Numerical Implementation of the Doyle-Fuller-Newman (DFN) Model

Prof. Scott Moura | Energy, Controls, and Applications Lab (eCAL) | UC Berkeley

IN PROGRESS | April 24, 2014

In this note we document the numerical implementation of the DFN model.

1 Doyle-Fuller-Newman Model

We consider the Doyle-Fuller-Newman (DFN) model in Fig. 1 to predict the evolution of lithium concentration in the solid $c_s^{\pm}(x,r,t)$, lithium concentration in the electrolyte $c_e(x,t)$, solid electric potential $\phi_s^{\pm}(x,t)$, electrolyte electric potential $\phi_e(x,t)$, ionic current $i_e^{\pm}(x,t)$, molar ion fluxes $j_n^{\pm}(x,t)$, and bulk cell temperature $T(t)$ [1]. The governing equations are given by

$$
\begin{aligned}
\frac{\partial c_s^{\pm}}{\partial t}(x,r,t) &= \frac{1}{r^2}\frac{\partial}{\partial r}\left[D_s^{\pm} r^2 \frac{\partial c_s^{\pm}}{\partial r}(x,r,t)\right], && (1)\\
\frac{\partial c_e}{\partial t}(x,t) &= \frac{\partial}{\partial x}\left[D_e \frac{\partial c_e}{\partial x}(x,t) + \frac{1-t_c^0}{\varepsilon_e F} i_e^{\pm}(x,t)\right], && (2)\\
0 &= \frac{\partial \phi_s^{\pm}}{\partial x}(x,t) - \frac{i_e^{\pm}(x,t) - I(t)}{\sigma^{\pm}}, && (3)\\
0 &= \frac{\partial \phi_e}{\partial x}(x,t) + \frac{i_e^{\pm}(x,t)}{\kappa} - \frac{2RT}{F}(1-t_c^0)\times\left(1 + \frac{d\ln f_{c/a}}{d\ln c_e}(x,t)\right)\frac{\partial \ln c_e}{\partial x}(x,t), && (4)\\
0 &= \frac{\partial i_e^{\pm}}{\partial x}(x,t) - a_s F j_n^{\pm}(x,t), && (5)\\
0 &= \frac{1}{F} i_0^{\pm}(x,t)\left[e^{\frac{\alpha_a F}{RT}\eta^{\pm}(x,t)} - e^{-\frac{\alpha_c F}{RT}\eta^{\pm}(x,t)}\right] - j_n^{\pm}(x,t), && (6)\\
\rho^{\mathrm{avg}} c_P \frac{dT}{dt}(t) &= h_{\mathrm{cell}}\left[T_{\mathrm{amb}}(t) - T(t)\right] + I(t)V(t) - \int_{0^-}^{0^+} a_s F j_n(x,t)\Delta T(x,t)\,dx, && (7)
\end{aligned}
$$

