




## Article

# A Novel Solver for an Electrochemical–Thermal Ageing Model of a Lithium-Ion Battery

Toshan Wickramanayake <sup>\*</sup>, Mehrnaz Javadipour  and Kamyar Mehran 

Real Time Power and Control Systems Laboratory, School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, UK; k.mehran@qmul.ac.uk (K.M.)

<sup>\*</sup> Correspondence: d.g.d.wickramanayake@qmul.ac.uk

**Abstract:** To estimate the state of health, charge, power, and safety (SoX) of lithium-ion batteries (LiBs) in real time, battery management systems (BMSs) need accurate and efficient battery models. The full-order partial two-dimensional (P2D) model is a common physics-based cell-level LiB model that faces challenges for real-time BMS implementation due to the complexity of its numerical solver. In this paper, we propose a method to discretise the P2D model equations using the Finite Volume and Verlet Integration Methods to significantly reduce the computational complexity of the solver. Our proposed iterative solver uses novel convergence criteria and physics-based initial guesses to provide high fidelity for discretised P2D equations. We also include both the kinetic-limited and diffusion-limited models for Solid Electrolyte Interface (SEI) growth into an iterative P2D solver. With these SEI models, we can estimate the capacity fade in real time once the model is tuned to the cell–voltage curve. The results are validated using three different operation scenarios, including the 1C discharge/charge cycle, multiple-C-rate discharges, and the Lawrence Livermore National Laboratory dynamic stress test. The proposed solver shows at least a 4.5 times improvement in performance with less than 1% error when compared to commercial solvers.

**Keywords:** lithium-ion battery; partial two-dimensional model; finite volume method (FVM); temperature modelling; solid electrolyte interphase growth; numerical iterative solver algorithm



**Citation:** Wickramanayake, T.; Javadipour, M.; Mehran, K. A Novel Solver for an Electrochemical–Thermal Ageing Model of a Lithium-Ion Battery. *Batteries* **2024**, *10*, 126. <https://doi.org/10.3390/batteries10040126>

Academic Editor: Yong Joon Park

Received: 6 February 2024

Revised: 22 March 2024

Accepted: 2 April 2024

Published: 9 April 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Lithium-ion batteries (LiBs) are used for energy storage for a range of applications, such as electric vehicles (EVs), consumer electronics, and even grid energy storage [1]. LiBs' high energy density, long cycle life, and reliable manufacturing processes are the main reasons for their wide application. Given the scale of their utility, accurate and efficient models of LiBs have gained significant importance [2,3].

The operation management of rechargeable battery systems is typically implemented using battery management systems (BMSs). In addition to basic battery operations, a BMS estimates two important states of battery operation, i.e., state of safety (SoS) and state of health (SoH), using LiB models representing the cell's internal states [4].

A battery model is a set of equations to mimic the cell/pack operation and provide information on the state of batteries, including the state of charge (SoC), state of health (SoH), state of power (SoP), state of function (SoF), and internal cell temperature [5]. Cell-level battery models are typically classified into three types: (i) empirical models, (ii) equivalent circuit models (ECMs), and (iii) electrochemical models (EMs) [6].

Empirical models are the simplest to implement, as they are based on functions derived from statistical curve fitting to experimental measurements of battery performance. Due to the low computational cost required, empirical models have been widely used in BMSs. However, they suffer from a lack of robustness and inaccuracy in non-optimal battery conditions [7].

ECMs model a LiB as a voltage source in series with a lumped circuit element network. Thus, ECMs also tend to require low computational processing power and are often used

on advanced BMSs of EVs. ECMs, however, provide primarily high-level information about a cell, such as its SoC and cell voltage [8].

EMs are the most accurate type of cell-level battery model [9], as they represent the internal physiochemical processes within a cell. The partial two-dimensional (P2D) model is the most accurate cell-level EM [10]. The P2D model was first proposed by Doyle et al. [11,12] and, in its basic form, captures three major physiochemical processes in a cell: the diffusion of lithium in the electrodes, reduction/oxidisation (redox) reactions at the electrode/electrolyte interphase, and the diffusion of lithium ions in the electrolyte. The P2D model captures the different physiochemical phenomena using a set of coupled Partial Differential Algebraic Equations (PDAEs) with high fidelity.

The P2D model can be adapted to capture more physiochemical processes than just the three fundamental processes. Examples of this include modelling the effects of heat generation/dissipation in different regions of the cell (non-isothermal cases) [13,14] or modelling ageing in cells by capturing Solid Electrolyte Interface (SEI) growth [15,16]. The adaptations of the P2D model tend to be additive, where additional processes are modelled in conjunction with the fundamental physiochemical processes of the P2D model. Despite its accuracy and adaptability, the P2D model is a complex mathematical model that has no analytical solution [17,18]. To implement the P2D model, we normally use two approaches: (1) develop reduced-order models (ROMs) of the P2D model [10] or (2) use numerical iterative solvers to implement the full-order P2D model.

Various forms of ROMs are suggested in the literature, e.g., the Single-Particle Model (SPM) [19], the extended Single-Particle Model (eSPM) [20], Multi-Scale Multi-Domain Models (MSMD) [21], and Electrode-Averaged Models (EAMs) [22].

The most commonly used of these ROMs are the SPM and eSPM [10]. The SPM models only the electrode material domains of a battery and considers each electrode to be a spherical particle. The physics processes modelled in the SPM are primarily lithium mass conservation in the spherical electrode particles. Like the SPM, the eSPM also models the electrode material domains as spherical particles, but it also has a reduced-order model of the electrolyte of a LiB. The eSPM considers the ionic current distribution in a LiB to follow a simple linear profile, enabling an analytic solution to be determined for the eSPM equations, thus making the model computationally simple to implement [23].

These existing ROMs show high accuracy only for a range of operating conditions and low accuracy for the entire range when compared to the full-order model (FOM) version of the P2D model. In this paper, our proposed iterative solver solves the P2D model with a significant improvement in solver performance for a broad range of operating conditions.

### 1.1. Current Progress in Full-Order P2D Model Solvers

As shown in Table 1, several researchers have attempted to develop a fast and accurate full-order P2D model solver. To numerically solve the P2D model, first, the PDAEs of the model need to be discretised and converted to Differential Algebraic Equations (DAEs).

Multiple discretisation methods are employed, including the Finite Difference Method (FDM), the Finite Volume Method (FVM), and the Finite Element Method (FEM), to name a few. Existing approaches may vary based on whether they provide a fundamental isothermal P2D model or additional model parameters.

Once the chosen model equations are discretised, they are iteratively solved for a time step until convergence to the DAE solution is achieved. The convergence mechanism is non-trivial and dependent on multiple factors, including the convergence criteria, the initial guess of the solver, and the implementation method used for the DAEs in the solver algorithm. Indeed, the method of convergence of the different solvers can be considered a major factor affecting solver accuracy and performance. Hence, we aim to provide a novel convergence criterion and iterative solver structure to achieve high accuracy and performance.

**Table 1.** Summary of recent P2D solver papers.

Paper	Discretisation Method	Parameters Solved		
		Isothermal P2D	Temperature Modelling	Ageing Model
Yin et al. (2023) [24]	FVM	×	-	-
Chen et al. (2023) [25]	FDM	×	-	-
Ai et al. (2023) [26]	FEM	×	-	-
Chayambuka et al. (2022) [27]	FDM + FVM at boundaries	×	-	-
Jiang et al. (2022) [28]	FVM	×	-	-
Han et al. (2021) [29]	FVM	×	ROM	-
Geng et al. (2021) [30]	FDM	×	-	-
Han et al. (2021) [31]	FDM	×	FOM	-
Noor et al. (2021) [32]	FEM	×	-	-
Lee et al. (2021) [33]	FDM	×	-	-
Esfahanian et al. (2019) [34]	FVM	×	FOM	-
Guo et al. (2017) [35]	Nonlinear State-Variable Method	×	-	-
Torchio et al. (2016) [36]	FVM	×	FOM	-
Tulsyan et al. (2016) [37]	State-Space Method	×	-	-
Doyle et al. (1993) [11]	FDM	×	-	-
Proposed solver	FVM	×	FOM	×

### 1.2. Contributions

Based on our previous work [38], we aim to provide the following contributions in this paper:

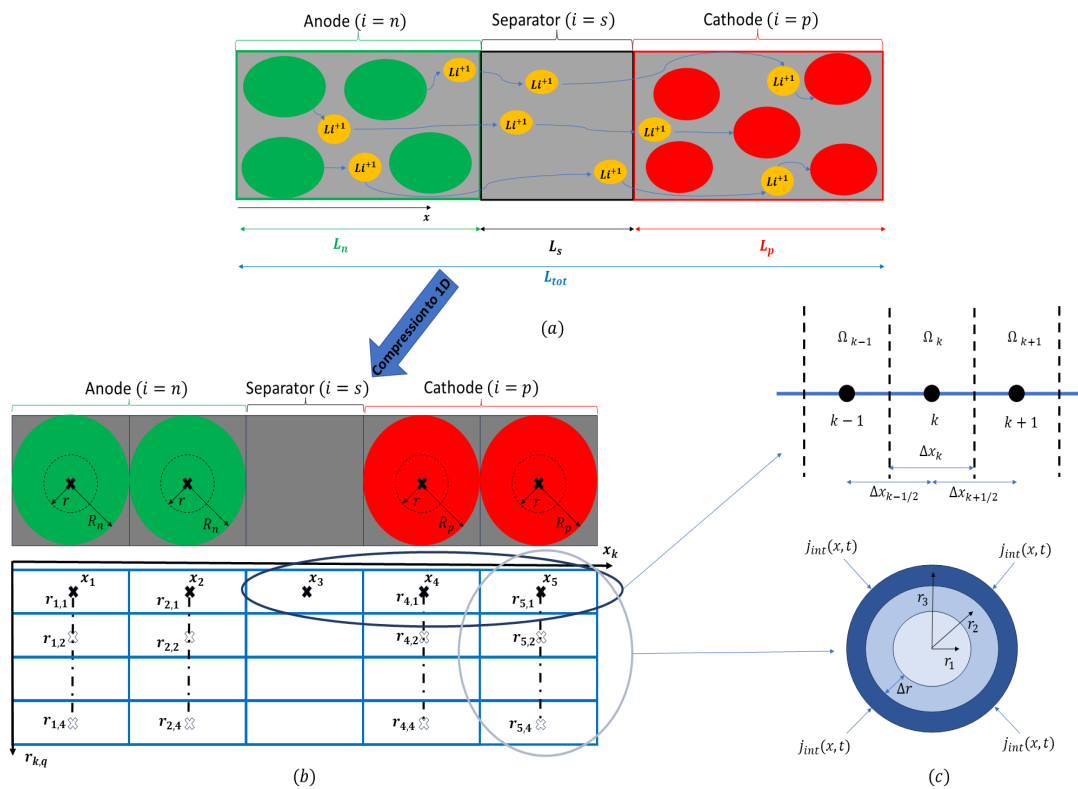
- A novel convergence criterion for solving the full-order P2D model of a LiB. The proposed solver shows at least a 4.5 times improvement in performance with less than 1% error when validated against commercial solvers. The MATLAB-based solver code [<https://github.com/twick07/Electrochemical-Thermal-P2D-Model-Iterative-Solver> (accessed on 1 April 2024)] will be open-source and available for use by other researchers in this field.
- A solver that is suitable for different LiB chemistries by providing multiple battery parameter sets for simulations.
- A full-order P2D model for temperature generation/dissipation in LiBs.
- The inclusion of multiple ageing models for the growth of the Solid Electrolyte Interface (SEI) in an iterative P2D solver: kinetic-limited and diffusion-limited models for SEI growth. These SEI models can be used to estimate the real-time capacity fade when tuned to cell performance data.

## 2. The Electrochemical–Thermal P2D Battery Model

The P2D model considers the battery to be composed of three physical material domains. These are the anode (negative electrode), the separator, and the cathode (positive electrode). We denote these material domains by the index  $i$ , such that  $i \in \{n, s, p\}$ , as shown in Figure 1a.

The P2D model also considers the battery to have two phases, namely, the solid electrode phase and the liquid electrolyte phase. The solid phase is considered to exist only in the electrode domains of  $i = \{n, p\}$ , and the electrolyte phase is considered to exist in all three domains of the battery model. The two phases in the model are denoted by the index  $j \in \{1, 2\}$ , such that the solid phase is considered when  $j = 1$  and the electrolyte phase is considered when  $j = 2$ .

Considering the dimensions of the basic P2D model, the model essentially compresses the dimensions of a cell such that the cell is one-dimensional in space (see Figure 1b). Therefore, the primary dimension is in  $x$ , which spans the entire thickness of the cell. However, in compressing the dimension of the cell, the electrode regions undergo a transformation from a three-dimensional porous structure to a one-dimensional porous structure. This 1D porous electrode structure is considered to consist of a set of spherical electrode particles at each point in  $x$ , and the particles are assumed to be surrounded by the liquid electrolyte. Hence, in the electrode domains, another ‘pseudo’ dimension is considered, which is the radial dimension,  $r$ , through the thickness of each spherical electrode particle.



**Figure 1.** An illustration of the P2D model structure and discretisation. (a) The P2D model structure, the green and red spheres represent anode and cathode electrode particles respectively and the small yellow spheres denote diffusing lithium-ions; (b) an illustration of a coarse mesh of the discretised P2D model where black cross marks denote discrete points in  $x$ , with a total number of points  $N_{tot} = 5$ , and white cross marks denote discrete points in  $r$ , with total number of points  $N_{shell} = 4$ ; (c) a 1D FVM control volume mesh representation in  $x$  (top) and in  $r$  (bottom).

## 2.1. Electrochemical–Thermal Model Equations

The P2D model consists of a coupled set of nonlinear PDAEs formed using fundamental physiochemical laws. The model was first proposed in 1993 by the Newman group [11] and, in its most basic isothermal form, models three processes occurring in a cell: (1) mass conservation, (2) charge conservation, and (3) mass transfer via reduction/oxidisation (redox) reactions. The model was further adapted by Rao et al. [13] to characterise heat generation and dissipation in the cell as well. Hence, this full set of coupled PDAEs is commonly referred to as the electrochemical–thermal model of a LiB (see Table 2 for the full set of model equations).

Mass conservation in the P2D model is applied to both the solid and electrolyte phases of the cell. To model solid-phase mass conservation, first, solid-phase diffusion is modelled using Fick’s second law of diffusion, as in Equation (1) in Table 2. Fick’s law of diffusion relates the rate of change in the concentration of neutral lithium in the solid electrode phase,  $c_1(x, r, t)$ , to the spatial derivative of the same concentration. Note that  $D_{1,i}^{eff}$  is the effective solid-phase diffusion coefficient of lithium, and its relation is shown in Equation (26) in Table 3. This model for the spherical diffusion of neutral lithium is applicable primarily to intercalation-type electrodes; when modelling other types of electrodes, adaptations to this spherical diffusion model will be required [39].

The conservation of mass in the electrolyte phase of the model is characterised by using concentrated solution theory, as shown in Equation (2) in Table 2. This model of ionic diffusion accounts for the variation in lithium-ion concentration in the electrolyte,  $c_2(x, t)$ , as being driven by variations in lithium-ion concentration gradients and charge gradients in the electrolyte. Note that  $a_i$  is the solid interfacial area of the domain, and  $D_{2,i}^{eff}$  is the effective

tive lithium-ion diffusion coefficient; their relations are shown in Equations (22) and (23), respectively, in Table 3.

The P2D model assumes that there are two currents flowing in a LiB simultaneously, namely, the solid-phase electronic current,  $i_1(x, t)$ , and the electrolyte-phase ionic current,  $i_2(x, t)$ . By applying Kirchoff's current law, at any point in  $x$  across the model, both currents must be added to the applied current  $I_{app}(t)$ , as shown in Equation (5) in Table 2. As currents flow in both phases of the model, charge conservation must be applied in each phase. In the solid phase, this is implemented using Ohm's law, and hence, the spatial variation in the solid-phase potential,  $\psi_1(x, t)$ , can be determined using Equation (3) in Table 2. Note that  $\sigma_i^{eff}$  is the effective electronic conductivity of the solid phase, and its relation is shown in Equation (24) in Table 3.

