TO: Professor Andreas Linninger / Grant Hartung

FROM: Kehinde Abioye

DATE: 10/26/2017

SUBJECT: Nonlinear Systems

Using knowledge of eigenvectors, we can solve for nonlinear systems independently to study flux across the Blood Brain Barrier using the diffusion and convection equations for concentration.

Introduction

The brain needs nutrients and needs to get rid of waste on a continuous basis. The blood control both transports. Water is an important nutrient that is handled by the blood brain barrier (BBB). This assignment describes the transport of solute and water and how it moves across a membrane with no linear equations. Eigenvectors can be used to solve these equations to study solute transport across the BBB.

Methods

Part 1:

Equation 2 was shown to satisfy equation 1 by evaluating equation 1 at steady state.

$$V\frac{dc}{dt} = \frac{d}{dx}\frac{js}{A_\perp} = \frac{d}{dx}\frac{(js^c + js^d)}{A_\perp} \qquad (1)$$

$$V\frac{dc}{dt} = \nabla \cdot (\mathrm{vc} - \mathrm{D}\nabla \mathrm{c})$$

$$V\frac{dc}{dt} = \frac{d}{dx}(vc - D\frac{d}{dx})$$

$$\mathrm{c(x)} = \frac{\mathrm{c_1} - \mathrm{c_2}}{1 - e^{pe}} e^{\frac{\mathrm{jw}(1-\sigma)}{DA_\perp}x} + \frac{\mathrm{c_2} - \mathrm{c_1}e^{pe}}{1 - e^{pe}} \quad (2)$$

$$Pe = \frac{vL}{D} = \frac{(1-\sigma)jw^L}{DA_\perp}$$

Part 2:

The concentration profile, c (x), was plotted with range of x between 0 and L (the given parameters were used) using for loops. Then the concentration profile was evaluated when jw = -5 μm3 · s-1 and plotted.

Part 3:

The solute flux, shown in equation 3:

$$\frac{js(x=0)}{A_\perp} = (1-\sigma)vc|_{x=0} - \frac{dc}{dx}|_{x=0} = \frac{jw}{A_\perp}(1-\sigma)[\frac{\mathrm{c_2} - \mathrm{c_1}e^{pe}}{1 - e^{pe}}] \quad (3)$$

$$Pe{=}\gamma Jw$$

$$\gamma{=}3.33{\cdot}10{-}5\ \mu\mathrm{m} \cdot \mathrm{s}\text{-}1$$

was evaluated at x=0, x=0.5·L, and x=L.

Part 4:

The solution for js and jw were found and the residual error contour and residual error surface was plotted. For loops with the ranges jw = -50 to jw = 30 and js = -60 to js = 800 μmol/s were used. The residual error was found by using equations 4 and 5:

$$g1(js,jw)=0 \quad (4)$$

$$g2(js,jw)=0 \quad (5)$$

which represents the residual form of js and jw given in equation 6 and 7:

$$\frac{jw}{A_\perp}=\alpha(p_1-p_2)-\beta(c_1-c_i) \quad (6)$$

$$ci=js/jw$$

$$\alpha=1.67\cdot10-4 \text{ μmmmHg·s}$$

$$\beta=RT\sigma=0.206365 \text{ μm4μmol·s}$$

$$\frac{js}{A_\perp}=\frac{jw}{A_\perp}(1-\sigma)\left[\frac{c_2-c_1e^{pe}}{1-e^{pe}}\right] \quad (7)$$

$$Pe=\gamma Jw$$

$$\gamma=0.02 \text{ s} \cdot \text{μm-3}$$

The product of g1 and the product of g2 were taken and the values were added. The matrix was transposed to plot the maps.

Part 5:

Equations 4 and 5 were solved for with MATLAB's fsolve function. The solution was verified using the value from part 4 and marked on the plot.

Part 6:

Equation 4 and 5 were solved for using different values of the concentration (5, 10,15 and 20).

