



A fast fixed-point solution framework for the P2D model of lithium-ion batteries

Downloaded from: <https://research.chalmers.se>, 2025-11-29 23:00 UTC

Citation for the original published paper (version of record):

Li, Y., Wik, T., Zhu, Q. et al (2025). A fast fixed-point solution framework for the P2D model of lithium-ion batteries. *Journal of Power Sources*, 661.
<http://dx.doi.org/10.1016/j.jpowsour.2025.238591>

N.B. When citing this work, cite the original published paper.



A fast fixed-point solution framework for the P2D model of lithium-ion batteries

Yang Li[✉]*, Torsten Wik[✉], Qingbo Zhu[✉], Yicun Huang[✉], Yao Cai[✉], Changfu Zou[✉]*

Department of Electrical Engineering, Chalmers University of Technology, Gothenburg 41296, Sweden

HIGHLIGHTS

- Reformulates the P2D model into a quasilinear PIDE system without algebraic states.
- Proposes a fixed-point operator with closed-form IMEX single-step updates.
- Stable, efficient simulation without Jacobians, robust to initial guess errors.
- Well-suited for battery systems under advanced control and dynamic loads.

ARTICLE INFO

Keywords:

Battery modeling
Pseudo-two-dimensional (P2D) model
Partial integro-differential equation (PIDE)
Gradient-free method
Lithium-ion battery
Fixed-point iterative method

ABSTRACT

This paper presents a novel algorithmic framework for efficiently solving the pseudo-two-dimensional (P2D) model of lithium-ion batteries. The proposed approach reformulates the original P2D model, typically expressed as a system of coupled nonlinear partial differential–algebraic equations, into a system of quasilinear *partial integro-differential equations (PIDEs)*. Through this reformulation, intermittent algebraic states, such as local potential and current terms, are effectively eliminated, thereby reducing the model complexity. This enables the identification of a generic fixed-point iterated function for solving the P2D model's nonlinear algebraic equations. To implement this iterated function, the finite volume method is employed to spatially discretize the PIDE system into a system of ordinary differential equations. An implicit–explicit (IMEX) time integration scheme is adopted, and the resulting quasilinear structure facilitates the development of a single-step numerical integration scheme that admits a closed-form update, providing stable, accurate, and computationally efficient solutions. Unlike traditional gradient-based approaches, the proposed framework does not require the Jacobian matrix and is insensitive to the initial guess error of the solution, making it easier to implement and more robust in practice. Due to its significantly reduced computational cost, the proposed framework is particularly well-suited for simulating large-scale battery systems operated under advanced closed-loop control strategies.

1. Introduction

Lithium-ion (Li-ion) batteries have demonstrated outstanding performance as power sources for electric vehicles (EVs) and energy storage systems in modern electric power grids [1,2]. These batteries offer the typical advantages of electrochemical devices, such as modularity and scalability, while also providing higher energy density compared to other battery chemistries [3]. However, their health and safety issues require careful consideration during both the design and operational stages. It is vital to predict both the measurable characteristics and internal, unmeasurable states to ensure the safety, reliability, and longevity of Li-ion batteries.

Mathematical models are valuable and often essential tools for quantifying the physical mechanisms that govern performance and

for understanding their evolution over a battery's lifetime [4]. Today, due to their mathematical simplicity, ease of implementation, and low computational cost, equivalent circuit models (ECMs) are most widely used in the development of the algorithms for battery management systems [5]. In an ECM, the dynamic electrochemical behavior of batteries is emulated using electrical circuits composed of basic components such as capacitors, resistors, inductors, and voltage/current sources. Low-order, empirically derived ECMs are computationally efficient and well-suited for applications operating within narrow current and temperature ranges. However, ECM parameters must be identified from available experimental data, and extrapolation beyond the observed range poses challenges for emerging applications such as ultra-fast charging [6] and low-temperature operation of electrified

* Corresponding authors.

E-mail addresses: yangli@ieee.org (Y. Li), changfu.zou@chalmers.se (C. Zou).

Nomenclature**Latin Letters**

F	Faraday's constant [C mol^{-1}]
R	Universal gas constant [$\text{J mol}^{-1} \text{K}^{-1}$]
a_s	Particle surface area to volume [m^{-1}]
c_{ss}	Li-ion concentration at the particle surface [mol m^{-3}]
c_e	Li-ion concentration in the electrolyte [mol m^{-3}]
c_e^0	Average Li-ion concentration in the electrolyte [mol m^{-3}]
c_p	Specific heat capacity [$\text{J kg}^{-1} \text{K}^{-1}$]
c_s	Li-ion concentration in the solid phase [mol m^{-3}]
$c_{s,\max}$	Theoretical maximum Li-ion concentration in the solid phase [mol m^{-3}]
c_{s0}	Initial solid-phase concentration [mol m^{-3}]
$D_{e,\text{eff}}$	Effective electrolyte diffusion coefficient [$\text{m}^2 \text{s}^{-1}$]
$D_{s,\text{eff}}$	Effective solid-phase diffusion coefficient [$\text{m}^2 \text{s}^{-1}$]
$f_{+/-}$	Activity coefficient [-]
h	Heat transfer coefficient [$\text{W m}^{-2} \text{K}^{-1}$]
i_0	Exchange current density [A m^{-2}]
i_e	Electrolyte current density [A m^{-2}]
i_s	Solid-phase current density [A m^{-2}]
i_{app}	Applied charging current density [A m^{-2}]
j_n	Intercalation molar flux [$\text{mol m}^{-2} \text{s}^{-1}$]
k_{eff}	Effective electrode rate constant [$\text{A m}^{2.5} \text{mol}^{-1.5}$]
L	Width of a domain
l	Width of a control volume (m)
N	Number of control volumes in a domain
p	Order of convergence
q	Error contraction ratio
q_{col}	Heat flux of current collectors [W m^{-2}]
q_{ohm}	Heat flux due to ohmic effects [W m^{-2}]
q_{rev}	Reversible heat flux [W m^{-2}]
q_{rxn}	Heat flux due to intercalation [W m^{-2}]
q_{sei}	Heat flux of SEI [W m^{-2}]
q_{tot}	Total heat generation rate [W m^{-2}]
r	Microscopic spatial coordinate [m]
r_{Σ}	Sum of r_{ct} and $r_{f,\text{eff}}$ [Ωm^2]
R_p	Radius of assumed spherical particle [m]
r_{col}	Areal resistance of current collectors [Ωm^2]
r_{ct}	Areal charge-transfer resistance [Ωm^2]
$r_{f,\text{eff}}$	Effective areal SEI film resistance [Ωm^2]
T	Battery temperature [K]
t	Temporal coordinate [s]
T_s	Time step size [s]
t_+	Transference number [-]
T_{amb}	Ambient temperature [K]
T_{ref}	Reference temperature [K]
U	Equilibrium potential of a reaction [V]
U_{ss}	Open-circuit potential of an electrode [V]
U_e	An overpotential term of electrolyte [V]
V_{bat}	Battery terminal voltage [V]

x Macroscopic spatial coordinate [m]

Greek Letters

α	Symmetric coefficient [-]
β	$= 2(1 - t_+)$ [-]
δ	Electrode/separator boundary [m]
η_{ct}	Charge-transfer overpotential [V]
κ_{eff}	Effective electrolyte conductivity [S m^{-1}]
Φ_e	Electrolyte potential [V]
Φ_s	Solid-phase potential [V]
ρ	Mass density [kg m^{-3}]
σ_{eff}	Effective solid-phase conductivity [S m^{-1}]
ε_e	Volume fraction of the electrolyte [-]
ε_s	Volume fraction of the solid phase [-]
ϱ	Combined resistivity of solid-phase and electrolyte [Ωm]

Superscript

+	Positive electrode
-	Negative electrode
-1	Inverse
$\pm$	Positive or negative electrode domain
T	Transpose
j	Electrode or separator domain
sep	Separator

Subscript

i	Control volume index
$i + \frac{1}{2}$	Interface between of the i th and $(i + 1)$ th control volumes
k	Iteration index

vehicles [7]. There is a growing need to design battery cells for higher current rates, more dynamic load profiles, and increasingly harsh environments. Meeting these demands with ECMs typically requires increased model order, more complex parameter-fitting functions, and greater experimental effort to identify the parameters accurately [8]. Furthermore, since ECM components do not necessarily have direct mechanistic relationships with underlying electrochemical processes, these models tend to provide limited insight into physically meaningful, time-varying parameters, degradation mechanisms, and internal safety constraints. As a result, the predictive accuracy of ECMs can degrade significantly if the evolving battery dynamics due to aging are not properly taken into account.

In contrast to ECMs, physics-based models are derived from fundamental principles of batteries, and they can overcome many limitations of ECMs and better exploit the full potential of Li-ion batteries [9,10]. A foundational framework in this category is the pseudo-two-dimensional (P2D) model, commonly referred as the “DFN model”, introduced by Doyle, Fuller, and Newman [11–13]. The P2D model is a system of partial differential-algebraic equations (PDAEs) that consists of a number of coupled nonlinear partial differential equations (PDEs) and algebraic equations (AEs). A major advantages of the P2D model is its extensibility: it can be readily modified to incorporate additional physical phenomena such as thermal effects, mechanical stress, and degradation mechanisms. However, the inherent complexity of the model makes it computationally demanding to solve using conventional numerical methods, particularly for large-scale, long-duration simulations and real-time battery management applications [14].

In recent years, various attempts have been made to reduce the computational burden of physics-based models by employing model

Table 1
Comparison of nonlinear algebraic equation solvers.

Method	Gradient-free	Jacobian-free	Robust to initial guess	Convergence rate	Memory usage	Large-scale suitable	Parameter tuning free
Newton	×	×	Very Poor	Quadratic	Medium	×	×
QN-Broyden	×	✓	Poor	Superlinear	Medium	✓	×
Gauss–Newton	×	×	Poor	Linear	Medium	×	×
Trust-region	×	×	Medium	Superlinear	High	×	×
LM	×	×	Medium	Superlinear	Medium	×	×
Proposed	✓	✓	Good	Approx. Linear	Low	✓	✓

order reduction (MOR) techniques [15]. One widely used approach is the single particle model (SPM), which simplifies each electrode of the battery cell as a lumped domain, ignoring electrolyte dynamics and temperature effects [16]. The SPM has been used extensively in the design of state-of-charge (SOC) estimators [17,18]. To improve the predictive accuracy, the model was subsequently extended to incorporate electrolyte dynamics and capture thermal behavior [19,20]. However, the underlying assumption of a “lumped particle” representation is generally valid only at low to moderate current rates and can be significantly violated under more demanding conditions. This limitation is particularly pronounced in the batteries with thick electrodes, a growing trend in modern battery design. The shortcomings of the SPM and its variants have driven the development of simplified P2D models that retain spatially distributed behaviors within the electrodes. Examples include spatially discretized P2D models [21–23] and polynomially approximated formulations [24]. Nevertheless, the accuracy of these reduced order models often depends on specific operating conditions and model assumptions, which limit their generalizability. Consequently, such models are generally unsuitable for predicting battery behavior under extreme operating scenarios, such as ultra-high current rates or high/low temperature environments, where batteries may approach their internal physical limits.

As a result, significant efforts have been directed towards developing more efficient algorithms for solving the full-order battery models [25–28], with the goal of enabling both online battery management and offline applications such as machine learning, parameter identification, and optimal design, where large numbers of simulations are often required. A commonly adopted strategy involves applying the method of lines (MOL), which converts the system of PDAEs into a set of highly stiff differential-algebraic equations (DAEs) via spatial discretization techniques, such as the finite volume method (FVM), or spectral methods like Galerkin [29] and orthogonal collocation [30] approaches. The resulting stiff DAEs are typically solved using well-established time integration schemes, such as the backward differentiation formula (BDF), Rosenbrock methods, implicit Runge–Kutta methods, and direct collocation approaches [31]. Many existing P2D simulation studies employ general-purpose solvers such as IDA from the SUNDIALS suite [32] or MATLAB’s ode15s, both of which are based on BDF. However, these solvers are designed for continuous-time systems and are not well aligned with the discrete-time and single-step structures preferred in the control systems, where real-time implementation and computational efficiency are crucial. Moreover, because these solvers are not specifically tailored to the structure of the P2D model, they often suffer from computational inefficiencies when applied to its highly nonlinear and stiff dynamics.

In practice, the computational efficiency of full-order model solvers is often limited by the techniques used to iteratively solve the nonlinear AEs. To explain this, consider a generic form of the iterated update used in many existing algorithms:

$$\mathbf{v}_{k+1}(t) = \mathbf{v}_k(t) + \mathbf{h}(\mathbf{x}(t), u(t), \mathbf{v}_k(t), \lambda), \quad (1)$$

where t denotes the time index, $\mathbf{v}_k$ and $\mathbf{v}_{k+1}$ represent the present and the next iterates of the algebraic state vectors, respectively, $\mathbf{x}$ denotes the system state vectors, u is the control input, $\mathbf{h}(\cdot)$ is a vector-valued function defining the update rules, and λ represents a set of

tuning parameters. The iterative process continues until convergence is achieved at $\mathbf{v}^*(t)$, the solution of the algebraic subproblem. These class of algorithms typically suffers from the following limitations.

(1) Many widely used algorithms, including Newton’s methods, quasi-Newton (QN) methods, Gauss–Newton (GN) methods, and various optimization-based approaches, such as trust-region (TR) and Levenberg–Marquardt (LM) methods, aim to directly reduce the update increment $\mathbf{v}_{k+1} - \mathbf{v}_k$. These methods are fundamentally gradient-based and require either the computation or approximation of Jacobian matrices within the update function $\mathbf{h}(\cdot)$. For large-scale systems, this process is computationally intensive and may introduce numerical stability issues.

