OPEN ACCESS

International
Journal
Of Modern Engineering Research (IJMER)

Matlab Source Code for Species Transport through Nafion Membranes in Direct Ethanol, Direct Methanol, and Direct Glucose Fuel Cells

Michael W. Manley[1,] John H. Summerfield[2]

Department Of Chemical and Physical Sciences Missouri Southern State University Joplin

ABSTRACT: *A simple simulation of chemical species movement is presented. The species traverse a Nafion membrane in a fuel cell. Three cells are examined: direct methanol, direct ethanol, and direct glucose. The species are tracked using excess proton concentration, electric field strength, and voltage. The Matlab computer code is provided.*

I. INTRODUCTION

An efficient fuel cell could replace combustion engines and advance the goal of being less oil dependent. For a fuel cell, an accurate mathematical model is an important tool for optimizing fuel cell efficiency, which makes fuel cell model a current research topic.[1,2] The novelty of this work is that three fuel cells are modelled apart from the typical hydrogen oxygen fuel cell. Furthermore, a Matlab computer program is presented so that others may investigate their own systems.

The systems studied in this work involve chemical species transport with an ion exchange polymer composite matrix supported on an electrode surface. The ion exchange polymer that is used to modify electrodes in this work is Nafion, the structure of which is shown in Figure 1.

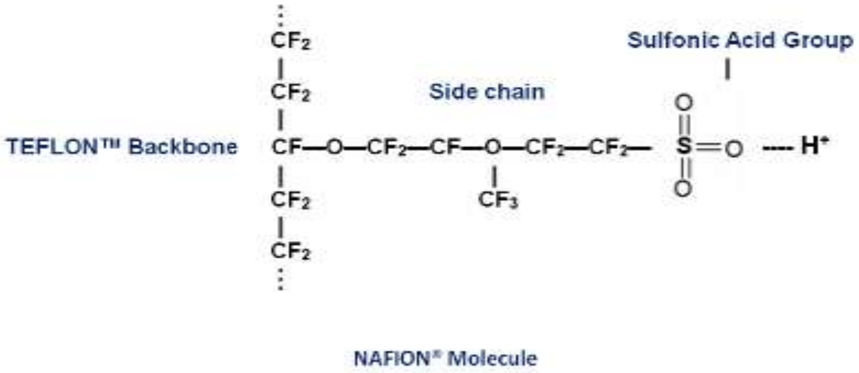

Figure 1: Molecular structure of Nafion.

Structurally Nafion is a Teflon-like, hydrophobic, fluorocarbon backbone with sidechains that terminate in a hydrophilic sulfonic acid. When Nafion is in contact with solution, the proton from the sulfonic acid can easily exchange with cationic species in solution. Nafion provides a cation selective matrix where cationic redox species are concentrated within the film. Because Nafion is a polymer, mass transport in the system is slowed. In the acid form, Nafion provides ion conduction through the acidic proton about the sulfonic acid. A fuel cell is similar to a battery in that it converts chemical energy into electrical energy and is better than a battery because it does not undergo charge/discharge cycles. A fuel cell provides power as long as it is provided fuel, similar to a combustion engine. A fuel cell is better than a combustion engine because it converts chemical energy directly into electrical energy without moving through a pressure-volume cycle and so is a more efficient process. The Carnot limitations restrict combustion engines to a theoretical maximum efficiency of 40%. Because a fuel cell converts chemical energy to electrical energy without mechanical cycles, there are no thermodynamic limitations and so the theoretical efficiency is 100%.

A fuel cell consists of two electrodes separated by an ion conducting membrane. As is typical, the membrane is Nafion and the electrodes are graphite. When catalyst coated electrodes are pressed against the membrane, interfacial zones are created. The electrochemical reactions occur only in these interfacial zones. A local difference in the concentration of anions and cations is produced because of these reactions. This separation of charge creates a potential difference across the cell. The slightly resistive nature of the electrodes and Nafion causes a potential loss in these regions. Table 1 shows the fuel cells investigated in this work.[3]

Fuel Cell Type	**Anode**	**Cathode**	**Voltage (V)**
Methanol	$CH_3OH + 6OH^- => CO_2 + 5H_2O + 6e^-$	$3/2O_{2\,+}\, 3H_2O + 6e^- => 6OH^-$	1.12
Ethanol	$CH_3CH_2OH + 2OH^- => CH_3COOH + 3H_2O + 4e^-$, $CH_3CH_2OH + 12OH^- => 2CO_2 + 9H_2O + 12e^-$	$3O_{2\,+}\, 6H_2O + 12e^- => 12OH^-$	1.17
Glucose	$C_6H_{12}O_6 + H_2O => C_6H_{12}O_7 + 2H^+ + 2e^-$	$O_2 + 2H^+ + 2e^- => H_2O$	1.30

Table1: The fuel cell type, the anode reaction, the cathode reaction, and the cell voltages for this work.

The Model

Consider a one dimensional model. Define $C_j(x, t)$ as the concentration in mol/cm^3,$J_j(x, t)$ as the flux (mol cm^{-2} s^{-1}) and φ(x, t) as the potential (volts). The Nernst-Plank equation relates these quantities.

$$J_j(x,t) = -D_j \frac{\partial C_j(x,t)}{\partial x} - \frac{z_j F D_j}{RT} C_j(x,t) \frac{\partial \varphi(x,t)}{\partial x} \quad (1)$$

Where $D_j$is the diffusion rate of species$j$ (cm^2 s^{-1}), z_j is the charge on species j, F is Faraday's constant (C mol^{-1}), R is the gas law constant (J K^{-1} mol^{-1}), and T is temperature (K).

