Table of Contents

Estimation of first cell distance to wall

The wall cell distance is crucial for accurate predictions of the RSM method in simulation of turbulent flow, so the distance is estimated using suggested formulas of ANSYS Fluent

```
% Pipe dimensions and fluid properties are first determined.
prompt = 'Enter fluid density (kg/m^3): ' ;
rho_f = input(prompt);

prompt = 'Enter dynamic viscocity (m^2/s): ' ;
mu = input(prompt);

prompt = 'Enter velocity (m/s): ' ;
U = input(prompt);

prompt = 'Enter pipe length (m): ';
L = input(prompt);

prompt = 'Enter pipe diameter (m): ';
d = input(prompt);

Re_p = rho_f * U * d / mu;
% Re_p = rho * U * L / mu;

Cf = 0.079 * Re_f^(-0.25);
tau_w = 0.5 * Cf * rho_f * U^2;
U_tau = sqrt(tau_w/rho_f);

% Choice of y_plus is made on recommendation by ANSYS. For RSM method,
% y_plus should not be higher than 1
prompt = 'Select y_plus from 1 to 200: ';
y_plus = input(prompt);

y = (mu * y_plus)/(rho_f * U_tau);

Y = sprintf('Cell distance should be %d meters away from the
 wall.',y);
disp(Y)
```

Compuation of deposition velocity

The dimensionless deposition velocity is computed based on number of inlet particles and total registered outlet particles. The relaxation time is set manually and is used to plot the deposition velocity later on.

```
% clear all
close all
% clc


% Fluid properties and pipe dimensions

M = 28.97; % Molecular weight of air [kg/kmol]
rho_f = 1.16; % fluid density [kg/m^3]
% rho_p = 3750; % aluminium oxide density
rho_p = 2000; % silica colloid density
nu = 1.589*10^-5; % Kinematic visc(v)
mu = nu * rho_f; % Dynamic visc (u)
R = 8314; % Universal gas constant [J/kmolK]
x = 5; % Dynamic shape factor
T = 373; % Fluid temp [Kelvin]
g = 0;

d = 0.1; % Channel diameter / height [metre]
L = 10; % Length of pipe [m]
dx = 5; % Length in which particles travel (minus the initial meter)
S = rho_p/rho_f; % Particle-fluid-ratio


% u = 5; % Avg. velocity, Tandberg
u = 48.0701; %  Avg velocity 50 m/s inlet sim
% u = 9.8015; % Avg velocity 10 m/s inlet sim

% Particle diameter
d_p = 5*10^-8; % Particle diameter [metre]
d_p_micro = d_p * 10^6; % Particle diameter [micrometre]

% Development of particle relaxation time

Re_f = (rho_f * u * d)/(mu); % Reynolds number
l = nu * ((pi*M)/(2*R*T))^0.5; % Mean free path
% l = 0.0664 * 10^-6 *(101/101)*(373/293)*((1+(110)/(293))/(1+(110)/
(373))); % Crowe
Kn = l ./ d_p; % Knudsen number
Cc = 1 + Kn.*(2.514 + 0.8 .* exp(-0.55./Kn)); % Cunningham slip
 correction factor
% Cc = 1 + Kn.*(2.34 + 1.05 .* exp(-0.39./Kn)); % Tandberg, uses
 0.5*Kn
Cd = (24/Re_f) * (1+0.15*Re_f^0.687); % Drag coefficient
f = (3*pi*mu.*d_p.*x)./(Cc); % Friction coefficient
k = 1.38054 * 10^-23; % Boltzmann constant
D_b = (k*T*Cc)./(3*pi*mu.*d_p); % Stokes-Einstein equation diffusion
 coeff.

g_plus = g*nu/(u_fric^3); % Dimensionless gravitational acceleration
L_1_plus = 3.08*rho_f./(rho_p.*d_p_plus); % Dim. less constant of
 Ahmadi
k_plus = 0; % Surface roughness, 0 for smooth surface
```

```
% Cf = 0.0791 * Re_f^(-0.25);
Cf = 2 * (2.236*log(Re_f) - 4.639)^(-2); % friction factor
tau_w = Cf * 0.5 * rho_f * u^2; % Wall shear stress
u_fric = sqrt(tau_w / rho_f); % Friction velocity
d_p_plus = d_p .* u_fric/mu; % Dimensionless particle diameter


tau_p = (rho_p * d_p^2) / (18*mu); % Particle inertia relaxation time
tau_p = tau_p * Cc; % Accounting for rarefied gas effects
tau_plus = tau_p * u_fric^2 / nu % Dimensionless particle realaxation
 time


% Computation of dimensionless deposition velocity

Ninn = 100000; % Particles at inlet
Nout = 100000-611; % Particles at outlet

v_dpp = (d*u)/(2*u_fric*dx) * log(Ninn/Nout) % Dim. less deposition
 vel.
```

