
fit_MSD = MSD_um[1:n_fit+1]
fit_err = MSD_err_um[1:n_fit+1]

# Handle zero errors
fit_err = np.where(fit_err > 0, fit_err, 1e-10)

try:
    popt_msd, pcov_msd = curve_fit(linear, fit_times, fit_MSD,
                                    sigma=fit_err, absolute_sigma=True,
                                    p0=[1, 0])
    perr_msd = np.sqrt(np.diag(pcov_msd))
    slope = popt_msd[0]
    slope_err = perr_msd[0]
except:
    # Simple linear fit without weights
    coeffs = np.polyfit(fit_times, fit_MSD, 1)
    slope = coeffs[0]

```

```

slope_err = 0

D_msd = slope / 4 # For 2D: MSD = 4*D*t
D_msd_err = slope_err / 4

print(f"\n" + "-" * 40)
print(f"METHOD 3: MSD Slope")
print("-" * 40)
print(f" Slope = {slope:.4f} +/- {slope_err:.4f} um^2/s")
print(f" D_MSD = {D_msd:.4f} +/- {D_msd_err:.4f} um^2/s")
print(f" Ratio = {D_msd/D_theory_um:.2f}x")

results['D_msd'] = D_msd
results['D_msd_err'] = D_msd_err
results['MSD_um'] = MSD_um
results['MSD_err_um'] = MSD_err_um
results['lag_times'] = lag_times
results['msd_slope'] = slope

return results

```

Analysis function defined.

### 1.3 Run Analysis for All Bead Sizes

```
[4]: # Store results for all bead sizes
all_results = {}

for bead_diameter, file_path in DATA_FILES.items():
    label = f"{bead_diameter} um beads"
    result = analyze_bead_size(file_path, bead_diameter, label)
    if result is not None:
        all_results[bead_diameter] = result
    print("\n")
=====
```

ANALYZING 1.0 um beads

File: 1mu-21c-1isto224w-0\_5p-trackresults.txt

Total frames: 240

Total particles detected: 75

Track splitting (threshold: 20 px/frame):

Valid segments extracted: 64

(Min segment length: 20 frames)

Using top 10 segments:

Seg	Orig	Part	Frames
-----	------	------	--------

25	13		235
26	14		235
27	15		235
32	19		234
38	22		234
11	5		230
20	10		230
31	18		230
16	8		225
28	16		218

Drift correction applied:

Mean dx removed: 0.0094 px (3.3 nm)  
 Mean dy removed: 0.0260 px (9.0 nm)

Total displacement steps: 2296

dx: std = 0.8068 px (278.3 nm)  
 dy: std = 0.8026 px (276.9 nm)

---

#### METHOD 1: Direct Variance

---

D\_x = 8.7545  $\mu\text{m}^2/\text{s}$   
 D\_y = 8.6638  $\mu\text{m}^2/\text{s}$   
 D\_avg = 8.7092  $\mu\text{m}^2/\text{s}$   
 D\_theory = 0.4414  $\mu\text{m}^2/\text{s}$   
 Ratio = 19.73x

---

#### METHOD 2: Gaussian Fit

---

sigma\_x = 0.8121 px  
 sigma\_y = 0.8154 px  
 D\_avg = 8.9058  $\mu\text{m}^2/\text{s}$   
 Ratio = 20.17x

---

#### METHOD 3: MSD Slope

---

Slope = 51.5090 +/- 0.8529  $\mu\text{m}^2/\text{s}$   
 D\_MSD = 12.8773 +/- 0.2132  $\mu\text{m}^2/\text{s}$   
 Ratio = 29.17x

---



---



---

ANALYZING 5.0  $\mu\text{m}$  beads

49

=====

File: 5mu-21c-1isto6\_5w-0\_5p-trackresults.txt  
Total frames: 239  
Total particles detected: 17

Track splitting (threshold: 20 px/frame):  
Valid segments extracted: 21  
(Min segment length: 20 frames)

Using top 10 segments:

Seg	Orig	Part	Frames
3	2		238
4	3		238
5	4		238
6	5		238
7	6		238
10	8		238
13	10		238
14	11		238
17	13		238
18	14		238

Drift correction applied:

Mean dx removed: 0.0049 px (1.7 nm)  
Mean dy removed: -0.0020 px (-0.7 nm)

Total displacement steps: 2370  
dx: std = 0.4379 px (151.1 nm)  
dy: std = 0.4631 px (159.8 nm)

-----  
METHOD 1: Direct Variance

-----  
D\_x = 2.5791 um^2/s  
D\_y = 2.8851 um^2/s  
D\_avg = 2.7321 um^2/s  
D\_theory = 0.0883 um^2/s  
Ratio = 30.95x

-----  
METHOD 2: Gaussian Fit

-----  
sigma\_x = 0.3620 px  
sigma\_y = 0.4187 px  
D\_avg = 2.0602 um^2/s  
Ratio = 23.34x

METHOD 3: MSD Slope

Slope = 14.9964 +/- 0.6024 um<sup>2</sup>/s  
D\_MSD = 3.7491 +/- 0.1506 um<sup>2</sup>/s  
Ratio = 42.47x

## 1.4 Visualize Particle Trajectories

2D tracks for each bead size.

```
[5]: # Plot trajectories for each bead size
n_sizes = len(all_results)
if n_sizes > 0:
    fig, axes = plt.subplots(1, n_sizes, figsize=(7*n_sizes, 6))
    if n_sizes == 1:
        axes = [axes]

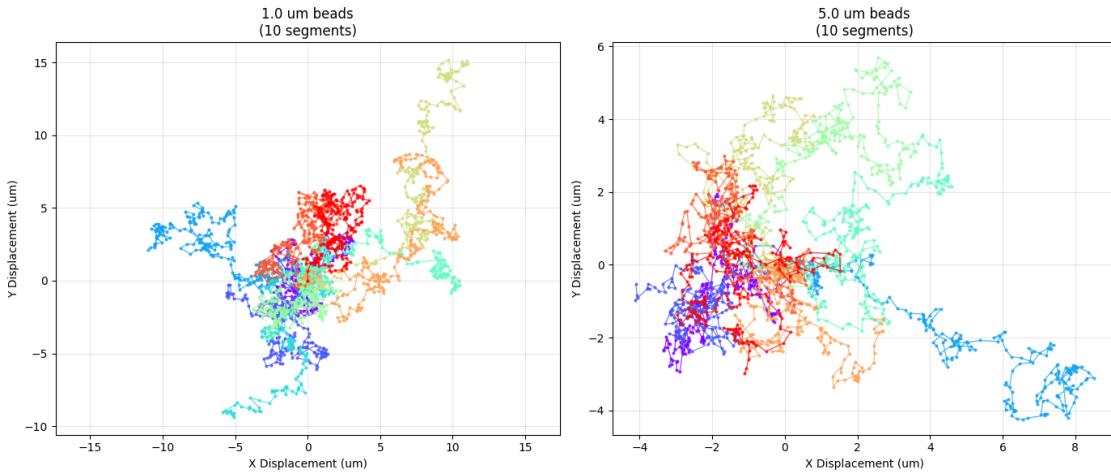
    for ax, (bead_d, res) in zip(axes, all_results.items()):
        segments = res['segments'][:10] # Show top 10
        colors = cm.rainbow(np.linspace(0, 1, len(segments)))

