Appendix A

```
   a=app.amEditField.Value; b=app.bmEditField.Value;
Distance_Rows=app.LateralspacingmEditField.Value; layout=app.LayoutEditField.Value;
D=app.LateraldiametermEditField.Value; A=(pi/4)*D^2;
s=app.EmitterspacingmEditField.Value; L=0.25*a; L=(round(L/s))*s;
qav=app.AveragedischargeofemitterLhrEditField.Value;
Hav=app.AverageemitterheadmEditField.Value;
y=app.EmitterexponentEditField.Value; n=round(L/s+1); v=1.01*10^-6; S=0;
Qmax=qav*(L/s+1); z=0;
B=3/(2*9.81*A^2); E=8*s/(pi^2*9.81*D^5);
c=qav/Hav^y;
q=zeros(1,n); Q=zeros(1,n); H=zeros(1,n);
Hmax=Hav+0.01; Vmax=Qmax*10^-3/(3600*A);

% manifold data
% No. of laterals in selected-2-calculate quarter
if layout==1
m=floor((b/2)/Distance_Rows)+1;
elseif layout==2
m=floor(b/Distance_Rows)+1;
end
%Manifold Diameter
Max_Manifold_Discharge=2*qav*n*m; %in m^3/s
%Max_Acceptable_Velocity=0.2;
%Manifold_diameter=(4*Max_Manifold_Discharge/(Max_Acceptable_Velocity*pi))^0.5;
Manifold_diameter=app.ManifoldDiametermEditField.Value;
Manifold_Area=(pi/4)*(Manifold_diameter)^2;
Relative_roughness=0.003*10^-3/Manifold_diameter;
%Up2Down||Bending
%Bending_Coefficient=1; Up_Down=0.5;

Diameter_submain=app.SubmainDiametermEditField.Value;
Area_submain=pi*Diameter_submain^2/4;

Diameter_main=app.MainDiametermEditField.Value; Area_main=pi*Diameter_main^2/4;

while z==0
Q1=Qmax; H1=Hmax; x=0.; deltad=0.01;
q1=c*H1^y;
q(1)=q1; Q(1)=Q1; H(1)=H1;
%loop
%First Inner Loop
for i=1:n
    x=x+s;
    Y=round((x),3);
    Z=round((L+s),3);
    if Z==Y
     disp('Alright then')
    end

    if (Y==Z)
       if Y<Z
            Hmax =Hmax +deltad;
      else
            Q2=Q1-q1;
            V=Q2*10^-3/(3600*A);
```

```
            Vrel=V/Vmax;
            if Vrel<0
                Hmax=Hmax-deltad;
            elseif Vrel>0 && Vrel<0.00666
                z=1;

                break;

            else
                Hmax=Hmax+deltad;
            end
        end
    else
   Q2=Q1-q1; Re=4*Q2*10^-3/(3600*pi*D*v);
   if Re<2400
       f=64/Re;
   elseif Re>2400 && Re<=10^5
       f=0.316*Re^-0.25;
   else
       f=130*Re^-0.172;
   end
   H2=H1+B*(10^-3/3600)^2*(Q1^2-(Q1-q1)^2)-E*f*(10^-3/3600)^2*(Q1-q1)^2+s*S;
   if H2<0
        if x< L+s
            Hmax =Hmax +deltad;
        else
            Q2=Q1-q1;
            V=Q2*(10^-3/3600)/A;
            Vrel=V/Vmax;
            if Vrel<0
                Hmax=Hmax-deltad;
            elseif Vrel>0 && Vrel<0.00666
                z=1;
                break;
            else
                Hmax=Hmax+deltad;
            end
        end
   else
       q2=c*H2^y;
       q(i+1)=q2; Q(i+1)=Q2; H(i+1)=H2;
       H1=H2; Q1=Q2; q1=q2;
   end
    end
end

end
```

```
                Q1=Qmax; H1=Hmax; x=0.; q1=c*H1^y;
                Hrel1=Hmax/Hav;
                Vrel=1; qrel1=q1/qav;
                Re=4*Q1*10^-3/(3600*pi*D*v);
               if Re<2400
                   f=64/Re;
               elseif Re>2400 && Re<=10^5
                   f=0.316*Re^-0.25;
               else
```

```
                    f=130*Re^-0.172;
                end
                %Matrices for plotting.
