Polymer Electrolyte Fuel Cell Model
based on the work given in:
T. E. Springer, T. A. Zawodzinski, and S. Gottesfeld, J. Electrochem. Soc., 138, 2334 (1991)
This code uses the cubic approximation for the corrected diffusivity for all values of the water content to provide an accurete enough guess that can be used when the piecewise version of Dlambda is used. Still unstable at high rates (singularity error).

```
> restart;
> with(plots):
> digits:=15;
```

$$digits := 15$$

```
> pars:={iapp=0.5,i0=0.01,Hflow=2, Oflow=3, Pa=3, Pc=3,tA=0.0365,
  tC=0.0365,tMem=0.0175,Tcell=80,Tsat=80,Voc=1.1,xON=0.21,
  xwA=0.1558,xwC=0.1558,PcO=49.8  ,PcH=12.8  ,PcN=33.5,   PcW=217.7
  ,TcO=155  ,TcH=33.2
  ,TcN=126,TcW=647.096,e=0.5,MO=32,MW=18,MN=14,MH=2,F=96485,s=0.0126
  ,R=8.3145,xON=0.21,xIwA=0.1558,xIwC=0.1558};
```

$$pars := \{F = 96485, MH = 2, MN = 14, MO = 32, MW = 18, Pa = 3, Pc = 3, PcH = 12.8, PcN = 33.5, PcO = 49.8, PcW = 217.7, R = 8.3145, TcH = 33.2, TcN = 126, TcO = 155, TcW = 647.096, Voc = 1.1, e = 0.5, i0 = 0.01, s = 0.0126, tA = 0.0365, tC = 0.0365, xON = 0.21, xwA = 0.1558, xwC = 0.1558, Hflow = 2, Oflow = 3, Tcell = 80, Tsat = 80, iapp = 0.5, tMem = 0.0175, xIwA = 0.1558, xIwC = 0.1558\}$$

Ratio of H2 supllied to H2 consumed

```
> nu[H]:=Hflow/iapp;
```

$$\nu_H := \frac{Hflow}{iapp}$$

Ratio of O2 supllied to O2 consumed

```
> nu[O]:=Oflow/iapp;
```

$$\nu_O := \frac{Oflow}{iapp}$$

## Diffusion Coefficients

Calculated from Equation 10

Diffusion Coefficient  for water (vapor) and hydrogen gas

```
> DwH:=evalf(subs(pars,0.000364*(Tcell/(TcW*TcH)^(1/2))^(2.334)*(PcW
  *PcH)^(1/3)*(TcH*TcW)^(5/12)*(1/MH+1/MW)^(1/2)*e^(3/2)/Pa));
```

$$DwH := 0.006989448527$$

Diffusion Coefficient  for water (vapor) and oxygen gas

```
> DwO:=evalf(subs(pars,0.000364*(Tcell/(TcW*TcO)^(1/2))^(2.334)*(PcW
  *PcO)^(1/3)*(TcO*TcW)^(5/12)*(1/MO+1/MW)^(1/2)*e^(3/2)/Pc));
```

$$DwO := 0.001367438403$$

Diffusion Coefficient  for water (vapor) and nitrogen gas

```
> DwN:=evalf(subs(pars,0.000364*(Tcell/(TcW*TcN)^(1/2))^(2.334)*(PcW
  *PcN)^(1/3)*(TcN*TcW)^(5/12)*(1/MN+1/MW)^(1/2)*e^(3/2)/Pc));
```

$$DwN := 0.001692835421$$

Diffusion Coefficient for oxygen and nitrogen gas

```
> DON:=evalf(subs(pars,0.0002745*(Tcell/(TcN*TcO)^(1/2))^(1.832)*(Pc
  N*PcO)^(1/3)*(TcO*TcN)^(5/12)*(1/MO+1/MN)^(1/2)*e^(3/2)/Pc));
```

$$DON := 0.002714423430$$

# Governing Equations

Governing equation for mole fraction for water in the anode (Eq.11) (alpha is the ratio of water flux in the membrane to that produced in the cathode)

```
> Eq1:=diff(xwa(z),z)=R*Tcell*iapp/(2*F)/(Pa*DwH)*(xwa(z)*(1+alpha)-
  alpha);
```

$$Eq1 := \frac{d}{dz}\,\mathrm{xwa}(z) = \frac{71.53640205\,R\,Tcell\,iapp\,(\mathrm{xwa}(z)\,(1+\alpha)-\alpha)}{F\,Pa}$$