More specifically, $C_j(x, t)$ is the concentration of j needed to neutralize the sulfonates concentration. Let N be the concentration of sulfonates in the Nafion. For simplification, the relation between flux and concentration is relied on,

$$\frac{\partial C_j(x,t)}{\partial t} = -\frac{\partial J_j(x,t)}{\partial x} .$$

(2)

Using Eq. (1), Eq. (2), and our newest definitions,

$$\frac{\partial C_j(x,t)}{\partial t} = D_j \frac{\partial^2 C_j(x,t)}{\partial x^2} + \frac{z_j F D_j}{RT} \left[\left(C_j(x,t) + N \right) \frac{\partial \varphi(x,t)}{\partial x} \right]_x \qquad (3)$$

The total current, i, at steady state is set by the steady state flux, J_j,

$$i = \sum_j i_j = -\sum_j nFAJ_i \qquad (4)$$

where$n$ is the number of electrons involved in the redox reaction and A is the surface area of the electrode. This work considers each chemical species independently so Eq. (4) reduces to

$$i = -nFAJ_j . \qquad (5)$$

Turning to the relation between concentration and electric potential energy, Poisson's equation is relied on

$$\frac{\partial^2 \varphi\left(x,t\right)}{\partial x^2} = -\frac{F}{\varepsilon} \sum_j z_j C_j\left(x,t\right)$$

(6)

where $\varepsilon = \varepsilon_0 \varepsilon_r$ and is the relative permittivity. ε_0 is the vacuum permittivity and ε_r is the dielectric constant. This is 20 for Nafion.[4] For the systems of interest, Eq.(6) becomes

$$\frac{\partial^2 \varphi\left(x,t\right)}{\partial x^2} = -\frac{F}{\varepsilon} C_j\left(x,t\right). \qquad (7)$$

Finally then, equations (3) and (7) are used to define the species movement over the Nafion membrane.[5]

Equation (3) provides the transport equation with a condition on $\varphi''\left(C\left(x\right)\right)$ as seen in Eq. (7).Instead of considering derivatives of the potential, it is replaced with a relation to the electric field, $E\left(C\left(x\right)\right)$,

$$E\left(C\left(x\right)\right) = -\frac{\partial\varphi\left(C\left(x\right)\right)}{\partial x}. \qquad (8)$$

This transforms Eq.(1) to

$$J = -D_j \frac{\partial C(x)}{\partial x} + \frac{FD_j}{RT} C(x) E\left(C\left(x\right)\right) \qquad (9)$$

where*J* is the time independent, steady state. Equation (7) can also be recast,

$$\frac{\partial E\left(C\left(x\right)\right)}{\partial x} = \frac{F}{\varepsilon}\left(C\left(x\right) - N\right).$$

(10)

Equations (9) and (10) appear to have no simulation problems. After a bit of examination they reveal themselves to be stiff partial differential equations. Standard methods were used to regularize them[6] and a Runge-Kutta scheme was used for the simulation's calculations.[7]

Let $C\left(x\right) = N\left(1 + \tilde{C}\left(x\right)\right)$ then the above become

$$J = -D_j N \frac{\partial \tilde{C}(x)}{\partial x} + \frac{FD_j N}{RT}\left(\tilde{C}(x) + 1\right) E\left(\tilde{C}\left(x\right)\right) (11)$$

and

$$\frac{\partial E\left(\tilde{C}\left(x\right)\right)}{\partial x} = \frac{FN}{\varepsilon}\tilde{C}\left(x\right). \qquad (12)$$

Let $x = l\tilde{x}$ where *l* is the Nafion film thickness in cm and the above can be rewritten as

$$J = -\frac{D_j N}{l}\frac{\partial \tilde{C}(x)}{\partial x} + \frac{FD_j N}{RT}\left(\tilde{C}(\tilde{x}) + 1\right) E\left(\tilde{C}\left(\tilde{x}\right)\right) \qquad (13)$$

and

$$\frac{\partial E\left(\tilde{C}\left(\tilde{x}\right)\right)}{\partial \tilde{x}} = \frac{FNl}{\varepsilon}\tilde{C}\left(\tilde{x}\right).$$

(14)

It was thought that the electric field values might be too large and create problems in the simulation so $E\left(\tilde{C}\left(\tilde{x}\right)\right)$ was scaled by 10^{-3}. That is $\tilde{E}\left(\tilde{C}\left(\tilde{x}\right)\right) = 10^{-3} E\left(\tilde{C}\left(\tilde{x}\right)\right)$. This scaling and rearrangement yields

$$10^{-3}\frac{\partial \tilde{C}(x)}{\partial x} = -10^{-3}\frac{lJ}{D_j N} + \frac{Fl}{RT}\left(\tilde{C}(\tilde{x}) + 1\right)\tilde{E}\left(\tilde{C}\left(\tilde{x}\right)\right)$$

(15)

and

$$\frac{\partial \tilde{E}\left(\tilde{C}\left(\tilde{x}\right)\right)}{\partial \tilde{x}} = 10^{-3}\frac{FNl}{\varepsilon}\tilde{C}\left(\tilde{x}\right). \qquad (16)$$

Simplifications can now take place. Equations (15) and (16) now become

$$\delta\frac{\partial \tilde{C}(x)}{\partial x} = \alpha + \beta\left(\tilde{C}(\tilde{x}) + 1\right)\tilde{E}\left(\tilde{C}\left(\tilde{x}\right)\right) \qquad (17)$$

and

$$\frac{\partial \tilde{E}\left(\tilde{C}\left(\tilde{x}\right)\right)}{\partial \tilde{x}} = \gamma \tilde{C}\left(\tilde{x}\right).$$

(18)

where $\alpha = -10^{-3}\, lJ/D_j N$, $\boldsymbol{\beta = Fl/RT}$, $\gamma = 10^{-3}\, FNl/\varepsilon$, and $\delta = 10^{-3}$.

