```
clc;
clear;
%close all;

conc = linspace(0,1,100);
gamma = -0.8757;
D0 = 6.22e-4;%0.585e-3;
rhoDensity = 2.157;
DiffCoeff = D0./(-gamma*conc+1).^2;
figure, plot(conc,DiffCoeff);
title(strcat('Diffusion Coefficient curve, \gamma = ',num2str(gamma)));
xlabel('Conc of bound iodine in v/v'), ylabel('Diffusion coefficient
(mm^2/s)');

close all;
%% Load Area plots and load concentration profiles

%dateexp = '20180409';
% dateexp = '20180621';
%dateexp = '20180820';
dateexp = 'MinMax';

if strcmp(dateexp,'20180409')
    disp('Reading 20180409 data')
    load('C:\Users\kw8745\Dropbox\ExtraData\20180409\Areat');
    load('C:\Users\kw8745\Dropbox\ExtraData\20180409\Aream');
    load('C:\Users\kw8745\Dropbox\ExtraData\20180409\Areav');
    A =
load('C:\Users\kw8745\Dropbox\ExtraData\20180409\STconc_oldCalib');
    B =
load('C:\Users\kw8745\Dropbox\ExtraData\20180409\SVconc_oldCalib');
    C =
load('C:\Users\kw8745\Dropbox\ExtraData\20180409\SMconc_oldCalib');
    load('C:\Users\kw8745\Dropbox\ExtraData\20180820\Reissmem');
    load('C:\Users\kw8745\Dropbox\ExtraData\20180820\BM');
    load('D:\Dropbox\Kevin scans\20180621\SMSLmem');
elseif strcmp(dateexp,'20180621')
    disp('Reading 20180621 data')
    load('D:\Dropbox\Kevin scans\20180621\Areat');
    load('D:\Dropbox\Kevin scans\20180621\Aream');
    load('D:\Dropbox\Kevin scans\20180621\Areav');
    A = load('D:\Dropbox\Kevin scans\20180621\STconc_oldCalib');
    B = load('D:\Dropbox\Kevin scans\20180621\SVconc_oldCalib');
    C = load('D:\Dropbox\Kevin scans\20180621\SMconc_oldCalib');
    load('D:\Dropbox\Kevin scans\20180621\Reissmem');
    load('D:\Dropbox\Kevin scans\20180621\BM');
    load('D:\Dropbox\Kevin scans\20180621\SMSLmem');
    BM(80:end) = BM(80);
elseif strcmp(dateexp,'20180820')
    disp('Reading 20180820 data')
    load('C:\Users\kw8745\Dropbox\ExtraData\20180820\Areat');
    load('C:\Users\kw8745\Dropbox\ExtraData\20180820\Aream');
    load('C:\Users\kw8745\Dropbox\ExtraData\20180820\Areav');
    A =
load('C:\Users\kw8745\Dropbox\ExtraData\20180820\STconc_oldCalib');
    B =
load('C:\Users\kw8745\Dropbox\ExtraData\20180820\SVconc_oldCalib');
```

```
    C =
load('C:\Users\kw8745\Dropbox\ExtraData\20180820\SMconc_oldCalib');
    load('C:\Users\kw8745\Dropbox\ExtraData\20180820\Reissmem');
    load('C:\Users\kw8745\Dropbox\ExtraData\20180820\BM');
    load('D:\Dropbox\Kevin scans\20180621\SMSLmem');
else
    disp('Reading 20180621 data')
    load('D:\Dropbox\Kevin scans\20180621\Areat');
    load('D:\Dropbox\Kevin scans\20180621\Aream');
    load('D:\Dropbox\Kevin scans\20180621\Areav');
    A = load('D:\Dropbox\Kevin scans\20180621\STconc_oldCalib');
    B = load('D:\Dropbox\Kevin scans\20180621\SVconc_oldCalib');
    C = load('D:\Dropbox\Kevin scans\20180621\SMconc_oldCalib');
    load('D:\Dropbox\Kevin scans\20180621\Reissmem');
    load('D:\Dropbox\Kevin scans\20180621\BM');
    load('D:\Dropbox\Kevin scans\20180621\SMSLmem');
end
%Aream(80:end) = Aream(80)*ones(101-80+1,1); for 0327
STArea = Areat;
SMArea = Aream;
SVArea = Areav;

STdist = distt;
SMdist = distm;
SVdist = distv;%SVdist;

STconc = 2*A.STconc_oldCalib; SVconc = 2*B.SVconc_oldCalib; SMconc =
2*C.SMconc_oldCalib;

% subtract baseline 'concentration'
STconc = STconc(:,2:end)-repmat(STconc(:,1),[1,5]);
SVconc = SVconc(:,2:end)-repmat(SVconc(:,1),[1,5]);
SVconc = SVconc-repmat(SVconc(:,1),[1,5]);
SVconc(SVconc<0)=2;
SMconc = SMconc(:,2:end)-repmat(SMconc(:,1),[1,5]);

%Getting rid of sharp spikes in BM
Basimem = BM;
SMSLmem = smooth(SMSLmem,0.2,'lowess')';
Reissmem = smooth(Reissmem,0.1,'lowess');
BM = smooth(BM,0.2,'lowess');

% Interpolate area and concentration profiles so that 0% is at max conc
% and anything less than that is taken to be -ve percent
% a: ST
% STconc: conc from point of max conc to 100%
% STwholeOrigConcProfile: whole conc profile as extracted from uCT

STwholeOrigConcProfile = STconc;
[~,id] = max(STconc(:,end));

% split the conc profiles; area vs distance plots; and membrane(BM)
plots at
% the point where concentration is maximum in the associated scala so
we
% get 2 sets of plots for each (same as in scala concentration plots).