Governing equation for mole fraction for oxygen in the cathode (Eq.12)

```
> Eq2:=diff(xO(z),z)=R*Tcell*iapp/(2*F)/Pc*((xO(z)*(1+alpha)+0.5*xwc
  (z))/DwO+(1-xwc(z)-xO(z))/DON);
```

$$Eq2 := \frac{d}{dz}\,\mathrm{xO}(z) = \frac{1}{2}\,R\,Tcell\,iapp\,(731.2943660\,\mathrm{xO}(z)\,(1+\alpha) - 2.7551779\,\mathrm{xwc}(z) + 368.4023609 - 368.4023609\,\mathrm{xO}(z))\,/\,(F\,Pc)$$

Governing equation for mole fraction for water in the cathode (Eq.13)

```
> Eq3:=diff(xwc(z),z)=R*Tcell*iapp/(2*F)/Pc*(((1-xwc(z)-xO(z))*(1+al
  pha))/DwN+(0.5*xwc(z)+xO(z)*(1+alpha))/DON);
```

$$Eq3 := \frac{d}{dz}\,\mathrm{xwc}(z) = \frac{1}{2}\,R\,Tcell\,iapp\,(590.7248794\,(1-\mathrm{xwc}(z)-\mathrm{xO}(z))\,(1+\alpha) + 368.4023609\,\mathrm{xO}(z)\,(1+\alpha) + 184.2011804\,\mathrm{xwc}(z))\,/\,(F\,Pc)$$

Mole fraction of water at cathode interface (Eq. 7)

```
> xw4:=subs(pars,(xIwC*nu[O]+2*(1+alpha)*(1-xIwC)*xON)/(nu[O]+(2*alp
  ha+1)*(1-xIwC)*xON));
```

$$xw4 := \frac{1.289364000 + 0.354564\,\alpha}{6.177282000 + 0.354564\,\alpha}$$

Mole fraction of oxygen at cathode interface (Eq. 7)

```
> xO4:=subs(pars,((nu[O]-1)*(1-xIwC)*xON)/(nu[O]+(2*alpha+1)*(1-xIwC
  )*xON));
```

$$xO4 := \frac{0.8864100000}{6.177282000 + 0.354564\,\alpha}$$

Mole fraction of water at anode interface (Eq. 8)

```
> xw1:=subs(pars,(nu[H]*xIwA-alpha*(1-xIwA))/(xIwA-alpha*(1-xIwA)+nu
  [H]-1));
```

$$xw1 := \frac{0.6232000000 - 0.8442\,\alpha}{3.155800000 - 0.8442\,\alpha}$$

Solve for the water mole fraction profile in the anode

```
> sol1:=subs(pars,dsolve({Eq1,xwa(0)=xw1}));
```

$$sol1 := \mathrm{xwa}(z) = \frac{\alpha}{1+\alpha} + \mathbf{e}^{(0.08219438804\,(1+\alpha)\,z)}\left(\frac{-3116+4221\,\alpha}{-15779+4221\,\alpha} - \frac{\alpha}{1+\alpha}\right)$$

Solve for the water and oxygen mole fraction profile in the cathodes

```
> sol2:=evalf(subs(pars,dsolve({Eq2,Eq3,xwc(0)=xw4,xO(0)=xO4}))):
> assign(sol1);
> assign(sol2);
```

Solve for water mole fraction at anode/membrane interface using above solution

```
> xw2:=(subs(z=tA,pars,xwa(z)));
```

$$xw2 := \frac{\alpha}{1+\alpha} + \mathbf{e}^{(0.003000095163\,+\,0.003000095163\,\alpha)}\left(\frac{-3116+4221\,\alpha}{-15779+4221\,\alpha} - \frac{\alpha}{1+\alpha}\right)$$

Solve for water mole fraction at cathode/membrane interface using above solution

```
> xw3:=subs(z=tC,pars,xwc(z)):
```

Solve for oxygen mole fraction at cathode/membrane interface using above solution

```
> xO3:=(subs(z=tC,pars,xO(z))):
```

