LAB 1: OPTICAL TWEEZERS

Alejandra Rosselli Calderon

October 2020

Contents

1 Abstract

The purpose of this experiment is to analyze the motion of a 1μm bead of latex suspended in water when confined by an optical trap. The optical trap used in this experiment was created by a 632nm laser. The optical setup consisted of a collection of mirrors and lenses that allowed me to focus the beam of light at a specific point in the glass screen. By using a polarizer, the power of the laser could be adjusted in order to observe the relationship between the power and the strength of the trap, which is observed when the bead is more confined and the Brownian motion decreases. To analyze the motion, I recorded videos with a camera and then used python to track the latex bead over time.

2 Introduction & Theory

Brownian motion is the random movement of a particle in a fluid due to the collisions of the atoms or molecules in the fluid against the particle [1] . In this experiment, I am looking to decrease the Brownian motion of a 1μm latex bead suspended in water. One way of doing this is through the use of lasers to create an optical trap. Optical traps use lasers to hold particles in place. This procedure is used across the different sciences, since it allows the trapping ranging from atoms or molecules to DNA and gas bubbles.

There are two main approaches to optical tweezers: the Rayleigh regime, where the wavelength of the light λ is greater than the radius of the particle R, and the Mie regime, where the radius R is much greater than λ [2].

The Rayleigh regime can be explained through electrodynamics. In this regime, the particle can be treated as a dialectric in an external electric field. The applied magnetic field polarizes the particle, which makes it behave as a point dipole. Since the applied electric field is slowly varying, the point dipole oscillates [3]. The energy of the particle is:

$$W = UV \tag{1}$$

where U is the electrostatic energy density and V is the volume. The electrostatic energy density U is defined as:

$$U = -\frac{1}{2}\vec{P} \cdot \vec{E}$$

where $\vec{P}$ is the polarizarion and $\vec{E}$ is the electric field. Since $U \propto -E^2$, it is also proportional to the intensity $-I$. For particles with a higher index of refraction than the surrounding medium, there is an attractive force towards the high intensity regions, which is the point where the trap is focused. [3]

The gradient of the force on the sphere is described by the polarizability α, speed of light c, index of refraction of the medium n_m and intensity of light I as follows:

$$F_{grad} = \frac{2\pi\alpha}{cn_m^2}\nabla I \tag{2}$$

The other force on this system is the scattering force that comes from the absorption and reradiation of the light. The oscillation of the dielectric in the applied magnetic field causes the dielectric to radiate electromagnetic energy. The effective force on the particle is:

$$F_{scat} = \frac{I\sigma n_m}{c} \tag{3}$$

where c is the speed of light.

One can understand the basics of optical tweezers in the Mie regime by analyzing the behavior of the momentum and simple optics. Light carries a momentum that is proportional to it's energy, $p = \frac{h}{\lambda}$, where λ is the wavelength of light and h is Planck's constant. When light hits a surface, it can refract or reflect. For the case of a clear sphere in water, light does both. This change in the direction of light changes its momentum. Since momentum needs to stay constant over time, the particle would need to have a change in momentum equal in magnitude and opposite in direction from the change in the momentum of the light.

A laser serves as a unidirectional ray of light, which allows us to know the original direction of the momentum of the light. To create a stable trap, we must aim to focus the laser close enough to the particle. This ensures that small movements about the trap stabilize and the particle stays fixed. [4] This can be observed in Figure 1. The net force due to the refraction increases with the angle θ while the net force due to the reflection decreases as θ increases. This forces will balance when the sphere gets to the point z of stable equilibrium since the force of refraction acts as a restoring force.

In this experiment, the radius of the latex bead is similar to the wavelength of the laser used. This means that we are in between the two regimes and the basic optics along with conservation of momentum is not enough to completely understand the behavior of the optical trap.

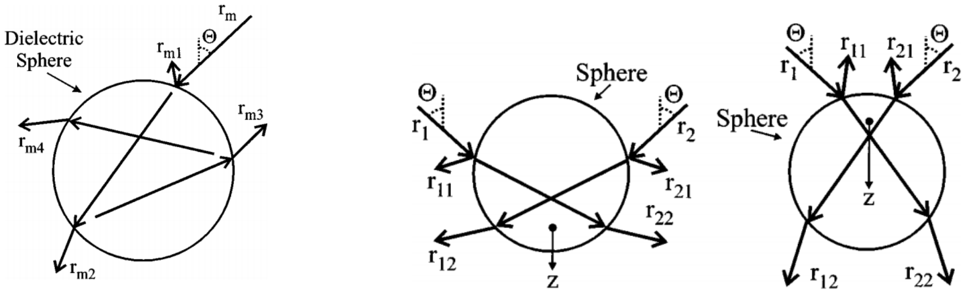

Figure 1: Left: Ray diagram for a sphere in the Mie regime. r_m is the incident ray, θ the incident angle, r_{m1} the reflected ray, r_{m2} the refracted ray and r_{m3} and r_{m4} the subsequent refracted rays. **Right:** Ray diagram of the net force acting on the dielectric sphere when above or below the focal point of the trap (z) [4]

The strength of the trap can be calculated by its equipartition function:

$$\langle U(x)\rangle = \frac{1}{2}k_x\langle x^2\rangle = \frac{1}{2}k_bT \tag{4}$$

where k_x is the spring constant of the optical trap, k_b is the Boltzmann constant, T is the temperature in Kelvin and $\langle x^2\rangle$ is the average square deviation from the mean particle position. This equation assumes that the potential of the trap is parabolic. To check this assumption we can make a Gaussian fit of the distributions using the following probability density:

$$\rho(x) = \frac{e^{-U(x)/k_bT}}{Z} \tag{5}$$

where Z is the partition function.