% The portions to the left are labelled with
```

```
% _left and those to the right just use regular names. Then
reinterpolate
% them so that the portion to the right are between 0 to 100%
%% ST scala
STconc = STconc(id:end,:);
STleng_left = round(100*id/(100-id)); %number of points in ST to the
left of delivery point
STdist = linspace(0,STdist(end),101); %distance array from delivery
point to apex
STArea_left = STArea(1:id-1); %Area at every point in distance array
from del point to CA
STArea_left = interp1(STdist(1:id-
1),STArea_left,linspace(STdist(1),STdist(id-1),STleng_left-1));
STArea_left = STArea_left(end:-1:1);
STArea = STArea(id:end);
STArea =
interp1(STdist(id:end),STArea',linspace(STdist(id),STdist(end)))';

STconc =
interp1(STdist(id:end)',STconc,linspace(STdist(id),STdist(end),101));
%a = interp1(distt2',a,linspace(distt2(1),distt2(end),100));
BM(14) = BM(15);
BM = smooth(BM,0.2);
BMleft = BM(1:id-1);
BM = BM(id:end);
BMleft = interp1(STdist(1:id-1),BMleft,linspace(STdist(1),STdist(id-
1),STleng_left-1))';
BMleft = BMleft(end:-1:1);
BM = interp1(STdist(id:end),BM,linspace(STdist(id),STdist(end)))';
STdist = linspace(STdist(id),STdist(end),100);

%% SV scala
SVWholeOrigConcProfile = SVconc;
[~,id] = max(SVconc(:,end));

if strcmp(dateexp,'20180327')
    id = 29;
elseif strcmp(dateexp,'20180621')
    id = id-10;
elseif strcmp(dateexp,'20180820')
    id = id-15;
else
    id = 25;
end
SVdist = linspace(0,SVdist(end),101);
SVconc = SVconc(id:end,:);
SVArea = smooth(SVArea);


%Areavwhole = smooth(Areavwhole);
SVleng_left = round(100*id/(100-id));
SVdist = SVdist(id:end-1);
%Areav = Areavwhole(id:end-1);
SVArea_left = SVArea(1:id-1);
SVArea = SVArea(id:end-1);

SVArea_left = interp1(SVdist(1:id-
```

```
1),smooth(SVArea_left),linspace(SVdist(1),SVdist(id-1),SVleng_left-1));
SVArea_left = SVArea_left(end:-1:1);
SVconc = interp1(id:101,SVconc,linspace(id,100,100));
SVArea =
interp1(SVdist,smooth(SVArea'),linspace(SVdist(1),SVdist(end)))';

SVdist = linspace(SVdist(1),SVdist(end),100);

%% SM scala
[~,id] = max(SMconc(:,end));
if strcmp(dateexp,'20180327')
    id = 25;%20180327
elseif strcmp(dateexp,'20180621')
    id = id-7;
elseif strcmp(dateexp,'20180820')
    id = 3; %id = id-5;
end

SMArea = smooth(SMArea,'lowess',0.3);
SMWholeOrigConcProfile = SMconc;
SMconc = SMconc(id:end,:);
SMleng_left = round(100*id/(100-id));
SMdist = linspace(0,SMdist(end),101);
if id>1
    SMArea_left = SMArea(1:id-1);
    SMArea_left = interp1(SMdist(1:id-
1),SMArea_left,linspace(SMdist(1),SMdist(id-1),SMleng_left-1));
    SMArea_left = SMArea_left(end:-1:1);
end
SMArea = SMArea(id:end);
SMArea =
interp1(SMdist(id:end),SMArea',linspace(SMdist(id),SMdist(end)))';
SMconc =
interp1(SMdist(id:end)',SMconc,linspace(SMdist(id),SMdist(end),101));
Reissmem = smooth(Reissmem,0.2);
if id>1
    Reissmem_left = Reissmem(1:id-1);
    Reissmem_left = interp1(SMdist(1:id-
1),Reissmem_left,linspace(SMdist(1),SMdist(id-1),SMleng_left-1))';
    Reissmem_left = Reissmem_left(end:-1:1);
end
Reissmem = Reissmem(id:end);
Reissmem =
interp1(SMdist(id:end),Reissmem,linspace(SMdist(id),SMdist(end)))';

BasiMem = BM;

SMSLmem = smooth(SMSLmem,0.2);
if id>1
    SMSLmem2 = SMSLmem(1:id-1);
    SMSLmem2 = interp1(SMdist(1:id-
1),SMSLmem2,linspace(SMdist(1),SMdist(id-1),SMleng_left-1))';

    SMSLmem2 = SMSLmem2(end:-1:1);
end
SMSLmem = SMSLmem(id:end);
SMSLmem =
```

```
interp1(SMdist(id:end),SMSLmem,linspace(SMdist(id),SMdist(end)))';

%% Initialization of the dx and dt for the forward model
% ha, hb, hc: unit length for ST, SV, SM resp.
