

1. Introduction

Lithium-ion batteries, featuring graphite electrodes, have dominated energy storage due to their higher efficiency, power density and calendar life compared to other rechargeable batteries, like nickel-cadmium or lead-acid. However, meeting demands for fast-charging and high capacity poses challenges due to the presence of complex degradation mechanisms. Battery degradation modeling due to lithium plating helps predict capacity loss, cycle life and predict the conditions to reduce electrochemical degradation of batteries. In this project, the SP model for a Li-ion battery has been simulated.

2. Mathematical Model

In the SP model, electrodes are assumed to consist of spherical intercalation particles with identical size. The key assumption in the single-particle model is that the current distribution is taken to be uniform along the thickness of the porous electrode.

The mass balance of lithium ions in an intercalation particle of the electrode is given by Fick's Diffusion Law:

$$\frac{\partial c_{s,j}}{\partial t} = \frac{D_{s,j}}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial c_{s,j}}{\partial r} \right)$$

with the boundary conditions:

$$c(r = 0) = c_0$$

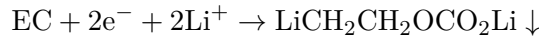
$$D \frac{\partial c}{\partial r} \Big|_{r=R} = -j$$

The electrochemical reactions for lithium-ion intercalation/deintercalation at the solid/ solution interface can be expressed as



where θ_s represents a vacant host on the solid particle surface.

Anodic degradation generally occurs with the formation and growth of SEI layer that consumes active lithium from the electrolyte. The reduction of ethylene carbonate demonstrates the parasitic reaction occurring at the negative electrode/electrolyte interface:



However, under conditions of fast charging where the anodic potential becomes negative, lithium metal starts to plate on the graphite, along with the SEI film. At the end of charging, the anodic potential becomes positive and stripping action starts.

Thus, in order to model the exchange current density at the solid/solution interface superposition of flux contributions from electrode intercalation i_{el} , SEI growth i_{SEI} , lithium plating i_{pl} and stripping i_{st} were taken.

$$i_{tot} = i_{el} + i_{SEI} + i_{pl} + i_{st}$$

These currents are modelled by the following equations:

- Butler Volmer Formulation

$$i_{el} = 2Fk_{el} \sqrt{c_{s,surf}(c_{s,max} - c_{s,surf})c_e} \sinh \left(\frac{\alpha F}{RT} (\eta - U_{OCP} - R_{film} i_{tot}) \right)$$

The plating initiates only when the anode potential becomes negative.

$$i_{pl} = \min \left(0, 2Fk_{pl}\sqrt{c_e} \sinh \left(\frac{\alpha_{pl}F}{RT} (\eta - U_{pl} - R_{film}i_{tot}) \right) \right); \eta < 0$$

The stripping current initiates when $\eta > 0$ and some amount of lithium metal $m_{pl} = 4\pi R_n^2 \delta_{pl} \rho_{pl}$ has deposited in the form of plated film on the electrode particle.

$$i_{st} = \max \left(0, 2Fk_{pl}\sqrt{c_e} \sinh \left(\frac{\alpha_{pl}F}{RT} (\eta - R_{film}i_{tot}) \right) \frac{m_{pl}}{\max(m_{pl})} \right); \eta > 0, \max(m_{pl}) > 0$$

- Tafel kinetics

$$i_{SEI} = -Fk_{SEI}c_{EC,surf} \exp \left(-\frac{\alpha_{SEI}F}{RT} (\eta - U_{SEI} - R_{film}i_{tot}) \right)$$

Fick's first law of diffusion is used to model the transport of ethylene carbonate concentration from the film to the bulk across the SEI plus plated lithium layer.

$$-D_{EC} \frac{\partial c_{EC,surf}}{\partial \delta_{film}} = \frac{i_{SEI}}{F}$$

The growth in thickness of the plated film during plating (reduction during stripping) is approximated as:

$$\frac{\partial \delta_{Li}}{\partial t} = -\frac{i_{pl}(\text{or } i_{st})M_{pl}}{F\rho_{pl}}$$

Similarly for change in thickness in SEI film,

$$\frac{\partial \delta_{SEI}}{\partial t} = -\frac{i_{SEI}M_{SEI}}{F\rho_{SEI}}$$

Then,

$$\delta_{film} = \delta_{SEI} + \delta_{Li}$$

The film resistance is modelled as:

$$\mathbb{R}_{film} = \frac{\delta_{film}}{\kappa_{SEI}}$$

Finally, the capacity loss at the end of a charge cycle is formulated as follows:

$$Q_{lost} = \int_0^{t_r} (i_{st} + i_{cr} + i_{SEI}) dt$$

3. Results and Discussion

The equations given in the above section were numerically solved using an Euler Implicit scheme in MATLAB R2022b. The model is simulated with the particle geometry and properties given in reference [2],[3] with a computational time step of 0.05 s. The open circuit potentials have been interpolated from experimentally available data.