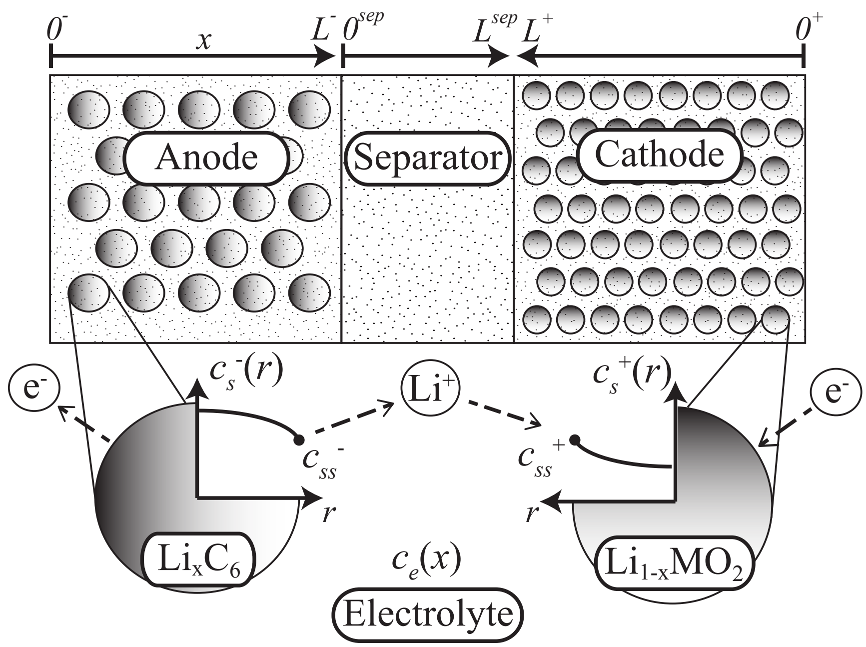

Figure 1: Schematic of the Doyle-Fuller-Newman model [1]. The model considers two phases: the solid and electrolyte. In the solid, states evolve in the x and r dimensions. In the electrolyte, states evolve in the x dimension only. The cell is divided into three regions: anode, separator, and cathode.

where $D_e, \kappa, f_{c/a}$ are functions of $c_e(x,t)$ and

$$
\begin{aligned}
i_0^{\pm}(x,t) &= k^{\pm}\left[c_{ss}^{\pm}(x,t)\right]^{\alpha_c}\left[c_e(x,t)\left(c_{s,\max}^{\pm}-c_{ss}^{\pm}(x,t)\right)\right]^{\alpha_a}, && (8)\\
\eta^{\pm}(x,t) &= \phi_s^{\pm}(x,t)-\phi_e(x,t)-U^{\pm}(c_{ss}^{\pm}(x,t))-FR_f^{\pm}j_n^{\pm}(x,t), && (9)\\
c_{ss}^{\pm}(x,t) &= c_s^{\pm}(x,R_s^{\pm},t), && (10)\\
\Delta T(x,t) &= U^{\pm}(\overline{c}_s^{\pm}(x,t))-T(t)\frac{\partial U^{\pm}}{\partial T}(\overline{c}_s^{\pm}(x,t)), && (11)\\
\overline{c}_s^{\pm}(x,t) &= \frac{3}{(R_s^{\pm})^3}\int_0^{R_s^{\pm}} r^2 c_s^{\pm}(x,r,t)dr && (12)
\end{aligned}
$$

Along with these equations are corresponding boundary and initial conditions. The boundary conditions for the solid-phase diffusion PDE (1) are

$$
\begin{aligned}
\frac{\partial c_s^{\pm}}{\partial r}(x,0,t) &= 0, && (13)\\
\frac{\partial c_s^{\pm}}{\partial r}(x,R_s^{\pm},t) &= -\frac{1}{D_s^{\pm}}j_n^{\pm}. && (14)
\end{aligned}
$$

The boundary conditions for the electrolyte-phase diffusion PDE (2) are given by

$$\frac{\partial c_e}{\partial x}(0^-,t) = \frac{\partial c_e}{\partial x}(0^+,t) = 0, \tag{15}$$

$$\varepsilon_e^- D_e^-(L^-)\frac{\partial c_e}{\partial x}(L^-,t) = \varepsilon_e^{\text{sep}} D_e^{\text{sep}}(0^{\text{sep}})\frac{\partial c_e}{\partial x}(0^{\text{sep}},t), \tag{16}$$

$$\varepsilon_e^{\text{sep}} D_e^{\text{sep}}(L^{\text{sep}})\frac{\partial c_e}{\partial x}(L^{\text{sep}},t) = \varepsilon_e^+ D_e^+(L^+)\frac{\partial c_e}{\partial x}(L^+,t), \tag{17}$$

$$c_e(L^-,t) = c_e(0^{\text{sep}},t), \tag{18}$$

$$c_e(L^{\text{sep}},t) = c_e(0^+,t). \tag{19}$$

The boundary conditions for the solid-phase potential ODE (3) are given by

$$\frac{\partial \phi_s^-}{\partial x}(L^-,t) = \frac{\partial \phi_s^+}{\partial x}(L^+,t) = 0. \tag{20}$$

The boundary conditions for the electrolyte-phase potential ODE (4) are given by

$$\phi_e(0^-,t) = 0, \tag{21}$$

$$\phi_e(L^-,t) = \phi_e(0^{\text{sep}},t), \tag{22}$$

$$\phi_e(L^{\text{sep}},t) = \phi_e(L^+,t). \tag{23}$$

The boundary conditions for the ionic current ODE (5) are given by

$$i_e^-(0^-,t) = i_e^+(0^+,t) = 0 \tag{24}$$

and also note that $i_e(x,t) = I(t)$ for $x \in [0^{\text{sep}}, L^{\text{sep}}]$.