Saturatation vapor presure of water (Eq. 15)

```
> Psat:=subs(pars,10^(-2.1794+0.02953*Tsat-9.1837e-5*Tsat^2+1.4454e-
  7*Tsat^3));
```

$$Psat := 0.4669255941$$

Water vapor activity

```
> a:=(xw(z)*Pc/Psat);
```

$$a := 2.141668850\,\mathrm{xw}(z)\,Pc$$

Water content (# water molecules per charge site) in membrane when a<=1 (Eq. 16)

```
> lambda1:=(subs(pars,0.043+17.81*a-39.85*a^2+36*a^3));
```

$$\lambda 1 := 0.043 + 114.4293667\,\mathrm{xw}(z) - 1645.036260\,\mathrm{xw}(z)^2 + 9548.237764\,\mathrm{xw}(z)^3$$

Water content in membrane when a>1 (Eq. 17)

```
> lambda2:=subs(pars,14+1.4*(a-1));
```

$$\lambda 2 := 12.6 + 8.995009170\,\mathrm{xw}(z)$$

Water content at membrane/cathode interface

```
> lambdaxw3:=subs(xw(z)=xw3,pars,piecewise(a<=1,lambda1,a>1,lambda2)
  ):
```

Water content at membrane/anode interface

```
> lambdaxw2:=subs(xw(z)=xw2,pars,piecewise(a<=1,lambda1,a>1,lambda2)
  );
```

$$lambdaxw2 := \{\, 0.043 + \frac{114.4293667\,\alpha}{1+\alpha} + 114.4293667\, \mathbf{e}^{(0.003000095163 + 0.003000095163\,\alpha)} \left( \frac{-3116+4221\,\alpha}{-15779+4221\,\alpha} - \frac{\alpha}{1+\alpha} \right) - 1645.036260 \left( \frac{\alpha}{1+\alpha} + \mathbf{e}^{(0.003000095163 + 0.003000095163\,\alpha)} \left( \frac{-3116+4221\,\alpha}{-15779+4221\,\alpha} - \frac{\alpha}{1+\alpha} \right) \right)^2 + 9548.237764 \left( \frac{\alpha}{1+\alpha} + \mathbf{e}^{(0.003000095163 + 0.003000095163\,\alpha)} \left( \frac{-3116+4221\,\alpha}{-15779+4221\,\alpha} - \frac{\alpha}{1+\alpha} \right) \right)^3 ,$$

$$\frac{6.425006550\,\alpha}{1+\alpha} + 6.425006550\, \mathbf{e}^{(0.003000095163 + 0.003000095163\,\alpha)} \left( \frac{-3116+4221\,\alpha}{-15779+4221\,\alpha} - \frac{\alpha}{1+\alpha} \right) \le 1$$

$$12.6 + \frac{8.995009170\,\alpha}{1+\alpha} + 8.995009170\, \mathbf{e}^{(0.003000095163 + 0.003000095163\,\alpha)} \left( \frac{-3116+4221\,\alpha}{-15779+4221\,\alpha} - \frac{\alpha}{1+\alpha} \right)$$

$$, 1 < \frac{6.425006550\,\alpha}{1+\alpha} + 6.425006550\, \mathbf{e}^{(0.003000095163 + 0.003000095163\,\alpha)} \left( \frac{-3116+4221\,\alpha}{-15779+4221\,\alpha} - \frac{\alpha}{1+\alpha} \right)$$

Number of water molecules per proton transfered

```
> ndrag:=2.5*lambda(z)/22;
```

$$ndrag := 0.1136363636\,\lambda(z)$$

Assume Dprime is linear for lambda<4. See Figure 3

```
> Dprime:=exp(2416*(1/303-1/(273+Tcell1)))*(a1*lambda(z)+b1);
```

$$Dprime := \mathbf{e}^{\left( \frac{2416}{303} - \frac{2416}{273 + Tcell1} \right)} (a1\,\lambda(z) + b1)$$

Solve for corrected diffusion coefficient for lambda less than 4 (Eq. 21)

```
> Dlambda1:=subs(pars,(1/(1+s*lambda(z))^2*lambda(z)/(a*(17.81-79.7*
  a+108*a^2))))*Dprime;
```

$$Dlambda1 := \frac{0.1556418647\,\lambda(z)\, \mathbf{e}^{\left( \frac{2416}{303} - \frac{2416}{273 + Tcell1} \right)} (a1\,\lambda(z) + b1)}{(1 + 0.0126\,\lambda(z))^2\, \mathrm{xw}(z)\,(17.81 - 512.0730219\,\mathrm{xw}(z) + 4458.316590\,\mathrm{xw}(z)^2)}$$

