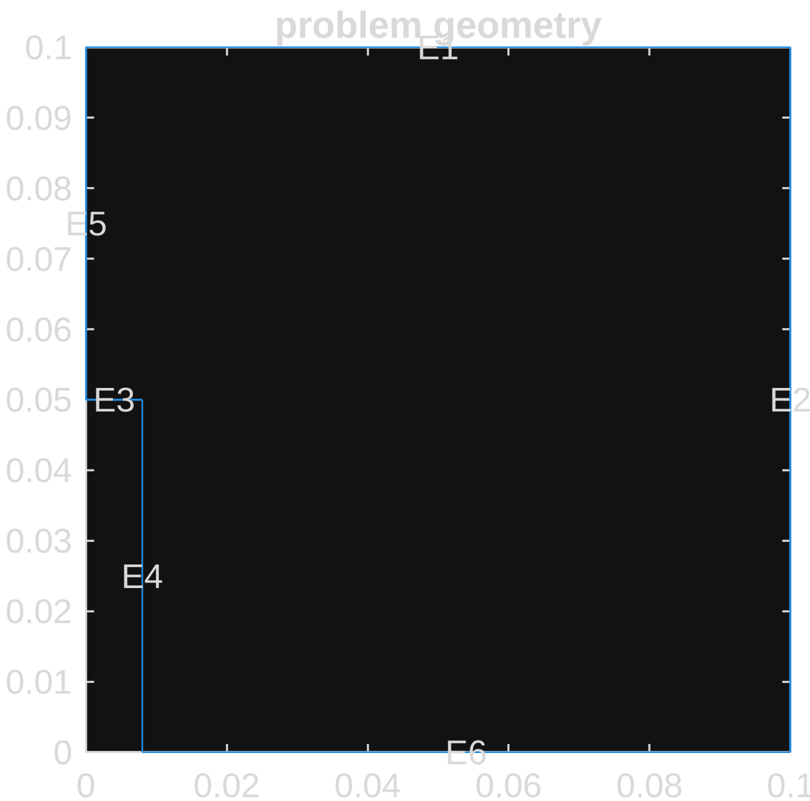
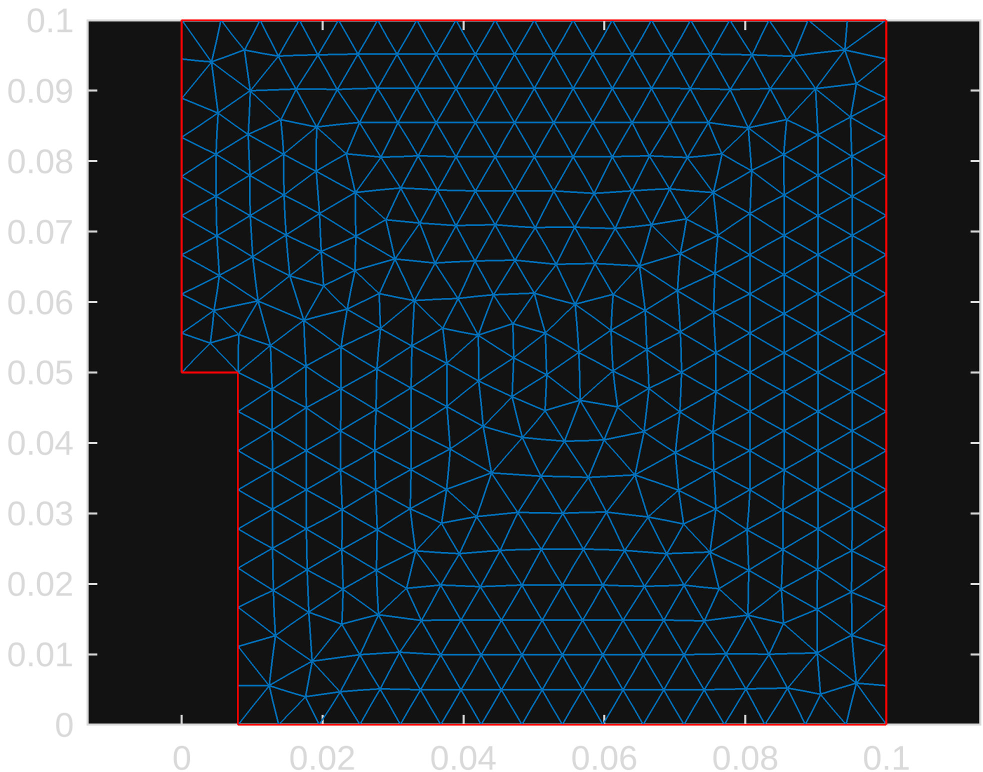
```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%DEFINE GEOMETRY
%Syntax and variable names derived from MATLAB documentation article
%entitled, '2-D Geometry Creation at Command Line'

%KEY MODEL PARAMETERS - D is defined in this case as k/(cp * rho)
k = 2.78*10^-6; %cm^2/s
rho = 1; %unitless
cp = 1; %unitless
q = 0;
tmodel=createpde('thermal','transient');

% First define geometry of domain (hydrogel region)
rect1 = [3 4 0 0 0.1 0.1 0 0.1 0.1 0]'; %coordinates of domain in cm

% Next define geometry of lumen
rect2 = [3 4 0 0 0.008 0.008 0 0.05 0.05 0]'; %coordinates of lumen in cm

% Concatenate the geometry matrices to create the basic shape matrix
gd = [rect1, rect2];

% The following explanation is from MATLAB documentation: "In order to
% create a formula describing the unions and intersections of basic shapes,