3.A. Half cycle model of discharge of battery

Initially only the discharge half cycle of the battery was simulated. Only the intercalation current was considered following Butler Volmer Kinetics. The potential at the anode was found to increase as a function of time. It took ~2100 s for the battery to fully charge for a current density of 0.5 A/m².

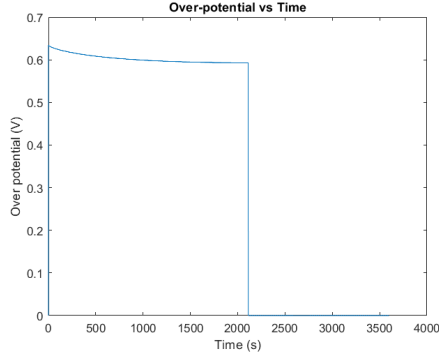


Figure 1: Overpotential vs Time

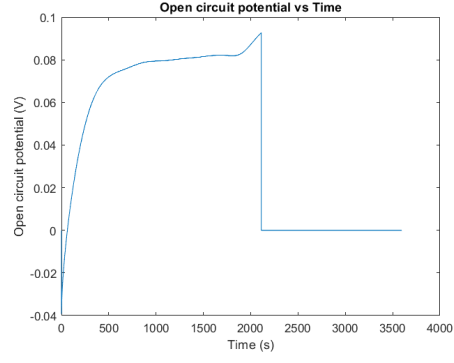


Figure 2: Open Circuit Potential vs Time

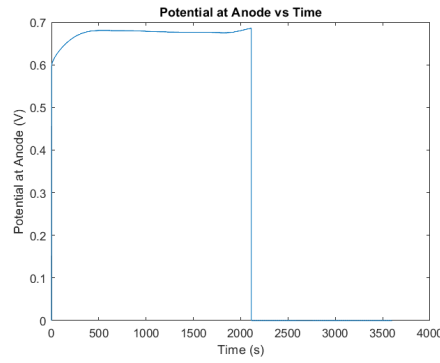


Figure 3: Anodic Potential vs Time

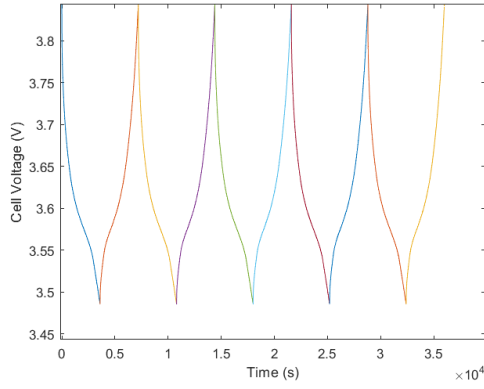
Graphs for one discharge cycle for the anode

3.B. Full Cycle model with capacity fade due to i_{sei}

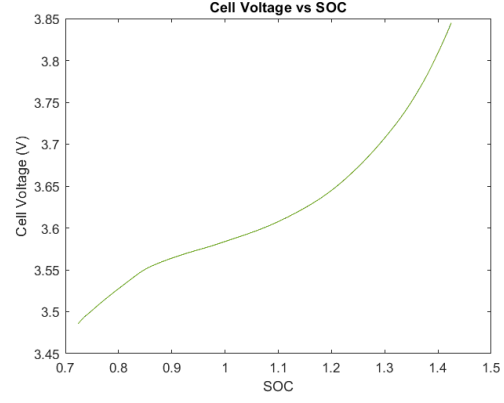
On modelling the full cycle, one can observe the cyclical nature of cell voltage [Fig. 5(a)]. The cell voltage decreases during discharge and increases during the charge cycle and varies between 3.4 and 3.8 V. In the case of no capacity fade, this cycle can go on indefinitely.

