

Simulation of light scattering from Brownian particles

Raymond Mullen
Dept. Physics, Cal Poly San Luis Obispo
March, 2015
© 2015 Raymond Mullen

Abstract

This project is a computational exploration of the light scattering from Brownian particles. We simulate laser light scattering to produce fluctuating light intensity at a detector located in the far-field. By analyzing the statistical properties of this intensity fluctuation, and knowing physical parameters of the system, we can deduce the size of the scattering particles. The computer simulation is in excellent agreement with theory.

Introduction

When viewed up close, microscopic particles in a fluid exhibit random motion due to collisions with molecules in the fluid. This is called Brownian motion, after the botanist Robert Brown who observed pollen granules, about five microns in diameter, under the microscope in motion that could not be explained by the currents or evaporation of the fluid [Mazo 2002]. When laser light is incident upon Brownian particles, it is scattered by each of them and in all directions. As the particles move in time, the intensity of the scattered light at some point also changes.

Smaller particles undergo more rapid Brownian motion, while larger particles move slower, so the scattered light intensity exhibits fluctuations depending on the particle size. By recording the intensity over time data, this fluctuation can be quantified using the autocorrelation function.

The parameters of the autocorrelation function can be used to determine the size of the original scattering particles.

In our simulations we initially put the Brownian particles at random positions within a small area (typically in this project, a cube of side 8mm). We picked a coordinate system whose origin is at the center of the cube. The detector is a distance L away from the origin lying on the z -axis. This layout is shown in Figure 1. Point A is an example source particle and B is the location of the detector. We are simulating the effect when the plane wave of laser light travels in the x -direction and scatters off the particles in our cube.

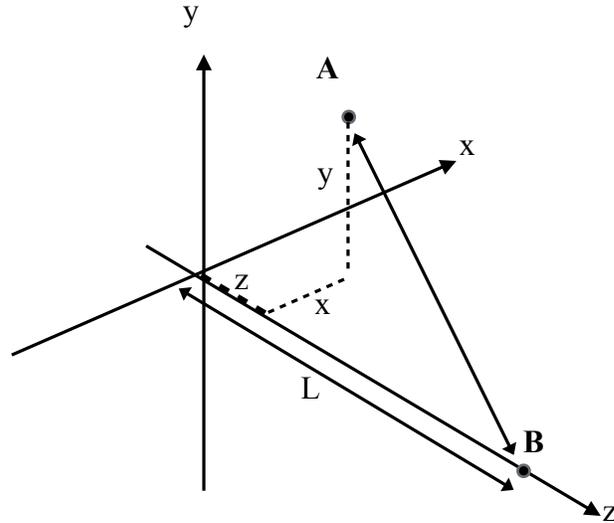


Figure 1. Sketch of the setup we are simulating. Light waves incoming in the x direction scatter off particle A and are recorded at B. All the source particles are constrained in an 8 mm cube whose center is the origin of the coordinate system.

In order to characterize the fluctuations of the light intensity we employ the autocorrelation function which quantifies the degree of correlation between successive values of the light intensity as time progresses. Figure 2 illustrates how the correlation function is calculated. First, one takes the signal of interest (in this case the time-varying light intensity) and, at least conceptually, makes a copy of this signal. Then one can take corresponding points, or points separated by some fixed time, and multiply them together and take their average. The sets of lines in the figure indicate which data points are being sampled and how they depend on the so-called lag time. For the first set (i), the lag time is zero. The same t -value is sampled on both plots. For the second set (ii), there is a small lag time, so for some position t on the first plot the position on the second plot that is sampled is $t + 0.05$ s. Finally, for the last set of lines (iii), the lag time is increased to 0.15 s. For the first drawn line, while the point is in a trough on the top plot, the time lagged point is on a hill in the second plot. As the lag time increases the

correlation between the plots decreases. The rate at which these signals become uncorrelated is what we are measuring. A mathematical statement of the autocorrelation is

$$R(m) = \frac{1}{N} \sum_{i=1}^{N-m} x_i x_{i+m} \quad (1)$$

where N is the total number of time steps and m is the lag.