Solve for corrected diffusion coefficient for lambda greater than 4 (Eq. 22)

```
> Dlambda4:=evalf(subs(pars,1e-6*exp(2416*(1/303-1/(273+Tcell1)))*(2
  .563-0.33*lambda(z)+0.0264*lambda(z)^2-0.000671*lambda(z)^3)));
```

$$Dlambda4 :=$$

$$0.1\;10^{-5}\, \mathbf{e}^{\left( 7.973597360 - \frac{2416.}{273. + Tcell1} \right)} (2.563 - 0.33\,\lambda(z) + 0.0264\,\lambda(z)^2 - 0.000671\,\lambda(z)^3)$$

Solve for the coefficeints for Dprime by assuming the corrected diffusion coefficient is continuous at Dlambda=4, and Dprime=0.6e-6 at lambda=2 and T=30 deg

```
> use RealDomain in
  Temp:=solve({4=lambda1,subs(lambda(z)=4,Tcell1=30,Dlambda1)=subs(l
  ambda(z)=4,Tcell1=30,Dlambda4),subs(lambda(z)=2,Tcell1=30,Dprime)=
  0.6e-6},{xw(z),a1,b1}) end use;
```

$$Temp := \{ a1 = 0.7357235260\ 10^{-6},\ b1 = -0.8714470520\ 10^{-6},\ \mathrm{xw}(z) = 0.09044297269 \}$$

Solve for the water mole fraction in terms of water content in the seperator-- since Eq. 16 is cubic, the solution is limited to the real solution

This ensures that the water content (lambda(z)) is the only dependent variable

```
> use RealDomain in Temp1:=solve({lambda(z)=lambda1},{xw(z)}) end
  use;
```

$$Temp1 := \{ \mathrm{xw}(z) = 0.1024000000\ 10^{-33}\ (0.2022301\ 10^{7}\ (-0.1767584599\ 10^{80} + 0.5896706604\ 10^{79}\ \lambda(z) + 0.5489512422\ 10^{74} \sqrt{0.1051024347\ 10^{12} - 0.6917547978\ 10^{11}\ \lambda(z) + 0.1153855687\ 10^{11}\ \lambda(z)^2}\ )\text{^}(2/3) - 0.3285467212\ 10^{59} + 0.5608297441\ 10^{33}\ (-0.1767584599\ 10^{80} + 0.5896706604\ 10^{79}\ \lambda(z) + 0.5489512422\ 10^{74} \sqrt{0.1051024347\ 10^{12} - 0.6917547978\ 10^{11}\ \lambda(z) + 0.1153855687\ 10^{11}\ \lambda(z)^2}\ )\text{^}(1/3)\ ) \Big/ (-0.1767584599\ 10^{80} + 0.5896706604\ 10^{79}\ \lambda(z) + 0.5489512422\ 10^{74} \sqrt{0.1051024347\ 10^{12} - 0.6917547978\ 10^{11}\ \lambda(z) + 0.1153855687\ 10^{11}\ \lambda(z)^2}\ )\text{^}(1/3)\ \}$$

```
> a1:=subs(Temp,a1);b1:=subs(Temp,b1);
```

$$a1 := 0.7357235260\ 10^{-6}$$

$$b1 := -0.8714470520\ 10^{-6}$$

```
> Dlambda1:=subs(Temp1,Tcell1=Tcell,pars,Dlambda1):
```