        for seg, c in zip(segments, colors):
            x_um = seg['x'] * PIXEL_SIZE
            y_um = seg['y'] * PIXEL_SIZE
            # Plot relative to start
            ax.plot(x_um - x_um[0], y_um - y_um[0], '-o',
                    markersize=2, linewidth=1, color=c, alpha=0.7)

        ax.set_xlabel('X Displacement (um)')
        ax.set_ylabel('Y Displacement (um)')
        ax.set_title(f'{res["label"]}\n{res["n_segments"]} segments')
        ax.axis('equal')
        ax.grid(True, alpha=0.3)

    plt.tight_layout()
    plt.show()
else:
    print("No data to plot.")
```

SI



## 1.5 Displacement Histograms

Plot histogram of X and Y displacements for each bead size.

**Source:** Histogram plotting and error bar calculation adapted from LabLevel References/Reference-code/Python demos/histogram\_demo\_using\_data.ipynb (Rev12 Feb. 18, 2021, JB, Physics SFU).

```
[6]: # Displacement histograms for each bead size
for bead_d, res in all_results.items():
    dx = res['dx']
    dy = res['dy']

    fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(14, 5))
    fig.suptitle(f'Displacement Distributions - {res["label"]}', fontsize=14)

    nbins = 20

    # X histogram
    counts_x, bins_x, _ = ax1.hist(dx, bins=nbins, density=True, alpha=0.7,
                                    color='lightblue', edgecolor='black', □
    ↪label='Data')
    bin_centers_x = (bins_x[:-1] + bins_x[1:]) / 2

    # Error bars (Poisson)
    counts_raw_x, _ = np.histogram(dx, bins=nbins)
    bin_widths_x = np.diff(bins_x)
    counts_err_x = np.sqrt(counts_raw_x) / (len(dx) * bin_widths_x)
    ax1.errorbar(bin_centers_x, counts_x, yerr=counts_err_x, fmt='ko',
                 markersize=4, capsize=4, label='Error bars')
```

```

# Gaussian overlay
mean_x = np.mean(dx)
std_x = np.std(dx)
x_range = np.linspace(dx.min(), dx.max(), 100)
gaussian_x = (1/(sqrt(2*pi)*std_x)) * exp(-(x_range - mean_x)**2 / (2*std_x**2))
ax1.plot(x_range, gaussian_x, 'r-', linewidth=2,
          label=f'Gaussian\n$\mu$={mean_x:.3f}\n$\sigma$={std_x:.3f}')

ax1.set_xlabel('Displacement in X (px)')
ax1.set_ylabel('Probability Density')
ax1.set_title('X Displacement')
ax1.legend()
ax1.grid(True, alpha=0.3)

# Y histogram
counts_y, bins_y, _ = ax2.hist(dy, bins=nbins, density=True, alpha=0.7,
                                 color='lightgreen', edgecolor='black', label='Data')
bin_centers_y = (bins_y[:-1] + bins_y[1:]) / 2

counts_raw_y, _ = np.histogram(dy, bins=nbins)
bin_widths_y = np.diff(bins_y)
counts_err_y = np.sqrt(counts_raw_y) / (len(dy) * bin_widths_y)
ax2.errorbar(bin_centers_y, counts_y, yerr=counts_err_y, fmt='ko',
              markersize=4, capsizes=4, label='Error bars')

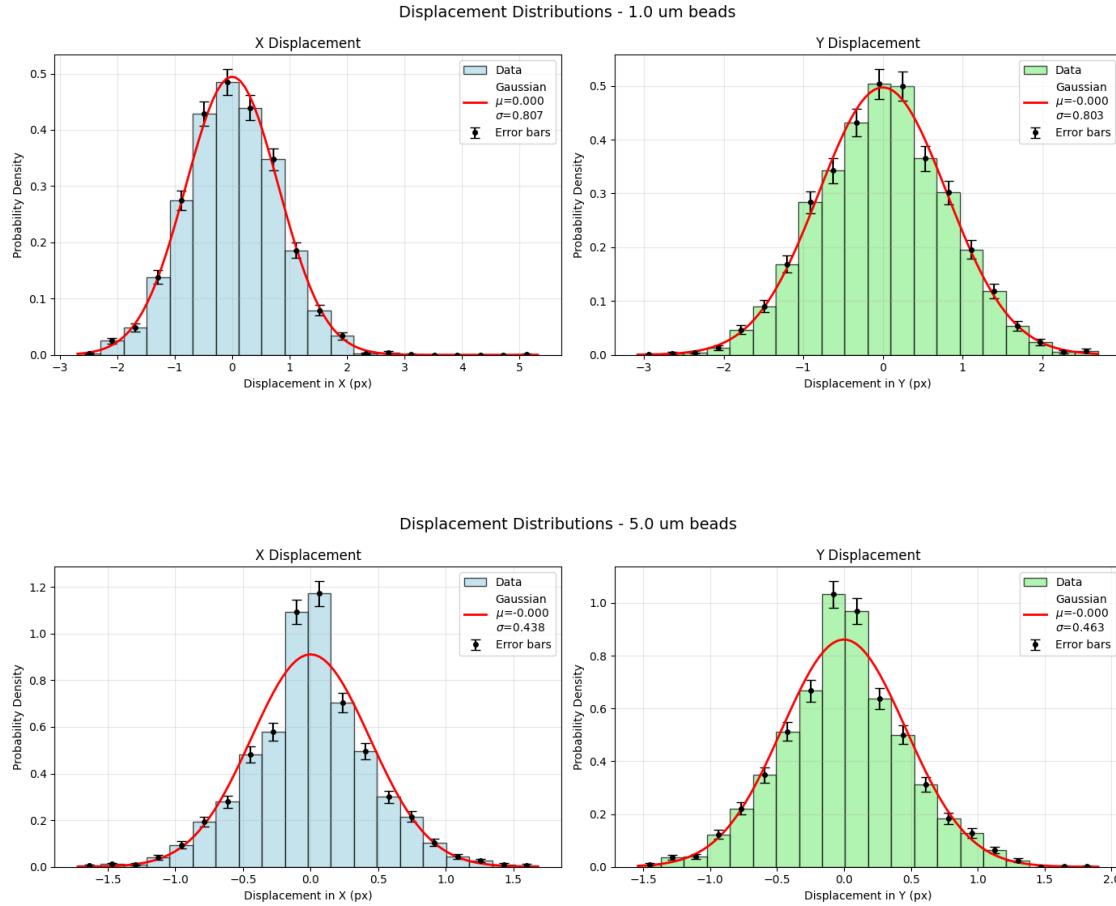
mean_y = np.mean(dy)
std_y = np.std(dy)
y_range = np.linspace(dy.min(), dy.max(), 100)
gaussian_y = (1/(sqrt(2*pi)*std_y)) * exp(-(y_range - mean_y)**2 / (2*std_y**2))
ax2.plot(y_range, gaussian_y, 'r-', linewidth=2,
          label=f'Gaussian\n$\mu$={mean_y:.3f}\n$\sigma$={std_y:.3f}')