dx_ST = STdist(2) - STdist(1); dx_SV = SVdist(2) - SVdist(1);
dx_SM = SMdist(2) - SMdist(1);

% save a, b, c for plotting later
STorig = STconc; SVorig = SVconc; SMorig = SMconc;

ConcAt0 = zeros(101,6);

trunc = 100; %truncate profile at less than 100 percent if needed
i = 1;
STconc = STconc(1:trunc,size(STconc,2)-i:size(STconc,2));SVconc =
SVconc(1:trunc,size(STconc,2)-i:size(STconc,2));SMconc =
SMconc(1:trunc,size(STconc,2)-i:size(STconc,2));
STArea = STArea(1:trunc); SVArea = SVArea(1:trunc); SMArea =
SMArea(1:trunc);
STArea = smooth(STArea,0.2,'lowess'); SMArea =
smooth(SMArea,0.2,'lowess'); SVArea = smooth(SVArea,0.2,'lowess');
Nst = length(STconc); Nsv=length(SVconc); Nsm = length(SMconc);

RM = Reissmem;
RM = RM(1:trunc); BM = BM(1:trunc);

SMSLmem(91:end) = SMSLmem(90)*ones(1,10);

%% Initialize the transport parameters.
k_x = [2;0.3;0.4]*1e-3;

if strcmp(dateexp,'20180409')
    k_x = [1;0.52;0.08]*1e-3;
elseif strcmp(dateexp,'20180621')
    k_x = [1.5;0.2;0.05].*1e-3;
elseif strcmp(dateexp,'20180820')
    k_x = [1;0.5;0.01].*1e-3;
end

% some experiment Kevin was trying
% xac = k_x(1).*DiffCoeff./D0;
% xbc = k_x(2).*DiffCoeff./D0;
% xcclear = k_x(2).*DiffCoeff./D0;

%% Learning rates before normalizing
lr1 = 1e-6;
lr2 = 5e-7;
lr3 = 5e-7;

%% Forward model with t_on and t_off
AreaT = STArea;
AreaM = SMArea;
TimeStep=0.1; % in seconds
t_on = 1:10; % in minutes
t_off = 10:10:100; % in minutes
```

```
% initializing arrays
min_base = zeros(25,25);
min_apex = zeros(25,25);
max_base =zeros(25,25);
max_apex = zeros(25,25);
min_all = zeros(50,50,550);
base_conc = zeros(25,25,60000);
apex_conc = zeros(25,25,60000);

% Loop begin
for t_on =1%:25
    for t_off = 3%:25%:10:100
        TimeStep=0.1;
        %Nsec = 200*60/T;%
        Nsec = 1*(t_on+t_off)*60/TimeStep; % total number of timesteps
in 1 cycle
        STconc = zeros(100,Nsec);
        SVconc=zeros(100,Nsec); SMconc=SVconc; d=SVconc;

        % To calculate the loss function, we consider only the part
from
        % max to apex. The argument here being that the concentration
to
        % the left is influenced by flow rate/advective flow.
        SVconc_left = zeros(SVleng_left,Nsec);
        SMconc_left = zeros(SMleng_left,Nsec);
        STconc_left = zeros(STleng_left,Nsec);

        % these arrays are used to store solely the diffusion (and not
the
        % transfer) part of the equations used, they are required when
        % computing the inverse model
        STdiffterm = STconc;
        SVdiffterm = SVconc;
        SMdiffterm = SMconc;

        % number of cycles required to complete 1000mins of delivery
        numcycles = ceil(1000/(t_on+t_off));
        for iter = 1:numcycles

            % Time step is 0.1 sec
            TimeStep=0.1;

            % velocity from max to apex, it is set to zero
            velocity = 0*ones(1,Nsec);

            % Setting the velocity or flow rate of solution in ST to
the
            % left of point of delivery.
            % velocity_left sets the flow rate over the entire cycle
            % Also Set initial concentration at point of delivery
            % ConcAt0 is the concentration at the point of delivery;
this
            % is usually a flat line around 240 but can be changed if
            % required

            if strcmp(dateexp,'20180409')
```

```
                % ConcAt0 is multiplied by 2 to get the concentration
of
                % iopamidol. We get conc. of iodine if not multiplied
by 2
                ConcAt0 = 2*270*ones(1,Nsec);

                velocity_left_percycle = 2*2.806e-
2*[2,6.4,7.1,7.34,12.83]/32;
                velocity_left =
[velocity_left_percycle(1)*ones(1,12000),...

linspace(velocity_left_percycle(1),velocity_left_percycle(2),6000),...
                    velocity_left_percycle(2)*ones(1,12000),...

linspace(velocity_left_percycle(2),velocity_left_percycle(3),6000),...
                    velocity_left_percycle(3)*ones(1,12000),...

linspace(velocity_left_percycle(3),velocity_left_percycle(4),6000),...
                    velocity_left_percycle(4)*ones(1,12000),...