Based on this theoretical background, I expected the trap to be stronger as the power of the laser increased. A stronger trap would reflect in a decrease in the oscillations in the position as a function of time.

An oil immersion microscope was used throughout this experiment. The immersion oil has an index of refraction of about 1.51. This index of refraction is similar to that of the cover slip. There is refraction at each one of the interfaces that the laser goes through, but having an interface where the indices of refraction of both materials is similar decreases the refraction. This is important because we have the whole setup well aligned and focused, and the refraction between interfaces might ruin the focus which won't allow us to create a trap.

3 Experiment

The first part of the experiment consisted on familiarizing myself with the optics apparatus and then aligning the optics. The optical trap setup starts with the laser. The laser in the lab has a wavelength 632 nm and can get up to a power of 17mW. The first optical piece it goes through is a polarizer. The polarizer is what allows us to control the power of the laser. One thing that I had to remember was that the polarization is continuous, which means that by turning the polarizer around the power first increases and then decreases to get to the same original value every full turn. The next lens was a 50 mm focusing length, which allows the beam to be expanded by the time it reaches the microscope objective and to focus the beam at the back focal point of the microscope. This is important because it allows the trap and the microscope to be in focus. This is needed to be able to trap a bead within the slide and be able to properly observe it.

After the focusing lens, the laser bounces off an silver mirror at a 90°angle and reaches a dichroic mirror. Dichroic mirrors are important because they reflect some wavelengths of lights but allow other wavelengths of light to pass. In this case, the dichroic mirror reflects the incident light from the laser but allows the white back light of the microscope to pass through. From this mirror, the light goes into the microscope objective. It it important to note back focal length of the microscope objective in order to properly align the optical trap. At the beginning I neglected this step and had to go back to correct the alignment when the trap was not trapping the latex beads correctly.

The microscope objective that I worked with had a back focal length of 160 mm. This means that the laser should be focused to a point 160mm before it reaches the lens. To ensure that this distance is achieves, the 50mm focal lens had to be 50mm before this focus point. By placing the focal lens 210mm before the microscope objective, we could ensure that the light would be focused to a point when reaching the microscope objective.

After passing the microscope objective, the light reaches the slide where it is focused to a point. The slide is placed on a XYZ stage in order to get the focused point of light to be in the proper location. This stage also allowed me to move the sample around in order to get the latex beads into the optical trap. The XYZ stage was not optimal and small movements in one direction would sometimes affect the other two but moving the three axis around slightly would get the laser to be in the right location to trap a bead.

Behind the slide there is a back light that allows the camera to see what is happening within the slide. The light from this back light passes straight through the slide, microscope objective and two dichroic mirrors before reaching the camera. The second dichroic mirror that it goes through has the primary purpose of filtering the light from the laser. When aligning the trap and looking for beads that could be confined, I found it useful to remove this mirror. This allowed me to see where the beads were relative to the optical trap. When I needed to collect data, I would put this mirror back in front of the camera. A picture of the setup is included in Figure 2.

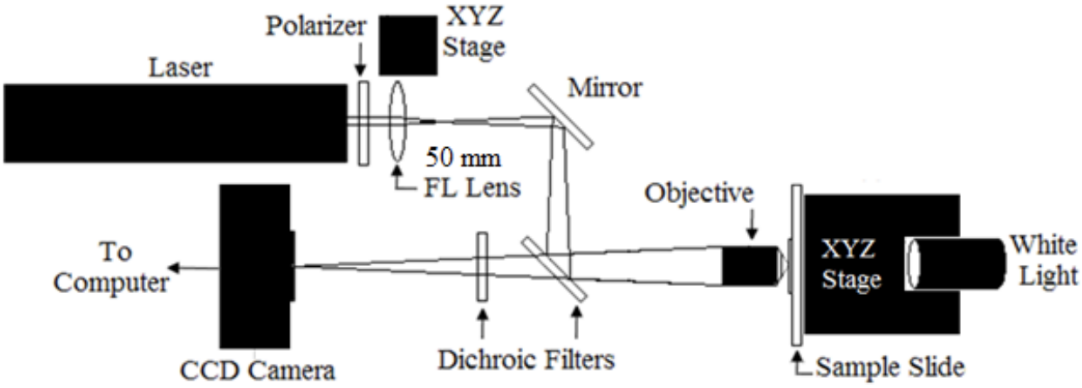

Figure 2: Diagram of the setup of the laser optics for optical tweezers.