ax2.set_xlabel('Displacement in Y (px)')
ax2.set_ylabel('Probability Density')
ax2.set_title('Y Displacement')
ax2.legend()
ax2.grid(True, alpha=0.3)

plt.tight_layout()
plt.show()

```

53



## 1.6 Mean-Squared Displacement (MSD) Analysis

Calculate MSD as a function of lag time:

$$\text{MSD}(\tau) = \langle [r(t + \tau) - r(t)]^2 \rangle$$

For 2D diffusion:  $\text{MSD} = 4\text{D} * \text{tau}$

**Source:** MSD calculation adapted from LabLevel References/Reference-code/Python demos/diffusion\_analysis\_demo.ipynb (Rev12 071119 DAL, Physics SFU).

```
[7]: # MSD plots for each bead size
for bead_d, res in all_results.items():
    lag_times = res['lag_times']
    MSD_um = res['MSD_um']
    MSD_err_um = res['MSD_err_um']
    D_msd = res['D_msd']
    D_msd_err = res['D_msd_err']
```

54

```

D_theory = res['D_theory']
slope = res['msd_slope']

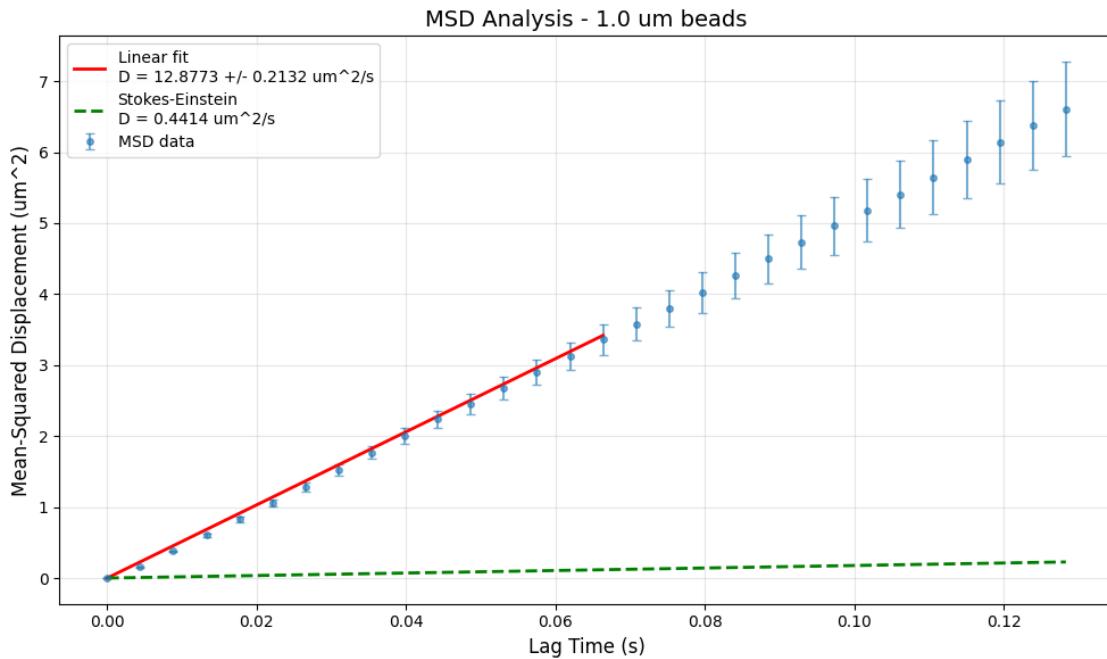
plt.figure(figsize=(10, 6))
plt.errorbar(lag_times, MSD_um, yerr=MSD_err_um, fmt='o',
              markersize=4, capsize=3, alpha=0.6, label='MSD data')

# Fit line
n_fit = min(15, len(lag_times) // 2)
fit_line = linear(lag_times[:n_fit+1], slope, 0)
plt.plot(lag_times[:n_fit+1], fit_line, 'r-', linewidth=2,
         label=f'Linear fit\nD = {D_msd:.4f} +/- {D_msd_err:.4f} um^2/s')

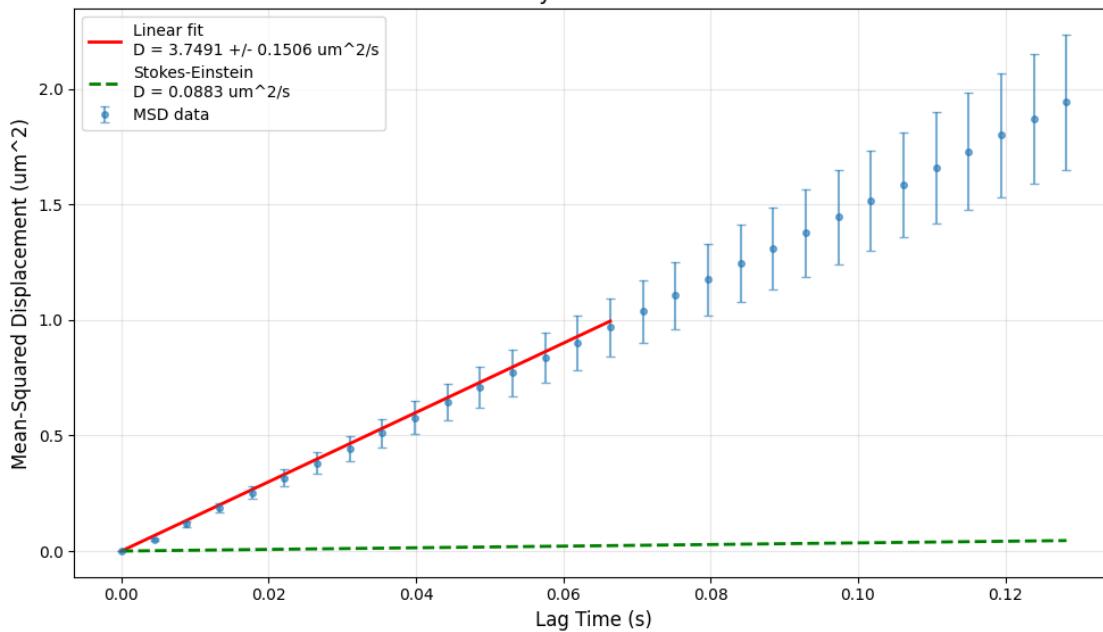
# Theory line
theory_msd = 4 * D_theory * lag_times
plt.plot(lag_times, theory_msd, 'g--', linewidth=2,
         label=f'Stokes-Einstein\nD = {D_theory:.4f} um^2/s')