Turning now to the theory of light scattering [Johnson and Gabriel1981] it can be shown that the second order autocorrelation function of the intensity of the scattered light is

$$g_2(\tau) = e^{-2\Gamma\tau} \quad (2)$$

where τ is the lag time and Γ is a decay constant given by

$$\Gamma = Dq^2 \quad (3)$$

where D is the diffusion coefficient given by the Stokes-Einstein equation:

$$D = \frac{k_B T}{6\pi\eta r} \quad (4)$$

k_B is Boltzmann's constant, T is the temperature, μ is the viscosity of the fluid, and r is the radius of the particle. q is the magnitude of the scattering wave vector given by:

$$q = \frac{4\pi n}{\lambda} \sin\left(\frac{\theta}{2}\right) \quad (5)$$

where n is the index of refraction of the fluid, λ is the wavelength of the incident light, and θ is the scattering angle [Johnson and Gabriel 1981]. For all our simulations the incident light is traveling in the x-direction and light is scattered in the z-direction so that $\theta = 90^\circ$.

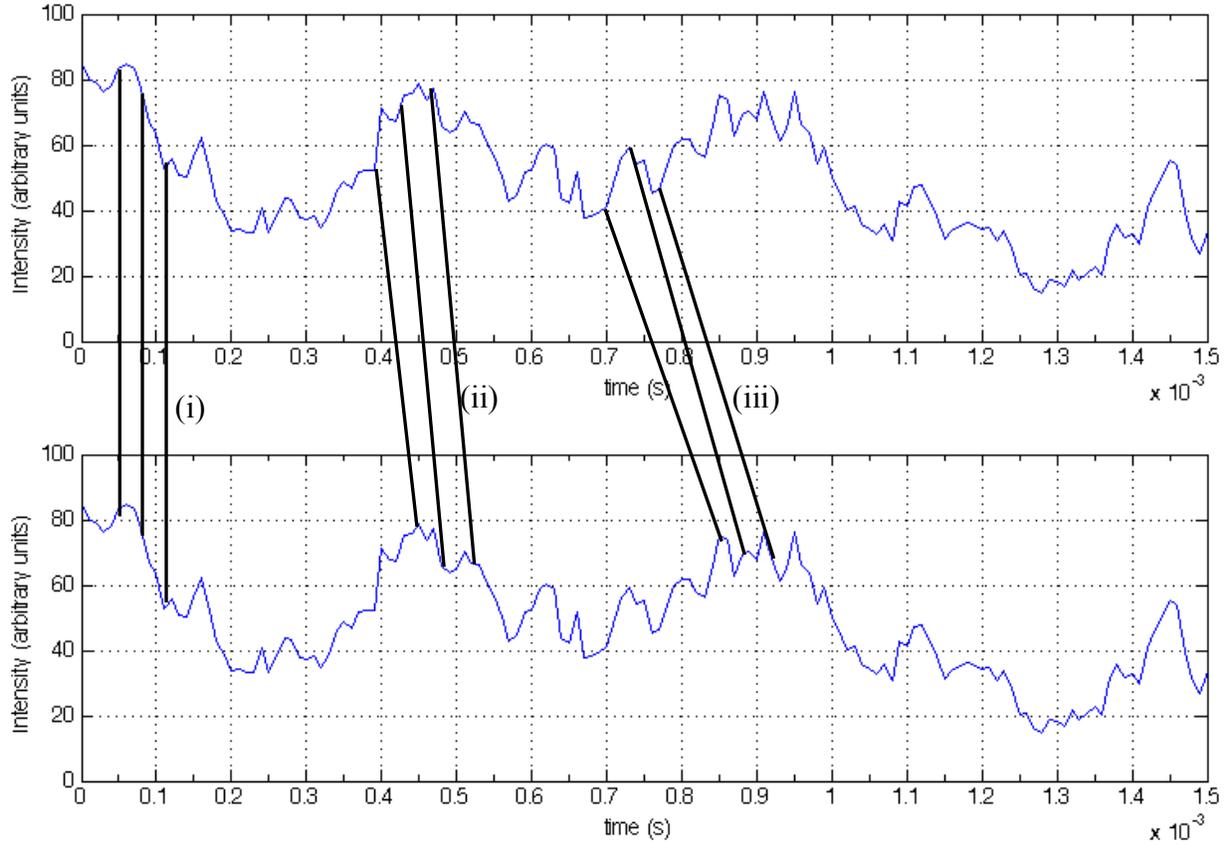


Figure 2. Two identical intensity plots used to demonstrate the autocorrelation function sampling for different lag times. (i) lag time = 0; (ii) lag time = 0.05 s; (iii) lag time = 0.15 s

