Annex

Annex 1: Code in Matlab of the cation exchange numerical solution with standard finite differences

```
clear;
clc;
close all;
% parameters:
V=0.018; % Velocity [dm/min] --> 1.8mm/min
D= 0.0001; % Dispersion Coefficient
L=1; %lenght of domain [dm]
Tf=900; %[min]-->15h
K12=0.2; %reaction rate coefficient [-]
CEC=10^(-5);% cation exchange capacity [mol/g]
teta=0.2; % porosity [-]
ro=840;  % bulk density [g/dm3]
% gamma=CEC*ro/teta;%[mol/l]
gamma=0.0003;

TN = 101; %total nodes
TT=3010; %total time
delx=L/(TN-1);
delt=Tf/(TT-1);
if delx>=2*D/V
    error ('delx')
end
if delt>=(delx^2)/(2*D+K12*delx^2)
    error ('delt')
end

VT= 0:delt:Tf;%time vector
VN= 0:delx:L;%lenght vector

%initial conditions
C2=2.512*10^(-6); %[mol/l]
B1=0.99;
B2=0.01;
C1=B1*C2/(K12*B2); %=0.001243;

%from the equation: [A]*{c}k+1={b}k --> {c}k+1=[A]-1*{b}
%matrix A, b, c in zeros:
A = zeros(TN*4,TN*4);
b = zeros(TN*4,TT);
c = zeros(TN*4,TT);


%finite differences are applied from node=1 to node=TN and the
following
%terms are obtained by the implicit method
R =(-V*delt/(2*delx))+(-D*delt/(delx^2)); %a(j,j-1)
S =1+(2*D*delt/(delx^2)); %a(j,j)
T =(V*delt/(2*delx))-(D*delt/(delx^2)); %a(j,j+1)


%BOUNDARY CONDITIONS
    %left boundary conditions (first line of matrix)
C2o=2.512*10^(-6); %H+ [mol/l] given pH
C1o=0.01711; %Na+[mol/l]given
B2o=1/(1+K12*C1o/C2o);
```

```
B1o=1-B2o;
    %right boudary conditions(last line of matrix)
Ro=-(0.5*V*delt/delx+D*delt/(delx^2));
So=0.5*V*delt/delx+D*delt/(delx^2)+1;

%definition of A:
%equation 1
for i=1:TN
    if i==1 %left b.c.
        A(i,i)=1;
    elseif i==TN %right b.c.
        A(i,i-1)=Ro;
        A(i,i)=So;
    else
        A(i,i-1)=R;
        A(i,i)=S;
        A(i,i+1)=T;
        A(i,2*TN+i)=gamma;
    end
%equation 2
    if i==1 %left b.c.
        A(TN+i,TN+i)=1;
    elseif i==TN %right b.c.
        A(TN+i,TN+i-1)=Ro;
        A(TN+i,TN+i)=So;
    else
        A(TN+i,TN+i-1)=R;
        A(TN+i,TN+i)=S;
        A(TN+i,TN+i+1)=T;
        A(TN+i,3*TN+i)=gamma;
    end
%equation 3
    if i==1 %left b.c.
        A(2*TN+i,2*TN+i)=1;
    else
        A(2*TN+i,2*TN+i)=1;
        A(2*TN+i,3*TN+i)=1;
    end
%equation 4
    if i==1 %left b.c.