linspace(velocity_left_percycle(4),velocity_left_percycle(5),6000),...
                    velocity_left_percycle(5)*ones(1,12000+1)];
            elseif strcmp(dateexp,'20180621')
                ConcAt0 = 2*240*ones(1,Nsec);
                velocity_left = [(1/2)*2*2.806e-
2*ones(1,Nsec/5),(2/2)*2*2.806e-2*ones(1,4*Nsec/5+1)];

            elseif strcmp(dateexp,'20180820')
                ConcAt0 = 2*260*ones(1,Nsec);
                velocity_left = (1)*2*2.806e-2*ones(1,Nsec+1);
            else
                ConcAt0 = 2*240*ones(1,Nsec);
                timespace =
[ones(1,t_on*60/TimeStep),zeros(1,t_off*60/TimeStep)];
                velocity_left = (8/32)*2*2.806e-2*(timespace);% to
change flow rate

            end

            % There isn't enough RAM to run the program for all 1000
            % minutes at one go, so instead, at the beginning of every
            % iteration, the concnetration profiles at the last 2 time
            % instances in the previous cycle are set as the profiles
            % for the first 2 instances in the current cycle
            if iter>1
                STconc(:,1:2) = STconc(:,end-1:end);STconc(:,3:end)=0;
                STconc_left(:,1) = STconc_left(:,end);
                SVconc(:,1:2) = SVconc(:,end-1:end);SVconc(:,3:end)=0;
                SVconc_left(:,1) = SVconc_left(:,end);
                SMconc(:,1:2) = SMconc(:,end-1:end);SMconc(:,3:end)=0;
                SMconc_left(:,1) = SMconc_left(:,end);
            end
            STArea = AreaT;%1*AreaT(end)*ones(100,1)];
            SMArea = AreaM;
            % CSA of tube, so that flow rate can be
            % adjusted at every point (volume flow rate/CSA)
            Area0 = (pi*(55e-3)^2);
```

```
            flag=0;

            % one cycle time. i.e: t_on plus t_off time
            rangetime = 1:Nsec;

            for i = rangetime

                for len=1:100
                    if len<=100
                        DiffcoeffST = D0/(1-gamma*STconc(n,i)*1e-
3/rhoDensity)^2;
                        DiffcoeffSV = D0/(1-gamma*SVconc(n,i)*1e-
3/rhoDensity)^2;
                        DiffcoeffSM = D0/(1-gamma*SMconc(n,i)*1e-
3/rhoDensity)^2;
                        Kst_sm = k_x(1)*BasiMem(len)*1;
%Basimem(n);0.0007
                        Ksm_sv = k_x(2)*Reissmem(len);     %0.001,(7e-
4)
                        Ksm_clear = k_x(3)*SMSLmem(len);   %0.001,
0.00006
                    end

                    % velocity converted into mm/s
                    v = velocity(i)*(pi*(55e-3)^2)/STArea(len);

                    %% Calculating concentrations along length over
time
                    % STdiffterm, SMdiffterm, SVdiffterm are those
terms
                    % that contain only the diffusion component of the
                    % concentration, saving them is necessary for the
                    % inverse model
                    % STconc, SMconc and SV conc will be concatenated
with
                    % STconc_left, SMconc_left and SVconc_left and
together
                    % reinterpolated between 0 and 100% for the purpose
of
                    % displaying
                    % for T_on, T_off experiments, concatenation won't
be
                    % needed, instead we can just look at STconc(1,:)
and
                    % STconc(100,:) for basal and apical concentrations

                    %% Concentrations in ST from point of delivery to
apex
                    if len==1

                        STconc(len,i+1) = STconc_left(len,i) +
TimeStep*(DiffcoeffST*((STArea(len) - STArea_left(len))*(STconc(len,i)
- STconc_left(len,i))...
                            +STArea(len)*(STconc_left(len,i) -
2*STconc(len,i) + STconc(len+1,i) )...
                            +STArea(len)*(STconc(len,i)-
STconc_left(len,i))*(DiffcoeffST-D0/(1-gamma*STconc_left(len,i)*1e-
```

```
3/rhoDensity)^2))/(STArea(len)*dx_ST*dx_ST)...
                                -
(1/(1*1*STArea(len)))*(Kst_sm*(STconc(len,i)-SMconc(len,i))));
                        if iter>numcycles-3
                            base_conc(t_off,t_on,i+1+Nsec*(iter-
numcycles+2))=STconc(len,i+1);
                        end
                    end
                    if len>=Nst
                        if len<100
                            STdiffterm(len,i+1) =
(DiffcoeffST*(STArea(len) - STArea(len-1))*(STconc(len,i) - STconc(len-
1,i))...

+DiffcoeffST*STArea(len)*(STconc(len+1,i) - 2*STconc(len,i) +
STconc(len-1,i))...
                                +1*STArea(len)*(STconc(len,i)-
STconc(len-1,i))*(DiffcoeffST-D0/(1-gamma*STconc(len-1,i)*1e-
3/rhoDensity)^2))/(STArea(len)*dx_ST*dx_ST);
                            STconc(len,i+1)=STconc(len,i) +
TimeStep*(STdiffterm(len,i+1)...