Simulation of Brownian Particles

The particles suspended in the fluid diffuse through the liquid due to the collisions with other molecules, a process called Brownian motion. The motion of a single particle in one dimension can be described by the following equation:

$$x_i = x_{i-1} + \sqrt{2D \Delta t} w_i \quad (6)$$

where its new position x_i is dependent on its previous position x_{i-1} . D is the diffusion coefficient from equation 4, Δt is the time between the steps of the particle, and w_i is a normally distributed random number with zero mean and unit variance [Volpe 2013]. In a Cartesian system, the motion of a particle in three dimensions is determined by applying equation 6 to each coordinate,

independently of the others. Figure 3 shows the positions of one particle, of radius $r = 0.8 \mu\text{m}$, as it diffuses through water with viscosity $\eta = 0.001 \text{ Nsm}^{-2}$ and temperature $T = 293 \text{ K}$. The particle took 10,000 steps with $\Delta t = 0.01 \text{ ms}$.

In order to evaluate how well equation 6 models the diffusive motion of the particles, we simulated the motion of 500 particles in one dimension. Figure 4 shows the position of the particles at each time step with all particles having an initial position of zero. Each particle takes a specific route and ends at some distance away from the origin. Our plot used particles of radius $1 \mu\text{m}$, in fluid of viscosity 0.001 Ns/m^2 , and at temperature 300 K . Brownian particles should diffuse away from the origin and end up distributed with standard deviation given by:

$$SD = \sqrt{2Dt} \quad (7)$$

where D is the diffusion coefficient and t is the time. Figure 5 shows a histogram of the positions at $t = 0.1 \text{ s}$; it follows the expected distribution. At $t = 0.1 \text{ s}$, the theoretical standard deviation is $2.096 \times 10^{-7} \text{ m}$. The calculated standard deviation for running the simulation twenty-five times is $(2.086 \pm 0.074) \times 10^{-7} \text{ m}$. The theoretical standard deviation is within the uncertainty. As the plots and calculations show, the particles are mostly concentrated within ± 1 standard deviation. Code snippet 1 in the appendix shows the script that produced these plots.

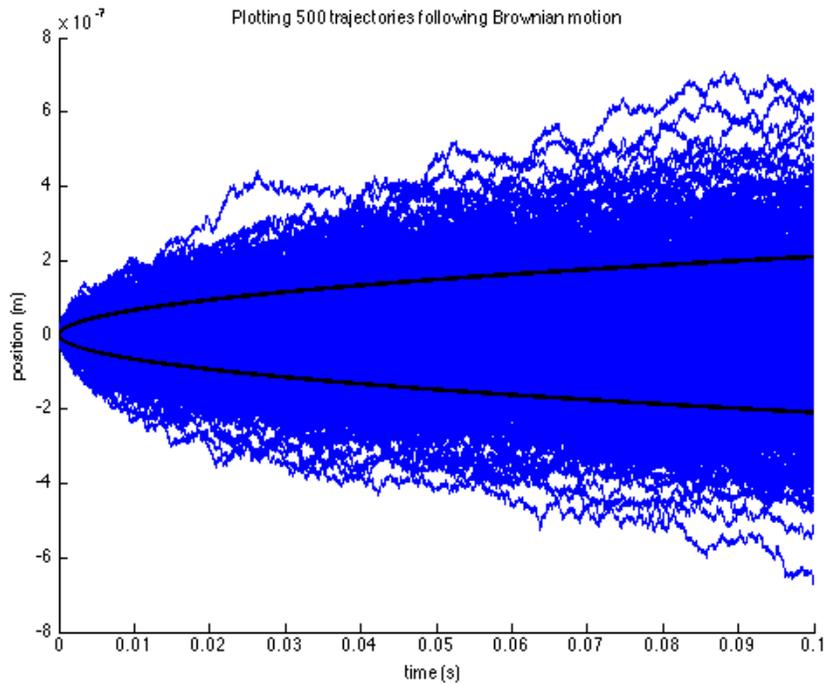


Figure 4. The positions of 500 particles over 10,000 time steps of 0.01 ms each. The plot of one standard deviation of the expected distance is in black. Over 25 runs, the calculated standard deviation of $(2.086 \pm 0.074) \times 10^{-7}$ m agrees with the theoretical standard deviation of 2.096×10^{-7} m.