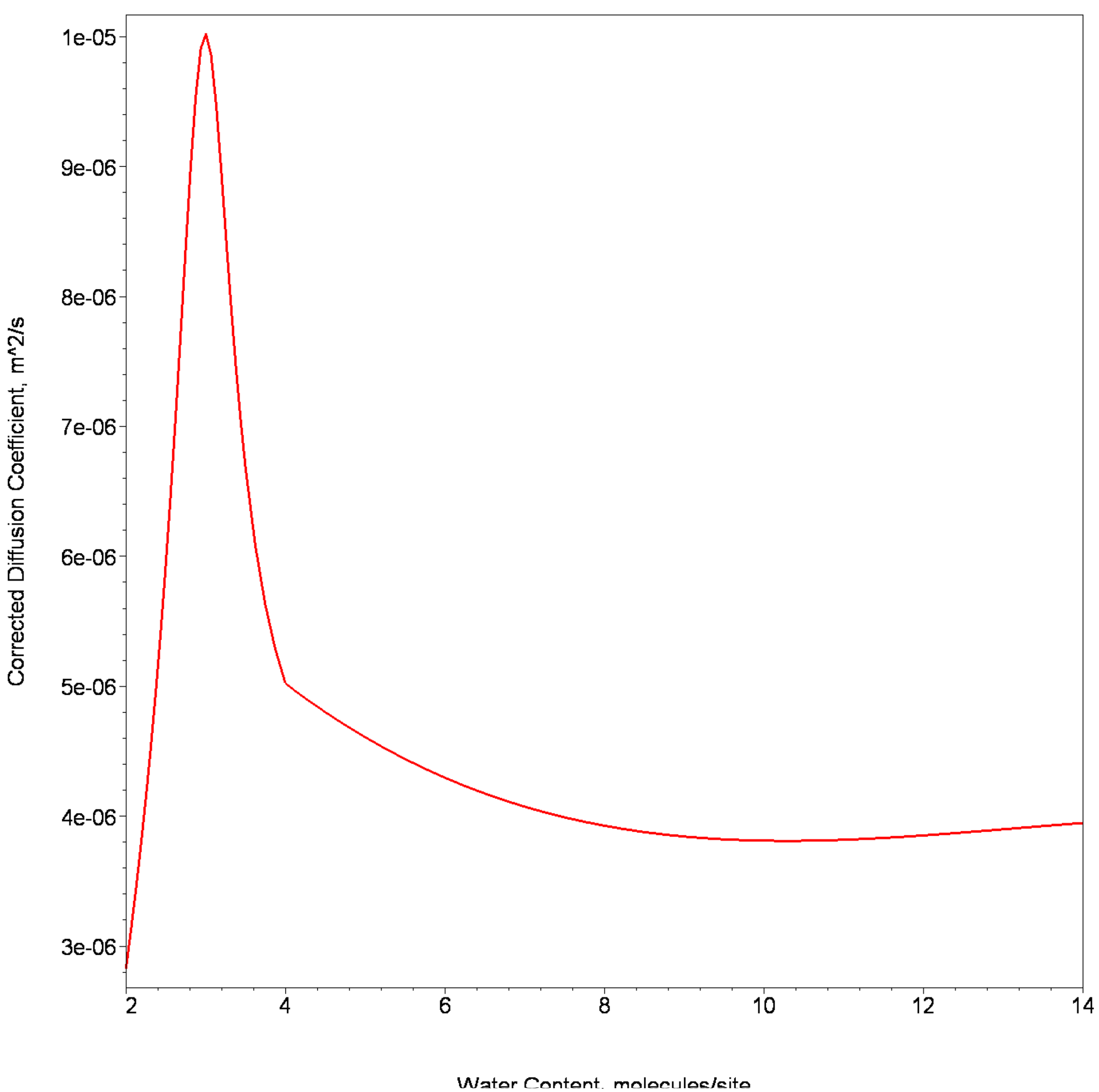
```
> Dlambda4:=subs(Tcell1=Tcell,pars,Dlambda4):
```

Total corrected diffusion coefficient--use cubic representation for entire domain

```
> Dlambda:=Dlambda4:
```

Plot of corrected diffusion coefficient with lambda

```
> plot(subs(lambda(z)=y,Dlambda),y=2..14,axes=boxed,thickness=3,labe
  ls=["Water Content, molecules/site","Corrected Diffusion
  Coefficient, m^2/s"],labeldirections=[Horizontal, Vertical]);
```



Governing equation for water content in the membrane

```
> Eq4:=subs(pars,diff(lambda(z),z)=(2*ndrag-alpha)*(iapp/(2*F))*1155
  /(1*Dlambda)):
>
```

Solve for water content in the membrane as a function of alpha

```
> soltest:=dsolve({Eq4,lambda(0)=lambdaxw2},numeric,parameters=[alph
  a],abserr=1e-12);
                         soltest := proc(x_rkf45)  ...  end proc
```

