

Contents

- [Clearing previous data](#)
- [Load the data and start post processing](#)
- [Binning and data portrayal](#)
- [The basic plot for force chains is ready, I need to now plot the streamlines](#)
- [Streamlines plot](#)

```
% Code Summary:
% This is the latest and probably most accurate attempt at force lines
% until now. This also forms most part in my Internship_Presentation.
% Correct representation of forces has been done vectorially for each bin.
% Prabhu Sir decimated this and also asked why there were streamlines
% outside as well. I'm not sure, I'll have to check it properly why they
% are. Making more bins should however sort that out as here it is much larger.
% Also understand how quiver and streamlines functions in MATLAB work,
% like the basic methodology. One issue is that I've to give few
% initialization points which is input to the meshgrid command at the
% bottom, so that is slightly user dependent. I will have to check that out
% very much. The first plot is a zoomed inplot of the original, and
% contains how the contact network looks like. I was proud of the
% zmat = adj.*(sqrt(zvec.^2./(xvec.^2 + zvec.^2))).*zvec./abs(zvec);
% line as the sgn(zvec) at the end is very concise. Contour, quiver and
% streamline plots have been obtained and plotted. If the number of bins is
% made as the standard 2dp*2dp size, then we will get a much accurate plot
% I feel.
```

Clearing previous data

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

Load the data and start post processing

```
file = importdata("post1\particles_115300.liggghts", " ", 9);
data = file.data;
clear file;

x = data(:, 3); y = data(:, 4); z = data(:, 5);
zbool = z<0.05; % Taking only particles in the heap

x = x(zbool); y = y(zbool); z = z(zbool);
radius = data(1, end-1);
dp = 2*radius;

adj = zeros(size(x, 1)); % This creates the adjacency matrix which will store the overlap between particles if any
xvec = zeros(size(x, 1)); % For each particle, this will store the x component of the interaction
zvec = zeros(size(x, 1));

for i=1:size(adj, 1)
    for j=i+1:size(adj, 2)
        adj(i, j) = dp - sqrt( (x(i) - x(j))^2 + (y(i) - y(j))^2 + (z(i) - z(j))^2 );
        % The above expression calculates the difference between dp and
        % inter-particle centre distance
        if(adj(i, j) < 0)
            adj(i, j) = 0; % particles do not touch
            continue
        end

        adj(j, i) = adj(i, j); % particles touch and the overlap is the same

        xvec(i, j) = x(i) - x(j); % however, as forces are equal and opposite, hence
        xvec(j, i) = x(j) - x(i); % the directions and components will be opposite
```

```

zvec(i, j) = z(i) - z(j);
zvec(j, i) = z(j) - z(i);

% As A->B distance is -ve of B->A distance, hence the signs have been
% accomodated likewise

end
end

xmat = adj.*(sqrt(xvec.^2./(xvec.^2 + zvec.^2))).*xvec./abs(xvec);
zmat = adj.*(sqrt(zvec.^2./(xvec.^2 + zvec.^2))).*zvec./abs(zvec);

% In the above vector operations, I firstly Have the amount of overlap
% stored in adj. Then, using the sqrt() sign, I get the cos or sin
% components of the force acting on the particle. Now, it comes out as +
% for A-B or B-A both interactions as sqrt is always +ve.
% So, I multiplied by sgn(xvec) or sgn(zvec) to get the sign of it as well.
% This will give the directed force components

xmat(isnan(xmat)) = 0;
zmat(isnan(zmat)) = 0;

% There were various NaN values (Not a Number) which were present. The
% above commands removes those NaNs and makes them 0

xadj = zeros(1, size(x, 1));
zadj = zeros(1, size(x, 1));

% Now in xmat and zmat, we have each column containing the vectoral
% forces acting on the particle in that specific direction.
for i=1:size(xadj, 2)

    xadj(i) = sum(xmat(:, i)); % xadj(i) will contain the vector summation of all the forces acting on it (vectorially) with signs
    zadj(i) = sum(zmat(:, i));

end

network = graph(adj); % This command make a graph with appropriate nodes and edges

figure(1);
scatter(x, z, 2, "red")
hold on
mainPlot = plot(network, "NodeColor", [0 0 0], "EdgeColor", [0 0 0], 'XData', x, 'YData', z);
% axis equal

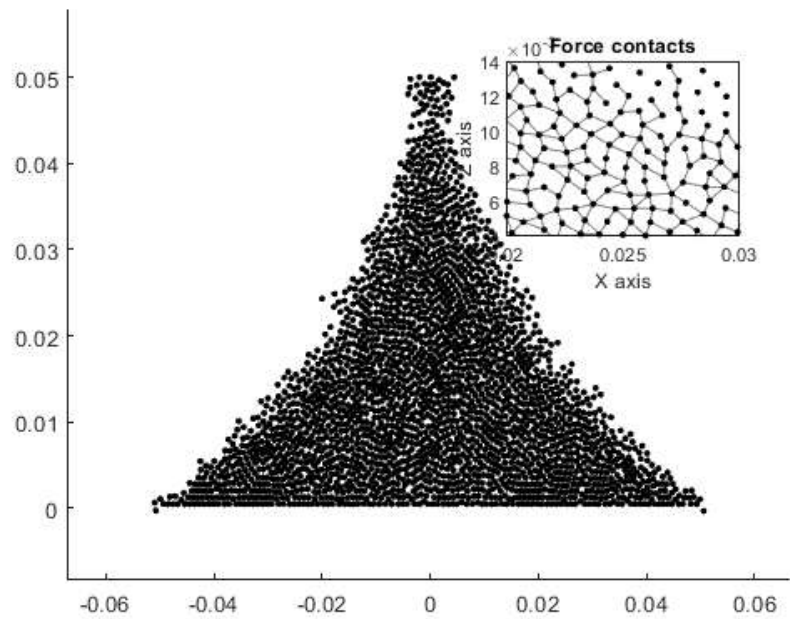
insetPosition = [0.6, 0.6, 0.25, 0.25]; % 0.6*xsize
insetAxes = axes('Position', insetPosition);
copyobj(mainPlot, insetAxes);

xlim(insetAxes, [0.020, 0.020 + 0.010]);
ylim(insetAxes, [0.004, 0.004 + 0.010]);
set(insetAxes, 'Box', 'on');
title(insetAxes, 'Zoomed Inset');

xlabel("X axis")
ylabel("Z axis")
title("Force contacts")

saveas(gcf, "Screen_joints.png")
% This is just to verify if the graph or the adjacency matrix is
% functioning correctly or not. I also wanted to try out the zoomed image
% part as well. One more issue is