(2) The convergence speed and reliability of nonlinear AE solvers are highly sensitive to the quality of initial guess $\mathbf{v}_0(t)$. For smooth current profiles, such as during constant-current charging and discharging, the solution from the previous time step $\mathbf{v}^*(t - T_s)$ (where T_s denotes the time step size) can be reused as an effective initial guess. This is justified by the fact that under the condition $u(t) \approx u(t - T_s)$ and $\mathbf{x}(t) \approx \mathbf{x}(t - T_s)$, the algebraic solution $\mathbf{v}^*$ is also expected to change gradually. However, during highly dynamic conditions, such as fast-changing load currents encountered in realistic driving cycles, this assumption no longer holds, leading to increased iteration counts and reduced computational efficiency.

(3) The performance of these iterative solvers often depends on appropriate selection of tuning parameters λ , which govern the trade-off between numerical stability and convergence speed. Identifying suitable values typically involves extensive trial-and-error and can complicate the design of robust, general-purpose solution schemes.

Is it possible to develop a nonlinear AE solver tailored specifically for the P2D model that can overcome the above-mentioned challenges? To answer this question we need a deeper investigation into the structure and inherent properties of the P2D model, an area that has received limited attention in the research community. One early attempt to improve the efficiency of solving the DAEs, without relying on the iterative form given in (1), involves reformulating the model into an equivalent circuit network [33]. In this approach, circuit theory was applied to transform the AEs obtained from the FVM into a system of linear equations. This allows the algebraic states to be calculated in closed form without the need for iteration. The solution takes the form:

$$\mathbf{v}^*(t) = [\mathbf{R}(\mathbf{x}(t), \tilde{\mathbf{v}}(t))]^{-1} \mathbf{b}(\mathbf{x}(t), u(t)), \quad (2)$$

where the matrix $\mathbf{R}$ depends on the state vector $\mathbf{x}$ and an intermediate approximation $\tilde{\mathbf{v}}(t)$, while the vector $\mathbf{b}$ is parameterized by the state and the input $u(t)$. Since the algebraic variables $\mathbf{v}(t)$ are calculated analytically, the resulting system is converted into an ODE system, which is much easier to solve than a DAE system. However, this method relies on several simplifying assumptions, such as a linearized expression for the activation overpotential (see detailed discussions in Section 3.1.1), and it is derived based on a specific discretization approach. In addition, the resulting stiff ODE system was solved using a variable time-step algorithm (ode23ts in MATLAB), which is not convenient for real-time implementation or for the development of advanced control and management algorithms. Due to the stiffness of the system, fixed-step simulation requires very small time steps,

typically less than 1 s, making the method computationally expensive, particularly for long-term simulations (e.g., aging studies) or large-scale batch simulations involving many battery cells.

To address the aforementioned drawbacks, this paper proposes a computationally-efficient, single-step numerical scheme to solve the P2D model with the following innovative ideas.

We first show that the original PDAE system of the P2D model can be reformulated as a system of partial integro-differential equations (PIDEs). We emphasize that the term *reformulation* in this work is used differently from its usage in the literature [34]. Here, it is not primarily intended for model reduction, but rather facilitates the identification of a general *fixed-point iterated operator* $\mathcal{G}$ to address the challenging algebraic constraints in the P2D model, i.e.,

$$v_{k+1}(x, t) = \mathcal{G}(\chi(x, t), u(t), v_k(x, t)), \quad (3)$$

where x is the spatial index, and $\chi(x, t)$ and $v(x, t)$ represent the spatiotemporal variables corresponding to the state vector $\mathbf{x}(t)$ and the algebraic variables $\mathbf{v}(t)$, respectively. Once the operator $\mathcal{G}$ is established, spatial discretization is applied to convert the PIDE system into an DAE system. Within this framework, the algebraic variables can be computed through a fixed-point iteration using the following update rule:

$$\mathbf{v}_{k+1}(t) = \mathbf{g}(\mathbf{x}(t), u(t), \mathbf{v}_k(t)), \quad (4)$$

where $\mathbf{g}(\cdot)$ is a vector-valued iterated function corresponding to the iterated operator $\mathcal{G}$. A single-step time-integration scheme with a closed-form update is developed to enable fast and accurate numerical integration in the time domain.

The main advantages of the proposed method are summarized as follows, among which the benefits of the PIDE-based nonlinear AE solver compared with some well known solvers in Table 1. We will demonstrate these advantages in later sections.

(1) The proposed method solves for the algebraic states using an analytical fixed-point iterated function $\mathbf{g}(\cdot)$. As a gradient-free algorithm, it does not require computation or approximate of the Jacobian, simplifying the implementation and reducing computational cost.

(2) The iteration exhibits approximately linear asymptotic convergence and demonstrate minimized sensitivity to initial guess of the algebraic variables. These characteristics ensure that only up to 2 iterations are required for practical applications with very dynamic load conditions, ensuring very lower computational burden and suitable for large-scale system simulation.

(3) The use of a single-step implicit-explicit method enables significantly larger time step sizes compared to traditional solvers. Numerical experiments confirm that stability can be maintained with time steps up to 30 s.

(4) The reformulation does not rely on simplifying model assumptions and is compatible with various spatial discretization schemes and time integration methods, enhancing its adaptability across applications.

The proposed algorithm can be readily used for testing battery performance and proficient in control algorithm design. The algorithm is superior to many existing methods especially when simulating the behaviors of the battery under dynamic current profiles.

2. Overview of the P2D model

2.1. Overview of electrochemical P2D model

The P2D model of Li-ion batteries was established based on the porous electrode theory and the concentrated solution theory [11–13]. It is a physics-based model which reflect the sandwich-like structure of the battery cell with three domains, namely the positive electrode (i.e. the cathode, denoted by “+”), the negative electrode (i.e. the anode, denoted by “−”), and the separator (sep) in between, as illustrated

in Fig. 1, and L^+ , L^- , and L^{sep} represent the width of corresponding domain. The lithium species are stored in a number of assumed spherical particles in the solid phase of the electrode, and transport of lithium species during charge/discharge cause the variation of the lithium concentrations in the particles as well as in the electrolyte. On the one hand, the P2D model describes the cell behaviors in the horizontal axis (x -direction) on the macro scale. As indicated in Fig. 1, we denote $0^+ = 0$, $\delta^+ = L^+$, $\delta^- = L^+ + L^{\text{sep}}$, and $0^- = L^+ + L^{\text{sep}} + L^-$ as the positions of four boundaries on the macro scale, and $\Omega^+ = [0^+, \delta^+]$, $\Omega^{\text{sep}} = [\delta^+, \delta^-]$, and $\Omega^- = [\delta^-, 0^-]$ as the three domains. On the other hand, the P2D model also describe the particle-level behavior along the pseudo radial axis (r -direction) on the micro scale, with $r \in [0, R_p^+]$ in the positive electrode and $r \in [0, R_p^-]$ in the negative electrode, where R_p^+ and R_p^- correspond to the surfaces of the assumed spherical particles. The model consists of the following tightly coupled and nonlinear PDAEs:

$$\frac{\partial c_e^j}{\partial t} = \frac{1}{\epsilon_e^j} \frac{\partial}{\partial x} \left(D_{e,\text{eff}}^j \frac{\partial c_e^j}{\partial x} \right) + \frac{1-t_+}{F \epsilon_e^j} \frac{\partial i_e^j}{\partial x}, \quad (5)$$

$$\frac{\partial c_s^\pm}{\partial t} = \frac{1}{r^2} \frac{\partial}{\partial r} \left(D_{s,\text{eff}}^\pm r^2 \frac{\partial c_s^\pm}{\partial r} \right), \quad (6)$$

$$\frac{\partial \Phi_e^j}{\partial x} = -\frac{i_e^j}{\kappa_{\text{eff}}^j} + \beta \frac{RT}{F} \left(1 + \frac{d \ln f_{+/-}^j}{d \ln c_e^j} \right) \frac{\partial \ln c_e^j}{\partial x}, \quad (7)$$

$$\frac{\partial i_e^\pm}{\partial x} = F a_s^\pm j_n^\pm, \quad \frac{\partial i_e^{\text{sep}}}{\partial x} = 0, \quad (8)$$

$$\frac{\partial \Phi_s^\pm}{\partial x} = -\frac{i_s^\pm}{\sigma_{\text{eff}}^\pm}, \quad (9)$$

$$\frac{\partial i_s^\pm}{\partial x} = -F a_s^\pm j_n^\pm, \quad (10)$$

$$j_n^\pm = \frac{i_0^\pm}{F} \left[\exp \left(\frac{\alpha F \eta_{\text{ct}}^\pm}{RT} \right) - \exp \left(-\frac{(1-\alpha) F \eta_{\text{ct}}^\pm}{RT} \right) \right], \quad (11)$$

$$\eta_{\text{ct}}^\pm = \Phi_s^\pm - \Phi_e^\pm - U_{\text{ss}}^\pm - F r_{f,\text{eff}}^\pm j_n^\pm, \quad (12)$$

subject to boundary conditions

$$\frac{\partial c_e^\pm}{\partial x} \Big|_{x=0^\pm} = 0, \quad D_{e,\text{eff}}^\pm \frac{\partial c_e^\pm}{\partial x} \Big|_{x=\delta^\pm} = D_{e,\text{eff}}^{\text{sep}} \frac{\partial c_e^{\text{sep}}}{\partial x} \Big|_{x=\delta^\pm}, \quad (13)$$

$$\frac{\partial c_s^\pm}{\partial r} \Big|_{r=0} = 0, \quad D_{s,\text{eff}}^\pm \frac{\partial c_s^\pm}{\partial r} \Big|_{r=R_p^\pm} = -j_n^\pm, \quad (14)$$

$$\frac{\partial \Phi_e^\pm}{\partial x} \Big|_{x=0^\pm} = 0, \quad \kappa_{\text{eff}}^\pm \frac{\partial \Phi_e^\pm}{\partial x} \Big|_{x=\delta^\pm} = \kappa_{\text{eff}}^{\text{sep}} \frac{\partial \Phi_e^{\text{sep}}}{\partial x} \Big|_{x=\delta^\pm}, \quad (15)$$

$$\sigma_{\text{eff}}^\pm \frac{\partial \Phi_s^\pm}{\partial x} \Big|_{x=0^\pm} = -i_{\text{app}}, \quad \frac{\partial \Phi_s^\pm}{\partial x} \Big|_{x=\delta^\pm} = 0, \quad (16)$$

and initial conditions

$$c_s^\pm|_{t=0} = c_{s0}^\pm, \quad c_e^\pm|_{t=0} = c_e^0, \quad (17)$$

where superscripts $j \in \{+, -, \text{sep}\}$ and $\pm \in \{+, -\}$ denote the domain in which the equation applies to. Electrolyte concentrations $c_e^j(x, t)$ and solid-phase concentrations $c_s^\pm(x, r, t)$ are spatio-temporal state variables. The surface solid-phase concentration is defined as $c_{s0}^\pm(x, t) := c_s^\pm(x, R_p^\pm, t)$. $\Phi_e^j(x, t)$, $\Phi_s^\pm(x, t)$, $i_e^j(x, t)$, $i_s^\pm(x, t)$, $j_n^\pm(x, t)$, and $\eta_{\text{ct}}^\pm(x, t)$ are spatio-temporal algebraic states. The applied current density $i_{\text{app}}(t)$ is defined as positive when charging and negative when discharging. Model parameters are functions of concentrations and/or battery temperature, i.e., $U_{\text{ss}}^\pm = f_1^\pm(c_{\text{ss}}^\pm, T)$, $i_0^\pm = f_2^\pm(c_{\text{ss}}^\pm, c_e^\pm, T)$, $D_{e,\text{eff}}^j = f_3^j(c_e^j, T)$, $\kappa_{\text{eff}}^j = f_4^j(c_e^j, T)$, $D_{s,\text{eff}}^\pm = f_5^\pm(T)$, $\sigma_{\text{eff}}^\pm = f_6^\pm(T)$, $r_{f,\text{eff}}^\pm = f_7^\pm(T)$, and $k_{\text{eff}}^\pm = f_8^\pm(T)$. $\beta = 2(1-t_+)$ is assumed to be a constant. Detailed expressions of these functions are provided in Appendix A.

By solving the presented PDAEs, other variables, such as the battery terminal voltage and SOC, can be calculated as model outputs. For example, the terminal voltage of the cell is calculated by

$$V_{\text{bat}}(t) = \Phi_s^+|_{x=0^+} - \Phi_s^-|_{x=0^-} + r_{\text{col}} i_{\text{app}}. \quad (18)$$

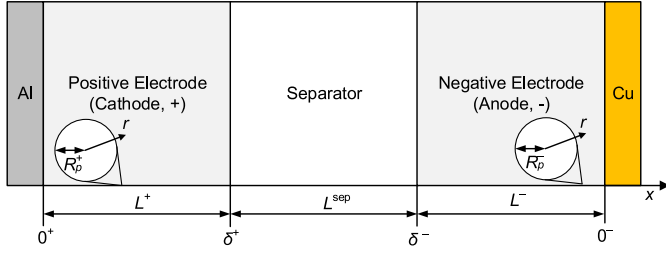


Fig. 1. Schematic of the P2D model structure of Li-ion batteries.

The physical meanings and the units of all symbols used in the P2D model are given in Nomenclature. More details about the P2D model can be found in various literature, such as [32,33,35].

