Contents


```
% Code Summary:
% This code tried to integrate my 2 codes serially and bin them and then
% radially average them. It was expected to be predicting Janssen's stress
% saturation, but I think multiple things failed me, such as 1. the
% geometry was not that of a cylindrical silo but of a hopper. the radial
% averaging would've been done correctly, but most likely there would not
% be a relation, as this has been done on a heap. If I'd have done the
% thing for the heap only, or only for the silo by keeping a zbool, then I
% might as well have gotten some actual physically correct results unlike
% this random thing.
```

Clearing previous data

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

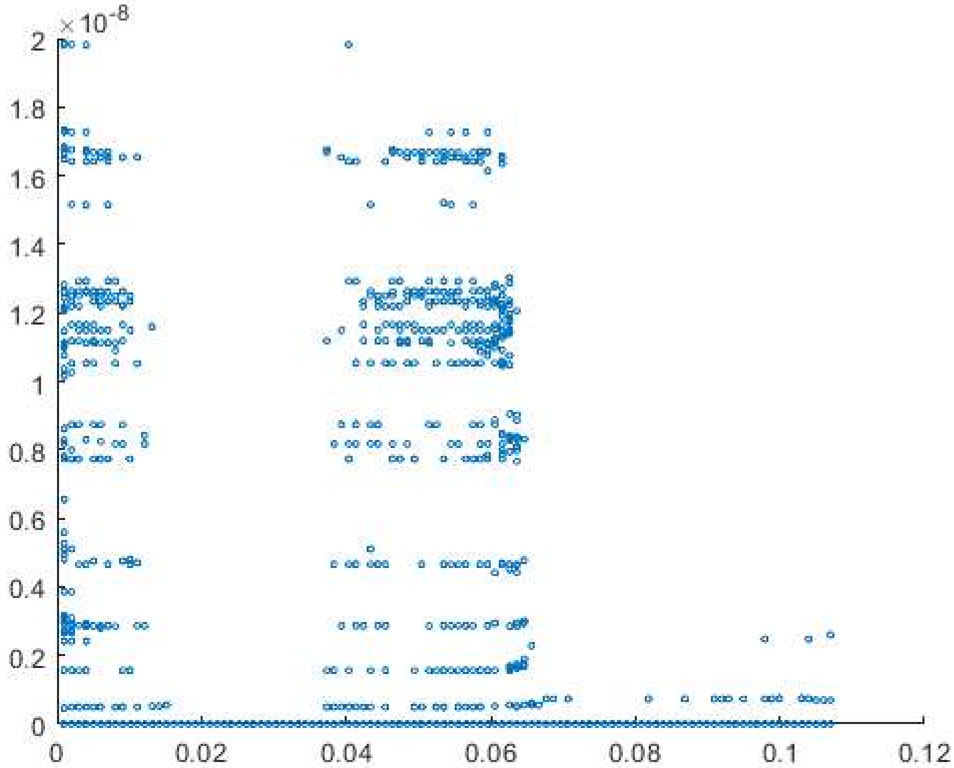
Load file and data

```
file = importdata("post\particles_359000.liggghts", " ", 9);
data = file.data;
clear file;

x = data(:, 3);
y = data(:, 4);
z = data(:, 5);
radius = data(1, end-1);
dp = 2*radius;

r = sqrt(x.^2 + y.^2);
rbins = linspace(0, max(r), 100);

xb=0.01; yb=0.01; zb=0.001; % dimensions of the bin
xl=min(x); yl=min(y); zl=min(z); % lower limits of the volume under consideration
xu=max(x); yu=max(y); zu=max(z); % 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;

xbins = linspace(xl, xu, tbx);
ybins = linspace(yl, yu, tby);
zbins = linspace(zl, zu, tbz);
bindata = zeros(Numbins, 16); % x y z vx vy vz fx fy fz c1 c2 c3 c12 c13 c23 r m

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;
```

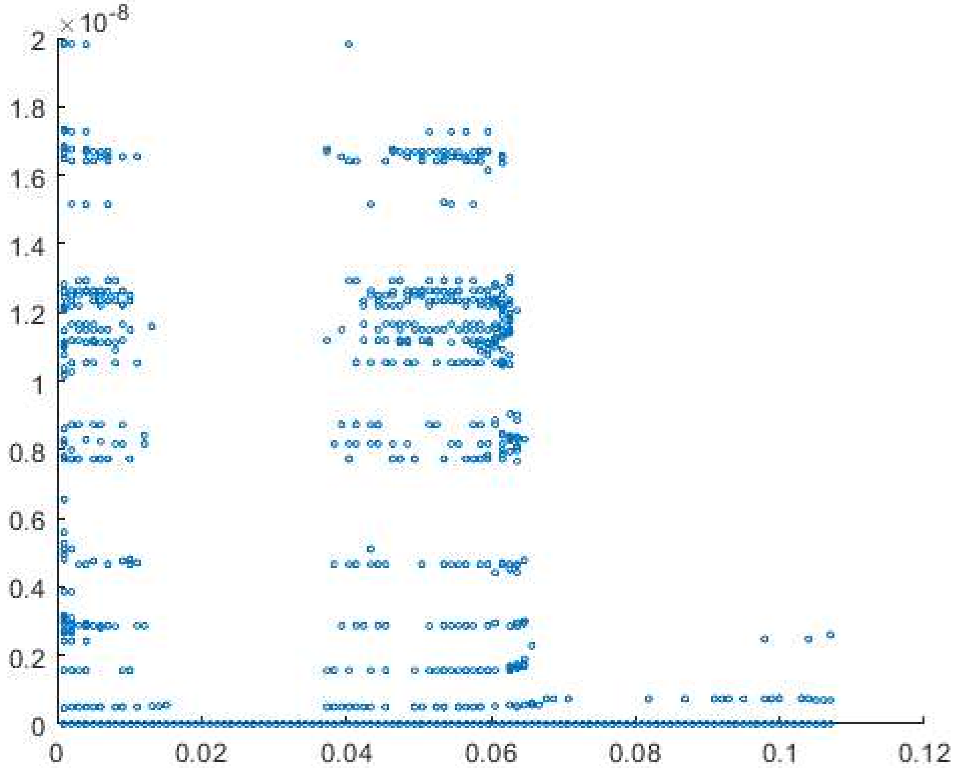
```
            a = (x > xbins(i) & x <= xbins(i+1));
            b = (y > ybins(j) & y <= ybins(j+1));
            c = (z > zbins(k) & z <= zbins(k+1));
            rows = a & b;
            rows = rows & c;
            rows = find(data(rows==1, :));

            if (isempty(rows)) % If there are no particles in the bin volume
                bindata(binno, 1) = xbins(i); bindata(binno, 2) = ybins(j); bindata(binno, 3) = zbins(k);
            else
                bindata(binno, 1) = xbins(i); bindata(binno, 2) = ybins(j); bindata(binno, 3) = zbins(k);
                bindata(binno, 4:15) = mean(data(rows, 6:17));
                bindata(binno, 16) = mean(sqrt(x(rows).^2 + y(rows).^2));
            end

        end
    end
end
```