        A(3*TN+i,3*TN+i)=1;
    else
        A(3*TN+i,3*TN+i)=1;
    end
end


for t=1 %initial conditions
    for i=1:TN
        c(i,t)=C1;
        c(TN+i,t)=C2;
        c(2*TN+i,t)=B1;
        c(3*TN+i,t)=B2;
    end
end
%definition of b and final calculation of c:
for t=2:TT
    for i=1:TN
        if i==1 %left boundary conditions
            b(i,t)=C1o; %eq 1
            b(TN+i,t)=C2o; %eq 2
```

```
            b(2*TN+i,t)=B1o; %eq 3
            b(3*TN+i,t)=B2o; %eq 4
        elseif i==TN %right boundary conditions
            b(i,t)=c(i,t-1);%eq 1
            b(TN+i,t)=c(TN+i,t-1);%eq 2
            b(2*TN+i,t)=1; %eq 3
            b(3*TN+i,t)=(c(2*TN+i,t-1)*c(TN+i,t-1))/(K12*c(i,t-1));
%eq 4
        else
            b(i,t)=c(i,t-1)+gamma*c(2*TN+i,t-1);%eq 1
            b(TN+i,t)=c(TN+i,t-1)+gamma*c(3*TN+i,t-1);%eq2
            b(2*TN+i,t)=1;%eq3
            b(3*TN+i,t)=(c(2*TN+i,t-1)*c(TN+i,t-1))/(K12*c(i,t-
1));%eq4
        end
    end
    %final calculation:
    c(:,t)=(A\b(:,t));

    %graphic concentration vs lenght in one time:
%     subplot(2,2,1); plot(VN,c(1:TN,t)')
%     title ('C1=Na+')
%     xlabel('length [dm]')
%     ylabel('Concentration [moles/L]')
%     subplot(2,2,2); plot(VN,c(TN+1:2*TN,t)')
%     title ('C2=H+')
%     xlabel('length [dm]')
%     ylabel('Concentration [moles/L]')
%     subplot(2,2,3); plot(VN,c(2*TN+1:3*TN,t)')
%     title ('B1')
%     xlabel('length [dm]')
%     ylabel('Concentration [moles/L]')
%     subplot(2,2,4); plot(VN,c(3*TN+1:4*TN,t)')
%     title ('B2')
%     xlabel('length [dm]')
%     ylabel('Concentration [moles/L]')

end



subplot (2,2,1); plot(VT,c(TN,:)); %C1
title ('progresion of C1=[Na+]')
xlabel('time(minutes)')
ylabel('Concentration [moles/L]')
hold on
subplot(2,2,2);
plot(VT,c(TN+TN,:)); %C2[H+]
title ('progresion of C2=[H+]')
xlabel('time (minutes)')
ylabel('Concentration [moles/L]')
hold on
subplot (2,2,3); plot (VT,c(2*TN+TN,:));%B1
title ('progresion of [B1]')
xlabel('time(minutes)')
ylabel('Concentration [moles/L]')
hold on
subplot (2,2,4); plot (VT,c(3*TN+TN,:)); %B2
title ('progresion of [B2]')
xlabel('time(minutes)')
```

```
ylabel('Concentration [moles/L]')
%pH:
% plot(VT,-log10(c(TN+TN,:))); %pH
% title ('progresion of pH vs time')
% xlabel('time (min)')
% ylabel('pH')

%[H+] values saved:
filetxt = fopen('prova','w');
fprintf(filetxt,'Concentration \n');
fprintf(filetxt,'VT \t c(TN+TN) \n');
for i = 1:TT
    fprintf(filetxt,'%G \t %G \n',VT(i),c(TN+TN,i));
end
fclose(filetxt);
```

Annex 2: Code in Matlab of the cation exchange numerical solution with operator splitting

```
clear;
clc;
close all;
% parameters:
V=0.018; % Velocity [dm/min] --> 1.8mm/min
D= 0.0001; % Dispersion Coefficient
L=1; %lenght of domain [dm]
Tf=900; %[min]-->15h
K12=0.2; %reaction rate coefficient [-]
CEC=10^(-5);% cation exchange capacity [mol/g]
teta=0.2; % porosity [-]
ro=840;  % bulk density [g/dm3]
% gamma=CEC*ro/teta;%[mol/l]
gamma=0.0003;

TN = 101;%total nodes
TT=3010;%total time
delx=L/(TN-1);
delt=Tf/(TT-1);
if delx>=2*D/V
    error ('delx')
end
if delt>=(delx^2)/(2*D+K12*delx^2)
    error ('delt')
end

VT= 0:delt:Tf; %time vector
VN= 0:delx:L; %lenght vector

%initial conditions
C2=2.512*10^(-6); %[mol/l]
B1=0.99;
B2=0.01;
C1=B1*C2/(K12*B2); %=0.001243;

%from the equation: [A]*{c}k+1={b}k --> {c}k+1=[A]-1*{b}
%matrix A, b, c in zeros:
A = zeros(TN*2,TN*2);
b = zeros(TN*2,TT);
c = zeros(TN*2,TT);
beta = zeros(TN*2,TT);
c_sub = zeros(TN*2,TT-1);

%finite differences are applied from node=1 to node=TN and the
following
%terms are obtained by the implicit method
R =(-V*delt/(2*delx))+(-D*delt/(delx^2)); %a(j,j-1)
S =1+(2*D*delt/(delx^2)); %a(j,j)
T =(V*delt/(2*delx))-(D*delt/(delx^2)); %a(j,j+1)

%BOUNDARY CONDITIONS
    %left boundary conditions(first line of matrix)
C2o=2.512*10^(-6); %H+ [mol/l] given pH
C1o=0.01711; %Na+[mol/l]given
B2o=1/(1+K12*C1o/C2o);
```

```
B1o=1-B2o;
    %right boundary conditions(last line of matrix)
Ro=-(0.5*V*delt/delx+D*delt/(delx^2));
So=0.5*V*delt/delx+D*delt/(delx^2)+1;

%definition of A:
%equation 1
for i=1:TN
    if i==1 %left b.c.