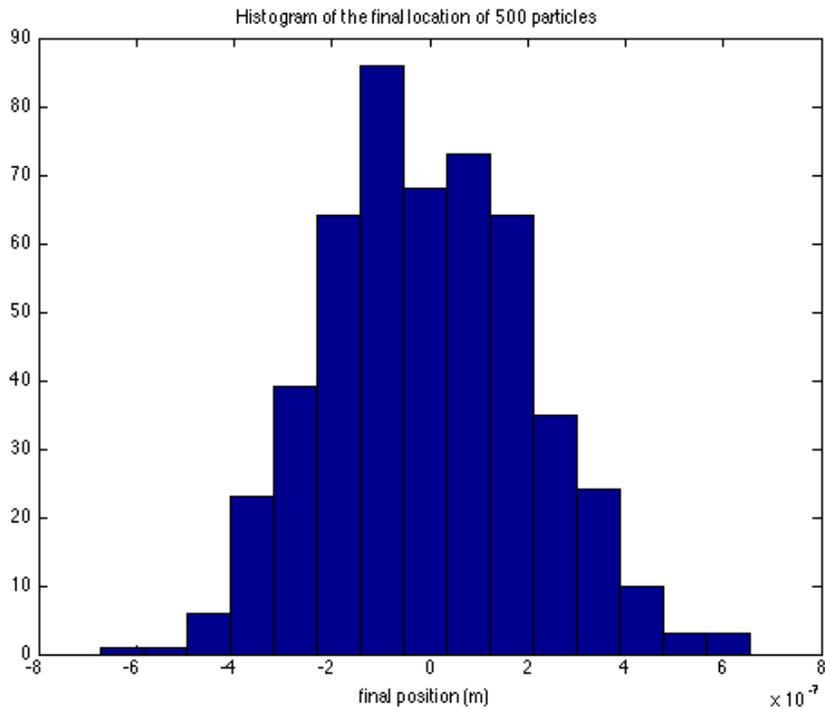


Figure 5. A histogram of the positions of all 500 particles at $t = 0.1$ s.

Simulation of Light Scattering

For scattered light detected in the far field, the electric field of the scattered light from one particle, with respect to some origin, is given by:

$$E_s = E_0 \exp\left(\frac{2i\pi n}{\lambda}(x + z)\right) \quad (8)$$

where n is the index of refraction, i is the square root of -1, λ is the wavelength of the incident light, and x and z are the positions of the scattering particle in the cube. We assume all the particles have the same scattering amplitude so we let $E_0 = 1$. The intensity at the detector is then got by:

$$I = EE^* \quad (9)$$

where E is the complex electric field [Johnson and Gabriel 1981].

The field at the detector can be calculated by summing the contributions from each particle that scatters light. To account for all the particles, we calculate equation (8) for each particle and add the results together to get the total field, E . To get the intensity at the detector, we plug E into equation 9. Figure 6 shows a typical intensity variation at the detector due to light scattering from 50 particles, each with radius 0.2 μm , for 30,000 time steps.

To calculate the autocorrelation of the intensity, we use the built in MATLAB function 'xcorr'. After finding the autocorrelation of the intensity signal, we want to measure the decay rate of that autocorrelated signal. Figure 7 shows the calculated autocorrelation function, the best fit of that curve, and the expected decay curve obtained by plotting equation 2.

