Journal of The Electrochemical Society

The Electrochemical Society
Advancing solid state & electrochemical science & technology

A Robust and Sleek Electrochemical Battery Model Implementation: A MATLAB® Framework

To cite this article: Seong Beom Lee and Simona Onori 2021 *J. Electrochem. Soc.* **168** 090527

View the article online for updates and enhancements.

Journal of The Electrochemical Society, 2021 **168** 090527
1945-7111/2021/168(9)/090527/17/$40.00 © 2021 The Electrochemical Society ("ECS"). Published on behalf of ECS by IOP Publishing Limited

A Robust and Sleek Electrochemical Battery Model Implementation: A MATLAB® Framework

Seong Beom Lee and Simona Onori[z]

Department of Energy Resources Engineering, Stanford University, Stanford, CA 94305, United States of America

A Doyle-Fuller-Newman electrochemical battery model implementation in a robust and sleek MATLAB® framework for lithium-ion batteries as an open-access MATLAB code is presented. The Doyle-Fuller-Newman (DFN) model, in the form of partial differential equations, is first numerically discretized then converted to a differential algebraic equation (DAE). The most efficient way to implement a DAE system is through the adoption of standard DAE solvers provided by available commercial software. MATLAB® is a widely used software in the control community, and to the best of our knowledge, its standard solvers have failed to successfully simulate the DFN model when the battery undergoes high C-rates of operations. One critical issue with DFN model simulation is related to the *inconsistency of initial conditions*. In 2015, a robust single-step iteration-free initialization approach, enabling solving DAE systems using a standard solver using Maple® symbolic environment, was proposed by *Lawder* et al. A symbolic environment enables direct and efficient derivation of implicit ordinary differential equations from algebraic equations during initialization. We perform the single-step iteration-free initialization approach in MATLAB® environment by adopting the MATLAB® symbolic toolbox and simulate the DFN model with the *ode15s* solver. This framework allows users to robustly simulate the DFN model and identify model parameters directly after numerical discretization, utilizing a standard solver.
[DOI: 10.1149/1945-7111/ac22c8]

Manuscript submitted May 26, 2021; revised manuscript received August 3, 2021. Published September 21, 2021.

Lithium-ion battery (LIB) systems take advantage of the high electrochemical potential of lithium (−3.040 V vs standard hydrogen electrode) and provide the highest specific power density (300–1500 W kg^{-1}) and specific energy density (100–250 Wh kg^{-1}) against all the electrochemical battery devices available on the market nowadays.[1] For this reason, LIBs are currently considered the best means to store energy to support sustainable transportation systems and renewable grid applications. In 2019, for example, a market of LIBs was valued $36.7 billion and is expected to grow by an annual rate of 18.0% to reach $129.3 billion by 2027.[2]

The growth of the LIB market is propelled by the increasing demand electric vehicles (EVs) and renewable grid applications. There were more than 1.18 million EVs on the U. S. roads in 2019. Total EV sales in 2018 were increased by 81% compared to 2017. In the first quarter of 2019, more than 61,000 electric vehicles were sold in the U.S., representing a 10% increase over the same period in 2018.[3] LIBs with different cathode chemistries, such as nickel-manganese-cobalt oxide (NMC) and nickel-cobalt-aluminum oxide (NCA), have been adopted for EV systems.[4]

A microgrid is a sub part of renewable grid systems, which is defined as a discrete energy system consisting of distributed energy sources and loads capable of operating independently from the main power grid.[5] The global microgrid market is currently valued at $28.6 billion and is expected to grow at an annual growth rate of 10.6% to reach $47.7 billion by 2025.[5] Electrochemical energy systems are the main storage technology implemented and used today for renewable grid services. Of electrochemical storage systems installed since 2003, nearly 80% is proved by LIBs.[6]

Motivation

Despite the progress made in materials and manufacturing processes that contributed to significantly reducing the cost of batteries over the years, LIBs are still the most expensive single components in EV systems and renewable grid applications (in 2018, EV batteries were at $156/kWh and grid batteries at $223–323/kWh).[7] When LIB systems are deployed for energy applications, mathematical battery models are used to understand, operate, and optimize their performance. The operation of battery systems by adequate battery models can achieve further cost reduction, enabling a greater depth of discharge, smaller size, and longer life. This helps accelerate adoption of sustainable transportation systems and renewable grid applications.

A Battery management system (BMS) is used in both EVs and stationary storage systems to monitor and optimize the system behavior while guaranteeing safety. Current practice employs empirical models to predict state of charge (SOC) and state of health (SOH) owing to their easy and straightforward implementation.[8] Typically, empirical models are calibrated upon laboratory-based experimental data (e.g., 1C-constant charging and discharging) and used to predict battery behavior.[9] The accuracy of such models is usually limited to the range over which they have been parameterized and the calibration effort that goes to design accurate empirical models is very high and chemistry-dependent, meaning that if a new chemistry is to be adopted then a completely new BMS has to be recalibrated. In addition, empirical models do not contain any physical insights nor account for any aging degradation mechanisms the battery undergoes to making them a rather simplistic tool.[8] The lack of physicochemical information of the system states and parameters causes inaccurate SOH predictions of the battery system.

Now, the following fundamental questions arise. If the current empirical models under-utilize the LIB system, would there be other advanced mathematical models most likely to be implemented in the next-generation BMS to maximize the usability and lifetime of the battery system? If so, what obstacles would exist for these battery models to be implemented in the BMS today? If such barriers delay the implementation of advanced battery models in BMS, what research approach would be required to accelerate the development of advanced model-based BMS?

Advanced battery models.—Advanced LIB models can be roughly categorized into molecular-level models, data-driven models, and physics-based models.[8] Molecular-level battery models are used to understand LIBs as they adopt stochastic approaches (e.g., kinetic Monte Carlo methods), molecular dynamics, and density functional theory calculations to provide useful analytical tools to understand chemical/kinetic phenomena at the molecular and atomic scale; however, molecular-level models cannot be used for battery controls due to their high computational cost (e.g., hour-scale simulation).[10]

Data-driven models formulated upon machine learning techniques have recently gained a great attention as tools to predict battery performance due to their potential in achieving high accuracy with low computational cost (e.g., millisecond-scale simulation). However, data-driven battery models typically rely on extensive

[z] E-mail: sonori@stanford.edu

experimental data sets to predict the battery output (e.g., 20000 ∼ 30000 data sets).[11] More importantly, data-driven battery models provide limited information on capacity/power fade as no information on the electrochemical internal states are provided.[12]

Physics-based battery models characterize internal dynamics such as transport and kinetic phenomena. Moreover, physics-based battery models can incorporate transport phenomena, chemical/electrochemical kinetics, side reactions, and thermal/stress/mechanical effects.[10] In LIBs, safety and capacity/power fade issues can always occur during the battery operation, including thermal runaway, electrolyte decomposition processes, lithium plating, and formation of the solid-electrolyte interphase layer. For this reason, an accurate estimate of the SOC and SOH is one of the key tasks of the BMS. The SOC is defined as the level of charge of battery relative to its capacity (0 ∼ 100%), and one of the SOH definitions is the capacity of the battery in its fully charged state compared to the capacity at the time of manufacturing industry (0 ∼ 100%).[8] However, simply providing numerical values for the SOC and SOH, as in empirical models, does not guarantee safety and full utilization of the battery system.

Physics-based models offer internal electrochemical information that can be used to maximize safety, usability and lifetime of the battery.[13] For example, lithium plating is considered one of the main causes of capacity fade and thermal runaway in LIBs which is known to occur during fast charging of LIBs when the overpotential is less than zero.[10] When EV batteries are charged at high rates, a large amount of lithium ions accumulates on the electrode/electrolyte interface because the lithium solid diffusion is significantly lower than lithium ion diffusion in the electrolyte. The lithium solid diffusion limitation in the lithium ion intercalation process will result in anode lithium plating. Lithium plating not only causes capacity/power fade but also poses a significant safety issue. In addition, recent works indicate that lithium plating can also occur at mild charge conditions in cells after extended cycling, leading to rapid aging of the cell.[14,15] This is a fundamental breakthrough in the field of SOH estimation. In the physics-based modeling framework, the overpotential can be mathematically restricted above zero during charge operation to minimize lithium plating thereby identifying optimal charging routines to operate the battery system with maximized usability, safety, and lifetime.[10] For this reason, physics-based models have been regarded as advanced mathematical tools to be used in the next-generation BMS.[11] Multiple modeling approaches, including reduced-order models, efficient simulations, and alternative modeling frameworks, have been attempted towards such an advanced model-based BMS.[11,16–20]

Physics-based models.—LIBs are composed of components at both the micro- and macro-scale. Micro-scale components are the solid particles in the positive and negative electrodes, whereas the macro-scale ones include the electrolyte between the positive electrode, separator, and negative electrode. The size of solid particles varies across the electrode, and in general, each particle has an irregular shape. During charging and discharging, lithium ions move across the cell by ions transport, by means of diffusion and migration.

Conventional physics-based battery models include the Doyle-Fuller-Newman (DFN) model and its reduced counterparts, such as the enhanced single particle model (ESPM) and single particle model (SPM), as shown in Fig. 1.[19] The DFN model is the most widely accepted physics-based model to predict electrochemical dynamics of LIBs in the battery community. The DFN model is formulated based on the porous electrode and concentrated solution theories, consisting of two porous electrodes, a separator, two current collectors, and electrolyte. The electrodes are considered to be a porous matrix, in which spherical solid particles surrounded by electrolytes are described. Diffusion and migration of lithium ions are modeled within the porous matrix. The DFN model is formulated on assumptions that electrodes include ideal spherical-shape particles, and the effective coefficients for ionic diffusion and conductivity are derived based on empirical laws.[21] These assumptions can make the model ineffective–with lack of predictability leading to underutilization (or overutilization) of the battery system–and inaccuracy at the condition of operations in which EV and grid batteries would experience, e.g., low SOC, medium-high temperature, and medium-high C-rate.[16]

The DFN model is referred to as the pseudo-two-dimensional (P2D) model because micro-scale components solve the diffusion of lithium occurring in a spherical solid particle at each local position of the macro-scale LIB system.[17] In the DFN framework, solid particles are distributed in one dimension along the *x-axis* of electrolyte, and the solid particle concentration is described along with the *r* dimension. Therefore, the concentration of each solid particle is solved along with the *r* dimension at a particular point on the *x-axis* in the electrolyte. This intrinsic characteristic of the DFN framework yields the same results as solving a two-dimensional model, leading to high computational cost. For this reason, within the DFN modeling framework, additional assumptions are added to generate simplified modeling formulations. For example, the ESPM model assumes that each electrode can be represented by a single spherical particle, which implies that all solid particles are uniform and have the same chemical properties. In SPM, the electrolyte dynamics is ignored.[10] These reduced order models lack in predictability under conditions of operation when the underlying assumptions are not verified, leading to underutilization (or overutilization) of the battery system and inaccuracy.[23]

To overcome the limitations of DFN model, the full homogenized macroscale (FHM) model was proposed.[16,24] In the FHM framework, homogenization methods to upscale pore-scale battery dynamics are adopted to design a macroscale model of LIBs. The FHM model is formulated under the assumption that the electrodes are composed of spatial unit cells, causing micro-scale continuity in the lithium-ion cell system. In addition, the FHM model's effective ionic properties are determined by resolving the closure problem in the unit cell of the electrode microstructure.[22] Compared to the DFN model, the FHM model provides more accurate predictions over low state of charge and medium-high temperature, as well as high C-rate. The implemented 1-D FHM modeling framework exhibits a decreased root mean square error (RMSE) up to 75% when compared to the error generated from the DFN model under medium-high temperature conditions.[16]

The DFN model.—Despite a lack of predictability of the DFN model at high temperatures, low SOC, over battery aging, and at high C-rates of operation, the DFN model has been and still is considered the most acceptable physics-based model to study, analyze and design lithium-ion batteries. The DFN model was developed in 1993, and from 1993 to 2015, accurate physics-based models, which can replace the DFN model, were not proposed. During this period, numerous studies based on the DFN model have been conducted on capacity fade, mechanical stress effects, and safety issues for LIBs. For example, capacity fade phenomena including the SEI layer growth and lithium plating have been added to the DFN model.[1,25] Mechanical stress and thermal effects have been incorporated into the DFN model to study safety issues.[26] In addition, although advanced model-based BMS has not yet been commercialized, several numerical approaches for efficient simulations (e.g., polynomial approximation, orthogonal collocation method, and non-linear model predictive control) have been attempted to implement the DFN model in the next-generation BMS.[17,27,28] Understanding the DFN model and its implementation is an enabler of advanced model-based BMS. The remaining of the paper studies, analyzes, and discusses model equations, numerical discretization methods, and implementation issues of the DFN model.

