

Contents

- [Clearing previous data](#)
- [Load data generally](#)
- [Making a video](#)

```
% Code for calculating the time series latest data case 4b
% Code Summary:
% Given the input sizes of the bins, discretize the xyz domain into small
% small bins and create meshing. In a dataarray named bindata of size nbins
% = xbins*ybins*zbins, correctly identify the bin number and calculate the
% properties of the atoms inside it such as stresses - contact and
% streaming and store it correctly. If the for loop is activated and the
% time=6*1e5 is removed, then we can get multiple figures for different
% different timesteps. The commented code at the end creates a video of the
% timestep data accordingly. One issue however is that as the bins are x,
% y, z and radius has been divided as  $r^2 = x^2 + y^2$ , hence The radii will
% also be discrete and probably in a square shape only. Will have to
% analyse that in detail as the obtained plots are not physically correct
% and am getting square type plots instead of radial. There is some kind of
% bias, will have to think over it and figure it out.
```

Clearing previous data

```
clc;
close all;
clear variables;
```

Load data generally

Trying it out with a very general mindset for a single timestep.

```
folder = "C:\IIsc\DEM\Task 2\case4b";
PostProcessing_folder = fullfile(folder, "Binning") ;
if ~exist(PostProcessing_folder, 'dir')
    mkdir(PostProcessing_folder)
end

xb=0.01; yb=0.01; zb=0.01; % dimensions of the bin
xl=-0.052; yl=-0.052; zl=-0.01; % lower limits of the volume under consideration
xu=0.052; yu=0.052; zu=0.12; % upper limits
tbx = floor((xu-xl)/xb); tby = floor((yu-yl)/yb); tbz = floor((zu-zl)/zb); % Num bins along each direction
Vb = xb*yb*zb;
Numbins = tbx*tby*tbz;

tstart = 100000; tstop = 300000; % time dimensions *****
m = 5; % m = streaming stress proportionality constant *****
rho=2500;
xbins = linspace(xl, xu, tbx);
ybins = linspace(yl, yu, tby);
zbins = linspace(zl, zu, tbz);

% for time = 400000:2000:600000 % Was trying out for only 1 timestep, so commented this line out
time = 600000;
name = "post\particles_" + time + ".liggghts";
file = importdata(name, " ", 1);
timestep = file.data;
% file = importdata(name, " ", 3);
% numatoms = file.data;

% if (timestep > tstop)
%     disp("We have crossed the timesteps asked for in the simulation. ")
%     % break; % will break from the time loop
% end

file = importdata(name, " ", 9);
data = file.data;
clear file;

r = data(1, end-1); % radius
x = data(:, 3);
y = data(:, 4);
z = data(:, 5);
```

```

dp = 2*r; % All atoms are of the same type i.e. type1
bindata = zeros(Numbins, 20); % Say, 20 properties will be calculated for each bin

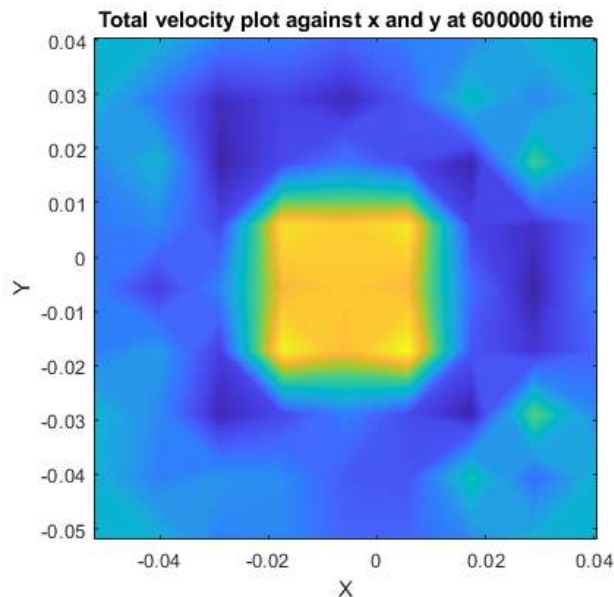
for i=1:1:(tbx-1)
    for j=1:1:(tby-1)
        for k=1:1:(tbz-1)
            binno = (i-1)*tby*tbz + (j-1)*tbz + k;
            a = (x > xbins(i) & x <= xbins(i+1));
            b = (y > ybins(j) & y <= ybins(j+1));
            c = (z > zbins(k) & z <= zbins(k+1));