plt.xlabel('Lag Time (s)', fontsize=12)
plt.ylabel('Mean-Squared Displacement (um^2)', fontsize=12)
plt.title(f'MSD Analysis - {res["label"]}', fontsize=14)
plt.grid(True, alpha=0.3)
plt.legend(fontsize=10)
plt.tight_layout()
plt.show()

```



MSD Analysis - 5.0 um beads



## 1.7 Summary of Results

```
[8]: # =====
# INDIVIDUAL BEAD SIZE SUMMARIES
# =====
for bead_d, res in all_results.items():
    print("=" * 60)
    print(f"DIFFUSION COEFFICIENT SUMMARY - {res['label'].upper()}")
    print("=" * 60)

    print(f"\nExperimental Setup:")
    print(f"  Bead diameter: {bead_d} um")
    print(f"  Temperature: {TEMPERATURE} K ({TEMPERATURE-273.15:.1f} C)")
    print(f"  Frame rate: {FRAME_RATE} fps")
    print(f"  Pixel size: {PIXEL_SIZE} um/px ({PIXEL_SIZE*1000:.0f} nm)")

    print(f"\nData Quality:")
    print(f"  Track segments used: {res['n_segments']}")
    print(f"  Total displacement steps: {res['n_steps']}")
    print(f"  Drift correction: {'ON' if SUBTRACT_DRIFT else 'OFF'}")

    print(f"\nDiffusion Coefficient Results:")
    print(f"  Method 1 (Direct variance):  D = {res['D_direct']:.4f} um^2/s")
    print(f"  Method 2 (Gaussian fit):      D = {res['D_fit']:.4f} um^2/s")
```

```

    print(f"  Method 3 (MSD slope):           D = {res['D_msd']:.4f} +/-_{-}
    ↪{res['D_msd_err']:.4f} um^2/s")
    print(f"  Stokes-Einstein prediction:   D = {res['D_theory']:.4f} um^2/s")

    print(f"\nComparison with Theory (D_exp/D_theory):")
    print(f"  Method 1: {res['D_direct']}/{res['D_theory']:.2f}x")
    print(f"  Method 2: {res['D_fit']}/{res['D_theory']:.2f}x")
    print(f"  Method 3: {res['D_msd']}/{res['D_theory']:.2f}x")
    print()

```

=====

#### DIFFUSION COEFFICIENT SUMMARY - 1.0 UM BEADS

=====

##### Experimental Setup:

Bead diameter: 1.0 um  
 Temperature: 294.15 K (21.0 C)  
 Frame rate: 226 fps  
 Pixel size: 0.345 um/px (345 nm)

##### Data Quality:

Track segments used: 10  
 Total displacement steps: 2296  
 Drift correction: ON

##### Diffusion Coefficient Results:

Method 1 (Direct variance): D = 8.7092 um^2/s  
 Method 2 (Gaussian fit): D = 8.9058 um^2/s  
 Method 3 (MSD slope): D = 12.8773 +/- 0.2132 um^2/s  
 Stokes-Einstein prediction: D = 0.4414 um^2/s

##### Comparison with Theory (D\_exp/D\_theory):

Method 1: 19.73x  
 Method 2: 20.17x  
 Method 3: 29.17x

=====

#### DIFFUSION COEFFICIENT SUMMARY - 5.0 UM BEADS

=====

##### Experimental Setup:

Bead diameter: 5.0 um  
 Temperature: 294.15 K (21.0 C)  
 Frame rate: 226 fps  
 Pixel size: 0.345 um/px (345 nm)

##### Data Quality:

Track segments used: 10

Total displacement steps: 2370  
Drift correction: ON

#### Diffusion Coefficient Results:

Method 1 (Direct variance):  $D = 2.7321 \text{ } \mu\text{m}^2/\text{s}$   
Method 2 (Gaussian fit):  $D = 2.0602 \text{ } \mu\text{m}^2/\text{s}$   
Method 3 (MSD slope):  $D = 3.7491 \pm 0.1506 \text{ } \mu\text{m}^2/\text{s}$   
Stokes-Einstein prediction:  $D = 0.0883 \text{ } \mu\text{m}^2/\text{s}$

#### Comparison with Theory ( $D_{\text{exp}}/D_{\text{theory}}$ ):

Method 1: 30.95x  
Method 2: 23.34x  
Method 3: 42.47x

## 1.8 Comparison Between Bead Sizes

According to Stokes-Einstein:  $D = \frac{k_B T}{6\pi\eta r}$

Therefore:  $D \propto 1/r$  (inverse relationship with radius)

Expected ratio:  $\frac{D_{1\mu\text{m}}}{D_{5\mu\text{m}}} = \frac{r_{5\mu\text{m}}}{r_{1\mu\text{m}}} = 5$

```
[9]: # =====
# COMPARISON BETWEEN BEAD SIZES
# =====
print("==" * 60)
print("COMPARISON BETWEEN BEAD SIZES")
print("==" * 60)

if len(all_results) >= 2:
    bead_sizes = sorted(all_results.keys())

    # Create comparison table
    print(f"\n{'Bead Size':<12} {'D_direct':<14} {'D_fit':<14} {'D_MSD':<14} "
        f"{'D_theory':<14}")
    print(f"{'(um)':<12} {'(um^2/s)':<14} {'(um^2/s)':<14} {'(um^2/s)':<14} "
        f"{'(um^2/s)':<14}")
    print("-" * 68)

    for bead_d in bead_sizes:
        res = all_results[bead_d]
        print(f"{bead_d:<12.1f} {res['D_direct']:<14.4f} {res['D_fit']:<14.4f} "
            f"{res['D_msd']:<14.4f} {res['D_theory']:<14.4f}")

    # Calculate ratios if we have both 1um and 5um
    if 1.0 in all_results and 5.0 in all_results:
        res_1 = all_results[1.0]
```