Governing equations.—The DFN model is in the form of partial differential equations (PDEs) including multivariable functions and their partial derivatives.[21] The model predicts the dynamics of

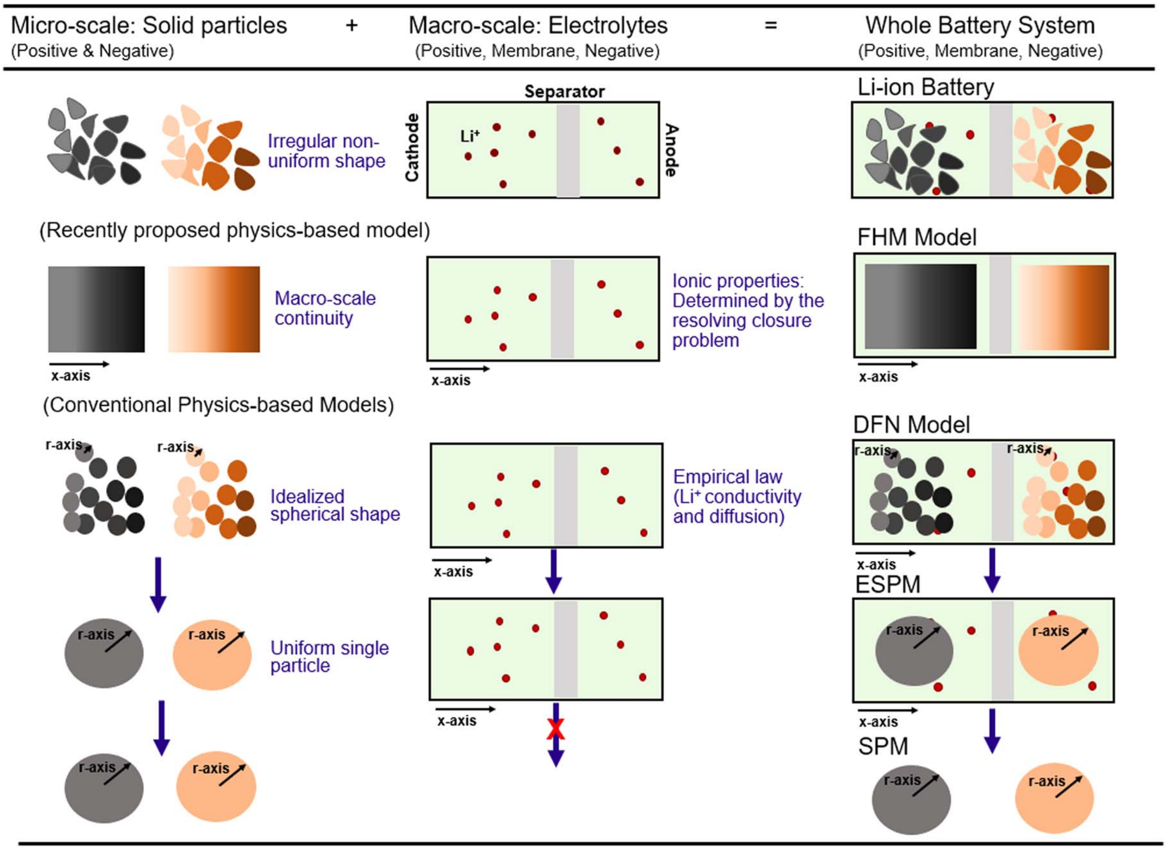
Micro-scale: Solid particles
(Positive & Negative)
+
Macro-scale: Electrolytes
(Positive, Membrane, Negative)
=
Whole Battery System
(Positive, Membrane, Negative)
Irregular non-uniform shape
Separator
Cathode
Anode
Li+
Li-ion Battery
(Recently proposed physics-based model)
FHM Model
Macro-scale continuity
Ionic properties: Determined by the resolving closure problem
x-axis
(Conventional Physics-based Models)
DFN Model
r-axis
Idealized spherical shape
Empirical law (Li+ conductivity and diffusion)
ESPM
Uniform single particle
SPM

Figure 1. Physics-based battery models. The DFN model has high computational cost and no analytical solutions. This has led to the formulation of ESPM and SPM by means of simplified descriptions of electrolyte dynamics and nonuniform reaction distributions. However, these reduced order models have predictive limitations under some usage conditions. The FHM model was proposed in 2015.[22] There are two main advantages of adopting the FHM model compared to the DFN model: (i) The FHM model consumes only 60% of the computational costs over the DFN model. (ii) Compared to the DFN model, the FHM model provides accurate predictions over the entire SoC range, over wide temperature window, and at higher C-rate.

lithium concentration in the solid phase $c_{s,j}(x, r, t)$ at the anode and at the cathode ($j = p, n$), lithium concentration in the electrolyte $c(x, t)$, electrolyte potential $\phi_e(x, t)$, and solid phase potential $\phi_{s,j}(x, t)$. All equations, variables, and parameters, which are adopted in the DFN implementation are described in Tables I–IV. Equations 1 $\sim$ 17 in Tables III–V describe the DFN model dynamics. Variables and parameters are shown in Tables I and II, respectively. Equations 1, 5, and 7 in Table III represent electrolyte dynamics in the positive electrode, separator, and negative electrode. Equations 2, 6, and 8 in Table III represent electrolyte potential in macro-scale systems. Equations 3 and 9 in Table III represent solid particle potentials in micro-scale systems, and Eqs. 4 and 10 in Table III represent solid particle concentrations in micro-scale systems.

Additional equations related to governing equations are also given in Tables IV and V. In Eqs. 11 and 12, Bruggeman coefficients are used to describe effective diffusivity and conductivity in the porous electrodes. The specific particle area (a_i), particle surface to volume (Eq. 13), is derived under the assumption that all solid particles have idealized spherical shape and uniform size. In Eqs. 14 and 15, the Butler-Volmer kinetic expression is adopted for the pole flux at the positive and negative electrode, describing the charge transfer processes, which occurs at the surface of solid particles–electrolyte interfaces. The open circuit potential (OCP) is defined as the difference of electrochemical potential between the positive and negative electrodes when the cell is disconnected from any circuit. Different cell chemistries produce different OCP equations, and three different OCPs at the positive and negative electrode are presented as examples in Table V. Equations 16–1 and 17–1 are experimentally obtained from Ref. 31 using LG M50 INR 21700 (Cathode chemistry: NMC, anode chemistry: graphite and silica), Equations 16–2 and 17–2 represent the OCP at the $LiCoO_2$ positive

Table I. List of variables for the DFN model.

Symbol	Variables	Units
c	Electrolyte concentration	mol/m^3
$c_{s,i}$	Solid phase concentration (i = *positive*, *negative*)	mol/m^3
ϕ_s	Solid phase potential	V
ϕ_e	Liquid phase potential	V
I	Applied current density	A/m^2
U_i	Open circuit potential at positive and negative (i = *pasitive*, *negative*)	V
J_i	Pore wall flux at positive and negative (i = *positive, separator, negative*)	$mol/m^2/s$
θ_i	State of charge at positive and negative (i = *positive*, *negative*)	—

Table II. Parameters for the DFN model.

Symbol	Parameter	
(Geometric parameters)		
σ_i	Solid phase conductivity (i = *positive*, *negative*)	S/m
ε_i	Porosity (i = *positive, separator*, *negative*)	—
Brugg	Bruggeman coefficient	—
D	Electrolyte diffusivity	m^2/s
$D_{s,i}$	Solid phase diffusivity (i = *positive*, *negative*)	m^2/s
$D_{s0,i}$	Reference solid phase diffusivity (i = *positive*, *negative*)	m^2/s
k_i	Reaction rate constant (i = *positive*, *negative*)	$mol/(m^2 \cdot s)$
$k_{0,i}$	Reference reaction rate constant (i = *positive*, *negative*)	$mol/(m^2 \cdot s)$
$c_{s,i,\max}$	Maximum solid phase concentration (i = *positive*, *negative*)	mol/m^3
A_{cell}	Electrode area	m^2
(Transport parameters)		
$R_{p,i}$	Particle radius (i = *positive*, *negative*)	m
a_i	Particle surface area to volume (i = *positive*, *negative*)	1/m
l_i	Region thickness (i = *posieive, separator*, *negative*)	C/mol
t_+	Transference number	—
F	Faraday's constant	C/mol
c	Electrolyte concentration	mol/m^3
(Kinetic/Thermal parameters)		
R	Gas constant	$J/(mol \cdot K)$
T	Temperature	K
A	Transfer coefficient	—
$E_{a,i,D}$	Activation energy for temperature dependent solid phase diffusion (i = *positive*, *negative*)	J/mol
$E_{a,i,k}$	Activation energy for temperature dependent reaction constant (i = *positive*, *negative*)	J/mol

Table III. Governing equations for the DFN model.

Governing equations

(Positive)

$$\varepsilon_p \frac{\partial c}{\partial t} = \frac{\partial}{\partial x}\left(D_{eff,p}\frac{\partial c}{\partial x}\right) + a_p(1 - t_+)j_p \tag{1}$$

$$-\sigma_{eff,p}\frac{\partial \phi_s}{\partial x} - \kappa_{eff,p}\frac{\partial \phi_e}{\partial x} + \frac{2\kappa_{eff,p}RT}{F}(1 - t_+)\frac{\partial lnc}{\partial x} = I \tag{2}$$

$$\frac{\partial}{\partial x}\left(\sigma_{eff,p}\frac{\partial \phi_s}{\partial x}\right) = a_p F j_p \tag{3}$$

$$\frac{\partial c_{s,p}}{\partial t} = -\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 D_{s,p}\frac{\partial c_{s,p}}{\partial r}\right) \tag{4}$$

(Separator)

$$\varepsilon_s \frac{\partial c}{\partial t} = \frac{\partial}{\partial x}\left(D_{eff,s}\frac{\partial c}{\partial x}\right) \tag{5}$$

$$-\kappa_{eff,s}\frac{\partial \phi_e}{\partial x} + \frac{2\kappa_{eff,s}RT}{F}(1 - t_+)\frac{\partial lnc}{\partial x} = I \tag{6}$$

(Negative electrode)

$$\varepsilon_n \frac{\partial c}{\partial t} = \frac{\partial}{\partial x}\left(D_{eff,n}\frac{\partial c}{\partial x}\right) + a_n(1 - t_+)j_n \tag{7}$$

$$-\sigma_{eff,n}\frac{\partial \phi_s}{\partial t} - \kappa_{eff,n}\frac{\partial \phi_l}{\partial x} + \frac{2\kappa_{eff,n}RT}{F}(1 - t_+)\frac{\partial lnc}{\partial x} = I \tag{8}$$

$$\frac{\partial}{\partial x}\left(\sigma_{eff,n}\frac{\partial \phi_s}{\partial x}\right) = a_n F j_n \tag{9}$$

$$\frac{\partial c_{s,n}}{\partial t} = -\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 D_{s,n}\frac{\partial c_{s,n}}{\partial r}\right) \tag{10}$$

Table IV. Additional equations for DFN model.

$$D_{eff,i} = D \cdot \varepsilon_i^{\,brugg}, \quad i = p,\ s,\ n \tag{11}$$

$$\sigma_{eff,i} = \sigma_i(1 - \varepsilon_i), \quad i = p,\ s,\ n \tag{12}$$

$$a_i = \frac{3}{R_i}(1 - \varepsilon_i), \quad i = p,\ n, \tag{13}$$

$$j_p = 2c^{0.5}k_p c_{sp,surf}(c_{sp,\max} - c_{sp,surf})^{0.5}\, sinh\left(\frac{\alpha F}{RT}(\Phi_s - \Phi_e - U_p)\right) \tag{14}$$

$$j_n = 2c^{0.5}k_n c_{sn,surf}(c_{sn,\max} - c_{sn,surf})^{0.5}\, sinh\left(\frac{\alpha F}{RT}(\Phi_s - \Phi_e - U_n)\right) \tag{15}$$

Table V. Open circuit potential at the positive and negative electrode.

LG INR 21700[29]	
$U_p = -0.8090\theta_p + 4.4875 - 0.0428\tanh(18.5138(\theta_p - 0.5542))$ $-17.7326\tanh(15.7890(\theta_p - 0.3117))$ $+17.5842\tanh(15.9308(\theta_p - 0.3120))$ $(0.2661 \leqslant \theta_p \leqslant 0.9084);\ \theta_p = \frac{c_{sp,surf}}{c_{sp,\max}}$	(16–1)
$U_n = 0.2482 + 1.97930\theta\exp(-39.3631\theta_n)$ $+0 - 0.0909\tanh(29.8538(\theta_n - 0.1234))$ $-0.04478\tanh(14.9159(\theta_n - 0.2769))$ $-0.0205\tanh(30.4444(\theta_n - 0.2769_n - 0.6103))$ $(0.0279 \leqslant \theta_n \leqslant 0.9014);\ \ \theta_n = \frac{c_{sn,surf}}{c_{sn,\max}};$	(17–1)
Cathode:$LiCoO_2$, Anode: Graphite[30]	
$U_p = \frac{-4.656 + 88.669{\theta_p}^2 - 401.119{\theta_p}^4 + 342.909{\theta_p}^6 - 462.471{\theta_p}^8 + 433.434{\theta_p}^{10}}{-1 + 18.933 - 79.532 + 37.311 - 73.083 + 95.96}$ $(0.4 \leqslant \theta_p \leqslant 1);\ \theta_p = \frac{c_{sp,surf}}{c_{sp,\max}}$	(16–2)
$U_n = 0.7222 + 0.1387\theta_n + 0.029{\theta_n}^{0.5}$ $-\frac{0.0172}{\theta_n} + \frac{0.0019}{{\theta_n}^{1.5}} + 0.2808\exp(0.90 - 15\theta_n)$ $-0.7984\exp(0.4465\theta_n - 0.4108)$ $(0.01 \leqslant \theta_n \leqslant 1);\ \ \theta_n = \frac{c_{sn,surf}}{c_{sn,\max}}$	(17–2)
LG INR 18650[15]	
$U_p = -10.72{\theta_p}^4 + 23.88{\theta_p}^3 - 16.77{\theta_p}^2 + 2.595\theta_p + 4.563$ $(0.3 \leqslant \theta_p \leqslant 1);\ \theta_p = \frac{c_{sp,surf}}{c_{sp,\max}}$	(16–3)
$U_n = -0.1493 + 0.8493\exp(-61.79\theta_n)$ $+0.3824\exp(-665\theta_n) - \exp(39.42\theta_n - 41.92)$ $-0.3131\arctan(25.59\theta_n - 4.099) - 0.009434\arctan$ $(32.49\theta_n - 15.74)(0 \leqslant \theta_n \leqslant 1)\ \theta_n = \frac{c_{sn,surf}}{c_{sn,\max}}$	(17–3)

electrode and graphite negative electrode, respectively.[29] Equations 16–3 and 17–3 are adopted for OCP at the positive and negative electrodes (LG INR 18650).[16]

Discretization.—For implementation reasons, the DFN model is spatially discretized and converted to differential algebraic equations (DAEs), a set of ordinary differential equations (ODEs) and algebraic equations (AEs) of the type:

$$\text{ODE system: } \frac{d\boldsymbol{y}}{dt} = \boldsymbol{f}(t, \boldsymbol{y}, z) \qquad [18]$$

$$\text{AE system: } \boldsymbol{g}(t, \boldsymbol{y}, z) = 0 \qquad [19]$$

where z represents the set of discretized AE variables, including the solid and liquid phase potential, y represents the set of discretized ODE variables, including the electrolyte and solid phase concentration, and t is the simulation time (note: the AE system is defined as a multivariate polynomial equation; ODE system is an equation involving ordinary derivatives of a function).