                                -velocity(i)*Area0*(STconc(len,i)-
STconc(len-1,i))/(STArea(len)*dx_ST));

                        else
                            STconc(len,i+1)=STconc(len-1,i+1);%
                            if i==199 && len==Nsec-1
                                temp=1;
                            end
                        end

                    elseif len>=2 && len<Nst
                        STdiffterm(len,i+1) = (DiffcoeffST*(STArea(len)
- STArea(len-1))*(STconc(len,i) - STconc(len-1,i))...
                            +DiffcoeffST*STArea(len)*(STconc(len+1,i) -
2*STconc(len,i) + STconc(len-1,i))...
                            +1*STArea(len)*(STconc(len,i)-STconc(len-
1,i))*(DiffcoeffST-D0/(1-gamma*STconc(len-1,i)*1e-
3/rhoDensity)^2))/(STArea(len)*dx_ST*dx_ST);
                        STconc(len,i+1) =  STconc(len,i) +
TimeStep*(STdiffterm(len,i+1)...
                            -
(1/(1*STArea(len)))*(Kst_sm*(STconc(len,i)-SMconc(len,i)))...
                            -velocity(i)*Area0*(STconc(len,i)-
STconc(len-1,i))/(STArea(len)*dx_ST));%...

                        if iter>numcycles-3 && len==Nst-1
                            apex_conc(t_off,t_on,i+1+Nsec*(iter-
numcycles+2))=STconc(len,i+1);
                        end

                        if STconc(len,i+1)<0
                            STconc(len,i+1)=0;
                        end
                    end

                    if i==Nsec-1 && len==199
```

```
                    temp=1;
                end

                %% Concentrations in ST (left), not used during
inverse model
                % Just the forward simulation of the 'left' part
from max
                % to base
                n_left = len;
                if n_left ==1
                    %v2 = velocity2(i)*(pi*(55e-
3)^2)/Areat2(lena2);
                    STconc_left(n_left,i+1) = STconc_left(n_left,i)
+ TimeStep*(DiffcoeffST*((STconc(1,i) - 2*STconc_left(n_left,i) +
STconc_left(n_left+1,i))*STArea_left(n_left)...
                        +(STArea_left(n_left) -
STArea(1))*(STconc_left(n_left,i) -
STconc(1,i)))/(STArea_left(n_left)*dx_ST*dx_ST)...
                        +0-
(1/(1*STArea_left(n_left)))*k_x(1)*BMleft(n_left)*(STconc_left(n_left,i
)-SMconc_left(n_left,i))...
                        -
velocity_left(i)*Area0*(STconc_left(n_left,i)-
ConcAt0(i))/(STArea_left(n_left)*dx_ST));%/(0.001);
                    if STconc_left(n_left,i+1)<0
                        STconc_left(n_left,i+1)=0;
                    end
                elseif n_left<STleng_left
                    %v2 = velocity2(i)*(pi*(55e-
3)^2)/Areat2(lena2);
                    STconc_left(n_left,i+1) = STconc_left(n_left,i)
+ TimeStep*(DiffcoeffST*((STconc_left(n_left+1,i) -
2*STconc_left(n_left,i) + STconc_left(n_left-
1,i))*STArea_left(n_left)...
                        +(STArea_left(n_left) - STArea_left(n_left-
1))*(STconc_left(n_left,i) - STconc_left(n_left-
1,i)))/(STArea_left(n_left)*dx_ST*dx_ST)...
                        +0-
(1/(1*STArea_left(n_left)))*k_x(1)*BMleft(n_left)*(STconc_left(n_left,i
)-SMconc_left(n_left,i))...
                        -
velocity_left(i)*Area0*(STconc_left(n_left,i)-STconc_left(n_left-
1,i))/(STArea_left(n_left)*dx_ST));%/(0.001);
                    if STconc_left(n_left,i+1)<0
                        STconc_left(n_left,i+1)=0;
                    end
                elseif n_left==STleng_left
                    STconc_left(n_left,i+1) = STconc_left(n_left-
1,i+1);
                end
                %end

                %% Concentrations in SM toards apex(right)

                if len==1
                    SMdiffterm(len,i+1) = (DiffcoeffSM*(SMArea(len)
- SMArea_left(len))*(SMconc(len,i) - SMconc_left(len,i))...
```

```
+DiffcoeffSM*SMArea(len)*(SMconc_left(len,i) - 2*SMconc(len,i) +
SMconc(len+1,i) )...
                            +1*SMArea(len)*(SMconc(len,i)-
SMconc_left(len,i))*(DiffcoeffSM-D0/(1-gamma*SMconc_left(len,i)*1e-
3/rhoDensity)^2))/(SMArea(len)*dx_SM*dx_SM);
                        SMconc(len,i+1) = SMconc(len,i) +
TimeStep*(SMdiffterm(len,i+1)...
                            -
(1/(1*1*SMArea(len)))*(Kst_sm*(SMconc(len,i)-
STconc(len,i))+Ksm_sv*(SMconc(len,i)-
SVconc(len,i))+Ksm_clear*(SMconc(len,i))));
                    end


                    if len>=2
                        if len<Nsm
                            SMdiffterm(len,i+1) =
(DiffcoeffSM*(SMArea(len) - SMArea(len-1))*(SMconc(len,i) - SMconc(len-
1,i))...

