Assignment 0

Integrated Workshop

Due Fri., Sept. 14, 2018 at 9:00 am

Code Structure Requirements

To obtain full credit, prepare your code with the following structure:

1. Your assignment should be callable from one script titled `Main.m`. It may reference other scripts and user defined functions in your working folder.
2. Segment each lettered section of the assignment with a line containing two percentage (%%) signs and the title of the section.
3. Comment thoroughly.

Introduction to Mean Square Displacements

This assignment is designed to reinforce fundamentals of MATLAB covered in the tutorial. You are given two files containing the time series data of a simulation of a 2-dimensional gas of hard disks. You will use this data to calculate the Mean Squared Displacement (MSD) of the particles in the gas at three packing fractions. The MSD is utilized to visualize collective motion and diffusion in a variety of systems. The packing fraction ϕ is defined as the ratio of the area of the particles to that of the box. In 2-D, $\phi = \sum_{i=1}^{N} A_i/A$, where A_i is the area of particle i, N is the number of particles, and A is the area of the box.

A) Read in Data

For this first assignment, suppose a colleague has been taking data for a system of 2-D monodisperse disks at different packing fractions, and they need you to analyze the data. They can tune their system to three different packing fractions, and they want to know the effect of packing fraction on how much each particle moves. As mentioned above, you decide to calculate the MSD of each system, and will use MATLAB to do it. Your colleague has given you the following files:

1. `posXphi0.2.csv`

2. `posYphi0.2.csv`
3. `posXphi0.5.csv`
4. `posYphi0.5.csv`
5. `posXphi0.8.csv`
6. `posYphi0.8.csv`

Each file is a trajectory through time of $N = 100$ particles at 3 different packing fractions ($\phi = 0.2, 0.5$ and 0.8). The first column in each file is the value of time in the experiment (in microseconds), and each column corresponds to a particle position (in microns) in a given dimension. Thus, if there are N_t points in time, each file has $N_t \times N + 1$ entries. Because the experiment runs in 2-D, we used separate files for X values and Y values (as indicated by the file names).

In this section, we'll read in the data for the files and put it into appropriate variables that we will use to process the data in subsequent sections.

1. Download the `.csv` files contained in this folder and place them into the folder in which you are writing your MATLAB script. Remember, if the files are somewhere else, MATLAB won't be able to read in the data.
2. Set your MATLAB working directory to this folder.
3. Open a new script. In the new script, use the `csvread` function to load the .csv files to the workspace. **Note:** `.csv` files are "comma separated value" files, which means that each column and row can be interpreted as distinct entities. You can use MATLAB or even Microsoft Excel to view a `.csv` file.
   (a) One way to do this is use string variables for the file names, and then use those variables as inputs to the `csvread` function. Store the output of the `csvread` function into individual variables. So you should have something like

   ```
   [your input variable] = 'posXphi0.2.csv';
   [your output variable] = csvread([your input variable]);
   ```

   for each file.
4. Put the x and y data into 3-D arrays called `x` and `y`. Each row should correspond to a point in time, and each column should correspond to a particle label. Each "stack" (accessed by the third element in the array) corresponds to a different packing fraction. Here's how to do it:
   (a) Because each data array from the `.csv` files have both time and position data, we need to extract the important information. First, extract the time values (first column in the arrays) from the experimental data and save them into new variables.

(b) Now extract the position data, and save them into two variables `x` and `y`.
To do this, we can use a 3-D array (or a rank 3 tensor). Remember that a 3-D array is an array of 2-D matrices, just as a 2-D matrix is a vector of 1-D vectors (and a vector is a set of scalars). So take the 2-D matrix from your input data variables and place it in the appropriate 2-D matrices contained in the 3-D arrays `x` and `y`.
Hint: If your `x` data for the first experiment is stored in an array called `dataX1`, then this should look something like

```
x(:,:,1) = dataX1(:,2:end);
```

where we have used `2:end` because all the position info is listed after the first column.

(c) Make sure you put in the x and y data from all three experiments into `x` and `y`.

B) Calculate the MSD

The MSD is the Mean Squared Displacement of the particles, plotted as a function of lag times, Δt:

$$MSD(\Delta t) = \langle |\vec{r}_i(t+\Delta t) - \vec{r}_i(t)|^2 \rangle_{t,i},$$

where $\vec{r}_i(t)$ is the position of particle i at time t. Equivalently, in 2-D, the MSD is defined as:

$$MSD(\Delta t) = \langle (x(t+\Delta t) - x(t))^2 + (y(t+\Delta t) - y(t))^2 \rangle_{t,i},$$

where x and y are the x and y coordinates of the particles. **Note:** Angled brackets ($\langle\rangle_\mu$), denote an average over some variable μ. For discrete variables, this is defined as:

$$\langle A \rangle_\mu = \frac{1}{N} \sum_{\mu=1}^{N} A_\mu$$

where N is the number of elements in A.

1. Calculate 200 values of Δt with which we will calculate the MSD.

   (a) First, find all of the indices over which you have data. The minimum index for any vector or matrix in MATLAB is 1. Use the `size` function to calculate the maximum index, N_t, of the data.

   (b) The MSD is traditionally plotted over a logarithmic timescale. To plot evenly spaced data, we will want to have logarithmically spaced time data. Create a vector of indices, `inds`, with which to make calculations. Use the `logspace` function to create a vector of 200 evenly spaced numbers from 1 to $N_t - 1$. Since these values will not all be integers, use the `round` function to convert these values to integers and the `unique` function to remove any repeated values in the vector. In this process, the number of indices will be reduced from 200, resulting in $N_{\Delta t}$ roughly logarithmically spaced indices.

(c) Use these indices and the times extracted from the data to assign the resultant Δt values, in seconds, to a vector, `delta_t`.

2. Calculate the MSD for the three packing fractions.

(a) Initialize a matrix of MSDs, `MSD`, using the `zeros` function. This matrix should have $N_{\Delta t}$ rows and 3 columns.

(b) Use a `for` loop to assign values to `MSD`. Here we will be looping over the array of indices (`inds`). Fill in the following code to calculate the MSD:

```
for  i  = 1:length(inds)
    ind = inds(i);
    dxs = x(1:end-ind, :, :) - x(1+ind:end, :, :);
    dys = ...;
    MSD(i,:) = ...;
end
```

Use the `mean(A,dim)` function to take the mean of an array along a dimension. Recall that for the MSD calculation, we will be taking the mean across times (t) and particles (i).

C) Plot the MSD

1. **[Output]** Plot the MSD for the three packing fractions using the `loglog` function. Label your axes, including units, and include a legend. Save the figure as a .png file.

2. **[Output]** What do you notice about the shape or size of the MSD plot? Are there distinct regimes as a function of Δt? If so, what might these correspond to?

3. **[Output]** How does the MSD change as a function of packing fraction (ϕ)? Provide an explanation for the observed behavior.

Submission

Compress your folder to a zip (or tar) file containing all of your code in `.m` format, all your graphs in `.png` format, and the answers to the questions in `.txt`, `.rtf`, or `.pdf` format. Upload to the canvas Assignments section, *named with the format* ***`LASTNAME-FIRSTNAME-0.zip`*** (or `.tar` / `.tgz`).