Once the laser was setup, I proceeded to make the slides. I used a glass slide as my base. Then I cut a square piece of parafilm and cut out a small square out from the center of it. This gave me a frame that would keep the cover slip slightly separated from the glass slide. I put some pressure on the parafilm frame to stick it to the glass slide and then put the glass slide over an incandescent light. This would heat up the wax which made it stick better to the glass slide and eventually cover slip. I found it useful to have an slide over the incandescent light at all times, because the slide covers would often fall off and having a slide that was almost ready saved me time. After heating up the slide, I carefully placed it on a wood board and applied a small droplet of water in the middle of the frame. The droplet cannot be too small, because it might dry up quickly. After the water was on the slide, I put the cover slip and applied pressure to stick it to the parafilm frame. A key here was to do this process quickly so that the parafilm would be warm and sticky and the cover slip would adhere better. After putting the cover slip, I put a drop of oil and places the slide on the microscope stage.

Once everything was set up and mounted, the actual trapping of the beads occurred. The trick to do this was to get the slide as close as possible to the microscope objective, to the point where it would start compressing. Then, moving the slide away from the objective. When doing this without the second dichroic mirror, the laser itself could be seen. This allowed me to move the slide to get the optical trap to the location where I needed it to be, right before the interface of the water to the cover slip. After this point was I reached, I would not more the stage in the y-direction so that the focal point would remain stable throughout the whole process. Then I moved the camera to make sure the image was in focus. Once everything seemed to be on focus,I put the dichroic mirror back in place, turned on the back light and I trapped the beads. This was done by chasing them around the slide to try to get them into the location where the optical trap was. I could verify that the bead was trapped because moving the slide slowly would make the stuff in the background move while keeping the bead fixed in place.

The optical trap I made was not very strong, since the circles were a bit off being completely concentric. To make it easier for me to trap the beads with the laser, I would start by having the intensity high and then decreasing it before recording the videos. This made it a little hard to know in which range of power the laser was, since I would have to use the photodiode to measure the power after recording the videos. It also happened that the bead would "jump" out of the trap before the recording ended. In this case, I cropped the videos to only analyze the motion of the trapped bead.

A problem I had at the beginning was that the mirrors and the camera had a lot of dirt on them, which was clearly visible on the videos. Using a can of compressed air to blow around and not directly into the optics helped significantly. I wished I would have had access to optical wipes to clean some of the optics, since Kimwipes leave some white residue behind. Another issue was that the lights from the lab would create a lot of background noise on the images collected from the camera. To solve this, I worked in the data collection with the lights off. That improved the image quality significantly. Once I set up the camera to an automatic gain an exposure, I would unselect the "auto" feature so that the gain and exposure parameters would be fixed throughout the experiment. I zoomed in to take the videos, but the software collected the whole screen. When analyzing the videos, this was too much data and would exceed the python notebook's allowed RAM. To solve this, I manually cropped the videos using the free online software "https://online-video-cutter.com/". This was particularly useful because I could not watch the videos as I was recording the data. The lab computer would not play the AVI videos, and neither would my laptop. The only place where I could see them was on Google Drive after they had been uploaded, and because of the size, it would take a while and it usually happened hours after I had recorded the data and left the lab.

4 Data Analysis

The data analysis process was done using a Google Colab python notebook. The whole code is available in Appendix 1.

The first part of the analysis consists in loading the AVI files from Google Drive and then extracting the frames. This changes our video into stacks of images which are easier to analyse. The frame rate of collection was kept consistent at 129 frames per second, so the time from one frame to the next was the same across all of the data. A color map of a single image of a trapped bead can be see in Figure 3.

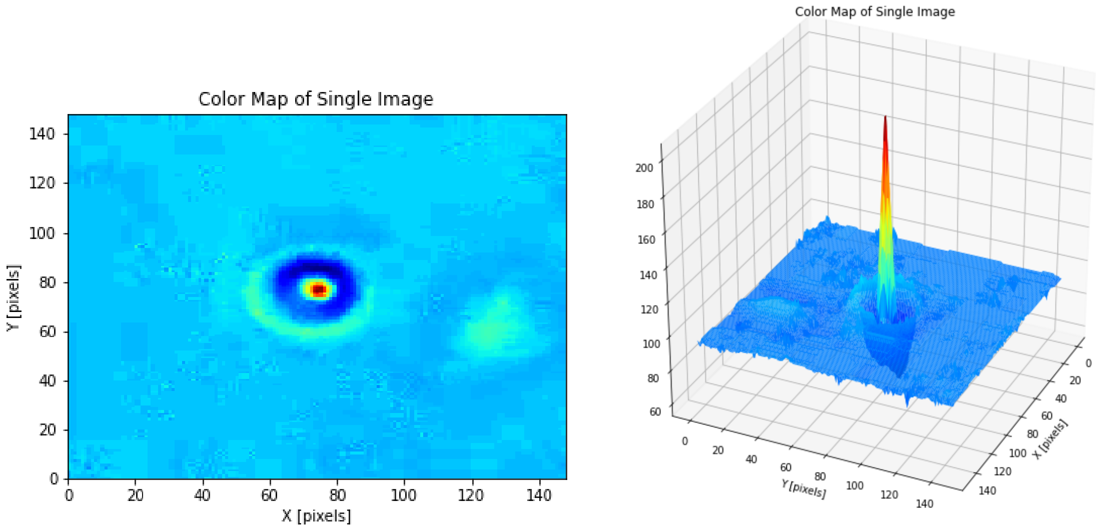

Figure 3: Left: Color map image of a single trapped bead. **Right:** Surface plot of the same bead showing the pixel light intensity across the image.