2.2. Thermal model

In an isothermal P2D model, the temperature T is considered as a constant parameter. A thermal model can be added to describe how T changes over time during battery operation. Since thermal modeling is not the focus of this paper, we use a simple lumped thermal model for demonstrating how the proposed reformulation of P2D model can affect the thermal model in latter sections. This lumped thermal model is given by

$$\sum_{j \in \{+, -, \text{sep}\}} (L^j \rho^j c_p^j) \frac{dT}{dt} = h(T_{\text{amb}} - T) + q_{\text{tot}}, \quad (19)$$

where T_{amb} is the ambient temperature, ρ is the mass density, c_p is the specific heat capacity, h is the heat transfer coefficient, and “ $\forall j$ ” means “ $j \in \{+, -, \text{sep}\}$ ”. The total heat flux q_{tot} generated by the battery is expressed as

$$q_{\text{tot}} = q_{\text{rxn}} + q_{\text{rev}} + q_{\text{ohm}} + q_{\text{sei}} + q_{\text{col}}, \quad (20)$$

where q_{rxn} , q_{rev} , q_{ohm} , q_{sei} , and q_{col} , are heat flux due to reaction, reversible entropy change, ohmic effects, solid-electrolyte interface (SEI) film, and the current collectors, given by

$$q_{\text{rxn}} = \sum_{\pm} \int_{\Omega^{\pm}} F a_s^{\pm} j_n^{\pm} \eta_{\text{ct}}^{\pm} dx, \quad (21)$$

$$q_{\text{rev}} = \sum_{\pm} \int_{\Omega^{\pm}} F a_s^{\pm} j_n^{\pm} T \frac{\partial U_{\text{ss}}^{\pm}}{\partial T} dx, \quad (22)$$

$$q_{\text{ohm}} = \sum_{\pm} \int_{\Omega^{\pm}} -i_s^{\pm} \frac{\partial \Phi_s^{\pm}}{\partial x} dx + \sum_{\pm} \int_{\Omega^{\pm}} -i_e^j \frac{\partial \Phi_e^j}{\partial x} dx, \quad (23)$$

$$q_{\text{sei}} = \sum_{\pm} \int_{\Omega^{\pm}} a_s^{\pm} (F j_n^{\pm})^2 r_{f,\text{eff}}^{\pm} dx, \quad (24)$$

$$q_{\text{col}} = i_{\text{app}}^2 r_{\text{col}}, \quad (25)$$

where “ $\forall \pm$ ” means “ $\pm \in \{+, -\}$ ”.

3. Methodology

As mentioned in the introduction, the MOL first converts the PDAE into a DAE system using spatial discretization. The PDEs (5) and (6) are reduced to ordinary differential equations (ODEs), whereas the spatial constraints (7)–(12) are converted to a set of AEs. A DAE solver is then required to solve the system in the time domain. Usually, DAE solvers use generic iterative methods which involves calculating the Jacobian of the DAE model and/or careful selection of initial guess of the solution. This can significantly reduce the computational speed and compromise numerical stability, being particularly problematic when the Jacobian differs greatly between successive time steps. This issue is pronounced during dynamic charging or discharging conditions with frequent current variations, where stored Jacobian matrices become less useful, as they do not closely resemble the new values needed for the next step.

3.1. A PIDE representation of P2D model

A problem-specific fixed-point iteration, though, is simple to implement and can be computationally very efficient. However, finding an iterated function is usually not straightforward for such a complex system. In this section, we show that for the P2D model, a form of iterated function (3) can be expressed as

$$j_n^{\pm} \leftarrow G(c_s^{\pm}, c_e^{\pm}, T, i_{\text{app}}, j_n^{\pm}), \quad (26)$$

where $c_s^{\pm}$, $c_e^{\pm}$, and T belong to the state variable $\mathcal{X}$, i_{app} is the input variable u , and $j_n^{\pm}$ is the algebraic variable v . Note that $\Phi_s^{\pm}$, $\Phi_e^{\pm}$, $i_s^{\pm}$, $i_e^{\pm}$, and $\eta_{\text{ct}}^{\pm}$ in the original P2D model are intermittent states that shall be eliminated in our reformulated model.

3.1.1. Reformulated kinetic reaction equation

According to the Butler–Volmer (BV) Eq. (11), the charge-transfer overpotential can be expressed as:

$$\eta_{\text{ct}}^{\pm} = \frac{2RT}{F} \sinh_{\alpha}^{-1}(\phi), \quad (27)$$

where $\phi = \frac{F j_n^{\pm}}{2i_{\text{ct0}}^{\pm}}$ denotes the normalized molar flux, and $\sinh_{\alpha}(\cdot)$ is a deformed hyperbolic sine function, defined by

$$\sinh_{\alpha}(\phi) := \frac{\exp(2\alpha\phi) - \exp(-2(1-\alpha)\phi)}{2}. \quad (28)$$

We can then introduce the charge-transfer resistance r_{ct} as

$$r_{\text{ct}}^{\pm} = \frac{\eta_{\text{ct}}^{\pm}}{F j_n^{\pm}} = \frac{RT}{F i_{\text{ct0}}^{\pm}} \frac{\sinh_{\alpha}^{-1}(\phi)}{\phi} = r_{\text{ct0}}^{\pm} \frac{\phi}{\sinh_{\alpha}(\phi)}, \quad (29)$$

where $r_{\text{ct0}}^{\pm} = \frac{RT}{F i_{\text{ct0}}^{\pm}}$ denotes the steady-state charge-transfer resistance. Note that (29) is only defined at $j_n^{\pm} \neq 0$ and $\phi \neq 0$. Considering that there is a removable discontinuity point $\lim_{\phi \rightarrow 0} (\phi / \sinh_{\alpha}(\phi)) = 1$, we define the charge-transfer resistance as

$$r_{\text{ct}}^{\pm}(j_n^{\pm}) = r_{\text{ct0}}^{\pm} \text{isinhc}_{\alpha}(\phi(j_n^{\pm})), \quad (30)$$

where the function $\text{isinhc}_{\alpha}(\phi)$ is defined as

$$\text{isinhc}_{\alpha}(\phi) = \begin{cases} 1, & \phi = 0 \\ \frac{\phi}{\sinh_{\alpha}(\phi)}, & \phi \neq 0. \end{cases}$$

Here, $\text{isinhc}_{\alpha}(\phi)$ describes the relationship between the normalized charge-transfer resistance, $r_{\text{ct}}^{\pm}/r_{\text{ct0}}^{\pm}$, and the normalized molar flux at different α ,

Note that in most existing literature, $r_{\text{ct}}^{\pm}$ is obtained by linearizing the BV equation [33] and/or assuming $\alpha = 0.5$ [25–28], which results in $r_{\text{ct}}^{\pm} \equiv r_{\text{ct0}}^{\pm}$, while this approximation is seen inaccurate under high current conditions especially when α deviates significantly from 0.5 (see Fig. 2).

Furthermore, we define the sum of $r_{\text{ct}}^{\pm}$ and $r_{f,\text{eff}}^{\pm}$ as a new resistance, r_{Σ} , which is a function of the molar flux, denoted by

$$r_{\Sigma}^{\pm}(j_n^{\pm}) := r_{\text{ct}}^{\pm}(j_n^{\pm}) + r_{f,\text{eff}}^{\pm}. \quad (31)$$

With (29) and (31), (12) can be written as

$$\Phi_s^{\pm} - \Phi_e^{\pm} - U_{\text{ss}}^{\pm} - F r_{\Sigma}^{\pm} j_n^{\pm} = 0. \quad (32)$$

3.1.2. Reformulated electrolyte potential equation

Next, we define two potential terms, $U_e^j(x, t)$ and $\Psi_e^j(x, t)$, as follows

$$U_e^j := \beta \frac{RT}{F} \ln \left(\frac{f_{+/-}^j c_e^j}{f_{+/-}^0 c_e^0} \right), \quad (33)$$

$$\Psi_e^j := \Phi_e^j - U_e^j, \quad (34)$$

where $f_{+/-}^0$ is the nominal value of the activity coefficient $f_{+/-}^j$ when $c_e^j(x, t) = c_e^0$ (i.e., the steady-state value).

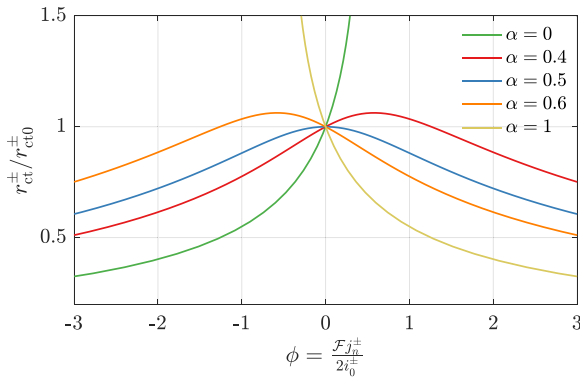


Fig. 2. Relationship between the normalized areal charge-transfer resistance and the normalized molar flux at different α .

Eq. (33) shows U_e^j is a function of the local electrolyte concentration c_e^j and temperature T . The gradient of (33) is

$$\begin{aligned} \frac{\partial U_e^j}{\partial x} &= \beta \frac{RT}{F} \frac{\partial}{\partial x} \left[\ln \left(\frac{f_{+/-}^j c_e^j(x, t)}{f_{+/-}^0 c_e^0} \right) \right] \\ &= \beta \frac{RT}{F} \frac{\partial \ln(f_{+/-}^j c_e^j)}{\partial x} \\ &= \beta \frac{RT}{F} \left(1 + \frac{d \ln f_{+/-}^j}{d \ln c_e^j} \right) \frac{\partial \ln c_e^j}{\partial x}. \end{aligned} \quad (35)$$

Substituting (34) and (35) into (7) yields

$$\frac{\partial \Psi_e^j}{\partial x} = -\frac{i_e^j}{\kappa_{\text{eff}}^j}. \quad (36)$$

Note that (36) is in agreement with that derived in [33], although in [33], the term $\frac{d \ln f_{+/-}^j}{d \ln c_e^j}$ in (35) was omitted by assuming $f_{+/-}^j \equiv f_{+/-}^0$.

3.1.3. Integro-differential equations for charge conservation

Substituting (34) into (32), we have

$$\Phi_s^\pm - \Psi_e^\pm - U_{\text{se}}^\pm - F r_{\Sigma}^\pm j_n^\pm = 0, \quad (37)$$

where

$$U_{\text{se}}^\pm = U_{\text{ss}}^\pm + U_e^\pm.$$

With (9) and (36), the gradient of (37) can be derived as:

$$\begin{aligned} \frac{\partial U_{\text{se}}^\pm}{\partial x} &= \frac{\partial \Phi_s^\pm}{\partial x} - \frac{\partial \Psi_e^\pm}{\partial x} - \frac{\partial (F r_{\Sigma}^\pm j_n^\pm)}{\partial x} \\ &= -\frac{i_s^\pm}{\sigma_{\text{eff}}^\pm} + \frac{i_e^\pm}{\kappa_{\text{eff}}^\pm} - \frac{\partial (F r_{\Sigma}^\pm j_n^\pm)}{\partial x}. \end{aligned} \quad (38)$$

Using (8) and boundary conditions, one can derive

$$i_e^\pm = F a_s^\pm \int_{0^\pm}^x j_n^\pm(x', t) dx', \quad (39)$$

$$i_e^{\text{sep}} = i_{\text{app}}. \quad (40)$$

Using (10) and boundary conditions, we have

$$i_s^\pm = i_{\text{app}} - F a_s^\pm \int_{0^\pm}^x j_n^\pm(x', t) dx'. \quad (41)$$

Substituting (39) and (41) into (38) yields an integro-differential equation (IDE):

$$a_s^\pm \rho^\pm \int_{0^\pm}^x j_n^\pm dx' - \frac{\partial (r_{\Sigma}^\pm j_n^\pm)}{\partial x} = \frac{1}{F} \left(\frac{i_{\text{app}}}{\sigma_{\text{eff}}^\pm} + \frac{\partial U_{\text{se}}^\pm}{\partial x} \right), \quad (42)$$

where $\rho^\pm = \frac{1}{\sigma_{\text{eff}}^\pm} + \frac{1}{\kappa_{\text{eff}}^\pm}$ denotes the combined resistivity of solid phase and electrolyte.

Each of the IDEs is subject to an integral boundary condition (IBC), which can be obtained by evaluating (39) at the electrode/separators boundary, i.e.,

$$F a_s^\pm \int_{0^\pm}^{\delta^\pm} j_n^\pm dx' = i_e^\pm|_{x=\delta^\pm} = i_e^{\text{sep}} = i_{\text{app}}. \quad (43)$$

To this end, the spatial constraints (7)–(12) and their BCs (15) and (16) are reformulated to the PIDE (42) and its IBC (43). Along with two PDEs modified from (5), (6), and (8), the reformulated P2D model as a PIDE system is summarized in Tables 2 and 3 for completeness. It can be seen that the algebraic states $\Phi_e^\pm$, $\Phi_s^\pm$, $i_e^\pm$, $i_s^\pm$, and $\eta_{\text{ct}}^\pm$ in the original P2D model are eliminated in this PIDE system, which significantly simplifies the model complexity.

3.2. A fixed-point iterative method for solving the IDEs

The structure of the reformulated P2D model as a PIDE system is illustrated in Fig. 3. It shows that the PDEs (5) and (6) update the solid-phase concentration $c_s^\pm$ and the electrolyte concentration c_e^j solely based on the molar flux $j_n^\pm$. Since there are various existing techniques for solving the diffusion PDEs (5) and (6), they are not the focus of the present study. In contrast, solving the IDEs is not straightforward and has not been discussed in the literature for the P2D model. Specifically, it can be seen that it needs $U_{\text{ss}}^\pm$, $U_e^\pm$, i_{app} , and $r_{\Sigma}^\pm$ to solve for $j_n^\pm$. Here, $U_{\text{ss}}^\pm$ and $U_e^\pm$ can be calculated as nonlinear functions of states $c_{\text{ss}}^\pm$, $c_e^\pm$, and T , while $r_{\Sigma}^\pm$ depends on the algebraic variable $j_n^\pm$. As a consequence, there is no closed-form solution for $j_n^\pm$. Hence, we focus our discussion on how to solve for the unknown $j_n^\pm$ iteratively.