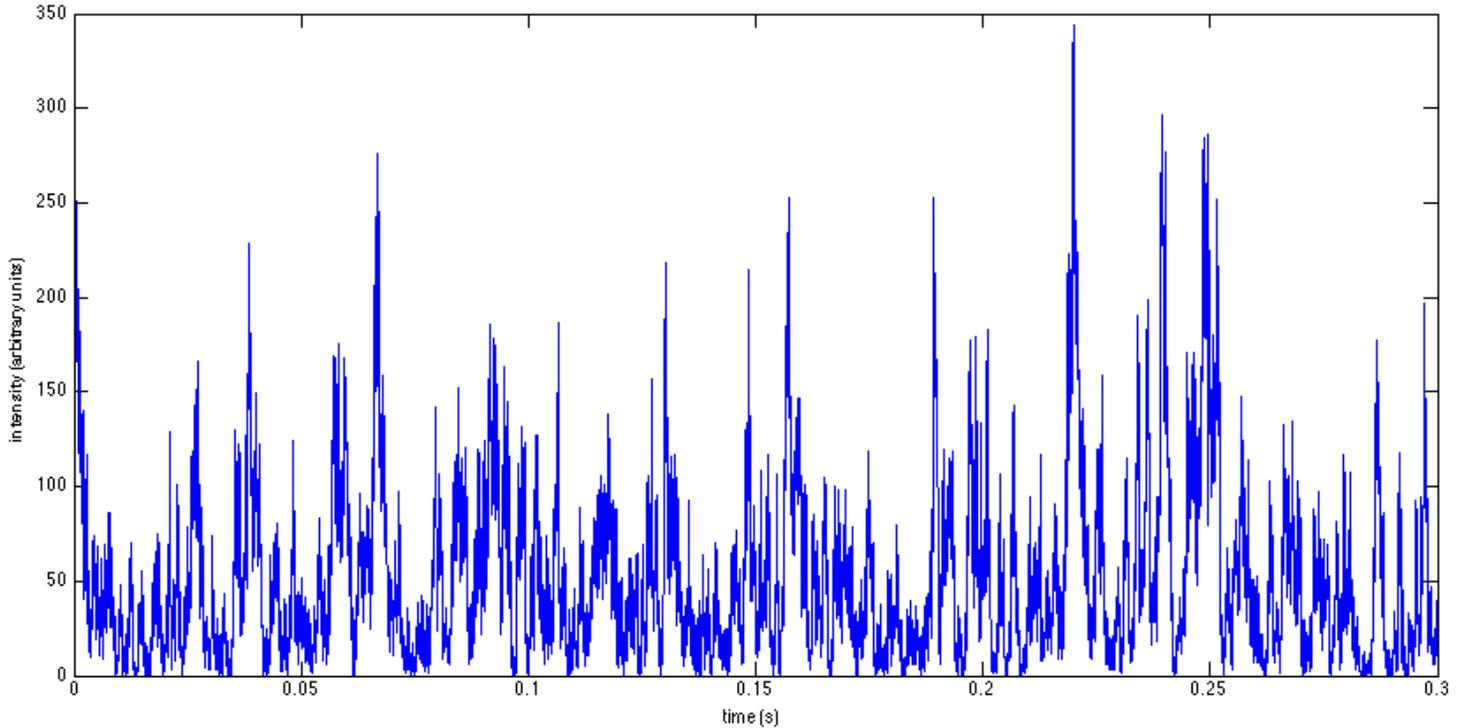


Figure 6. A plot of the intensity of scattered light off of 50 particles of radius $0.2 \mu\text{m}$, taken at the detector, with data recorded at each time step for 30,000 steps.

We fit an exponential curve to the calculated autocorrelation function to obtain a value for Γ .

We can then calculate a value for r , the radius of the scattering particles. Figure 7 shows two decay curves for different sized particles. Plot (a) shows the decay curves for particles of radius $0.2005 \mu\text{m}$. For this case, the particle radius from the fitted curve was calculated to be $0.208 \mu\text{m}$, a 3.75 % error. Plot (b) was generated from particles of radius $0.8 \mu\text{m}$. The radius was found to be $0.812 \mu\text{m}$, a 1.48 % error. Code snippets 2 and 3 in the appendix show the script and function used to generate the plots in figures 5 and 6.

