Chapter 2

Friction Loss Chart

An ASHRAE like Friction loss chart is generated by solving the equations discussed in chapter 1.

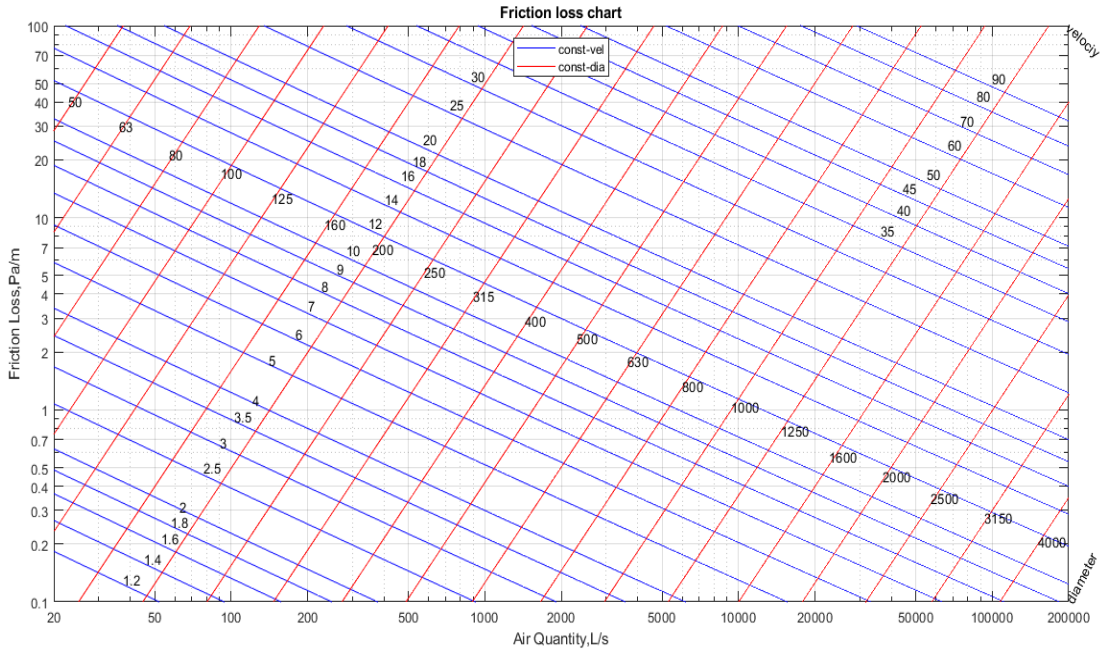
2.1 MATLAB code to generate ASHRAE friction loss chart

```
close all
clear all

e=0.09e-3;              %roughness in SI
q=20:50:250000;     %flow rate in l/s
mu=1.81e-5;          %viscosity in SI
rho=1.2;               %density in SI
z=length(q)

%axis labels values
yax=[0.1 0.2 0.3 0.4 0.5 0.7 1 2 3 4 5 7 10 20 30 40 50 70 100];
xax=[20 50 100 200 500 1000 2000 5000 10000 20000 50000 100000 200000];

v=[1.2 1.4 1.6 1.8 2.0 2.5 3 3.5 4 5 6 7  8 9 10 12 14 16 18 20 25 30 35 40 45 50 60 70 80 90 ];
for j=1:length(v)
   w=1;
   for i=1:z
      %reynolds no.
      d=sqrt(q(i)*4/(pi*v(j)*1000));
      re=rho*v(j)*d/mu;
     if re<2300
         f=64/re;
      else
        % friction factor by  Churchill equation
         b=(37530/re)^16;
         a=(2.457*log(1/((7/re)^0.9+0.27*e/d)))^16;
         f=8*((8/re)^12+1/(a+b)^1.5)^(1/12);
      end

      %friction loss
      p=f*rho*v(j)^2/(2*d);
      Dp(w)=p;
      fl(w)=q(i);
      vel(w)=v(j);
      w=w+1;
   end
   datatable=table(fl',Dp',vel');
   datatable.Properties.VariableNames= {'flow' 'loss' 'velocity'}
   flo=datatable.flow;
   l=datatable.loss;

   %friction loss chart
   figure(1)
   h1=loglog(flo,l,'b')
   xlim([20 200000])
   ylim([0.1 100])
  % label the axis and change the axis values
   ax = gca;
   ax.XTick = xax
   ax.YTick = yax
   title('Friction loss chart')
```