                X_final=zeros(1,n); Vrel_final=zeros(1,n); Hrel_final=zeros(1,n);
qrel_final=zeros(1,n); R_final=zeros(1,n);
                Vrel_final(1)=1;
                X_final(1)=0;
                Hrel_final(1)=Hrel1;
                qrel_final(1)=qrel1;
                R_final(1)=Re;
                q(1)=q1; Q(1)=Q1; H(1)=H1;
                %starting from fiirst emitter .

                 x=x+s;
                for j=1:n-1
                 Q2=Q1-q1;
                 V=Q2*10^-3/(3600*A);
                 Re=4*Q2*10^-3/(3600*pi*D*v);
                if Re<2400
                    f=64/Re;
                elseif Re>2400 && Re<=10^5
                    f=0.316*Re^-0.25;
                else
                    f=130*Re^-0.172;
                end

                H2=H1+B*(10^-3/3600)^2*(Q1^2-(Q1-q1)^2)-E*f*(10^-3/3600)^2*(Q1-q1)^2+s*S;
                q2=c*H2^y;
                %Storing Matrices Data @ Posterior Location
                Vrel_final(j+1)=V/Vmax;
                X_final(j+1)=x;
                Hrel_final(j+1)=H2/Hav;
                qrel_final(j+1)=q2/qav;
                R_final(j+1)=Re;
                q(j+1)=q2; Q(j+1)=Q2; H(j+1)=H2;
                x=x+s;
                if x>L
                    UC=1-(sum(abs(q-qav)))/(n*qav);
                    %DU_ = symsum(q, q, [3*n/4 n]);


                else
                    Q1=Q2;
                    H1=H2;
                    q1=q2;
                end
                 end


%Requires Variables for Data storage| Dicharges From Different Emitters | entering with
Hmax From Sec2 Execution.
H=zeros(1,m);
H(1)=Hmax;
Emitters_Flow=zeros(m,n); Emitters_Flow(1,:)=q;
Relative_Velocity=zeros(m,n); Relative_Head=zeros(m,n); Relative_Discharge=zeros(m,n);%%
Manifold_Reynolds_No=zeros(1,m);
Manifold_friction_factor=zeros(1,m);
```

```
Max_Velocity_Laterals_input=zeros(1,m);
% Loop passing over all Quarter Laterals
 for i=1:m-1
     Max_Velocity_Laterals_input(1)=Vmax; % HINT CHECH FOR ITS UPDATE | Update for
different laterals
     %Calculating Reynolds No. | loopy Friction factor | Darcy Weibach-
     %Friction Factor
     Manifold_Reynolds_No(i+1)=4*(2*sum(sum(Emitters_Flow)))*10^-
3/(3600*pi*Manifold_diameter*v);
     %Darcy-Weibach Friction Factor.