```



Binning and data portrayal

```

xbins = linspace(min(x), max(x), 10);
zbins = linspace(0, max(z), 10);

% I have discretized the X-Z plane into 100 small small bins
% 81 nodes will be allotted values

node_value = zeros(size(xbins, 2)-1, size(zbins, 2)-1);

u = zeros(size(xbins, 2)-1, size(zbins, 2)-1); % 9*9
v = zeros(size(xbins, 2)-1, size(zbins, 2)-1); % 9*9

for i=1:size(xbins, 2)-1
    for j=1:size(zbins, 2)-1

        rows = find(x > xbins(i) & x < xbins(i+1) & z > zbins(j) & z < zbins(j+1));
        if (isempty(rows))
            continue % If no particles lie in that bin, then let it remain 0
        end

        node_value(i, j) = mean(mean(adj(:, rows))); % Averaging the amount of overlap

        u(i, j) = mean(xadj(1, rows));
        v(i, j) = -abs(mean(zadj(1, rows)));

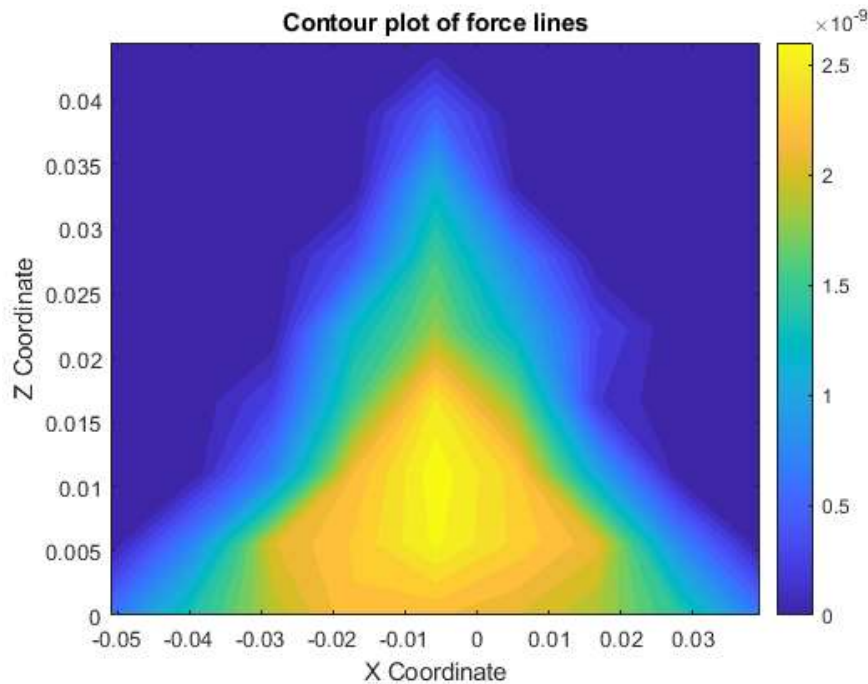
        % abs stands for absolute. This is done so that we get only the -ve
        % z component and that it is directed downwards. We take the mean
        % so that those particles inside the box are averaged correctly

    end
end

[X, Z] = meshgrid(xbins(1:end-1), zbins(1:end-1)); % Restricting the domain to only the domain points at which we have data
figure(2)
levels = 0:1e-10:max(max(node_value));
shading flat
contourf(X, Z, node_value', 500, 'LineColor', 'none', 'LevelList', levels)
xlabel("X Coordinate")
ylabel("Z Coordinate")
title("Contour plot of force lines")
colorbar