The input to the model is the applied current density $I(t)$, and the output is the voltage measured across the current collectors

$$V(t) = \phi_s^+(0^+,t) - \phi_s^-(0^-,t) \tag{25}$$

Further details, including notation definitions, can be found in [1, 2].

2 Time-stepping

Ultimately, the equations are discretized to produce a DAE in the following format:

$$\dot{x} = f(x,z,u), \tag{26}$$

$$0 = g(x,z,u) \tag{27}$$

with initial conditions $x(0), z(0)$ that are consistent. That is, they verify (27). The time-stepping is done by solving the nonlinear equation

$$0 = F(x(t+\Delta t), z(t+\Delta t)), \tag{28}$$

$$0 = \begin{bmatrix} x(t) - x(t+\Delta t) + \frac{1}{2}\Delta t\left[f(x(t),z(t),u(t)) + f(x(t+\Delta t),z(t+\Delta t),u(t+\Delta t))\right] \\ g\left(x(t+\Delta t),z(t+\Delta t),u(t+\Delta t)\right) \end{bmatrix} \tag{29}$$

for $x(t+\Delta t), z(t+\Delta t)$. The function `cfn_dfn.m` returns the solution $(x(t+\Delta t), z(t+\Delta t))$ of (28)-(29), given $x(t), z(t), u(t), u(t+\Delta t)$. Note that we solve (28)-(29) using Newton's method, meaning analytic Jacobians of $F(\cdot,\cdot)$ are required w.r.t. x, z.

$$J = \begin{bmatrix} F_x^1 & F_z^1 \\ F_x^2 & F_z^2 \end{bmatrix} \tag{30}$$

$$= \begin{bmatrix} -I + \frac{1}{2}\Delta t \cdot \frac{\partial f}{\partial x}(x(t+\Delta t), z(t+\Delta t), u(t+\Delta t)) & \frac{1}{2}\Delta t \cdot \frac{\partial f}{\partial z}(x(t+\Delta t), z(t+\Delta t), u(t+\Delta t)) \\ \frac{\partial g}{\partial x}(x(t+\Delta t), z(t+\Delta t), u(t+\Delta t)) & \frac{\partial g}{\partial z}(x(t+\Delta t), z(t+\Delta t), u(t+\Delta t)) \end{bmatrix} \tag{31}$$

3 DAEs

To perform the time-stepping in the previous section, we must compute functions $f(x, z, u)$ and $g(x, z, u)$. These functions, which represent the RHS of (26)-(27), are calculated by the Matlab function `dae_dfn.m`, given the inputs x, z, u. The role of variables x, z, u are played by the DFN variables shown in Table 1.

Table 1: DAE notation for DFN states in Matlab Code

DAE Variable	**DFN Variable**
x	$c_s^-, c_s^+, c_e = [c_e^-, c_e^{sep}, c_e^+], T$
z	$\phi_s^-, \phi_s^+, i_e^-, i_e^+, \phi_e = [\phi_e^-, \phi_e^{sep}, \phi_e^+], j_n^-, j_p^+$
u	I

In the subsequent sections, we go through each DFN variable listed in Table 1 and document its numerical implementation.