Manifold_friction_factor(i+1)=(0.3086/(log((6.9/Manifold_Reynolds_No(i))+(Relative_roughn
ess/3.7)^1.11))^2);
     %Remember to modify last section of the maifold as ther's no up2down
     Discharge_Summation=2*sum(sum(Emitters_Flow));
     %BENDING CALCULATION |
    down_stream_flow=2*sum(sum(Emitters_Flow));
    average_lateral_flow=n*qav;
    up_stream_flow=down_stream_flow+average_lateral_flow;
    upstream_Re=4*up_stream_flow*(10^-3/3600)/(v*pi);
    Up_Down=-0.219*log(upstream_Re)+2.148;
    Flow_Ratio=average_lateral_flow/up_stream_flow;
    Const_a=-34.57*(Flow_Ratio^2)-1.921*(Flow_Ratio)-0.12;
    Const_b=494*(Flow_Ratio^2)+40.71*(Flow_Ratio^2)+3.08;
    Bending_Coefficient=Const_a*log(upstream_Re)+Const_b;
     %|--------------------------------|
     H(i+1)=H(i)+((Bending_Coefficient+Up_Down)*(((Discharge_Summation)^2)*((10^-
3)/3600)^2)/(2*9.81*(Manifold_Area)^2))+(Manifold_friction_factor(i))*(Distance_Rows/Mani
fold_diameter)*(((Discharge_Summation)^2)*((10^-3)/3600)^2)/(2*9.81*(Manifold_Area)^2);
     % Now it's Required to Calculate Emitters' dicharges @ H(i+1)
     %|-------------------------------------------------------------|
     % in this section a loop required to update Qmax as Q1,Q2 become
     % nagative
     for k=1:200
                Q1=Qmax; H1=H(i+1); x=0.; q1=c*H(i+1)^y;
                Max_Velocity_Laterals_input(i+1)=Qmax*((10^-3)/3600)/((pi/4)*D^2);
                Hrel1=H1/Hav; Emitters_Flow(i+1,1)=q1;
                qrel1=q1/qav;
                Re=4*Q1*10^-3/(3600*pi*D*v);
               %Smooth Lateral Pipe of Diameter D || Estimating Re No.| f
               if Re<2400
                   f=64/Re;
               elseif Re>2400 && Re<=10^5
                   f=0.316*Re^-0.25;
               else
                   f=130*Re^-0.172;
               end
               %Useful Matrices for visualization|x,f,vrel,qrel
               Emitter_Distances=zeros(m,n);  %Relative_Reynolds=zeros(m,n);
               Relative_Velocity(1,:)=Vrel_final;
               Emitter_Distances(1,:)=X_final;
               Relative_Head(1,:)=Hrel_final;
               Relative_Discharge(1,:)=qrel_final;
               %=====> Relative_Reynolds(1)=Re;
               % Update m-1 Laterals Head,Dischares
               Dummy_Emitter_Discharge=zeros(1,n);
               Dummy_Lateral_Discharges=zeros(1,n);
```

```
                Dummy_Relative_Velocity=ones(1,n);
                Dummy_relative_head=zeros(1,n);
                Laterals_Heads=zeros(1,n);
                Dummy_Emitter_Discharge(1)=q1;
                Dummy_Lateral_Discharges(1)=Q1;
                Laterals_Heads(1)=H(i+1);
                % q(1)=q1; Q(1)=Q1; H(1)=H(i+1);
                %starting from Second Emitter .
                x=s;
                % Looping over the whole lateral
                % Here Another loop must be added to converge lateral
                % dicharge as it's higher than Qmax
                for j=1:n-1
                    % Finding Discharge and Velocity @ Next Emitter
                 Q2=Q1-q1;
                 V=Q2*10^-3/(3600*A);
                    % Estimating Re No. || Friction Factor
                 Re=4*Q2*10^-3/(3600*pi*D*v);
                if Re<2400
                    f=64/Re;
                elseif Re>2400 && Re<=10^5
                    f=0.316*Re^-0.25;
                else
                    f=130*Re^-0.172;
                end
                    %Updating Head for the next lateral
                H2=H1+B*(10^-3/3600)^2*(Q1^2-(Q1-q1)^2)-E*f*(10^-3/3600)^2*(Q1-q1)^2+s*S;
                q2=c*H2^y;
            %  Storing Matrices Data @ Posterior Location
            %  Vrel_final(j+1)=V/Vmax;
            %  X_final(j+1)=x;
            %  Hrel_final(j+1)=H2/Hav;
            %  qrel_final(j+1)=q2/qav;
            %  R_final(j+1)=Re;
                Dummy_Emitter_Discharge(j+1)=q2; Dummy_Lateral_Discharges(j+1)=Q2;
Laterals_Heads(j+1)=H2;
                Dummy_Relative_Velocity(j+1)=V/Max_Velocity_Laterals_input(i+1);
                %Update Emitter Loaction
                x=x+s;
                if x>L
                    % Calculating Uniformity Coefficient for the lateral
                    % UC=1-(sum(abs(q-qav)))/(n*qav);
                    if Dummy_Lateral_Discharges(n)<0
                       % Add Extra flow for the lateral input to suit its
                       % demand
                       Qmax=Qmax+qav;
                       fprintf('qav is added to balance the discharge @ iteration %d For
lateral %d \n' ,k,i);
                    else
                        % Last emitter is reached and Updating Matrices is
                        % required .