The simplifications continue. Let $\hat{E}\left(\tilde{C}\left(\tilde{x}\right)\right) = \tilde{E}\left(\tilde{C}\left(\tilde{x}\right)\right) - E_e$ where $E_e = \alpha/\beta$. Then

$$\begin{aligned}\delta \frac{\partial \tilde{C}(x)}{\partial x} &= \alpha + \beta\left(\tilde{C}(\tilde{x})+1\right)\left(\hat{E}\left(\tilde{C}\left(\tilde{x}\right)\right) + E_e\right) \\ &= \alpha + \beta E_e + \beta E_e\, \tilde{C}\left(\tilde{x}\right) + \beta\left(\, \tilde{C}\left(\tilde{x}\right)+1\right)\tilde{E}\left(\tilde{C}\left(\tilde{x}\right)\right) \\ &= \alpha + \beta\left(-\frac{\alpha}{\beta}\right) + \beta\left(-\frac{\alpha}{\beta}\right)\tilde{C}\left(\tilde{x}\right) + \beta\left(\, \tilde{C}\left(\tilde{x}\right)+1\right)\tilde{E}\left(\tilde{C}\left(\tilde{x}\right)\right) \\ &= -\alpha\tilde{C}\left(\tilde{x}\right) + \beta\left(\, \tilde{C}\left(\tilde{x}\right)+1\right)\tilde{E}\left(\tilde{C}\left(\tilde{x}\right)\right)\end{aligned} \qquad (19)$$

and

$$\begin{aligned}&\frac{\partial \hat{E}\left(\tilde{C}\left(\tilde{x}\right)\right)}{\partial \tilde{x}} = \gamma \tilde{C}\left(\tilde{x}\right) \\ &\hat{E}\left(\tilde{C}\left(\tilde{x}\right)\right) = \tilde{E}\left(\tilde{C}\left(\tilde{x}\right)\right) - E_e\end{aligned} \quad . \qquad (20)$$

Finally let $\boldsymbol{a = \alpha/\delta}$, $\boldsymbol{b = \beta/\delta}$, and $c = \gamma$ along with writing each equation as a function of $\tilde{C}$ and $\hat{E}$.

$$\frac{\partial \tilde{C}\left(\tilde{x}\right)}{\partial \tilde{x}} = -a\tilde{C}\left(\tilde{x}\right) + b\hat{E}\left(\tilde{C}\left(\tilde{x}\right)\right) + b\tilde{C}\hat{E}\left(\tilde{C}\left(\tilde{x}\right)\right) \qquad (21)$$

and

$$\begin{aligned}&\frac{\partial \hat{E}\left(\tilde{C}\left(\tilde{x}\right)\right)}{\partial \tilde{x}} = c\tilde{C}\left(\tilde{x}\right) \\ &\tilde{E}\left(\tilde{C}\left(\tilde{x}\right)\right) = \hat{E}\left(\tilde{C}\left(\tilde{x}\right)\right) - E_e\end{aligned} \quad . \qquad (22)$$

Equations (21) and (22) are the equations used in the Appendix A program. The diffusion coefficients shown in Table 2.

Fuel Cell Type	**Diffusion Coefficient (cm^2/s^{-1})**	**Reference**
Methanol	$2.1x10^{-5}\exp\left\{-\frac{20,460}{R}\left(\frac{1}{T}-\frac{1}{313}\right)\right\}$	[8]
Ethanol	$1.83x10^{-5}$	[1]
Glucose	$6.5x10^{-6}$	[9]

Table 2: Diffusion coefficients for the simulations. *R* is the ideal gas constant and *T* is temperature. The other parameters are found in the program.

To verify that the simulation results were accurate, an equation with a fixed point in the center of the interval over which the simulation was ran was needed. The equation used was

$$\bar{C}\left(x\right) = \bar{E}\left(x\right) + \bar{V}\left(x\right) = \left(x - \frac{l}{2}\right)^2 + \frac{V_0}{l}x - \frac{V_0}{2}$$

(23)

where

$\bar{E}(x)=\left(x-\frac{l}{2}\right)^2$ and

$\bar{V}(x)=\frac{V_0}{l}x-\frac{V_0}{2}$

(24)

The eigenvalue used in the simulation was the value that resulted from $\frac{d\bar{C}}{dx}$ evaluated at x = $l/2$. These equations were entered into the simulation in the form

$dy_1 = y_2 + y_3$

$dy_2 = 2$

$dy_3 = 0$

(25)

where $y_1 = \bar{C}(x), y_2 = \frac{d\bar{E}(x)}{dx}$ and $y_3 = \frac{d\varphi(x)}{dx}$. The resulting curves were as expected, creating the correct parabola for y_1 and the correct lines for y_2 and y_3.

II. RESULTS AND DISCUSSION

For a 52 micrometer Nafion membrane, the excess proton concentration at the membrane, the electric field strength at the membrane, and the voltage at the membrane are shown for all three fuel cells in Figures 2 through 10. The simulation currently does not allow a thickness of 51 micrometers or less. This is likely because of the fixed point assumption and for a membrane 51 micrometers or less, one cannot assume the excess proton concentration is zero in the middle of the film or the position of the zero concentration is too uncertain. It is interesting the simulation breaks down at around the same thickness used in a fuel cell.

It is not clear if this is the result of the simulation method or a characteristic of the modeling equations or the constraints on the system. Without the fixed point assumption, there is no interval within the film where the concentration of protons and Nafion is neutral. This is why the simulation breaks down and could be why such thin films are not stable in fuel cells.

The shapes of the curves across the membrane are similar cell to cell. All are essentially invariant across the membrane. For the excess protons, the membranes show a neutral region across the membrane with steep gradients at boundaries that yield concentration polarizations. That is, protons build up at the anode edge and deplete at the cathode edge.