For implementation of the DFN model, conventional discretization methods, such as the finite difference method (FDM), finite volume method (FVM), and finite element method (FEM), have been widely adopted.[30,32] The FDM is the most direct approach to discretizing the PDE system. In the FDM framework, a point on the *x-axis* (or *r-axis*) represented by the continuum representation of the governing equations is converted into a set of discrete equations. The FVM divides the spatial axis domain in the PDE system into finite-sized elements of control volumes. The FVM is established by the integral conservation law on each of the control volumes where the flux entering a given volume is identical to leaving the adjacent volume. The FEM subdivides the PDE system into pieces of the finite-sized elements, so-called finite-elements mesh. After that, the FEM reconnects the elements at nodes defined as points at which lines of the elements intersect. The FEM is known to require quite sophisticated mathematics for its formulation.

The FDM is considered the easiest and most straightforward discretization method while producing an accurate performance on regular geometries (e.g., one-dimensional setting and rectangular-shaped models). In the FDM, for example, time and space domains of the DFN model are converted into time domains at each local point; the first and second order *x-* and *r-derivatives* of PDEs are converted into ODEs as follows (*see* Fig. 2):

$$\text{First order: } \frac{\partial Y_i(x, t)}{\partial x} = \frac{Y_{i+1}(t) - Y_{i-1}(t)}{h} \qquad [20\text{-}1]$$

$$\frac{\partial c_{i,j}(x, r)}{\partial r} = \frac{c_{i+1,j}(t) - c_{i-1,j}(t)}{h_r} \qquad [20\text{-}2]$$

$$\text{Second order: } \frac{\partial^2 Y_i(x, t)}{\partial x^2} = \frac{Y_{i-1}(t) - 2Y_i(t) + Y_{i+1}(t)}{h^2} \qquad [21\text{-}1]$$

$$\frac{\partial^2 c_{i,j}(x, t)}{\partial r^2} = \frac{c_{i-1,j}(t) - 2c_{i,j}(t) + c_{i+1,j}(t)}{{h_r}^2} \qquad [21\text{-}2]$$

where discretized values of variables along the *x-axis* is represented as Y_i (solid phase potential, electrolyte phase potential, and electrolyte concentration), discretized values of solid concentration is represented as $c_{j,i}$, the node spacing h is defined as $h = \frac{1}{N+1}$, the

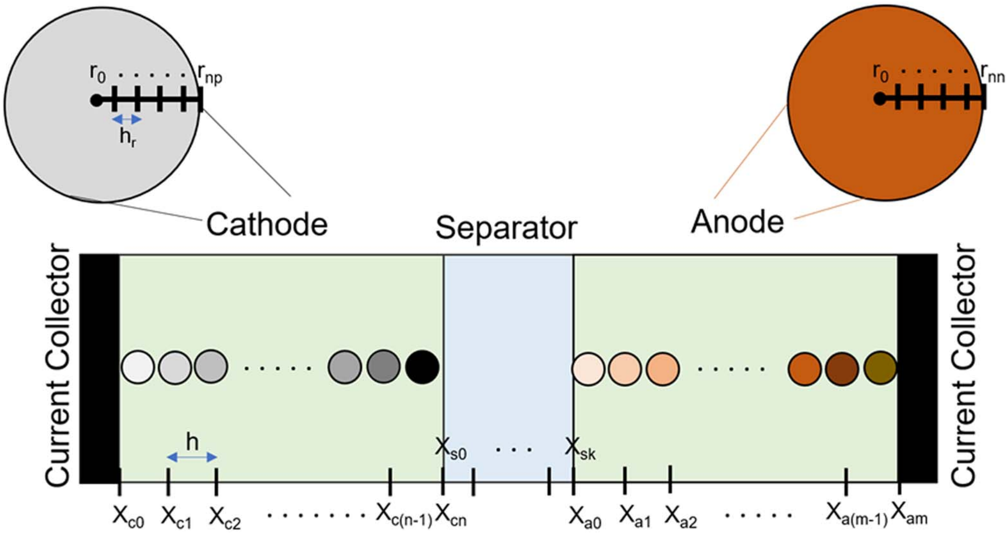

Figure 2. Finite difference discretization of the DFN model. The spatial coordinate position is described, using the one-dimensional coordinate. $X_{ci,}$ $X_{si,}$ $X_{ai,}$ and r_i represent the local node at cathode, separator, anode, and solid particle, respectively (cathode: i = 0…n, separator: i = 0…k, and anode: i = 0…m, solid particle at cathode: i = 0…np, and solid particle at anode: i = 0…nn). The number of node points is defined as discrete points in *x*- and *r*- domains (Number of node points for cathode: n, separator: k anode: m, and solid particle at cathode: np, and solid particle at anode: nn). The node spacing (h) represents a distance between the nearest two nodes.

Table VI. Parabolic profiles of solid particles.

$$\frac{\partial c_{sp,ave}}{\partial t} = -3\frac{j_p}{R_p} \qquad \frac{D_{s,p}}{R_p}(c_{sp,surf} - c_{sp,ave}) = -\frac{j_p}{5} \tag{22}$$

$$\frac{\partial c_{sn,ave}}{\partial t} = -3\frac{j_n}{R_n} \qquad \frac{D_{s,n}}{R_n}(c_{sn,surf} - c_{sn,ave}) = -\frac{j_n}{5} \tag{23}$$

spatial coordinate of *x-axis* and *r-axis* is expressed as *x* and *r*, respectively. *i* represents the node point along the *x-axis* (*i = 0…N*), and *j* represents the node points along the *r-axis* (*j = 0…N*); N is the number of node point.

Implementation issues with the DFN model.—High computational cost has been one of the critical issues with the DFN models.[16] Multiple DAEs are spatially created, which causes high computational demands in conventional numerical discretization methods.[17] Assume that 20 node points on the linear length scale are used to discretize the cathode, the separator, and the anode. In this case, the cathode generates 20 ODEs for the electrolyte concentration and a total of 40 AEs for the electrolyte and solid phase potential combined. In addition, assume that the solid particle is discretized with the number of node points 5. All solid particles in the cathode generates a total number of 100 ODEs. The solid particles are placed at each discretized point along with *x-axis* of the LIB (5 × 20). This results in a system of 160 DAEs for the cathode.

The anode is discretized in the same manner resulting in 160 DAEs. In the separator, using 20 node points will result in a total of 40 DAEs, consisting of 20 ODEs for the electrolyte concentration and 20 AEs for the electrolyte potential. Thus, the total number of DAEs of the full-order DFN model becomes 160 + 160 + 40 = 360. Given the large number of DAEs to be solved, simulation of the DFN model is computationally expensive. This has been typically considered one of the main obstacles to incorporate the DFN model into the advanced model-based BMS.[17,28]

Further, in the DAE system, ODEs are represented as governing equations (*see* Eq. 18), and AEs act as constraints (*see* Eq. 19) where consistent initial conditions (ICs) are defined to be initial values of $\boldsymbol{y_0}$ and $\boldsymbol{z_0}$ satisfying the AE.[33] In the DFN model, consistent initial conditions for all electrochemical variables are not known *a priori.*[33] For this reason, initial guesses are typically determined by the equilibrium states of LIBs. Initial guesses of solid-phase potentials at positive and negative electrodes are obtained from the values of negative and positive open-circuit potentials, respectively. Open-circuit potentials are a function of solid particle concentrations at the surface, and therefore, initial guesses of open-circuit potentials are determined by initial guesses of solid particle concentrations at the surface. The initial guesses of solid concentrations at the surface are given by the state of charge of the battery, and the initial guesses of average solid phase concentrations can be set up to be equal to the solid phase concentrations at the surface. However, when the current is applied to the battery (at time $t = 0^+$ from a situation of rest, $i_{app} = 0$ at $t = 0^-$) the battery exits from its equilibrium states, and the aforementioned initial guesses no longer satisfy the AE constraints.

Typically, standard solvers provided with commercial software include their own capabilities to identify consistent ICs for DAE systems.[34] For example, MATLAB® *ode15s* is known to simulate DAE systems with such inconsistent ICs.[34] In the DFN model, however, when high C-rates are applied to the battery, the difference between initial guesses based on the equilibrium status and consistent ICs can be considerable. In addition, when using high number of node points (note: the number of node points is defined in Fig. 2), conventional numerical discretization methods, including the FDM, FVM, and FEM, generate numerous numbers of discretized variables along with spatial scales. In this case, if inconsistent ICs are given, standard solvers might not identify the consistent ICs between the spatial electrochemical variables, leading to an unsuccessful simulation of the DFN model.[34] In the case of the DFN model, many standard solvers are known to cause lack of convergence. This usually happens under high C-rate operating conditions or at a high number of node points in conventional numerical discretization methods. In order to determine consistent ICs, researchers have typically implemented iterative-based initialization approaches (e.g., *fsolve* and bisection algorithms); however, these iterative algorithms often come with high computational cost or fail to obtain consistent ICs.[34]

Lastly, multiple parameters identification also remains a challenge in the DFN model.[35] The DFN model requires over 20 parameters to fully describe the physical, chemical, and electrochemical properties of LIBs.[31] In commercial LIBs, model parameter values are generally not provided because battery manufacturers treat them as trade secrets.[35] Multiple parameters in the DFN models, such as geometric, kinetic, and transport parameters, can be mathematically identified simultaneously from available measurements of voltage, current, and temperatures.[35] However, given the complexity of the DFN model and the large number of parameters, identifying multiple parameters often is not a simple research task unless very good initial guesses of the parameters are provided in advance. In addition, not all model parameters can be guaranteed to be identified uniquely and with satisfactory confidence.[35,36]

Ideally, the best way of identifying parameters is to conduct separate experiments. In a recent study, most parameters of the DFN model were experimentally obtained, and they were applied to the DFN model to show the fit to voltage data obtained by experiment measurements from a commercialized cell.[31] Geometric parameters of the DFN model include the electrode and separator thickness, particle radius, electrode, and separator porosity. The structure of two electrodes and a separator was obtained by scanning electron microscopy (SEM). Porosity and pore size were investigated with combinations of several experimental techniques including the Brunauer–Emmett–Teller theory, X-ray tomography, and SEM. Transport parameters involve diffusivity and conductivity of the solid phase and electrolyte. The solid phase diffusion coefficients were determined by observation of the change of SOC through the galvanostatic intermittent titration technique (GITT) from half-cell measurements. The GITT procedure consists of a series of current pulses, each followed by a relaxation time. It has been known to be difficult to directly obtain diffusivity and conductivity of the electrolyte after cell opening, due to its evaporation, oxidation, and reduction. The empirical relation between diffusivity/conductivity and electrolyte concentration was obtained from the literature.[37] In this way, diffusivity and conductivity were indirectly calculated. Kinetic parameters contain the activation energy and exchange current coefficient. Those parameters were investigated through electrochemical impedance spectroscopy measurements. In addition, the thermodynamic OCP and lithium stoichiometry in the electrode are obtained using the GITT.

However, these experimental protocols are a time-consuming destructive process. In addition, it is not straightforward to obtain all model parameters experimentally. In Ref. 31, some parameters are assumed or adjusted to be fitted with experimental voltage data, and other parameters are calculated by empirical formulas. More importantly, the multiple parameters identification techniques are essential to update the transport and kinetic parameters or any degradation inputs of LIBs within a very short span of time during battery operation.[37] During the LIB operation, capacity fade occurs, and therefore kinetic/transport parameters are continuously changed over cycles. To accurately predict the battery performance, model parameters must be updated by the adequate parameter identification technique. Any mismatch between model and experimental data over cycles of operation can be resolved with the updated parameters. Therefore, accurate identification of model parameters is a crucial technique to formulate precise physics-based models.

Current open-access codes for the DFN model.—From a user' perspective, the most efficient way of implementing DFN battery models is to adopt standard DAE solvers provided by prevalent commercial software. Among many other software, MATLAB® is the most popular control-oriented software; however, its own standard solvers typically present issues related to the initialization in conventional numerical discretization methods. For this reason, other numerical implementation techniques/software have been utilized. COMSOL® is one of the popular modeling software equipped with robust DAE solvers to implement and solve physics-based battery models. However, its expensive license fee is a barrier to collaboration and software sharing. In addition, it provides limited flexibility for different numerical discretization methods and does not allow to directly script parameter identification techniques without additional software.[16,38]

Several open-source battery modeling software packages offer alternative options; Examples include DUALFOIL,[39] FastDFN,[40] LIONSIMBA,[41] and PyBAMM.[42] DUALFOIL is written in FOTRAN, which has become a dead computational language. FastDFN and LIONSIMBA are MATLAB® based software packages, and PyBAMM is written in python language. FastDFN simulates the DFN model by directly deriving time-stepping analytical Jacobian based on the Newton's method. LIONSIMBA and PyBAMM import external solvers-e.g., the Implicit Differential-Algebraic (IDA) solver as part of SUNDIALS, which has been known to be one of the robust solvers today; however, it has been reported that these solvers also fail when inconsistent ICs are provided in extreme operating conditions.[41,43] In addition, these software packages lack the parameter identification routines required to allow for BMS implementation of physics-based battery models. Each of these packages implements the DFN model under a specific choice of operating condition based on model parameters obtained from the literature.