We now propose a generic fixed-point iterative method for solving the IDEs (42) and (43). First, for ease of notation, the following linear operation $\mathcal{L}$ is defined on a spatiotemporal variable $v(x, t)$, parameterized by $\beta_1(x, t)$ and $\beta_2(x, t)$:

$$\mathcal{L}(v; \beta_1, \beta_2) := \beta_1 \int_{0^\pm}^x v(x', t) dx' + \beta_2 \frac{\partial v}{\partial x}. \quad (44)$$

Assuming $r_{\Sigma}^\pm$ is known, the IDE (42) can be expressed as

$$\mathcal{L}(v; \beta_1, \beta_2) = \beta_0, \quad (45)$$

where

$$v = j_n^\pm, \quad (46)$$

$$\beta_0 = \frac{1}{F} \left(\frac{i_{\text{app}}}{\sigma_{\text{eff}}^\pm} + \frac{\partial U_{\text{se}}^\pm}{\partial x} \right), \quad (47)$$

$$\beta_1 = a_s^\pm \rho^\pm, \quad (48)$$

$$\beta_2 = -r_{\Sigma}^\pm. \quad (49)$$

Note that (47)–(49) are only applicable for $x \in [0^\pm, \delta^\pm]$. For $x = \delta^\pm$, we need to use the IBC (43) to obtain

$$\beta_0 = i_{\text{app}}, \quad (50)$$

$$\beta_1 = F a_s^\pm, \quad (51)$$

$$\beta_2 = 0. \quad (52)$$

In (45), if $\mathcal{L}$ has a unique inverse operation with respect to $j_n^\pm$, and if all parameters are known, we can solve the equation by its inverse operation $\mathcal{L}^{-1}$, denoted by

$$j_n^\pm(x, t) = \mathcal{L}^{-1}(\beta_0(x, t); \beta_1(x, t), \beta_2(x, t)). \quad (53)$$

One can view (53) as the iterated operation in (3), i.e., $\mathcal{G} = \mathcal{L}^{-1}$. This is because, according to (29)–(31), $r_{\Sigma}^\pm$ in (49) is parameterized by $j_n^\pm$, and thus $j_n^\pm$ is a fixed point. Therefore, given an initial guess $j_{n,0}^\pm(x, t)$, we can solve $j_n^\pm(x, t)$ iteratively:

$$r_{\Sigma, k}^\pm = r_{\text{ct0}}^\pm \text{isinhc}_a \left(F j_{n, k}^\pm / (2i_0^\pm) \right), \quad (54)$$

Table 2
Reformulated P2D model as a PIDE systems.

PDE	BC
$\frac{\partial c_s^\pm}{\partial t} = \frac{1}{\epsilon_s^\pm} \frac{\partial}{\partial x} \left(D_{s,\text{eff}}^\pm \frac{\partial c_s^\pm}{\partial x} \right) + \frac{1-l_s}{\epsilon_s^\pm} a_s^\pm j_n^\pm,$ $\frac{\partial c_{s,\text{sep}}}{\partial t} = \frac{1}{\epsilon_{s,\text{sep}}} \frac{\partial}{\partial x} \left(D_{s,\text{eff}}^{\text{sep}} \frac{\partial c_{s,\text{sep}}}{\partial x} \right),$ $\frac{\partial c_e^\pm}{\partial t} = \frac{1}{r^\pm} \frac{\partial}{\partial r} \left(D_{s,\text{eff}}^\pm r^2 \frac{\partial c_e^\pm}{\partial r} \right),$	$\frac{\partial c_s^\pm}{\partial x} \Big _{x=0^\pm} = 0, D_{s,\text{eff}}^\pm \frac{\partial c_s^\pm}{\partial x} \Big _{x=\delta^\pm} = D_{s,\text{eff}}^{\text{sep}} \frac{\partial c_{s,\text{sep}}}{\partial x} \Big _{x=\delta^\pm},$ $D_{s,\text{eff}}^{\text{sep}} \frac{\partial c_{s,\text{sep}}}{\partial x} \Big _{x=\delta^\pm} = D_{s,\text{eff}}^\pm \frac{\partial c_s^\pm}{\partial x} \Big _{x=\delta^\pm},$ $\frac{\partial c_e^\pm}{\partial r} \Big _{r=0} = 0, D_{s,\text{eff}}^\pm \frac{\partial c_e^\pm}{\partial r} \Big _{r=R_p^\pm} = -j_n^\pm$
IDE	IBC
$a_s^\pm \theta^\pm \int_{0^\pm}^x j_n^\pm dx' - \frac{\partial(r_{\Sigma}^\pm j_n^\pm)}{\partial x} = \frac{1}{F} \left(\frac{i_{\text{sep}}}{\sigma_{\text{eff}}^\pm} + \frac{\partial U_{\Sigma}^\pm}{\partial x} \right),$	$F a_s^\pm \int_{0^\pm}^{\delta^\pm} j_n^\pm dx' = i_{\text{app}}$
Additional Equations	
$U_{\text{se}}^\pm = U_{\text{ss}}^\pm + U_e^\pm = f_1^\pm(c_{\text{ss}}^\pm, T) + \beta \frac{RT}{F} \ln \left(\frac{f_{s,j}^\pm c_{\text{ss}}^\pm}{f_{s,j}^0 c_e^0} \right),$	$\theta^\pm = \frac{1}{\sigma_{\text{eff}}^\pm} + \frac{1}{\kappa_{\text{eff}}^\pm}, \quad r_{\Sigma}^\pm = r_{\text{ct0}}^\pm \text{isinhc}_a \left(\frac{F j_n^\pm}{2i_0^\pm} \right) + r_{f,\text{eff}}^\pm$

Table 3
Output equations of the reformulated P2D model.

$i_e^\pm = F a_s^\pm \int_{0^\pm}^x j_n^\pm(x', t) dx', \quad i_e^{\text{sep}} = i_{\text{app}}, \quad i_s^\pm = i_{\text{app}} - F a_s^\pm \int_{0^\pm}^x j_n^\pm(x', t) dx'$ $V_{\text{bat}} = (U_{\text{se}}^\pm + F r_{\Sigma}^\pm j_n^\pm) \Big _{x=0^+} - (U_{\text{se}}^- + F r_{\Sigma}^- j_n^-) \Big _{x=0^-} + \sum_{\pm} F a_s^\pm \int_{\Omega^\pm} \int_{0^\pm}^x \frac{j_n^\pm}{\kappa_{\text{eff}}^\pm} dx' dx + \left(\int_{\Omega^{\text{sep}}} \frac{1}{\kappa_{\text{eff}}^{\text{sep}}} dx + r_{\text{col}} \right) i_{\text{app}}.$

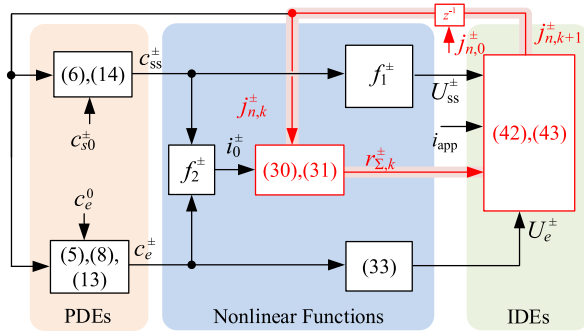


Fig. 3. Block diagram of the proposed reformulated P2D model as a system of PIDEs. The red lines indicate the algebraic loop that can be used for iteratively calculating $j_n^\pm$. The z^{-1} block indicates the memory for iterative calculating the algebraic states in one time step (rather than time delay), and it involves an initial guess, $j_{n,0}^\pm$.

$$j_{n,k+1}^\pm = \mathcal{L}^{-1}(\beta_0; \beta_1, \beta_2(r_{\Sigma,k}^\pm)). \quad (55)$$

Here, the subscript k represents the k th iteration. The iterations terminate when a stopping criteria is met, for example, by evaluating the norm

$$\left\| 1 - \frac{j_{n,k}^\pm(x, t)}{j_{n,k-1}^\pm(x, t)} \right\|_2 \leq \epsilon, \quad (56)$$

where $0 < \epsilon \ll 1$ is the termination tolerance and we denote the corresponding iteration number as N_{iter} . Therefore, the solution $j_n^{\pm*} = j_{n,k=N_{\text{iter}}}^\pm$.

The loop highlighted in red in Fig. 3 indicates the path where the molar flux is iteratively solved. Since this method does not require to calculate the local potentials, current densities, and charge-transfer overpotentials, the computational burden is significantly reduced than the original P2D model.

3.3. Reformulated voltage equation

Since the PIDE in Table 2 does not involve potential terms, the battery voltage Eq. (18) needs to be modified.

Using (34), the voltage Eq. (18) can be rewritten as

$$V_{\text{bat}} = (U_{\text{se}}^+ + F r_{\Sigma}^+ j_n^+) \Big|_{x=0^+} - (U_{\text{se}}^- + F r_{\Sigma}^- j_n^-) \Big|_{x=0^-} + (\Psi_e^+ \Big|_{x=0^+} - \Psi_e^- \Big|_{x=0^-}) + r_{\text{col}} i_{\text{app}}. \quad (57)$$

Integrating (36) gives

$$\Psi_e^+ \Big|_{x=0^-} - \Psi_e^- \Big|_{x=0^+} = \sum_{\forall j} \int_{\Omega^j} \frac{i_e^j}{\kappa_{\text{eff}}^j} dx. \quad (58)$$

Substituting (58) into (57) and considering the expressions i_e^j in (39)–(40), the battery voltage Eq. (18) can be expressed as

$$V_{\text{bat}} = (U_{\text{se}}^\pm + F r_{\Sigma}^\pm j_n^\pm) \Big|_{x=0^+} - (U_{\text{se}}^- + F r_{\Sigma}^- j_n^-) \Big|_{x=0^-} + F a_s^+ \int_{\Omega^+} \int_{0^+}^x \frac{j_n^+}{\kappa_{\text{eff}}^+} dx' dx + F a_s^- \int_{\Omega^-} \int_{0^-}^x \frac{j_n^-}{\kappa_{\text{eff}}^-} dx' dx + \left(\int_{\Omega^{\text{sep}}} \frac{1}{\kappa_{\text{eff}}^{\text{sep}}} dx + r_{\text{col}} \right) i_{\text{app}}. \quad (59)$$

3.4. Reformulated thermal model

Considering (7), (9), and (33), the ohmic heat flux Eq. (23) can be simplified to

$$q_{\text{ohm}} = \sum_{\forall \pm} \int_{\Omega^\pm} \frac{(i_s^\pm)^2}{\sigma_{\text{eff}}^\pm} dx + \sum_{\forall j} \int_{\Omega^j} \left[\frac{(i_e^j)^2}{\kappa_{\text{eff}}^j} - \frac{\partial U_e^j}{\partial x} i_e^j \right] dx, \quad (60)$$

and $i_s^\pm$ and i_e^j can be further obtained using $j_n^\pm$ and i_{app} according to (39)–(41).

Considering (21), (24), (29), and (31), the sum of the reaction heat flux and SEI heat flux can be simplified to

$$q_{\text{rxn}} + q_{\text{sei}} = \sum_{\forall \pm} \int_{\Omega^\pm} F^2 a_s^\pm r_{\Sigma}^\pm (j_n^\pm)^2 dx. \quad (61)$$

Therefore, the total heat generation is a function of $j_n^\pm$, i_{app} , U_e^j , and $\partial U_{\text{ss}}^\pm / \partial T$. The relationship between the reformulated thermal model and the isothermal P2D model is shown in Fig. 4.