However due to SEI plate formation the total current keeps decreasing as there is loss of Li each charging cycle. In the obtained graphs, capacity fade can't be observed on the cell voltage vs SOC graph [Fig. 5(b)] for a small number of cycles because of the small value of i_{sei} (of the order of 10^{-5}).

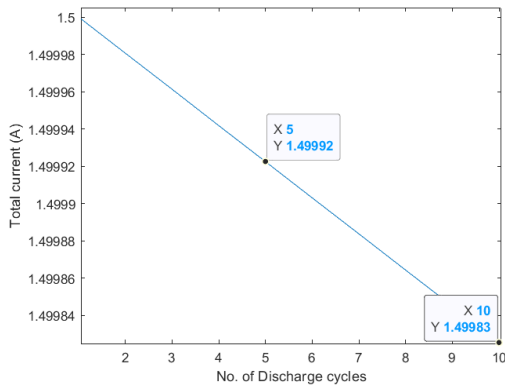
It can also be seen that the amount of charge lost per cycle remains constant leading to a linear profile for the total current [Fig. 5(b),(c)]. This is because a constant prefactor has been taken in the Tafel equation initially. The current density taken in the case was 1.5 A/m^2 .



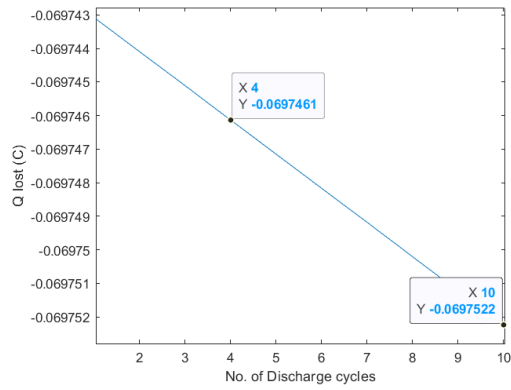
(a) Cell Voltage vs Time



(b) Cell Voltage vs SOC during charging



(c) Total current vs No. of Charge cycles



(d) Charge lost vs No. of Charge cycles

Figure 5: Graphs for full cycle SP model with capacity fade

4. Conclusion

An anodic degradation model has been developed that observes the growth of the SEI film and subsequent loss of capacity over charge and discharge cycles. The mathematical model for coupling the SEI film growth with lithium plating/stripping based on the anodic potential value has been studied and will be fully implemented in the future. A variable prefactor for the Tafel equation based on the film thickness will also be implemented.

The model developed in this project can help in identifying the parameters and preventing the conditions that lead to anodic degradation during fast-charging of lithium-ion batteries.

5. Code

5.A. Full Cycle SP model with capacity fade

The flow of execution is as follows:

1. **Initialization:** Universal constants, time parameters, cathode, anode, and other parameters are initialized.
2. **Simulation Start:** The simulation loop begins, iterating over each cycle of the battery operation.
3. **Cycle Type Determination:** Based on the counter value, the type of cycle (discharge or charge) is determined.

4. **Propagation Loop:** Within each cycle, a propagation loop runs through each time step to update the concentration profiles of lithium ions within the cathode and anode.
5. **Calculation of parameters:** Open circuit voltage, SOC and other parameters are calculated based on the concentration profiles.
6. **Newton-Raphson Method:** Depending on the cycle type, the Newton-Raphson method is applied to solve for electric potentials and currents at each time step.
7. **Voltage Cutoff Check:** The code checks for voltage cutoff conditions to determine if the cycle should be terminated early.
8. **Reinitialization:** After each cycle, parameters are updated for the next cycle.
9. **Plotting:** Finally, the code plots the thickness of the solid electrolyte interphase (SEI) layer and the total current over cycles.
10. **End of Simulation:** The simulation loop ends after all cycles are completed.

The code simulates the behavior of a lithium-ion battery over multiple charge-discharge cycles.