We address the consistent initialization problem by relying on a single-step iteration-free initialization approach developed in

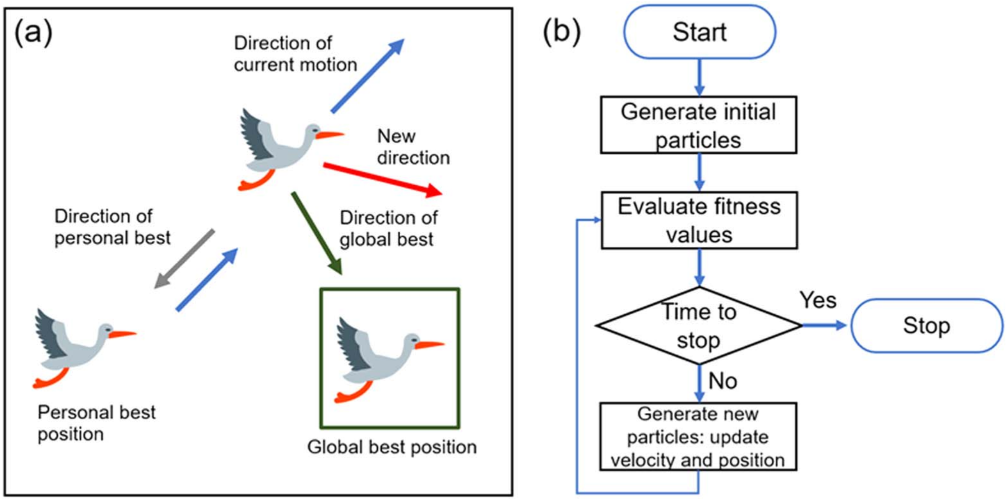

Figure 3. Particle swarm optimizaiton. (a) The particle swarm optimization algorithm (b) Flow chart of the particle swarm optimization. The particle swarm optimization algorithm was developed in 1995. In particle swarm optimizaiton, a solution is represented as a particle, and each particle include two vectors of position and velocity. Each particle moves to a new position using velocity based on the global best position in the PSO algorithm. Once a new position is reached, the best positions of each particle are updated.

Table VII. Comparison between DEARLIBS and implementation of *ode15s* without single-step iteration free initialization (NMC).

	Case I) Simulation results and computational time of the DEARLIBS (NMC)				
C-rate	Number of node point (anode, separator, cathode)				
	(2, 2, 2)	(5, 2, 5)	(10, 5, 10)	(20, 10, 20)	(30, 20, 30)
1C	0.49 s	1.68 s	2.37 s	40.25 s	46.58 s
2C	0.25 s	0.98 s	3.92 s	40.61 s	35.48 s
5C	0.23 s	1.43 s	2.54 s	31.22 s	29.42 s
8C	0.18 s	0.86 s	2.93 s	18.83 s	31.32 s
10C	0.22 s	0.53 s	1.88 s	21.38 s	23.10 s
	Case II) Simulation results and computational time of *ode15s implementation* (NMC)				
C-rate	Number of node point (anode, separator, cathode)				
	(2, 2, 2)	(5, 2, 5)	(10, 5, 10)	(20, 10, 20)	(30, 20, 30)
1C	0.50 s	0.61 s	2.54 s	Simulation Failure (not converging)	Simulation Failure (not converging)
2C	0.15 s	1.33 s	8.63 s		
5C	0.27 s	0.08 s	1.94 s		
8C	0.39 s	0.07 s	Simulation Failure (not converging)		
10C	0.19 s	0.08 s			

Refs. 33 and 44 based on a symbolic software environment. A symbolic environment enables direct and efficient derivation of implicit ODEs from algebraic equations, which is an essential step for the iteration-free initialization (*see the detailed process in the DEARLIBS implementation section*).[33] To date, the DFN model has been successfully implemented with such an initialization approach in Maple®.[29,45] We perform the single-step iteration-free initialization approach in the MATLAB® environment by adopting the MATLAB® symbolic toolbox and simulate the DFN Model with the *ode15s* solver. Moreover, a closed-loop optimization-based identification method is provided that allows for the selection of model parameters upon voltage and current data. To the best of the authors' knowledge, this is the first time that implementation of the DFN models using the MATLAB® *ode15s* solver has been documented along with a closed-loop identification routine. The single-step iteration-free initialization method implemented in the DEARLIBS was developed and implemented in MAPLE environment in Ref. 33 and related patent.[44]

The reader must be reminded that the DEARLIBS is intended solely for academic purposes, and an academic use only license is available at https://github.com/DEARLIBS. Commercial use of DEARLIBS might infringe the original patent of the single-step iteration-free initialization.

DEARLIBS.—In this paper, we provide the **DEARLIBS** (**D**oyle-Fuller-Newman **E**lectrochemical **B**attery Model Implementation in **R**obust and Sleek MATLAB® Framework for **Li**thium-ion **B**atterie**s**). The objective of DEARLIBS is to provide a MATLAB® framework for robust and sleek implementation of physics-based battery model, where the DFN model is adopted as an example (note: "robust" in this context refers to the ability of the solver to generate a successful simulation without a crash- e.g., even under extreme fast charging and/or dynamic battery operation). The DEARLIBS consists of DAEs after numerical discretization, a standard MATLAB® solver, and a parameter identification routine with voltage data from experimental measurements, as shown in Algorithm 1. The DEARLIBS adopts the finite difference method, over which the single-step iteration-free initialization approach is applied. The MATLAB® standard *ode15s* solver is then adopted as the solver of choice to solve the problem. The robustness of the DEARLIBS allows to simulate the DFN model under extreme operating conditions as well as couple parameter identification techniques because the DEARLIBS includes the single-step iteration-free initialization approach.

The DEARLIBS will enable users' straightforward access to such a computational framework, thereby addressing the aforementioned implementation issues of the DFN model for use in the next-generation BMS. For example, users can easily replace DAEs on Line 1 of the pseudo code in Algorithm 1 with control-oriented alternative models such as FHM for LIBs. Different discretized DAE equations derived from various numerical discretization methods, such as FVM, FEM, and the orthogonal collocation can also be implemented on Line 1 in the pseudo code. For parameter identification, the DEARLIBS allows users to continuously identify model parameters with different optimizers to arrive at robust optimization approaches. Different optimization routines can be attempted from Line 5 of the pseudo code.

Paper structure.—This paper consists of four sections: the DEARLIBS implementation, Fast charging simulation, Parameter identification, and Conclusion and Perspective sections. In the DEARLIBS implementation section, the single-step iteration-free initialization approach and parameter identification technique are described in detail. The DEARLIBS possesses two strengths; (i) successful simulation under fast-charging operating conditions and (ii) straightforward and robust implementation of parameter identification routines. In the fast-charging simulation section, we demonstrate the DEARLIBS performance over high C-rates. Model parameters of the $LiCoO_2$ cathode/graphite anode chemistry and NMC/graphite 18650 cylindrical battery systems were tested. The objective of the Parameter identification section is to show the parameter identification performance of the DEARLIBS. The experimental voltage profile at the constant C-rate is used to identify model parameters, and the model validation is confirmed by comparing model voltage outputs with voltage data from experimental measurements using the urban dynamic driving schedule (UDDS). Geometric model parameters were adopted from Ref. 31 and the particle swarm optimization (PSO) was integrated into the DFN model on the DEARLIBS to mathematically identify kinetic/transport parameters.[16]

DEARLIBS Implementation

This section describes details of the DEARLIB pseudo code presented in the Introduction section. The DEARLIBS was carried out on a computing system with Intel Xeon Gold 6136 CPU @ 3.00 GHz, 12 Core(s), 24 Logical Processors, using the global optimization and symbolic math toolbox of the MATLAB® software.

Line 1 in Algorithm 1: equations.—*For the macro-scale system*, governing Eqs. 1, 2, 3, 5, 6, 7, 8 and 9 in Table I are adopted. Next,

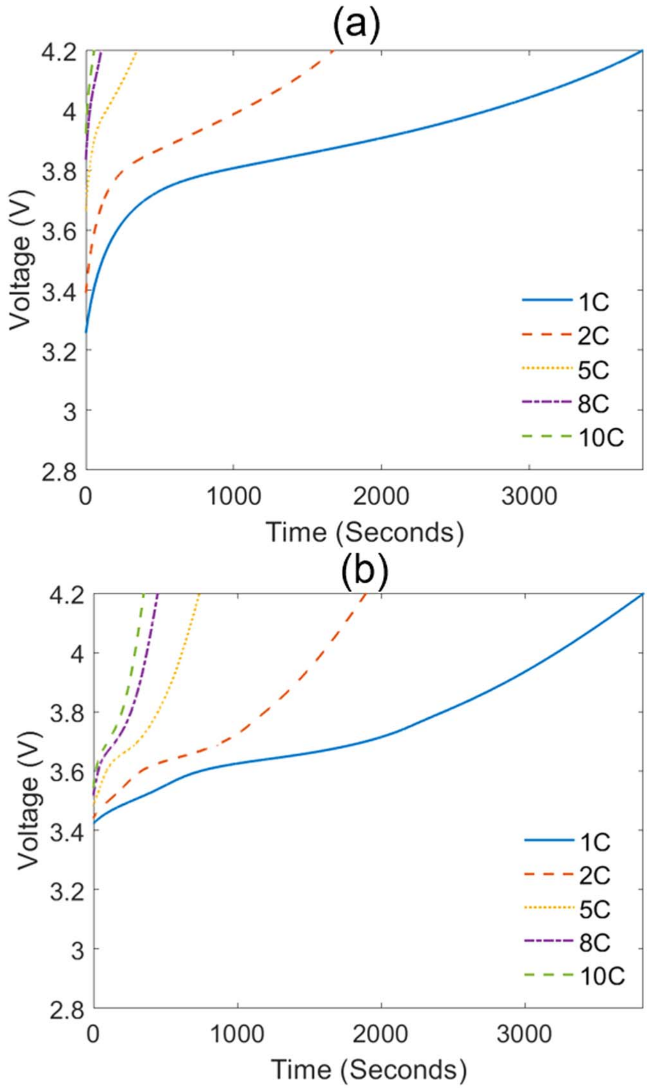

Figure 4. Robust and Sleek MATLAB® Implementation under fast charging conditions when the number of node point is (20,10,20). (a) $LiCoO_2$ cathode and graphite anode battery system (b) NMC/graphite 18650 cylindrical battery system. A blue-color solid line, orange-color dashed line, yellow-color dotted line, purple-color dash-dot line, and green-color dashed line represent simulation results of 1C, 2C, 5C, 8C, and 10C, respectively.

finite difference Eqs. 20 and 21 are inserted into the governing equations to obtain the DAE system. The DAEs are directly coded in a symbolic form. *For the micro-scale system*, the concentration of solid particles, which is approximated by parabolic profiles (Equations 22 and 23 in Table VI), is implemented by replacing Eqs. 4 and 10 in Table III.[18] The solid phase concentration is represented by the solid particle concentration at the surface ($c_{si,surf}$) and solid particle average concentration ($c_{si,ave}$), rather than numerically discretizing solid particles along with a radial axis ($i = p, n$). This type of volume-averaging combined with the parabolic profiles, and solid concentration profiles are expressed as ODEs, as shown in Table VI. Equations 22 and 23 are directly incorporated into Line 1 in the ODE symbolic form. Equations 11–17 are coded in a symbolic form as well.

Line 2 in Algorithm 1: Single-step iteration-free initialization.—In the single-step iteration-free initialization method, the DAE system is separated into AEs and ODEs, and AEs are converted into implicit ODEs by the perturbation method.[46] Next, a switch function is multiplied by the original ODEs in the DAE system. Lastly, the original ODEs including the switch function and implicit ODEs are simultaneously simulated with a MATLAB® standard *ode15s* solver.[33] During a solving process, while implicit ODE systems identify the consistent ICs, the switch function converts the original ODEs into zero. Once consistent ICs are determined, the switch function is set to one, meaning that the ODEs are reverted to the original equations. In this way, consistent ICs are guaranteed with a single-step iteration-free process. This section describes the detailed driving process of the single-step iteration-free initialization.[33,46]

First, we show a process of AEs being converted into implicit ODEs. Here, $\boldsymbol{g}(t') = 0$ represents a set of AEs, as a function of time. When a perturbation parameter (ε) is a very small constant ($\ll 1$), $\lim_{\varepsilon \to 0} \boldsymbol{g}(t' + \varepsilon)$ becomes zero as follows:

$$\boldsymbol{g}(t') = \lim_{\varepsilon \to 0} \boldsymbol{g}(t' + \varepsilon) = 0 \tag{24}$$

When $t' = t + \varepsilon$, Eq. 24 becomes

$$\boldsymbol{g}(t + \varepsilon) = \lim_{\varepsilon \to 0} \boldsymbol{g}(t' + 2\varepsilon) = 0 \tag{25}$$

From Eq. 25,

$$\boldsymbol{g}(t + \varepsilon) = 0 \tag{26}$$

Combining Eq. 26 with Eq. 24 produces the following:

$$\boldsymbol{g}(t') = \boldsymbol{g}(t + \varepsilon) = 0 \tag{27}$$

The original AEs to be solved is $\boldsymbol{g}(t') = 0$. From Eq. 27, however, solving $g(t + \varepsilon) = 0$ in confirmed to be equal to solving $\boldsymbol{g}(t') = 0$. By the Taylor series expansion,

$$\boldsymbol{g}(t + \varepsilon) = \boldsymbol{g}(t) + \varepsilon \frac{d\boldsymbol{g}(t)}{dt} + \frac{1}{2}\varepsilon^2 \frac{d^2\boldsymbol{g}(t)}{dt^2} + \ldots \approx \boldsymbol{g}(t) + \varepsilon \frac{d\boldsymbol{g}(t)}{dt} \tag{28}$$

Combining Eq. 27 with Eq. 28, leads to

$$\varepsilon \frac{d\boldsymbol{g}(t)}{dt} - \boldsymbol{g}(t) = 0 \tag{29}$$

or

$$\varepsilon \frac{d\boldsymbol{g}(t)}{dt} = -\boldsymbol{g}(t) \tag{30}$$

We have demonstrated that solving the original set of AEs ($\boldsymbol{g}(t) = 0$) equals to solving $\varepsilon \frac{d\boldsymbol{g}(t)}{dt} = -\boldsymbol{g}(t)$ from Eqs. 26, 27, and 30.