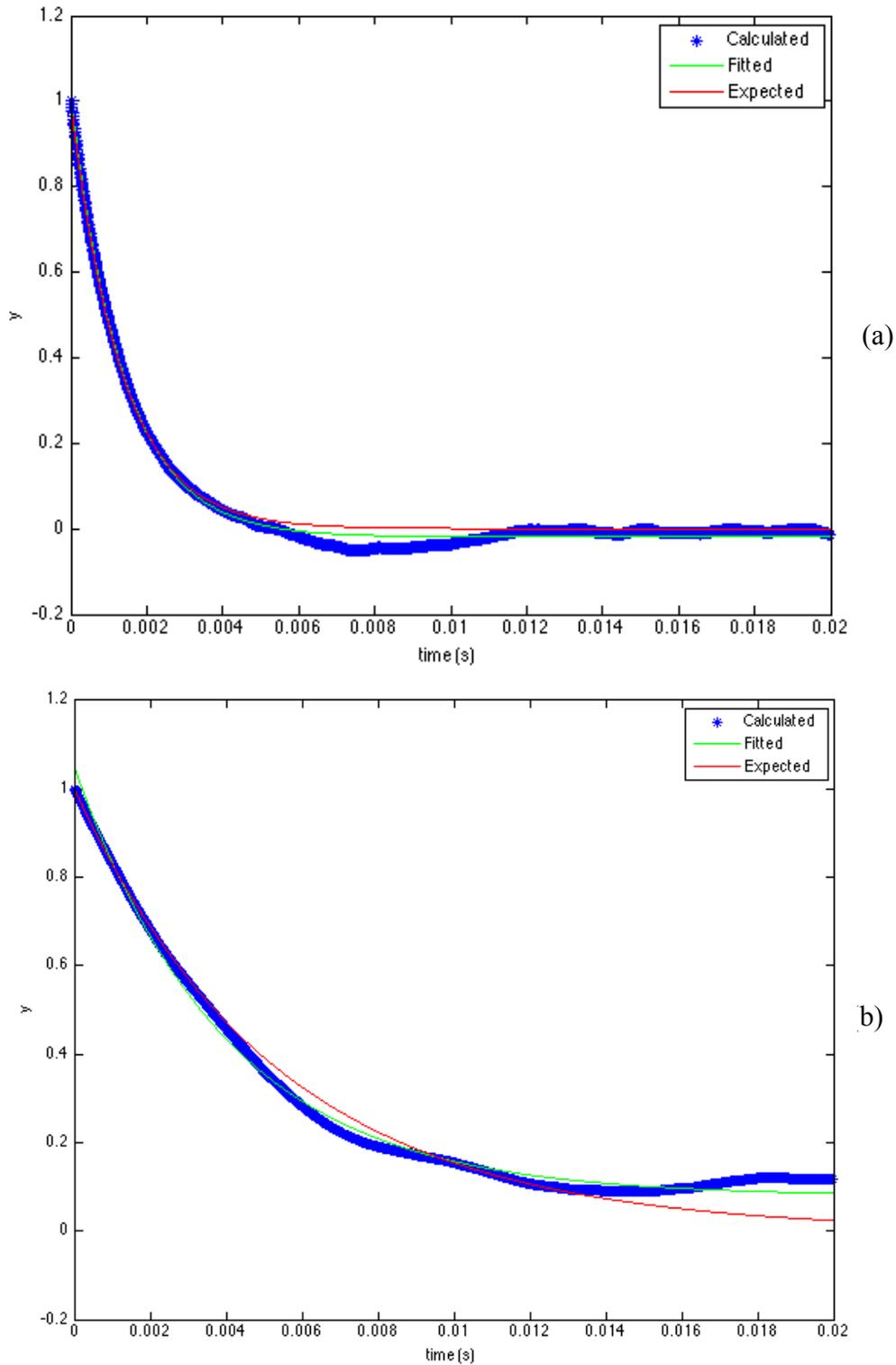


Figure 7. Typical results for a single run showing the data points of the calculated autocorrelation function along with the expected second order autocorrelation function. The exponential fitted to the calculated autocorrelation function can be used to find the radius of the scattering particles. (a) The decay curves when light is scattered off particles with radius $0.2005 \mu\text{m}$. Using the fitted curve, the radius was calculated to be $0.208 \mu\text{m}$. (b) The decay curves when light is scattered off particles with radius $0.8 \mu\text{m}$. From the fitted curve, the radius was calculated to be $0.812 \mu\text{m}$.

Results

From the fit of an exponential to the autocorrelation function, we can obtain a value for the decay constant Γ as defined in equation 3. Along with equations 4 and 5, we can solve for r , the radius of the scattering particles.

Table 1 shows parameters that were constant through all the runs. For each of the four sets of parameters shown in Table 2, twenty runs were made. The fitted radius was calculated for each run. The calculated radius given in Table 2 is an average of the twenty radii and its uncertainty is their standard deviation.