```

res_5 = all_results[5.0]

print(f"\n" + "-" * 60)
print(f"RATIO D(1um) / D(5um) - Expected: 5.0")
print("-" * 60)
print(f" Method 1 (Direct): {res_1['D_direct']/res_5['D_direct']:.2f}")
print(f" Method 2 (Fit): {res_1['D_fit']/res_5['D_fit']:.2f}")
print(f" Method 3 (MSD): {res_1['D_msd']/res_5['D_msd']:.2f}")
print(f" Theory: {res_1['D_theory']/res_5['D_theory']:.2f}")

# Bar chart comparison
fig, ax = plt.subplots(figsize=(10, 6))

x = np.arange(len(bead_sizes))
width = 0.2

d_direct = [all_results[b]['D_direct'] for b in bead_sizes]
d_fit = [all_results[b]['D_fit'] for b in bead_sizes]
d_msd = [all_results[b]['D_msd'] for b in bead_sizes]
d_theory = [all_results[b]['D_theory'] for b in bead_sizes]

bars1 = ax.bar(x - 1.5*width, d_direct, width, label='Direct Variance', color='steelblue')
bars2 = ax.bar(x - 0.5*width, d_fit, width, label='Gaussian Fit', color='darkorange')
bars3 = ax.bar(x + 0.5*width, d_msd, width, label='MSD Slope', color='forestgreen')
bars4 = ax.bar(x + 1.5*width, d_theory, width, label='Stokes-Einstein', color='crimson', alpha=0.7)

ax.set_xlabel('Bead Diameter (um)', fontsize=12)
ax.set_ylabel('Diffusion Coefficient (um^2/s)', fontsize=12)
ax.set_title('Diffusion Coefficient Comparison by Bead Size', fontsize=14)
ax.set_xticks(x)
ax.set_xticklabels([f'{b} um' for b in bead_sizes])
ax.legend()
ax.grid(True, alpha=0.3, axis='y')

plt.tight_layout()
plt.show()

else:
    print("\nNeed at least 2 bead sizes for comparison.")
    print("Check that data files exist and can be loaded.")

```

```
print("\n" + "=" * 60)
```

---

**COMPARISON BETWEEN BEAD SIZES**


---

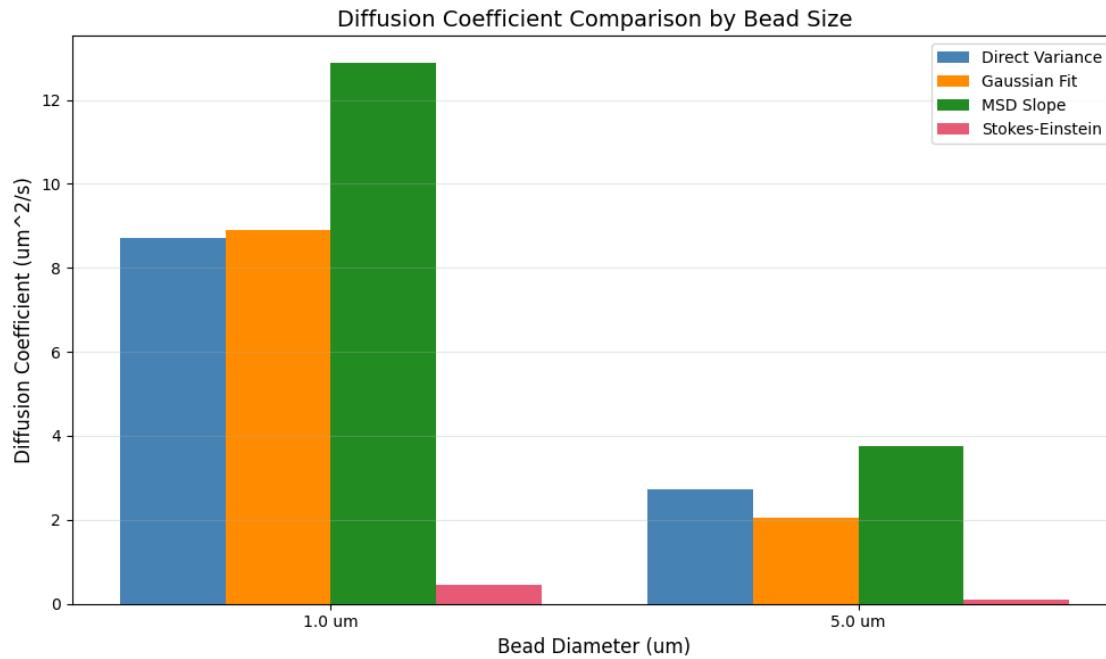
Bead Size (um)	D_direct (um^2/s)	D_fit (um^2/s)	D_MSD (um^2/s)	D_theory (um^2/s)
1.0	8.7092	8.9058	12.8773	0.4414
5.0	2.7321	2.0602	3.7491	0.0883

---

RATIO D(1um) / D(5um) - Expected: 5.0

---

Method 1 (Direct): 3.19  
 Method 2 (Fit): 4.32  
 Method 3 (MSD): 3.43  
 Theory: 5.00



## 1.9 Final Summary Table

```
[10]: # =====
# FINAL SUMMARY TABLE
# =====
print("=" * 70)
print("FINAL SUMMARY")
print("=" * 70)

print(f"\nExperimental Conditions:")
print(f" Temperature: {TEMPERATURE} K ({TEMPERATURE-273.15:.1f} C)")
print(f" Viscosity: {VISCOSITY:.6f} Pa.s")
print(f" Frame rate: {FRAME_RATE} fps")
print(f" Pixel size: {PIXEL_SIZE} um/px")

print(f"\nAnalysis Parameters:")
print(f" Track split threshold: {MAX_DISPLACEMENT_PER_FRAME} px/frame")
print(f" Min segment length: {MIN_TRACK_LENGTH} frames")
print(f" Max segments used: {NUM_BEST_SEGMENTS}")
print(f" Drift correction: {'ON' if SUBTRACT_DRIFT else 'OFF'}")

print(f"\n" + "-" * 70)
print(f"{'Bead':<8} {'Segments':<10} {'Steps':<8} {'D_MSD':<16} {'D_theory':<14} {'Ratio':<14}")
print(f"{'(um)':<8} {'used':<10} {'total':<8} {'(um^2/s)':<16} {'(um^2/s)':<14} ↴{'exp/theory':<14}")
print("-" * 70)

for bead_d in sorted(all_results.keys()):
    res = all_results[bead_d]
    d_str = f"{res['D_msd']:.4f} +/- {res['D_msd_err']:.4f}"
    ratio = res['D_msd'] / res['D_theory']
    print(f"{bead_d:<8.1f} {res['n_segments']:<10} {res['n_steps']:<8} "
          f"{d_str:<16} {res['D_theory']:<14.4f} {ratio:.2f}x")