Listing 1: SP_model_cap_fade.m

```
1 clc
2 clearvars
3 close all
4
5 %Universal values
6 cycles = 10; % Total number of cycles
7 cyc_vec = 1:2:cycles+1; % Vector representing cycles
8 F = 96485; % Faraday's constant
9 C_Li = 1000; % Lithium concentration
10 T = 298; % Temperature (Kelvin)
11 Rgc = 8.314; % Gas constant
12
13 % Time parameters
14 dt = 0.05; % Time step
15 t = 0:dt:3600; % Time vector
16 Nt = numel(t); % Number of time steps
17 voltage = zeros(Nt,1); % Initialize voltage vector
18
19 %CATHODE - constants
20
21 R_p = 8.5E-6; % Cathode radius
22 dr_p = R_p/20; % Cathode step size
23 r_p = 0:dr_p:R_p; % Cathode radius vector
24 D_p = 1E-14; % Diffusion coefficient for cathode
25 Nr_p = numel(r_p); % Number of cathode steps
26 Cmax_p = 51410; % Maximum lithium concentration in cathode
27 curr_den_p = 1.5; % Initial current density for cathode
28 a_p = 0.5; % Alpha parameter for cathode
29 k_p = 6.67E-11; % Rate constant for cathode
30
31
32 %ANODE - constants
33
34 R_n = 12.5E-6; % Anode radius
35 dr_n = R_n/20; % Anode step size
36 r_n = 0:dr_n:R_n; % Anode radius vector
```

```
37 D_n = 3.9E-14; % Diffusion coefficient for anode
38 Nr_n = numel(r_n); % Number of anode steps
39 Cmax_n = 31833; % Maximum lithium concentration in anode
40 curr_den_n = curr_den_p; % Initial current density for anode
41 a_n = 0.5; % Alpha parameter for anode
42 k_n = 1.764E-11; % Rate constant for anode
43
44
45 % Parameters for parasitic reaction
46 Us_ocp = 0.4; % Open circuit potential
47 Js = 0.75E-2; % Exchange current density
48 a_s = 0.5; % Alpha parameter for parasitic reaction
49 rho_sei = 1690; % SEI density
50 cond_sei = 5E-6; % SEI conductivity
51 M_sei = 0.162; % SEI molar mass
52
53 % Parameters for lithium plating
54 a_pl = 0.5; % Alpha parameter for plating
55 U_pl = 0; % Plating potential
56 c_e = 1000; % Lithium concentration
57 k_pl = 2.23E-7; % Rate constant for plating
58 rho_pl = 534; % Plating density
59 cond_pl = 1.1E7; % Plating conductivity
60 M_pl = 0.007; % Plating molar mass
61 J_pl = F*k_pl*(c_e^(0.5)); % Plating current density
62
63 %Parameters for lithium stripping
64 m_pl = zeros(Nt,1); % Plating thickness vector
65
66 %CATHODE - variables
67 C_p = zeros(Nr_p,Nt); % Cathode concentration matrix
68 over_pot_p = zeros(Nt,1); % Overpotential for cathode
69 U_ocp_p = zeros(Nt,1); % Open circuit potential for cathode
70 phi_p = zeros(Nt,1); % Electric potential for cathode
71
72 %ANODE - variables
73 C_n = zeros(Nr_n,Nt); % Anode concentration matrix
74 over_pot_n = zeros(Nt,1); % Overpotential for anode
75 U_ocp_n = zeros(Nt,1); % Open circuit potential for anode
76 phi_n = zeros(Nt,1); % Electric potential for anode
77 SOC = zeros(Nt,1); % State of charge
78
79 %Parasitic reaction - variables
80 i_tot = zeros(cycles,1); % Total current vector
81 i_int = zeros(Nt,1); % Intercalation current vector
82 i_sei = zeros(Nt,1); % SEI current vector
83 i_pl = zeros(Nt,1); % Plating current vector
84 i_str = zeros(Nt,1); % Stripping current vector
85 Q_lost = zeros(cycles,1); % Lost charge vector
86
87 %Resistance model - variables
88 t_film = zeros(cycles,1); % Film thickness vector
89 t_sei = zeros(cycles,1); % SEI thickness vector
90 t_li = zeros(cycles,1); % Lithium thickness vector
91
92 %Initialisation
93 C_p(:,1) = 0.5*Cmax_p; % Initial cathode concentration
94 C_n(:,1) = 0.95*Cmax_n; % Initial anode concentration
```

```

95 i_tot(1,1) = curr_den_p; % Initial total current
96 t_film(1,1) = 10E-9; % Initial film thickness
97 t_sei(1,1) = 10E-9; % Initial SEI thickness
98
99 % Start of simulation
100 counter = 0; % Counter for cycle type
101 for cycle = 1:cycles
102     % Current density remains constant for one cycle
103
104     % Determine type of cycle
105     if counter == 0 % Discharge cycle
106         counter = 1;
107         if curr_den_p < 0
108             curr_den_p = i_tot(cycle,1);
109             curr_den_n = i_tot(cycle,1);
110         end
111
112     elseif counter == 1 % Charge cycle
113         counter = 0;
114         if curr_den_p > 0
115             curr_den_p = -i_tot(cycle,1);
116             curr_den_n = -i_tot(cycle,1);
117         end
118     end
119
120
121     % Propagation loop
122     for j=1:Nt % Time loop
123
124         if j~=1 % Not at boundary
125             % Cathode concentration update
126             for i=2:Nr_p-1 % space loop for anode
127                 C_p(i,j)=C_p(i,j-1)+D_p*dt*(((C_p(i+1,j-1)-2*C_p(i,j-1)+C_p(i-1,j-1))
128             end
129             % Anode concentration update
130             for i=2:Nr_n-1 % space loop for cathode
131                 C_n(i,j)=C_n(i,j-1)+D_n*dt*(((C_n(i+1,j-1)-2*C_n(i,j-1)+C_n(i-1,j-1))
132             end
133
134             % Boundary conditions
135             C_p(1,j)=C_p(2,j);
136             C_p(Nr_p,j)=C_p(Nr_p-1,j)+curr_den_p*dr_p/(D_p*F);
137
138             C_n(1,j)=C_n(2,j);
139             C_n(Nr_n,j)=C_n(Nr_n-1,j)-curr_den_n*dr_n/(D_n*F);
140         end
141
142         % Some variables for current calculation
143         J_0_p = k_p*((Cmax_p-C_p(Nr_p,j))*C_p(Nr_p,j)*C_Li)^a_p;
144         J_0_n = k_n*((Cmax_n-C_n(Nr_n,j))*C_n(Nr_n,j)*C_Li)^a_n;
145         SOC_p = C_p(Nr_p,j)/Cmax_p;
146         SOC_n = C_n(Nr_n,j)/Cmax_n;
147         SOC(j,1) = SOC_n;
148         U_ocp_p(j,1) = Uocp_interp2(SOC_p); %:)
149         U_ocp_n(j,1) = Uocp_interp(SOC_n);
150
151         cond_film = cond_sei;
152         R_film = t_film(cycle,1)/cond_film;
    
```

```

153
154     if curr_den_p <= 0 % Charge cycle
155
156         if j~=1 && phi_n(j-1,1)< 0
157             % Plating
158
159             result = Newton_Raphson_pl(curr_den_n, J_0_n, a_n, Js,
160                 Us_ocp, U_ocp_n(j,1),a_s, J_pl, U_pl, R_film);
161             phi_n(j,1) = result(1,1);
162             i_int(j,1) = result(1,2);
163             i_sei(j,1) = result(1,3);
164             i_pl(j,1) = result(1,4);
165
166         elseif j~=1 && phi_n(j-1,1) > 0 && max(m_pl) > 0
167             %Stripping
168
169             result = Newton_Raphson_str(0, J_0_n, a_n, Js, Us_ocp,
170                 U_ocp_n(j,1),a_s, J_pl, R_film, m_pl(j-1,1),
171                 max(m_pl)); %CHANGE
172             phi_n(j,1) = result(1,1);
173             i_int(j,1) = result(1,2);
174             i_sei(j,1) = result(1,3);
175             i_str(j,1) = result(1,4);
176
177         else
178
179             result = Newton_Raphson2(curr_den_n, J_0_n, a_n, Js,
180                 Us_ocp, U_ocp_n(j,1),a_s);
181             phi_n(j,1) = result(1,1);
182             i_int(j,1) = result(1,2);
183             i_sei(j,1) = result(1,3);
184
185         end
186
187         over_pot_p(j,1) =
188             (Rgc*T/(a_p*F))*asinh(i_int(j,1)/(2*J_0_p));
189         phi_p(j,1) = over_pot_p(j,1) + U_ocp_p(j,1);
190
191         % Calculate cell voltage
192         voltage(j,1) = phi_p(j,1) - phi_n(j,1);
193         %Loss of Li ions
194         Q_lost(cycle,1) = Q_lost(cycle,1) + i_sei(j,1)*dt +
195             i_pl(j,1)*dt + i_str(j,1)*dt; %negative
196
197         % Thickness of SEI layer + plating/stripping
198         t_sei(cycle,1) = t_sei(cycle,1) -
199             (i_sei(j,1)*M_sei*dt)/(F*rho_sei);
200         t_li(cycle,1) = t_li(cycle,1) -
201             (i_pl(j,1)*M_pl*dt)/(F*rho_pl);
202         t_film(cycle,1) = t_sei(cycle,1) + t_li(cycle,1);
203         m_pl(j,1) = 4*pi*R_n*R_n*t_li(cycle,1)*rho_pl;
204
205     elseif curr_den_p > 0 % Discharge cycle
206
207         % Calculate overpotentials
208         over_pot_p(j,1) =
209             (Rgc*T/(a_p*F))*asinh(curr_den_p/(2*J_0_p));