# Determine alpha

Solves for alpha so that the water content calculated at the anode/membrane interface as calculated by sol1 above (lambdaxw3) agrees with water content calculated by soltest.
If solution fails, try a different initial alpha guess--this uses a cubic approximation for Dlambda over the entire domain

```
> #alphaguess:=subs(fsol,alpha);
  alphaguess:=0.8:
  errout:=1:
  soltest('parameters'=[alphaguess]):
  soltest(subs(pars,tMem)):
  evalf(subs(alpha=alphaguess,lambdaxw3)):
  err:=evalf(rhs(soltest(subs(pars,tMem))[2])-subs(alpha=alphaguess,
  lambdaxw3));
  while errout>1e-5 do
  soltest('parameters'=[1.000001*alphaguess]):
> derr1:=(evalf(rhs(soltest(subs(pars,tMem))[2])-subs(alpha=1.000001
  *alphaguess,lambdaxw3))):
  soltest('parameters'=[0.999999*alphaguess]):
> derr2:=(evalf(rhs(soltest(subs(pars,tMem))[2])-subs(alpha=0.999999
  *alphaguess,lambdaxw3))):
  derr:=(derr1-derr2)/(2*0.000001*alphaguess);
  alphaguessnew:=alphaguess-err/derr;
  cont:=true:
  while cont=true do
  s11:='s11':
  try
  soltest('parameters'=[alphaguessnew]);
  s11:=evalf(rhs(soltest(subs(pars,tMem))[2])-subs(alpha=alphaguessn
  ew,lambdaxw3));
  catch:
  end try:

  if is(s11,numeric)=true then

  err:=s11:
  alphaguess:=alphaguessnew:
  cont:=false:
  else
  alphaguessnew:=(alphaguess+alphaguessnew)/2:
  print(alphaguessnew);
  cont:=true:
  end:
  end:
```

```
  errout:=sqrt(err^2):
  print(subs((soltest(subs(pars,tMem)),lambda(z))),evalf(subs(alpha=
  alphaguess,lambdaxw3)),errout);

  end:
```

```
err := -24.90723782
0.6058482875
2.29553962854013616, 14.85092828, 12.55538865
0.5379074094
11.7260800519317706, 14.81999008, 3.093910030
14.9347054171019788, 14.81230778, 0.1223976400
14.8126632561494258, 14.81258837, 0.00007489000000
14.8125885944261154, 14.81259046, 0.1870000000 10^-5
```

Redefine Dlambda as piecwise accross the domain

```
> Dlambda:=piecewise(lambda(z)<=4,Dlambda1,lambda(z)>4,Dlambda4):
```

Governing equation for water content in the membrane

```
> Eq4:=subs(pars,diff(lambda(z),z)=(2*ndrag-alpha)*(iapp/(2*F))*1155
  /(1*Dlambda)):
> alphaguess;
```

```
0.5217342266
```

Solve for water content in the membrane as a function of alpha using the piecwise Dlambda

```
> soltest:=dsolve({Eq4,lambda(0)=lambdaxw2},numeric,parameters=[alph
  a],abserr=1e-12):
```

Use the alpha value determined by using the cubic Dlamba as an initial guess when using the piecewise Dlambda

```
> errout:=1:
  soltest('parameters'=[alphaguess]):
  soltest(subs(pars,tMem)):
  evalf(subs(alpha=alphaguess,lambdaxw3)):
  err:=evalf(rhs(soltest(subs(pars,tMem))[2])-subs(alpha=alphaguess,
  lambdaxw3));
  while errout>1e-5 do
  soltest('parameters'=[1.000001*alphaguess]):
> derr1:=(evalf(rhs(soltest(subs(pars,tMem))[2])-subs(alpha=1.000001
  *alphaguess,lambdaxw3))):
  soltest('parameters'=[0.999999*alphaguess]):
> derr2:=(evalf(rhs(soltest(subs(pars,tMem))[2])-subs(alpha=0.999999
  *alphaguess,lambdaxw3))):
  derr:=(derr1-derr2)/(2*0.000001*alphaguess);
  alphaguessnew:=alphaguess-err/derr;
  cont:=true:
```

```
while cont=true do
s11:='s11':
try
soltest('parameters'=[alphaguessnew]);
s11:=evalf(rhs(soltest(subs(pars,tMem))[2])-subs(alpha=alphaguessn
ew,lambdaxw3));
catch:
end try:

if is(s11,numeric)=true then

err:=s11:
alphaguess:=alphaguessnew:
cont:=false:
else
alphaguessnew:=(alphaguess+alphaguessnew)/2:
print(alphaguessnew);
cont:=true:
end:
end:


errout:=sqrt(err^2):
print(subs((soltest(subs(pars,tMem)),lambda(z))),evalf(subs(alpha=
alphaguess,lambdaxw3)),errout);

end:
```

$$err := -2.80727014$$

$$14.9354224324942990,\ 14.80714675,\ 0.1282756800$$

$$14.8081421451606518,\ 14.80737296,\ 0.0007691900000$$

$$14.8073771452633238,\ 14.80737485,\ 0.2300000000\ 10^{-5}$$

```
> asol:=alpha=alphaguess;
```

$$asol := \alpha = 0.5103550391$$

Water profile in anode

```
> xwa(z):=subs(asol,xwa(z));
```

$$\mathrm{xwa}(z) := 0.3379040199 - 0.2673127460\ \mathbf{e}^{(0.1241427082\ z)}$$