4 Solid Concentration, c_s^-, c_s^+

[DONE] The PDEs (1) governing Fickian diffusion in the solid phase are implemented using third order Padé approximations of the two transfer functions from $j_n^\pm$ to $c_{ss}^\pm$ and $\overline{c}_s^\pm$.

$$\frac{C_{ss}^\pm(s)}{J_n^\pm(s)} = \frac{-\frac{21}{R_s^\pm}s^2 - \frac{1260 D_s^\pm}{(R_s^\pm)^3}s - \frac{10395(D_s^\pm)^2}{(R_s^\pm)^4}}{s^3 + \frac{189 D_s^\pm}{(R_s^\pm)^2}s^2 + \frac{3465(D_s^\pm)^2}{(R_s^\pm)^4}s}, \tag{32}$$

$$\frac{\overline{C}_s^\pm(s)}{J_n^\pm(s)} = \frac{-3R_s^\pm}{s}. \tag{33}$$

These transfer functions are converted into controllable canonical state-space form , thus producing the subsystem:

$$\frac{d}{dt}\begin{bmatrix} c_{s1}^{\pm}(t) \\ c_{s2}^{\pm}(t) \\ c_{s3}^{\pm}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -\frac{3465(D_s^{\pm})^2}{(R_s^{\pm})^4} & -\frac{189 D_s^{\pm}}{(R_s^{\pm})^2} \end{bmatrix} \begin{bmatrix} c_{s1}^{\pm}(t) \\ c_{s2}^{\pm}(t) \\ c_{s3}^{\pm}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} j_n^{\pm}(t) \tag{34}$$

$$\begin{bmatrix} c_{ss}^{\pm}(t) \\ \overline{c}_s^{\pm}(t) \end{bmatrix} = \begin{bmatrix} -\frac{10395(D_s^{\pm})^2}{(R_s^{\pm})^5} & -\frac{1260 D_s^{\pm}}{(R_s^{\pm})^3} & -\frac{21}{R_s^{\pm}} \\ -\frac{3}{R_s^{\pm}} \cdot \frac{3465(D_s^{\pm})^2}{(R_s^{\pm})^4} & -\frac{3}{R_s^{\pm}} \cdot \frac{189 D_s^{\pm}}{(R_s^{\pm})^2} & -\frac{3}{R_s^{\pm}} \cdot 1 \end{bmatrix} \begin{bmatrix} c_{s1}^{\pm}(t) \\ c_{s2}^{\pm}(t) \\ c_{s3}^{\pm}(t) \end{bmatrix} \tag{35}$$

for each discrete point in x.

5 Electrolyte Concentration, c_e

[DONE] The electrolyte concentration is implemented using the central difference method, which ultimately produces the matrix differential equation:

$$\frac{d}{dt} c_e(t) = A_{ce}(c_{e,x}) \cdot c_e(t) + B_{ce}(c_{e,x}) \cdot i_{e,x}(t) \tag{36}$$

where $c_e, i_{e,x}$ are vectors whose elements represent discrete points along the x-dimension of the DFN model. In particular $i_{e,x}$ and $c_{e,x}$ represent the entire electrolyte current and concentration, respectively, across the entire battery, including boundary values,

$$i_{e,x}(t) = \left[0,\ i_e^{-}(x,t),\ I(x,t),\ i_e^{+}(x,t),\ 0\right]^T, \tag{37}$$

$$c_{e,x}(t) = \left[c_{e,bc,1}(t),\ c_e^{-}(x,t), c_{e,bc,2}(t),\ c_e^{sep}(x,t), c_{e,bc,3}(t),\ c_e^{+}(x,t),\ c_{e,bc,4}(t)\right]^T, \tag{38}$$

$$c_{e,bc}(t) = C_{ce}\ c_e(t) \tag{39}$$

Note that the system matrices (A_{ce}, B_{ce}) are also state-varying, but C_{ce} is not. These state matrices are computed online by Matlab function `c_e_mats.m`. Matrix C_{ce} can be computed offline. The state matrices are computed by

$$A_{ce} = (M1) - (M2)(N2)^{-1}(N1), \tag{40}$$

$$B_{ce} = (M3), \tag{41}$$

$$C_{ce} = -(N2)^{-1}(N1) \tag{42}$$

The first term on the RHS of PDE (2) is implemented by

$$(M1) = \text{BlkDiag}\left((M1n), (M1s), (M1p)\right), \tag{43}$$

$$(M2) = \begin{bmatrix} (M2n_{col1}) & (M2n_{col2}) & 0 & 0 \\ 0 & (M2s_{col1}) & (M2s_{col2}) & 0 \\ 0 & 0 & (M2p_{col1}) & (M2p_{col2}) \end{bmatrix} \tag{44}$$