```



```

202         over_pot_n(j,1) =
203             (Rgc*T/(a_n*F))*asinh(curr_den_n/(2*J_0_n));
204
205         % Calculate electric potentials
206         phi_p(j,1) = over_pot_p(j,1) + U_ocp_p(j,1);
207         phi_n(j,1) = over_pot_n(j,1) + U_ocp_n(j,1);
208
209         % Calculate cell voltage
210         voltage(j,1) = phi_p(j,1) - phi_n(j,1);
211
212     end
213
214     % Check for voltage cutoff
215
216     if (voltage(j,1)<3 || SOC_n<0.05) && counter == 1 %Discharge
217         cutoff
218         break
219     elseif (voltage(j,1)>4.2 || SOC_p<0.5) && counter == 0 %Charge
220         cutoff
221         break
222     end
223
224     % Reinitialization
225     i_tot(cycle+1,1) = i_tot(cycle,1) + Q_lost(cycle,1)/3600;
226     t_film(cycle+1,1) = t_film(cycle,1);
227     t_sei(cycle+1,1) = t_sei(cycle,1);
228     t_li(cycle+1,1) = t_li(cycle,1);
229     m_pl = zeros(Nt,1);
230     C_p(:,1) = C_p(Nr_p, j);
231     C_n(:,1) = C_n(Nr_n, j);
232
233     % Plot variation of voltage with time
234     figure(1)
235     timevec = 0+(cycle-1)*3600:dt:3600+(cycle-1)*3600;
236     plot(timevec,voltage,'-');
237     xlabel("Time (s)")
238     ylabel("Cell Voltage (V)")
239     pause(1)
240     hold on
241
242
243     % Plot variation of voltage with SOC
244     if mod(cycle,2) == 0
245         figure(2)
246         timevec2 = transpose(0:dt:3600);
247         answer = SOC./timevec2;
248         plot(answer*abs(curr_den_p), voltage);
249         title("Cell Voltage vs SOC")
250         xlabel("SOC")
251         ylabel("Cell Voltage (V)")
252         pause(1)
253         hold on
254     end
255
256 end
    
```

