Biological Physics Problem Set 02

Table of Contents

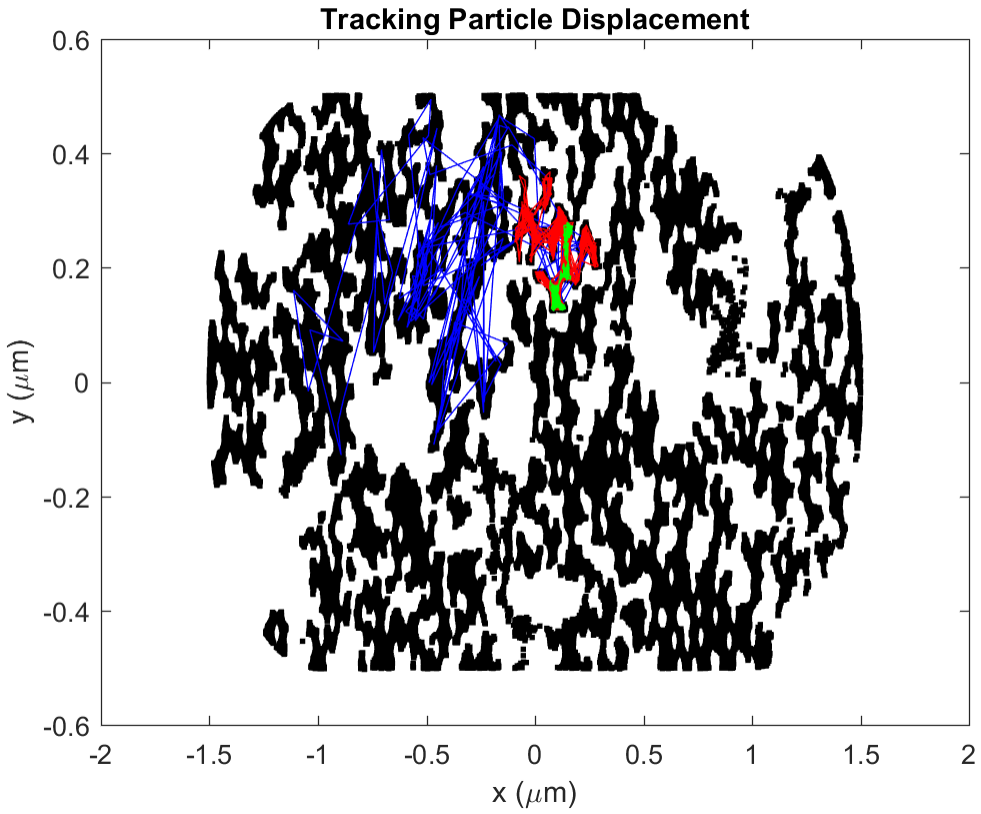
Patrick O'Brien

Due Thurs., Feb. 18, 2016

```
% They are displayed below in their corresponding spots, but for this
 PS I
% calculated a Diffusion coefficient of D = 7.73e-6, c = 7.31e-5, and
% n = 0.7039.
```

First Look at the Data

```
close all
clear


load PS2data

figure(1)

x1 = [locs1us(1:end,1:end,1); locs100us(1:end,1:end,1);
 locs10ms(1:end,1:end,1)];
y1 = [locs1us(1:end,1:end,2); locs100us(1:end,1:end,2);
 locs10ms(1:end,1:end,2)];

plot(x1, y1, 'k.')
hold on
plot(locs10ms(1:end, 1, 1), locs10ms(1:end, 1, 2), 'b-')
plot(locs100us(1:end, 1, 1), locs100us(1:end, 1, 2), 'r-')
plot(locs1us(1:end, 1, 1), locs1us(1:end, 1, 2), 'g-')

xlabel('x (\mum)')
ylabel('y (\mum)')
title('Tracking Particle Displacement')
```