print("=" * 70)
```

=====

FINAL SUMMARY

=====

Experimental Conditions:

Temperature: 294.15 K (21.0 C)  
 Viscosity: 0.000976 Pa.s  
 Frame rate: 226 fps  
 Pixel size: 0.345 um/px

Analysis Parameters:

61

Track split threshold: 20 px/frame

Min segment length: 20 frames

Max segments used: 10

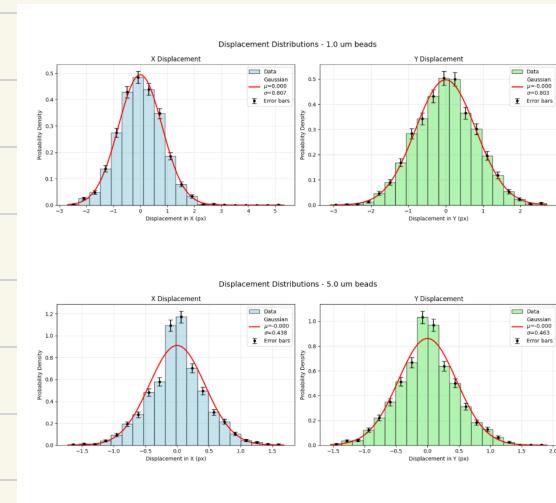
Drift correction: ON

Bead (um)	Segments used	Steps total	D_MSD (um^2/s)	D_theory (um^2/s)	Ratio exp/theory
1.0	10	2296	12.8773 +/- 0.2132	0.4414	29.17x
5.0	10	2370	3.7491 +/- 0.1506	0.0883	42.47x

## 10. Discussions, reflections and suggestions

Figure 1:  
Displacement  
Distribution for  
both Particles

Refer to pg 53 of Lab Notebook



The Distributions are Gaussian. However the  $x$  displacement for 1.0 micro beads is not Symmetrical tho.

Refer to pg 59

Table 1 : comparing  
3 Different methods  
for finding  $D$  for 2  
particle sizes.

COMPARISON BETWEEN BEAD SIZES				
Bead Size (μm)	$D_{direct}$ ( $\mu\text{m}^2/\text{s}$ )	$D_{fit}$ ( $\mu\text{m}^2/\text{s}$ )	$D_{MSD}$ ( $\mu\text{m}^2/\text{s}$ )	$D_{theory}$ ( $\mu\text{m}^2/\text{s}$ )
1.0	8.7092	8.9058	12.8773	0.4414
5.0	2.7321	2.0602	3.7491	0.0883

Critical Finding:  $D$  is 20-40x Higher Than Theory

Most Likely Cause: Pixel Calibration Error  
since  $D$  scales as (pixel size)<sup>2</sup> if actual pixel size  
is 5x larger than the assumed 0.345  $\mu\text{m}/\text{px}$ , the  
values would match theory.

Check calibration images!

## What is Working:

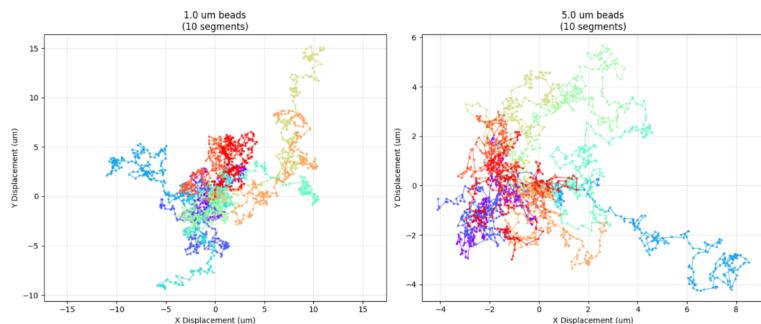
1. Displacements are Gaussian - histograms are bell-shaped, symmetric
2. MSD is linear - confirms diffusive motion
3. Size dependence exists - 1um beads move more than 5um beads
4. Methods agree - Variance and Gaussian fit give similar results

## What Needs Attention:

1.  $D(1\text{um})/D(5\text{um}) = 3.5$  instead of expected 5.0
2. Absolute D values are 20-40x too high
3. 5um beads show more discrepancy between methods (possible tracking issues)

Refer Pg 51

Figure 2: 2D movement of particles



Well to me, this does look pretty Brownian, that I am not able to understand is, why the paths interfere with each other. When we observed the .avi files, the particles do not move in such a way, they have their own distinct path where they wiggle. Investigate this.

## Recommended Action:

Verify pixel calibration using the stage micrometer images (calibration-feb5-100x.tif). The 0.345  $\mu\text{m}/\text{px}$  value may be incorrect for actual magnification setting.

## 11. Goal Reflections

1. Yes we re did Dilution and although we did not get the right number of particles in FOV yet, we are getting close
2. Yes, we did take multiple Calibration Values and also found error
3. Verified true Camera Pixel Size with TA
4. While we were able to collect 240 frames, we were not able to collect 1800 frames.
5. Quantitative Analysis done for two particles sizes
6. Work flow is better now
7. Could not explore Live cells but really excited for it!!!!

# Lab Notebook Submission Summaries

Mobility & Mortality Lab (Lab 2) – Sessions 1 & 2

PHYS 332

## Pre-Lab Questions (Pages i–iii)

### Pre-Lab Overview

**Questions 1–5:** Dilution calculation, RMS speed, random walk statistics, polymer end-to-end distance, diffusion timescale.

Item	Page
Q1: Dilution factor for $\sim 20$ beads in FOV; calibration $S = 100 \text{ nm}/\text{px}$	i
Volume of field: $128.8 \times 96.4 \times 5 \mu\text{m}^3$ ; dilution $\approx 1:28$	i
Q2: RMS speed $v_{rms} = \sqrt{3k_B T/m}$ for $\text{H}_2$ and $E. \text{coli}$ at $T = 300 \text{ K}$	ii
Q3: All 4-step random walk sequences; $\langle r_x^2 \rangle = N$ for $N$ steps	ii
Q4: Polymer end-to-end distance: $r_{rms} = b\sqrt{N} \approx 7.7 \text{ nm}$	ii
Q5: Diffusion time for protein across bacterium ( $r = 2 \mu\text{m}$ , $D = 10 \mu\text{m}^2/\text{s}$ ): $t \approx 0.067 \text{ s}$	iii

## Session 1: Exploring Microscope and Imaging (Pages 1–16)

### Session 1 Overview

**Date:** 3-Feb-2026

**Lab Partner:** Nathan Unruh

**Pages:** 1–16

**Objective:** Set up bright-field microscope with Köhler illumination; calibrate  $100\times$  oil objective ( $\leq 2\%$  uncertainty); prepare  $1 \mu\text{m}$  bead sample; capture Brownian motion video; extract trajectories and compute diffusion coefficient.

Item	Page
<b><i>Preparation</i></b>	
Goals (6 items), Apparatus table, Variables table	2–3
References: Lab documents, protocols, textbooks, GitHub link	3
<b><i>Setup &amp; Köhler Illumination</i></b>	
Image 1: Microscope photo (eyepiece, condenser, fine focus labelled)	4
Image 2: Translational stage close-up; calibration slide	4
Köhler procedure: 7 steps; aperture diaphragm to 70% of objective NA	5
Note: Field diaphragm → illumination area; aperture → NA/contrast	5
<b><i>Camera Calibration</i></b>	
Error: Lab AccuSion not detecting microscope → Solution: restart	6
Image 3: Letter “A” used to locate grating; screenshot of working window	6
8 lines in 100× objective, edge to edge: 1168.706 pixels	7
Calibration: 73.044 px/line ⇒ <b>68.45 nm/px</b>	7