As expected, taking more time steps and gathering a larger data set returns better statistics with a smaller uncertainty. In each case, the actual radius is within the uncertainty of the calculated radius. With a larger particle size the uncertainty is also larger.

tDel (s)	1.0×10^{-5}
wavelength (nm)	633
viscosity (Ns/m ²)	0.001002
scattering angle (deg)	90
refractive index	1.33
temperature (K)	293
number of scattering particles	100

Table 1. These are the parameters used that stayed constant in each run of our simulation.

Time steps	Actual radius (m)	Calculated radius (m)	Average percent error
1000000	2.005×10^7	$(2.004 \pm 0.07) \times 10^7$	-0.04
1000000	8.0×10^7	$(7.865 \pm 0.77) \times 10^7$	-1.69
2000000	2.005×10^7	$(2.011 \pm 0.04) \times 10^7$	0.30
2000000	8.0×10^7	$(8.097 \pm 0.51) \times 10^7$	1.21

Table 2. An overview of our results. Scattered light intensity was simulated for the given number of time steps. Using the autocorrelation function, that intensity data was analyzed and we were able to calculate what particle radius should have generated that intensity data. We were more accurate for smaller particles on longer time scales.

Conclusion

This simulation aimed to model laser light scattering from Brownian particles and use the gathered intensity data to make a measurement of the radius of the original scattering particles.

This simulation aimed to model laser light scattering from Brownian particles and use the gathered intensity data to make a measurement of the radius of the original scattering particles.

We confirmed our method of simulating Brownian particles by looking at the paths of 500 particles in one dimension and checking that their standard deviation matched with the theoretical. By plotting the autocorrelation function of the measured intensity, we were able to fit an exponential curve to obtain a measured value for the decay constant. This was easily solved for the radius of the scattering particles. For the each radii simulated, the actual radius was within the uncertainty of the calculated radius.

References

- Johnson, Charles, and Don Gabriel. *Laser Light Scattering*. New York: Dover Publications, Inc, 1981.
- Mazo, Robert. *Brownian Motion: Fluctuations, Dynamics, and Applications*. Oxford: Oxford University Press, 2002.

Volpe, Giorgio, and Giovanni Volpe. "Simulation of a Brownian particle in an optical trap."

American Journal of Physics 81.3 (2013): 224-230.

Appendix

Code 1: Brownian motion standard deviation check - script

```
clear all
numTraj = 500;           %total number of trajectories to plot
tDel = 1e-5;            %time between steps [s]
timeSteps = 10000;     %number of steps taken

R = 1e-6;               %radius of particle [m]
vis = 0.001;           %viscosity of fluid [Ns/m^2]
gamma = 6*pi*vis*R;    %[kg/s]
kB = 1.38e-23;         %Boltzmann constant [J/K]
T = 300;               %Temperature [K]
D = (kB * T)/gamma;    %[m^2/s]

pos = zeros(1,timeSteps); %holds path of one particle at each time step
final = zeros(1,numTraj); %the final position of each particle

figure(1)
clf
time = 0:tDel:(timeSteps - 1)*tDel;
xlabel('time (s)')
ylabel('position (m)')
title(['Plotting ' num2str(numTraj) ' trajectories following Brownian
motion'])
hold on

for i = 1:numTraj
    for j = 2:timeSteps %the initial position is 0
        pos(j) = pos(j-1) + sqrt(2*D*tDel)*randn();
    end
    final(i) = pos(1,timeSteps);
    plot(time,pos)
end

SD = sqrt(2*D*time); %Expected standard deviation
plot(time,SD,'black',time,-1*SD,'black','LineWidth',2.5)
hold off

figure(2)
hist(final,15)
xlabel('final position (m)')
title(['Histogram of the final location of ' num2str(numTraj) ' particles'])

calcSD = std(final) %standard deviation of recorded data
expectSD = SD(timeSteps) %the expected std deviation at final time
```