Calculating the MSD

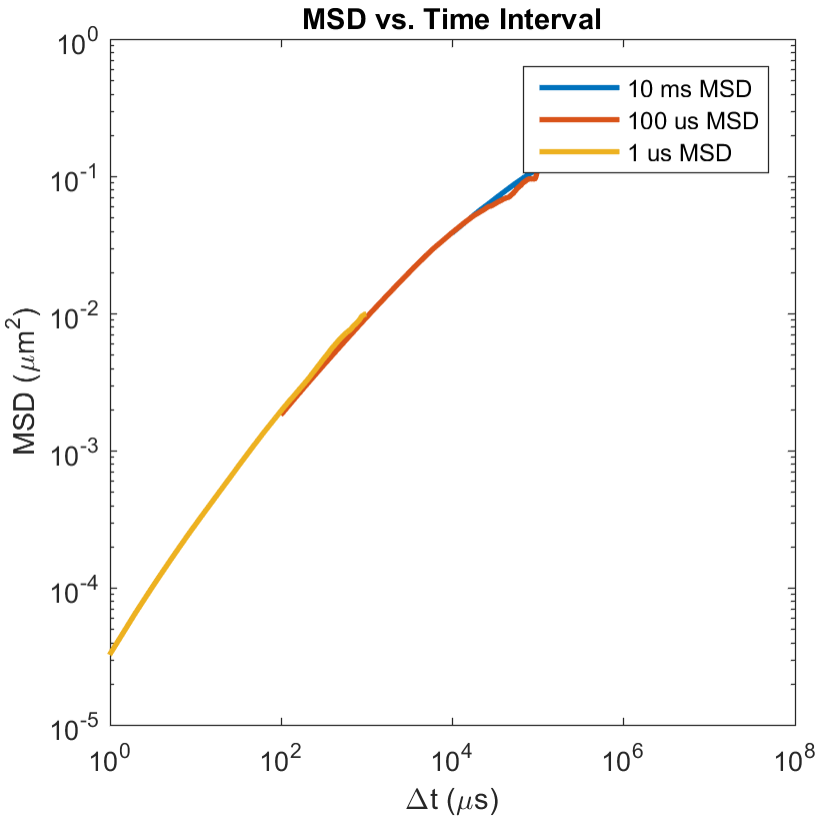
```
% MSD = < | r(t + dt) - r(t)| ^2>

% part 1. Calculate approximately 200 dt values for which we can
 calculate
% the MSD

Nlarge = size(locs100us,1);
Nsmall = size(locs10ms,1);

% create time vector
t10ms = (1:1:Nsmall)*10*1000;
t100us = (1:1:Nlarge)*100;
t1us = (1:1:Nlarge);

% Used more than 200 here to because some points lost from 'unique'.
ixs10ms  = unique(round(logspace(log10(1),log10(Nsmall-1),700)));
disp(length(ixs10ms))
ixs100us = unique(round(logspace(log10(1),log10(Nlarge-1),346)));
%length(ixs100us)
ixs1us = unique(round(logspace(log10(1),log10(Nlarge-1),346)));
%length(ixs1us)

for i=1:length(ixs10ms)-1
```

```
    dt10ms(i) = t10ms(ixs10ms(i+1))-t10ms(ixs10ms(1));
end

for i=1:length(ixs1us)-1
    dt1us(i) = t1us(ixs1us(i+1))-t1us(ixs1us(1));
end

for i=1:length(ixs100us)-1
    dt100us(i) = t100us(ixs100us(i+1))-t100us(ixs100us(1));
end

% part 2. Calculate the MSD for 2 arrays

MSD10ms = zeros(size(ixs10ms));
for j = 1:length(ixs10ms)
    ix = ixs10ms(j);
    drs = locs10ms(1:end-ix, :, :) - locs10ms(1 + ix:end, :, :);
    drs2 = drs(:,:,1).^2+drs(:,:,2).^2;
    MSD10ms(j) = mean(mean(drs2));
end

MSD1us = zeros(size(ixs1us));
for j = 1:length(ixs1us)
    ix = ixs1us(j);
    drs = locs1us(1:end-ix, :, :) - locs1us(1 + ix:end, :, :);
    drs2 = drs(:,:,1).^2+drs(:,:,2).^2;
    MSD1us(j) = mean(mean(drs2));
end

MSD100us = zeros(size(ixs100us));
for j = 1:length(ixs100us)
    ix = ixs100us(j);
    drs = locs100us(1:end-ix, :, :) - locs100us(1 + ix:end, :, :);
    drs2 = drs(:,:,1).^2+drs(:,:,2).^2;
    MSD100us(j) = mean(mean(drs2));
end
```

149

Plot figure 2

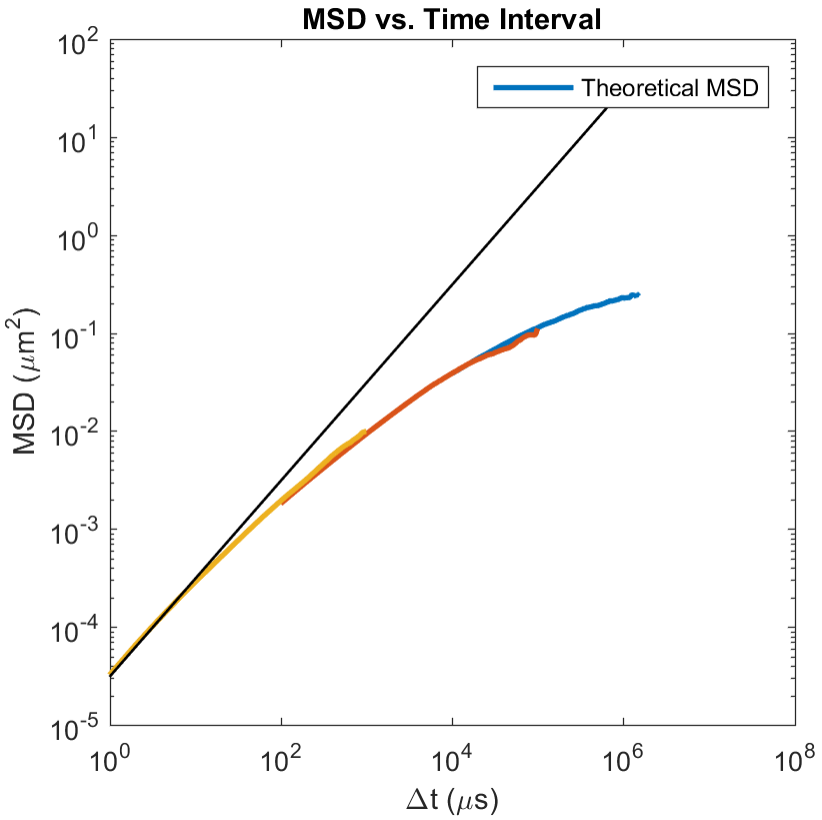
```
figure(2)
loglog(t10ms(ixs10ms),MSD10ms, 'LineWidth', 2)
hold on, axis square
loglog(t100us(ixs100us),MSD100us,'LineWidth',2)
loglog(t1us(ixs1us),MSD1us, 'LineWidth',2)
xlabel('\Deltat (\mus)')
ylabel('MSD (\mum^2)')
legend('10 ms MSD', '100 us MSD', '1 us MSD')
title('MSD vs. Time Interval')
```