**Table 2.** Electrochemical–thermal P2D model equations.

Differential Equations	Boundary Conditions
<p>(P1) Solid-Phase Diffusion</p> $\frac{\partial c_1(x, t)}{\partial t} = \frac{D_{1,i}^{eff}}{r^2} \frac{\partial}{\partial r} \left( r^2 \frac{\partial c_1}{\partial r} \right)$	$\left. \frac{\partial c_1(x, t)}{\partial r} \right _{r=0} = 0,$ $\left. \frac{\partial c_1(x, t)}{\partial r} \right _{r=R_i} = -\frac{j_{int}(x, t)}{D_{1,i}^{eff}} \quad (1)$
<p>(P2) Electrolyte-Phase Diffusion</p> $e_{2,i} \frac{\partial c_2(x, t)}{\partial t} = D_{2,i}^{eff} \frac{\partial^2 c_2(x, t)}{\partial x^2} + a_{1,i}(1 - t_+) j(x, t)$	$\left. \frac{\partial c_2(x, t)}{\partial x} \right _{x=0, x=L_{tot}} = 0,$ $D_{2,i}^{eff} \left. \frac{\partial c_2(x, t)}{\partial x} \right _{x=L_n^-} = D_{2,i}^{eff} \left. \frac{\partial c_2(x, t)}{\partial x} \right _{x=L_n^+},$ $D_{2,i}^{eff} \left. \frac{\partial c_2(x, t)}{\partial x} \right _{x=L_n+L_s} = D_{2,i}^{eff} \left. \frac{\partial c_2(x, t)}{\partial x} \right _{x=L_n+L_s^+} \quad (2)$
<p>(P3) Ohm's Law in Solid Phase</p> $\sigma_{1,i}^{eff} \frac{\partial^2 \psi_1(x, t)}{\partial x^2} = a_{1,i} F j(x, t)$	$\left. \frac{\partial \psi_1}{\partial x} \right _{x=0, L_{tot}} = -\frac{I_{app}(t)}{\sigma_{1,i}^{eff}},$ $\left. \frac{\partial \psi_1}{\partial x} \right _{x=L_n, L_n+L_s} = 0 \quad (3)$
<p>(P4) Ohm's Law in Electrolyte Phase</p> $\kappa_2^{eff} \frac{\partial^2 \psi_2(x, t)}{\partial x^2} = \frac{2RT(x, t)}{F} (1 - t_+) \kappa_2^{eff} \frac{\partial^2 \ln(c_2(x, t))}{\partial x^2} - a_1 F j(x, t)$	$\psi_2(0, t) = 0,$ $\left. \frac{\partial \psi_2}{\partial x} \right _{x=0, L_{tot}} = 0 \quad (4)$
<p>(P5) Kirchhoff's Current Law</p> $i_1(x, t) + i_2(x, t) = I_{app}(t)$	$i_1(x, t) = 0 \text{ for } L_n < x < L_n + L_s \quad (5)$
$\frac{\partial i_1(x, t)}{\partial x} = -a_{1,i} F j(x, t)$	$i_2(0, t) = 0, i_2(L_{tot}, t) = 0 \quad (6)$
$\frac{\partial i_2(x, t)}{\partial x} = a_{1,i} F j(x, t)$	$i_2(0, t) = 0, i_2(L_{tot}, t) = 0 \quad (7)$
<p>(P6) Redox Reaction Exchange Flux</p> $j(x, t) = \frac{i_0(x, t)}{F} \sinh \left( \frac{a_i F}{RT(x, t)} \eta(x, t) \right)$	$i_0(x, t) = 2Fk_i(c_1(x, R_i, t)(c_{1,i}^{max} - c_1(x, R_i, t))(c_2(x, t)))^{0.5} \quad (8)$
$\eta(x, t) = \psi_1(x, t) - \psi_2(x, t) - U_i(x, t) - FR_{SEI} j(x, t)$	$R_{SEI} = 0 \text{ for } x > L_n \quad (9)$
<p>(P7) Thermal Energy Balance</p> $\rho_i C_{p,i} \frac{\partial T(x, t)}{\partial t} = \lambda_i \frac{\partial^2 T(x, t)}{\partial x^2} + Q_{gen,i}(x, t)$	$\left. \frac{\partial T}{\partial x} \right _{x=0} = -h_{ext}(T_{amb} - T(0, t))$ $\left. \frac{\partial T}{\partial x} \right _{x=L_{tot}} = -h_{ext}(T(L_{tot}, t) - T_{amb}) \quad (10)$
	$\left. \frac{\partial T}{\partial x} \right _{x=L_{tot}} = -h_{ext}(T(L_{tot}, t) - T_{amb}) \quad (11)$

Charge conservation in the electrolyte phase is modelled using Ohm's law as well. However, in this phase, the spatial variation in the electrolyte potential,  $\psi_2(x, t)$ , is due not only to the ionic current in the electrolyte but also to the potential gradients formed by the variation in ionic concentration in the electrolyte, hence explaining the two terms in

Equation (4) in Table 2. Also note that the term  $\kappa_2^{eff}$  is the effective ionic conductivity in the electrolyte, and its relation is shown through Equations (20) and (21) in Table 3.

**Table 3.** Additional P2D model equations for a Lithium Manganese Oxide/Graphite cell.

Open-Circuit Potentials	
$U_n(x, t) = U_{n,ref}(x, t) + (T(x, t) - T_{amb}) \left. \frac{\partial U_n(x, t)}{\partial T} \right _{T_{amb}}$	(12)
$U_p(x, t) = U_{p,ref}(x, t) + (T(x, t) - T_{amb}) \left. \frac{\partial U_p(x, t)}{\partial T} \right _{T_{amb}}$	(13)
Stoichiometry	
$\theta_n(x, t) = \frac{c_1(x, R_n, t)}{c_{1,n}^{max}}$	(14)
$\theta_p(x, t) = \frac{c_1(x, R_p, t)}{c_{1,p}^{max}}$	(15)
Open-Circuit Potential Function	
$^1 U_{n,ref}(x, t) = -0.16 + 1.32 \exp(-3\theta_n(x, t)) + 10 \exp(-2000\theta_n(x, t))$	(16)
$^1 U_{p,ref}(x, t) = 4.19829 + 0.0565661 \tanh(-14.5546\theta_p(x, t) + 8.60942) - \frac{0.0275479}{(0.998432 - \theta_p(x, t))^{0.492465} - 1.90111} - 0.157123 \exp(0.04738(\theta_p(x, t)^8)) + 0.810239 \exp(-40(\theta_p(x, t) - 0.133875))$	(17)
Entropy Change	
$\left. \frac{\partial U_n(x, t)}{\partial T} \right _{T_{amb}} = 0$	(18)
$\left. \frac{\partial U_p(x, t)}{\partial T} \right _{T_{amb}} = 0$	(19)
Electrolyte Conductivity	
$^1 \kappa_2 = 1.0793 \times 10^{-2} + 6.7461 \times 10^{-4} c_2(x, t) - 5.2245 \times 10^{-7} c_2(x, t)^2 + 1.3605 \times 10^{-10} c_2(x, t)^3 - 1.1724 \times 10^{-14} c_2(x, t)^4$	(20)
$\kappa_2^{eff} = \epsilon_2^{brugsi} \kappa_2$	(21)
Particle Surface-Area-to-Volume Ratio	
$a_{1,i} = \frac{3\epsilon_1}{R_i}$	(22)
Bruggeman Parameter Corrections	
$D_{2,i}^{eff} = \epsilon_2^{brugsi} D_{2,i}$	(23)
$\sigma_{1,i}^{eff} = \epsilon_1^{brugsi} \sigma_{1,i}$	(24)
Arrhenius Relationships	
$k_i^{eff} = k_i \exp[-\frac{E_a^i}{R} (\frac{1}{T(x, t)} - \frac{1}{T_{amb}})]$	(25)
$D_{1,i}^{eff} = D_{1,i} \exp[-\frac{E_{D1,i}}{R} (\frac{1}{T(x, t)} - \frac{1}{T_{amb}})]$	(26)
Heat Generation	
$Q_{gen}(x, t) = Q_{jh}(x, t) + Q_{rev}(x, t) + Q_{irrev}(x, t)$	(27)
$Q_{jh}(x, t) = i_1(x, t) \frac{\partial \psi_1(x, t)}{\partial x} + i_2(x, t) \frac{\partial \psi_2(x, t)}{\partial x}$	(28)
$Q_{rev}(x, t) = a_{1,i} F j(x, t) T(x, t) \left. \frac{\partial U_i(x, t)}{\partial T} \right _{T_{amb}}$	(29)
$Q_{irrev}(x, t) = a_{1,i} F j(x, t) \eta(x, t)$	(30)

<sup>1</sup> Function is taken from work presented in Liu et al. [40].

Thus far, the P2D model has represented the physiochemical processes occurring in a cell as independently occurring in the two phases of the cell: i.e., in each phase, there is a concentration variable  $c_j(x, t)$ , a potential variable  $\psi_j(x, t)$ , and a current variable  $i_j(x, t)$ . The intercoupling of the phases is modelled using the Butler–Volmer relation for a reversible redox reaction [5], and its exact form is shown in Equations (8)–(10) in Table 2.