In these images, it is clear that the center of the bead is the brightest spot on the image. Taking these into account, I wrote the code to go over each one of the frames of the video and find the location of the brightest pixel. Since the accuracy is measured in pixels, the location is quantized. In some cases, more than one pixel would have the maximum brightness. In order to take a single point as the center of the bead, I found the average of the locations of the brightest pixels. Tracking the location from one frame to the next gave me the information about the oscillation of the bead from one frame to the next.

In Figure 4 we can see the location of the beads over the length of the video at different powers of the laser. In a qualitative sense, this plot matches with the hypothesis that at higher powers of the laser, the optical trap would be stronger and the bead would be more localized to a fixed location. The overall movement seems to vary more in the x-axis than the y-axis, which makes sense with the observations of the optical trap, since the circles were a little misaligned in the x-axis. One observation is that the purple dots for the laser with power 4.56mW seems to be too constrained. I believe that in this case the tap was very close to the cover slip from the slide, so the bead was against the glass. This could have caused it to be "better trapped" since there is a force applied by the bead on the glass.

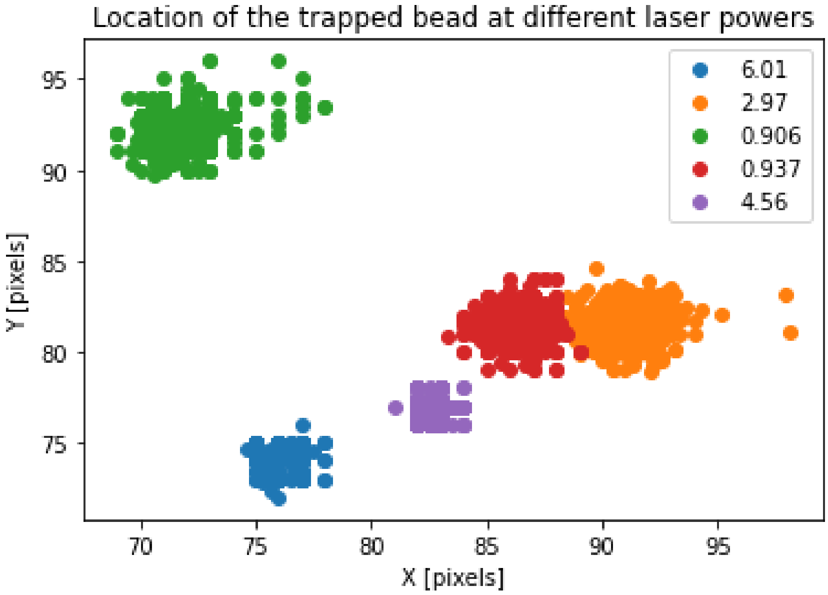

Figure 4: Diagram of the locations of the beads at different powers over the length of the videos. The power of the laser is given in mW.

One thing to note is that the code reads in the information given by the camera in pixels. In order to have this information in sensible units, I needed to do a transformation from pixels to μms. To do this, I estimated the conversion basing myself off the fact that the diameter of the bead was about 1μm and 12 pixels. These is based entirely on what I defined as a bead for the computer and trusting that the diameter of the latex bead is pretty standard. Any errors from these approximation and conversion factor will be propagated throughout the rest of the analysis.

In order to analyze the movement of the particle, I found the difference in position from one frame to the next. I squared this number to ensure that I was getting the absolute value of the oscillation. Then I averaged over these differences to get the average value of the change in position squared. These calculations can be seen in Table 1. The general trend seems to show a decrease in this value as the power of the laser increases.

Laser Power [mW]	$\langle r^2 \rangle$ [μm]
0.906	0.0059729
0.937	0.0047915
2.97	0.0047153
4.56	0.0030844
6.01	0.0040554

Table 1: Average of the absolute value of the oscillations of the bead at different laser powers.

In order to see this trend more clearly, I plotted it and found a linear model that would fit the data. Figure 5 shows this relationship, along with the error bars of the average value of the oscillations. We can see that the general trend of the data shows a decrease in the average value of the oscillation as the power of the laser increases. This supports our hypothesis that the oscillations would decrease as the power of the trap increased. The error bars were calculated with the formula for the standard error, $SE = \frac{\sigma}{\sqrt{N}}$ where σ is the standard deviation and N the number of data points.