Results

Part 2:

Figure one shows the concentration profile for jw=5 and -5:

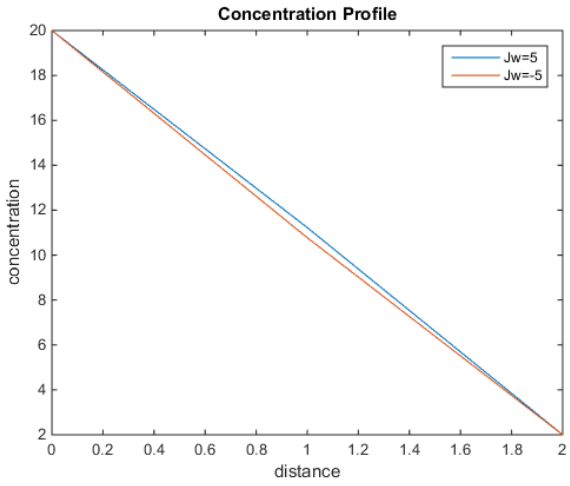

Figure 1 Concentration profile when jw is changed

Part 4:

Figure 2 and 3 show the residual error surface and residual error contour of js and jw:

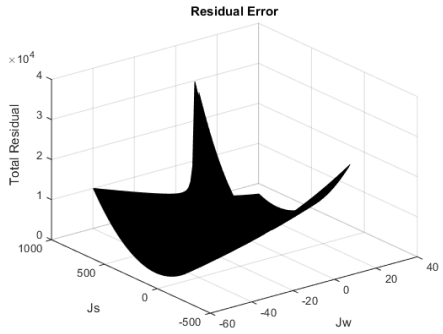

Figure 2 Surface plot of residual error

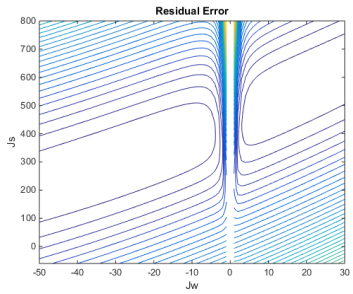

Figure 3 Contour plot of residual error

Conclusion

Convection and concentration equations were used to find the concentration profile in a membrane. A contour and surface map were produced to visualize the error between two membranes.

Discussion

Eigenvectors are useful when solving for non-linear equations which usually describe biological systems. This assignment was effective in learning how to use this knowledge to study solute across membranes.