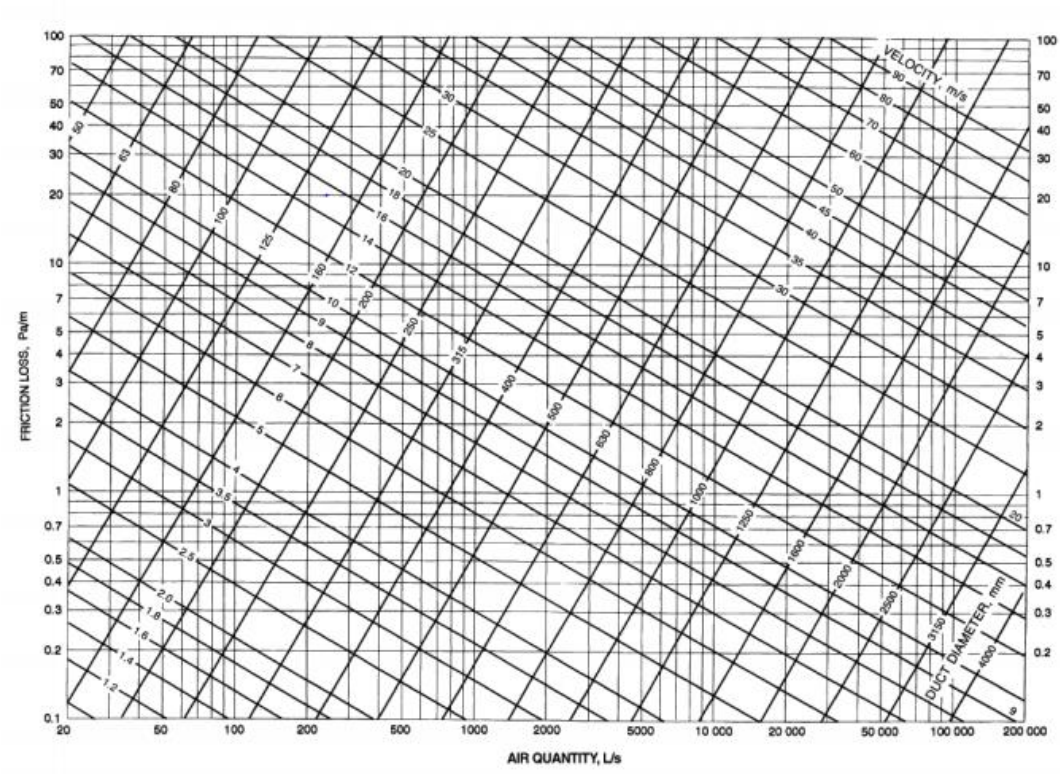
```
   xlabel('Air Quantity,L/s')
   ylabel('Friction Loss,Pa/m')
   hold on
   grid on
 end

%diameter in mm
d=[50 63 80 100 125 160 200 250 315 400 500 630 800 1000 1250 1600 2000 2500 3150 4000];
for j=1:length(d)
   w=1;
   for i=1:z
      % reynolds no.
      v=4*q(i)*1000/(pi*d(j)*d(j));
      re=rho*d(j)*v/mu;

      if re<2300
         f=64/re;
      else
         % friction factor by Churchill equation
         b=(37530/re)^16;
         a=(2.457*log(1/((7/re)^0.9+0.27*e*1000/d(j))))^16;
         f=8*((8/re)^12+1/(a+b)^1.5)^(1/12);
      end

      %friction loss
      p=f*rho*v^2*1000/(2*d(j));
      Dp(w)=p;
      fl(w)=q(i);
      dia(w)=d(j);
      w=w+1;
   end
   datatable=table(fl',Dp',dia');
   datatable.Properties.VariableNames= {'flow' 'loss' 'diameter'};
   flo=datatable.flow;
   l=datatable.loss;
   h2=loglog(flo,l,'r')
   hold on
end
% text
ylim=get(gca,'ylim');
xlim=get(gca,'xlim');
txt=text(xlim(2),ylim(1),'diameter')
txt1=text(xlim(2),ylim(2),'velocity')
angle=60;
angle1=320;
set(txt,'Rotation',angle)
set(txt1,'Rotation',angle1)

% labelling constant velocity and diameter lines
arr1=[1.2,1.4,1.6, 1.8, 2.0, 2.5, 3, 3.5, 4, 5,6,7, 8, 9, 10, 12, 14, 16, 18, 20, 25, 30, 35, 40, 45, 50, 60, 70, 80, 90 ];
for m=1:length(arr1)
  f=string(arr1);
  gtext(f(m));
end
arr2=[50,63,80,100,125,160,200,250 ,315, 400, 500, 630, 800, 1000, 1250, 1600, 2000, 2500, 3150, 4000];
for m=1:length(arr2)
  f=string(arr2);
  gtext(f(m));
end

 %legend
legend(([h1(1), h2(1)]),'const-vel','const-dia','location','north')
```

2.2 Friction loss chart

Following Friction loss chart is generated from the above MATLAB code.



(a)



(b)

Fig 2.1 Friction loss chart from (a) our MATLAB code (b) ASHRAE Handbook

Chapter 3

Mobile App

ASHRAE Friction Loss Chart was reproduced successfully with the help of MATLAB code in Chapter 2. W the help of basic equations discussed in section 1.1, it is being coded in java to calculate friction factor and converted into a Mobile App.

3.1 JAVA Code and its structure

The Code is divided into two parts i.e., for

1. Circular cross-section
2. Rectangular cross-section

3.1.1 Round Duct

In circular cross-section it is further subdivided into two options based on data being fed.

a. flow rate(l/s) and velocity(m/s)
b. flow rate(l/s) and diameter(mm)

Now after taking input from the user, we are giving flow rate(l/s), velocity(m/s), diameter(mm) and friction loss (Pa/m) as output.

3.1.2 Rectangular Duct

In rectangular cross-section it is further subdivided into two options based on data being fed.

a. flow rate(l/s), velocity and one of the dimensions
b. flow rate(l/s), dimension 1 and dimension 2

Now after taking input from the user we are giving flow rate(l/s), velocity(m/s), dimension1 and dimension 2 and friction loss as output.

3.1.3 Activities

For the Backend portion, it is being coded in JAVA language and there are five activities used in the app which are linked to corresponding XML layout.

1. MainActivity.java

   This activity is linked to Mainactivity.xml and gets the intent of user to select type of duct
2. CircularActivity.java

   Gets user input into different editable text fields and buttons for circular duct.
3. CircularActivityResults.java