+DiffcoeffSM*SMArea(len)*(SMconc(len+1,i) - 2*SMconc(len,i) +
SMconc(len-1,i) )...
                                +1*SMArea(len)*(SMconc(len,i)-
SMconc(len-1,i))*(DiffcoeffSM-D0/(1-gamma*SMconc(len-1,i)*1e-
3/rhoDensity)^2))/(SMArea(len)*dx_SM*dx_SM);
                            SMconc(len,i+1) = SMconc(len,i) +
TimeStep*(SMdiffterm(len,i+1)...
                                -
(1/(1*SMArea(len)))*(Kst_sm*(SMconc(len,i)-STconc(len,i)) +
Ksm_clear*(SMconc(len,i)-0) + Ksm_sv*(SMconc(len,i)-SVconc(len,i))));
                            if SMconc(len,i+1)<0
                                SMconc(len,i+1)=0;
                            end
                        elseif len==Nsm
                            SMconc(len,i+1) = SMconc(len-1,i+1);
                        end

                    end

                    %% concentrations in SM (left)
                    n_left = len;%0:0.01:1.18
                    if n_left ==1

                        SMconc_left(n_left,i+1) = SMconc_left(n_left,i)
+ TimeStep*(DiffcoeffSM*((SMconc(1,i) - 2*SMconc_left(n_left,i) +
SMconc_left(n_left+1,i))*SMArea_left(n_left)...
                            +(SMArea_left(n_left) -
SMArea(1))*(SMconc_left(n_left,i) -
SMconc(1,i)))/(SMArea_left(n_left)*dx_SM*dx_SM)...
                            +0 -
(1/(1*SMArea_left(n_left)))*(k_x(1)*BMleft(n_left)*(SMconc_left(n_left,
i)-STconc(1,i))...
                            +
k_x(2)*Reissmem_left(n_left)*(SMconc_left(n_left,i)-
SVconc_left(n_left,i))+
k_x(3)*SMSLmem2(n_left)*SMconc_left(n_left,i)));
                        if SMconc_left(n_left,i+1)<0
```

```
                        SMconc_left(n_left,i+1)=0;
                    end
                elseif n_left<SMleng_left
                    if n_left<length(BMleft)
                        SMconc_left(n_left,i+1) =
SMconc_left(n_left,i) + TimeStep*(DiffcoeffSM*((SMconc_left(n_left+1,i)
- 2*SMconc_left(n_left,i) + SMconc_left(n_left-
1,i))*SMArea_left(n_left)...
                                +(SMArea_left(n_left) -
SMArea_left(n_left-1))*(SMconc_left(n_left,i) - SMconc_left(n_left-
1,i)))/(SMArea_left(n_left)*dx_SM*dx_SM)...
                                +0-
(1/(1*SMArea_left(n_left)))*(k_x(1)*BMleft(n_left)*(SMconc_left(n_left,
i)-STconc_left(n_left,i))...
                                +
1*k_x(2)*Reissmem_left(n_left)*(SMconc_left(n_left,i)-
SVconc_left(n_left,i))+
k_x(3)*SMSLmem2(n_left)*SMconc_left(n_left,i)));
                    elseif n_left<size(SVconc_left,1)
                        SMconc_left(n_left,i+1) =
SMconc_left(n_left,i) + TimeStep*(DiffcoeffSM*((SMconc_left(n_left+1,i)
- 2*SMconc_left(n_left,i) + SMconc_left(n_left-
1,i))*SMArea_left(n_left)...
                                +(SMArea_left(n_left) -
SMArea_left(n_left-1))*(SMconc_left(n_left,i) - SMconc_left(n_left-
1,i)))/(SMArea_left(n_left)*dx_SM*dx_SM)...
                                +0-(1/(1*SMArea_left(n_left)))*(
1*k_x(2)*Reissmem_left(n_left)*(SMconc_left(n_left,i)-
SVconc_left(n_left,i))+1*k_x(3)*SMSLmem2(n_left)*SMconc_left(n_left,i))
);
                    else
                        SMconc_left(n_left,i+1) =
SMconc_left(n_left,i) + TimeStep*(DiffcoeffSM*((SMconc_left(n_left+1,i)
- 2*SMconc_left(n_left,i) + SMconc_left(n_left-
1,i))*SMArea_left(n_left)...
                                +(SMArea_left(n_left) -
SMArea_left(n_left-1))*(SMconc_left(n_left,i) - SMconc_left(n_left-
1,i)))/(SMArea_left(n_left)*dx_SM*dx_SM)...
                                -
(1/(1*SMArea_left(n_left)))*k_x(3)*SMSLmem2(n_left)*SMconc_left(n_left,
i));

                    end
                    if SMconc_left(n_left,i+1)<0
                        SMconc_left(n_left,i+1)=0;
                    end
                elseif n_left==SMleng_left
                    SMconc_left(n_left,i+1) = SMconc_left(n_left-
1,i+1);
                end

                %% Concentrations in SV (right)

                if len==1
                    SVdiffterm(len,i+1) = (DiffcoeffSV*(SVArea(len)
- SVArea_left(len))*(SVconc(len,i) - SVconc_left(len,i))...

+DiffcoeffSV*SVArea(len)*(SVconc_left(len,i) - 2*SVconc(len,i) +
```

```
SVconc(len+1,i) )...