Note that the term  $j(x, t)$  is the pore wall exchange flux and represents the number of lithium atoms that are being oxidised at the surfaces of the solid electrode particles and will then be converted into lithium ions in the electrolyte surrounding the electrode particle.

The exchange flux is the key term to solving the P2D model, as it not only represents the rate of redox reactions occurring at any point but also is related to the spatial derivative of ionic and solid-phase currents, as shown in Equations (6) and (7) in Table 2. The exchange flux is also coupled directly to the potentials across the model via the electrode overpotential variable  $\eta(x, t)$ , as in Equation (10). Note that, in this term, the potential drop due to the resistance of the SEI layer in the anode  $R_{SEI}$  is also considered. However, in the cathode, no SEI growth is assumed, and thus, it is ignored in the cathode.

The final physiochemical process in the electrochemical–thermal model is the thermal energy balance process in Equation (11) in Table 2. The equation models the generation and dissipation of heat/temperature,  $T(x, t)$ , in the cell model. The dissipation of heat is modelled using the second-order spatial derivative of the temperature and thermal conductivity,  $\lambda_i$ , of the different domains of the cell. The generation of heat,  $Q_{gen,i}(x, t)$ , can be composed of multiple different heat sources assumed to be present during the operation of a LiB. We consider three different heat sources. The first heat source is due to the joule heating effect,  $Q_{jh}(x, t)$ , caused by the flow of currents in the different phases of the model and is represented using Equation (28) in Table 3. The second heat source is due to the reversible heat generated during redox reactions,  $Q_{rev}(x, t)$ , in the electrode domains and is calculated using Equation (29) in Table 3. Note that to calculate this heat source, the entropy change in the electrodes due to redox reactions,  $\left. \frac{\partial U_i(x, t)}{\partial T} \right|_{T_{amb}}$ , must be known, as shown in Equations (18) and (19) in Table 3. The final heat source considered is the irreversible heat generation in the electrodes due to redox reactions,  $Q_{irrev}(x, t)$ , and is modelled using Equation (30) in Table 3.

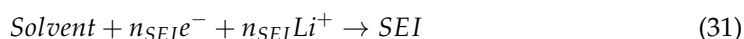
## 2.2. SEI Growth Models

During the operation of a LiB, multiple different degradation (‘ageing’) processes can occur simultaneously. These include processes such as lithium plating, where lithium dendrites are observed to grow on the surfaces of anode electrode particles, particularly under operating conditions like fast charging or under low-temperature conditions. Cathode Electrolyte Interphase (CEI) growth is another significant degradation process, where electrolyte particles undergo side reactions with cathode electrode particles to form an interphase on the surface of the cathode [41]. Electrode particle cracking is also an important degradation mechanism, where the consistent intercalation/deintercalation of electrode particles induces fatigue, leading to particle cracking and the loss of electrode active material [42].

Multiple cell-level physics models of these different degradation mechanisms have been proposed in the literature; however, the most comprehensively validated models have been presented for work related to the degradation process of Solid Electrolyte Interphase (SEI) growth [43,44].

Solid Electrolyte Interphase (SEI) growth is widely considered by researchers to be one of the principal mechanisms that induce capacity fade in LiBs [44]. SEI growth is the process by which a passivation layer grows on the surface of an electrode, and it is considered to commonly occur on the surfaces of graphite anodes [16].

The formation of the SEI layer occurs due to side reactions occurring at the interface between the anode particle and the electrolyte. This side reaction is irreversible and thus has the general form shown in Equation (31) below:



Note that  $n_{SEI}$  refers to the number of electrons involved in the SEI formation reaction. As the side reaction is irreversible, the formation of SEI products leads to a loss in lithium inventory and thus a loss of capacity. Furthermore, as we consider the electrolyte used in

the cell model to be Ethylene Carbonate (EC), this side reaction has the following specific form for our model:

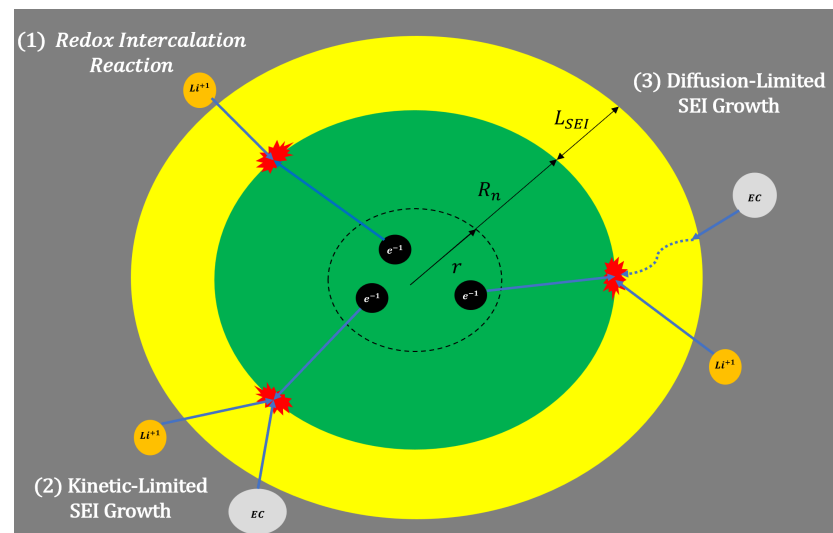


This side reaction occurs during the charging of the cell, when there is a negative overpotential at the anode particle surface [15]. Hence, during charging, there are two competing reactions occurring simultaneously in the anode: the intercalation reduction reaction represented by the exchange flux,  $j_{int}(x, t)$ , and the irreversible SEI formation reaction represented by the exchange flux,  $j_{SEI}(x, t)$ . Thus, the sum of these two fluxes must be equal to the total exchange flux as follows:

$$j(x, t) = j_{int}(x, t) + j_{SEI}(x, t) \quad (33)$$

Note: We do not consider an SEI side reaction to be occurring at the cathode, and thus,  $j_{SEI}(x, t)$  would be zero outside of the anode domain.

When choosing to model SEI growth in the continuum-scale P2D model, there are two models that are commonly used in the literature [44]: the kinetic-limited SEI growth model and the diffusion-limited SEI growth model, as shown in Figure 2.



**Figure 2.** An illustration of the different possible reactions on a graphite anode particle during charging. These include: (1) The reduction/oxidisation (redox) intercalation reactions occurring at the SEI/anode interphase, (2) Kinetic-limited SEI growth occurring at the SEI/anode interphase and (3) Diffusion-limited SEI growth, displaying the diffusion of solvent particles through the SEI layer to react at the SEI/anode interphase.

For both models, the SEI reaction exchange flux,  $j_{SEI}(x, t)$ , is modelled using the cathodic Tafel equation and the SEI overpotential,  $\eta_{SEI}(x, t)$ , as shown below:

$$j_{SEI}(x, t) = -\frac{i_{0,SEI}}{F} \exp\left(-\frac{\alpha_{SEI}F}{RT(x, t)} \eta_{SEI}(x, t)\right) \quad (34)$$

$$\eta_{SEI}(x, t) = \psi_1(x, t) - \psi_1(x, t) - U_{SEI} - FR_{SEI}j(x, t) \quad (35)$$

The Tafel equation is commonly used in electrochemical modelling to approximate the exchange flux of irreversible reactions, such as the SEI formation reaction. Thus, Equations (34) and (35) are common for both the kinetic-limited model and the diffusion-limited model. The two models just have a major difference in the expression for the SEI exchange current,  $i_{0,SEI}$ .



### 2.2.1. Kinetic-Limited Reaction Model

The kinetically limited SEI growth model was first presented by Ramadass et al. [15] and has since been used copiously [16,45–47]. The model proposes that the SEI growth rate is only driven by the SEI overpotential,  $\eta_{SEI}(x, t)$ . Hence, for the kinetic-limited case, the SEI exchange current,  $i_{0,SEI}$ , is considered a fitting parameter and takes a constant value.