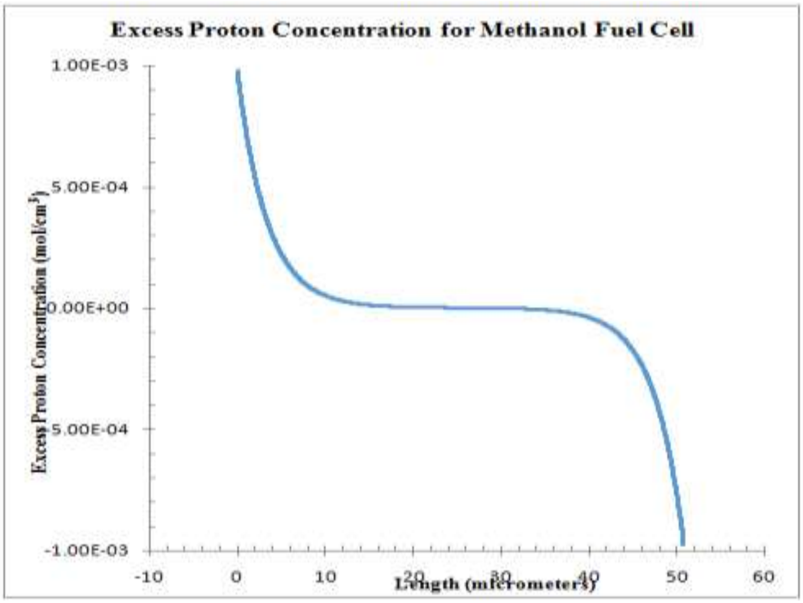

Figure 2: Excess proton concentration in a 52 micrometer Nafion membrane in a methanol fuel cell.

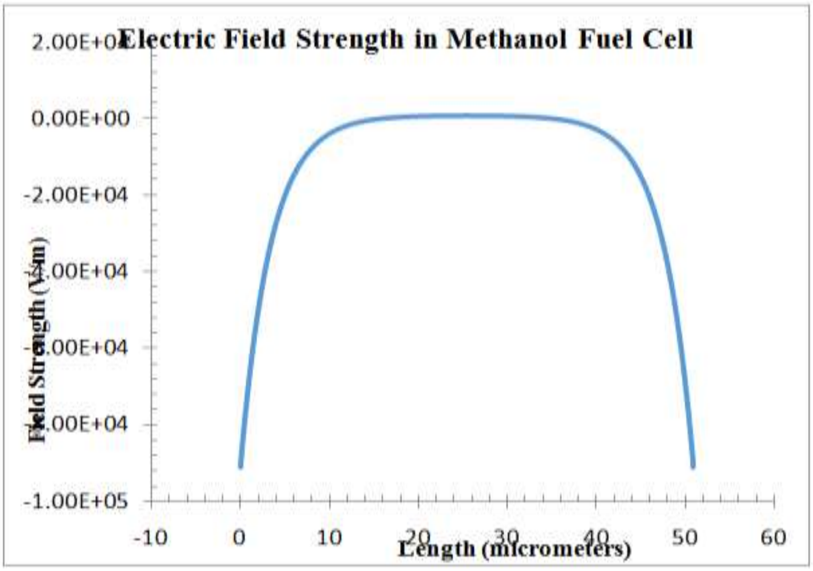

Figure 3: Electric field strength in a 52 micrometer Nafion membrane in a methanol fuel cell.

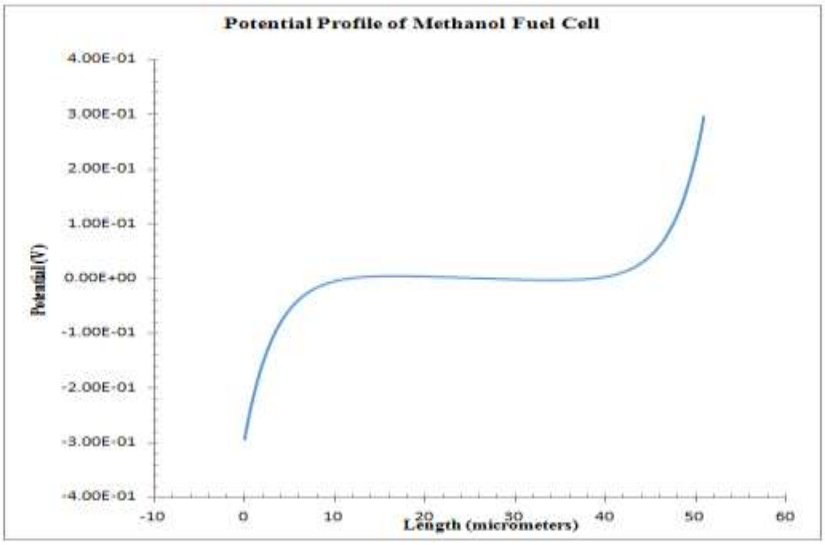

Figure 4: Voltage in a 52 micrometer Nafion membrane in a methanol fueul cell.