5.B. Newton Raphson for calculating anodic/cathodic cell potential and currents

This function estimates the i_{int} , i_{sei} and V when the plating/stripping condition is not met.

Listing 2: Newton_Raphson2.m

```

1 function [result] = Newton_Raphson2(curr_den, J_0, a, Js, Us_ocp, U_ocp,
   a_s)
2     % Constants
3     F = 96485; % Faraday's constant (C/mol)
4     T = 298;   % Temperature (K)
5     R = 8.314; % Gas constant (J/(mol*K))
6
7     % Initial guess for V
8     V = U_ocp + 0.5;
9
10    % Newton-Raphson iteration loop
11    while true
12        % Calculate function value and its derivative
13        val = -2*J_0*sinh(a*F*(V - U_ocp)/(R*T)) - Js*exp(-a_s*F*(V -
           Us_ocp)/(R*T)) - curr_den;
14        derivative = -2*J_0*(a*F/(R*T))*cosh(a*F*(V - U_ocp)/(R*T)) +
           Js*(a_s*F/(R*T))*exp(-a_s*F*(V - Us_ocp)/(R*T));
15
16        % Update voltage using Newton-Raphson method
17        phi_new = V - (val/derivative);
18
19        % Check for convergence
20        if norm(phi_new - V) < 1e-10
21            break
22        end
23
24        % Update V for the next iteration
25        V = phi_new;
26    end
27
28    % Calculate final values of i_int and i_sei
29    i_int = 2*J_0*sinh(a*F*(V - U_ocp)/(R*T));
30    i_sei = -Js*exp(-a_s*F*(V - Us_ocp)/(R*T));
31
32    % Return the result as a vector
33    result = [V, i_int, i_sei];
34 end