### 2.2.2. Diffusion-Limited Reaction Model

The diffusion-limited model is also used in the literature, e.g., by Kamyab et al. [48], Das et al. [49], and Kane et al. [43]. This growth model takes into account the diffusion of the solvent from the interface of the SEI layer/electrolyte to the interface of the SEI layer/anode surface, as shown in Figure 2. As such, its SEI exchange current term is modelled as follows:

$$i_{0,SEI} = Fk_{SEI}c_{sol}(x, R_n, t)^{\alpha_{SEI}} \quad (36)$$

In this model, the SEI reaction rate,  $k_{SEI}$ , is chosen as the constant fitting parameter, and its value varies based on the specific LiB being modelled. The term  $c_{sol}(x, R_n, t)$  is the concentration of the Ethylene Carbonate (EC) solvent at the interface of the SEI layer/anode particle. To determine this value, a second-order partial differential equation is used to model the diffusion of the EC solvent across the SEI layer as follows:

$$\frac{\partial c_{sol}(x, r, t)}{\partial t} = D_{sol} \frac{\partial^2 c_{sol}(x, r, t)}{\partial r^2} \quad (37)$$

This diffusion equation holds for the following boundary conditions at the SEI layer/anode particle surface:

$$D_{sol} \left. \frac{\partial c_{sol}(x, r, t)}{\partial r} \right|_{r=R_n} = j_{SEI}(x, t) \quad (38)$$

The following boundary condition holds at the interface between the SEI layer and the electrolyte:

$$c_{sol}(x, R_n + L_{SEI}, t) = \epsilon_{SEI}c_{sol,2} \quad (39)$$

The above boundary condition stems from the assumption that the SEI layer is a porous structure that allows the diffusion of solvent particles through it [48]. Hence, the concentration of the solvent at the interphase of the SEI layer and the electrolyte is assumed to be the product of the SEI layer's porosity,  $\epsilon_{SEI}$ , and the concentration of the EC solvent in the bulk electrolyte,  $c_{sol,2}$ .

To reduce the complexity of solving this model, many works have considered the solvent diffusion profile across the SEI layer to be linear [44]. This assumption reduces the order of Equation (37) and can be simply solved as follows:

$$c_{sol}(x, R_n, t + \Delta t) = \epsilon_{SEI}c_{sol,2} + \frac{L_{SEI}}{D_{sol}}j_{SEI}(x, t) \quad (40)$$

The reduced-order model of Equation (40) is used in this work to model solvent diffusion in the diffusion-limited SEI growth model.

### 2.2.3. SEI Thickness, Resistance, and Capacity Lost

For both growth models, SEI formation is considered to occur at the SEI layer/anode interface, as shown in Figure 2. Thus, the rates of growth of the SEI layer thickness, resistance, and capacity loss can be determined as follows:

$$\frac{\partial L_{SEI}(x, t)}{\partial t} = -\frac{M_{SEI}}{n_{SEI}\rho_{SEI}}j_{SEI}(x, t) \quad (41)$$

$$\frac{\partial R_{SEI}(x, t)}{\partial t} = -\frac{M_{SEI}}{n_{SEI}\rho_{SEI}\kappa_{SEI}}j_{SEI}(x, t) \quad (42)$$

$$\frac{\partial Q_{lost}(x, t)}{\partial t} = -\frac{Fa_{1,n}}{n_{SEI}}j_{SEI}(x, t) \quad (43)$$

Also note that, in both the models of SEI growth, we have chosen to ignore the electric potential change across the electric double layer that forms at the interface of the electrolyte and SEI layer. This order reduction assumption was made as this potential drop was observed to have a negligible effect on SEI growth during nominal operational currents [49]. Furthermore, as the role of the electric double layer in the formation rate of the SEI layer is still an ongoing field of active research [50], the authors have not been able to determine a validated cell-level model of the double-layer formation mechanism on the SEI/electrolyte interphase for use in our modelling work.

### 3. Model Discretisation

As mentioned previously, the full-order P2D model has no analytical solution without using order reduction assumptions, and we normally need an iterative numerical approach to solve a P2D model in practice [17,18]. To implement a numerical solver, the continuous PDAEs described in Table 2 must be converted into Discrete Algebraic Equations (DAEs) to be solved at discrete points in space and time. In this work, the Finite Volume Method (FVM) is used for spatial discretisation, and the Verlet Integration Method (VIM) is used for the temporal discretisation of the P2D model.

#### 3.1. Mesh Generation

Before applying the FVM and VIM, the continuous P2D model must be spatially discretised into a fixed number of elements in the two spatial dimensions of the model, i.e., in the  $x$  dimension across the thickness of the cell model and in the  $r$  dimension radially across spherical electrode particles, as shown in Figure 1b.

##### 3.1.1. Mesh Generation in $x$ Dimension

When discretising the model in the  $x$  dimension, the number of elements for each material domain must be chosen. For example, in Figure 1b, a coarse mesh is illustrated with two elements for the anode domain,  $N_n = 2$ , one element for the separator domain,  $N_s = 1$ , and two elements for the cathode domain,  $N_p = 2$ . Hence, the total number of  $x$ -dimension elements is five, i.e.,  $N_{tot} = 5$ . Note that the position of the element in  $x$  is denoted by the index  $k$ , and thus,  $1 \leq k \leq N_{tot}$ .

The meshing algorithm implemented in this work requires only the total number of  $x$ -dimension elements,  $N_{tot}$ , to be selected. The number of elements for the different domains is allocated according to the ratio of the thicknesses of the different domains of the model to ensure that the width of each element in the model is of a similar thickness. Hence, the width of an element in the  $x$  dimension in a specific material domain  $i$  has the following thickness:

$$\Delta x_i = \frac{L_i}{N_i} \approx \frac{L_{tot}}{N_{tot}} \quad (44)$$

##### 3.1.2. Mesh Generation in $r$ Dimension

When discretising the model in the  $r$  dimension, the meshing algorithm selects the number of elements in both the anode and the cathode to be equal. In the example mesh in Figure 1b, a radial mesh of four elements is shown, i.e.,  $N_{shell} = 4$ . Note that the positions of radial elements in  $r$  are denoted by the index  $q$ , and thus,  $1 \leq q \leq N_{shell}$ . The meshing algorithm works by choosing the number of radial elements,  $N_{shell}$ , and then generates

a radial mesh with an equal number of elements for all electrode particles in the model. Thus, the width of a radial element in an electrode material domain  $i$  is given as follows:

$$\Delta r_i = \frac{R_i}{N_{shell}} \quad (45)$$

### 3.2. Finite Volume Method

There are multiple different methods that can be used to spatially discretise continuous PDAEs. Commonly used approaches include the Finite Difference Method (FDM), the Finite Element Method (FEM), and the Finite Volume Method (FVM). All of these methods have been used in the literature to solve a P2D model, as shown in Table 1.

The conceptually simplest discretisation method is the Finite Difference Method (FDM), which first discretises the model in the spatial and temporal dimensions and then directly implements the model's continuous PDAEs as finite differences. While this method is theoretically simple to implement, for the P2D model, it poses challenges when implementing boundary conditions, especially those at material discontinuity boundaries [27].

The Finite Element Method (FEM) is another method for PDAE discretisation, which works by first dividing the spatial model into a set of finite elements. Thereafter, using the selected method, a trial function is determined to solve the PDAEs in each finite element. The FEM method has been used efficiently to solve the P2D model, such as in the works by Ai et al. [26] and Noor et al. [32]; however, the method often involves expressing the model equations in specific forms for solving in each model element. This method, while accurate and computationally efficient, is conceptually complex, and the effectiveness of the method depends on factors such as the method of determining the trial function and the method of discretising model equations in the finite elements.

In this work, the Finite Volume Method (FVM) is chosen for the spatial discretisation of the P2D model PDAEs. The FVM implements a differential equation by considering fluxes flowing into and out of a control volume (CV) and relating the difference in the fluxes to variable changes in the control volume. Hence, the FVM is particularly effective at applying variable conservation equations across a given control volume [51]. This makes the FVM particularly useful for the discretisation of the P2D model due to its accurate implementation of boundary conditions, particularly the mass transfer boundary conditions in Equation (2) in Table 2.

#### 3.2.1. PDAE Discretisation in the $x$ Dimension

To illustrate how the FVM can be used to implement differential equations, consider a generic three-element grid in the  $x$  dimension, as shown in Figure 1c.