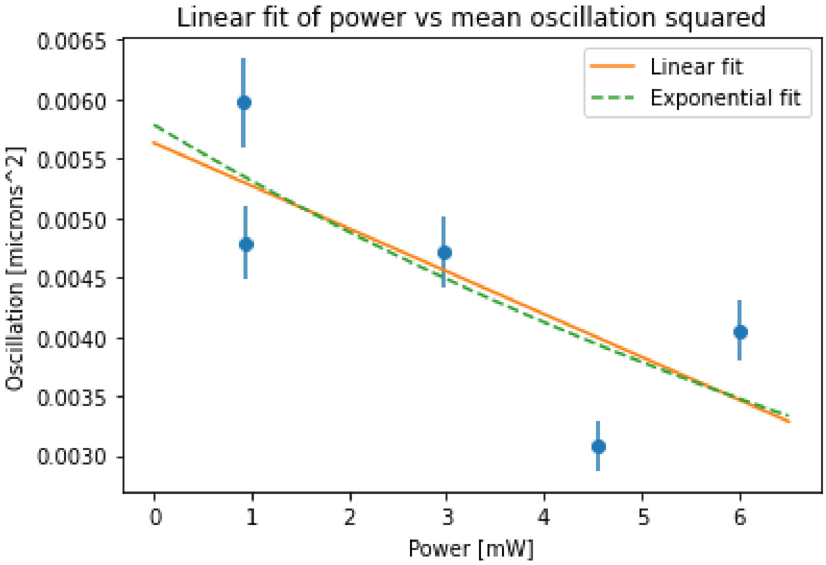

Figure 5: Linear and exponential fits of the absolute value of the oscillation vs power of the laser

From equation (4) we can calculate the spring constant of the trap to be $k_x = \frac{k_b T}{\langle x^2 \rangle}$. Figure 6 shows that there is an increase in the value of the spring constant as the power of the laser is increased. This supports our hypothesis, as a higher spring constant means a stronger optical trap. A high spring constant means that a greater force needs to be applied in order to cause a displacement. The beads at a higher power would need a stronger force in order to oscillate. Because of this, we see a reduction in the Brownian motion as the power is increased.

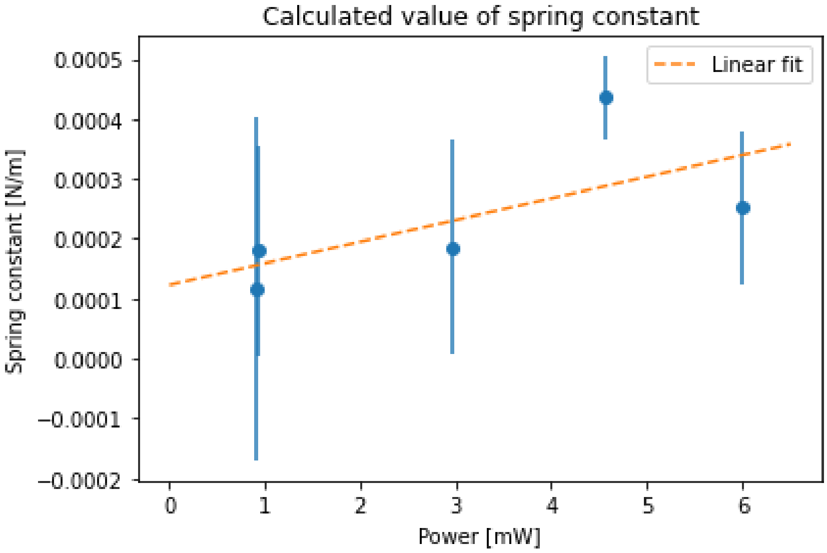

Figure 6: Calculation of the spring constant in pN/micron

Plotting a histogram of the probability distribution of the location of the beads shows a Gaussian trend which we can see in Figure 7. This shows that our assumption that the potential of the optical trap is parabolic holds true.

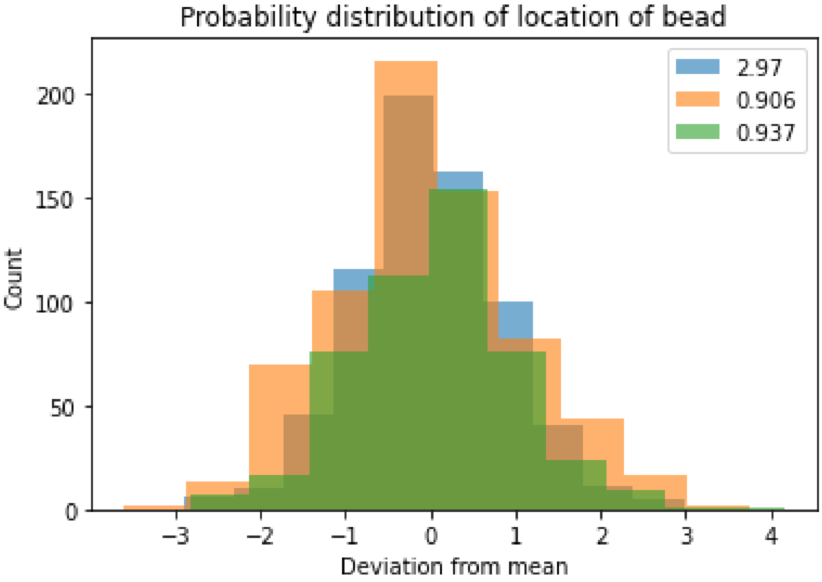

Figure 7: Distribution histogram of the oscillation of the bead at different laser powers in mW

5 Conclusion

The results of this experiment confirmed my initial theory. The collected data shows that the power is inversely related to the square of the average oscillation, $P \propto 1/x^2$. I also saw that the power increased linearly as a function of the spring constant, $P \propto k_x$. I was able to observe the Brownian motion of the latex beads, and the decrease in this motion when the bead was optically trapped. I was able to track the motion of the bead over time to study its motion. I was able to calculate the standard deviation of the oscillations and use this value to calculate the standard error of my measurements. Using these values, I was able to calculate the error propagation by using the techniques and formulas that we have studied in the lectures.