% you need a name for each basic shape. Give the names as a matrix whose
% columns contain the names of the corresponding columns in the basic
% shape matrix. One easy way to create the names is by specifying a
% character array whose rows contain the names, and then taking the
% transpose. Use the char function to create the array.

ns = char('rect1', 'rect2');
ns=ns';

% Use boolean operation to create the domain that is the set difference %
between the hydrogel and the lumen.
sf = 'rect1-rect2';


% Create the geometry as in PS06 & PS07 using the decsg command
[dl,bt]=decsg(gd,sf,ns);


% Set the geometry for the PDE model
geometryFromEdges(tmodel, dl);

% define "thermal properties" for the geometry
thermalProperties(tmodel,
'ThermalConductivity',k,'MassDensity',rho,'SpecificHeat',cp);
% Display resulting geometry
figure
pdegplot(tmodel, 'EdgeLabels','on')
```

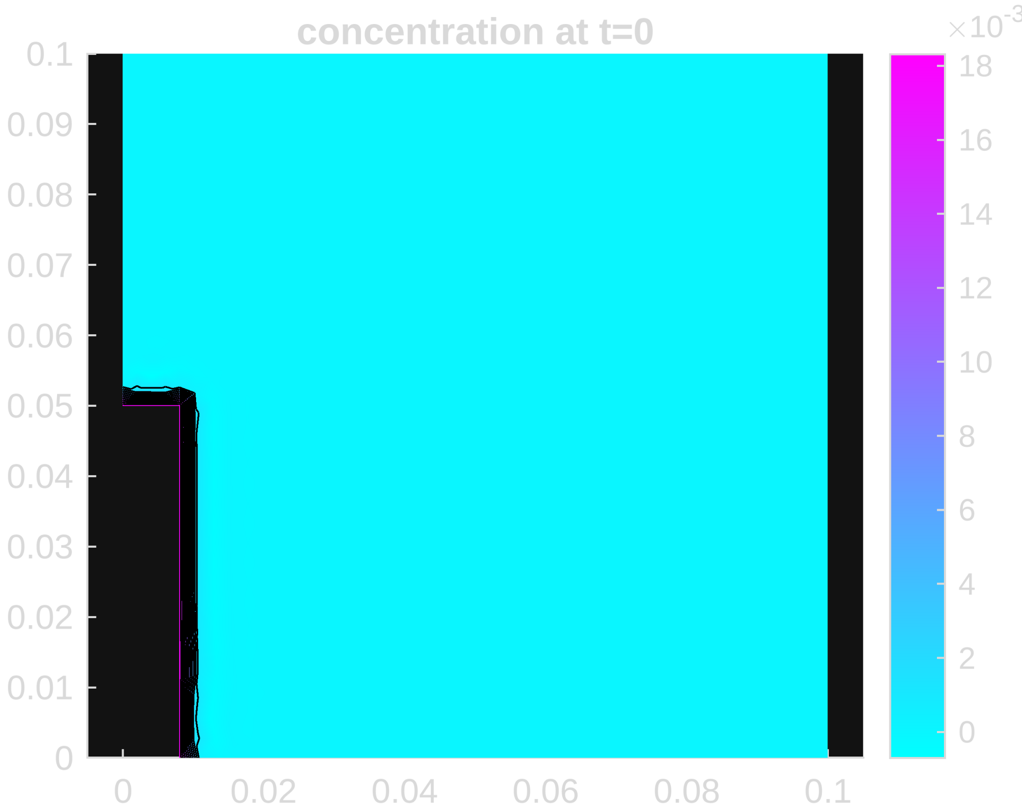
```
title('problem geometry')
axis equal
```



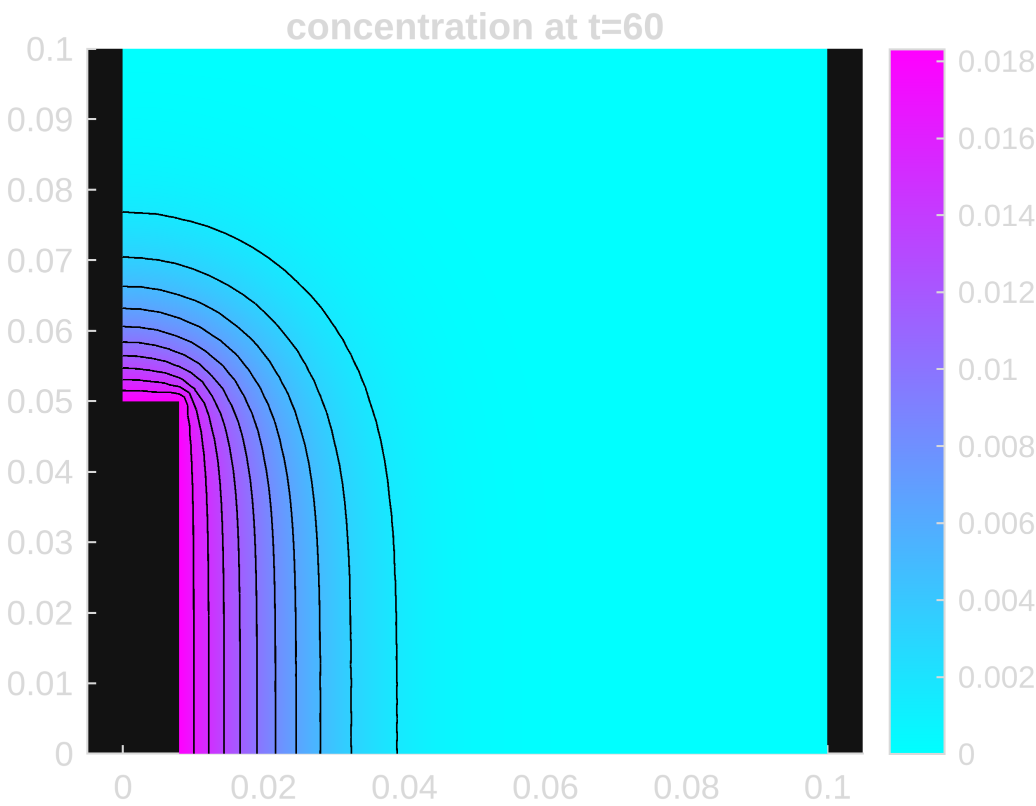

```
% Define boundary conditions
thermalBC(tmodel, 'Edge',1,'Temperature',0);%vessel surface edge 1, mg/mL
thermalBC(tmodel, 'Edge',3,'Temperature',1 * exp(-4)); %duct surface edge 3,
mg/mL
thermalBC(tmodel, 'Edge',4,'Temperature',1 * exp(-4)); %duct surface edge 4,
mg/mL
thermalBC(tmodel, 'Edge',2,'Temperature',0); %No flux at lateral boundary
thermalBC(tmodel, 'Edge',6,'HeatFlux',0); %No flux at PDMS edge 6
thermalBC(tmodel, 'Edge',2,'HeatFlux',0); %No flux at PDMS edge 2
thermalBC(tmodel, 'Edge',5,'HeatFlux',0); %No flux at symmetry plane

% Creation of geometry with Meshes
msh = generateMesh(tmodel);
mshpts=msh.Nodes;
figure
pdeplot(tmodel);
axis equal
```



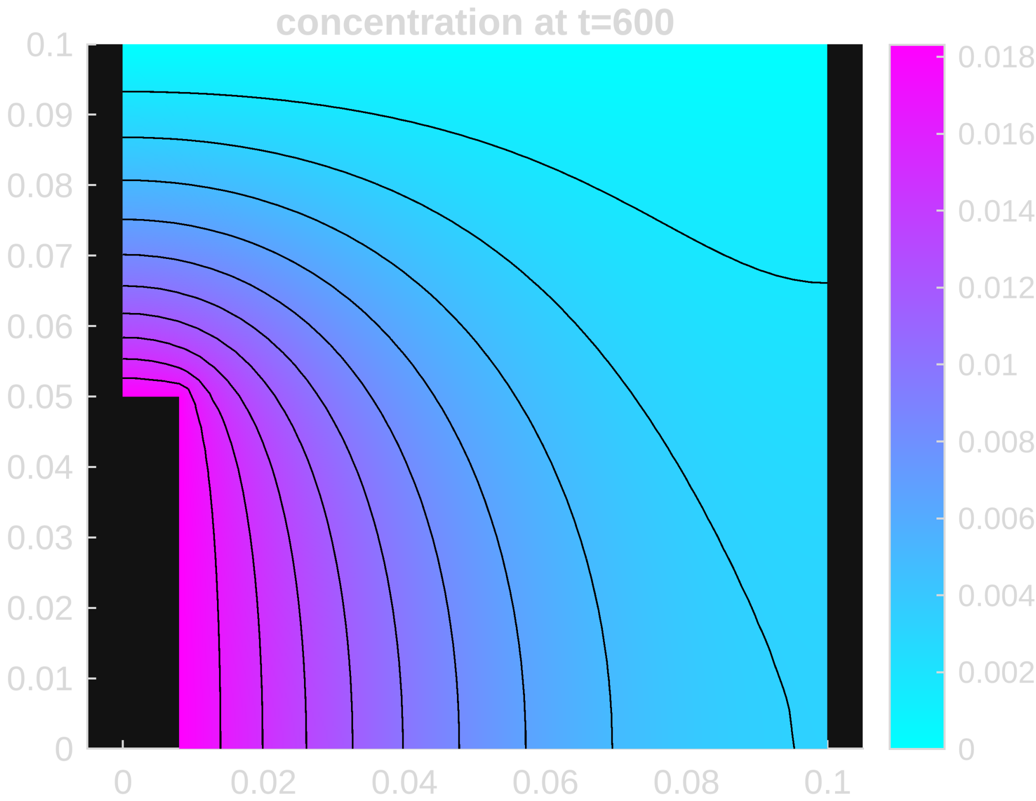

```
% Transient Solution modeling at 0 seconds'
