Process and Governing equation

The process is described for discontinuous milling. The governing equation is given as:

$$\frac{dm_i}{dt} = -s_i m_i(t) + \sum_{j=1}^{i-1} s_j m_j(t) b_{i,j} \quad (1)$$

1. Determination of Missing Mass Fractions:

For the missing mass fraction values, we can evaluate that, in any specific time interval, mass is distributed in various i^{th} size categories in rows for each j^{th} column of matrix, so for each column sum of all i^{th} mass fractions should be unity, with this idea matrix $b_{i,j}$ can be expressed as:

$$\boldsymbol{b_{i,j}} = \begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 \\ 0.4 & 0.0 & 0.0 & 0.0 \\ 0.3 & 0.4 & 0.0 & 0.0 \\ 0.2 & 0.3 & 0.3 & 0.0 \\ 0.1 & 0.3 & 0.7 & 1.0 \end{bmatrix}$$

The set of Governing Differential Equations for each size category is derived from equation (1) as:

$$\frac{dm_1}{dt} = -s_1 m_1(t)$$

$$\frac{dm_2}{dt} = \left[b_{2,1} s_1 m_1(t)\right] - s_2 m_2(t)$$

$$\frac{dm_3}{dt} = \left[b_{3,1} s_1 m_1(t) + b_{3,2} s_2 m_2(t)\right] - s_3 m_3(t)$$

$$\frac{dm_4}{dt} = \left[b_{4,1} s_1 m_1(t) + b_{4,2} s_2 m_2(t) + b_{4,3} s_3 m_3(t)\right] - s_4 m_4(t)$$

$$\frac{dm_5}{dt} = \left[b_{5,1} s_1 m_1(t) + b_{5,2} s_2 m_2(t) + b_{5,3} s_3 m_3(t) + b_{5,4} s_4 m_4(t)\right] - s_5 m_5(t)$$

2. Complete Algorithm based on Runge- Kutta (RK4)

Step 1, Initialization of variable for number of iterations with changing time step: The time step is user defined and might be changed for assessing the performance of the program and analyse errors. In the matlab file Crushingsand.m, n iterations are done each by varying the value of time step by a factor 10^{-1} per iteration.

```
%%To run the program for different values of time step, n is initialized
%which can be changed according to the time step required by the user
N=5;
```

Step 2, Running loop for various time intervals 10^-2,10^-3,10^-4,10^-5,10^-6: The time step changes for each iteration, n. The time vector thus, changes for each value of time step.

```
for n=1:N
    dt=10^(-2-(n-1));
    delt(n)=dt;
    % Start time
    tstart = 0;
    % End time
    tend = 50;
    % Time vector
    t = tstart:dt:tend;
```

Step 3, Mass fraction, communition rate, grain size category variables initialization.

```
%Mass-dependent comminution rate s1, s2, s3, s4, s5 written in a 1-D array
s = [0.5 0.4 0.3 0.2 0.0];

% Mass fraction matrix written in 4x5 matrix
b = [0.0 0.0 0.0 0.0 ; 0.4 0.0 0.0 0.0 ;0.3 0.4 0.0 0.0 ;0.2 0.3 0.3 0.0 ; 0.1 0.3
0.7 1.0];


% Allocation of the solution vector, each vector will contain N no. of solutions
% corresponding to the number of time stepsdetermined by length(t)
    m1 = zeros(1,length(t)); m2 = zeros(1,length(t)); m3 = zeros(1,length(t));
    m4 = zeros(1,length(t)); m5 = zeros(1,length(t));

% Initial mass for each grain size category
    m1(1) = 100; m2(1) = 0; m3(1) = 0; m4(1) = 0; m5(1) = 0;
```

Step 4, Initializing 5 variables which will assume one solution per iteration for the five grain size categories: This is done because assigning m1, m2... vectors to the function handlers f1, f2... complicates the code and vector elements with changing index cannot be assigned to function handlers.

1st iteration: y1=m1(1) , y2=m2(1) .....

2nd iteration: y1=m1(2) , y2=m2(2) ..... and so on.

```
    y1 = 100; y2 = 0; y3 = 0; y4 = 0; y5 = 0;
```

Step 5, Assigning the RHS of five governing differential equations to function handlers f1, f2, f3, f4, f5. The governing equations are given in section 1.