There are several sources of error that have to be considered in this experiment. The first one was that the alignment of the trap was not perfect, which we could observe in Figure 4, because the movement seems more pronounced in the x axis than in the y axis. I think this affected the experiment, as I needed a higher power in order to keep the bead trapped. I would have liked to have a better optical alignment so that I could have measured the oscillations at lower values of the power. In this experiment, when I would lower the power to less than about 900 μW, the bead would stop being trapped. Starting by trapping the bead with a high power and then decreasing it seemed to be the only way to get to these values, but it did not work to go to lower values. Another possible source of error was the photodiode used to measure the power of the laser, as sometimes it was hard to know if the value was correct and measuring this value would free the trapped bead. Another possible error was focusing the laser trap too close to the cover slip of the slide. I think this was the case of the measurement for 4.56 mW, as it has some deviation from the other values and we observe that it barely moves on Figure 4 compared to the other measurements. Since the XYZ stage where the slide was placed would shift in z when moved along the xy plane, it was hard to know exactly if the trap was too close to the cover slip.

In future experiments I would want to use a different technique to make a frame for my slide. The parafilm method worked some times, but a lot of the times the cover slip would fall during the experiment. When aligning the optics, Dean Hudek told me to press the microscope lens to contract it, but many times this would cause the cover slip to get attached to the objective and ruin the alignment and slide. I think updating the lab computer or software (even though they were up to date) would be useful, as I had no way to see the videos until hours after the data collection has occurred and I was blindly trusting the process. This would have been useful to know which videos worked and which videos didn't and change my method of data collection as I was collecting the data and not days after. Having optic wipes would have been useful, as there was a lot of dust on the optics and lenses and it was hard at the beginning to get clear images until I insisted on having someone clean my setup.

Appendices

1 Python Code

```python
# -*- coding: utf-8 -*-
"""ROSSELLI_OpticalTweezers.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/1TfvBG6xuzvZoRjbBn9dPTM9j0qOKw_aL

# Lab 1: Optical Tweezers

## Alejandra Rosselli Calderon

## Setup:
"""

# Set up Google infrastructure for loading files from Google Drive
!pip install -U -q PyDrive
import os
from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
from google.colab import auth
from google.colab import files
from oauth2client.client import GoogleCredentials

# Authenticate and create the PyDrive client.
auth.authenticate_user()
gauth = GoogleAuth()
gauth.credentials = GoogleCredentials.get_application_default()
gdrive = GoogleDrive(gauth)

import numpy as np

import matplotlib as mpl
import matplotlib.pyplot as plt
from mpl_toolkits import mplot3d # Allow 3D surface plots

import math as mt
import time

```

```python
from sklearn.linear_model import LinearRegression
from scipy.optimize import curve_fit

import statistics as stat

# To save images as a movie
import io
import cv2                      # Video processor
import skimage                    # Image processor
import base64                   # Help convert avi to mp4
from IPython.display import HTML # Play video in colab

"""## Return Frames from Video"""

def getFrame(sec,filename):
    vidcap = cv2.VideoCapture(filename)
    vidcap.set(cv2.CAP_PROP_POS_MSEC,sec*1000)
    hasFrames,image = vidcap.read()
    return hasFrames, np.array(image)

"""## Play AVI in Colab"""

def convert_avi_to_mp4(avi_file_path, output_name):
    os.popen("ffmpeg -i '{input}' -ac 2 -b:v 2000k -c:a aac -c:v libx264 -b:a 160k -vprofile high -bf 0 -strict experimental -f mp4 '{output}.mp4'".format(input = avi_file_path, output = output_name))
    return True

def playvideo(video_filename):
    video = io.open(video_filename, 'r+b').read()
    encoded = base64.b64encode(video)
    return HTML(data='''<video alt="test" controls>
                    <source src="data:video/mp4;base64,{0}" type="video/mp4"/>
                 </video>'''.format(encoded.decode('ascii')))

"""## Load AVI files from from GDrive and Extract Frames"""

# Create list of .avi file ID's. File IDs should look something like '1BMZjjmLnQRydIBGqQjEkRuHRuKjsX24J'

#Gain 200
```

```
#Exposure 3648
AVI_FILE_ID_LIST = [
                    ’1QGSomPxCI27t1nlVLSfUG8kRIbUcMEzJ’,
                    ’1x7Br-hdwNTNBHemCx_XPCCPBygZsIpmk’,
                    ’1n0F2x5bAoBN3hoXUib6WmH26O-r-5R82’,
                    ’1u32Xt8S1XtoT07-4WFFRFhiann4asJrW’,
                    ’1btv5Bjvl8KIIv4IjfLAToq4Q3YrDBovc’
                   ]

# choose a local (colab) directory to store the data.
colab_folder_path = "/content/"
local_download_path = os.path.expanduser(colab_folder_path)
try:
    os.makedirs(local_download_path)
except: pass


# Initialize frames[# videos, #frames in video, #pixel height, #pixel width]
# This method can be used if all the files you have are of the same dimensions
# frames = np.zeros([len(AVI_FILE_ID_LIST)
#                     ,1000 # #frames
#                     ,130 # pixel height
#                     ,120 # pixel width
#                    ])

# Initialize frames for datasets that do not have the same dimensions
frames = []


# Initialize frameRates (1/fps) for each avi file
# User should manually input the recorded framerates to output the correct number of frames
frameRate_list = 1/np.array([129,
                             129,
                             129,
                             129,
                             129
                             #1241,
                             #76
                             ])
```

```


# Loop through all .tif files and create data array
for file_index in range(len(AVI_FILE_ID_LIST)):
    # This opens the file. It doesn't create a new file
    opened_drive_file = gdrive.CreateFile({'id': AVI_FILE_ID_LIST[file_index]}) # File Instance
    file_path = os.path.join(local_download_path, opened_drive_file['title'])
    opened_drive_file.GetContentFile(file_path)

    frames_in_image = []
    # Convert video to frames
    sec = 0
    frame_number = 0
    frameRate = frameRate_list[file_index] # frame rate (1/fps)
    success, image = getFrame(sec,file_path)

    while success:
#         frames[file_index, frame_number] = image[:,:,0]
        frames_in_image.append(image[:,:,0])
        frame_number += 1
        sec += frameRate
        success, image = getFrame(sec,file_path)
    frames.append(np.array(frames_in_image))

    print('Title: {:s}, id: {:s}'.format(opened_drive_file['title']
                                         ,AVI_FILE_ID_LIST[file_index]))
    print('     Frames in file = {:d}'.format(np.shape(frames[file_index])[0]) )
    print('     Images have dimension = h{:d} x w{:d}\n'.format(np.shape(frames[file_index])[1],np.shape(frames[file_index])[2]) )

frames = np.array(frames)

"""## Look at a frame"""

# frames[ file_index][frame_index , h_index , w_index ]
print( type(frames) )
print( frames.shape , frames[0].shape)