                            +1*SVArea(len)*(SVconc(len,i)-
SVconc_left(len,i))*(DiffcoeffSV-D0/(1-gamma*SVconc_left(len,i)*1e-
3/rhoDensity)^2))/(SVArea(len)*dx_SV*dx_SV);
                        SVconc(len,i+1) = SVconc(len,i) +
TimeStep*(SVdiffterm(len,i+1)...
                            -
(1/(1*1*SVArea(len)))*(Ksm_sv*(SVconc(len,i)-SMconc(len,i))));
                    end
                    if len>=2
                        if len<Nsv
                            SVdiffterm(len,i+1) =
(DiffcoeffSV*(SVArea(len) - SVArea(len-1))*(SVconc(len,i) - SVconc(len-
1,i))...

+DiffcoeffSV*SVArea(len)*(SVconc(len+1,i) - 2*SVconc(len,i) +
SVconc(len-1,i) )...
                                +1*SVArea(len)*(SVconc(len,i)-
SVconc(len-1,i))*(DiffcoeffSV-D0/(1-gamma*SVconc(len-1,i)*1e-
3/rhoDensity)^2))/(SVArea(len)*dx_SV*dx_SV);
                            SVconc(len,i+1) = SVconc(len,i) +
TimeStep*(SVdiffterm(len,i+1)...
                                -
(1/(1*SVArea(len)))*(Ksm_sv*(SVconc(len,i)-SMconc(len,i))));
                            if SVconc(len,i+1)<0
                                SVconc(len,i+1)=0;
                            end
                        elseif len==Nsv
                            SVconc(len,i+1)=SVconc(len-1,i);
                        end
                    end

                    %% concentrations in SV (left)
                    n_left = len;
                    if n_left ==1
                        SVconc_left(n_left,i+1) = SVconc_left(n_left,i)
+ TimeStep*(D0/(1-gamma*SMconc_left(n_left,i)*1e-3/rhoDensity)^2*...
                            ((SVconc(1,i+1) - 2*SVconc_left(n_left,i) +
SVconc_left(n_left+1,i))*SVArea_left(n_left)...
                            +(SVArea_left(n_left) -
SVArea(1))*(SVconc_left(n_left,i) -
SVconc(1,i)))/(SVArea_left(n_left)*dx_SV*dx_SV)...
                            +0-
(1/(1*SVArea_left(n_left)))*(k_x(2)*Reissmem_left(n_left)*(SVconc_left(
n_left,i)-SMconc_left(n_left,i))));%/(0.001);
                        if SVconc_left(n_left,i+1)<0
                            SVconc_left(n_left,i+1)=0;
                        end
                    elseif n_left<SVleng_left
                        if n_left<length(Reissmem_left)%c2len
                            SVconc_left(n_left,i+1) =
SVconc_left(n_left,i) + TimeStep*(D0/(1-gamma*SMconc_left(n_left,i)*1e-
3/rhoDensity)^2*...
                                ((SVconc_left(n_left+1,i) -
2*SVconc_left(n_left,i) + SVconc_left(n_left-
1,i))*SVArea_left(n_left)...
                                +(SVArea_left(n_left) -
SVArea_left(n_left-1))*(SVconc_left(n_left,i) - SVconc_left(n_left-
```

```
1,i)))/(SVArea_left(n_left)*dx_SV*dx_SV)...
                                    -
(1/(1*SVArea(n_left)))*(1*k_x(2)*Reissmem_left(n_left)*(SVconc_left(n_l
eft,i)-SMconc_left(n_left,i))));%/(0.001);
                        else
                            SVconc_left(n_left,i+1) =
SVconc_left(n_left,i) + TimeStep*(DiffcoeffSV*((SVconc_left(n_left+1,i)
- 2*SVconc_left(n_left,i) + SVconc_left(n_left-
1,i))*SVArea_left(n_left)...