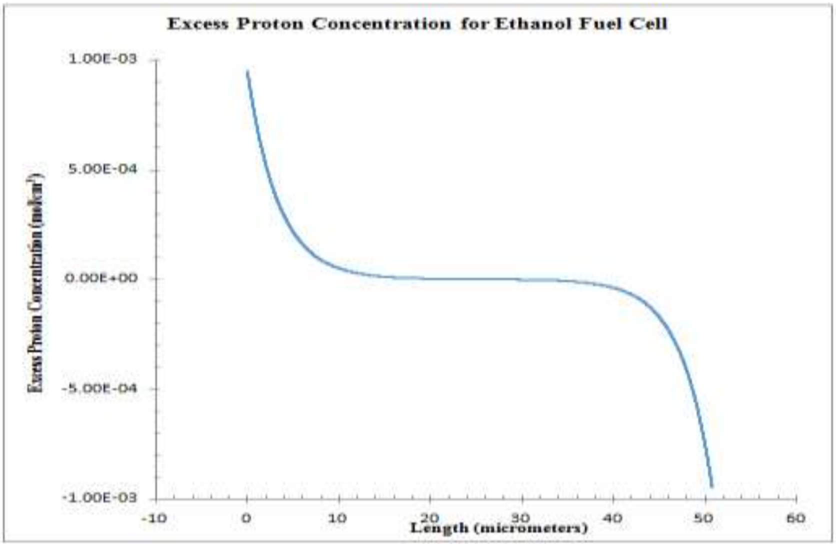

Figure 5: Excess proton concentration in a 52 micrometer Nafion membrane in an ethanol fuel cell.

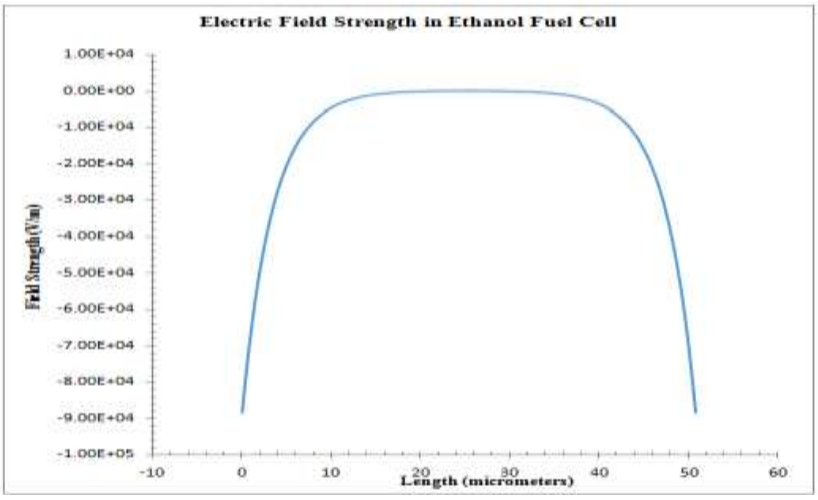

Figure 6: Electric field strength in a 52 micrometer Nafion membrane in an ethanol fuel cell.

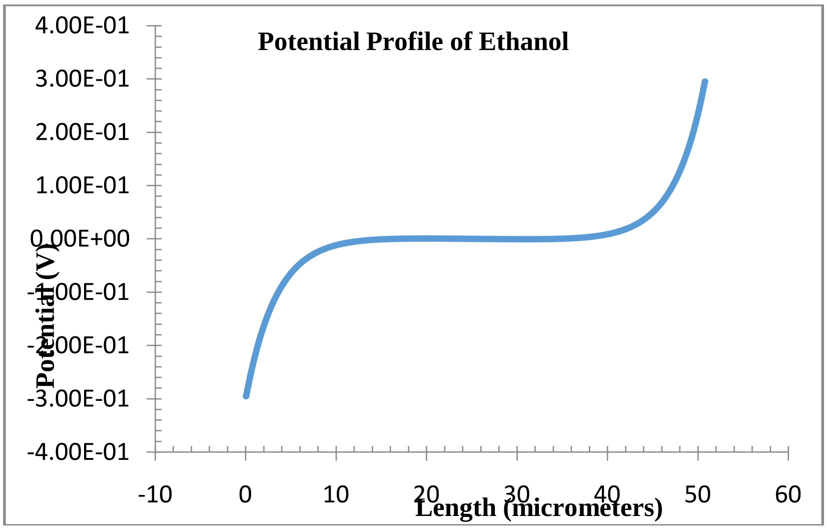

Figure 7: Voltage in a 52 micrometer Nafion membrane in an ethanol fuel cell.

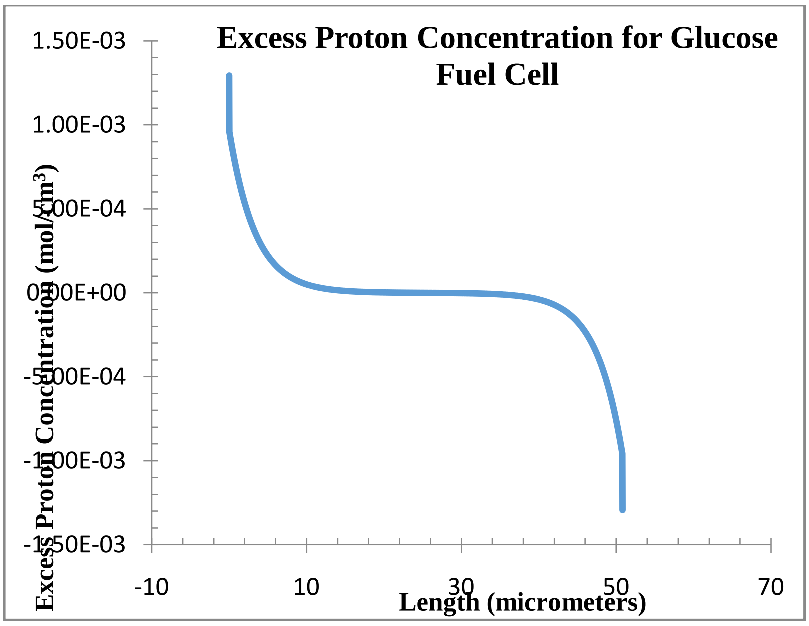

Figure 8: Excess proton concentration in a 52 micrometer Nafion membrane in a glucose fuel cell.

