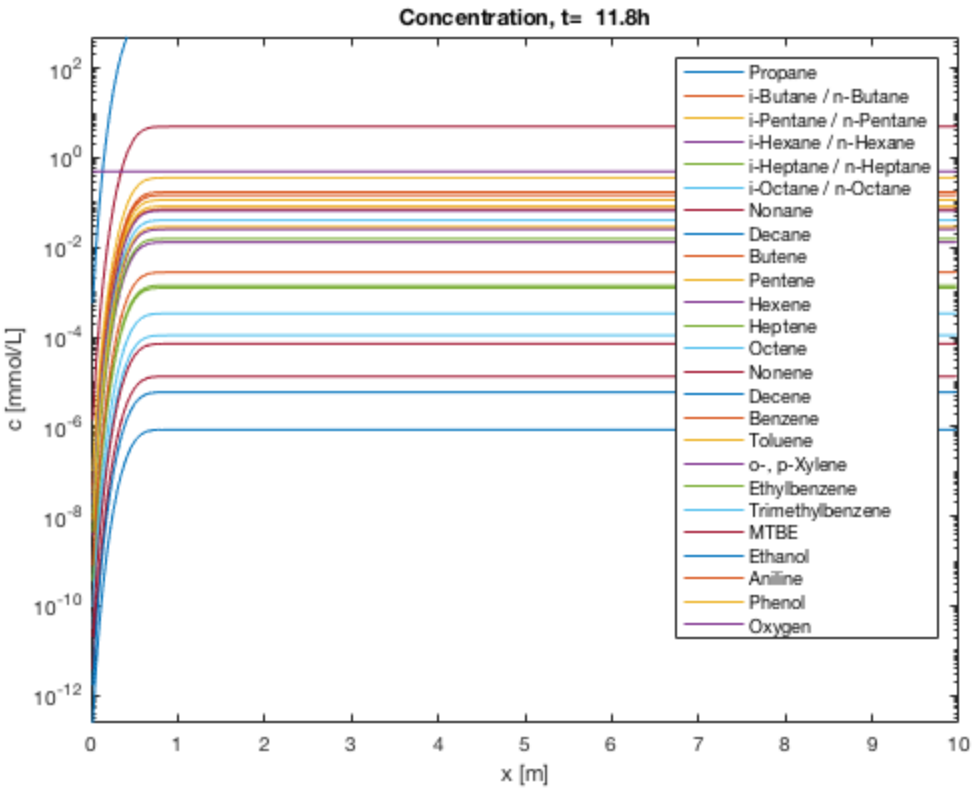
```
%
 ========================================================================
% 1-D Simulation of Transport and Fate of BTEX
% Hannah McNeil and Nicole Smith
% University of Tübingen
%
% December, 2018
%
 ========================================================================

clear all

% Transport Coefficients
m_tot = 8000;       % total mass of NAPL [kg]
c_in = 0.5;       % Dissolved oxygen inflow concentration [mol/m3]

L = 10;             % length of aquifer [m]
W = 20;             % width of aquifer [m]
H = 2;              % depth [m]
poros = 0.35;       % porosity [-]
q = 1.1574e-05;     % Specific Discharge [m/s]
alpha = 0.01;       % Longitudinal Dispersivity [m]
Dp = 1e-9;          % pore diffusion coefficient [m2/s] -- assumed

tin = 0;            % begin time [s]
% te = 3600*(1*240);  % end time [s]
te = 3600*(1*12)

% Derived Coefficients
A = H*W;            % area [m2]
v = q/poros;        % seepage velocity [m/s]
D = alpha*v+Dp;     % dispersion coefficient [m2/s]

% Load individual gasoline component data
% Compounds         % compound names for legend
% MW                % molecular weight [kg/mol]
% rho               % density [kg/m3]
% Si                % solubility constants at 20-25 C [kg/m3]
% wt                % weight percent in initial mixture [fraction]
load('data.mat')

% Invidual gasoline compounds - derived initial conditions
n = wt*m_tot./MW;   % moles of each compound in initial mixture [mol]
Xo = n./sum(n);     % molar fraction [-]
c_aq_i = Xo.*Si./MW;% initial water conc. at equilibrium  [mol/m3]

% calculate cnapl eq. 7.9

% Requirement 1: Neumann-Number = 1/4
% Requirement 2: Courant-Number Cr = dt*v/dx = 1
% Solve for dx:
dx = 4*D/v;
```

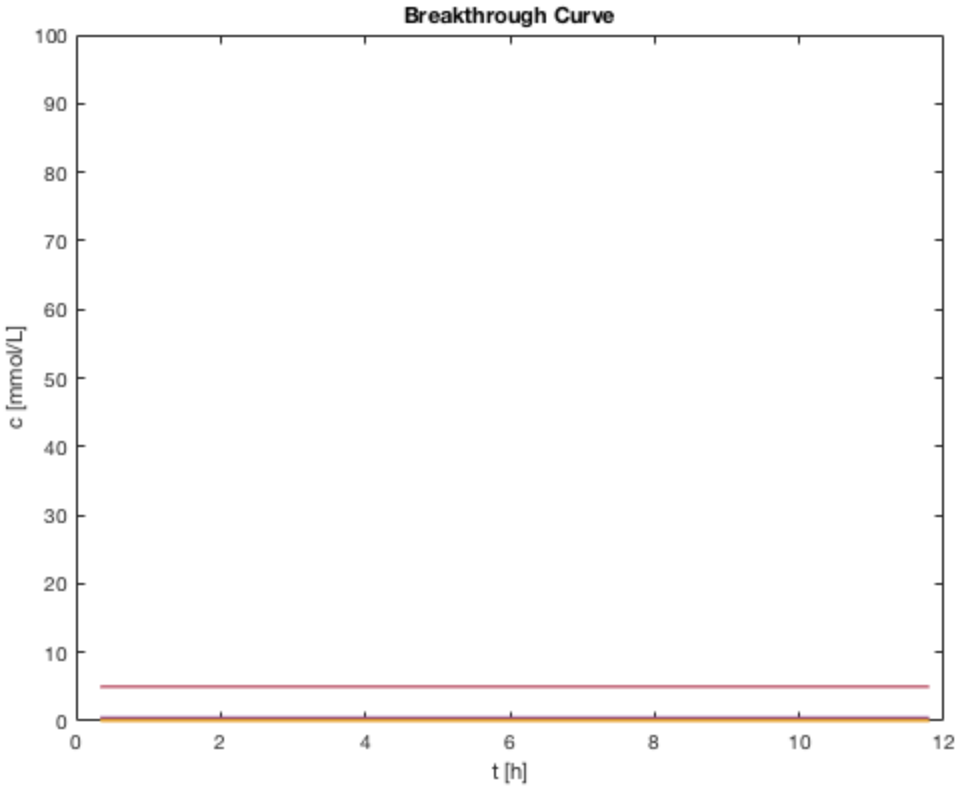
```
% Solve for dt
dt = dx/v;

x=0.5*dx:dx:L;
nx = length(x);

% Number of Components [given compounds + oxygen]
ncomp = 25;

% Matrix of Aqueous Concentrations - Compound and Oxygen
% Rows related to length coordinates
% Columns related to components
%c_aq = zeros(nx,ncomp);
c_aq = ones(nx,ncomp).*[c_aq_i,c_in(:,end)];

% Matrix of Initial Total NAPL Mass - Compound and Oxygen
% Rows related to length coordinates
% Columns related to components
m = ones(nx,1)*[(m_tot.*dx/L.*wt), 0];

% Matrix of Inflow Concentrations - Compound and Oxygen
% Columns related to components
c_in = [zeros(1,ncomp-1), c_in];

% Initial estimate of area for water flow
%Aw = ones(nx,ncomp).*poros.*H.*W;

% initialize breakthrough curve
BTC=zeros(0,ncomp);

% Open figure and delete its content