Considering a general dependent variable,  $\Omega$ , in a 1D mesh,  $\Omega$  would have finite values at the centre of each control volume. Hence, at the centre of a CV with node point  $k$ , the discrete value of the variable will be  $\Omega_k$ , as shown in Figure 1c. In the FVM, a flux is typically a first-order spatial derivative of a variable at one of the boundaries of a CV. Hence, the two fluxes at the left boundary,  $k - 1/2$ , and right boundary,  $k + 1/2$ , would have the following algebraic forms:

$$\left. \frac{\partial \Omega}{\partial x} \right|_{k-1/2} = \frac{\Omega_k - \Omega_{k-1}}{\Delta x_{k-1/2}} \quad (46)$$

$$\left. \frac{\partial \Omega}{\partial x} \right|_{k+1/2} = \frac{\Omega_{k+1} - \Omega_k}{\Delta x_{k+1/2}} \quad (47)$$

Note that the element widths at the boundaries are simply the means of the widths of the elements in the surrounding boundaries, for example,

$$\Delta x_{k+1/2} = \frac{\Delta x_{k+1} + \Delta x_k}{2} \quad (48)$$

As some P2D model equations require logarithmic derivatives, these derivatives can also be calculated in a similar manner as follows:

$$\left. \frac{\partial \log(\Omega)}{\partial x} \right|_{k+1/2} = \frac{\ln(\Omega_{k+1}) - \ln(\Omega_k)}{\Delta x_{k+1/2}} \quad (49)$$

Most of the P2D model equations require evaluations of second-order spatial derivatives at node points; the FVM can simply compute the higher-order derivatives as follows:

$$\left. \frac{\partial^2 \Omega}{\partial x^2} \right|_k = \frac{1}{\Delta x_k} \left[ \frac{\partial \Omega}{\partial x} \right]_{k-1/2}^{k+1/2} = \frac{1}{\Delta x_k} \left[ \left. \frac{\partial \Omega}{\partial x} \right|_{k+1/2} - \left. \frac{\partial \Omega}{\partial x} \right|_{k-1/2} \right] \quad (50)$$

Thus, using the FVM in the  $x$  dimension, the time derivative equations of lithium-ion diffusion and thermal energy balance can be converted into discrete algebraic equations at each grid point  $k$  as follows:

$$\epsilon_{2,i} \left. \frac{\partial c_{2,k}(t)}{\partial t} \right|_k = \frac{1}{\Delta x_k} \left[ D_{2,k}^{eff} \frac{\partial c_{2,k}(t)}{\partial x} \right]_{k-0.5}^{k+0.5} + a_{1,i}(1-t_+)j_k(t) \quad (51)$$

$$\rho_i C_{p,i} \left. \frac{\partial T_k(t)}{\partial t} \right|_k = \frac{1}{\Delta x_k} \left[ \lambda_k \frac{\partial T_k(t)}{\partial x} \right]_{k-0.5}^{k+0.5} + Q_{gen,k}(t) \quad (52)$$

Note that the key term of the exchange flux,  $j_k(t)$ , which is essential in calculating Equations (51) and (52), can be found at grid point  $k$  using the following DAEs:

$$j_k(t) = \frac{i_{o,k}(t)}{F} \sinh \left( \frac{\alpha_i F}{RT_k(t)} \eta_k(t) \right) \quad (53)$$

$$i_{o,k}(t) = 2Fk_i^{eff}(c_{1,k,N_{shell}}(t)(c_{1,i}^{max} - c_{1,k,N_{shell}}(t))(c_{2,k}(t)))^{0.5} \quad (54)$$

$$\eta_k(t) = \psi_{1,k}(t) - \psi_{2,k}(t) - U_{OCp,k} - FR_{SEI,k}j_k(t) \quad (55)$$

One of the powerful implications of using the FVM is that second-order spatial derivative equations can be reformulated as sequential DAEs. This is illustrated below for the  $x$ -dimension PDAEs of Ohm's law in the solid phase and of Ohm's law in the electrolyte in Equations (56) and (57), respectively:

$$\sigma_{1,k+1/2}^{eff} \left. \frac{\partial \psi_1}{\partial x} \right|_{k+1/2} = \sigma_{1,k-1/2}^{eff} \left. \frac{\partial \psi_1}{\partial x} \right|_{k-1/2} + \Delta x_k a_{1,i} F j_k(t) \quad (56)$$

$$\kappa_{2,k+1/2}^{eff} \left. \frac{\partial \psi_2}{\partial x} \right|_{k+1/2} = \kappa_{2,k-1/2}^{eff} \left. \frac{\partial \psi_2}{\partial x} \right|_{k-1/2} + \frac{2RT_k(t)}{F}(1-t_+)\kappa_{2,k}^{eff} \left[ \frac{\partial \ln(c_2)}{\partial x} \right]_{k-1/2}^{k+1/2} - \Delta x_k a_{1,i} F j_k(t) \quad (57)$$

By constructing the above sequential DAEs, an iterative solver can be set up to solve the discretised model equations if certain boundary values are estimated, as described in Section 4.

### 3.2.2. PDAE Discretisation in the $r$ Dimension

The PDAE for solid-phase diffusion in Equation (1) radially resolves the concentration of neutral lithium,  $c_1(x, r, t)$ , across spherical electrode particles. Hence, the FVM needs to be applied across the spherical electrode particles in the electrode domains.

To implement the FVM radially, the spherical electrode particles are first split into a set of non-overlapping control volumes, where each radial control volume is a concentric shell, as shown in Figure 1c.

Using this mesh, to apply the FVM, a term for the flux of neutral lithium must first be derived if the surface area of each concentric shell is known. The area of each shell can be calculated as shown in Equation (58).

$$A_{1,k,q+0.5} = 4\pi r_{q+0.5}^2 \quad (58)$$

Using this surface area and knowing the effective solid-phase diffusivity of lithium,  $D_{1,k}^{eff}$ , the flux at the surface of a shell can be calculated as follows:

$$Flux_{q+0.5} = D_{1,k}^{eff} A_{1,k,q+0.5} \left. \frac{\partial c_1}{\partial r} \right|_{q+0.5} \quad (59)$$

Further, the spatial derivative of the neutral lithium concentration can be calculated using the FVM as follows:

$$\left. \frac{\partial c_1}{\partial r} \right|_{q+0.5} = \frac{c_{1,k,q+1}(t) - c_{1,k,q}(t)}{\Delta r} \quad (60)$$

Thereafter, using the flux in and out of each radial CV and using the volume of a radial CV as shown in Equation (61), the PDAE for solid-phase diffusion can be expressed as a DAE, as in Equation (62).

$$V_{k,q} = \frac{4}{3}\pi r_{q+0.5}^3 - \frac{4}{3}\pi r_{q-0.5}^3 \quad (61)$$

$$\frac{\partial c_{1,k,q}(t)}{\partial t} = \frac{D_{1,k}^{eff}}{V_{k,q}} \left[ A_{1,k,q+0.5} \left. \frac{\partial c_1}{\partial r} \right|_{q+0.5} - A_{1,k,q-0.5} \left. \frac{\partial c_1}{\partial r} \right|_{q-0.5} \right] \quad (62)$$

### 3.3. Spatial Boundary Condition Implementation

For any set of PDAEs to have a physically meaningful solution, boundary conditions are needed. In the P2D model, all the boundary conditions are spatial boundary conditions that occur in at least one of the four material domain boundaries. The four boundaries are the negative current collector/anode ( $ncc/n$ ), the anode/separator ( $n/s$ ), the separator/cathode ( $s/p$ ), and the cathode/positive current collector ( $p/pcc$ ).

One of the key problems in numerically solving the P2D model is in evaluating the material parameters at these material boundaries. For example, a parameter such as the effective electrolyte-phase lithium-ion diffusivity,  $D_{2,i}^{eff}$ , can vary by an order of magnitude between one of the electrode domains and the separator domain. The significant variation in model parameters causes instability in solvers and leads to extremely fine time steps to solve the model.

Several approaches are proposed in the literature to solve the material discontinuity problem, e.g., using a simple numerical mean between domain parameters [52], using a harmonic mean between domain parameters [36], or, in some cases, even using an extended control volume across material boundaries [27].

In this work, we used a combination of a simple algebraic mean and a harmonic mean to evaluate the material parameters at boundaries. The method used depends on whether the boundary is between CVs within a specific material domain or whether the boundary is between the CVs of two different material domains.

#### 3.3.1. Boundaries between Material Domains

In our P2D model, there are two inter-material boundaries, which are the anode/separator ( $n/s$ ) boundary and the separator/cathode boundary ( $s/p$ ). At these boundaries, there are two non-overlapping CVs from two different domains, whose boundaries intersect exactly. When applying the FVM to these two CVs, material parameters need to be evaluated at this inter-material boundary. Due to the significant material parameter discontinuities at the inter-material boundaries, an intermediate value for these parameters

is required to be numerically stable for implementation in a solver. The harmonic mean approach can effectively manage large outliers in data sets [53] and is therefore employed in this work to find intermediate parameter values at inter-material boundaries.