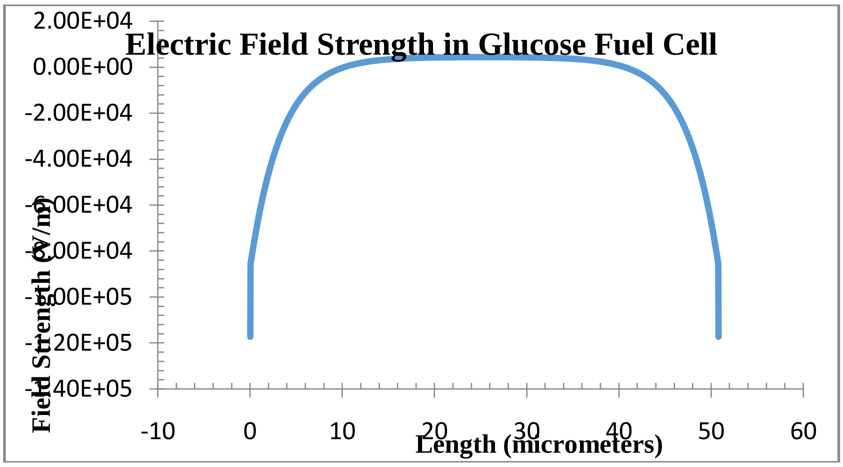

Figure 9: Electric field strength in a 52 micrometer Nafion membrane in a glucose fuel cell.

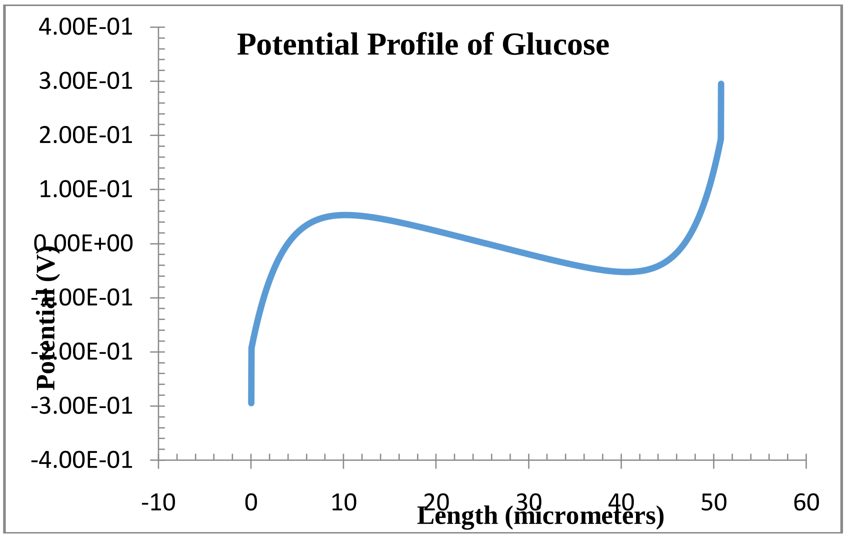

Figure 10: Voltage in a 52 micrometer Nafion membrane in a glucose fuel cell.

Conflict of Interest Statement

The authors declare that there is no conflict of interest regarding the publication of this paper.

