
NOTE:
Correction: The definition of the Peclet number (Pe) has been modified to:
Pe = (Mw * i0 * L)/(rho * F * D)

Caveats:
This code takes the final form of the model after Landau transformation.
The model solves the PDEs for concentration and potential using numerical method of lines.
A 3 point forward/backward difference is used at the boundaries for first derivatives. Second order
central difference is used in the spatial direction for second derivatives. A cell-centered finite
difference/finite volume method may be more robust for general discretization of PDEs. In general,
when convection (or first derivatives are involved) proper upwind schemes may be needed. N = 10
internal nodes points are used for demonstration. Higher number of node points may be needed for
convergence.
This code is not optimized for speed/efficiency/memory usage.
The same model can be solved more efficiently using collocation in X (Code is available upon request).
Maple can't solve DAEs directly. So, the algebraic equations are differentiated and then solved using
consistent initial condition found from Leastsquare minimization of residues. There are more robust
approaches for solving battery models [M. T. Lawder, V. Ramadesigan, B. Suthar and V. R.
Subramanian, “Extending explicit and linearly implicit ODE solvers for index-1 DAEs”, Computers
Running the same code in C using SUNDIALS IDA will be 2 orders of magnitude faster.
Cycling codes (without documentation) is plotted here.

The equations below are the set of equations after transformation to non dimensional form.

Governing Equations
Equation for potential after Landau Transformation

```maple
eq := (diff(c(Z,tau),Z)*diff(phi(Z,tau),Z)+c(Z,tau)*diff(phi(Z,tau),Z$2))/(1-S(tau))^2 = 0;
eq := \left( \frac{\partial}{\partial Z} c(Z, \tau) \right) \left( \frac{\partial}{\partial Z} \phi(Z, \tau) \right) + c(Z, \tau) \left( \frac{\partial^2}{\partial Z^2} \phi(Z, \tau) \right) = 0
```

(1)

Equation for concentration after Landau transformation

```maple
c_eq := diff(c(Z,tau),tau) = -Pe/tD*CC[0](tau)^.5*(Va(tau)-UU[0](tau))*
       (1-Z)/(1-S(tau))*diff(c(Z,tau),Z)+diff(c(Z,tau),Z$2)/(tD*(1-S(tau))^2);
c_eq := \frac{\partial}{\partial \tau} c(Z, \tau) = -\frac{Pe \sqrt{CC_0(\tau) \left( Va(\tau) - UU_0(\tau) \right) (1 - Z) \left( \frac{\partial}{\partial Z} c(Z, \tau) \right)}}{tD \left( 1 - S(\tau) \right)}
```

(2)

Boundary Conditions
Boundary condition for potential at X=0
\[ bc1 := \frac{\partial \phi(Z, \tau)}{1 - S(\tau)} = -\frac{Da}{\sqrt{c(Z, \tau)}} (V_a(\tau) - \phi(Z, \tau)) \] (3)

Boundary conditions for concentration at X=0
\[ cbc1 := \frac{\partial c(Z, \tau)}{1 - S(\tau)} = -Da \sqrt{c(Z, \tau)} (V_a(\tau) - \phi(Z, \tau)) + c_v P_e c(Z, \tau)^{1.5} (V_a(\tau) - \phi(Z, \tau)) \] (4)

Boundary condition for potential at X=1
\[ bc2 := c(Z, \tau) \frac{\partial \phi(Z, \tau)}{1 - S(\tau)} = -\delta (1 - S(\tau)) \] (5)

Boundary condition for concentration at X=1
\[ cbc2 := \frac{\partial c(Z, \tau)}{1 - S(\tau)} = -\delta (1 - S(\tau)) \] (6)

List of parameters
\[ MV = \frac{M_w}{\rho} \]
\[ pars0 := [iapp=10, i0=20, i0cathode=20, D1=1e-11, F = 96487, R=8.314, T=298, Lx=50e-6, c0=1000, MV=1.2998e-5] \]