figure(1);clf
%
% % Open video
% v = VideoWriter('transport_model.avi');
% open(v);

%
 ========================================================================
% Loop over all timepoints
%
 ========================================================================
for t=dt:dt:te

    BTC=[BTC;c_aq(end,:)];

    %
 ========================================================================
    % ADVECTION
    %
 ========================================================================
    % Advection a Courant-number 1 implies that the concentrations are
    % moved by exactly one box. The values in the last box are moved
 out.
```

```
    % The first box receives the inflow concentration.

    c_aq(2:end,:)= c_aq(2:end,:) + dt*(q/dx)*(c_aq(1:end-1,:) -
 c_aq(2:end,:));
    c_aq(1,:)= c_aq(1,:) + dt*(q/dx)*(c_in - c_aq(1,:));

    %
 ========================================================================
    % DISPERSION
    %
 ========================================================================

%     Jd=(c_aq(1:end-1,:)-c_aq(2:end,:))/dx*D;
%     % add a dispersive flux of zero at the inflow boundary and
 assume that
%     % the dispersive flux at the outflow is identical to that at the
 last
%     % internal interface
%     Jd =[zeros(1,ncomp);Jd;Jd(end,:)];
%
%     % concentration change due to divergence of dispersive flux
%     c_aq = c_aq + dt/dx*(Jd(1:end-1,:)-Jd(2:end,:));

    %
 ========================================================================
    % REACTION
    %
 ========================================================================
   %
 ========================================================================
    % EQUILIBRATION
    %
 ========================================================================

    %
 ========================================================================
    % GRAPHICAL OUTPUT
    %
 ========================================================================

    % Graphical output every 10 minutes
    figure(1)
    semilogy(x,c_aq);
    xlabel('x [m]');
    ylabel('c [mmol/L]');
    ylim([0 500]);
    legend(compound)
    title(sprintf('Concentration, t=%6.1fh',t/3600));
    drawnow

%     % Graphical video
%         frame = getframe(gcf);
%         writeVideo(v,frame)
```

```
end

% % End video
% close(v)

figure(3)
%subplot(2,1,2)
plot([dt:dt:te]/3600,BTC)
xlabel('t [h]')
ylabel('c [mmol/L]')
ylim([0 100]);
title(sprintf('Breakthrough Curve'))% n = %3.1f')),n))
saveas(gcf, 'Breakthrough Curve.jpeg')
```

```
te =

       43200
```





Published with MATLAB® R2017b