The parameters that the harmonic mean is applied to are  $D_{2,i}^{eff}$ ,  $\lambda_i$ , and  $\kappa_2^{eff}$ . An example of how the harmonic mean is calculated is shown for the thermal conductivity parameter,  $\lambda_i$ , at the anode/seperator boundary.

$$\lambda_{N_n+0.5} = \frac{(\lambda_{N_n})(\lambda_{N_n+1})}{\gamma\lambda_{N_n+1} + (1-\gamma)\lambda_{N_n}} \quad (63)$$

Note that the mean CV width is found as  $\gamma = \frac{\Delta x_n + \Delta x_s}{2}$ .

### 3.3.2. Boundaries within a Material Domain

Within a material domain, there are no significant variations in these material parameters. Hence, the parameters  $D_{2,i}^{eff}$ ,  $\sigma_{1,i}^{eff}$ ,  $\lambda_i$ , and  $\kappa_2^{eff}$  can be evaluated at intra-material CV boundaries using a simple arithmetic mean, as shown below for ionic diffusivity:

$$D_{2,k+0.5}^{eff} = \frac{D_{2,k+1}^{eff} + D_{2,k}^{eff}}{2} \quad (64)$$

### 3.3.3. Other Boundary Conditions

For solving the discretised electrochemical–thermal model, the entire set of spatial boundary conditions to be implemented can be found in Table 4.

**Table 4.** Electrochemical–thermal model’s discretised boundary conditions.

Boundary Condition	Grid Position	
(P1) Solid-Phase Diffusion		
$\left. \frac{\partial c_{1,k,q}(t)}{\partial r} \right _{q=0.5} = 0$	$1 \leq k \leq N_n,$ $N_n + N_s + 1 \leq k \leq N_{tot},$ $q = 0.5$	(65)
$\left. \frac{\partial c_{1,k,q}(t)}{\partial r} \right _{q=N_{shell}+0.5} = -\frac{j_{int,k}(t)}{D_{1,k}^{eff}}$	$1 \leq k \leq N_n,$ $N_n + N_s + 1 \leq k \leq N_{tot},$ $q = N_{shell} + 0.5$	(66)
(P2) Electrolyte-Phase Diffusion		
$\left. \frac{\partial c_{2,k}}{\partial x} \right _{k=0.5} = 0$	$k = 0.5$	(67)
$\left. \frac{\partial c_{2,k}}{\partial x} \right _{k=N_{tot}+0.5} = 0$	$k = N_{tot} + 0.5$	(68)
(P3) Ohm’s Law in Solid Phase		
$\left. \frac{\partial \psi_1}{\partial x} \right _{k=0.5} = -\frac{I_{app}(t)}{\sigma_{1,0.5}^{eff}}$	$k = 0.5$	(69)
$\left. \frac{\partial \psi_1}{\partial x} \right _{k=N_{tot}+0.5} = -\frac{I_{app}(t)}{\sigma_{1,0.5}^{eff}}$	$k = N_{tot} + 0.5$	(70)
(P4) Ohm’s Law in Electrolyte Phase		
$\psi_2(0, t) = 0,$	$k = 1$	(71)
$\left. \frac{\partial \psi_2}{\partial x} \right _{k=0.5} = 0$	$k = 0.5$	(72)



Table 4. Cont.

Boundary Condition	Grid Position	
(P5) Kirchhoff's Current Law		
$i_{1,0.5} = I_{app}(t)$	$k = 0.5$	(73)
$i_{1,N_{tot}+0.5} = I_{app}(t)$	$k = N_{tot} + 0.5$	(74)
$i_{2,k} = I_{app}(t)$	$N_n + 0.5 \leq k \leq N_n + N_s + 0.5$	(75)
(P6) Redox Reaction Exchange Flux		
$j_k(t) = 0$	$N_n + 0.5 \leq k \leq N_n + N_s + 0.5$	(76)
(P7) Thermal Energy Balance		
$\left. \frac{\partial T}{\partial x} \right _{k=0.5} = -h_{ext}(T_{amb} - T_1(t))$	$k = 0.5$	(77)
$\left. \frac{\partial T}{\partial x} \right _{k=N_{tot}+0.5} = -h_{ext}(T_{amb} - T_1(t))$	$k = N_{tot} + 0.5$	(78)

### 3.4. Verlet Integration

After solving the discretised P2D model equations for a given time step,  $t$ , the discretised equations need to be integrated in time to find the next set of values of the dependent variables. One of the simplest approaches for discretised time integration is the Euler method [54], as shown below for temperature:

$$T_k(t + \Delta t) = T_k(t) + \Delta t \left. \frac{\partial T_k(t)}{\partial t} \right|_k \quad (79)$$

Note that  $\Delta t$  here refers to the time step and is a key parameter for solver stability. The Euler method, while simple to implement, is a first-order integrator and, as a result, can have stability issues and require fine time steps for certain models [54]. To improve the speed and stability of the solver, we use the Verlet Integration Method [55] to integrate the DAEs of the P2D model:

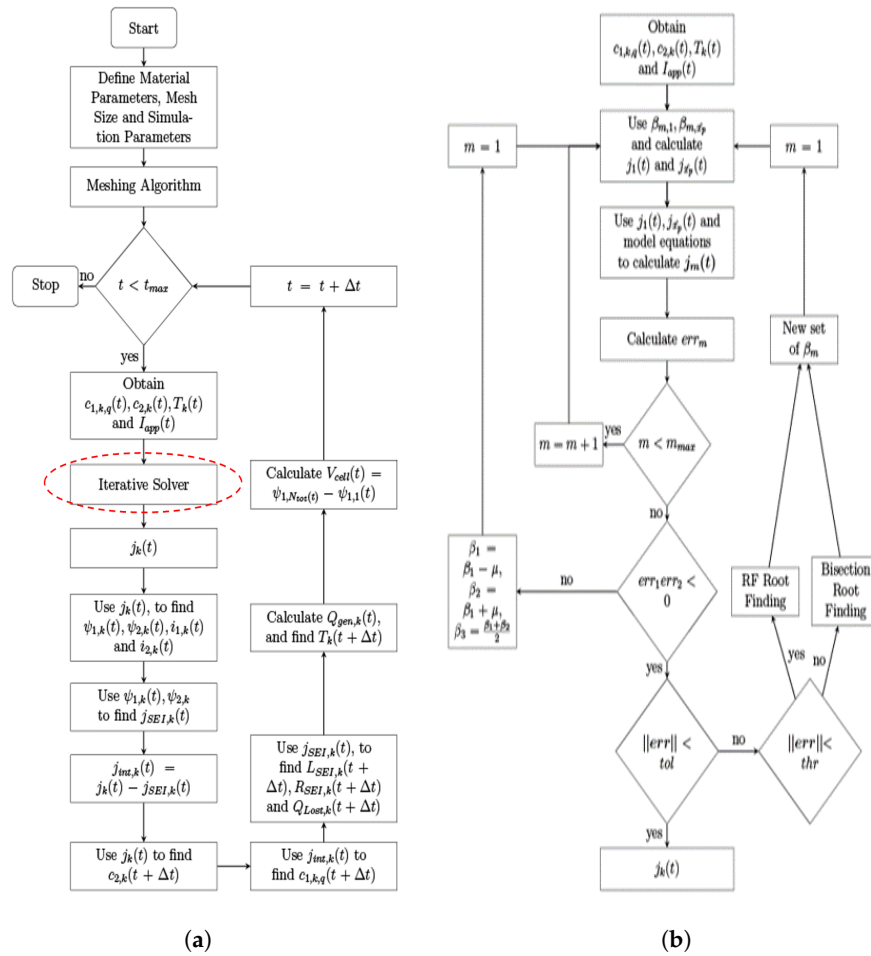
$$\left. \frac{\partial^2 T_k(t)}{\partial t^2} \right|_k = \frac{T_k(t + \Delta t) - 2T_k(t) + T_k(t - \Delta t)}{\Delta t^2} \quad (80)$$

$$T_k(t + \Delta t) = 2T_k(t) - T_k(t - \Delta t) + \frac{\Delta t^2}{2} \left. \frac{\partial^2 T_k(t)}{\partial t^2} \right|_k \quad (81)$$

As shown in Equations (80) and (81), Verlet Integration is clearly a second-order integrator. However, the second-order time derivative in Equation (80) requires the next time-step value to be known. Hence, to estimate the next value, the MATLAB *ode15s* differential equation integrator is used to determine this value and thereafter calculate the second-order time derivative of Equation (80). Due to the use of a second-order time integrator, larger time steps become more stable when integrating the equations of the P2D model. To the best of the authors' knowledge, our proposed solver is the first attempt to use Verlet Integration to integrate the equations of the P2D model.