Camera model: BFS-U3-16S2C-CS; FOV: 1440×1080 pixels	7
<b><i>Sample Preparation</i></b>	
Materials: 0.5 wt% stock, glass slides, 22×22 mm coverslips, Parafilm spacers	9
Dilution: 1 μL stock + 30 μL water (1:30); pipette mix 10×	9
Chamber: thick method, Parafilm 100 μm spacers, sealed with nail polish	10
Quick check: 25–30 beads in FOV at 100×; no aggregation	10
<b><i>Data Collection</i></b>	
Room temperature: $22.3 \pm 0.5^\circ\text{C}$	11
Recorded 120 frames at 1 fps (two files: 1-frame and 5-frame versions)	11
Issue 1: Drift (all beads same direction) → more nail polish, waited 5 min	11
Issue 2: Beads in/out of focus → track only in-focus beads	12
Conclusion: Successfully recorded 60 s video at 1 fps (60 frames)	12
<b><i>Analysis (In-Lab)</i></b>	
MTrack2 plugin in ImageJ; background subtraction (rolling ball = 60 px)	13
File delimiter and code errors → fixed in <b>MTrack2Loader-fixed.ipynb</b>	14
Figure 1: 1D particle tracks — unidirectional flow (not random)	15
Figure 2: 2D tracks — small local motion, unclear if Brownian	15
<b><i>Session 1 Conclusions</i></b>	
Pipeline works; concentration too high; MTrack2 parameters need tuning	16
Sampling duration may be too short	16

**Key Files:** home/Data/2026-02-03/1um-1frame.avi, 1um-5frame.avi, imagecalibration.tif

## Post-Lab (Session 1): Second Attempt (Pages 17–22)

### Post-Lab Overview

**Recorder:** Ahilan (solo) **Pages:** 17–22  
**Goal:** Download ImageJ/Fiji locally; explore MTrack2 parameters; obtain better track data.

Item	Page
Downloaded Fiji (contains MTrack2 under Trackers folder)	17
Threshold: Upper 138, Lower 255; uniform background	18
Particle size range: 13–37 px; max velocity set very large	19
Minimum Track Length most sensitive parameter	19
Parameter sweep table (4 combinations): min size vs min track	20
Observation: Min Track Length is most influential parameter	20
Saved: 1um-post-lab-trackresults.txt (min size 10, min track 11)	20
Figure 3: 2D post-lab plot — Brownian motion visible	21
Figure 4: 1D post-lab plot — linear flow still present, need more beads	21
<b>Post-Lab Analysis / Plan for Next Lab</b>	
Recheck dilution; need calibration error from multiple measurements	22
Verify true camera pixel size; collect $\geq 3$ min of data	22
Establish solid workflow; move to larger beads; compare $D$	22

## Session 2: Larger Particles & Refining Workflow (Pages 23–33)

### Session 2 Overview

**Date:** 5-Feb-2026      **Lab Partner:** Nathan Unrhu      **Pages:** 23–33  
**Objective:** Redo dilution; take multiple calibration values; verify camera pixel size; collect more frames; analyse 1  $\mu\text{m}$  and 5  $\mu\text{m}$  beads; refine workflow.

Item	Page
<b><i>Preparation</i></b>	
Goals (7 items), Apparatus (additionally: 10 $\times$ , 40 $\times$ , 100 $\times$ oil objectives)	24–25
Software: ImageJ/Fiji + MTrack2; Python analysis code by Ahilan	25
Samples: 1 $\mu\text{m}$ and 5 $\mu\text{m}$ beads; glass slides, Parafilm/double-sided tape	25
<b><i>Procedure (Recorder: Ahilan)</i></b>	
Verified with TA: Labscript FOV is inaccurate; true FOV = 1440 $\times$ 1080 px	26
Learned: focus progressively 10 $\times$ $\rightarrow$ 100 $\times$ (faster than jumping)	26
Clean calibration slide with ethanol + lens paper (dab, not wipe)	26
Lens liquid contains additives — avoid direct contact	26
Redid small beads: 200 $\mu\text{L}$ of 1 $\mu\text{m}$ at 0.5%, factor 1:17	26
Large beads (5 $\mu\text{m}$ ): factor 1:6.5	27
<b><i>Procedure (Recorder: Nathan)</i></b>	
Camera errors: fixed by removing device + switching USB port	28
16.57 $\times$ dilution of 1 $\mu\text{m}$ beads; bead count increased (unexpected)	28
Memory limit: only $\sim$ 120 frames recordable	28
Data files: 5mu-0_5p.avi, initial-1mu-0_5p.avi	28
<b><i>Recalibration</i></b>	
100 $\times$ objective, 8 lines, calibration slide 1 div = 0.06 mm	29
Figure 1: Calibration lines photo at 100 $\times$	29
<b>Result:</b> 1 px $\approx$ 0.345 $\mu\text{m}$ (from 0.2 $\times$ 0.05/159.3 px)	29
<b><i>Calibration Uncertainty</i></b>	
Table 1: 5 measurements of edge-to-edge pixel values	30
Mean pixel length: $\bar{L} = 1156.98$ px; $\sigma = 8.40$ px	30
Standard error of mean: $\sigma_{\bar{L}} = 3.76$ px	30
<b><i>Investigating Small and Large Particles</i></b>	
1 $\mu\text{m}$ solution needed 1:224 dilution for 10–20 particles (17.5 $\times$ off calculation)	31
Sources of error: sample prep; drawing from surface increases concentration	31
5 $\mu\text{m}$ : 1:6.5 dilution confirmed	31
Better tracking by using dark rings (not bright centres) for 1 $\mu\text{m}$	32
Figure 2: Tracking parameters for 1 $\mu\text{m}$ ; Figure 3: for 5 $\mu\text{m}$	33

**Key Files:** 5mu-21C-1isto224w-0\_5p.avi, 5mu-21C-1isto6\_5w-0\_5p.avi; Data folder: Data/06-02-2026

## Diffusion Analysis Code (Pages 34–58)

### Analysis Code Overview

**Author:** Ahilan Kumaresan      **Date:** February 6, 2026      **Pages:** 34–58  
**Description:** Brownian Motion Diffusion Analysis notebook. Built from reference code + LLM assistance. Analyses multiple bead sizes for diffusion coefficient comparison.

Item	Page