and

$$(M1n) =$$
$$\alpha^{-} \cdot \begin{bmatrix} -(D_{e,0}+D_{e,n,2}) & D_{e,n,2} & & \\ D_{e,n,1} & -(D_{e,n,1}+D_{e,n,3}) & D_{e,n,3} & \\ \ddots & \ddots & \ddots & \\ & D_{e,n,i-1} & -(D_{e,n,i-1}+D_{e,n,i+1}) & D_{e,n,i+1} \\ & \ddots & \ddots & \ddots \\ & & D_{e,n,Nxn-2} & -(D_{e,n,Nxn-2}+D_{e,ns}) \end{bmatrix} \tag{45}$$

$$(M1s) =$$
$$\alpha^{sep} \cdot \begin{bmatrix} -(D_{e,ns}+D_{e,s,2}) & D_{e,s,2} & & \\ D_{e,s,1} & -(D_{e,s,1}+D_{e,s,3}) & D_{e,s,3} & \\ \ddots & \ddots & \ddots & \\ & D_{e,s,i-1} & -(D_{e,s,i-1}+D_{e,s,i+1}) & D_{e,s,i+1} \\ & \ddots & \ddots & \ddots \\ & & D_{e,s,Nxs-2} & -(D_{e,s,Nxs-2}+D_{e,np}) \end{bmatrix} \tag{46}$$

$$(M1p) =$$
$$\alpha^{+} \cdot \begin{bmatrix} -(D_{e,sp}+D_{e,p,2}) & D_{e,p,2} & & \\ D_{e,p,1} & -(D_{e,p,1}+D_{e,p,3}) & D_{e,p,3} & \\ \ddots & \ddots & \ddots & \\ & D_{e,p,i-1} & -(D_{e,p,i-1}+D_{e,p,i+1}) & D_{e,p,i+1} \\ & \ddots & \ddots & \ddots \\ & & D_{e,p,Nxp-2} & -(D_{e,p,Nxp-2}+D_{e,N}) \end{bmatrix} \tag{47}$$

$$(M2n) = \alpha^{-} \begin{bmatrix} D_{e,0} & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & D_{e,ns} \end{bmatrix},\ (M2s) = \alpha^{sep} \begin{bmatrix} D_{e,ns} & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & D_{e,sp} \end{bmatrix},\ (M2p) = \alpha^{+} \begin{bmatrix} D_{e,sp} & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & D_{e,N} \end{bmatrix} \tag{48}$$

$$
(M3) = \left[\begin{array}{ccc|c|ccc|c|ccc}
-\beta^- & 0 & \beta^- & & & & & & & & \\
 & \ddots & \ddots & \ddots & & & & & & & \\
 & & -\beta^- & 0 & \beta^- & & & & & & \\
\hline
 & & & -\beta^{sep} & 0 & \beta^{sep} & & & & & \\
 & & & & \ddots & \ddots & \ddots & & & & \\
 & & & & & -\beta^{sep} & 0 & \beta^{sep} & & & \\
\hline
 & & & & & & & -\beta^+ & 0 & \beta^+ & \\
 & & & & & & & & \ddots & \ddots & \ddots \\
 & & & & & & & & -\beta^+ & 0 & \beta^+
\end{array}\right] \tag{49}
$$

and

$$
\alpha^j = \frac{1}{(L^j \Delta x^j)^2}, \qquad \beta^j = \frac{1-t_c^0}{2\varepsilon_e^j F L^j \Delta x^j}, \tag{50}
$$

$$
D_e(c_{e,x}(x,t)) = [D_{e,0} \mid D_{e,n}(x) \mid D_{e,ns} \mid D_{e,s}(x) \mid D_{e,sp} \mid D_{e,p}(x) \mid D_{e,N}] \tag{51}
$$