        A(i,i)=1;
    elseif i==TN %right b.c.
        A(i,i-1)=Ro;
        A(i,i)=So;
    else
        A(i,i-1)=R;
        A(i,i)=S;
        A(i,i+1)=T;
    end
    %equation 2
    if i==1 %left b.c.
        A(TN+i,TN+i)=1;
    elseif i==TN %right b.c.
        A(TN+i,TN+i-1)=Ro;
        A(TN+i,TN+i)=So;
    else
        A(TN+i,TN+i-1)=R;
        A(TN+i,TN+i)=S;
        A(TN+i,TN+i+1)=T;
    end
end


for t=1 %initial conditions
    for i=1:TN
        c(i,t)=C1;
        c(TN+i,t)=C2;
        beta(i,t) = B1;
        beta(TN+i,t) = B2;
    end
end
%definition of b and final calculation of c:
for t=2:TT
    for i=1:TN
        if i==1 %left boundary conditions
            b(i,t)=C1o; %eq 1
            b(TN+i,t)=C2o; %eq 2
        elseif i==TN %right boundary conditions
            b(i,t)=c(i,t-1);%eq 1
            b(TN+i,t)=c(TN+i,t-1);%eq 2
        else
            b(i,t)=c(i,t-1);
            b(TN+i,t)=c(TN+i,t-1);
        end
    end
    %final calculation
    c_sub(:,t)=(A\b(:,t));
    for i=1:TN
        beta(i,t) = K12*c_sub(i,t)/(K12*c_sub(i,t)+c_sub(TN+i,t));
        beta(TN+i,t) = 1-beta(i,t);
        c(i,t) = c_sub(i,t) - gamma*(beta(i,t)-beta(i,t-1));
```

```
        c(TN+i,t) = c_sub(TN+i,t) - gamma*(beta(TN+i,t)-beta(TN+i,t-
1));
    end

    % graphic concentrartion vs lenght in one time:
%     subplot(2,2,1); plot(VN,c(1:TN,t)')
%     title ('C1=Na+')
%     xlabel('length [dm]')
%     ylabel('Concentration [moles/L]')
%     subplot(2,2,2); plot(VN,c(TN+1:2*TN,t)')
%     title ('C2=H+')
%     xlabel('length [dm]')
%     ylabel('Concentration [moles/L]')
%     subplot(2,2,3); plot(VN,beta(1:TN,t)')
%     title ('B1')
%     xlabel('length [dm]')
%     ylabel('Concentration [moles/L]')
%     subplot(2,2,4); plot(VN,beta(TN+1:2*TN,t)')
%     title ('B2')
%     xlabel('length [dm]')
%     ylabel('Concentration [moles/L]')
end
```

```
subplot (2,2,1); plot(VT,c(TN,:)); %C1
title ('progresion of C1=[Na+]')
xlabel('time(minutes)')
ylabel('Concentration [moles/L]')
hold on
subplot(2,2,2);
plot(VT,c(TN+TN,:)); %C2[H+]
title ('progresion of C2=[H+]')
xlabel('time(minutes)')
ylabel('Concentration [moles/L]')
hold on
subplot (2,2,3); plot (VT,beta(TN,:));%B1
title ('progresion of [B1]')
xlabel('time(minutes)')
ylabel('Concentration [moles/L]')
hold on
subplot (2,2,4); plot (VT,beta(2*TN,:)); %B2
title ('progresion of [B2]')
xlabel('time(minutes)')
ylabel('Concentration [moles/L]')

%pH:
% plot(VT,-log10(c(TN+TN,:))); %pH
% title ('progresion of pH vs time')
% xlabel('time(min)')
% ylabel('Concentration [moles/L]')
```