```

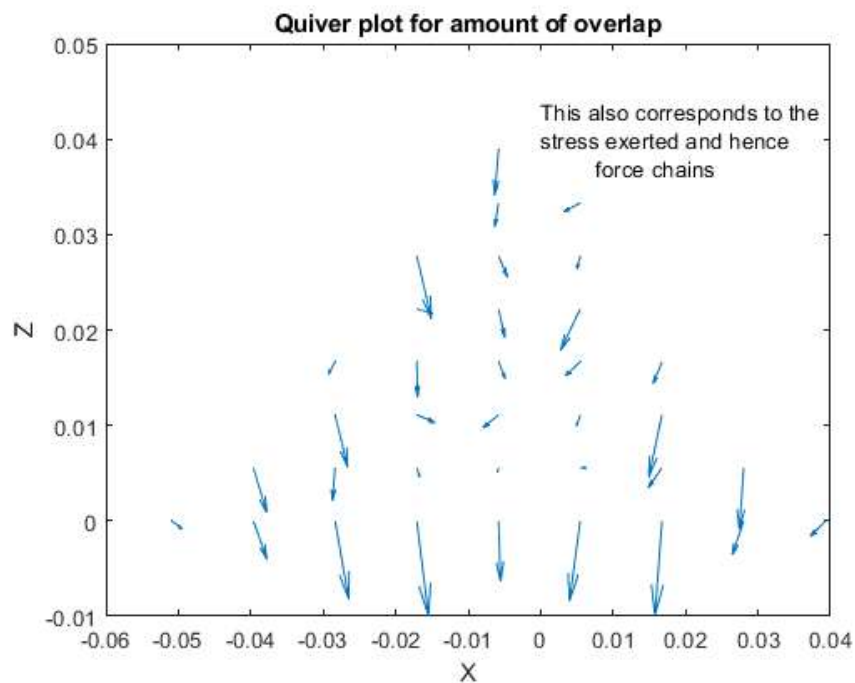
```
% filename = fullfile(folder, "folders", 'Contour.png');
saveas(gcf, 'Contour.png')
```



The basic plot for force chains is ready, I need to now plot the streamlines

```
figure(3);
quiver(X, Z, u', v')
xlabel("X")
ylabel("Z")
title("Quiver plot for amount of overlap")
str = {'This also corresponds to the ', 'stress exerted and hence ', 'force chains'};
text(0, 0.04, str);

saveas(gcf, 'Screen_quiver.png')
```



Streamlines plot

```
% xstart = xbins; % for 10 representative bins
xstart = linspace(min(x), max(x), 50); % for say a higher number of bins, eg 50

[startX,startY] = meshgrid(xstart, zbins(1));

figure(4);
axis equal
verts = stream2(X, Z, (u'), abs(v'), startX, startY);
lineobj = streamline(verts);

hold on
A = scatter(x, z, 5, "red", "o", "MarkerEdgeColor","flat" );
hold off
xlabel("X")
ylabel("Z")
title("Streamlines plot")
```