Plotting deposition velocity

The following script is fed the deposition velocities extracted from FLUENT and computed by the previous script. The deposition velocities are plotted over a parameter of particle relaxation times set by the fluid flow properties and particle sizes. Lastly, the empirical formulae of Papavergos & Hadley (1984) and Ahmadi (1994) are computed for the same particle size range and compared.

```
d_p = linspace(4*10^-9,2.5*10^-5,1000000); % Particle diameter [m]
 parameter


% Papavergos & Hadley (1984)

Vd_p = zeros(1,length(tau_p_plus));

for i = 1:length(tau_p_plus)

    j = tau_p_plus(i);

    if j < 0.2
        Vd_p(i) = 0.065*(Sc_p(i))^(-2/3);
    elseif 0.2 < j && j <= 20.0
        Vd_p(i) = 3.5 * 10^-4 * j^2;
    elseif j > 20
        Vd_p(i) = 0.18;
    end
end


% Ahmadi (1994)
Vd_p_a = zeros(1,length(tau_p_plus));
```

```
for i = 1:length(tau_p_plus)

    j = tau_p_plus(i);

    if i == 1

        a = 0.084 * Sc_p(i)^(-2/3);
        b = ((0.64*k_plus+0.5*d_p_plus(i))^2 ...
            + (tau_p_plus(i)^2*g_plus*L_1_plus(i))/(0.01085*...
            (1+tau_p_plus(i)^2*L_1_plus(i))))/
(3.42+((tau_p_plus(i)^2*...
            g_plus*L_1_plus(i))/
(0.01085*(1+tau_p_plus(i)^2*L_1_plus(i)))));
        c = (1/(1+tau_p_plus(i)^2*L_1_plus(i)));
        e = (1+8*exp(-(tau_p_plus(i)-10)^2/(32)));
        f = (0.037)/(1-tau_p_plus(i)^2*L_1_plus(i)*(1+(g_plus)/
(0.037)));

        Vd_p_a(i) = a + 0.5*b^c * e * f;


    elseif Vd_p_a(i-1) < 0.14

        a = 0.084 * Sc_p(i)^(-2/3);
        b = ((0.64*k_plus+0.5*d_p_plus(i))^2 ...
            + (tau_p_plus(i)^2*g_plus*L_1_plus(i))/(0.01085*...
            (1+tau_p_plus(i)^2*L_1_plus(i))))/
(3.42+((tau_p_plus(i)^2*...
            g_plus*L_1_plus(i))/
(0.01085*(1+tau_p_plus(i)^2*L_1_plus(i)))));
        c = (1/(1+tau_p_plus(i)^2*L_1_plus(i)));
        e = (1+8*exp(-(tau_p_plus(i)-10)^2/(32)));
        f = (0.037)/(1-tau_p_plus(i)^2*L_1_plus(i)*(1+(g_plus)/
(0.037)));

        Vd_p_a(i) = a + 0.5*b^c * e * f;


    else
        Vd_p_a(i) = 0.14;
    end
end

% Dim. less relaxation times and deposition velocities computed from
% results provided by FLUENT DPM for listed particle sizes.

d_p_match_sst = [4e-9, 6e-9, 2.6e-8, 8e-8, 2e-7, 6e-7, 8e-7, 1e-6,...
    2e-6, 3e-6, 6e-6, 3e-5];
tau_p_match_sst = [0.0012, 0.0018, 0.0084, 0.0304, 0.1067, 0.6681,...
    1.1271, 1.7045, 6.3656, 13.9833, 54.5758, 1000];
v_d_p_match_sst = [0.0099, 0.0095, 0.0091, 0.0093, 0.0078, 0.0048,...
    0.0045, 0.0051, 0.0298, 0.2150, 0.2005, 0.1940];
```

```
v_d_p_rsm = [0.0190, 0.0185, 0.0184, 0.0185, 0.0180, 0.0140,
 0.0139,...
    0.0152, 0.0292, 0.1392, 0.1774, 0.1855];


% Dim. less deposition velocity plotted against dim. less relaxation
 time
figure(1)
loglog(tau_p_plus, Vd_p, '-', tau_p_plus, Vd_p_a, '-',
 tau_p_match_sst,...
    v_d_p_match_sst, 'go', tau_p_match_sst, v_d_p_rsm, 'ro')
grid on
xlabel('Dim. less relaxation rate')
ylabel('Dim. less deposition velocity')
title('Dim. less deposition velocity for a 10 m pipe at 48 m/s')
legend('Papavergos & Hedley (1984)','Ahmadi (1993)',...
    'k-\omega SST simulation','RSM simulation', 'location','best')
axis([10^-3 10^3 10^-5 10^0])

% Plot of Cunningham slip correction factor
figure(2)
loglog(d_p,Cc)
grid on
title('Rarefied gas effect by particle size, 48 m/s')
xlabel('Particle diameter')
ylabel('Cunningham correction slip factor')

% Plot of particle Schmidt number
figure(3)
loglog(d_p,Sc_p)
grid on
title('Schmidt number by particle size, 48 m/s')
xlabel('Particle diameter')
ylabel('Particle schmidt number')
```