```

When the plating/stripping conditions are met, a slightly different form is used. The form used for plating has been given below:

Listing 3: Newton_Raphson_pl.m

```

1 function [result] = Newton_Raphson_pl(curr_den, J_0, a, Js, Us_ocp, U_ocp,
   a_s, J_pl, U_pl, R_film)
2     % Constants
3     F = 96485; % Faraday's constant (C/mol)
4     T = 298;   % Temperature (K)
5     R = 8.314; % Gas constant (J/(mol*K))
6
7     % Initial guess for V
8     V = U_ocp + 0.5;
9
10    % Newton-Raphson iteration loop
11    while true

```

```

12     % Calculate function value and its derivative
13     val = 2*J_0*sinh(a*F*(V - U_ocp - R_film*curr_den)/(R*T)) -
        Js*exp(-a_s*F*(V - Us_ocp - R_film*curr_den)/(R*T)) + min(0,
        2*J_pl*sinh(a*F*(V - U_pl - R_film*curr_den)/(R*T))) -
        curr_den;
14     derivative = 2*J_0*(a*F/(R*T))*cosh(a*F*(V - U_ocp -
        R_film*curr_den)/(R*T)) + Js*(a_s*F/(R*T))*exp(-a_s*F*(V -
        Us_ocp - R_film*curr_den)/(R*T));
15
16     % Adjust derivative if parasitic current density is negative
17     if 2*J_pl*sinh(a*F*(V - U_pl - R_film*curr_den)/(R*T)) < 0
18         derivative = derivative + 2*J_pl*(a*F/(R*T))*cosh(a*F*(V -
        U_pl - R_film*curr_den)/(R*T));
19     end
20
21     % Update voltage using Newton-Raphson method
22     phi_new = V - (val/derivative);
23
24     % Check for convergence
25     if norm(phi_new - V) < 1e-10
26         break
27     end
28
29     % Update V for the next iteration
30     V = phi_new;
31 end
32
33 % Calculate final values of i_int, i_sei, and i_pl
34 i_int = 2*J_0*sinh(a*F*(V - U_ocp - R_film*curr_den)/(R*T));
35 i_sei = -Js*exp(-a_s*F*(V - Us_ocp - R_film*curr_den)/(R*T));
36 i_pl = min(0, 2*J_pl*sinh(a*F*(V - U_pl - R_film*curr_den)/(R*T)));
    % Parasitic current density
37
38 % Return the result as a vector
39 result = [V, i_int, i_sei, i_pl];
40 end

```

5.C. OCP interpolation Function for anode and cathode

Listing 4: Uocp_interp.m

```

1 function [U_ocp] = Uocp_interp(SOC)
2 % Uocp_interp: Interpolates open circuit voltage (U_ocp) based on state
  of charge (SOC)
3 % Inputs:
4 %   SOC: State of charge (scalar or array)
5 % Outputs:
6 %   U_ocp: Interpolated open circuit voltage corresponding to the input
  SOC
7
8 % Data points for SOC and corresponding U_ocp
9 x = [0.15511559664628977, 0.050825157963702156, 0.06006604083815564, ...
10      0.3399340598799139, 0.07194724790393106, 0.1155116402390845, ...
11      0.19075911712554655, 0.22376241413155087, 0.26864691820333064, ...
12      0.3095709863471621, 0.3742574686225445, 0.40594065389192246, ...
13      0.44158417437117936, 0.47722769485043615, 0.5102309918564405, ...
14      0.5379538419159398, 0.5735973623951964, 0.6052805476645746, ...
15      0.6369637329339526, 0.6726072534132095, 0.7148514332936673, ...