Appendix A: Matlab Code for Transport Through Nafion

```
functionGeneralTest
clearall%#ok<CLALL>
global a
global b
global c
global E0
formatlong
length = 50.8; %number of microns interested in
i = 1*10^4; %Electric Density, A/m^2, (from 1 amp/cm^2)
Vo = 0.59; %V lost over membrane
lo = 175; %length in microns
Eo = 1.23;
area = 5*10^(-4); %m^2, 5 cm^2, from Wayneís thesis
ir = i*area; %A, current used for resistance
Vr = Eo-Vo; %V lost to resistance
Ro = Vr/ir; %resistance, Ohms
Res = Ro*length/lo;
I = Vr/Res; %A, adjusts current for the length of interest
ne = 6; %electrons in the reactions
F = 9.64853*10^7; %C/kmol
```

```
R = 8314.47; %J/K*kmol
l = lo*10^(-6)*length/lo; %m, data taken from 50 micron length
J = I/(area*ne*F); %kmol/m^2 s
D = 1.41*10^(-9); %m^2/sec – Change according to molecules Diffusion Coefficient
rho = 1.74*10^6; %g/m^3, 1.74 g/cm^3 average taken from Oberbroeckling/Leddy paper
MW = 1.100*10^6; %g/kmol, taken from 1100 g/mol
N = rho/MW; %kmol/m^3
T = 298; %K
er = 20;
eo = 8.85419*10^(-12); %C^2/(J m)
e = er*eo;
alpha = -10^(-3)*l*J/(D*N);
beta = F*l/(R*T);
gamma = 10^(-3)*F*N*l/e;
delta = 10^(-3);
a=alpha/delta;
b=beta/delta;
c=gamma;
lam1 = (-a+sqrt(a*a+4*b*c))/2;
lam2 = (-a-sqrt(a*a+4*b*c))/2;
M1 = [-a-lam1 b; c -lam1];
M1(2,2) = M1(1,2)*-M1(2,1)/M1(1,1) + M1(2,2);
M1(2,1) = 0; %changes to rref
ratio1 = abs(M1(1,2)/M1(2,2));
ratio2 = abs(M1(1,1)/M1(2,2));
%if M(2,2) is small enough, have one free variable to determine the eigenvectors
if ratio1 > 10^6
if ratio2 > 10^6
PosEigVec = [-b/(-a-lam1) 1]/10000;
end
end
M2 = [-a-lam2 b; c -lam2];
M2(2,2) = M2(1,2)*-M2(2,1)/M2(1,1) + M2(2,2);
M2(2,1) = 0; %changes matrix to rref
ratio1 = abs(M2(1,2)/M2(2,2));
ratio2 = abs(M2(1,1)/M2(2,2));
%if M(2,2) is small enough, have one free variable to determine the eigenvectors
if ratio1 > 10^6
if ratio2 > 10^6
NegEigVec = [-b/(-a-lam2) 1]/10000;
end
end
E0 = -alpha/beta;
n=1000;
Ee = -alpha/beta;
IC = Vo/(10^3*l);
del1= l/2;
del2= l/2;
X1=linspace(0,del1,n);
X2=linspace(0,del2,n);
try
XI1=[NegEigVec(1,1) -.1 0]';
[X1,Y]=ode15s(@F2,X1,XI1);
catch
del1=.999*del1;
X1=linspace(0,del1,n);
end
C1=Y*[1 0 0]';
E1=Y*[0 1 0]'+E0;
```

```
V1=-Y*[0 0 1]';
C1 = N*(1+C1);
E1 = 10^(3)*E1;
V1 = V1*10^(3);
C1min = min(C1);
C1max = max(C1);
E1min = min(E1);
E1max = max(E1);
deltax1 = del1/n;
try
XI2=[PosEigVec(1,1) -.1 0]';
[X2,Y]=ode15s(@F1,X2,XI2);
catch
del2=.999*del2;
X2=linspace(0,del2,n);
end
C2=Y*[1 0 0]';
E2=Y*[0 1 0]'+E0;
V2=-Y*[0 0 1]';
C2 = N*(1+C2);
E2 = 10^(3)*E2;
V2 = V2*10^(3);
C2min = min(C2);
C2max = max(C2);
E2min = min(E2);
E2max = max(E2);
deltax2 = del2/n;
A = [E1min, E2min];
Amin = max(A);
h1 = size(X1);
h2 = size(X2);
n1 = h1(1,1);
n2 = h2(1,1);
VaAll = V1(n1)-V2(n2);
x1 = 0;
while E1(n1-x1)<Amin
x1 = x1+1;
end
x2 = 0;
while E2(n2 - x2)<Amin
x2 = x2+1;
end
for x2 = 1:n2-1
x1 = 0;
Amin = E2(n2-x2);
while E1(n1-x1)<Amin
x1 = x1+1;
end
Va(n2-x2,1) = V1(n1-x1)-V2(n2-x2);
end
xv=1;
whileVa(n2-xv,1)> Vo
xv = xv+1;
end
x1 = 0;
x2 = xv-1;
Amin = E2(n2-x2);
while E1(n1-x1)<Amin
x1 = x1+1;
```

```
end
Creal = zeros(2*n, 1);
Ereal = zeros(2*n, 1);
Vreal = zeros(2*n, 1);
xfinal = zeros(2*n, 1);
Cfinal = zeros(2*n, 1);
Efinal = zeros(2*n, 1);
Vfinal = zeros(2*n, 1);
xreal = zeros(2*n,1);
for xx = 1:n2-x2
xreal(xx) = deltax2*(xx-1);
Creal(xx) = C2(n2-x2-xx+1);
Ereal(xx) = E2(n2-x2-xx+1);
Vreal(xx) = V2(n2-x2-xx+1);
end
zro = 2*n-(n2-x2+n1-x1);
deltaxmid = (l-(n2-x2)*deltax2-(n1-x1)*deltax1)/zro;
for xx = 1:zro
xreal(n2-x2+xx) = xreal(n2-x2+xx-1)+deltaxmid;
Creal(n2-x2+xx) = N;
Ereal(n2-x2+xx) = Ee*10^3;
Vreal(n2-x2+xx) = 0;
end
for xx = 1:n1-x1
xreal(n2-x2+zro+xx) = l-deltax1*(n1-x1-xx);
Creal(n2-x2+zro+xx) = C1(xx);
Ereal(n2-x2+zro+xx) = E1(xx);
Vreal(n2-x2+zro+xx) = V1(xx);
end
line = 5; %number of points used to make the linearization
FE = zeros(2,line);
LE = zeros(2,line);
FV = zeros(2,line);
LV = zeros(2,line);
FC = zeros(2,line);
LC = zeros(2,line);
for xx = 1:line
FE(1,xx) = (xx-1)*deltax2;
FE(2,xx) = Ereal(xx);
LE(1,xx) = l-(line-xx)*deltax1;
LE(2,xx) = Ereal(2*n-line+xx);
FV(1,xx) = (xx-1)*deltax2;
FV(2,xx) = Vreal(xx);
LV(1,xx) = l-(line-xx)*deltax1;
LV(2,xx) = Vreal(2*n-line+xx);
FC(1,xx) = (xx-1)*deltax2;
FC(2,xx) = Creal(xx);
LC(1,xx) = l-(line-xx)*deltax1;
LC(2,xx) = Creal(2*n-line+xx);
end
RFE = corrcoef(FE'); %gives correlation coefficient for line
RLE = corrcoef(LE'); %made at the boundaries
RFV = corrcoef(FV');
RLV = corrcoef(LV');
RFC = corrcoef(FC');
RLC = corrcoef(LC');
pFE = polyfit(FE(1,:),FE(2,:),1);
pLE = polyfit(LE(1,:),LE(2,:),1);
pFV = polyfit(FV(1,:),FV(2,:),1);
```

```
pLV = polyfit(LV(1,:),LV(2,:),1);
pFC = polyfit(FC(1,:),FC(2,:),1);
pLC = polyfit(LC(1,:),LC(2,:),1);
clearM;
M(1,1) = pFE(1,1);
M(1,2) = -pLE(1,1);
M(1,3) = pLE(1,2) - pFE(1,2);
M(2,1) = -pFV(1,1);
M(2,2) = pLV(1,1);
M(2,3) = Vo - pLV(1,2) + pFV(1,2);
Xs = rref(M);
xi = Xs(1,3);
xf = Xs(2,3);
cleartotal
xfinal(1) = 0;
Cfinal(1) = pFC(1,1)*xi + pFC(1,2);
Efinal(1) = pFE(1,1)*xi + pFE(1,2);
Vfinal(1) = pFV(1,1)*xi + pFV(1,2);
xfinal(2) = deltax2 - xi;
Cfinal(2) = Creal(2);
Efinal(2) = Ereal(2);
Vfinal(2) = Vreal(2);
for x = 3:2*n-1
Cfinal(x) = Creal(x);
Efinal(x) = Ereal(x);
Vfinal(x) = Vreal(x);
end
for x = 3:n2-x2
xfinal(x) = xfinal(x-1) + deltax2;
end
deltaxmid = (l-(n2-x2)*deltax2-(n1-x1)*deltax1)/zro;
for xx = 1:zro
xfinal(n2-x2+xx) = xfinal(n2-x2+xx-1)+deltaxmid;
end
for xx = 1:n1-x1-2
xfinal(n2-x2+zro+xx) = l-deltax1*(n1-x1-xx);
end
xfinal(2*n-1) = 2*l-deltax1-xf;
Cfinal(2*n-1) = Creal(2*n-1);
Efinal(2*n-1) = Ereal(2*n-1);
Vfinal(2*n-1) = Vreal(2*n-1);
xfinal(2*n) = l;
Cfinal(2*n) = pLC(1,1)*xf + pLC(1,2);
Efinal(2*n) = pLE(1,1)*xf + pLE(1,2);
Vfinal(2*n) = pLV(1,1)*xf + pLV(1,2);
Vfinal(2*n)-Vfinal(1) %check potential drop
Efinal(2*n)-Efinal(1) %checks charge over membrane
sum1 = 0;
for x = 1:n2-x2
sum1 = sum1 + Cfinal(x);
end
sum1 = sum1*deltax2;
sum2 = 0;
for x = 1:n1-x1
sum2 = sum2 + Cfinal(2*n-x+1);
end
sum2 = sum2*deltax1;
sum3 = 0;
if n2-x2 < n
```

```
for x = n2-x2+1:2*n-n1+x1;
sum3 = sum3 + Cfinal(x);
end
sum3 = sum3*deltaxmid;
end
sum = sum1 + sum2 + sum3 - N*l; %gives excess protons, not good bc trap rule
%checks that ignoring b*Ctilde*Ehat is ok
Ctilde = Cfinal/N - 1;
Ehat = 10^(-3)*Efinal + alpha/beta;
for x = 1:2*n
Other(x,1) = Ctilde(x,1)*Ehat(x,1);
end
clearA
clearB
clearC
for x = 1:2*n
A(x,1) = abs(Ehat(x,1)/Other(x,1));
B(x,1) = abs(b*Ehat(x,1)/(-a*Ctilde(x,1)));
C(x,1) = abs(b*Other(x,1)/(-a*Ctilde(x,1)));
end
maxA = max(A);
minA = min(A);
maxB = max(B);
minB = min(B);
maxC = max(C);
minC = min(C);
for x = 1:2*n
All(x,1) = -a*Ctilde(x,1)+b*Ehat(x,1)*(1+Ctilde(x));
Assume(x,1) = -a*Ctilde(x,1)+b*Ehat(x,1);
Diff(x,1) = 100*(All(x,1)-Assume(x,1))/All(x,1);
end
maxDiff = max(Diff); %tells what % off
minDiff = min(Diff);
xfinal; %Output of these values are retrieved by removing the semicolon
Cfinal-N; %excess proton concentration
Vfinal; %potential
Efinal; %electric field
end
function F1=F1(T,Y)
global a
global b
global c
global E0
F1=-[-a*Y(1)+b*Y(2)*(1+Y(1));c*Y(1);Y(2)+E0];
end
function F2=F2(T,Y)
global a
global b
global c
global E0
F2=[-a*Y(1)+b*Y(2)*(1+Y(1));c*Y(1);Y(2)+E0];
end
```