Water profile in cathode

```
> xwc(z):=subs(asol,xwc(z));
```

$$\mathrm{xwc}(z) := -0.14791362\ \mathbf{e}^{(0.4232592634\ z)} - 1.115715672\ \mathbf{e}^{(-0.4071110271\ z)} + 1.4948756$$

Oxygen profile in cathode

```
> xO(z):=subs(asol,xO(z));
```

$$\mathrm{xO}(z) := 0.6364145560\,\mathbf{e}^{(0.4232592634\,z)} - 0.002127692692\,\mathbf{e}^{(-0.4071110271\,z)} - 0.4948755444$$

Conductivity

```
> sigma30:=0.005139*lambda(z)-0.00326;
```

$$\sigma 30 := 0.005139\,\lambda(z) - 0.00326$$

```
> sigma:=evalf(subs(pars,exp(1268*(1/303-1/(273+Tcell)))*sigma30));
```

$$\sigma := 0.009296230552\,\lambda(z) - 0.005897200156$$

```
> lambda[0]:=rhs(soltest(0)[2]);
```

$$\lambda_0 := 3.25231448106013$$

```
> Nstep:=99:
  h:=subs(pars,tMem/(Nstep+1));
  hC:=subs(pars,tC/(Nstep+1));
  hA:=subs(pars,tA/(Nstep+1));
```

$$h := 0.0001750000000$$

$$hC := 0.0003650000000$$

$$hA := 0.0003650000000$$

```
> xcoord:=seq(n,n=0..Nstep+1):
```

Discretize behavior in the membrane to determine membrane resistance--this is needed due to the piecewise functions involved

```
> for j from 0 to Nstep+1 do
   lambdan[j]:=rhs(soltest(j*h)[2]);
   #print(lambdan[j]);
   if lambdan[j]<14 then
    EqX:=lambdan[j]=subs(pars,0.043+17.81*a-39.85*a^2+36*a^3);
    xwm[j]:=solve(EqX,xw(z))[1];
   else
    EqX:=lambdan[j]=subs(pars,14+1.4*(a-1));
    xwm[j]:=solve(EqX,xw(z));
   end;
  xwcat[j]:=subs(z=(Nstep+1-j)*hC,xwc(z)):
  xwan[j]:=subs(z=(j)*hA,xwa(z)):
  xOx[j]:=subs(z=(Nstep+1-j)*hC,xwc(z)):
  sig[j]:=subs(lambda(z)=lambdan[j],sigma);
  dR[j]:=1/sig[j];
  od:
```

Membrane resistance (Eq. 26)

```
> Rm:=evalf((2*sum(subs(z=2*k*h,dR[2*k]),k=1..(Nstep+1)/2-1)+(4*sum(
  subs(x=(2*k-1)*h,dR[2*k-1]),k=1..((Nstep+1)/2)))+(subs(x=0,dR[0])+
  subs(x=1,dR[Nstep+1]))))*h/(3);
```

$$Rm := 0.4227840513$$

```
> xO3:=evalf(subs(z=tC,pars,xO(z)));
```

$$xO3 := 0.1493509858$$

Equation for voltage of the fuel cell for a given applied current

```
> EqV:=subs(pars,iapp=i0*Pc*xO3*exp(0.5*F/(R*Tcell)*(Voc-Vcell-Rm)))
  ;
```

$$EqV := 0.5 = 0.004480529574\, \mathbf{e}^{(49.11688976 - 72.52766250\, Vcell)}$$