4. Numerical solution method

4.1. Solving IDEs

The IDEs derived in Section 3 can be solved using various techniques of numerical integration and differentiation. We use (42) to demonstrate the method to solve the IDEs. We divide the positive electrode domain into N^+ control volumes, denoted by $\{1, 2, \dots, N^+\}$, along the electrode thickness, and $x_{i+\frac{1}{2}}$ denotes the boundary between element i and $i+1$. The width of the i th control volume is denoted as l_i . The

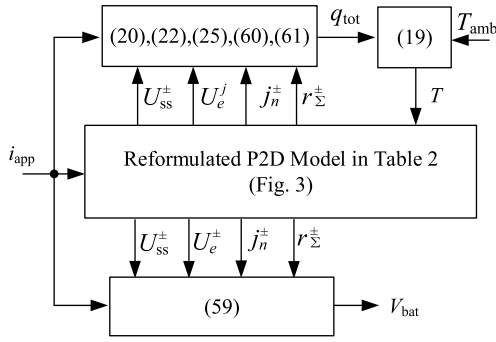


Fig. 4. Block diagram for the reformulated P2D model coupled with a thermal model and its output voltage.

following discretization rules are applied to (42) at the interface of its i th and $(i + 1)$ th control volumes, $i = \{1, 2, \dots, N^+ - 1\}$:

$$\int_0^{x_{i+\frac{1}{2}}} X(x, t) dx = \sum_{n=1}^i X_n(t) l_n, \quad (62)$$

$$\frac{\partial X}{\partial x} \Big|_{x=x_{i+\frac{1}{2}}} = \frac{X_{i+1} - X_i}{0.5(l_i + l_{i+1})}, \quad (63)$$

$$X|_{x=x_{i+\frac{1}{2}}} = \frac{l_i + l_{i+1}}{\frac{l_i}{X_i} + \frac{l_{i+1}}{X_{i+1}}}, \quad (64)$$

which yields $N^+ - 1$ linear equations (an underdetermined system) as given in the matrix form (65).

$$\begin{bmatrix} r_{\Sigma,1}^+ + a_s^+ l_1^+ \frac{(\phi_{s,1}^+ l_1^+ + \phi_{s,2}^+ l_2^+)}{2} & -r_{\Sigma,2}^+ & 0 \\ a_s^+ l_1^+ \frac{(\phi_{s,1}^+ l_1^+ + \phi_{s,2}^+ l_2^+)}{2} & r_{\Sigma,2}^+ + a_s^+ l_2^+ \frac{(\phi_{s,2}^+ l_2^+ + \phi_{s,3}^+ l_3^+)}{2} & -r_{\Sigma,3}^+ \\ a_s^+ l_1^+ \frac{(\phi_{s,1}^+ l_1^+ + \phi_{s,2}^+ l_2^+)}{2} & a_s^+ l_2^+ \frac{(\phi_{s,2}^+ l_2^+ + \phi_{s,3}^+ l_3^+)}{2} & r_{\Sigma,3}^+ + a_s^+ l_3^+ \frac{(\phi_{s,3}^+ l_3^+ + \phi_{s,4}^+ l_4^+)}{2} \\ \vdots & \vdots & \vdots \\ a_s^+ l_1^+ \frac{(\phi_{s,N^+-1}^+ l_{N^+-1}^+ + \phi_{s,N^+}^+ l_{N^+}^+)}{2} & a_s^+ l_2^+ \frac{(\phi_{s,N^+-1}^+ l_{N^+-1}^+ + \phi_{s,N^+}^+ l_{N^+}^+)}{2} & a_s^+ l_3^+ \frac{(\phi_{s,N^+-1}^+ l_{N^+-1}^+ + \phi_{s,N^+}^+ l_{N^+}^+)}{2} \\ \dots & 0 & 0 \\ \dots & 0 & 0 \\ \dots & 0 & 0 \\ \ddots & \vdots & \vdots \\ \dots & r_{\Sigma,N^+-1}^+ + a_s^+ l_{N^+-1}^+ \frac{(\phi_{s,N^+-1}^+ l_{N^+-1}^+ + \phi_{s,N^+}^+ l_{N^+}^+)}{2} & -r_{\Sigma,N^+}^+ \end{bmatrix} \begin{bmatrix} j_{n,1}^+ \\ j_{n,2}^+ \\ j_{n,3}^+ \\ \vdots \\ j_{n,N^+}^+ \end{bmatrix} = \frac{1}{2F} \begin{bmatrix} \frac{l_1^+}{\sigma_{eff,1}^+} + \frac{l_2^+}{\sigma_{eff,2}^+} \\ \frac{l_2^+}{\sigma_{eff,2}^+} + \frac{l_3^+}{\sigma_{eff,3}^+} \\ \vdots \\ \frac{l_{N^+-1}^+}{\sigma_{eff,N^+-1}^+} + \frac{l_{N^+}^+}{\sigma_{eff,N^+}^+} \end{bmatrix} i_{app} + \frac{1}{F} \begin{bmatrix} -1 & 1 & 0 & \dots & 0 & 0 \\ 0 & -1 & 1 & \dots & 0 & 0 \\ 0 & 0 & -1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -1 & 1 \end{bmatrix} \begin{bmatrix} U_{se,1}^+ \\ U_{se,2}^+ \\ U_{se,3}^+ \\ \vdots \\ U_{se,N^+}^+ \end{bmatrix} \quad (65)$$

In (65), $\mathbf{j}^+ = [j_{n,1}^+, j_{n,2}^+, \dots, j_{n,N^+}^+]^T$ is a vector containing all local volume-averaged molar fluxes, $\mathbf{u}^+ = [U_{se,1}^+, U_{se,2}^+, \dots, U_{se,N^+}^+]^T$ is a vector containing all local volume-averaged potentials, and $\mathbf{L}^+$, $\mathbf{b}^+$, and $\mathbf{C}^+$ are matrices or vector of appropriate dimensions associated with local resistive components.

The IBC gives the N th equation:

$$(F a_s^+) \mathbf{1}_{N^+}^T \mathbf{j}^+ = i_{app}, \quad (66)$$

where $\mathbf{1}_{N^+} \in \mathbb{R}^{N^+}$ is a unit vector.

Table 4

Comparison of the components in the general form and the FVM implementation in the iterated function for the positive electrode.

Symbol	General	FVM
$\mathcal{L}$	(48), (49), (51), (52)	$\begin{bmatrix} \mathbf{L}^+ \\ (F a_s^+) \mathbf{1}_{N^+}^T \end{bmatrix}$
$v(x, t)$	$j_n(x, t)$	$\mathbf{j}^+$
$\beta_0(x, t)$	(47), (50)	$\begin{bmatrix} \mathbf{b}^+ \\ 1 \end{bmatrix} i_{app} + \begin{bmatrix} \mathbf{C}^+ \\ \mathbf{0}_{N^+}^T \end{bmatrix} \mathbf{u}^+$

Due to the quasilinear nature, (65) and (66) can be solved simultaneously by matrix inversion, i.e.,

$$\mathbf{j}^+ = \begin{bmatrix} \mathbf{L}^+ \\ (F a_s^+) \mathbf{1}_{N^+}^T \end{bmatrix}^{-1} \left(\begin{bmatrix} \mathbf{b}^+ \\ 1 \end{bmatrix} i_{app} + \begin{bmatrix} \mathbf{C}^+ \\ \mathbf{0}_{N^+}^T \end{bmatrix} \mathbf{u}^+ \right), \quad (67)$$

where $\mathbf{0}_{N^+} \in \mathbb{R}^{N^+}$ is a zero vector. This represents the FVM implementation of the iterated function (53). The correspondence between the general form and its FVM implementation for the positive electrode is summarized in Table 4. A similar procedure can be applied to calculate $\mathbf{j}^- = [j_{n,1}^-, j_{n,2}^-, \dots, j_{n,N^+}^-]^T$ for the negative electrode, but the details are omitted here for brevity.

4.2. Solving PDEs

As can be seen from Fig. 3, the IDEs are discretized into AEs and solved iteratively together with the nonlinear functions. The inputs of this process, i.e., the solid-phase and electrolyte concentrations, are obtained by solving the PDEs, while the outputs of the AEs (molar fluxes) serve as inputs of the PDEs. Consequently, the numerical scheme used to solve the PDEs must provide the information required by the AE solver.

4.2.1. Solving the electrolyte diffusion equation

A generic state-space representation for solving the diffusion equations is given by

$$\dot{\mathbf{x}}_e(t) = \mathbf{A}_e \mathbf{x}_e(t) + \mathbf{B}_e \mathbf{j}(t), \quad (68)$$

$$\mathbf{c}_e(t) = \mathbf{C}_e \mathbf{x}_e(t), \quad (69)$$

where $\mathbf{x}_e$ denotes the state vector of the electrolyte diffusion equation, $\mathbf{c}_e$ is the vector of local electrolyte concentrations, and $\mathbf{j} = [\mathbf{j}^+, \mathbf{j}^-]^T$ collects all local molar fluxes. The matrices $\mathbf{A}_e$, $\mathbf{B}_e$, and $\mathbf{C}_e$ are parameterized by system state variables, such as $\mathbf{x}_e$ and the battery temperature. It should be noted that this representation omits the feedthrough term in order to avoid algebraic loops between the input and output of the diffusion equation, which would otherwise restrict the choice of numerical algorithms. In the present study, for the FVM scheme adopted in the previous section, a compatible scheme for solving the electrolyte concentration Eq. (5) is also using the FVM method. The corresponding expressions of the matrices $\mathbf{A}_e$, $\mathbf{B}_e$, and $\mathbf{C}_e$ are omitted for brevity.

Note that the ODE system derived from the FVM method is typically highly stiff. To enhance numerical stability, we discretize the ODE at time t using an implicit-explicit (IMEX) scheme, as described in [9]:

$$\frac{\mathbf{x}_e(t + T_s) - \mathbf{x}_e(t)}{T_s} = \mathbf{A}_e \mathbf{x}_e(t + T_s) + \mathbf{B}_e \mathbf{j}(t), \quad (70)$$

where T_s denotes the sampling period. Note that this formulation does not correspond to the standard backward Euler scheme but rather an IMEX method, since $\mathbf{j}(t)$ depends on $\mathbf{x}_e(t)$. Owing to the linear structure of the system, the solution of (70) can be expressed in closed form using matrix inversion:

$$\mathbf{x}_e(t + T_s) = (\mathbf{I} - T_s \mathbf{A}_e)^{-1} [\mathbf{x}_e(t) + T_s \mathbf{B}_e \mathbf{j}(t)], \quad (71)$$

where $\mathbf{I}$ is the identity matrix of appropriate dimension.

In contrast, the Crank–Nicholson (CN) method, which is based on the trapezoidal rule, can also be expressed into closed form due to the linear structure of the equation:

$$\mathbf{x}_e(t + T_s) = \left(\mathbf{I} - \frac{T_s}{2} \mathbf{A}_e \right)^{-1} \left(\mathbf{x}_e(t) \left(\mathbf{I} + \frac{T_s}{2} \mathbf{A}_e \right) + T_s \mathbf{B}_e \mathbf{j}(t) \right). \quad (72)$$

4.2.2. Solving the solid-phase diffusion equation

Solving the solid-phase diffusion equation is more flexible. At each macro-scale location ($x = x_i$), a PDE is formulated that only involves the micro-scale coordinate, yielding $c_s^\pm(x_i, r, t) = c_{s,i}^\pm(r, t)$ and $c_{ss,i}^\pm(x_i, t) = c_{ss,i}^\pm(t)$. Various methods exist for solving this PDE, including polynomial profile approximation, Padé approximation, FVM, and finite difference method [15]. However, the choice of method must be made carefully, since as mentioned above, we aim to avoid introducing the feedthrough terms in the present algorithm. Under this requirement, the discretized model takes the following form

$$\dot{\mathbf{x}}_{s,i}^\pm(t) = \mathbf{A}_{s,i}^\pm \mathbf{x}_{s,i}^\pm(t) + \mathbf{B}_{s,i}^\pm j_{n,i}^\pm(t), \quad (73)$$

$$c_{ss,i}^\pm(t) = \mathbf{C}_{s,i}^\pm \mathbf{x}_{s,i}^\pm(t), \quad (74)$$

where the matrices $\mathbf{A}_{s,i}^\pm$, $\mathbf{B}_{s,i}^\pm$, and $\mathbf{C}_{s,i}^\pm$ are parameterized by the battery temperature and can also depend on the state vector $\mathbf{x}_{s,i}^\pm$.

Although the FVM can be applied in a manner similar to that used for the electrolyte diffusion equation, it typically leads to a high-order system to might significant increase the computational burden. To address this, we adopt the moment matching method [10]. For simplicity, we assume the order of moment matching, denoted by M , is the same for both electrode. The resulting system matrices are expressed as

$$\mathbf{A}_{s,i}^\pm = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & -a_1/\tau_{s,i}^\pm & 0 & \cdots & 0 \\ 0 & 0 & -a_2/\tau_{s,i}^\pm & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -a_M/\tau_{s,i}^\pm \end{bmatrix},$$

$$\mathbf{B}_{s,i}^\pm = \begin{bmatrix} 3 & -b_1/R_p^\pm & -b_2/R_p^\pm & \cdots & -b_M/R_p^\pm \end{bmatrix}^\top,$$

$$\mathbf{C}_{s,i}^\pm = \mathbf{1}_{1 \times (M+1)},$$

where $\tau_{s,i}^\pm = (R_p^\pm)^2/D_{s,\text{eff}}^\pm$ and $\mathbf{1}_{1 \times (M+1)}$ is an $(M+1)$ -dimensional row vector of ones. The parameters $a_1, a_2, \dots, a_M$ and $b_1, b_2, \dots, b_M$ depend on the selection of M .

5. Results and discussion

5.1. System configuration

In this section, the effectiveness of the proposed solution method is evaluated through numerical simulations. All algorithms are implemented in MATLAB R2019b and executed on a system equipped with an Intel Core i7 processor and 16 GHz of RAM.

The benchmark model used for comparison is a spatially discretized P2D model formulated as a system of DAE based on the model parameters given in Appendix B. This model is implemented using the same FVM rules as applied in the proposed framework. The number of the control volumes in the positive electrode, separator, and negative electrode domains are set to $N^+ = 10$, $N^{\text{sep}} = 5$, and $N^- = 10$, respectively. The order of moment matching is set to $M = 2$, leading to $a_1 = 20.57$, $a_2 = 168.42$, $b_1 = 2.18$, and $b_2 = 15.82$. The resulting DAE system consists of ODEs derived from (5) and (6), and AEs resulting from (7)–(12). The proposed method employs the single-step explicit technique described in (71) (whereas we still use IMEX to indicate this method) to solve the ODEs with sampling time $T_s = 1$ s. The benchmark model uses MATLAB's `fsolve` function to solve the AEs, specifically for computing the molar flux $j_n^\pm$ with high accuracy.

5.2. Initialization methods

As mentioned in Section 3.2, at time t , the iterative method for solving the nonlinear AEs needs to be initialized by providing an initial guess $j_{n,0}^\pm$ of the solution. Several initialization schemes are investigated here.