Code 2: Brownian motion and autocorrelation - script

```
clear all

wavelength = 633e-9;    %wavelength of the scattering light [m]
numPoints = 50;        %number of sources

timeSteps = 30000;     %number of steps taken
tDel = 1.0e-5;         %time between steps [s]

R = 0.2e-6;            %radius of the particles [m]
vis = 0.001002;        %viscosity of the fluid [Ns/m^2]
gamma = 6*pi*vis*R;    %[kg/s]
kB = 1.3806488e-23;    %Boltzmann constant [J/K]
temp = 293;           %Temperature [K]
th = 90;               %scattering angle [deg]
n = 1.33;              %refractive index of the fluid
D = (kB * temp)/gamma; %[m^2/s]
q = 4*pi*n*sind(th/2)/wavelength; %magnitude of the scattering wave vector
decayC = D * q^2;      %decay constant [1/s]

elapsedTime = timeSteps*tDel;    %[s]

%Generate an array of uniformly distributed random numbers between 0 and 1
%to be used as the initial positions of the particles, one row for each
%coordinate.
ptRandArr = rand(3,numPoints);

%Scale and shift the random numbers so they all lie in an 8mm x 8mm x 8mm
%region
xInitial = 1e-3 - (2e-3) * ptRandArr(1,:);
yInitial = 1e-3 - (2e-3) * ptRandArr(2,:);
zInitial = 1e-3 - (2e-3) * ptRandArr(3,:);

%Concatenate into a 3 by numPoints array with the first row holding the
%x-coordinates, row 2 holds the y-coordinates, and row 3 with the
%z-coordinates
ptArray = [xInitial; yInitial; zInitial];

%An array holding the intensity over time values at a single point
intenTime = zeros(1,timeSteps+1);
%Find the intensity due to the initial positions of the particles
intenTime(1,1) = ACsimFn(ptArray,wavelength,n);

%Will be used to calculate the Brownian motions of the particles
sq = sqrt(2*D*tDel);
```

```

%At each timeStep, move the particles according to the Brownian motion
%equation and calculate the intensity from the new
%arrangement of the particles
for t = 1:timeSteps
    %Array of normally distributed random numbers
    randArr = randn(3,numPoints);
    ptArray(1,:) = ptArray(1,:) + sq.*randArr(1,:); %Updating the x-coord
    ptArray(2,:) = ptArray(2,:) + sq.*randArr(2,:); %Updating the y-coord
    ptArray(3,:) = ptArray(3,:) + sq.*randArr(3,:); %Updating the z-coord
    %Finding the intensity from the new arrangement of
    %the particles
    intenTime(1,t+1) = ACsimFn(ptArray,wavelength,n);
end

%Plot the intensity versus time
figure(1)
tm = 0:tDel:elapsedTime; %Setting a time axis
plot(tm,intenTime,'-')
title('Intensity at One Point on the Screen')
xlabel('time (s)')
ylabel('intensity')

%Plot the autocorrelation function
intenTime = intenTime - mean(intenTime);
figure(3)
[ac,lags] = xcorr(intenTime,'unbiased');
ac=ac/ac(timeSteps);
plot(lags(timeSteps:(timeSteps+2000))*1e-5,ac(timeSteps:(timeSteps+2000)),
'*')
xlim([0 0.02])
title('Autocorrelated')
xlabel('time (s)')

%Calculate and plot the fit to the autocorrelated intensity data
hold on
fitEqn = fittype('A+B*exp(-2*alpha*x)'); %Define the equation to fit to
expFit = fit((lags(timeSteps:(timeSteps+2000))*1e-5)',...
    ac(timeSteps:(timeSteps+2000))',fitEqn,'StartPoint',[0 1 800])
plot(expFit,'g')
fitVal = [coeffvalues(expFit);confint(expFit)];
calcR = (kB * temp * q^2)/(6 * pi * vis * fitVal(1,3))

perErr = (calcR - R)*100/R %Percent error of calculated particle radius

%Plot the expected decay rate of the autocorrelation function
x = 0:tDel:elapsedTime;
pre = exp(-2*decayC*x);
plot(x,pre,'r')
xlim([0 0.02])
title('Expected Exponential')
xlabel('time (s)')
legend('Calculated','Fitted','Expected')
hold off

```

Code 3: Intensity calculation of a single point - function (called by Code 2)

```
function [ ptIntensity ] = ACsimFn(ptArray, wavelength, n)
%Calculate the intensity at a single point far from a region of
%scattering particles.
% ptArray has three rows, corresponding to the x, y, and z corridantes of
% a source point, and a number of columns equal to the number of source
% points.
%
% wavelength is the wavelength of the scattering light, in meters.
%
% n is the index of refraction of the fluid

ii = sqrt(-1);
pha=(2*pi*n)/wavelength*(ptArray(3,:)+ptArray(1,:));
ENet=sum(exp(ii*pha));
ptIntensity = ENet * conj(ENet);

end
```