Next, we apply a switch function to a set of ODEs. Here, $y(t)$ represents the set of differential variables, and $\boldsymbol{f}(t)$ represents a set of functions and Eq. 31 represents the ODE systems

$$\frac{d\boldsymbol{y}(t)}{dt} = \boldsymbol{f}(t) \tag{31}$$

Now, a switch function, T_H, is introduced

$$T_H = \frac{1}{2}(1 + tanh(q(t - t_j))) \tag{32}$$

where q is a weighting factor and t_j is the time allowed for the perturbation approach to find consistent ICs. If q has enough large values (e.g., 1000), the switch function becomes zero when $t < t_j$. When $t > t_j$ the switch function becomes one. Now, when Eq. 33 is applied to Eq. 34 it leads to:

Table VIII. Geometric, operational, and electrochemical parameters.

Symbol	Parameter	Values	Units
Geometrical parameters			
l_p	Thickness of positive electrode	7.56×10^{-5}	m
l_s	Thickness of separator	1.2×10^{-5}	m
l_n	Thickness of negative electrode	8.52×10^{-5}	m
R_p	Particle radius at positive electrode	5.22×10^{-6}	m
R_n	Particle radius at negative electrode	5.86×10^{-6}	m
A_{cell}	Electrode area	0.11	m^2
ε_p	Porosity at positive electrode	0.35	—
ε_s	Porosity at separator	0.47	—
ε_n	Porosity at negative electrode	0.25	—
Operational parameters			
i_{app}	Applied Current	1C: 5	A
Electrochemical parameters			
F	Faraday constant	96485	sAmol^{-1}
R	Ideal gas constant	8.314	$\mathrm{JK}^{-1}\mathrm{mol}^{-1}$
T	Temperature	278.5, 298.5, 308.5	K

$$\frac{dy(t)}{dt} = \boldsymbol{f}(t, y, z)\frac{1}{2}(1 + tanh(q(t - t_j))) \qquad [33]$$

In Line 2 (*see* Algorithm 1), Eqs. 30 and 33 are coded based on DAE system implemented in Line 1. That is,

(i) When $t < t_j$, $\varepsilon\frac{dg(t)}{dt} = -\boldsymbol{g}(t)$ will find consistent initial conditions.
(ii) When $t > t_j$, $\frac{dy(t)}{dt} = \boldsymbol{f}(t)$ and $\varepsilon\frac{dg(t)}{dt} = -\boldsymbol{g}(t)$ are solved simultaneously.

The single-step iteration-free initialization approach was originally developed in a symbolic software environment. The symbolic environment enables to efficiently derive Eq. 30, which is an essential step in the approach. In the DFN model, for example, governing Eqs. 2 and 8 in Table III are converted to Eqs. 34 and 35 as AEs after the FDM discretization, as follows:

$$-\frac{\sigma_{eff,p}}{2}\frac{\Phi_{1,\ i+1}(t) - \Phi_{1,i-1}(t)}{h} - \frac{\kappa_{eff,p}}{2}\frac{\Phi_{2,i+1}(t) - \Phi_{2,i-1}(t)}{h} + \frac{2\kappa_{eff,p}RT(1 - t_+)}{Fh}\frac{c_{i+1}(t) - c_{i-1}(t)}{c_i(t)} - I = 0 \qquad [34]$$

$$-\frac{\sigma_{eff,n}}{2}\frac{\Phi_{1,\ i+1}(t) - \Phi_{1,i-1}(t)}{h} - \frac{\kappa_{eff,n}}{2}\frac{\Phi_{2,i+1}(t) - \Phi_{2,i-1}(t)}{h} + \frac{2\kappa_{eff,n}RT(1 - t_+)}{Fh}\frac{c_{i+1}(t) - c_{i-1}(t)}{c_i(t)} - I = 0 \qquad [35]$$

The non-linear third term of Eq. 34 $\left(\frac{2\kappa_{eff,p}RT(1 - t_+)}{Fh}\frac{c_{i+1}(t) - c_{i-1}(t)}{c_i(t)}\right)$ and 35 $\left(\frac{2\kappa_{eff,n}RT(1 - t_+)}{Fh}\frac{c_{i+1}(t) - c_{i-1}(t)}{c_i(t)}\right)$ can impose restrictions on the direct derivation of implicit ODEs (Eq. 30), either manually or using the non-symbolic format of MATLAB®. The direct conversion of AEs to implicit ODE systems can be efficiently achieved with the help of a symbolic environment.

Line 3 in Algorithm 1: Odefunction.—The *odeFunction* converts a system of symbolic first-order DAEs to a MATLAB® function handle acceptable as input arguments to the numerical MATLAB® DAE solver *ode15s*.[47]

Line 4 in Algorithm 1: Initial guesses.—Initial guesses based on the equilibrium status as initial conditions.

Line 5 in Algorithm 1: Parameter identification.—In this part, simultaneous identification of multiple parameters is carried out, adopting experimental voltage profiles of constant current-discharge. The model voltage outputs are fitted with the voltage data by experimental measurements to identify parameter values, using evolution optimization techniques.

In the DEARLIBS, the PSO algorithm is implemented (The default PSO setting: the maximum iteration number is 50 and the particle size is 30). In the PSO, the optimal value of the objective function is identified by an iterative process that allows a candidate solution called particles to reach a better objective function value. These particles include the objective function value and velocity. In the first iteration, all particles involve an objective function value consisting of random parameters within the lower and upper bound, and the PSO identifies the particle representing the best objective function value among these particles. In the next iteration, the other particles change their velocity direction to be directed towards the position of the particle including the best objective function value in the first iteration. The same process is repeated until the optimal objective function is identified (*see* Fig. 3).

Various parameter identification techniques are currently used to identify the parameters of LIBs, including Kalman filter methods, recursive least squares methods, non-recursive least squares methods, and evolutionary computation methods.[38,48,49] Among many other identification techniques, the PSO algorithm has several advantages; (i) It does not require the optimization problem to be differentiable, as required by classic optimization methods such as gradient descent and quasi-newton methods. (ii) It can therefore also be used on optimization problems that are not even continuous, noisy, and change over time. The optimization routine finds the optimal values for which the objective function subject to the model equations, initial conditions, and bounds for the parameters, is minimized. The final identified parameters are obtained by solving an optimization problem where the sum of squares of the differences in the voltage outputs between the model and experiment divided by the total number of experimental data points of the system is minimized inn the Root Mean Square Error sense, and unknown parameters are used as decision variables.

Line 6 and 7.—The *ode15s* solver is adopted.

Line 8.—Voltage data by experimental measurement is embedded.

Line 9.—The objective function of the optimization approach is coded as follows:

$$Obj = \text{Min}\ \frac{1}{N}\sum_{j=1}^{N}\sqrt{(V_{exp,j} - V_{model,j}(\boldsymbol{p}))^2} \quad [36]$$

$$s.t.\ \boldsymbol{p}^L < \boldsymbol{p} < \boldsymbol{p}^U$$

where N is the total number of experimental data points for charging and discharging, n_p is the number of estimating parameters, $V_{exp,j}$ and $V_{model,j(\boldsymbol{p})}$ indicate the experimental and model predicted voltage value for the *jth* data point, $\boldsymbol{p}$ is the vector of the estimating parameters, and $\boldsymbol{p}^L$ and $\boldsymbol{p}^U$ represent the lower and upper bounds for the vector of the estimating parameters (p).

Fast Charging Simulation

One of the most critical issues that still prevents the spread of EVs is the long charging time. However, during the fast-charging process, issues with battery life, efficiency, and safety can occur by unwanted side reactions and consequent thermal runaway. To address these issues, material scientists have worked on developing new materials to accommodate fast charging, and control engineers have designed charging optimization and thermal management strategies. The DEARLIBS can provide insights to understand the chemical/kinetic phenomena without physical destruction of the battery system. This section shows the robustness of the DEARLIBS under fast charging operation. Because DEARLIBS is a flexible physics-based battery modeling framework, once more accurate alternative modeling tools are implemented, they can be used simultaneously to provide an insight into battery design development and to develop optimal control techniques.

In this section, two lithium-ion battery cells ($LiCoO_2$ cathode/graphite anode and NMC/graphite 18650 cylindrical battery systems) are simulated by adopting model parameters from Refs. 15 and 30. Model parameters of the NMC/graphite 18650 cylindrical battery system are obtained from our previous studies,[16] and these parameters were confirmed to be validated with experimental data up to 15C. Model parameters of the $LiCoO_2$/graphite battery system have been widely used in the literature when new algorithms and codes are developed.[40,42] The OCPs Eqs. 16–2, 17–2 and 16–3, 17–3 in Table V are used for two battery systems, respectively.

To the best of the authors' knowledge, there have been no attempts to qualitatively define stiff operating conditions in the DFN model. In the DFN model, reasons for simulation failure are: 1) high C-rates and 2) high the number of node points. In this section, the stiff operating condition is addressed by employing the *ode15s* solver without the single-step integration-free initialization approach (*see Case II: Simulation results and computational time of ode15s (NMC) in* Table VII). Simulations were conducted for the NMC/graphite 18650 cylindrical battery system. The *ode15s* solver simulates the DFN model well from 1C to 5C rates for low number of node points (∼ (10, 5, 10)). However, within the same C-rate range, when the number of node points increases and reaches (20, 10, 20), the solver fails. At 8C and 10C rates, the DFN model does not converge even for the case of number of node points equal to (10, 5, 10). High number of node points while operating the battery at high C-rates should be successfully simulated, to properly reproduce fast charging simulation scenarios.

To investigate the performance of the DEARLIBS, Lines 1–4 and 6 in Algorithm 1 are implemented without parameter identification routines presented in Line 5, 7, 8, 9 of Algorithm 1, and the *ode 15 s* solver is directly utilized to solve the DFN model after numerical discretization and single-step iteration-free initialization. The proposed DEARLIBS successfully simulates the DFN model over the entire C-rate range (up to 10C), as shown in Fig. 4. Table VII shows the computational time as a function of C-rate and grid points (*see Case I: Simulation results and computational time of the DEARLIBS (NMC) in* Table VII).

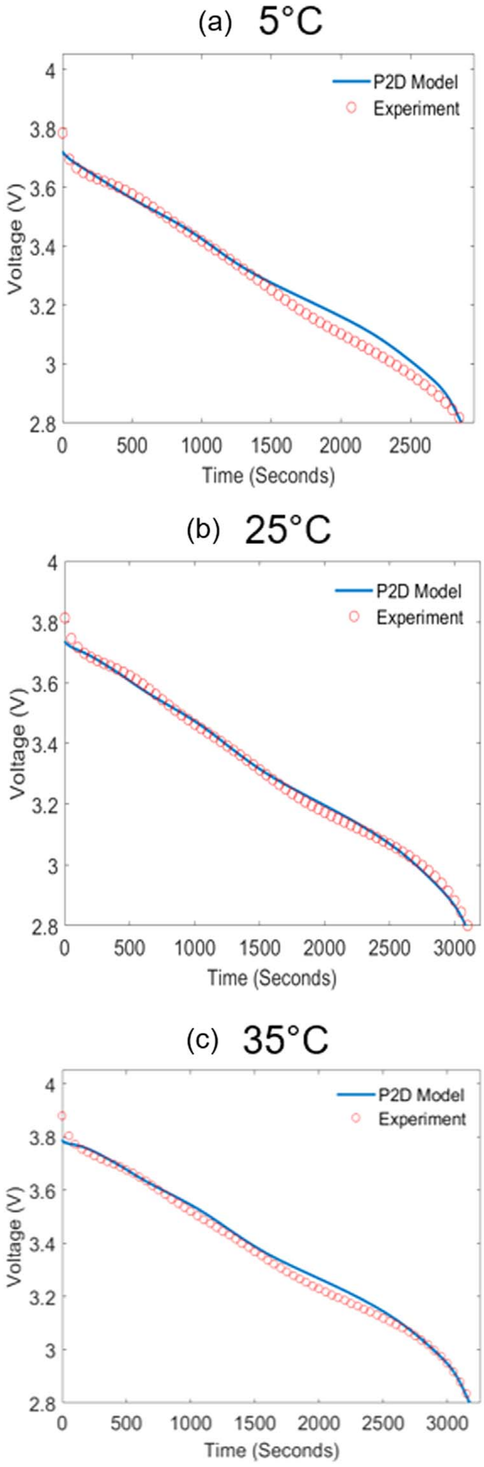

Figure 5. Comparison of voltage profiles between experimental data and battery model outputs from the DFN model at 1C constant discharge condition. Voltage profiles from experimental measurement and DFN model outputs are compared at (a) 5 °C, (b) 25 °C, and (c) 35 °C (Experimental data: empty circle dots, the P2D model: blue-color straight line).

Parameter Identification

In this section, we demonstrate that the proposed DEARLIBS can couple parameter identification routines with the DFN model. All experimental results are based on a commercially available cylindrical cell manufactured by LG Chem (model: LG M50 INR 21700 5000 mAh).[50] The cell voltage was obtained using the Arbin battery

Table IX. Identified Kinetic/Transport Parameters at Different Temperatures.