(1) Initialization with zero values:

$$j_{n,0}^\pm(x, t) = 0, \quad \forall x \in \Omega^\pm.$$

(2) Initialization with the solution in the previous time step $t - T_s$:

$$j_{n,0}^\pm(x, t) = j_n^\pm(x, t - T_s), \quad \forall x \in \Omega^\pm.$$

(3) Initialization with physically consistent uniform profile satisfying

$$j_{n,0}^\pm(x, t) = \frac{\pm i_{\text{app}}(t)}{F a_s^\pm L^\pm}, \quad \forall x \in \Omega^\pm.$$

(4) Initialization with physically consistent quadratic profile satisfying

$$j_{n,0}^\pm(x, t) = \alpha_0^\pm(t) + \alpha_1^\pm(t)x + \alpha_2^\pm(t)x^2, \quad \forall x \in \Omega^\pm,$$

where the coefficients α_0 – α_2 are determined by solving the following three equations based on the boundary conditions

$$F a_s^\pm \int_{0^\pm}^{\delta^\pm} j_{n,0}^\pm(x', t) dx' = i_{\text{app}}(t),$$

$$\left. \frac{\partial U_{\text{se}}^\pm}{\partial x} \right|_{x=0^\pm} = -\frac{i_{\text{app}}}{\sigma_{\text{eff}}^\pm} - F r_\Sigma^\pm \left. \frac{\partial j_{n,0}^\pm}{\partial x} \right|_{x=0^\pm},$$

$$\left. \frac{\partial U_{\text{se}}^\pm}{\partial x} \right|_{x=\delta^\pm} = \frac{i_{\text{app}}}{\kappa_{\text{eff}}^\pm} - F r_\Sigma^\pm \left. \frac{\partial j_{n,0}^\pm}{\partial x} \right|_{x=\delta^\pm}.$$

5.3. Convergence test

We examine the convergence behavior of the proposed fixed-point iteration method for solving the IDEs in Section 3.2. To measure and analyze both the speed and quality of convergence, the error contraction ratio q is calculated:

$$q_k(t) = \frac{\|\mathbf{j}_{k+1} - \mathbf{j}_k\|_2}{\|\mathbf{j}_k - \mathbf{j}_{k-1}\|_2}$$

which quantifies how much the iteration approaches the solution at each step. In practice, we are particularly interested in the first-step and second-step contraction ratios, q_1 and q_2 , since minimizing the number of iterations is crucial for reducing computational cost.

The nature of convergence is further characterized by the estimated order of convergence, given by

$$p_k(t) = \frac{\log \|\mathbf{j}_{k+1} - \mathbf{j}_k\|_2 - \log \|\mathbf{j}_k - \mathbf{j}_{k-1}\|_2}{\log \|\mathbf{j}_k - \mathbf{j}_{k-1}\|_2 - \log \|\mathbf{j}_{k-1} - \mathbf{j}_{k-2}\|_2}$$

Theoretically, the estimated order of convergence should converge to the true order as k increases. However, in practice, numerical limitations may reduce the accuracy when the residual becomes very small, causing the estimated order of convergence to become unstable at higher iterations. In this study, we use the estimate at $k = 8$, which was observed to be stable and representative across simulations.

To evaluate the convergence characteristics of the proposed solver, termination a very small termination tolerance is used, specifically, $\epsilon = 1 \times 10^{-20}$, to avoid premature termination of iteration process. The calculated contraction ratios and estimated order of convergence for 1C and 3C discharge tests are presented in Fig. 5. It shows that the initial contraction ratios depicted in Fig. 5(a) are very small, suggesting that the solver can reach a highly accurate solution within just first one or two iterations in practice. Furthermore, in Fig. 5(b), most of the time the estimated order of converge remains slightly below 1

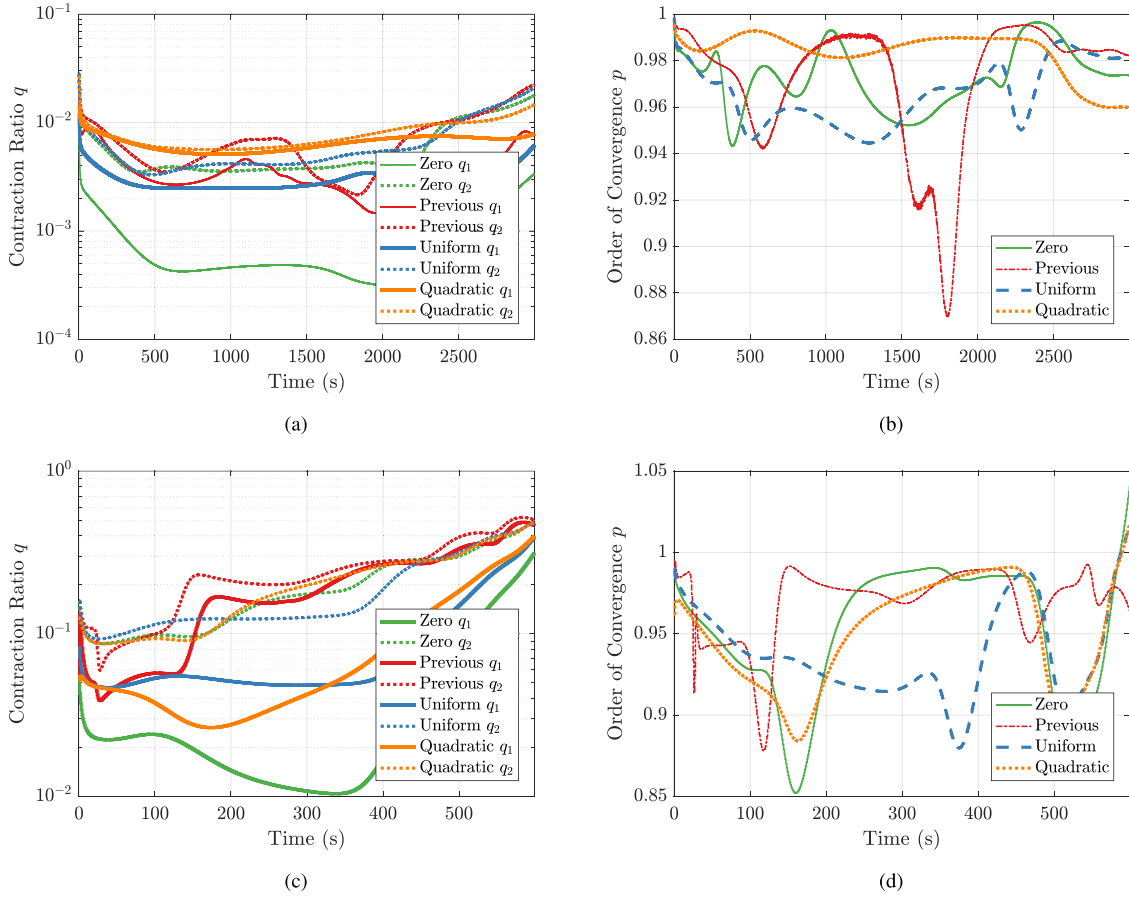


Fig. 5. Results of convergence test with different initialization methods. (a) First- and second-step contraction ratio under 1C-discharge. (b) Estimated order of convergence under 1C-discharge. (c) First- and second-step contraction ratio under 3C-discharge. (d) Estimated order of convergence under 3C-discharge.

and higher than 0.95, indicating the algorithm exhibits approximately linear convergence, comparable to many state-of-the-art AE solvers.

Although some differences are observed among the presented initialization strategies, e.g., zero initialization appears to yield a slightly better first-step contraction ratio as seen from Fig. 5(a), the overall impact of the initial guess is not significant. This indicates that the proposed AE solver exhibits low sensitivity to initial estimate of the algebraic states, which is advantageous for practice implementation.

Fig. 5(c) and (d) show the convergence performance under a 3C discharge condition. This represents an extremely high current rate, given that the maximum rated C-rate for the battery chemistry under investigation is approximately 2C. Under such higher current conditions, the battery experience stronger polarization effects and greater heterogeneity in internal state distributions (concentrations, molar fluxes, etc.). Consequently, both the contraction ratio and order of convergence are somewhat degraded in general. Nonetheless, the observed values ($0.01 < q < 0.5$, $0.85 < p < 0.99$) remain within satisfactory bounds, ensuring that the proposed fixed-point iteration still converges rapidly.

5.4. Performance comparison with existing AE solvers

In this section, the performance of the proposed gradient-free AE solver is compared with MATLAB's `fsolve` function, which is based on the trust-region dogleg algorithm combining Newton's method and gradient descent. To evaluate the trade-off between computational speed and solution accuracy, different termination tolerance values ϵ are tested. Three solver configurations are considered: (1) $\epsilon = 1 \times 10^{-5}$, (2) $\epsilon = 1 \times 10^{-3}$, both without any limit on the maximum number of iterations, and (3) a single-iteration setting, where the maximum

number of iterations is limited to one. For all cases, the molar fluxes are initialized to zero at every time step.

Figs. 6 and 7 show the simulation results under a dynamic discharging profile based on the Federated Urban Driving Schedule (FUDS). In Fig. 6, the original FUDS profile is applied, where the maximum current rate is 1C. In Fig. 7, the original current profile is amplified by a factor of three, and it is repeated applied until the voltage reaches the cut-off voltage of 3.0 V to evaluate performance under more extreme conditions. The evaluation metrics involve voltage prediction error (relative to the gradient-based `fsolve` baseline), the number of iterations, and CPU runtime per simulation time step. Table 5 further provides a summary of the comparison results.

As can be seen in Fig. 6 and Table 5, under the original FUDS current profile, all three solver settings yield high predictive accuracy compared to the benchmark. For $\epsilon = 1 \times 10^{-5}$, the number of iterations is between 1 and 4, while for $\epsilon = 1 \times 10^{-3}$, the number of iterations ranges from 1 to 3. These results suggest that both settings offer sufficiently accurate solutions, potentially exceeding the accuracy requirements for practical applications. Notably, even in the single-iteration case, the solver achieves a voltage root-mean-square error (RMSE) of less than 1×10^{-5} V, and maximum absolute error (MaxArr) of 0.151 mV, indicating the minimal iteration is often adequate.

As expected, higher current magnitudes, such as those in the amplified FUDS profile, demand for more iterations to attain the similar levels of accuracy, as shown from Fig. 7(h) and Table 5. Nevertheless, across all test conditions, the proposed solver consistently demonstrates significantly lower computational cost than `fsolve`, with CPU run-times reduced by more than one order of magnitude. This performance enhancement is primarily due to the fact that the proposed method

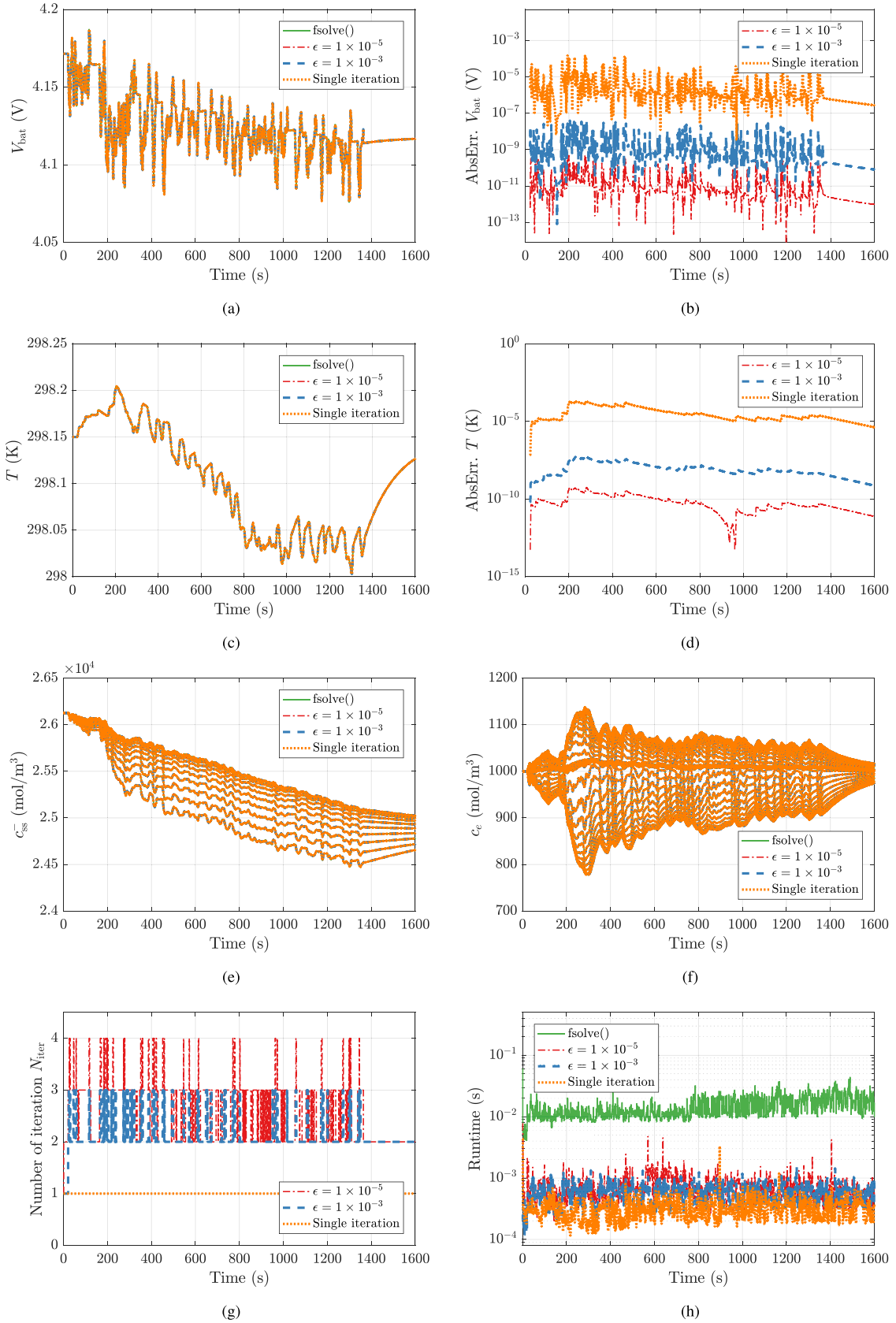


Fig. 6. Comparison of gradient-based and proposed gradient-free AE solvers under FUDS profile with the maximum current of 1C. (a) Voltage. (b) Voltage error. (c) Battery temperature. (d) Temperature error. (e) Solid-phase concentrations in the negative electrode. (f) Electrolyte concentrations. (g) Iteration number. (h) CPU runtime for solving AEs.