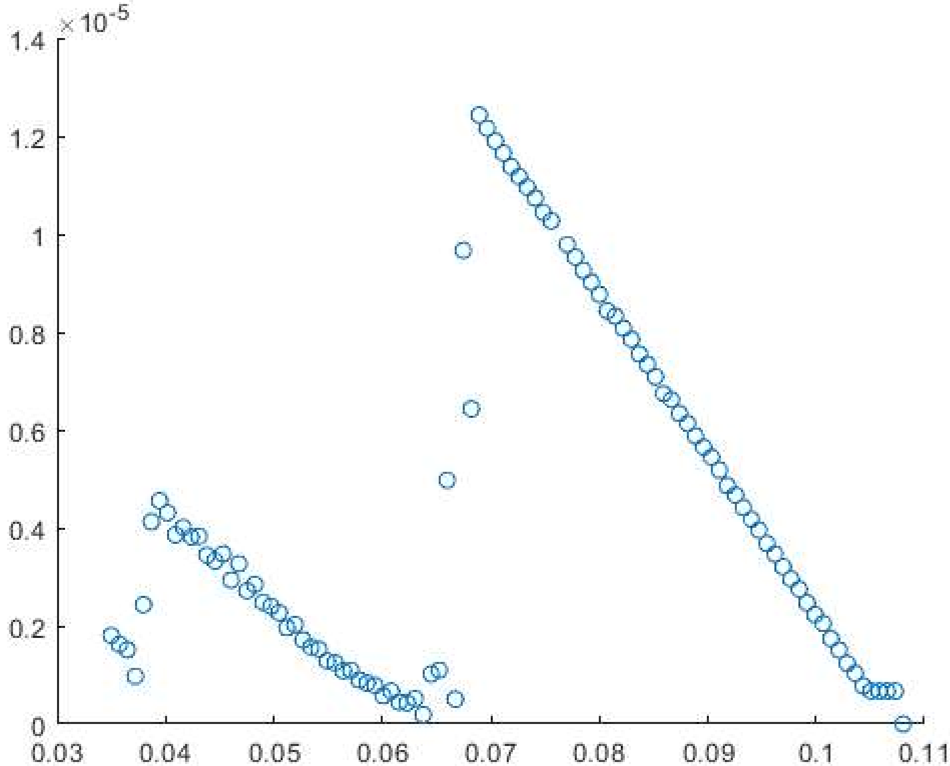
```
radco = sqrt(bindata(:, 1).^2 + bindata(:, 2).^2);
raduni = unique(radco);
zuni = unique(bindata(:, 3));

num1 = size(raduni, 1);
num2 = size(zuni, 1);
radavgdata = zeros(num1*num2, 17);

k=1;
for i=1:num1

    samedist = find(radco == raduni(i)); % get the indices of unique radii

    for j=1:num2

        samez = find(bindata(:, 3) == zuni(j)); % get the indices of unique z
        common = intersect(samedist, samez); % get the coordinates of common radial distance and z
        if(isempty(common))
            continue;
        end
        radavgdata(k, 1:16) = mean(bindata(common, :), 1); % take mean of the common data points
        radavgdata(k, 17) = raduni(i); % also saving the radius for ease
        k = k + 1; % update counter of row index

    end

end
```

```
rows = find(radavgdata(:, 17) > raduni(10) & radavgdata(:, 17) < raduni(170));
z1 = radavgdata(rows, 3);
tauzz = radavgdata(rows, 15);
scatter(z1, (abs(tauzz)), 5)
```




```
zbins = linspace(0.035, max(z), 100);
tau = zeros(1, 100);

for i=1:100-1
    rows = find(z > zbins(i) & z < zbins(i+1));
    tau(i) = mean(data(rows, 14));
end

figure;
scatter(zbins, abs(tau))
```





Trying to see zz stress contour along planes using the binned data will be exactly similar to the earlier versions, so I'll take it lite

---

Published with MATLAB® R2023a