```

```
# Pick a specific file and frame
file_index = 0 # Select a file
frame_index = 25 # Select a frame in the file

fig,ax = plt.subplots( figsize=(6,4.5) )
plt.pcolormesh( frames[ file_index ][ frame_index , : , : ] , cmap='jet' );
plt.title( 'Color Map of Single Image' );
plt.xlabel( 'X [pixels]' );
plt.ylabel( 'Y [pixels]' );

file_index = 0

fig = plt.figure( figsize=(10,10) )
ax = plt.axes(projection='3d')
X, Y = np.meshgrid( np.arange(frames[file_index].shape[2]) , np.arange(frames[file_index].shape[1]) )

surf = ax.plot_surface( X , Y , frames[file_index][20,:,:] , rstride=1, cstride=1, cmap='jet', edgecolor='none')
ax.view_init(elev=30., azim=25.)

plt.title( 'Color Map of Single Image' );
plt.xlabel( 'X [pixels]' );
plt.ylabel( 'Y [pixels]' );

!ls -a

"""## Play Video in Colab
Convert .avi to .mp4 and play the video in colab.
"""

# Convert avi to mp4
file_index = 0
opened_drive_file = gdrive.CreateFile({'id': AVI_FILE_ID_LIST[file_index]}) # File Instance

filepath = opened_drive_file['title']
#mp4_output_video_name = 'test_video'
#convert_avi_to_mp4(filepath,mp4_output_video_name)
!ls