Symbol	Parameter	Temp.	Lower bound	Upper bound	Initial guess	Final value (Units)
D	Electrolyte diffusion coefficient	5 °C	0.42×10^{-9}	0.78×10^{-9}	0.6×10^{-9}	0.71×10^{-9}
		25 °C	0.7×10^{-9}	1.3×10^{-9}	1×10^{-9}	0.7×10^{-9} (m^2 s^{-1})
		35 °C	0.84×10^{-9}	1.56×10^{-9}	1.2×10^{-9}	1.12×10^{-9}
κ	Electrolyte conductivity	5 °C				1.51
		25 °C	0.82	1.52	1.17	1.33 (S m^{-1})
		35 °C				1.35
σ_p	Solid phase conductivity at positive	5 °C				0.13
		25 °C	0.13	0.23	0.18	0.15 (S m^{-1})
		35 °C				0.23
σ_n	Solid phase conductivity at negative	5 °C				157
		25 °C	150.5	279.5	215	279.5 (S m^{-1})
		35 °C				279.5
$D_{s,p}$	Solid phase diffusivity at positive	5 °C	2.8×10^{-15}	5.2×10^{-15}	4×10^{-15}	3.6×10^{-15}
		25 °C	2.8×10^{-15}	5.2×10^{-15}	4×10^{-15}	4.9×10^{-15} (m^2 s^{-1})
		35 °C	4.55×10^{-15}	8.45×10^{-15}	6.5×10^{-15}	5.5×10^{-15}
$D_{s,n}$	Solid phase diffusivity at negative	5 °C	0.56×10^{-14}	1.04×10^{-14}	0.8×10^{-14}	0.78×10^{-14}
		25 °C	2.31×10^{-14}	5.94×10^{-14}	3.3×10^{-14}	4.3×10^{-14} (m^2 s^{-1})
		35 °C	4.2×10^{-14}	10.8×10^{-14}	6×10^{-14}	7.3×10^{-14}
k_p	Reaction rate constant at positive	5 °C	0.14×10^{-11}	0.26×10^{-11}	0.2×10^{-11}	0.19×10^{-11}
		25 °C	0.49×10^{-11}	0.91×10^{-11}	0.7×10^{-11}	0.8×10^{-11} mol/(m^2·s)
		35 °C	0.91×10^{-11}	1.69×10^{-11}	1.3×10^{-11}	1×10^{-11}
k_n	Reaction rate constant at negative	5 °C	0.63×10^{-12}	1.17×10^{-12}	0.9×10^{-12}	1.02×10^{-12}
		25 °C	0.49×10^{-12}	0.91×10^{-12}	0.7×10^{-12}	0.5×10^{-12} mol/(m^2·s)
		35 °C	0.91×10^{-12}	1.69×10^{-12}	1.3×10^{-12}	1.19×10^{-12}

Table X. Root Mean Square Errors of voltage profiles at 1C between Different Number of Node points.

	(2, 2, 2)	(5, 5, 5)	(10, 5, 10)	(20, 10, 20)
5 °C	19 mV	16 mV	17 mV	30 mV
25 °C	21 mV	17.5 mV	18 mV	16 mV
35 °C	23 mV	20 mV	20.5 mV	20 mV

testing system equipped with highly accurate sensors capable of measuring the load current and cell voltage (Model: Arbin LBT21024).[51] A thermal chamber (Model: Amerex IC500R Thermal Chamber) was used to conduct temperature-controlled experiments.[52] The battery cell was charged and discharged in a thermal chamber at different temperatures (5 °C, 25 °C, and 35 °C).

Voltage measurements were acquired from CC discharging and dynamics current profiles: First, a CC-CV profile (CC: Constant Current, CV: Constant Voltage) was applied to the cell to reach full charge capacity. Then, the battery cell was charged at 1C, and once the cell voltage reached 4.2 V, it was charged with the CV until the applied current decreases to C/10 (note: a C-rate of 1C is defined as the applied current when the battery is charged for one hour). After CC-CV charging, the cell was at rest for 30 min. The next experimental procedure is described as follows:

(i) For constant discharge, C/20 was applied to the cell until it reaches 2.8 V.

(ii) For dynamic discharging, the battery was discharged at C/5 until it reaches SOC 85% (SOC was calculated by Arbin battery testing instrument). After that, the UDDS was applied to the cell until the cell voltage value reaches 2.8 V.

The geometric parameters of LG M50 INR 21700 are presented in Table VIII.[31] Using this information, the remaining eight kinetic and transport parameters are identified, adopting the PSO algorithm. The identified eight kinetic/transport parameters are the electrolyte diffusion coefficient, electrolyte conductivity, solid phase conductivity at the positive and negative electrode, solid phase diffusivity at the positive and negative electrode, and reaction rate constants at the positive and negative electrode. Parameter identification has been simultaneously carried out using voltage data by experimental measurement of discharging (1C) at 5 °C, 25 °C, and 35 °C. Also, for the best performance of the optimization problem, the diffusion and kinetic rate constants are expressed in term of a decimal function ($D = 10^{-A}$, $D_{s,p} = 10^{-B}$, $D_{s,n} = 10^{-C}$, $k_p = 10^{-D}$ and $k_n = 10^{-E}$), and indices (A, B, C, D, and E) of the decimal function are used as optimizing variables since the original parameters (diffusivity and reference rate constants) are too small ($\sim 10^{-15}$) to be identified properly.

Initial guess and upper/lower bounds.—Initial guesses were set up based on Ref. 31 and lower and upper bounds were adjusted based on the initial guesses (±30% from initial guesses) to obtain converged values of the unknown parameters while minimizing the RMSE. The initial guesses for the electrolyte diffusion coefficient, the electrolyte conductivity, solid phase conductivity at the positive and negative electrodes, the solid phase diffusivity at the positive and negative electrode, and the reaction rate at the positive and negative electrode were set up as follows (*see* Table IX): At 5 °C, initial guesses are determined by $0.6 \times 10^{-9} m^2 s^{-1}$, 1.17 S m^{-1}, 0.18 S m^{-1}, 215 S m^{-1}, $4 \times 10^{-15} m^2 s^{-1}$, $0.8 \times 10^{-14} m^2 s^{-1}$, 0.2×10^{-11}mol (m^{-2}·s), and 0.9×10^{-12}mol (m^{-2}·s), respectively. At 25 °C, initial guesses are set up to $1 \times 10^{-9} m^2 s^{-1}$, 1.17 S m^{-1}, $0.18 m^2 s^{-1}$, 215 S m^{-1}, $4.0 \times 10^{-15} m^2 s^{-1}$, $3.3 \times 10^{-14} m^2 s^{-1}$, 0.7×10^{-11}mol (m^{-2}·s), and 0.7×10^{-12}mol (m^{-2}·s), respectively. At 35 °C, initial guesses are $1.2 \times 10^{-9} m^2 s^{-1}$, 1.17 S m^{-1}, 0.18 S m^{-1}, 215 S m^{-1}, $6.5 \times 10^{-15} m^2 s^{-1}$, $6.0 \times 10^{-14} m^2 s^{-1}$, 1.3×10^{-11}mol/(m^2·s), and 1.3×10^{-12}mol/(m^2·s), respectively. The lower and upper bounds for kinetic/transport parameters were given as ±30% from the initial guesses.

Identified parameters.—The converged parameters for the electrolyte diffusion coefficient, the electrolyte conductivity, solid phase conductivity at the positive and negative electrodes, the solid phase

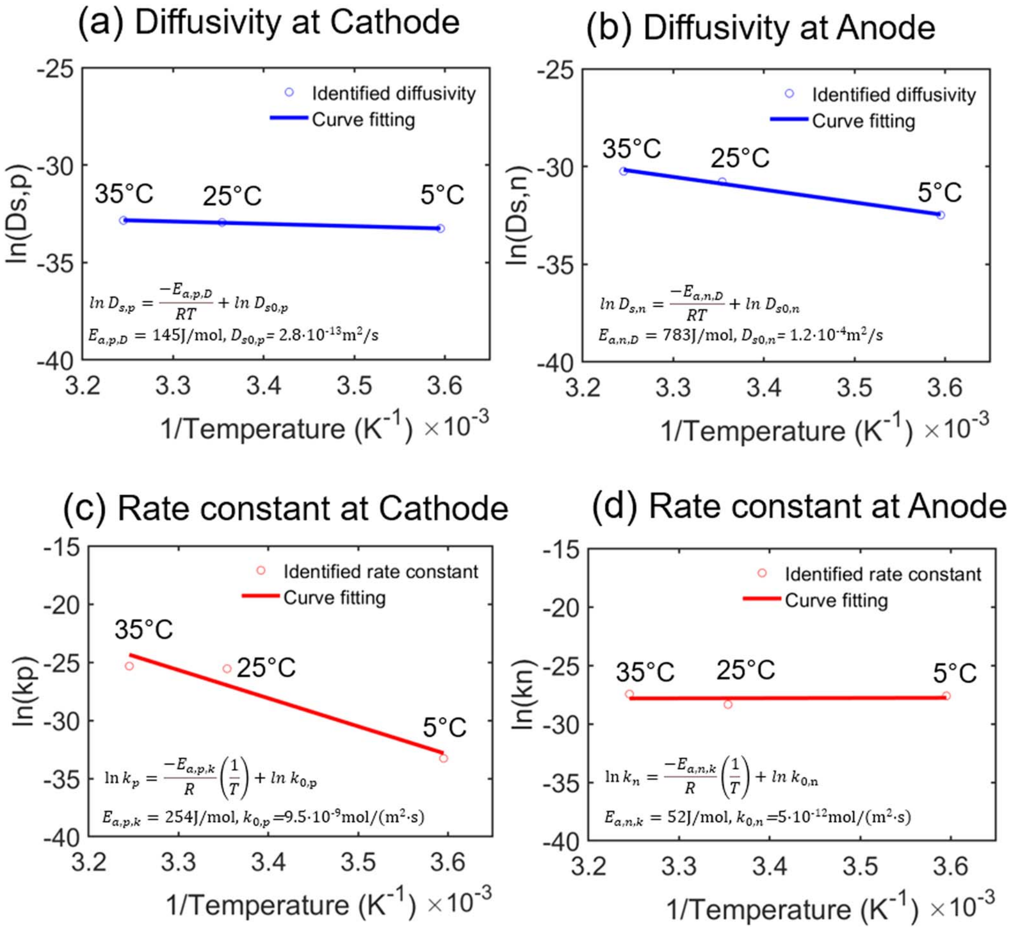

Figure 6. Arrhenius relationships between identified solid phase diffusion coefficients and rate constants at different temperatures. The identified parameters in this work exhibit the Arrhenius relationships.

diffusivity at the positive and negative electrode, and the reaction rate at the positive and negative electrode are as follows (*see* Table IX). At 5 °C, identified values are 0.71 × 10^{-9}m^2 s^{-1}, 1.51 S m^{-1}, 0.13 S m^{-1}, 157 S m^{-1}, 3.6 × 10^{-15}m^2 s^{-1}, 0.78 × 10^{-14}m^2 s^{-1}, 0.19 × 10^{-11}mol (m^{-2}·s), and 1.02 × 10^{-12}mol (m^{-2}·s), respectively. At 25 °C, identified parameters are confirmed as 0.7 × 10^{-9}m^2 s^{-1}, 1.33 S m^{-1}, 0.15 S m^{-1}, 279 S m^{-1}, 4.9 × 10^{-15}m^2 s^{-1}, 4.3 × 10^{-14}m^2 s^{-1}, 0.8 × 10^{-11}mol/(m^2·s), and 0.5 × 10^{-12}mol/(m^2·s), respectively. At 35 °C, converged parameters are determined as 1.12 × 10^{-9}m^2 s^{-1}, 1.35 S m^{-1}, 0.23 S m^{-1}, 279 S m^{-1}, 5.46 × 10^{-15}m^2 s^{-1}, 7.34 × 10^{-14}m^2 s^{-1}, 1 × 10^{-11}mol/(m^2·s), and 1.19 × 10^{-12}mol/(m^2·s), respectively.

Performance of the DFN model.—In this parameter identification test, we used 20 node points for the negative electrode, 10 node points for the separator, and 20 node points for the positive electrode (notation: (cathode, separator, anode) = (20, 10, 20)). The minimized RMSEs were 30 mV, 16 mV, and 20 mV at 5 °C, 25 °C, and 35 °C, respectively, as shown in Fig. 5. The RMSE was investigated for different numbers of node points (*see* Table X). The general trend is that the RMSE decreases while the number of node points increases. In most cases, the RMSE was decreased with higher numbers of node points and converged at (20, 10, 20). However, the RMSE of (20, 10, 20) at 5 °C was confirmed to be a higher value compared to the lower number of node points. One of the possible reasons can be the stiffness characteristic of simulations at 5 °C. During the parameter identification process, different sets of parameters are attempted repeatedly to fit model outputs to the experimental voltage profiles, and the simulation might not be converged to produce better RMSE values. This will be investigated further as part of future work.