```

```

16     0.7504950544909936, 0.7993398937726521, 0.8442243978444317, ...
17     0.8838284549697064, 0.9141913270663193, 0.9353135177246177, ...
18     0.9511550600002721, 0.04554461029912765, 0.04158417437117928, ...
19     0.03630372742467419, 0.033663403233352124, 0.02838285556877762, ...
20     0.025742632095524992, 0.021782196167576706, 0.01914197269432416,
    ...
21     0.017821860957697916];
22 U = [0.16367350561227423, 0.28693879203366895, 0.25428574097746526, ...
23     0.1171429016291087, 0.21755105853923606, 0.19387752178658568, ...
24     0.1440816936627777, 0.12775516813467586, 0.11795924658973952, ...
25     0.11795924658973952, 0.11469392902796871, 0.11387761520771406, ...
26     0.11061229764594324, 0.10326537984252307, 0.09591836861797454, ...
27     0.09020414073581598, 0.08285722293239581, 0.082040877971765, ...
28     0.082040877971765, 0.08122456415151035, 0.08040821919087959, ...
29     0.07959187423024884, 0.07877556040999413, 0.07551024284822332, ...
30     0.07142861146619786, 0.06244903488189233, 0.04938782691556128, ...
31     0.03469392902796868, 0.3065306039831655, 0.32857141967417824, ...
32     0.3506122353651909, 0.3726530354860156, 0.3979591531686112, ...
33     0.424897914061904, 0.44612240036247397, 0.4689795454439294, ...
34     0.4869387063976346];
35
36 % Interpolate U_ocp using piecewise cubic Hermite interpolation
37 U_ocp = interp1(x, U, SOC, 'pchip');
38 end
    
```

Similarly for the cathode,

Listing 5: Uocp_interp2.m

```

1 function [U_ocp] = Uocp_interp2(SOC)
2
3 x = [0, 0.0505263157894737, 0.101052631578947; ...
4     0.151578947368421, 0.202105263157895, 0.252631578947368; ...
5     0.303157894736842, 0.353684210526316, 0.404210526315789; ...
6     0.454736842105263, 0.505263157894737, 0.555789473684211; ...
7     0.606315789473684, 0.656842105263158, 0.707368421052632; ...
8     0.757894736842105, 0.808421052631579, 0.858947368421053; ...
9     0.909473684210526, 0.960000000000000];
10 U = [2.11080000000000, 3.12419643036539, 3.78640080467038; ...
11     4.18132750493529, 4.37972261739175, 4.44019975687525; ...
12     4.41027589121773, 4.32740716564021, 4.22002472714531; ...
13     4.10857054890990, 4.00653325467765, 3.92148394315164; ...
14     3.85611201238692, 3.80926098418315, 3.77696432847713; ...
15     3.75348128773538, 3.73233270134684, 3.70733683001532; ...
16     3.67364518015216, 3.62877832826878];
17
18 U_ocp = interp1(x,U,SOC,'pchip');
19 end
    
```

6. References

1. Meng Guo et al.; Single-Particle Model for a Lithium-Ion Cell: Thermal Behavior 2011
2. Gang Ning and Branko N. Popov; Cycle Life Modeling of Lithium-Ion Batteries 2004
3. Abhishek Sarkar et al.; Performance degradation due to anodic failure mechanisms in lithium-ion batteries 2021

7. Nomenclature

- Solid phase concentration of lithium (c_i)
- Solid phase concentration of lithium at electrode/electrolyte interface ($c_{s,surf}$)
- Maximum solid phase concentration of lithium ($c_{s,max}$)
- Initial solid phase concentration of lithium (c_0)
- Diffusion coefficient (D)
- Current density (j)
- Faraday's constant (F)
- Total current (i_{tot})
- Intercalation current (i_{el})
- SEI growth current (i_{SEI})
- Plating current (i_{pl})
- Stripping current (i_{st})
- Reaction rate constant (k)
- Capacity lost due to parasitic reaction (Q_s)
- Radial coordinate (r)
- Universal gas constant (R)
- Radius of particles (R_i)
- Resistance of newly formed film (R_{film})
- Superficial surface area (S)
- Time (t)
- Temperature (T)
- Anodic and cathodic transfer coefficients of electrochemical reaction (α_a, α_c)
- Volume fraction of a solid phase (ε)
- Potential of a phase (φ)
- Open circuit potential (U_{OCP})
- Overpotential of electrochemical reaction (η)
- Conductivity of the newly formed surface film (κ)
- Density of active material (ρ)
- Film thickness (δ)
- Film mass (m)
- Molecular weight (M)