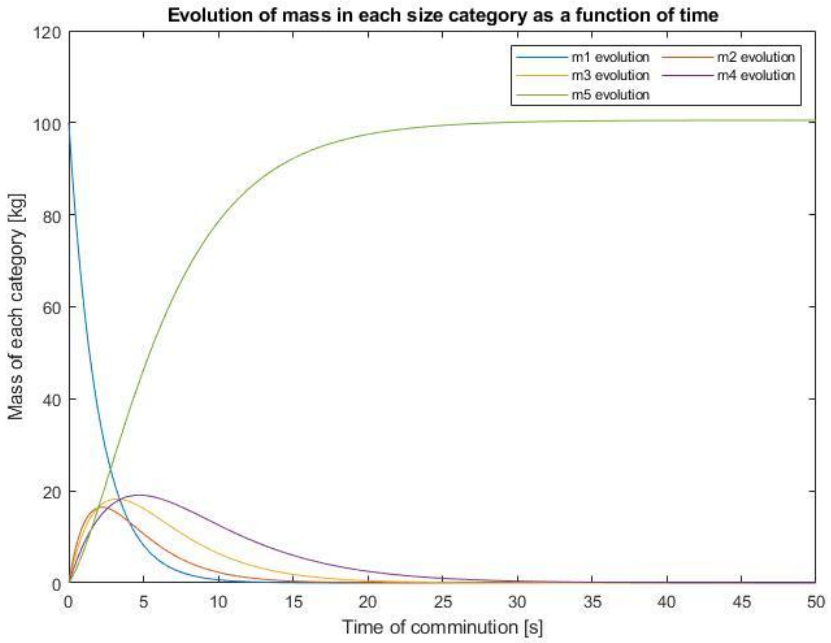
```
f1=@(y1) -s(1)*y1;
f2=@(y1,y2) b(2,1)*s(1)*y1 - s(2)*y2 ;
f3=@(y1,y2,y3) b(3,1)*s(1)*y1 + b(3,2)*s(2)*y2 - s(3)*y3 ;
f4=@(y1,y2,y3,y4) b(4,1)*s(1)*y1 + b(4,2)*s(2)*y2 + b(4,3)*s(3)*y3 - s(4)*y4
f5=@(y1,y2,y3,y4,y5) b(5,1)*s(1)*y1 + b(5,2)*s(2)*y2 + b(5,3)*s(3)*y3 +
b(5,4)*s(4)*y4 -s(5)*y5 ;
```

Step 6, Computing the mass evolution of each grain size categories using Runge-Kutta 4 method. The functions for computing mass evolution depends on the grain size categories. The RHS side of each of the function does not contain time variable and hence is not required to be included while calling the functions. For every subsequent grain size category, evolution of mass depends on the mass evolved from the previous grain size category. Hence, the number of variables for computing using RK-4 increases as we go from m1 to m5.

```
    for i=1:length(t)-1

        y1=m1(i);
        %RK4 method to compute mass evolution for m1 size category(depends only on
        %m1)
        k1=f1(y1);
        k2=f1(y1+(dt*k1/2));
        k3=f1(y1+(dt*k2/2));
        k4=f1(y1+dt*k3);
        m1(i+1)=m1(i)+(dt/6)*(k1+(2*(k2+k3))+k4);
```

```
        y2=m2(i);
        %RK4 method to compute mass evolution for m2 size category(depends on
        %m1 and m2)
        k1=f2(y1,y2);
        k2=f2(y1+(dt*k1/2),y2+(dt*k1/2));
        k3=f2(y1+(dt*k2/2),y2+(dt*k2/2));
        k4=f2(y1+dt*k3,y2+dt*k3);
        m2(i+1)=m2(i)+(dt/6)*(k1+(2*(k2+k3))+k4);

        y3=m3(i);
        %RK4 method to compute mass evolution for m3 size category (depends on
        %m1,m2 and m3)
        k1=f3(y1,y2,y3);
        k2=f3(y1+(dt*k1/2),y2+(dt*k1/2),y3+(dt*k1/2));
        k3=f3(y1+(dt*k2/2),y2+(dt*k2/2),y3+(dt*k2/2));
        k4=f3(y1+dt*k3,y2+dt*k3,y3+dt*k3);
        m3(i+1)=m3(i)+(dt/6)*(k1+(2*(k2+k3))+k4);

        y4=m4(i);
        %RK4 method to compute mass evolution for m4 size category (depends on
        %m1,m2, m3 and m4)
        k1=f4(y1,y2,y3,y4);
        k2=f4(y1+(dt*k1/2),y2+(dt*k1/2),y3+(dt*k1/2),y4+(dt*k1/2));
        k3=f4(y1+(dt*k2/2),y2+(dt*k2/2),y3+(dt*k2/2), y4+(dt*k2/2));
        k4=f4(y1+dt*k3,y2+dt*k3,y3+dt*k3, y4 +dt*k3);
        m4(i+1)=m4(i)+(dt/6)*(k1+(2*(k2+k3))+k4);

        y5=m5(i);
        %RK4 method to compute mass evolution for m5 size category (depends on
        %m1,m2, m3, m4 and m5)
        k1=f5(y1,y2,y3,y4,y5);
        k2=f5(y1+(dt*k1/2),y2+(dt*k1/2),y3+(dt*k1/2),y4+(dt*k1/2),y5+(dt*k1/2));
        k3=f5(y1+(dt*k2/2),y2+(dt*k2/2),y3+(dt*k2/2),y4+(dt*k2/2),y5+(dt*k2/2));
        k4=f5(y1+dt*k3,y2+dt*k3,y3+dt*k3, y4 +dt*k3,y5+dt*k3);
        m5(i+1)=m5(i)+(dt/6)*(k1+(2*(k2+k3))+k4);

    end
```