                        fprintf('UPDATING DATA \n');
                        Emitters_Flow(i+1,:)=Dummy_Emitter_Discharge;
                        Relative_Velocity(i+1,:)=Dummy_Relative_Velocity;
                        Relative_Head(i+1,:)=Laterals_Heads./Hav;
                        Relative_Discharge(i+1,:)=Dummy_Emitter_Discharge./qav;
```

```
                    end

                else
                    %Updating Next Emitter inputs.
                    Q1=Q2;
                    H1=H2;
                    q1=q2;
                end
                end
                %Escaping lateral loop at apropraiate discarge.
                if Dummy_Lateral_Discharges(n)>0

                    fprintf('Lateral No. %d |Loop is @ iteration %d |Accumulated
Dischargeat last Section is %d \n' ,i,k,Dummy_Lateral_Discharges(n));
                    break;
                end
     end
     %|------------------------------------------------------------|
 end

 const_Bending_Coefficient=1;
```

Submain and Main calculations

```
 if layout==1
    Discharge_two_quarters=2*Discharge_Summation;
    Submain_Reynolds_No=4*(Discharge_two_quarters)*10^-3/(3600*pi*Diameter_submain*v);

Submain_friction_factor=(0.3086/(log((6.9/Submain_Reynolds_No)+(Relative_roughness/3.7)^1
.11))^2);
    Head_two_quarters=H(end)+const_Bending_Coefficient*(Discharge_two_quarters*(10^-
3/3600))^2/(2*9.81*Area_submain^2)...
    +Submain_friction_factor*((0.25*a)/Diameter_submain)*(Discharge_two_quarters*(10^-
3/3600))^2/(2*9.81*Area_submain^2);


    Main_discharge= 2*Discharge_two_quarters;
    Main_Reynolds_No=4*(Main_discharge)*10^-3/(3600*pi*Diameter_main*v);

Main_friction_factor=(0.3086/(log((6.9/Main_Reynolds_No)+(Relative_roughness/3.7)^1.11))^
2);
    Head_main=Head_two_quarters+const_Bending_Coefficient*(Main_discharge*(10^-
3/3600))^2/(2*9.81*Area_main^2)...