<b>Code Structure</b>	
Control variables: data files, frame rate, pixel size, temperature, viscosity	34–35
Helper functions: <code>load_mtrack2_data</code> , <code>split_tracks_at_jumps</code>	36–38
Main analysis function: <code>analyze_bead_size</code>	38–44
<b>Parameters Used</b>	
Frame rate: 228 fps; Pixel size: $0.345 \mu\text{m}/\text{px}$ ; $T = 294.15 \text{ K}$ ( $21.0^\circ\text{C}$ )	35
Viscosity: $0.000976 \text{ Pa}\cdot\text{s}$ ; Track split threshold: 20 px/frame	35
Min track length: 20 frames; Top 10 segments used	35
<b>Results: <math>1.0 \mu\text{m}</math> Beads</b>	
75 particles detected, 64 valid segments after splitting	44–45
Method 1 (Direct Variance): $D = 8.7002 \mu\text{m}^2/\text{s}$	45
Method 2 (Gaussian Fit): $D = 8.9058 \mu\text{m}^2/\text{s}$	45
Method 3 (MSD Slope): $D = 12.8773 \pm 0.2132 \mu\text{m}^2/\text{s}$	45
Stokes–Einstein prediction: $D_{\text{theory}} = 0.4414 \mu\text{m}^2/\text{s}$	45
<b>Results: <math>5.0 \mu\text{m}</math> Beads</b>	
17 particles detected, 20 valid segments	46
Method 1 (Direct Variance): $D = 2.7321 \mu\text{m}^2/\text{s}$	46
Method 2 (Gaussian Fit): $D = 2.0603 \mu\text{m}^2/\text{s}$	46–47
Method 3 (MSD Slope): $D = 3.7481 \pm 0.1506 \mu\text{m}^2/\text{s}$	47
Stokes–Einstein prediction: $D_{\text{theory}} = 0.0883 \mu\text{m}^2/\text{s}$	47
<b>Visualisations</b>	
2D particle trajectories (both sizes)	47–48
Displacement histograms with Gaussian fits and error bars	48–50
MSD vs lag time plots with linear fit and Stokes–Einstein line	51–52
Bar chart: diffusion coefficient comparison by bead size	55–56
<b>Comparison Between Bead Sizes</b>	
$D(1 \mu\text{m})/D(5 \mu\text{m})$ ratio: 3.19 (Direct), 4.32 (Fit), 3.43 (MSD)	56
Expected ratio (Stokes–Einstein): 5.0	56
Final summary table	57–58

## Post-Lab: Discussions & Reflections (Pages 59–61)

Item	Page
Figure 1: Displacement distributions — Gaussian but $x$ for 1 $\mu\text{m}$ asymmetric	59
Table 1: Comparison of 3 methods for both particle sizes	59
<b>Critical finding:</b> $D$ is 20–40× higher than theory	59
Most likely cause: pixel calibration error ( $D \propto (\text{pixelsize})^2$ )	59
What is working: Gaussian displacements, linear MSD, size dependence	60
What needs attention: $D$ ratio = 3.5 instead of 5.0; absolute $D$ too high	60
Recommended action: verify pixel calibration using stage micrometer images	60
Goal reflections: dilution improved, calibration done, workflow better	61
Could not explore live cells — excited for future sessions	61

## Executive Summary

### Main Figures

Image 1	Microscope setup (labelled components)	p. 4
Image 2	Translational stage close-up	p. 4
Figure 1	Calibration lines at 100 $\times$ (Session 2)	p. 29
Figure 2	MTrack2 parameters for 1 $\mu\text{m}$ tracking	p. 33
Figure 3	MTrack2 parameters for 5 $\mu\text{m}$ tracking	p. 33
Figures	2D trajectories, histograms, MSD plots, bar chart	p. 47–56

### Main Tables

Table (Pre-lab)	Dilution calculation	p. i
Table (Session 1)	Variables: dependent, control	p. 3
Table (Session 2)	5 calibration measurements	p. 30
Table (Post-lab 1)	MTrack2 parameter sweep	p. 20
Table (Analysis)	Final summary: $D$ for both bead sizes	p. 56–58

### Boxed Conclusions

#### Calibration (p. 29–30):

1 px  $\approx$  0.345  $\mu\text{m}$  (100 $\times$  oil objective)  
 Mean:  $\bar{L} = 1156.98$  px,  $\sigma = 8.40$  px, SEM = 3.76 px  
 Camera: BFS-U3-16S2C-CS, FOV: 1440 $\times$ 1080 px

#### Diffusion Coefficients (p. 56–58):

	$D_{MSD}$ ( $\mu\text{m}^2/\text{s}$ )	$D_{theory}$ ( $\mu\text{m}^2/\text{s}$ )	Ratio
1.0 $\mu\text{m}$	$12.88 \pm 0.21$	0.4414	29 $\times$
5.0 $\mu\text{m}$	$3.75 \pm 0.15$	0.0883	42 $\times$

Measured  $D$  is 20–40 $\times$  higher than Stokes–Einstein.  
 Most likely cause: **pixel calibration error** ( $D \propto s^2$ ).

#### Key Physics (p. 59–60):

Displacement distributions are Gaussian  $\Rightarrow$  confirms diffusive motion.  
 MSD is linear in lag time  $\Rightarrow$  consistent with Brownian motion.  
 $D(1\,\mu\text{m})/D(5\,\mu\text{m}) \approx 3.5$  (expected 5.0) — size dependence observed but off.

### Summary

Sessions 1–2 established the Brownian motion measurement pipeline: Köhler illumination setup, camera calibration (0.345  $\mu\text{m}/\text{px}$ ), sample preparation with dilution optimization, video capture, MTrack2 particle tracking, and Python-based diffusion analysis. Three independent methods (direct variance, Gaussian fit, MSD slope) consistently yield diffusion coefficients 20–40 $\times$  above Stokes–Einstein predictions, with the  $D(1\,\mu\text{m})/D(5\,\mu\text{m})$  ratio at  $\sim 3.5$  instead of 5.0. The dominant systematic error is likely

pixel calibration. The displacement distributions are Gaussian and MSD is linear, confirming diffusive motion qualitatively.

**Not completed:** Live cell exploration; achieving target 20 beads in FOV; resolving calibration discrepancy; collecting 1800+ frames.

**Repository:** <https://github.com/Ahilan-Bucket/phys332W-sfu/tree/main/Lab2-Microscopy-and-Motil>