initialtemp = 0; % temperature here will be considered as concentration
thermalIC(tmodel, initialtemp);
tfinal0 = 0.1;
tlist0 = linspace(initialtemp, tfinal0, 10000);
result0 = solve(tmodel,tlist0);

T0 = result0.Temperature;

% Plotting the data
figure
pdeplot(tmodel, 'XYData', T0(:, end), 'Contour','on');
axis equal
title 'concentration at t=0'
```



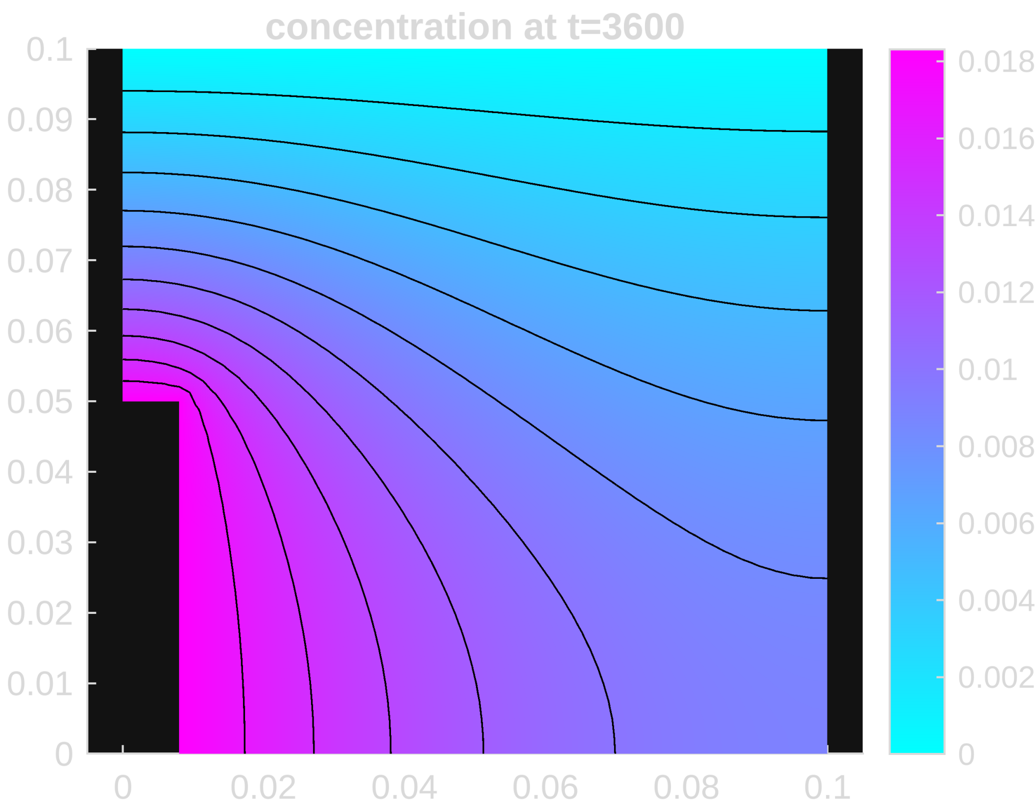

```
% Transient solutino modeling at 60 seconds
tfinal60 = 60;
tlist60 = linspace(initialtemp, tfinal60, 10000);
result60 = solve(tmodel, tlist60);
T60 = result60.Temperature;
% Plotting the data
figure
pdeplot(tmodel, 'XYData', T60(:, end), 'Contour','on');
axis equal
title 'concentration at t=60'
```




```
% Transient solutino modeling at 600 seconds
tfinal600 = 600;
tlist600 = linspace(initialtemp, tfinal600, 10000);
result600 = solve(tmodel, tlist600);
T600 = result600.Temperature;