    +Main_friction_factor*((0.5*b)/Diameter_main)*(Main_discharge*(10^-
3/3600))^2/(2*9.81*Area_main^2);
         %surfing
        flip_Emitters_Flow=fliplr(Emitters_Flow);
        flip_Relative_Head=fliplr(Relative_Head);
        All_Land_Discharge=zeros(2*m,4*n);
        All_Land_Relative_Head=zeros(2*m,4*n);
        Quarter_Flow=zeros(m,2*n);
        Quarter_Relative_Head=zeros(m,2*n);
        Quarter_Flow(:,1:n)=flip_Emitters_Flow;
        Quarter_Relative_Head(:,1:n)=flip_Relative_Head;
        Quarter_Flow(:,n+1:2*n)=Emitters_Flow;
        Quarter_Relative_Head(:,n+1:2*n)=Relative_Head;
```

```
        flip_Quarter_Flow=flipud(Quarter_Flow);
        flip_Relative_Head=flipud(Quarter_Relative_Head);
        Half_Land_Discharge=zeros(2*m,2*n);
        Half_Land_Relative_Head=zeros(2*m,2*n);
        Half_Land_Discharge(1:m,:)=Quarter_Flow;
        Half_Land_Relative_Head(1:m,:)=Quarter_Relative_Head;
        Half_Land_Discharge(m+1:2*m,:)=flip_Quarter_Flow;
        Half_Land_Relative_Head(m+1:2*m,:)=flip_Relative_Head;
        All_Land_Discharge(:,1:2*n)=fliplr(Half_Land_Discharge);
        All_Land_Relative_Head(:,1:2*n)=fliplr(Half_Land_Relative_Head);
        All_Land_Discharge(:,2*n+1:4*n)=Half_Land_Discharge;
        All_Land_Relative_Head(:,2*n+1:4*n)=Half_Land_Relative_Head;
        Emitters_Number=4*(m-1)*n+4*m*n;
        Laterals_Number=4*m+(m-1)*4;
        Lateral_Length=Laterals_Number*L;

 elseif layout==2
    Discharge_two_quarters=Discharge_Summation;
    Submain_Reynolds_No=4*(Discharge_two_quarters)*10^-3/(3600*pi*Diameter_submain*v);

Submain_friction_factor=(0.3086/(log((6.9/Submain_Reynolds_No)+(Relative_roughness/3.7)^1
.11))^2);
    Head_two_quarters=H(end)+const_Bending_Coefficient*(Discharge_two_quarters*(10^-
3/3600))^2/(2*9.81*Area_submain^2)...
    +Submain_friction_factor*((0.25*a)/Diameter_submain)*(Discharge_two_quarters*(10^-
3/3600))^2/(2*9.81*Area_submain^2);

    Main_discharge=2*Discharge_two_quarters;
    Main_Reynolds_No=4*(Main_discharge)*10^-3/(3600*pi*Diameter_main*v);

Main_friction_factor=(0.3086/(log((6.9/Main_Reynolds_No)+(Relative_roughness/3.7)^1.11))^
2);
    Head_main=Head_two_quarters+const_Bending_Coefficient*(Main_discharge*(10^-
3/3600))^2/(2*9.81*Area_main^2)...
    +Main_friction_factor*((0.5*a)/Diameter_main)*(Main_discharge*(10^-
3/3600))^2/(2*9.81*Area_main^2);
        %surfing
        flip_Emitters_Flow=fliplr(Emitters_Flow);
        flip_Relative_Head=fliplr(Relative_Head);
        All_Land_Discharge=zeros(m,4*n);
        All_Land_Relative_Head=zeros(m,4*n);
        All_Land_Discharge(:,1:n)=flip_Emitters_Flow;
        All_Land_Relative_Head(:,1:n)=flip_Relative_Head;
        All_Land_Discharge(:,n+1:2*n)=Emitters_Flow;
        All_Land_Relative_Head(:,n+1:2*n)=Relative_Head;
        All_Land_Discharge(:,2*n+1:3*n)=flip_Emitters_Flow;
        All_Land_Relative_Head(:,2*n+1:3*n)=flip_Relative_Head;
        All_Land_Discharge(:,3*n+1:4*n)=Emitters_Flow;
        All_Land_Relative_Head(:,3*n+1:4*n)=Relative_Head;
        Emitters_Number=4*m*n;
        Laterals_Number=4*m;
        Lateral_Length=Laterals_Number*L;

 end
            app.TotaldischargeLhrEditField.Value=Main_discharge;
            app.TotalheadmEditField.Value=Head_main;
            app.LaterallengthmEditField.Value=L;
```

```
            app.EmittersNumberEditField.Value=Emitters_Number;
            app.LateralsNumberEditField.Value=Laterals_Number;
            app.TotalLateralLengthmEditField.Value=Lateral_Length;
```

Published with MATLAB® R2019a