The boundary conditions (15)-(19) are implemented as

$$
(N1) = \left[\begin{array}{cccc|cccc|cccc}
\frac{1}{L^-\Delta x^-} & 0 & \cdots & 0 & 0 & \cdots & & 0 & 0 & \cdots & & 0 \\
0 & \cdots & 0 & \frac{D_{e,ns}}{L^-\Delta x^-} & \frac{D_{e,ns}}{L^{sep}\Delta x^{sep}} & 0 & \cdots & 0 & 0 & \cdots & & 0 \\
0 & 0 & \cdots & 0 & 0 & \cdots & 0 & \frac{D_{e,sp}}{L^{sep}\Delta x^{sep}} & \frac{D_{e,sp}}{L^+\Delta x^+} & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 & 0 & \cdots & & 0 & 0 & \cdots & 0 & \frac{-1}{L^+\Delta x^+}
\end{array}\right], \tag{52}
$$

$$
(N2) = \begin{bmatrix}
\frac{-1}{L^-\Delta x^-} & 0 & 0 & 0 \\
0 & -\frac{D_{e,ns}}{L^-\Delta x^-} - \frac{D_{e,ns}}{L^{sep}\Delta x^{sep}} & 0 & 0 \\
0 & 0 & -\frac{D_{e,sp}}{L^{sep}\Delta x^{sep}} - \frac{D_{e,sp}}{L^+\Delta x^+} & 0 \\
0 & 0 & 0 & \frac{1}{L^+\Delta x^+}
\end{bmatrix} \tag{53}
$$

6 Temperature, T

[DONE] Temperature is scalar, so the ODE is directly implemented as:

$$
\rho^{\text{avg}} c_P \frac{dT}{dt}(t) = h_{\text{cell}} \left[T_{\text{amb}}(t) - T(t)\right] + I(t)V(t) - \int_{0^-}^{0^+} a_s F j_n(x,t) \Delta T(x,t) dx, \tag{54}
$$

$$
\Delta T(x,t) = U^\pm(\overline{c}_s^\pm(x,t)) - T(t) \frac{\partial U^\pm}{\partial T}(\overline{c}_s^\pm(x,t)), \tag{55}
$$

$$
\overline{c}_s^\pm(x,t) = \frac{3}{(R_s^\pm)^3} \int_0^{R_s^\pm} r^2 c_s^\pm(x,r,t) dr \tag{56}
$$

7 Solid Potential, ϕ_s^-, ϕ_s^+

[DONE] The solid potential is implemented using the central difference method, which ultimately produces the matrix equation:

$$\begin{aligned}
\frac{d}{dt}\phi_s^-(t) &= F_{psn}^1\ \phi_s^-(t) + F_{psn}^2\ i_{e,aug}^-(t) + G_{psn}\ I(t) && (57)\\
\frac{d}{dt}\phi_s^+(t) &= F_{psp}^1\ \phi_s^+(t) + F_{psp}^2\ i_{e,aug}^+(t) + G_{psp}\ I(t). && (58)
\end{aligned}$$

where $i_{e,aug}^{\pm}$ are

$$i_{e,aug}^-(t) = \begin{bmatrix} 0 \\ i_e^-(x,t) \\ I(t) \end{bmatrix}, \qquad i_{e,aug}^+(t) = \begin{bmatrix} I(t) \\ i_e^+(x,t) \\ 0 \end{bmatrix} \tag{59}$$

This section also computes the terminal voltage $V(t)$ from (25) using matrix equations

$$\begin{aligned}
\phi_{s,bc}^-(t) &= C_{psn}\ \phi_s^-(t) + D_{psn}\ I(t), && (60)\\
\phi_{s,bc}^+(t) &= C_{psp}\ \phi_s^+(t) + D_{psp}\ I(t), && (61)\\
V(t) &= \phi_{s,bc,2}^+(t) - \phi_{s,bc,1}^-(t) && (62)
\end{aligned}$$

where the following matrices are computed a priori by Matlab function `phi_s_mats.m`