Dimensionless groups
\[ pars := [\delta = iapp*Lx*Crate/2/F/D1/c0, \]
\[ tD = \text{abs}(\text{Crate}) * Lx^2 / (3600*D1), \]
\[ Da = i0*Lx / (2. * F * D1 * c0), \]
\[ Pe = MV * i0 * Lx / (D1 * F), \]
\[ phi0 = R * T / F] \]

Discretizing the equations using finite difference
Input number of node points
\[ N1 := 10; \]

Node spacing
\[ h := 1.0 / (N1 + 1); \]
\[ h := 0.09090909091 \]

The variables of c and phi are renamed to be u1 and u2, where u1 = c, u2 = phi

> #u1 = C, u2 = phi #

\[ eq \]

> for i from 1 to \( N_1 \) do:

\[ eq_{11}[i] := (u1[i+1](tau) - u1[i-1](tau))/(2*h)* (u2[i+1](tau) - u2[i-1](tau))/(2*h)/(1+S(tau))^2 + u1[i](tau) * (u2[i-1](tau) - 2*u2[i](tau) + u2[i+1](tau))/h^2/(1+S(tau))^2 = 0; \]

end:

Discretized boundary conditions

bc1

> eq11[0] := (-3*u2[0](tau) + 4*u2[1](tau) - u2[2](tau))/(2*h) = -1*Da*(Va(tau) - u2[0](tau))*(1-S(tau))/u1[0](tau)^.5;

\[ eq_{110} := -16.50000000 \ u_{20}(\tau) + 22.00000000 \ u_{21}(\tau) - 5.500000000 \ u_{22}(\tau) = \]

\[ \frac{Da \ (Va(\tau) - u_{20}(\tau)) \ \left( 1 - S(\tau) \right) \ \sqrt{u_{10}(\tau)} }{u_{11}(\tau)} \]

bc2

> eq11[N1+1] := (-3*u2[N1+1](tau) + 4*u2[N1](tau) - u2[N1-1](tau))/(2*h) = -1*delta*(1-S(tau))/u1[N1+1](tau);

\[ eq_{111} := -16.50000000 \ u_{211}(\tau) + 22.00000000 \ u_{210}(\tau) - 5.500000000 \ u_{20}(\tau) = \]

\[-\ \frac{\delta \ (1 - S(\tau)) }{u_{111}(\tau)} \]

cceq

> cceq;

\[ \frac{\partial}{\partial \tau} \ c(Z, \tau) = - \frac{Pe \ \sqrt{CC_0(\tau)} \ \left( Va(\tau) - UU_0(\tau) \right) \ \left( 1 - Z \right) \ \left( \frac{\partial}{\partial Z} \ c(Z, \tau) \right) }{tD \ (1 - S(\tau))} \]

\[ + \ \frac{\partial^2}{\partial Z^2} \ c(Z, \tau) \]

\[ \ \frac{tD \ (1 - S(\tau))}{} \]

> for i from 1 to \( N_1 \) do:

\[ eq_{21}[i] := \text{diff}(u1[i](tau), tau) = Pe/tD*u1[0](tau)^.5*(u2[0](tau) - Va(tau))*(1-i*h)/(2*h)* (u1[i+1](tau) - u1[i](tau))/(1-S(tau)) + (u1[i-1](tau) - 2*u1[i](tau) + u1[i+1](tau))/h^2/(tD*(1-S(tau))^2); \]

end:

cbc1

> eq21[0] := (-3*u1[0](tau) + 4*u1[1](tau) - u1[2](tau))/(2*h) = (-Da*u1[0](tau)^.5*(Va(tau) - u2[0](tau)) + cv*Pe*u1[0](tau)^1.5*(Va(tau) - u2[0](tau)))*(1-S(tau));