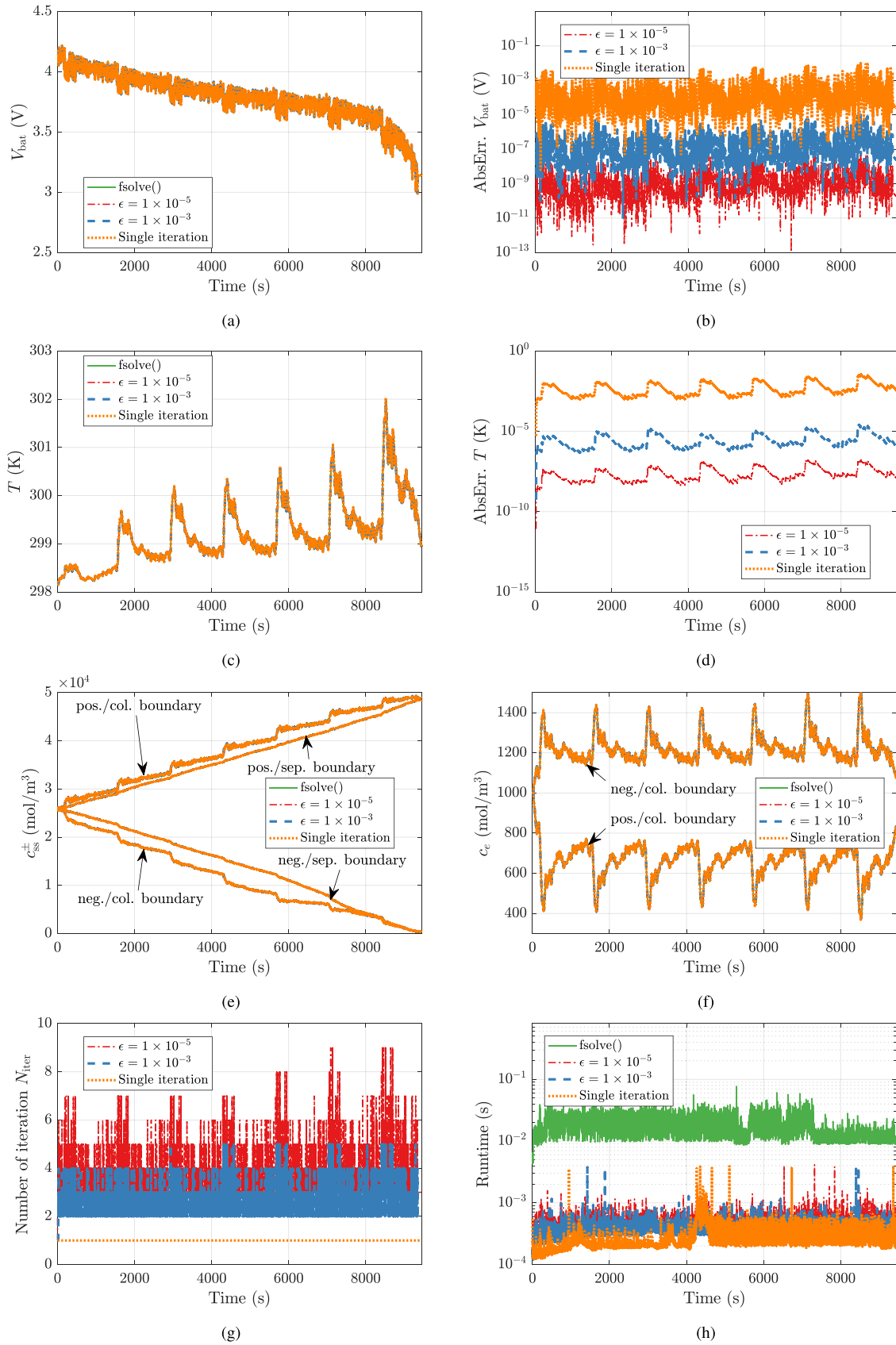


Fig. 7. Comparison of gradient-based and proposed gradient-free AE solvers under a repeated, modified FUDS profile, where the magnitude of the current is amplified by 3 times. (a) Voltage. (b) Voltage error. (c) Battery temperature. (d) Temperature error. (e) Solid-phase concentrations at electrode boundaries. (f) Electrolyte concentrations at electrode/current collector (col.) boundaries. (g) Iteration number. (h) CPU runtime for solving AEs.

Table 5
Numerical performance comparison of gradient-based and proposed gradient-free AE solvers.

Current Profile	Solver Config.	Voltage (V)		Iteration		CPU runtime (s)	
		RMSE	MaxArr	Average	Max	Average	Max
FUDS (Orig.)	fsolve	–	–	–	–	0.0105	0.029
	(1)	1.87×10^{-11}	5.90×10^{-10}	2.67	4	0.00035	0.0037
	(2)	1.86×10^{-9}	4.73×10^{-8}	2.08	3	0.00029	0.0013
	(3)	4.87×10^{-6}	1.51×10^{-4}	1	1	0.00016	0.0007
FUDS (Mod.)	fsolve	–	–	–	–	0.0155	0.0466
	(1)	1.51×10^{-9}	7.14×10^{-8}	3.67	9	0.00068	0.0045
	(2)	1.82×10^{-7}	1.06×10^{-5}	2.57	5	0.00051	0.0044
	(3)	2.48×10^{-4}	1.05×10^{-2}	1	1	0.00025	0.0038

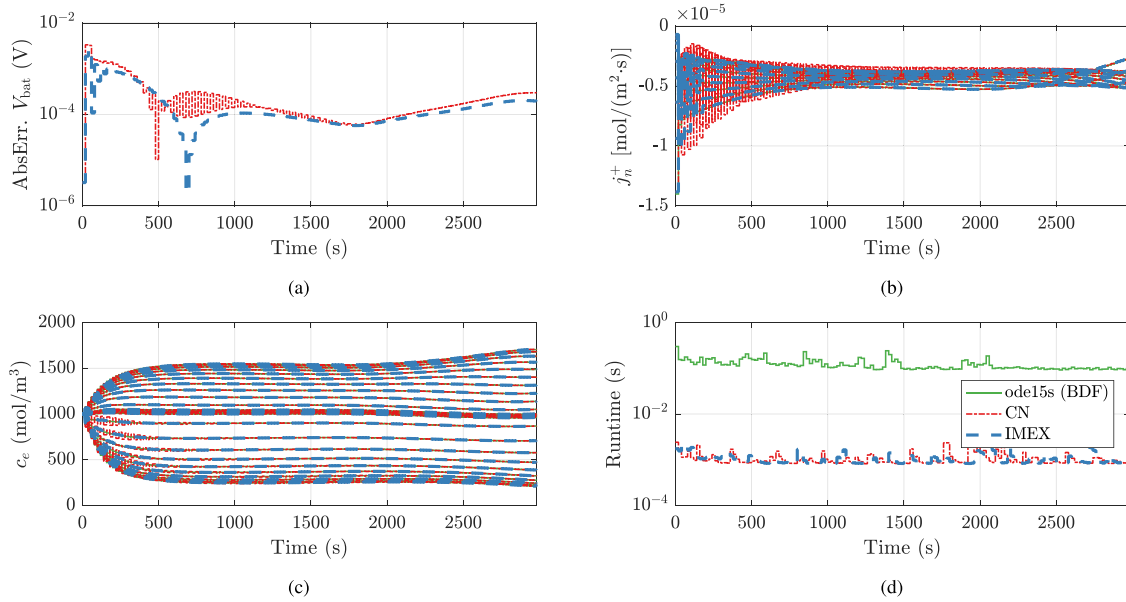


Fig. 8. Comparison of BDF, CN, and IMEX methods under a 1C discharging with a simulation time step of 20 s. (a) Voltage error. (b) Molar flux in the positive electrode. (c) Electrolyte concentration. (d) CPU Runtime for solving the full battery models.

only solves a linear equation for each electrode at every iteration and eliminates the requirement for Jacobian evaluation.

It is worth noting that MATLAB's `fsolve` is internally optimized using C/Fortran via MEX. The proposed algorithm, if similarly compiled, is expected to achieve further improvements in computational efficiency. This aspect, however, is left for future investigation, while the current results already demonstrate significant advantages.

5.5. Comparison of time integration schemes

In previous section, all simulation were conducted using the FVM implementation described in Sections 4.1 and 4.2, with the time-domain integration based on the single-step explicit method derived from the IMEX formulation, as defined in (71).

In this section, we compare that scheme with two alternatives: the CN scheme described in (72) and a widely used multistep method for solving the stiff DAEs, i.e. the BDF, implemented in MATLAB's `ode15s` function. To test each method's capability to handle longer time steps (important for accelerating large-scale battery system simulations), the integration step size is increased to 20 s. Based on previous findings, the number of fixed-point iterations is limited to two for both the IMEX and CN schemes, with each iteration initialized with zero guess of molar fluxes. Simulation results are presented in Fig. 8.

Although not shown in the illustrated examples, the IMEX and CN methods exhibits similar accuracy when using small step sizes (e.g., 1 s), as used in the previous study cases. However, as the step size increases, the CN scheme begins to exhibit oscillatory behavior in

the calculated molar flux and the electrolyte concentration, quantities that contribute significantly to the stiffness of the P2D model. Further increasing the step size leads to numerical instability and crash of the simulation. In contrast, the IMEX scheme remains stable and deliver accurate voltage predictions, achieving an RMSE below 0.02 mV.

In addition, as expected, both the IMEX and CN schemes with the proposed gradient-free framework demand much lower computational burden, i.e., about two orders of magnitude lower than the BDF method, thanks to their single-step nature and avoidance of Jacobian evaluations.

6. Conclusion

This paper presents a numerical framework for solving the pseudo-two-dimensional (P2D) model of lithium-ion batteries using a gradient-free, single-step implicit-explicit (IMEX) method. A general iterative formulation is derived by reformulating the original partial differential-algebraic equation (PDAE) system into a partial integro-differential equation (PIDE) system. The PIDE system is then spatially discretized using the FVM method at the macro-scale, and a single-step numerical scheme with a closed-form update is developed for time-domain integration, leveraging the quasi-linear structure of the discretized model. Simulation results demonstrate that the proposed method yields numerically stable and robust solutions while significantly reducing computational cost. The framework consistently outperforms several widely used solvers in battery research community, which offers a

promising approach for large-scale simulation and real-time applications. Future work will focus on incorporating more advanced techniques, such as spectral-based methods such as the Galerkin method and physics-informed data-driven techniques, into the solution of the PIDE system. These enhancements are expected to further improve computational efficiency while maintaining high accuracy within the proposed framework.

CRedit authorship contribution statement

Yang Li: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Torsten Wik:** Writing – review & editing, Supervision, Investigation, Funding acquisition. **Qingbo Zhu:** Writing – review & editing, Validation, Investigation, Formal analysis. **Yao Cai:** Validation, Investigation, Formal analysis. **Changfu Zou:** Writing – review & editing, Supervision, Investigation, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was funded by the Compel platform in Chalmers University of Technology and the Swedish Research Council (Grant No. 2023-04314).

Appendix A. Additional model equations

The following nonlinear functions are used in the P2D model to describe the temperature- and concentration-dependent parameters.

$$f_1^\pm(c_{ss}^\pm, T) = U^\pm \pm \frac{\partial U_{ss}^\pm}{\partial T} (T - T_{ref}), \quad (A.1)$$

$$f_2^\pm(c_{ss}^\pm, c_e^\pm, T) = F k_{eff}^\pm (c_e^\pm)^\alpha (c_{s,max}^\pm - c_{ss}^\pm)^\alpha (c_{ss}^\pm)^{1-\alpha}, \quad (A.2)$$

$$f_3^j(c_e^j, T) = (\epsilon_e^j)^{brugg^j} D_{e0}^j, \quad (A.3)$$

$$f_4^j(c_e^j, T) = (\epsilon_e^j)^{brugg^j} \kappa_0^j, \quad (A.4)$$

$$f_5^\pm(T) = D_{s0}^\pm \exp\left(-\frac{E_{a,D_s}^\pm}{R} \left(\frac{1}{T} - \frac{1}{T_{ref}}\right)\right), \quad (A.5)$$

$$f_6^\pm(T) = \sigma_0^\pm \exp\left(-\frac{E_{a,\sigma}^\pm}{R} \left(\frac{1}{T} - \frac{1}{T_{ref}}\right)\right), \quad (A.6)$$

$$f_7^\pm(T) = r_{f0}^\pm \exp\left(-\frac{E_{a,r_f}^\pm}{R} \left(\frac{1}{T} - \frac{1}{T_{ref}}\right)\right), \quad (A.7)$$

$$f_8^\pm(T) = k_0^\pm \exp\left(-\frac{E_{a,k}^\pm}{R} \left(\frac{1}{T} - \frac{1}{T_{ref}}\right)\right), \quad (A.8)$$

where $U^\pm = f_9^\pm(c_{ss}^\pm/c_{s,max}^\pm)$, $\frac{\partial U_{ss}^\pm}{\partial T} = f_{10}^\pm(c_{ss}^\pm/c_{s,max}^\pm)$, $D_{e0}^j = f_{11}(c_e^j, T)$, and $\kappa_0^j = f_{12}(c_e^j, T)$. The nonlinear functions $f_9^\pm$ and $f_{10}^\pm$ are based on electrodes' materials, f_{11} and f_{12} are based on electrolyte's material, and they are usually fitted from experimental data.