Arrhenius relationship.—The Arrhenius equation, a formula for the temperature dependence of reaction rates, describes the dependence of the rate constant of a chemical reaction on the absolute temperature, the reference reaction rate constant and other constants as following:

$$k_i = k_{0,i}e^{-E_{a,i,k}/RT} \tag{37}$$

where k_i is the reaction rate constant, T is the absolute temperature, $k_{0,i}$ is the reference reaction rate constant, $E_{a,i,k}$ is the activation energy for the reaction, i is the positive and negative electrode, and R is the gas constant. Taking logarithms of both right- and left sides yields Eq. 30 as shown in below:

$$\ln k_i = \frac{-E_{a,i,k}}{R}\left(\frac{1}{T}\right) + ln\ k_{0,i} \tag{38}$$

In Eq. 38, the Arrhenius Eq. 37 is converted into the linear equation whose slope is $\frac{-E_{a,i,k}}{R}$ when ln k_i is plotted as a function of $\left(\frac{1}{T}\right)$. Likewise, the solid phase diffusion coefficient can also be expressed as Arrhenius relationships (*see* Eq. 39).

$$D_{s,i} = D_{s0,i}e^{-E_{a,i,D}/RT} \tag{39}$$

where $D_{s,i}$ is the solid phase diffusion coefficient, $D_{s0,i}$ is the reference solid phase diffusion coefficient, $E_{a,i,D}$ is the activation energy for the diffusion, T is the absolute temperature, i is the

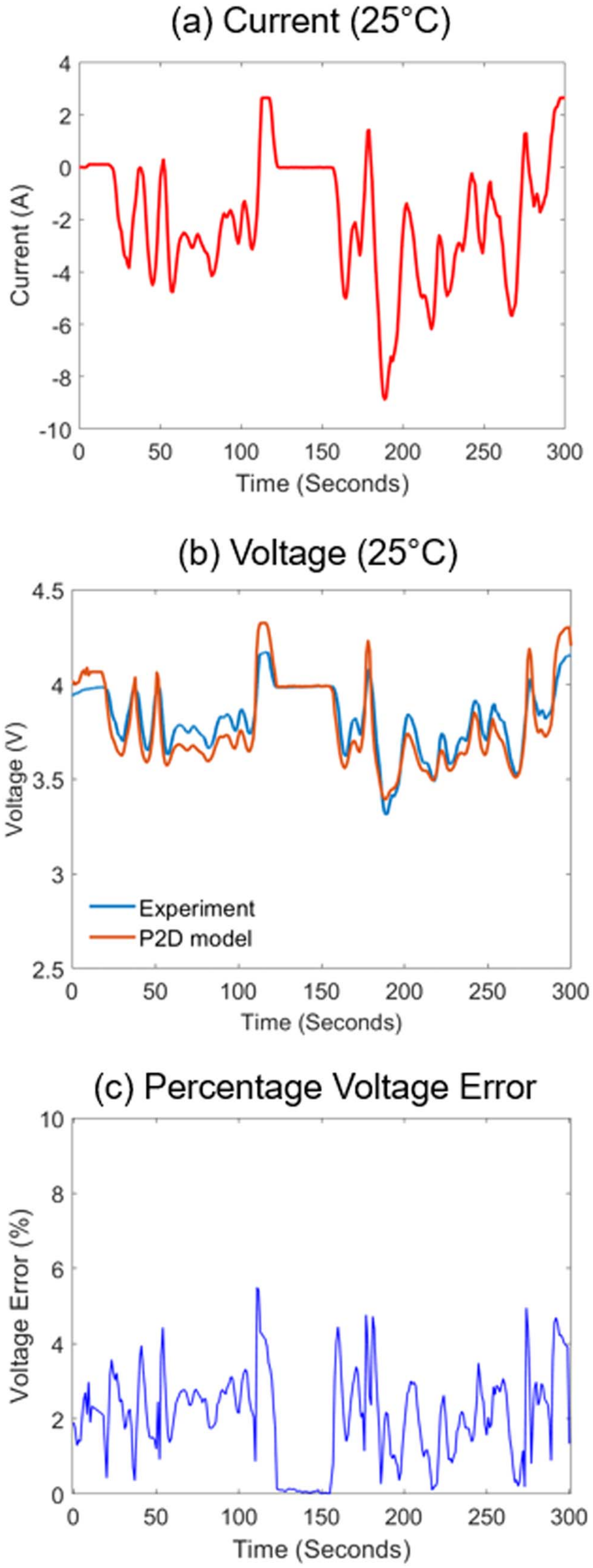

Figure 7. Simulations of the DEARLIBS under dynamic operational conditions. (a) UDDS current profiles (red) (b) voltage profile by experimental measurment (blue) and simulation results of DFN implementation (orange) (c) percentage voltage error of voltage profiles between model outputs and experimental measurment over time (blue). Initial state of charge is 85%, and the number of node point is (20,10,20) = (Cathode, Separator, Anode).

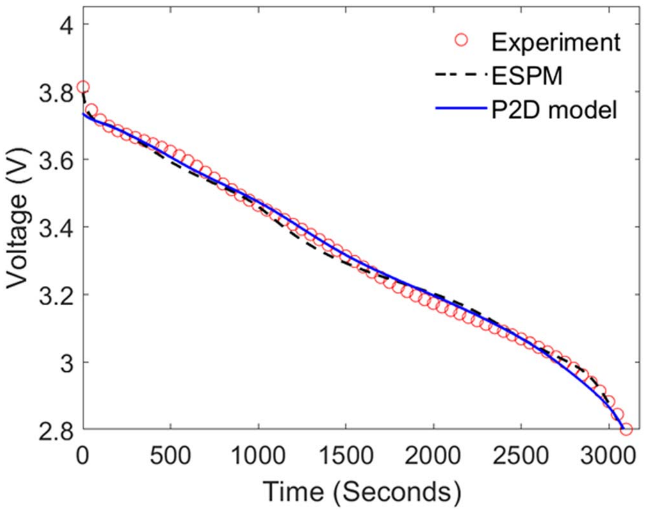

Figure 8. Comparison of voltage profiles at 1C constant discharge between the ESPM and DFN model, and experimental measurments. In this results, ESPM are implemented using the same MATLAB® framework. Governing equations are implemented on the same framework.

positive and negative electrode, and R is the gas constant. Taking logarithms of both right- and left side terms produces a plot of ln $D_{s,i}$ vs $\frac{1}{T}$, whose gradient is $\frac{-E_{a,i,D}}{R}$ and intersection is $ln\ D_{s,0}$, as shown in Eq. 40.

$$\ln D_{s,i} = \frac{-E_{a,i,D}}{RT} + ln\ D_{s0,i} \tag{40}$$

Equations 38 and 40 can be utilized to determine the reference reaction rate constant at the positive and negative electrode ($k_{0,p}$ and $k_{0,n}$), the activation energy for the reaction at the positive and the negative electrode ($E_{a,\ p,\ k}$ and $E_{a,\ n,\ k}$), the reference solid phase diffusion coefficient at the positive and the negative electrode ($D_{s0,p}$ *and* $D_{s0,n}$), and the activation energy for the diffusivity at the positive and the negative electrode ($E_{a,p,D}$ *and* $E_{a,n,D}$) as presented in Fig. 6. In addition, Arrhenius relationships between identified solid phase diffusion coefficients and rate constants at different temperature were investigated to confirm model parameters' behavior. The results shown in Fig. 6 indicate that the solid phase diffusion coefficients and reactions rates follow the Arrhenius relationship closely.

Validation.—In this section, we validate identified parameters in the previous section. We simulate the DFN model in DEARLIBS with UDDS schedule profiles at 25 °C, using (20, 10, 20). The UDDS profile represents city driving conditions for light-duty vehicle tests (*see* Fig. 7a), including dynamic operational conditions that operate the battery system every 0.1 s time step (e.g., 300-second simulation will include 3000 steps). Thought the UDDS was originally created as a reference for fossil-fueled vehicles, it is also used to estimate how many miles an EV can travel on a single charge. The DFN implementation is successfully validated with the experimental UDDS voltage profiles, as shown in Fig. 7b. The percentage voltage error is illustrated as a function of time in Fig. 7c, where the error value is calculated as shown below:

$$\%\ \text{voltage error} = |V_{exp,j} - V_{model,j}| \cdot \frac{100 \cdot N}{\sum_{i=1}^{N} V_{exp,j}}$$

N represents the total number of experimental data points for UDDS profiles, $V_{exp,j}$ and $V_{model,j}$ indicate the experimental and model

Table XI. Identified Kinetic/Transport Parameters between ESPM and DFN model.

Symbol (Unit)	Parameter	Lower bound	Upper bound	Initial guess	Final value		% Error
					ESPM	DFN	
D (m^2 s^{-1})	Electrolyte diffusion coefficient	0.7×10^{-9}	1.3×10^{-9}	1×10^{-9}	1×10^{-9}	0.7×10^{-9}	42%
κ (S m^{-1})	Electrolyte conductivity	0.82	1.52	1.17	1.33	1.33	0%
σ_p (S m^{-1})	Solid phase conductivity at positive	0.13	0.23	0.18	N/A	0.15	N/A
σ_n (S m^{-1})	Solid phase conductivity at negative	150.5	279.5	215	N/A	279.5	N/A
$D_{s,p}$ (m^2 s^{-1})	Solid phase diffusivity at positive	2.8×10^{-15}	5.2×10^{-15}	4×10^{-15}	5.2×10^{-15}	4.9×10^{-15}	6.1%
$D_{s,n}$ (m^2 s^{-1})	Solid phase diffusivity at negative	2.31×10^{-14}	5.94×10^{-14}	3.3×10^{-14}	2.53×10^{-14}	4.3×10^{-14}	41%
k_p mol/(m^2·s)	Reaction rate constant at positive	0.49×10^{-11}	0.91×10^{-11}	0.7×10^{-11}	0.7×10^{-11}	0.8×10^{-11}	12.5%
k_n mol/(m^2·s)	Reaction rate constant at negative	0.49×10^{-12}	0.91×10^{-12}	0.7×10^{-12}	0.53×10^{-12}	0.5×10^{-12}	6%

Algorithm 1. DEARLIBS in Pseudo-code.

```
Inputs: Applied current and voltage data by experiment measurement
Output: Voltage profiles from model outputs and identified parameter values
(Symbolic form DAEs after numerical discretization)
Line 1: Set up model parameters, applied current, and number of node points
Ordinary differential equations (ODEs)
$\frac{dy}{dt} = \boldsymbol{f}(t, \boldsymbol{y}, z)$;
Algebraic different equations (AEs)
$\boldsymbol{g}(t, \boldsymbol{y}, z) = 0$
where $z$ represents the set of AE variables, $\boldsymbol{y}$ represents the set of ODE variables, and $t$ represents simulation time (Symbolic form ODEs after single-step iteration-free algorithm)
Line 2: ODEs $\frac{dy}{dt} = f(t, \boldsymbol{y}, z)\frac{1}{2}(1 + tanh(q(t - t_j)))$
Implicit ODEs $\varepsilon\frac{d\boldsymbol{g}}{dt} = -\boldsymbol{g}(t, \boldsymbol{y}, z)$ where $\varepsilon$ represents the perturbation coefficient, $t_j$ represents the simulation starting time, and $q$ represents the switch function coefficient
Line 3: Convert symbolic form into function MATLAB® handles: odeFunction
Line 4: Input $y = y_{0}, g = g_{0}$(initial guesses based the equilibrium status)
(Parameter identification)
Line 5: Set up lower and upper bounds for parameters to be identified
Call in-built optimization:
1st iteration: Go to objective function (line 9)
2nd iteration ~ end: Compare present and previous objective function and go to line 9
Once completing parameter identification routines, go to line 6
Line 6: Obtain optimal battery performance by ode15s ($\boldsymbol{y}, \boldsymbol{g}, \boldsymbol{y}_{0}, g_{0}$)
(Objective function)
Line 7: Receive parameters and simulate the model by ode15s ($\boldsymbol{y}, \boldsymbol{g}, \boldsymbol{y}_{0}, g_{0}$)
Line 8: Call voltage data from experiment measurement
Line 9: Calculate objective function and go to line 5
    $obj = \frac{1}{N}\sum_{j=1}^{N}\sqrt{(V_{model,j} - V_{exp,j})^2}$ where $V_{model,j}$ represents jth voltage data from model outputs, $V_{exp,j}$ represents jth voltage data obtained by experiment, and N represents the total number of experimental data
```

predicted voltage value for the *jth* data point. The percentage voltage error is confirmed to be less than 5% in Fig. 7c.

Conclusion and Perspective

This work provides a robust and sleek DFN MATLAB® implementation- DEARLIBS. The robustness of proposed DFN implementation was tested under fast charging conditions with high numbers of discretization node points. Besides, the DFN model parameters were identified with constant current-discharging data at different temperatures, and confirmed to follow Arrhenius relationships, and validated using UDDS profiles.

The significant advantage of DEARLIBS compared to current open-access codes is that it maximizes users' convenience to include a parameter identification routine along with a standard solver provided by MATLAB®. In DEARLIBS, for example, model parameters in different chemistries can be identified to obtain different sets of parameters. Different cathode chemistries have been developed for LIBs available today (e.g., NCA, NMC, lithium-manganese, lithium-titanite, and lithium-iron-phosphate). In the EV systems, NCA and NMC cathode chemistries have been mostly adopted.[4] In the electricity market, batteries can provide up to 13 different services, where each of them would require different energy and power requirements. In the back-up power application, for example, the battery system must provide energy quickly upon users' needs, which would call for batteries with high specific power.[4] On the other hand, in the daily charging and discharging residential photovoltaic/battery application, the battery system must provide a more stable charging and discharging performance.[4] In this case, the batteries with a longer life cycle and higher durability are preferred. In DEARLIBS, users can implement their own parameter routines with in-house experimental data for different chemistries with minimized efforts. A direct implementation of standard solvers enables coupling parameter identification routines with minimized efforts, and this will help accelerate spread of the battery models as well as renewable energy systems. In contrast, current open-access codes lack parameter identification techniques in their modeling framework, and it might not be efficient to couple it with parameter identification.

Parameter identification is an essential component and key developmental phase to maximize battery performance for a specific energy application. A robust and efficient initialization algorithm is one of the most crucial parts to efficiently implement parameter identification techniques. During the parameter identification process, a set of model parameter values are changed at every iteration. The iteration process requires different consistent initial conditions depending on a set of updated parameter values. To overcome the initialization issue, the efficient and robust initialization algorithm, which was proposed by Matt Lawder et al. in 2015, is implemented in DEARLIBS.[33] The study of parameter identification in various cathode chemistries will be reported in the next work by utilizing the DEARLIBS.