Integration of deposition rate along pipe wall

Having recieved deposition rates from MATLAB using the UDF in FLUENT, the deposited particles are plotted against pipe length to reveal the concentration of deposited particles. The deposition rates are then integrated across each cell, revealing the deposition velocity of all cells.

```
figure(4)

subplot(1,2,1)
plot(pipe_length, deposited)
grid on
xlabel('Pipe wall position [m]')
ylabel('# deposited particles')
title(['Deposited ',num2str(d_p),' particles, long pipe'])

Ninn = 100000; % Particles at inlet

for i = 1:length(pipe_length)
```

```
    if i == 1

        Nout = Ninn - deposited(i);
        v_dep_p(i) = (d*u)/(2*u_fric*pipe_length(i)) * log(Ninn/Nout);
        Ninn = Nout;

    elseif pipe_length(i) == pipe_length(i-1)

        v_dep_p(i) = v_dep_p(i-1);

    else

        Nout = Ninn - deposited(i);
        v_dep_p(i) = (d*u)/(2*u_fric*(pipe_length(i)-
pipe_length(i-1))) * log(Ninn/Nout);
        Ninn = Nout;

    end

end

% The integration of dep. velocit over all cells should provide an
 average
% deposition velocity equal to that computed in previous script.

correlation = sum(v_dep_p)/(length(pipe_length)); % Check result

plot(pipe_length, v_dep_p)
grid on
xlabel('pipe length [m]')
ylabel('Dim. less depisition velocity [-]')
title(['Dep. velocity per cell for ', num2str(d_p), ' diameter
 particles'])
```

Dimensionless velocity profile

Dimenionless velocity profile is plotted close to wall to evaluate the different turbulence models.

```
% The wall shear stress is retrieved at the beginning of deposition
 region
% tau_w = 0.27; % wall shear stress for long pipe vel 9.8 RSM (0.26
 SST)
tau_w_sst = 4.63; % long pipe wel 48 SST
tau_w_rsm = 4.8; % long pipe wel 48 RSM

% Friction velocities are computed
u_fric_sst = sqrt(tau_w_sst / rho_f);
u_fric_rsm = sqrt(tau_w_rsm / rho_f);

% The number of cells from the wall to y+ = 10^3 are measured, with
 the
% data provided for by FLUENT results.
```

```
for i = 1:length(wall_distance_)

    if i == 1

        y(i) = wall_distance_(i);
        y_plus_sst(i) = (u_fric_sst * y(i)) / nu;
        y_plus_rsm(i) = (u_fric_rsm * y(i)) / nu;

    elseif y_plus_sst(i-1) < 10^3 % Not necessary to measure for hiher
 val.

        y(i) = wall_distance_(i);
        y_plus_sst(i) = (u_fric_sst * y(i)) / nu;
        y_plus_rsm(i) = (u_fric_rsm * y(i)) / nu;

    else
        % Velocity is fetched for the measured cells.
        u_plus_rsm = vel_rsm(1:i-1)./u_fric_rsm;
        u_plus_sst = vel_sst(1:i-1)./u_fric_sst;

        break

    end
end

figure(4)
% plot(y_plus, u_plus_sst, y_plus, u_plus_rsm)
semilogx(y_plus_sst, u_plus_sst, y_plus_sst, u_plus_rsm)
axis([10^-1 10^3 0 25])
grid on
title('The velocity law of the wall, long pipe 48 m/s')
xlabel('y+')
ylabel('u+')
legend('\kappa - \omega SST simulation','RSM
 simulation','location','best')
```

Published with MATLAB® R2018a