                                    +(SVArea_left(n_left) -
SVArea_left(n_left-1))*(SVconc_left(n_left,i) - SVconc_left(n_left-
1,i)))/(SVArea_left(n_left)*dx_SV*dx_SV));%/(0.001);
                        end
                        if SVconc_left(n_left,i+1)<0
                            SVconc_left(n_left,i+1)=0;
                        end
                    elseif n_left==SVleng_left
                        SVconc_left(n_left,i+1) = SVconc_left(n_left-
1,i+1);
                    end

                end
            end

            %%
            if strcmp(dateexp,'20180327')
                maxST = 274.2282;
            elseif strcmp(dateexp,'20180409')
                maxST = 282.424;
            elseif strcmp(dateexp,'20180621')
                maxST = 258.827;
            elseif strcmp(dateexp,'20180820')
                maxST = 279.7971;
            else
                maxST =240;% = D(0);
            end
            %             x
            if ~strcmp(dateexp,'MinMax')
                truncdisp = 99;
                amush = interp1([-(STleng_left-
2):100],amushy2(1:(100+STleng_left-1),:),linspace(-(STleng_left-
2),100,100));%interp1([1:142],bmushy2,linspace(1,142,100));
                figure, title('ST'),subplot(1,2,2),plot(amush(:,end-
4:end)/(2*maxST)),
                xlabel('% along length'),ylabel('Conc ratio wrt
max'),axis([0,truncdisp,0,1]);%300]);
                title('ST simulated'); grid on;
                legend('10 min','40 min','70 min','100 min','130 min');
                %figure,subplot(1,2,2), plot(amushy(1:100,end-
4:end)),xlabel('% along length'),ylabel('Conc
mg/ml'),axis([0,truncdisp,0,300]);  title('ST simulated'); legend('40
min','70 min','100 min','130 min');
                subplot(1,2,1), plot(STwholeOrigConcProfile(:,end-
4:end)/(2*maxST)),
                xlabel('% along length'),ylabel('Conc ratio wrt
max'),axis([0,truncdisp,0,1]);%300]);
                title('ST extracted'); grid on;
                legend('10 min','40 min','70 min','100 min','130 min');
```

```
                cmush = interp1([-(SMleng_left-
2):100],cmushy2,linspace(-(SMleng_left-
2),100,100));%interp1([1:142],bmushy2,linspace(1,142,100));
                figure, title('SM'),subplot(1,2,2),plot(cmush(:,end-
4:end)/(2*maxST)),
                xlabel('% along length'),ylabel('Conc ratio wrt
max'),axis([0,truncdisp,0,1]);%300]);
                title('SM simulated'); grid on;
                legend('10 min', '40 min','70 min','100 min','130
min');
                subplot(1,2,1), plot(SMWholeOrigConcProfile(:,end-
4:end)/(2*maxST)),
                xlabel('% along length'),ylabel('Conc ratio wrt max'),
axis([0,truncdisp,0,1]);%300]);
                title('SM extracted'); grid on;
                legend('10 min','40 min','70 min','100 min','130 min');

                bmush = interp1([-(SVleng_left-
2):100],bmushy2,linspace(-(SVleng_left-
2),100,100));%interp1([1:142],bmushy2,linspace(1,142,100));
                figure, title('SV'),subplot(1,2,2),plot(bmush(:,end-
4:end)/(2*maxST)),
                xlabel('% along length'),ylabel('Conc ratio wrt
max'),axis([0,truncdisp,0,1]);%300]);
                title('SV simulated'); grid on;
                legend('10 min','40 min','70 min','100 min','130 min');
                %subplot(1,2,1), plot([-(b2len-
2):100],interp1([1:101],bORIG(:,end-3:end),linspace(1,100,100+b2len-
1))), xlabel('% along length'),ylabel('Conc mg/ml'),axis([-(b2len-
1),truncdisp,0,260]); title('SV extracted');legend('40 min','70
min','100 min','130 min');
                subplot(1,2,1), plot(SVWholeOrigConcProfile(:,end-
4:end)/(2*maxST)),
                xlabel('% along length'),ylabel('Conc ratio wrt max'),
axis([0,truncdisp,0,1]);%300]);
                title('SV extracted'); grid on;
                legend('10 min','40 min','70 min','100 min','130 min');

                figure,  plot((STwholeOrigConcProfile(1:100,:) -
amush(:,end-4:end))/(2*maxST));
                xlabel('% along length'),ylabel('Conc ratio wrt
max'),axis([0,truncdisp,-0.5,0.5]);
                title('ST Error plot'),grid on; legend('10 min','40
min','70 min','100 min','130 min');
                figure,  plot((SMWholeOrigConcProfile(1:100,:) -
cmush(:,end-4:end))/(2*maxST));
                xlabel('% along length'),ylabel('Conc ratio wrt
max'),axis([0,truncdisp,-0.5,0.5]);
                title('SM Error plot'),grid on; legend('10 min','40
min','70 min','100 min','130 min');
                figure,  plot((SVWholeOrigConcProfile(1:100,:) -
bmush(:,end-4:end))/(2*maxST));
                xlabel('% along length'),ylabel('Conc ratio wrt
max'),axis([0,truncdisp,-0.5,0.5]);
                title('SV Error plot'),grid on; legend('10 min','40
min','70 min','100 min','130 min');
```

```
                figure, plot(linspace(0,Nsec+1,Nsec+1)/600,32 -
velocity_left*32/(2*2.806e-2));
                axis([0,150,0,35]),ylabel('Flow rate
nL/min'),xlabel('Time in min');
                title('Leakage rate');
                figure,
plot(linspace(0,Nsec+1,Nsec+1)/600,velocity_left*32/(2*2.806e-2));
                axis([0,150,0,35]),ylabel('Flow rate
nL/min'),xlabel('Time in min');
                title('Flow rate');
            else

                truncdisp = 99;
                amush = interp1(-(STleng_left-1):100,[STconc_left(end:-
1:1,(t_on+t_off-1)*60/0.1);STconc(1:100,(t_on+t_off-
1)*60/0.1)],linspace(-(STleng_left-
1),100,100));%interp1([1:142],bmushy2,linspace(1,142,100));

                min_all(t_off,t_on,iter) = min(amush/(2*maxST));

            end
        end
        max_apex(t_off,t_on) =
apex_conc(t_off,t_on,t_on*60/0.1)/(2*maxST);
        min_apex(t_off,t_on) = min(amush/(2*maxST));
        max_base(t_off,t_on) = max(amush/(2*maxST));
        min_base(t_off,t_on) = base_conc(t_off,t_on,Nsec)/(2*maxST);
    end
end
```