3. Plot for evolution of mass in each category as a function of time



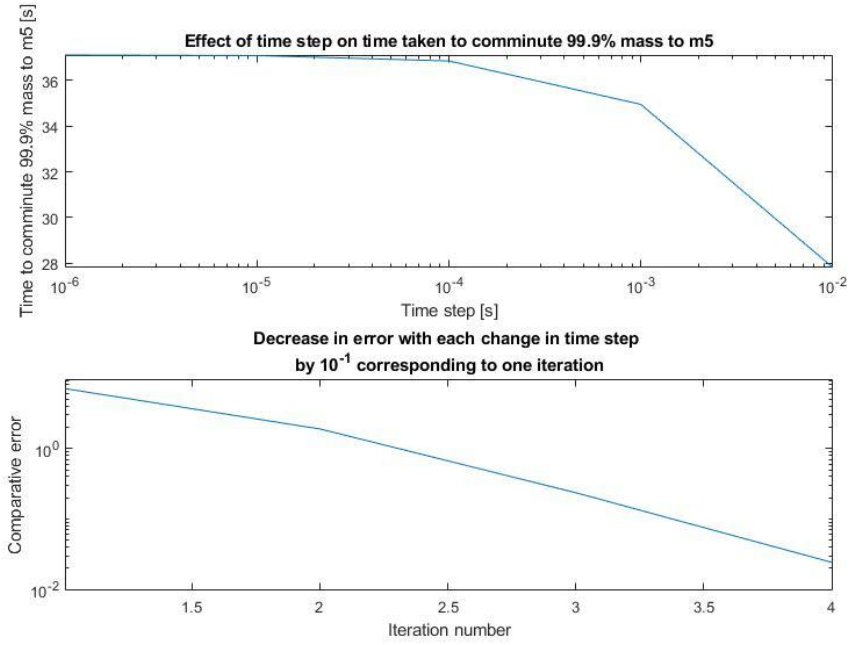
4. Time required to comminute 99.9% of the sand grains to m5

Mass evolved in each time step for the size category m5 is scanned using for loop till the allotted time for comminution and using if condition, the elements of the m5 array are subjected to a condition of being equal to or greater 99.9% of the initial mass of sand grains. As soon as the criteria is fulfilled in the i^{th} iteration, the element m5(i) is saved in a separate variable to be checked later. The mass m5(i) was however comminuted in the previous iteration (i-1). Thus, t(i-1) is the required time. The loop is exited using 'break' as soon as the time required is found out. **For a time step of 0.01 the time required is computed as 27.82 s**.

```
for i=1:length(t)-1
    if m5(i)>=99.9000       %Condition for 99.9% initial mass to comminute to m5
        m99_9(n)=m5(i);
        t99_9(n)=t(i-1);
        break
    end
end
```

5. Evolution of result with variation of time step

With the decrease in time step, time required for 99.9% mass to comminute to m5 category is observed to have changed drastically initially and with further decrease in time step up to 10^{-6}, the result stabilizes with an error of order 10^{-2}. This can be observed in the figure below. The first figure suggests that the result does not vary significantly by decreasing the time step from 10^{-5} to 10^{-6}.



Decrease in time step decreases computation efficiency but increases accuracy. In the second plot above, the comparative error is the difference of result between the subsequent time steps. Each iteration number corresponds to each change in time step from 10^{-2} to 10^{-6}.

```
% Error calculation by comparing with the previous iteration result
if(n>=2)
    err(n)=t99_9(n)-t99_9(n-1);
    itn(n)=n-1;
end
```

The change in time step from 10^{-5} to 10^{-6} (4^{th} iteration) results in a comparative error of the order 10^{-2} which is acceptable. **Thus the optimal time step is thus determined as 10^{-5}**.