$$\begin{aligned}
(F1n) &= (M1n) - (M2n)(N2n)^{-1}(N1n), && (63)\\
(F2n) &= (M3n), && (64)\\
(Gn) &= 1 - (M2n)(N2n)^{-1}(N4n), && (65)\\
(F1p) &= (M1p) - (M2p)(N2p)^{-1}(N1p), && (66)\\
(F2p) &= (M3p), && (67)\\
(Gp) &= 1 - (M2p)(N2p)^{-1}(N4p), && (68)\\
(Cn) &= -(N2n)^{-1}(N1n), && (69)\\
(Dn) &= -(N2n)^{-1}(N4n), && (70)\\
(Cp) &= -(N2p)^{-1}(N1p), && (71)\\
(Dp) &= -(N2p)^{-1}(N4p), && (72)
\end{aligned}$$

where the (Mij) and $N(ij)$ matrices result from central difference approximations of the ODE in space (3) and boundary conditions (20).

$$(M1j) = \begin{bmatrix} 0 & \alpha_j & 0 & \dots & 0 \\ -\alpha_j & 0 & \alpha_j & \dots & 0 \\ 0 & -\alpha_j & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \\ 0 & 0 & \cdots & \cdots & \alpha_j \\ 0 & 0 & \cdots & -\alpha_j & 0 \end{bmatrix}, \quad (M2j) = \begin{bmatrix} -\alpha_j & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & \alpha_j \end{bmatrix}, \tag{73}$$

$$(M3j) = \begin{bmatrix} 0 & -1 & 0 & \dots & & 0 \\ 0 & 0 & -1 & \dots & & 0 \\ \vdots & \vdots & \vdots & & & \vdots \\ 0 & 0 & \cdots & -1 & 0 & 0 \\ 0 & 0 & \cdots & 0 & -1 & 0 \end{bmatrix}, \tag{74}$$

$$(N1j) = \begin{bmatrix} 2\alpha_j & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & -2\alpha_j \end{bmatrix}, \quad (N2j) = \begin{bmatrix} -2\alpha_j & 0 \\ 0 & 2\alpha_j \end{bmatrix}, \tag{75}$$

$$(N4n) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad (N4n) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{76}$$

for $j \in \{n, p\}$, $\alpha_j = \sigma^j/(2L^j \Delta x^j)$.

8 Electrolyte Current, i_e^-, i_e^+

[DONE] The electrolyte current is implemented using the central difference method, which ultimately produces the matrix equation:

$$\frac{d}{dt} i_e^-(t) = F_{ien}^{1-}\, i_e^-(t) + F_{ien}^{2-}\, j_n^-(t) + F_{ien}^{3-}\, I(t) \tag{77}$$

$$\frac{d}{dt} i_e^+(t) = F_{iep}^{1+}\, i_e^+(t) + F_{iep}^{2+}\, j_n^+(t) + F_{iep}^{3+}\, I(t) \tag{78}$$

where the following matrices are computed a priori by Matlab function `i_e_mats.m`

$$F_{ien}^{1-} = (M1n) - (M2n)(N2n)^{-1}(N1n), \tag{79}$$

$$F_{ien}^{2-} = (M3n) - (M2n)(N2n)^{-1}(N3n), \tag{80}$$

$$F_{ien}^{3-} = (M2n)(N2n)^{-1}(N4n), \tag{81}$$

$$F_{iep}^{1+} = (M1p) - (M2p)(N2p)^{-1}(N1p), \tag{82}$$

$$F_{iep}^{2+} = (M3p) - (M2p)(N2p)^{-1}(N3p), \tag{83}$$

$$F_{iep}^{3+} = (M2p)(N2p)^{-1}(N4p) \tag{84}$$

where the (Mij) and $N(ij)$ matrices result from central difference approximations of the ODE in space (5) and boundary conditions (24).

$$(M1j) = \begin{bmatrix} 0 & \alpha_j & 0 & \dots & 0 \\ -\alpha_j & 0 & \alpha_j & \dots & 0 \\ 0 & -\alpha_j & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \\ 0 & 0 & \cdots & \cdots & \alpha_j \\ 0 & 0 & \cdots & -\alpha_j & 0 \end{bmatrix}, \quad (M2j) = \begin{bmatrix} -\alpha_j & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & \alpha_j \end{bmatrix}, \quad (M3j) = -\beta_j \mathbb{I}, \tag{85}$$

$$(N1j) = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}, \qquad (N2j) = \mathbb{I}, \qquad (N3j) = (N1j), \tag{86}$$

$$(N4n) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad (N4n) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{87}$$

for $j \in \{n, p\}$, $\alpha_j = (2L^j \Delta x^j)^{-1}$, $\beta_j = a_s^j F$.