Appendix B. Model parameters

The Li-ion battery electrochemical parameters are obtained from [32,33] and are given in Table B.6.

Data availability

Data will be made available on request.

Table B.6

Electrochemical parameters of Li-ion battery.

Sym.	Parametric value/expression		
	+	sep	-
$R_p^\pm$	2×10^{-6}	–	2×10^{-6}
$D_{s0}^\pm$	1.0×10^{-14}	–	3.9×10^{-14}
$a_e^\pm$	8.85×10^5	–	7.236×10^5
L^j	8.0×10^{-5}	2.5×10^{-5}	8.8×10^{-5}
$\epsilon_e^\pm$	0.59	–	0.4824
ϵ_e^-	0.385	0.724	0.485
$\sigma_0^\pm$	100	–	100
$brugg^j$	4	4	4
$c_{s,max}^\pm$	51,554	–	30,555
$c_{s0}^\pm$	25,545	–	26,128
$k_0^\pm$	2.334×10^{-11}	–	5.031×10^{-11}
ρ^j	2500	1100	2500
c_p^j	700	700	700
$E_{a,D_s}^\pm$	5000	–	5000
$E_{a,\sigma}^\pm$	5000	–	5000
$E_{a,r_f}^\pm$	5000	–	5000
$E_{a,k}^\pm$	5000	–	5000
$f_{f0}^\pm$	0	–	0.01
F	96,487	–	–
T_{ref}	298.15	–	–
R	8.314	–	–
c_e^0	1000	–	–
t_+	0.364	–	–
$f_{+/-}$	1	–	–
h	2	–	–
f_9^+	$\frac{-4.656+88.669\theta^2-401.119\theta^4+342.909\theta^6-462.471\theta^8+433.434\theta^{10}}{-14.18.933\theta^2-79.532\theta^4+37.311\theta^6-73.083\theta^8+95.96\theta^{10}}$		
f_9^-	$\frac{0.7222+0.1387\theta+0.029\theta^3-0.0172\theta^5+0.0019\theta^7-0.2808 \exp(0.9-15\theta)}{0.7984 \exp(0.4465\theta-0.4108)}$		
f_{10}^+	$\frac{-0.001(0.199521039-0.928373822\theta+1.364550689\theta^2-0.611544894\theta^3)}{1-5.661479887\theta+11.47636191\theta^2-9.824312136\theta^3+3.048755063\theta^4}$		
f_{10}^-	$\frac{0.001 \begin{pmatrix} 0.005269056+3.299265709\theta-91.79325798\theta^2 \\ +1004.911008\theta^3-5812.278127\theta^4+19329.7549\theta^5 \\ -37147.8947\theta^6+38379.18127\theta^7-16515.05308\theta^8 \end{pmatrix}}{1-48.09287227\theta+1017.234804\theta^2-10481.80419\theta^3 \\ +59431.3\theta^4-195881.6488\theta^5+374577.3152\theta^6 \\ -385821.1607\theta^7+165705.8597\theta^8}$		
f_{11}	$10^{-4} \times c_e \begin{pmatrix} -10.5+0.668 \times 10^{-3}c_e+0.494 \times 10^{-6}c_e^2 \\ +(0.074-1.78 \times 10^{-5}c_e-8.86 \times 10^{-10}c_e^2)/T \\ +(-6.96 \times 10^{-5}+2.8 \times 10^{-8}c_e)/T^2 \end{pmatrix}$		
f_{12}	$10^{-4} \times 10^{\frac{-4.43-}{T-229-5.0 \times 10^{-3}c_e}-0.22 \times 10^{-3}c_e}$		

References

- [1] T. Kim, W. Song, D.-Y. Son, L.K. Ono, Y. Qi, Lithium-ion batteries: outlook on present, future, and hybridized technologies, *J. Mat. Chem. A* 7 (2019) 2942–2964.
- [2] Y. Li, M. Vilathgamuwa, S.S. Choi, B. Xiong, J. Tang, Y. Su, Y. Wang, Design of minimum cost degradation-conscious lithium-ion battery energy storage system to achieve renewable power dispatchability, *Appl. Energy* 260 (2020) <https://doi.org/10.1016/j.apenergy.2019.114282>.
- [3] Z. Wei, K. Liu, X. Liu, Y. Li, L. Du, F. Gao, Multi-level data-driven battery management: From internal sensing to big data utilization, *IEEE Trans. Transp. Electrification* 9 (4) (2023) 4805–4823, <https://doi.org/10.1109/TTE.2023.3301990>.
- [4] U. Krewer, F. Röder, E. Harinath, R.D. Braatz, B. Bedürftig, R. Findeisen, Dynamic models of Li-ion batteries for diagnosis and operation: A review and perspective, *J. Electrochem. Soc.* 165 (16) (2018) A3656–A3673.
- [5] Y. Wang, J. Tian, Z. Sun, L. Wang, R. Xu, M. Li, Z. Chen, A comprehensive review of battery modeling and state estimation approaches for advanced battery management systems, *Renew. Sustain. Energy Rev.* 131 (2020) 110015.
- [6] Y. Li, T. Wik, Y. Huang, C. Zou, Nonlinear model inversion-based output tracking control for battery fast charging, *IEEE Trans. Control Syst. Technol.* 32 (1) (2024) 225–240, <https://doi.org/10.1109/TCST.2023.3306240>.
- [7] X. Hu, Y. Zheng, D.A. Howey, H. Perez, A. Foley, M. Pecht, Battery warm-up methodologies at subzero temperatures for automotive applications: Recent advances and perspectives, *Prog. Energy Combust. Sci.* 77 (2020) 100806.
- [8] N.A. Chaturvedi, R. Klein, J. Christensen, J. Ahmed, A. Kojic, Algorithms for advanced battery-management systems, *IEEE Control Syst. Mag.* 30 (3) (2010) 49–68.
- [9] Y. Li, Z. Wei, C. Xie, D.M. Vilathgamuwa, Physics-based model predictive control for power capability estimation of lithium-ion batteries, *IEEE Trans. Ind. Informat.* 19 (11) (2023) 10763–10774, <https://doi.org/10.1109/TII.2022.3233676>.

- [10] Y. Li, T. Wik, C. Xie, Y. Huang, B. Xiong, J. Tang, C. Zou, Control-oriented modeling of all-solid-state batteries using physics-based equivalent circuits, *IEEE Trans. Transp. Electrification*, 8 (2) (2022) 2080–2092, <http://dx.doi.org/10.1109/TTE.2021.3131147>.
- [11] M. Doyle, T.F. Fuller, J. Newman, Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell, *J. Electrochem. Soc.* 140 (6) (1993) 1526–1533.
- [12] T.F. Fuller, M. Doyle, J. Newman, Simulation and optimization of the dual lithium ion insertion cell, *J. Electrochem. Soc.* 141 (1) (1994) 1–10.
- [13] M. Doyle, J. Newman, Modeling the performance of rechargeable lithium-based cells: design correlations for limiting cases, *J. Power Sources* 54 (1) (1995) 46–51.
- [14] A. Jokar, B. Rajabloo, M. Désilets, M. Lacroix, Review of simplified pseudo-two-dimensional models of lithium-ion batteries, *J. Power Sources* 327 (2016) 44–55, <http://dx.doi.org/10.1016/j.jpowsour.2016.07.036>.
- [15] Y. Li, D. Karunathilake, D.M. Vilathgamuwa, Y. Mishra, T.W. Farrell, S.S. Choi, C. Zou, Model order reduction techniques for physics-based lithium-ion battery management: A survey, *IEEE Ind. Electron. Mag.* 16 (3) (2022) 36–51, <http://dx.doi.org/10.1109/MIE.2021.3100318>.
- [16] D. Zhang, B.N. Popov, R.E. White, Modeling lithium intercalation of a single spinel particle under potentiodynamic control, *J. Electrochem. Soc.* 147 (3) (2000) 831–838, <http://dx.doi.org/10.1149/1.1393279>.
- [17] D. Zhang, L.D. Couto, S.J. Moura, Electrode-level state estimation in lithium-ion batteries via Kalman decomposition, *IEEE Control. Syst. Lett.* 5 (5) (2021) 1657–1662, <http://dx.doi.org/10.1109/LCSYS.2020.3042751>.
- [18] N. Lotfi, R.G. Landers, J. Li, J. Park, Reduced-order electrochemical model-based SOC observer with output model uncertainty estimation, *IEEE Trans. Control Syst. Technol.* 25 (4) (2017) 1217–1230, <http://dx.doi.org/10.1109/TCST.2016.2598764>.
- [19] M. Guo, G. Sikha, R.E. White, Single-particle model for a lithium-ion cell: thermal behavior, *J. Electrochem. Soc.* 158 (2) (2011) A122–A132, <http://dx.doi.org/10.1149/1.3521314>.
- [20] S.J. Moura, N.A. Chaturvedi, M. Krstic, Adaptive partial differential equation observer for battery state-of-charge/state-of-health estimation via an electrochemical model, *J. Dyn. Syst. Meas. Control.* 136 (1) (2014) 011015, <http://dx.doi.org/10.1115/1.4024801>.
- [21] R. Klein, N.A. Chaturvedi, J. Christensen, J. Ahmed, R. Findeisen, A. Kojic, Electrochemical model based observer design for a lithium-ion battery, *IEEE Trans. Control Syst. Technol.* 21 (2) (2011) 289–301.
- [22] A.M. Bizeray, S. Zhao, S.R. Duncan, D.A. Howey, Lithium-ion battery thermal-electrochemical model-based state estimation using orthogonal collocation and a modified extended Kalman filter, *J. Power Sources* 296 (2015) 400–412.
- [23] M. Corno, N. Bhatt, S.M. Savaresi, M. Verhaegen, Electrochemical model-based state of charge estimation for Li-Ion cells, *IEEE Trans. Control Syst. Technol.* 23 (1) (2015) 117–127.
- [24] J. Sturm, H. Ennifar, S.V. Erhard, A. Rheinfeld, S. Kosch, A. Jossen, State estimation of lithium-ion cells using a physicochemical model based extended Kalman filter, *Appl. Energy* 223 (2018) 103–123.
- [25] S. Han, Y. Tang, S. Khaleghi Rahimian, A numerically efficient method of solving the full-order pseudo-2-dimensional (P2D) Li-ion cell model, *J. Power Sources* 490 (2021) 229571.
- [26] R. Han, C. Macdonald, B. Wetton, A fast solver for the pseudo-two-dimensional model of lithium-ion batteries, 2021, [arXiv:2111.09251](https://arxiv.org/abs/2111.09251).
- [27] Q. Chen, X.S. Chen, Z. Li, A fast numerical method with non-iterative source term for pseudo-two-dimension lithium-ion battery model, *J. Power Sources* 577 (2023) 233258.
- [28] I. Korotkin, S. Sahu, S.E.J. O’Kane, G. Richardson, J.M. Foster, Dandelion v1: An extremely fast solver for the Newman model of lithium-ion battery (dis)charge, *J. Electrochem. Soc.* 168 (6) (2021) 060544.
- [29] D.W. Limoge, P.Y. Bi, A.M. Annaswamy, A. Krupadanam, A reduced-order model of a lithium-ion cell using the absolute nodal coordinate formulation approach, *IEEE Trans. Control Syst. Technol.* 26 (3) (2018) 1001–1014, <http://dx.doi.org/10.1109/TCST.2017.2692743>.
- [30] S. Kosch, Y. Zhao, J. Sturm, J. Schuster, G. Mulder, E. Ayerbe, A. Jossen, A computationally efficient multi-scale model for lithium-ion cells, *J. Electrochem. Soc.* 165 (10) (2018) A2374–A2388, <http://dx.doi.org/10.1149/2.1241810jes>.
- [31] L. Xia, E. Najafi, Z. Li, H. Bergveld, M. Donkers, A computationally efficient implementation of a full and reduced-order electrochemistry-based model for Li-ion batteries, *Appl. Energy* 208 (2017) 1285–1296, <http://dx.doi.org/10.1016/j.apenergy.2017.09.025>.
- [32] M. Torchio, L. Magni, R.B. Gopaluni, R.D. Braatz, D.M. Raimondo, LIONSIMBA: A Matlab framework based on a finite volume model suitable for Li-ion battery design, simulation, and control, *J. Electrochem. Soc.* 163 (7) (2016) A1192–A1205.
- [33] Y. Li, M. Vilathgamuwa, T. Farrell, S.S. Choi, N.T. Tran, J. Teague, A physics-based distributed-parameter equivalent circuit model for lithium-ion batteries, *Electrochim. Acta* 299 (2019) 451–469, <http://dx.doi.org/10.1016/j.electacta.2018.12.167>.
- [34] P.W.C. Northrop, B. Suthar, V. Ramadesigan, S. Santhanagopalan, R.D. Braatz, V.R. Subramanian, Efficient simulation and reformulation of lithium-ion battery models for enabling electric transportation, *J. Electrochem. Soc.* 161 (8) (2014) E3149–E3157, <http://dx.doi.org/10.1149/2.018408jes>.
- [35] N.A. Chaturvedi, R. Klein, J. Christensen, J. Ahmed, A. Kojic, Algorithms for advanced battery-management systems, *IEEE Control Syst. Mag.* 30 (3) (2010) 49–68.