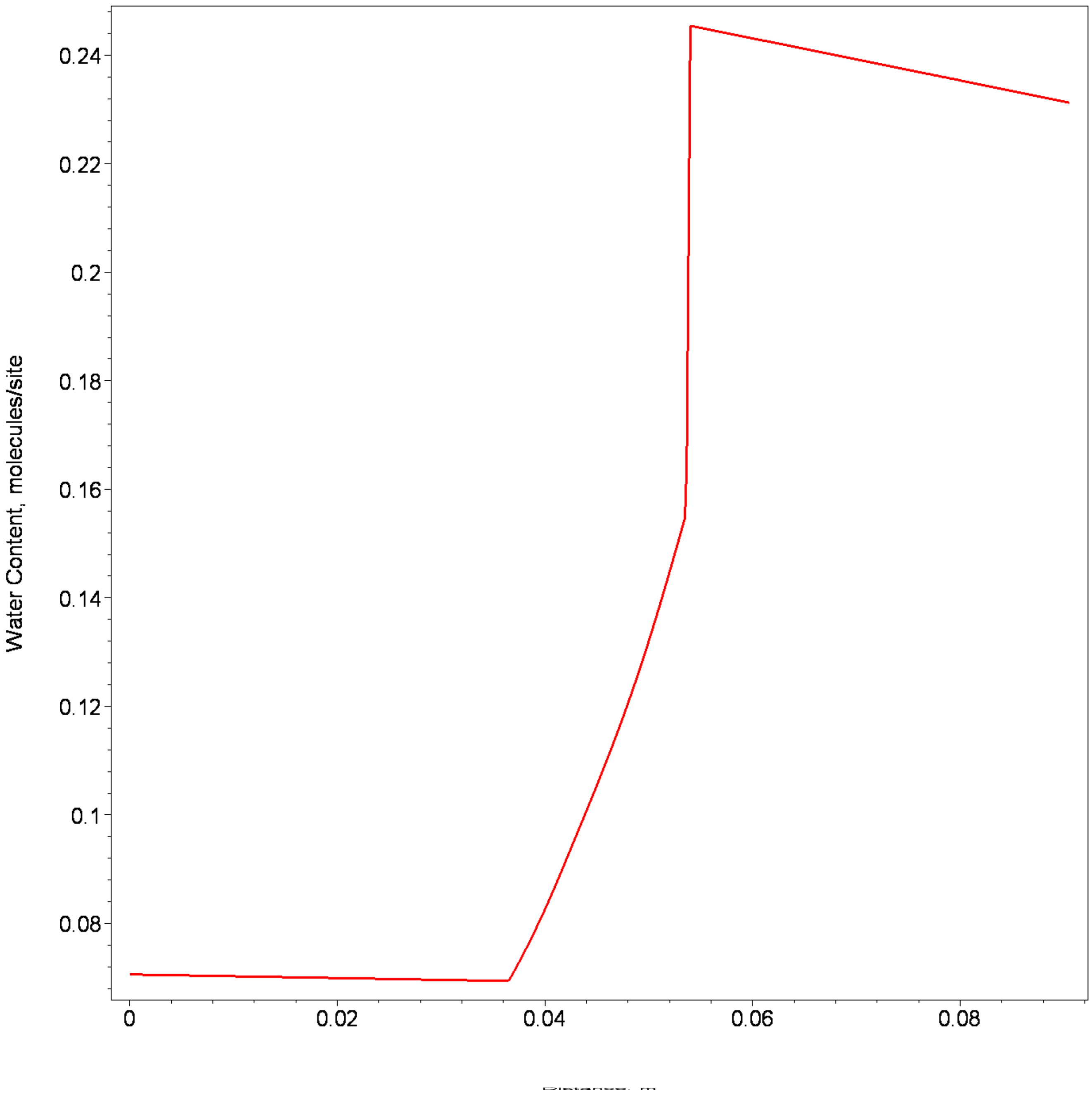
```
> Vcell:=solve(EqV,Vcell);
```

$$Vcell := 0.6122081062$$

```
>
```

Plot the water mole fraction profile across the fuel cell

```
> p1:=plot([xcoord*hA],[seq(xwan[n],n=0..Nstep+1)]):
  p2:=plot([seq(xcoord[n]*h+hA*(Nstep+1),n=1..Nstep+2)],[seq(xwm[n],
  n=0..Nstep+1)]):
  p3:=plot([seq(xcoord[n]*hC+(h+hA)*(Nstep+1),n=1..Nstep+2)],[seq(xw
  cat[n],n=0..Nstep+1)]):
> display({p1,p2,p3},axes=boxed,thickness=3,labels=["Distance, m",
  "Water Content, molecules/site"],labeldirections=[Horizontal,
  Vertical]);
```



```
> alphaguess;
                                         0.5103550391
```

```
> Rm;
                                         0.4227840513
```

If Vcell<0, the fuel cell cannot provide the required current

```
> Vcell;
                                         0.6122081062
```