```

```
# Play mp4 in colab
playvideo(filepath)

"""## Image Analysis

One can see in the original images that the center of the trapped bead is the brightest part of the video. It can also be seen in the colormap, where the center of the bead shows a bright red spike while the rest of the video frame has a dark blue color.

### Find brightest pixel in one image
"""

max = np.amax(frames[0][1])

result = np.where((frames[0][1]) == max)

result

coords = np.append(result[0],(result[1]))

t1 = np.asarray([1,1])

t1.shape

centroid_coords = np.append([coords], [t1], axis=0)

centroid_coords

"""### Find brightest pixel and tracking it"""

# frames[ file_index][frame_index , h_index , w_index ]
coordinates = []
num_coords = []

average = []
variance = []

for file_index in range(frames.shape[0]):

  centroid_coords = []

  #centroid_coords = np.zeros((frames.shape[0], frames[file_index].shape[0]))

  for frame_index in range(frames[file_index].shape[0]):
    max = np.amax(frames[file_index][frame_index])
    location = np.where(frames[file_index][frame_index] == max)
    x_avg = np.mean(location[0])
    y_avg = np.mean(location[1])
    coords = np.append(x_avg, y_avg)

    if frame_index == 0:
      centroid_coords= coords
      centroid_num = x_avg+1j*y_avg
    else:
      #centroid_coords = np.append([centroid_coords], [coords], axis=0)
      centroid_coords = np.vstack([centroid_coords, coords])
      centroid_num = np.append(centroid_num, (x_avg+1j*y_avg))

  var = stat.variance(abs(centroid_num))
  avg = np.mean(abs(centroid_num))

  coordinates.append(centroid_coords)
  num_coords.append(centroid_num)

  average.append(avg)
  variance.append(var)

laser_power = np.asarray((6.01, 2.97, 0.906, 0.937, 4.56))

for file_index in range(frames.shape[0]):

  plt.plot(np.real(num_coords[file_index]), np.imag(num_coords[file_index] ),’o’, label=str(laser_power[file_index]))


plt.title( ’Location of the trapped bead at different laser powers’ );
plt.xlabel( ’X [pixels]’ );
plt.ylabel( ’Y [pixels]’ );
plt.legend()
plt.show()

variance
```

```

oscillation = []

avg_oscillation = []
stdev_oscillation = []

SE = []

for file_index in range(frames.shape[0]):
  difference = np.diff(np.abs(num_coords[file_index]))

  oscillation.append(difference)
  avg_oscillation.append(np.mean(np.abs(oscillation[file_index])/12**2))
  stdev_oscillation.append(stat.stdev(oscillation[file_index])/12)
  standard_e = stat.stdev(oscillation[file_index])/(np.sqrt(frames[0].shape[0])*12**2)
  SE.append(standard_e)

print(’average absolute value of the oscillation in microns’)
for file_index in range(frames.shape[0]):
  v = np.mean(np.abs(oscillation[file_index]))
  v = v/(12**2)

  print(v)

SE

avg_oscillation

"""## Plotting and Fitting the Data"""

#plt.plot(laser_power, avg_oscillation, ’o’)
m, b = np.polyfit(laser_power, avg_oscillation, 1)

x = np.sort(laser_power)
y = np.sort(avg_oscillation)
stdev = np.sort(stdev_oscillation)



x_new = np.linspace(0, np.max(laser_power)+0.5, 100)
```

```
y_new1 = m*x_new + b
#y_new2 = np.exp(p*x_new) + np.exp(q)

def exponential(x, a, b):
    return a*np.exp(b*x)

pars, cov = curve_fit(f=exponential, xdata=laser_power, ydata=avg_oscillation, p0=[0, 0.0065], bounds=(-np.inf, np.inf))

#mpl.pyplot.errorbar(x, y, yerr=stdev, fmt=’o’)
mpl.pyplot.errorbar(laser_power, avg_oscillation, yerr=SE, fmt=’o’)

plt.title( ’Linear fit of power vs mean oscillation squared’ );
plt.xlabel( ’Power [mW]’ );
plt.ylabel( ’Oscillation [microns^2]’ );

plt.plot(x_new, y_new1, label=’Linear fit’)
plt.plot(x_new, exponential(x_new, *pars), linestyle=’--’, label=’Exponential fit’)
plt.legend()
plt.show()

plt.loglog(laser_power, avg_oscillation, ’o’)

p, q = np.polyfit(np.log(laser_power), np.log(avg_oscillation), 1)

p

q

"""## Kx Calculation"""

kb = (1.38064852 * 10**-23)

Kx = []
SE_kx = []

for file_index in range(frames.shape[0]):
  kx = 300*kb/((avg_oscillation[file_index]*10**-6)**2)
  se_kx = 1/(2*(avg_oscillation[file_index])**-3*(SE[file_index]))


  Kx.append(kx)
  SE_kx.append(se_kx)

m, b = np.polyfit(laser_power, Kx, 1)
y_new3 = m*x_new + b



Kx

SE_kx

plt.title( ’Calculated value of spring constant’ );
plt.xlabel( ’Power [mW]’ );
plt.ylabel( ’Spring constant [N/m]’ );

#plt.plot(laser_power, Kx, ’o’)
mpl.pyplot.errorbar(laser_power, Kx, yerr=SE_kx, fmt=’o’)
plt.plot(x_new, y_new3, ’--’, label=’Linear fit’)

plt.legend()
plt.show()

np.histogram(np.abs(oscillation[3]), 10)

#plt.hist(oscillation[0], bins = 10, alpha=0.6)
plt.hist(oscillation[1], bins = 10, alpha=0.6, label=str(laser_power[1]))
plt.hist(oscillation[2], bins = 10, alpha=0.6, label=str(laser_power[2]))
plt.hist(oscillation[3], bins = 10, alpha=0.6, label=str(laser_power[3]))
#plt.hist(oscillation[4], bins = 10, alpha=0.6)

plt.title( ’Probability distribution of location of bead’);
plt.xlabel( ’Deviation from mean’ );
plt.ylabel( ’Count’ );

plt.legend()
plt.show()
```

Bibliography

[1] Anne Marie Helmenstine. An introduction to brownian motion. *ThoughtCo*, 2019.

[2] Block Lab at Stanford University. An introduction to optical tweezers. 2020.

[3] John Bechhoefera and Scott Wilson. Faster, cheaper, safer optical tweezers for the undergraduate laboratory. *American Journal of Physics*, 70(393), 2002.

[4] Anne L. Brody Benjamin L. Brown Edward K. Boyda Stephen P. Smith, Sameer R. Bhalotra and Mara Prentiss. Inexpensive optical tweezers for undergraduate laboratories. *American Journal of Physics*, 67(26), 1999.