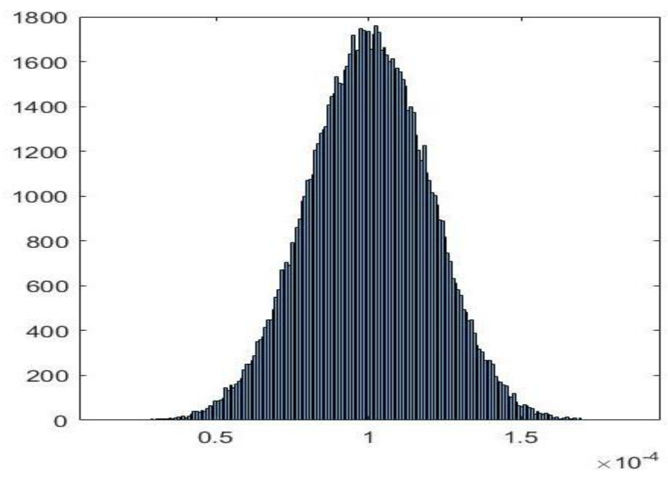
SHUBHAM WARE

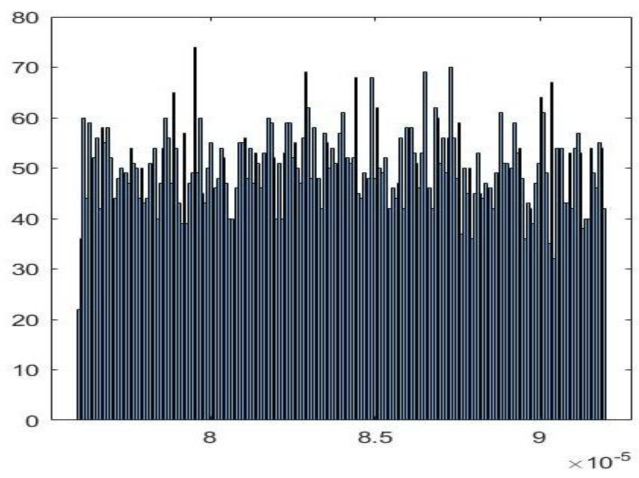
M.Nr2211211

Ware.shubham@st.ovgu.de

%%first particle and sieve are created using matlab function rand and normrand, this also allowed us to get random values of particles.

```
Particle=normrnd(d,sigma,[1,Np]);
Sieve=Wi+(Wi-Wo)*rand(1,Ns);
Particle0=Particle;
```