% Plotting the data
figure
pdeplot(tmodel, 'XYData', T600(:, end), 'Contour','on');
axis equal
title 'concentration at t=600'
```



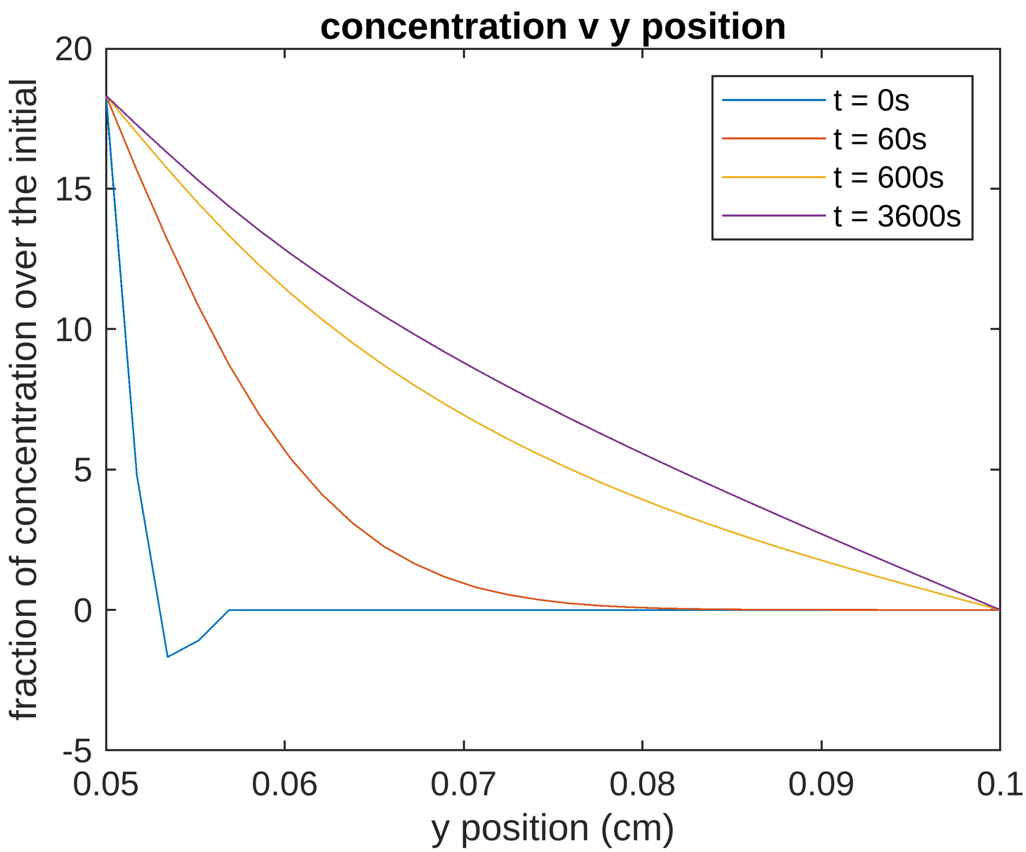

```
% Transient solutino modeling at 3600 seconds
tfinal3600 = 3600;
tlist3600 = linspace(initialtemp, tfinal3600, 10000);
result3600 = solve(tmodel, tlist3600);
T3600 = result3600.Temperature;
% Plotting the data
figure
pdeplot(tmodel, 'XYData', T3600(:, end), 'Contour','on');
axis equal
title 'concentration at t=3600'
```



Plotting Concentration lines as a function of y position

```
ylimits = [0.05, 0.1]; % distance between duct and vessel surface in mm
xlimits = [0, 1]; % length of diagram in mm


y = linspace(ylimits(1), ylimits(2), 30);
x = linspace(xlimits(1), xlimits(1), 30);

yreshaped = reshape(y, 30,1); % shape array to match indices

%temperature data arrayed
T = interpolateTemperature(result3600,x,y,1:length(tlist3600));

%plotting concentration from duct to vessel surface
figure
plot(yreshaped,(T(:,1)/100));
hold on
plot(yreshaped,(T(:,61)/100));
hold on
plot(yreshaped, (T(:,601)/100));
hold on
```

```
plot(yreshaped,(T(:,3600)/100));
hold on
legend('t = 0s', 't = 60s', 't = 600s', 't = 3600s')
xlabel('y position (cm)');
ylabel('fraction of concentration over the initial');
title('concentration v y position')
```



Differences in comparison to the paper

```
% generated graphs somewhat resemble the paper. however, t=0 curve dips
% below 0 whereas the in the paper it just asymptotically reaches 0. this
%is likely from how the t=0 data was originally plotted. mapping the
%solution at t=0 wasn't actually possible since matlab required a
%difference between your limits. you ended up having to plot solutions at
%times that approximately the times listed in the paper.
```