Appendix A

```
clc
clear all

%Problem 2
p1 =30; %mmHg
p2 =4; %mmHg
c1 =20; %µmol*µm-3
c2 =2; %µmol*µm-3
D =10; %µm*s-1
sigma= 0.5;
T =298; %K
LP= 1.67e-4; %µm*mmHg-1*s-1
L =2; %µm
Ap= 5; %µm2
jw=5; %µm3*s-1
jw2=-5
Pe=((1-sigma)*jw*L)/(D*Ap);
Pe2=((1-sigma)*jw2*L)/(D*Ap);
czero=((c1-c2)/(1-exp(Pe)))*exp(0)+(c2-c1*exp(Pe))/(1-exp(Pe))
czero2=((c1-c2)/(1-exp(Pe2)))*exp(0)+(c2-c1*exp(Pe2))/(1-exp(Pe2))
%  can't put a zero in for loop. I get an error message that I can only put
%  positive integers or logicals
for i=1:L
    c(i)=((c1-c2)/(1-exp(Pe)))*exp((Pe*i)/L)+(c2-c1*exp(Pe))/(1-exp(Pe))
end

for i=1:L
    cAlt(i)=((c1-c2)/(1-exp(Pe2)))*exp((Pe2*i)/L)+(c2-c1*exp(Pe2))/(1-
exp(Pe2))
end

c= [czero c]
cAlt= [czero2 cAlt]
%same values whether jw is negative or positive

x=0:L;

plot(x,c)
hold on
plot(x,cAlt)
title('Concentration Profile');
xlabel('distance');
ylabel('concentration');
legend('Jw=5','Jw=-5')

%Question 3
% x=0;
% x=0.5*L;
x=L;
gamma= 0.0000333;
Pe3= gamma*jw;
u= ((1-sigma)*jw);
sFlux= (u*c1)-((u*((c1-c2)/(1-exp(Pe3))))*exp((u/D)*x))
```

```
%%Question 4
p1= 30;
p2= 4;
beta = .206365;
gamma1 = .02;

jw4 = (-50:30);
js4 = (-60:800);

for i = 1:length(jw4);
    for j = 1:length(js4);
        Pe3 = gamma1*jw4(i);
        ci = js4(j)/jw4(i);
        g1(i,j) = (LP*(p1-p2)-beta*(c1-ci)-(jw4(i)/Ap));
        g2(i,j) = ((jw4(i)/Ap)*(1-sigma)*(c2-c1*exp(Pe3))/(1-exp(Pe3))-
(js4(j)/Ap));
        res(i,j) = g1(i,j)*g1(i,j);
        res2(i,j) = g2(i,j)*g2(i,j);
        sum(i,j) = res(i,j)+res2(i,j);
    end
end

sum = sum';

figure;
surf(jw4,js4,sum)
title('Residual Error');
xlabel('Jw');
ylabel('Js');
zlabel('Total Residual');
figure;
contour(jw4,js4,sum,50)
title('Residual Error');
xlabel('Jw');
ylabel('Js');

%Problem 6
%c2=5
c2=5;
for i = 1:length(jw4);
    for j = 1:length(js4);
        Pe3 = gamma1*jw4(i);
        ci = js4(j)/jw4(i);
        g1(i,j) = (LP*(p1-p2)-beta*(c1-ci)-(jw4(i)/Ap));
        g2(i,j) = ((jw4(i)/Ap)*(1-sigma)*(c2-c1*exp(Pe3))/(1-exp(Pe3))-
(js4(j)/Ap));
        res(i,j) = g1(i,j)*g1(i,j);
        res2(i,j) = g2(i,j)*g2(i,j);
        sum(i,j) = res(i,j)+res2(i,j);
    end
end

sum = sum';
```

```
%c2=10
c2=10
for i = 1:length(jw4);
    for j = 1:length(js4);
        Pe3 = gamma1*jw4(i);
        ci = js4(j)/jw4(i);
        g1(i,j) = (LP*(p1-p2)-beta*(c1-ci)-(jw4(i)/Ap));
        g2(i,j) = ((jw4(i)/Ap)*(1-sigma)*(c2-c1*exp(Pe3))/(1-exp(Pe3))-
(js4(j)/Ap));
        res(i,j) = g1(i,j)*g1(i,j);
        res2(i,j) = g2(i,j)*g2(i,j);
        sum(i,j) = res(i,j)+res2(i,j);
    end
end

sum = sum';

%c2=15
c2=15;
for i = 1:length(jw4);
    for j = 1:length(js4);
        Pe3 = gamma1*jw4(i);
        ci = js4(j)/jw4(i);
        g1(i,j) = (LP*(p1-p2)-beta*(c1-ci)-(jw4(i)/Ap));
        g2(i,j) = ((jw4(i)/Ap)*(1-sigma)*(c2-c1*exp(Pe3))/(1-exp(Pe3))-
(js4(j)/Ap));
        res(i,j) = g1(i,j)*g1(i,j);
        res2(i,j) = g2(i,j)*g2(i,j);
        sum(i,j) = res(i,j)+res2(i,j);
    end
end

sum = sum';

%c2=20
c2=20;
for i = 1:length(jw4);
    for j = 1:length(js4);
        Pe3 = gamma1*jw4(i);
        ci = js4(j)/jw4(i);
        g1(i,j) = (LP*(p1-p2)-beta*(c1-ci)-(jw4(i)/Ap));
        g2(i,j) = ((jw4(i)/Ap)*(1-sigma)*(c2-c1*exp(Pe3))/(1-exp(Pe3))-
(js4(j)/Ap));
        res(i,j) = g1(i,j)*g1(i,j);
        res2(i,j) = g2(i,j)*g2(i,j);
        sum(i,j) = res(i,j)+res2(i,j);
    end
end

sum = sum';
```