C Part I. Calculate the Diffusion Coefficient

```
maximum_limit = 100;

long_MSD = [MSD1us(dt1us < maximum_limit), MSD100us(dt100us <
 maximum_limit*10), MSD10ms];
long_dt = [dt1us(dt1us < maximum_limit), dt100us(dt100us <
 maximum_limit*10), dt10ms];

D = mean(long_MSD(long_dt < 10)./(4*long_dt(long_dt < 10)));
MSD_theory = 4*D.*long_dt;
figure(2)
loglog(long_dt, MSD_theory,'k', 'Linewidth', 1)
legend('Theoretical MSD')
display(D)
```

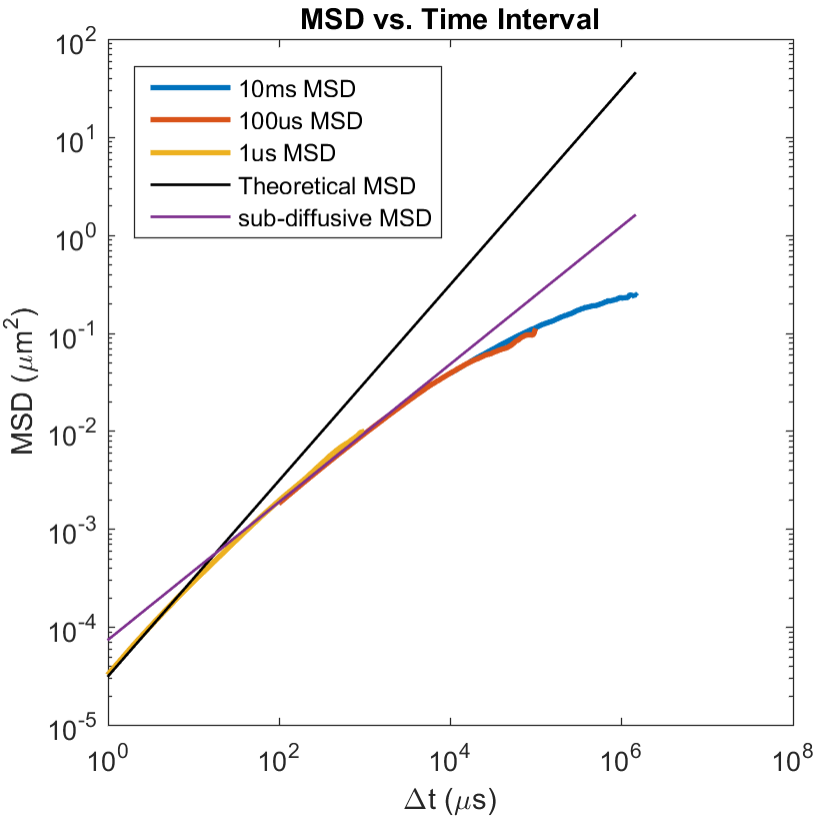
```
D =

   7.7337e-06
```



C Part II. Perculation

```
% n = 0.707;
% We see this behavior between 50 us and 1 ms. Let's find the exponent
% a. create two arrays we can fit to x = dt and y = MSD

x = [dt1us(dt1us>50 & dt1us<100), dt100us(dt100us>100 &
 dt100us<1000)];
y = [MSD1us(dt1us>50 & dt1us<100), MSD100us(dt100us>100 &
 dt100us<1000)];

% b. find parameter with polyfit.
p = polyfit(log10(x), log10(y), 1);
n = p(1);
c = 10.^(p(2));
display(n)
display(c)
MSDt = c*(long_dt.^(n));
figure(2)
loglog(long_dt, MSDt, 'Linewidth', 1)
legend('10ms MSD', '100us MSD','1us MSD','Theoretical MSD', 'sub-
diffusive MSD', 'location', 'northwest')


n =
```

```
    0.7039

c =

    7.3132e-05
```



Published with MATLAB® R2015b