Another possible research task that can be performed by DEARLIBS is the comparison and analysis between different types of physics-based models along with different lithium-ion chemistries. In the current practice, battery performance between the DFN model, ESPM, and SPM has been compared without the addition of experimental data.[53–55] In these studies, model parameters for specific chemistry such as $LiCoO_2$ have typically relied on the literature. It is assumed that the DFN model produces actual battery performance, and the ESPM is known to maintain only a few mV errors up to 5C for the discharging voltage profiles obtained from the DFN model.[53–55] However, the DFN model includes the aforementioned inaccuracy at the operating condition of low SOC, medium-high temperature, and medium-high C-rate.[16] In addition, the current research is conducted with model parameters of several specific chemistries collected from the literature.[53–55] More detailed studies on the comparison between battery models are required for different lithium-ion chemistries. For this reason, parameter identification routine implemented in DEARLIBS is a necessary technique to accelerate the relevant research topics.

Figure 8 shows an example of the voltage comparison between ESPM and DFN models with experimental voltage data, using DEARLIBS (1C constant discharge at 25 °C). In the ESPM framework, the solid particle concentration is described along with the *r* dimension, and it is formulated upon the assumption that each electrode can be represented by a single spherical particle. This implies that all solid particles are uniform and have the same chemical properties. The variation of electrochemical potential in the solid particles along with *x*-axis is ignored whereas the electrolyte concentration and potential are considered. In the ESPM framework, the molar flux of the electrolyte for the solid particles is considered to be the average value across each electrode as follows:[53,56]

$$j_p = \frac{i_{app}}{a_p l_p F} = 2c^{0.5} k_p {c_{sp,surf}}^{0.5} \times (c_{sp,\max} - c_{sp,surf})^{0.5} \sinh\left(\frac{\alpha F}{RT}(\Phi_{s,p} - \Phi_{e,p} - U_p)\right) \qquad [41\text{-}1]$$

$$j_n = -\frac{i_{app}}{a_n l_n F} = 2c^{0.5} k_n {c_{sn,surf}}^{0.5} \times (c_{sn,\max} - c_{sn,surf})^{0.5} \sinh\left(\frac{\alpha F}{RT}(\Phi_{s,n} - \Phi_{e,n} - U_n)\right) \qquad [41\text{-}2]$$

In addition, this assumption can lead to the analytical solution of liquid phase potential as shown below:[53,56]

$$\Phi_{e,p}(x) = (1 - t_+)\frac{2RT}{F} \ln\frac{c_{s,p}(x)}{c_{s,p}(0)} + \frac{i_{app}}{2}\left(\frac{l_n}{\kappa_{eff,n}} + 2\frac{l_s}{\kappa_{eff,s}} + \frac{l_p}{\kappa_{eff,p}}\right) \qquad [42]$$

where *x* represents the spatial coordinate of the electrode.

In Fig. 8, the ESPM implements Eqs. 1, 4, 5, 7, 10 in Table III, Eqs. 11–13 in Table IV, Eqs. 16–1 and 17–1 in Table V, and Eqs. 41–42. The kinetic/transport parameters in ESPM are identified with experimental voltage data with 1C CC-discharge, using the same upper and lower bounds from the DFN model (*see* Table XI). The RMSEs of the ESPM and the DFN model are 18 mV and 17 mV, respectively. Future work will involve detailed simulation and optimization studies on the ESPM, SPM, and DFN models along with model parameters of different chemistries obtained from parameter identification routines in DEARLIBS.

In addition to the aforementioned advantages, physics-based battery models can incorporate thermal/stress/mechanical effects, chemical/electrochemical kinetics, transport phenomena, and side reactions in DEARLIBS. The safety issues and capacity/pawer fade can always occur during the battery operation, including thermal runaway, electrolyte decomposition processes, lithium plating, and formation of the solid-electrolyte interphase layer. The proposed DEARLIBS can be used by both modelers, battery experimentalist, and control engineers, to understand lithium-ion battery systems and get accurate estimates under various operating conditions and across different chemistries.

Acknowledgments

This work was supported by the Bits and Watts Initiative within the Precourt Institute for Energy at Stanford University. The authors thank Anirudh Allam and Edoardo Catenaro for the help to conduct the battery cell experimental protocols. The authors also thank Razvan Carbunescu and Jacek Kierzenka at MathWorks for their suggestion to improve the robustness of DEARLIBS. S.B. Lee would like to pay his respects to Professor Venkat Subramanian and his pioneering work on the initialization approach adopted in this paper.

This paper is dedicated to all researchers who devoted their time, effort and energy to the development of electrochemical models for lithium-ion batteries.

ORCID

Seong Beom Lee https://orcid.org/0000-0003-2411-9017
Simona Onori https://orcid.org/0000-0002-6556-2608

References

1. S. Xia, X. Wu, Z. Zhang, Y. Cui, and W. Liu, *Chem*, **5**, 753 (2019).
2. E. Prasad (2020), Available: (https://apnews.com/press-release/business-wire/0f4e09f0cb524a5d83c32c4e9be5dade) AP NEWS.
3. J. PYPER (2019), Available: (https://greentechmedia.com/articles/read/us-electric-vehicle-sales-increase-by-81-in-2018) Grrentech Media.
4. A.-I. Stan, M. Świerczyński, D.-I. Stroe, R. Teodorescu, and S. J. Andreasen, *2014 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM),* 713 (2014).
5. H. Jiayi, J. Chuanwen, and X. Rong, *Renew. Sustain. Energy Rev.*, **12**, 2472 (2008).
6. EIA (U.S. Energy Information Administration Tech. Rep.), (2018), U.S. battery storage market trends.
7. K. Mongird, V. V. Viswanathan, P. J. Balducci, M. J. E. Alam, V. Fotedar, V. S. Koritarov, and B. Hadjerioua, *Pacific Northwest National Lab.(PNNL)*, Richland, WA, USA (2019), https://energystorage.pnnl.gov/pdf/PNNL-28866.pdf.
8. M.-F. Ng, J. Zhao, Q. Yan, G. J. Conduit, and Z. W. Seh, *Nature Machine Intelligence*, **2**, 161 (2020).
9. R. Ahmed, J. Gazzarri, S. Onori, S. Habibi, R. Jackey, K. Rzemien, J. Tjong, and J. LeSage, *SAE International Journal of Alternative Powertrains*, **4**, 233 (2015).
10. V. Ramadesigan, P. W. Northrop, S. De, S. Santhanagopalan, R. D. Braatz, and V. R. Subramanian, *J. Electrochem. Soc.*, **159**, R31 (2012).
11. M. T. Lawder, B. Suthar, P. W. Northrop, S. De, C. M. Hoff, O. Leitermann, M. L. Crow, S. Santhanagopalan, and V. R. Subramanian, *Proc. IEEE*, **102**, 1014 (2014).
12. K. A. Severson, P. M. Attia, N. Jin, N. Perkins, B. Jiang, Z. Yang, M. H. Chen, M. Aykol, P. K. Herring, and D. Fraggedakis, *Nat. Energy*, **4**, 383 (2019).
13. M. A. Xavier, A. K. De Souza, K. Karami, G. L. Plett, and M. S. Trimboli, *IEEE Control Systems Letters*, **5**, 1387 (2020).
14. G. Pozzato, S. B. Lee, and S. Onori, *IEEE Conference on Control Technology and Applications* (2021).
15. X.-G. Yang, Y. Leng, G. Zhang, S. Ge, and C.-Y. Wang, *J. Power Sources*, **360**, 28 (2017).
16. H. Arunachalam and S. Onori, *J. Electrochem. Soc.*, **166**, A1380 (2019).
17. A. M. Bizeray, S. Zhao, S. R. Duncan, and D. A. Howey, *J. Power Sources*, **296**, 400 (2015).
18. P. W. Northrop, V. Ramadesigan, S. De, and V. R. Subramanian, *J. Electrochem. Soc.*, **158**, A1461 (2011).
19. A. Allam and S. Onori, *IEEE Trans. Ind. Electron.*, **65**, 7311 (2018).
20. A. Allam, E. Catenaro, and S. Onori, *Iscience*, **23**, 101847 (2020).
21. M. Doyle, T. F. Fuller, and J. Newman, *J. Electrochem. Soc.*, **140**, 1526 (1993).
22. S. Korneev, H. Arunachalam, S. Onori, and I. Battiato, *Transp. Porous Media*, **134**, 173 (2020).
23. B. Guest, M. S. Trimboli, and G. L. Plett, *J. Electrochem. Soc.*, **167**, 160546 (2020).
24. H. Arunachalam, S. Onori, and I. Battiato, *J. Electrochem. Soc.*, **162**, A1940 (2015).
25. S. Carelli and W. G. Bessler, *J. Electrochem. Soc.*, **167**, 100515 (2020).
26. W. Ai, L. Kraft, J. Sturm, A. Jossen, and B. Wu, *J. Electrochem. Soc.*, **167**, 013512 (2019).
27. Z. Deng, L. Yang, H. Deng, Y. Cai, and D. Li, *Energy*, **142**, 838 (2018).
28. S. Kolluri, S. V. Aduru, M. Pathak, R. D. Braatz, and V. R. Subramanian, *J. Electrochem. Soc.*, **167**, 063505 (2020).
29. S. B. Lee, C. Pathak, V. Ramadesigan, W. Gao, and V. R. Subramanian, *J. Electrochem. Soc.*, **164**, E3026 (2017).
30. J. Peiró and S. Sherwin, *Handbook of Materials Modeling* (Springer, Berlin) 2415 (2005).
31. C.-H. Chen et al., *J. Electrochem. Soc.*, **167**, 080534 (2020).
32. T. Ashwin, A. McGordon, W. D. Widanage, and P. A. Jennings, *J. Power Sources*, **341**, 387 (2017).
33. M. T. Lawder, V. Ramadesigan, B. Suthar, and V. R. Subramanian, *Comput. Chem. Eng.*, **82**, 283 (2015).
34. A. Bizeray, *PhD Thesis*, University of Oxford (2016), Available: (https://ora.ox.ac.uk/objects/uuid:f326b332-b912–4bf6-a9b3-d5e61d3a9596).
35. Y. Qi, S. Kolluri, D. T. Schwartz, and V. R. Subramanian, *ECS Trans.*, **75**, 121 (2017).

36. S. B. Lee, K. Mitra, H. D. Pratt, T. M. Anderson, V. Ramadesigan, B. R. Chalamala, and V. R. Subramanian, *Journal of Electrochemical Energy Conversion and Storage*, **17**, 011008:1-19 (2020).
37. A. Nyman, M. Behm, and G. Lindbergh, *Electrochim. Acta*, **53**, 6356 (2008).
38. J. Li, L. Zou, F. Tian, X. Dong, Z. Zou, and H. Yang, *J. Electrochem. Soc.*, **163**, A1646 (2016).
39. J. Newman (2014), Fortran Programs for the Simulation of Electrochemical Systems (Dualfoil), http//www. cchem. berkeley. edu/jsngrp.
40. S. Moura, (2016), Energy, Controls, and Applications Lab (eCAL) University of Californian Berkeley, Available: (https://github.com/scott-moura/fastDFN).
41. M. Torchio, L. Magni, R. B. Gopaluni, R. D. Braatz, and D. M. Raimondo, *J. Electrochem. Soc.*, **163**, A1192 (2016).
42. V. Sulzer, S. G. Marquis, R. Timms, M. Robinson, and S. J. Chapman, *Journal of Open Research Software*, **9**, 14 (2020).
43. P. W. Northrop, B. Suthar, V. Ramadesigan, S. Santhanagopalan, R. D. Braatz, and V. R. Subramanian, *J. Electrochem. Soc.*, **161**, E3149 (2014).
44. D. Sonawane, M. Pathak, V. Subramanian, and M. Lawder, (2020), Vol. US Patent 10769236B2, USA.
45. N. Dawson-Elli, S. B. Lee, M. Pathak, K. Mitra, and V. R. Subramanian, *J. Electrochem. Soc.*, **165**, A1 (2018).
46. R. N. Methekar, V. Ramadesigan, J. C. Pirkle Jr, and V. R. Subramanian, *Comput. Chem. Eng.*, **35**, 2227 (2011).
47. MathWorks, https://mathworks.com/help/symbolic/daefunction.html.
48. S. Park, D. Kato, Z. Gima, R. Klein, and S. Moura, *J. Electrochem. Soc.*, **165**, A1309 (2018).
49. J. C. Forman, S. J. Moura, J. L. Stein, and H. K. Fathy, *J. Power Sources*, **210**, 263 (2012).
50. LG M50 21700, 5000mAh 7.3A Battery, https://18650batterystore.com/21700-p/lg-m50.htm.
51. Arbin LBT21024, Battery Test System, Advanced Test Equipment Rentals, https://atecorp.com/products/arbin/lbt21024.
52. IncuMax, IC500R refrigerated incubator from Amerex Instruments, Inc., https://biocompare.com/12104-Equipment/2262254-IncuMax-IC500R-refrigerated-incubator/.
53. S. J. Moura, F. B. Argomedo, R. Klein, A. Mirtabatabaei, and M. Krstic, *IEEE Trans. Control Syst. Technol.*, **25**, 453 (2016).
54. P. Kemper and D. Kum, *2013 IEEE Vehicle Power and Propulsion Conference (VPPC)* (IEEE, Piscataway, NJ) 1 (2013).
55. T. R. Tanim, C. D. Rahn, and C.-Y. Wang, *J. Dyn. Syst. Meas. Control*, **137** (2015).
56. E. Prada, D. Di Domenico, Y. Creff, J. Bernard, V. Sauvant-Moynot, and F. Huet, *J. Electrochem. Soc.*, **159**, A1508 (2012).