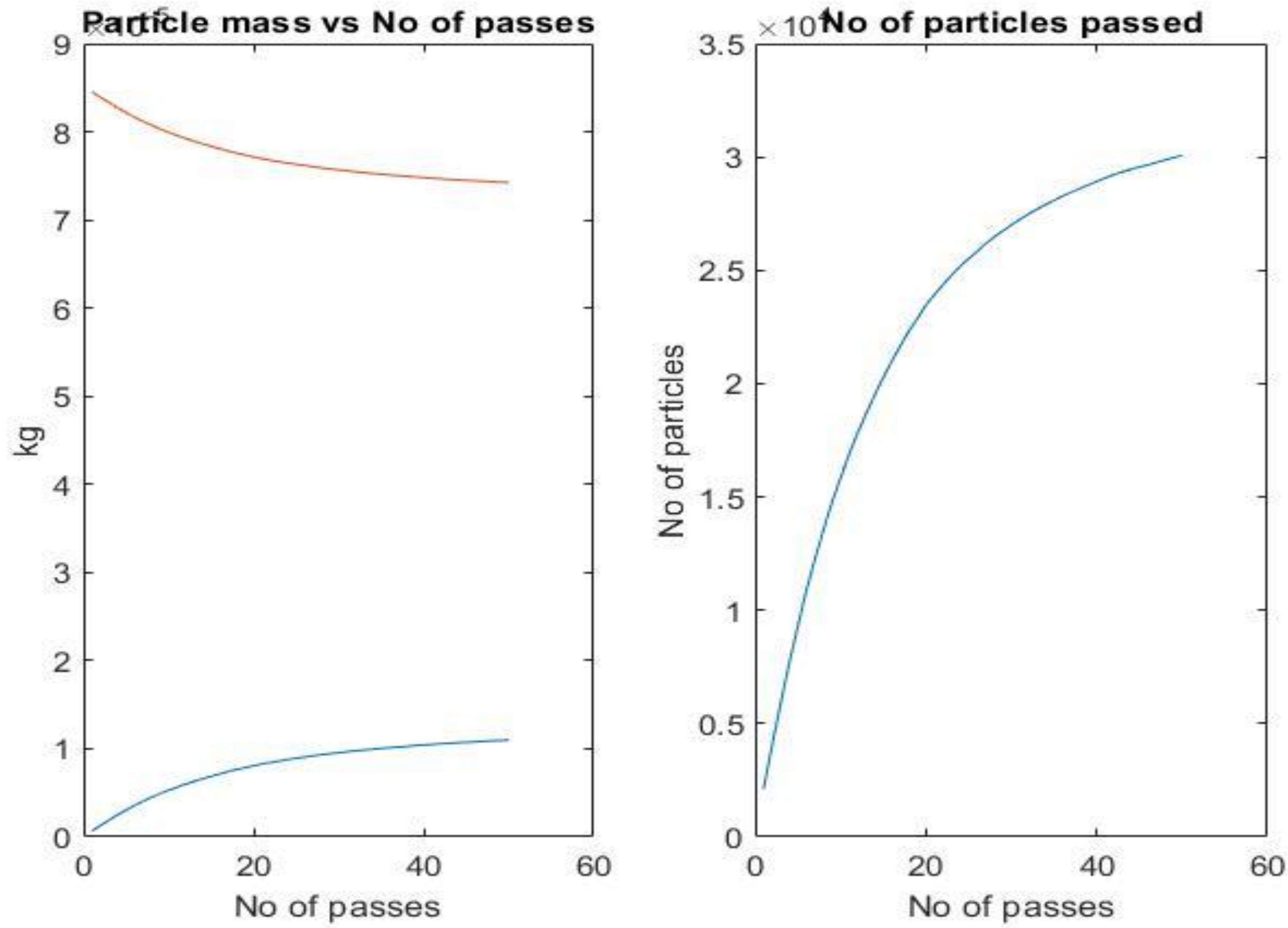


%%then to simulate sieving, we simulated 50 no of passes in which each sieve meets at least one particle

```
for i=1:No_of_passes

            for j=1:Ns
               k= randperm(length(Particle),1);  %%randperm is used to take random
.                                                  sampling from whole set of particles
                if Particle(k)<=Sieve(j);
                   Fine_particle(pass)=Particle(k);
                   Particle(k)=[];              %%particle which pass are then deleted from .
.                                                original set of particles
                   pass=pass+1;
                end
            end
 %%then mass of coarse and fine particles are calculated using standard formulas.
        Coarse_particle_mass(i)=sum(Ro*pi/6.*Particle.^3);
        Fine_particle_mass(i)=sum(Ro*pi/6.*Fine_particle.^3);
        No_of_particles_passed(i)=pass;

    End
```