\[ eq_{210} := -16.50000000 \ u_{10}(\tau) + 22.00000000 \ u_{11}(\tau) - 5.500000000 \ u_{12}(\tau) = \]
\[- Da \sqrt{u_1(\tau)} (V_a(\tau) - u_2(\tau)) + cvPe u_1(\tau)^{1.5} (V_a(\tau) - u_2(\tau)) \right) (1 - S(\tau)) \]

\[ cbc2 \]

\[ \text{eq21}[N1+1] := \frac{-3u_1[N1+1](tau) + 4u_1[N1](tau) - u_1[N1-1](tau)}{2h} = -\delta(1 - S(tau)) \]

\[ \text{eq21}_{11} := -16.50000000 u_1_{11}(\tau) + 22.00000000 u_1_{10}(\tau) - 5.500000000 u_1_{0}(\tau) = -\delta(1 - S(\tau)) \]

Reference electrode potential

\[ > \text{Va}(\tau) := 0; \]

\[ V_a(\tau) := 0 \]

Moving boundary velocity

\[ > \text{eqz1} := \frac{d}{dt} S(\tau) = \frac{Pe u_2(\tau) \sqrt{u_1(\tau)}}{tD} \]

\[ eqz1 := \frac{d}{dt} S(\tau) = \frac{Pe u_2(\tau) \sqrt{u_1(\tau)}}{tD} \]

Switch to enable/disable the convective term in the boundary equation cbc1

\[ cv = 1 : \text{enable} / 0 : \text{disable} \]

\[ > cv := 1; \]

\[ cv := 1 \]

Separating the Ordinary Differential Equations (ODEs) and the Algebraic Equations (AEs)

\[ > \text{eqodes} := \text{subs}(\text{pars}, \text{pars0}, [\text{seq}(\text{eq21}[i], i=1..N1), \text{eqz1}]); \]

\[ > \text{eqaes} := \text{subs}(\text{pars}, \text{pars0}, [\text{seq}(\text{eq11}[i], i=0..N1+1), \text{eq21}[0], \text{eq21}[N1+1]]); \]

\[ > \text{eqaes} := \text{map}(\text{rhs} - \text{lhs}, \text{eqaes}); \]

\[ > \text{eqaes2} := \text{subs}(\text{eqodes}, \text{diff}(\text{eqaes}, \tau)); \]

List of variables

\[ > \text{aevars} := [\text{seq}(u_2[i](\tau), i=0..N1+1), u_1[0](\tau), u_1[N1+1](\tau)]; \]

\[ > \text{odevars} := [\text{seq}(u_1[i](\tau), i=1..N1), S(\tau)]; \]

\[ > \text{aesubs} := [\text{seq}(\text{aevars}[i] = ZZ[i], i=1..\text{nops(aevars))}]; \]

\[ > \text{single\_phase} := \text{proc}(\text{cycle, phase}): \]

\[ \text{global sol, dsol}; \]

\[ \text{local i, rate, icodes, eqic, solic, icaes, ics}; \]

\[ \text{if phase = c then rate := 1; } \]

\[ \text{elif phase = d then rate := -1; } \]

\[ \text{fi}; \]
if cycle = 1 and phase = c then icodes:=[seq(u1[i](tau)=1,i=1..N1),S(tau)=.02]:
elif phase = c then icodes:=[seq(odevars[i]=subs(sol[cycle-1,d],odevars[i]),i=1..N1+1)]:
else icodes:=[seq(odevars[i]=subs(sol[cycle,c],odevars[i]),i=1..N1+1)]:
fi:
eqic:=subs(icodes,aesubs,Crate=rate,eqaes):
solic:=Optimization:-LSSolve(eqic):
icaes:=subs(solic[2],aesubs):
ics:=op(subs(tau=0,icodes)),op(subs(tau=0,icaes)):
dsol:=dsolve({op(subs(Crate=rate,eqodes)),op(subs(Crate=rate,eqaes2)),ics},type=numeric,implicit=true,maxfun=0,stiff=true,compile=true):
sol[cycle,phase]:=dsol(1):
end proc:

makeplot:=proc(cycle,phase)
global p1,p2,p3,p4,p5;
local offset;
if phase = d then offset:=1; else offset:=0; fi:
p1[cycle,phase]:=plots:-odeplot(dsol,[(2*(cycle-1)+offset+tau)*3600,u1[0](tau)*1000],0..1,axes=boxed,legend= C_at_MB ,thickness=3,color=red):
p2[cycle,phase]:=plots:-odeplot(dsol,[(2*(cycle-1)+offset+tau)*3600,u1[N1+1](tau)*1000],0..1,axes=boxed,legend= C_at_top,thickness=3,color=blue):
p3[cycle,phase]:=plots:-odeplot(dsol,[(2*(cycle-1)+offset+tau)*3600,u2[0](tau)],0..1,axes=boxed,legend=Phi_at_MB ,thickness=3,color=red):
p4[cycle,phase]:=plots:-odeplot(dsol,[(2*(cycle-1)+offset+tau)*3600,u2[N1+1](tau)],0..1,axes=boxed,legend=Phi_at_top,thickness=3,color=blue):
p5[cycle,phase]:=plots:-odeplot(dsol,[(2*(cycle-1)+offset+tau)*3600,S(tau)*50],0..1,axes=boxed,thickness=3,color=red):
print(Cycle=cycle,Phase=phase,ended);
end proc:

runcycle:=proc(cycle)
global pc,pphi,pmb;
local i;
for i from 1 to cycle do
single_phase(i,c);
      makeplot(i,c);
      single_phase(i,d);
      makeplot(i,d);
      od:

      pc := plots:-display({seq(op([p1[i,c],p1[i,d],p2[i,c],p2[i,d]]), i=1..testcycle)},
                        title="Li Concentration during charging phase",
                        labels=["time [seconds]","Li concentration [mol/m^3]"],
                        labeldirections=[horizontal, vertical]):
      pphi:=plots:-display({seq(op([p3[i,c],p3[i,d],p4[i,c],p4[i,d]]), i=1..testcycle)},
                        title="Electrolyte potential during charging phase",
                        labels=["time [seconds]","Potential [V]"],
                        labeldirections=[horizontal, vertical]):
      pmb := plots:-display({seq(op([p5[i,c],p5[i,d]]), i=1..testcycle)},
                        title="Li Electrode Thickness during charging phase",
                        labels=["time [seconds]","Thickness [1e-6 m]"],
                        labeldirections=[horizontal, vertical]):
      print("simulation complete"):
end proc:

> testcycle:=10;

\texttt{testcycle := 10} \hfill (19)

> runcycle(testcycle);

\texttt{Cycle = 1, Phase = c, ended}
\texttt{Cycle = 1, Phase = d, ended}
\texttt{Cycle = 2, Phase = c, ended}
\texttt{Cycle = 2, Phase = d, ended}
\texttt{Cycle = 3, Phase = c, ended}
\texttt{Cycle = 3, Phase = d, ended}
\texttt{Cycle = 4, Phase = c, ended}
\texttt{Cycle = 4, Phase = d, ended}
\texttt{Cycle = 5, Phase = c, ended}
\texttt{Cycle = 5, Phase = d, ended}
\texttt{Cycle = 6, Phase = c, ended}
\texttt{Cycle = 6, Phase = d, ended}
\texttt{Cycle = 7, Phase = c, ended}
\texttt{Cycle = 7, Phase = d, ended}
\texttt{Cycle = 8, Phase = c, ended}
\texttt{Cycle = 8, Phase = d, ended}
\texttt{Cycle = 9, Phase = c, ended
Cycle = 9, Phase = d, ended
Cycle = 10, Phase = c, ended
Cycle = 10, Phase = d, ended
"simulation complete"

> plots:-display(pc);

> plots:-display(pphi);

> plots:-display(pmb);
Li Electrode Thickness during charging phase