REFERENCES

[1]. Andreadis, G et al. (2006) Direct Ethanol Fuel Cell Anode Simulation Model. J. of Power Sources, 157, 657-665.

[2]. Deng, H. et al. (2015) Modeling of Passive Alkaline Membrane Direct Methanol Fuel Cell. Electro. Acta, 154, 430-446.

[3]. Yu, E. H. et al. (2012) Direct Oxidation Fuel Cells: from Materials to Systems. Energy Environ. Sci., 5, 5668-5680.

[4]. Paddison, S.J. et al. (1998) High Frequency Dielectric Studies of Hydrated Nafion. J. Electroanalytical Chem. 459, 91-97.

[5]. Schmidt, S. (2010) Mathematical Models of Ion Transport Through Nafion Membranes in Modified Electrodes and Fuel Cells Without Electroneutrality. Ph. D. Dissertation, University of Iowa, Ames.

[6]. Djilia, N. et al. (2007) Numerical Optimization of Proton Exchange Membrane Fuel Cell Cathodes. Electro. Acta 52, 2668-2682.

[7]. Kassam, A. et al. (2005) Fourth-Order Time Stepping for Stiff PDEs. SIAM J. of Sci. Comput. 4, 1214-1233.

[8]. Rosenthal N. et al. (2012) A Comprehensive Yet Comprehensible Analytical Model For The Direct Methanol Fuel Cell. J. Power Sources 206, 129-143.

[9]. Ranoo, P. et al. (2013) Mathematical Modeling And Experimental Verification of Direct Glucose Anion Exchange Membrane Fuel Cell. 113, 42-53.