            rows = a & b;
            rows = rows & c;
            rows = find(data(rows==1, :));
            if (isempty(rows)) % If there are no particles in the bin volume do nothing, already initialised to zero
                % bindata(binno, 1) = 12121; % writing so that will know which have 0 atoms
                bindata(binno, 3) = xbins(i); bindata(binno, 4) = ybins(j); bindata(binno, 5) = zbins(k);
            else
                bindata(binno, 3) = xbins(i); bindata(binno, 4) = ybins(j); bindata(binno, 5) = zbins(k);
                bindata(binno, 6:13) = mean(data(rows, 6:13));
                bindata(binno, 14) = m*mean((data(rows, 6) - bindata(binno, 6)).^2); % xx streaming stress
                bindata(binno, 15) = m*mean((data(rows, 7) - bindata(binno, 7)).^2); % yy streaming stress
                bindata(binno, 16) = m*mean((data(rows, 8) - bindata(binno, 8)).^2); % zz streaming stress
                bindata(binno, 17) = m*mean((data(rows, 6) - bindata(binno, 6)).*(data(rows, 7) - bindata(binno, 7))); % xy streaming stress
                bindata(binno, 18) = m*mean((data(rows, 6) - bindata(binno, 6)).*(data(rows, 8) - bindata(binno, 8))); % xz streaming stress
                bindata(binno, 19) = m*mean((data(rows, 7) - bindata(binno, 7)).*(data(rows, 8) - bindata(binno, 8))); % yz streaming stress
                % Data of contact stresses is not there, else calculate that as well from liggghts and find the mean
                bindata(binno, 20) = size(rows, 1) * (4/3) * pi * r^3 / Vb; % Volume fraction
            end
        end
    end
end

row = find(bindata(:, 5) == min(zbins(zbins>=0))); % This is the obtained ground plane
V = sqrt(bindata(row, 6).^2 + bindata(row, 7).^2 + bindata(row, 8).^2); % velocity along ground plane
Vfluc = ((bindata(row, 6) - mean(bindata(row, 6))).^2 + (bindata(row, 7) - mean(bindata(row, 7))).^2 + (bindata(row, 8) - mean(bindata(row, 8))).^2);
want = [bindata(row, 3), bindata(row, 4), V];

[X, Y] = meshgrid(unique(want(:, 1)), unique(want(:, 2)));
Vre = reshape(Vfluc, size(xbins, 2)-1, size(ybins, 2)-1);
levels = 0:max(max(Vre))/1000:max(max(Vre));
A = contourf(X, Y, Vre', 5, 'LineColor', 'none', 'LevelList', levels);
axis equal
xlabel("X")
ylabel("Y")
str = "Total velocity plot against x and y at " + time + " time";
title(str)

filename= fullfile (folder,"Binning",sprintf("%d.png", time));
saveas(gcf, filename);
% end % time loop

```



```
images    = cell(100,1);
%
% for i=1:100
%     num = 400000 + 2000*i;
%     imname = "Binning\" + num + ".png"; % These are my file names of the
%     images
%     images{i} = imread(imname);
% end
%
% writerObj = VideoWriter('Presentation.avi');
% writerObj.FrameRate = 10;
%
% secsPerImage = ones(100, 1);
%
% open(writerObj);
%
% for u=1:length(images)
%     % convert the image to a frame
%     frame = im2frame(images{u});
%     for v=1:secsPerImage(u)
%         writeVideo(writerObj, frame);
%     end
% end
%
% close(writerObj);
```