9 Electrolyte Potential, ϕ_e

The electrolyte potential is implemented using the central difference method, which ultimately produces the matrix equation:

$$\frac{d}{dt}\phi_e^-(t) = F_{pe}^1(c_{e,x}) \cdot \phi_e(t) + F_{pe}^2(c_{e,x}) \cdot i_{e,x}(t) + F_{pe}^3(c_{e,x}) \cdot \ln(c_{e,x}(t)) \tag{88}$$

where vectors $i_{e,x}, c_{e,x}$ are given by (37),(38). Note that the system matrices $F_{pe}^1, F_{pe}^2, F_{pe}^3$ are state-varying. These state matrices are computed online by Matlab function `phi_e_mats.m` as follows

$$\begin{aligned} F_{pe}^1 &= (M1) - (M2)(N2)^{-1}(N1), & (89) \\ F_{pe}^2 &= (M3) - (M2)(N2)^{-1}(N3), & (90) \\ F_{pe}^3 &= (M4) - (M2)(N2)^{-1}(N4) & (91) \end{aligned}$$

DO BOUNDARY CONDITIONS NEXT
and

$$\begin{aligned} \alpha^j &= \frac{1}{2L^j \Delta x^j}, \qquad \beta^j = \frac{RT}{\alpha F}(1 - t_c^0)\frac{1+0}{2L^j \Delta x^j}, \qquad \gamma = \frac{RT}{\alpha F}(1 - t_c^0)(1+0) & (92) \\ \kappa(c_{e,x}(x,t)) &= [\kappa_0 \mid \kappa_n(x) \mid \kappa_{ns} \mid \kappa_s(x) \mid \kappa_{sp} \mid \kappa_p(x) \mid \kappa_N] & (93) \end{aligned}$$

where the 0 is β^j and γ arises when $\frac{d \ln f_{c/a}}{d \ln c_e}(x,t)$ in (3) is zero.

10 Molar ion fluxes, i.e. Butler-Volmer Current, j_n^-, j_n^+

[DONE] Since the Butler-Volmer equation (6) is algebraic, and we always assume $\alpha_a = \alpha_c = 0.5 = \alpha$, it is trivially implemented as:

$$\frac{d}{dt} j_n^-(t) = \frac{2}{F} i_0^-(t) \sinh\left[\frac{\alpha F}{RT}\eta^-(t)\right] - j_n^-(t), \tag{94}$$

$$\frac{d}{dt} j_n^+(t) = \frac{2}{F} i_0^+(t) \sinh\left[\frac{\alpha F}{RT}\eta^+(t)\right] - j_n^+(t) \tag{95}$$

where

$$i_0^\pm(t) = k^\pm \left[c_{ss}^\pm(t) c_e(t) \left(c_{s,\max}^\pm - c_{ss}^\pm(t)\right)\right]^\alpha, \tag{96}$$

$$\eta^\pm(t) = \phi_s^\pm(t) - \phi_e(t) - U^\pm(c_{ss}^\pm(t)) - FR_f^\pm j_n^\pm(t) \tag{97}$$

for each discrete point in x, in the electrodes only. Note that $\frac{d}{dt} j_n^\pm(t)$ is a dummy variable used to save the corresponding element of vector $g(x, z, t)$.

References

[1] K. Thomas, J. Newman, and R. Darling, *Advances in Lithium-Ion Batteries.* New York, NY USA: Kluwer Academic/Plenum Publishers, 2002, ch. 12: Mathematical modeling of lithium batteries, pp. 345–392.

[2] N. A. Chaturvedi, R. Klein, J. Christensen, J. Ahmed, and A. Kojic, “Algorithms for advanced battery-management systems,” *IEEE Control Systems Magazine*, vol. 30, no. 3, pp. 49 – 68, 2010.