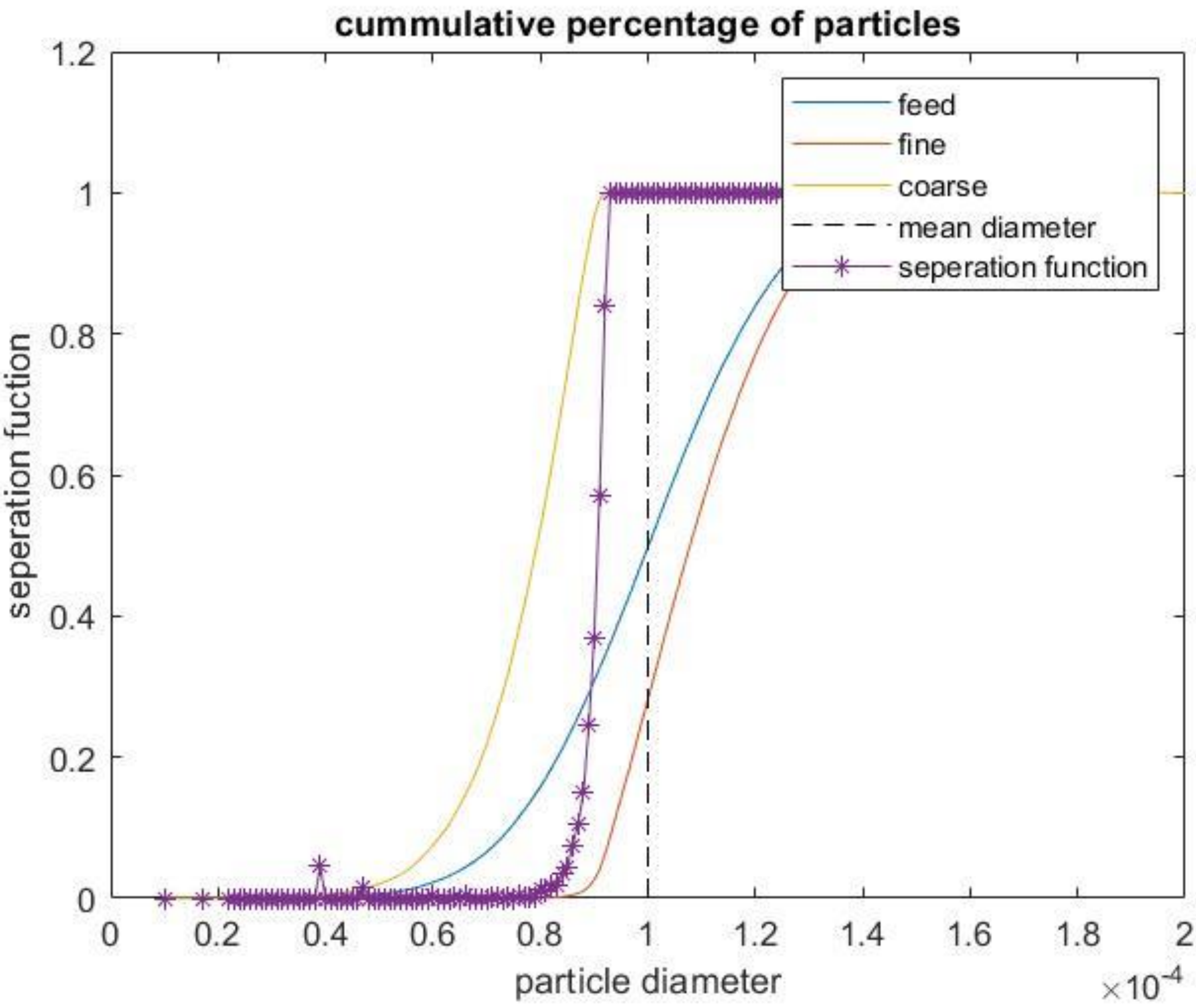



```
Particles
%%cumulative distribution are calulated by for loop
    for i=2:length(range)
  %%logical operator creates logical vector for respective values
        p= Particle<=range(i)&Particle>=range(i-1);
  %%using that lofical vector percentage is calculated
        Percentage_coarse(i)=Percentage_coarse(i-1)+sum(p)/length(Particle);
    end

 %%same procedure is repeted for fine and coarse particle
   for i=2:length(range)
        p= Fine_particle<=range(i)&Fine_particle>=range(i-1);
        Percentage_fine(i)=Percentage_fine(i-1)+sum(p)/length(Fine_particle);
    end


   for i=2:length(range)
        p= Particle0<=range(i)&Particle0>=range(i-1);
        Percentage(i)=Percentage(i-1)+sum(p)/length(Particle0);
   end

%%plotting result for cumulative distribution
```



Particles

%%calulation of seperation fuction is done by first defining function

%%handle for mass and seperation function

```
Seperation_function=@(Mc,Mf) (Mc/(Mc+Mf));
mass=@(d) (Ro*pi/6.*d.^3);
```

%%then actual function is calulated using for loop

```
for i=2:length(range)
    %%logical operator creates logical vector containg one when parameter
    %%are fullfilled for give calss.
        c= Particle<=range(i)&Particle>=range(i-1);
        f= Fine_particle<=range(i)&Fine_particle>=range(i-1);
    %%to check if we have maintained intigrity of number of particles
    %%particles in each calss are stored
        Particles_in_range(i)=length(find(c))+length(find(f));
        Mc=sum(mass(c.*(Particle)));
        Mf=sum(mass(f.*(Fine_particle)));
        T(i)=Seperation_function(Mc,Mf);
end
```

%%verifying no particles to check integrity of programm

```
varifying_no_of_particles =

  logical

   1
```

particles

%%now finally finding values of cut diameter and value of k

```
for i=1*10^4:3*10^4
```

%%values are rounded off as exact match is near impossible, here round function is used to different rounding values considering some values were not converging.

```
    if round(Particle(i),7)==round(Fine_particle(i),7);
       cut_off(count_cut)=Particle(i);
       count_cut=count_cut+1;
    elseif round(Particle(i),4)==0.334*round(Fine_particle(i),4);
        d25(count_25)=Particle(i);
        count_25=count_25+1;
    elseif round(Particle(i),6)==3*round(Fine_particle(i),6);
        d75(count_75)=Particle(i);
        count_75=count_75+1;
    end
end
```

And finally results were obtained by using putting values in formulas that we have obtained form equation.

```
D50 =

   7.4648e-05


D25 =

   8.9000e-05


D75 =

   9.4997e-05


k =

    0.9369
```

CONCLUSION:

We see very Ideal separation factor which is near about 1,hence giving us limitation to which we can apply simulation to real life problems, however point to be noted is even with simulation we can get near to practical situation by introducing randomness in data, for example here by using RAND function. Thus simulation can be a great tool for engineer to get generalized idea about the procedure and then to check its practical viability.