## 4. Solver Algorithm

Given that the model equations are discretised using the FVM and VIM, the discretised P2D model can now be implemented in a modelling algorithm. The P2D model algorithm implemented in this work has the structure shown by the flowchart in Figure 3a.



**Figure 3.** Flowcharts of solver algorithm. (a) Flowchart for electrochemical–thermal ageing model algorithm; (b) Flowchart of iterative solver.

The algorithm first requires a set of material parameters for a specific LiB cell. In this work, we are primarily simulating a Lithium Manganese Oxide/Graphite LiB, and the model parameters are listed in Table 5. After defining the parameter set, the independent simulation variables are defined. These independent variables include the maximum duration of the simulation,  $t_{max}$ , and the applied current,  $I_{app}(t)$ . Thereafter, the mesh size needs to be defined, and a meshing algorithm is run according to the process described in Section 3.1.

After the meshing is completed, the main simulation loop is started, where at each point in time,  $t$ , six standard dependent variables are determined, which are  $\psi_{1,k}(t)$ ,  $\psi_{2,k}(t)$ ,  $j_k(t)$ ,  $c_{1,k}(t)$ ,  $c_{2,k,q}(t)$ , and  $T_k(t)$ . If SEI growth is chosen to be modelled, an additional set of dependent variables specific to SEI growth are also calculated; these include  $j_{SEI,k}(t)$ ,  $j_{int,k}(t)$ ,  $L_{SEI,k}(t)$ ,  $R_{SEI,k}(t)$ , and  $Q_{Lost,k}(t)$ .

As shown in Figure 3a, the most critical process of the modelling algorithm is the iterative solver; therefore, we propose a novel iterative solver using a combination of physics-informed initial guesses and optimal root-finding algorithms to produce accurate and rapid solutions.

**Table 5.** Electrochemical–thermal model parameter values.

Parameter	Description	Units	Anode	Separator	Cathode
<sup>2</sup> $SoC_{init}$	Initial State of Charge	-	0.58	-	0.19
<sup>1</sup> $c_{1,i}^{max}$	Maximum Solid-Phase Lithium Concentration	mols/m <sup>3</sup>	26,390	-	22,860
<sup>2</sup> $c_{1,i}^{init}$	Initial Solid-Phase Lithium Concentration	mols/m <sup>3</sup>	$SoC_{n,i}^{init} c_{1,n}^{max}$	-	$SoC_{p,i}^{init} c_{1,p}^{max}$
<sup>1</sup> $c_{2,i}^{init}$	Initial Electrolyte-Phase Lithium-ion Concentration	mols/m <sup>3</sup>	2000	2000	2000
<sup>1</sup> $D_{1,i}$	Solid-Phase Lithium Diffusivity	m <sup>2</sup> /s	$3.9 \times 10^{-14}$	-	$1 \times 10^{-13}$
<sup>1</sup> $D_{2,i}$	Electrolyte-Phase Lithium-ion Diffusivity	m <sup>2</sup> /s	$7.5 \times 10^{-11}$	$7.5 \times 10^{-11}$	$7.5 \times 10^{-11}$
<sup>1</sup> $\sigma_{1,i}$	Solid-Phase Electrical Conductivity	S/m	100.0	-	3.8
<sup>1</sup> $L_i$	Thickness of Domain	m	$100 \times 10^{-6}$	$52 \times 10^{-6}$	$183 \times 10^{-6}$
<sup>1</sup> $R_i$	Electrode Particle Radius	m	$12.5 \times 10^{-6}$	-	$8.5 \times 10^{-6}$
<sup>1</sup> $\epsilon_{1,i}$	Electrode Volume Fraction	-	0.471	-	0.2970
<sup>1</sup> $\epsilon_{2,i}$	Porosity	-	0.3570	1.0	0.4440
<sup>1</sup> $brugg_i$	Bruggeman Coefficient	-	1.5	1.5	1.5
<sup>2</sup> $k_i$	Intercalation Reaction Rate Constant	m <sup>2.5</sup> /(mol <sup>0.5</sup> s)	$2 \times 10^{-10}$	-	$2 \times 10^{-10}$
<sup>1</sup> $\alpha_i$	Intercalation Reaction Transfer Coefficient	-	0.5	-	0.5
<sup>1</sup> $t_+$	Lithium-ion Transference Number	-	0.3630	0.3630	0.3630
$F$	Faraday Constant	Col/mol	96,487	96,487	96,487
$R$	Ideal Gas Constant	J/(mol K)	8.314	8.314	8.314
<sup>1</sup> $\rho_i$	Domain Density	kg/m <sup>3</sup>	2500	1200	1500
<sup>1</sup> $C_{p,i}$	Domain Specific Heat Capacity	J/(Kg K)	700	700	700
<sup>1</sup> $\lambda_i$	Domain Thermal Conductivity	W/(m K)	5	1	5
<sup>1</sup> $E_a^i$	Intercalation Reaction Rate Activation Energy	J/mol	30,000	-	30,000
<sup>1</sup> $E_a^{D_{1,i}}$	Solid-Phase Diffusivity Activation Energy	J/mol	4000	-	20,000
<sup>1</sup> $h_{ext}$	Heat Exchange Coefficient	W/(m <sup>2</sup> K)	5	-	5
<sup>1</sup> $T_{amb}$	Ambient Temperature	C	25.15	-	25.15
<sup>1</sup> $T_{init}$	Initial Cell Temperature	C	25.15	25.15	25.15
<sup>1</sup> $l_{SEI}^{init}$	Initial SEI Thickness	m	$1 \times 10^{-9}$	-	-
<sup>2</sup> $R_{SEI}^{init}$	Initial SEI Resistance	$\Omega m^2$	$2 \times 10^{-4}$	-	-
<sup>1</sup> $\kappa_{SEI}$	SEI Lithium Conductivity	S/m	$5 \times 10^{-6}$	-	-
<sup>1</sup> $M_{SEI}$	SEI Molar Mass	kg/mol	0.1620	-	-
<sup>1</sup> $\rho_{SEI}$	SEI Density	kg/m <sup>3</sup>	1690	-	-
<sup>1</sup> $\alpha_{SEI}$	SEI Reaction Transfer Coefficient	-	0.5	-	-
<sup>2</sup> $n_{SEI}$	Number of Electrons Involved in SEI Reaction	-	2	-	-
<sup>1</sup> $U_{SEI}$	Open-Circuit Potential of SEI Growth	V	0	-	-
<sup>2</sup> $i_{0,SEI}$	Kinetic-Limited SEI Exchange Current	A/m <sup>2</sup>	$8.8 \times 10^{-4}$	-	-
<sup>2</sup> $\kappa_{SEI}$	SEI Reaction Rate Coefficient	m/s <sup>2</sup>	$2 \times 10^{-12}$	-	-
<sup>2</sup> $D_{sol}$	Diffusivity of Electrolyte Solvent in SEI	m <sup>2</sup> /s	$3.5 \times 10^{-20}$	-	-
<sup>1</sup> $c_{sol,2}$	Concentration of Solvent in Bulk Electrolyte	mols/m <sup>3</sup>	4541	-	-
<sup>2</sup> $\epsilon_{SEI}$	Porosity of SEI Layer	-	0.03	-	-

<sup>1</sup> Parameters are taken from work presented in Liu et al. [40]. <sup>2</sup> Parameters are tuned for this work. In addition to these model parameters, the Battery\_Parameters.m file in the solver suite also contains battery parameter sets for Lithium Cobalt Oxide/Graphite (LCO/C6) [36] and Nickel Manganese Cobalt/Graphite (NMC111/C6) [56].

#### 4.1. The Iterative Solver

As shown in Figure 3b, as input, the iterative solver requires the cell-wide distribution of the lithium-ion concentration,  $c_{1,k}(t)$ ; the solid lithium concentration,  $c_{2,k,q}(t)$ ; the cell-wide temperature distribution,  $T_k(t)$ ; and the applied current,  $I_{app}(t)$ , at the current time step,  $t$ . The iterative solver will then use these values to determine the cell-wide total exchange flux,  $j_k(t)$ , for the current time step  $t$ . The total exchange flux,  $j_k(t)$ , is the key variable that can be used to solve all other dependent variables of the P2D model; hence, the solution to the P2D model at any time step is the cell-wide distribution of  $j_k(t)$ .

By finding this distribution, all other discretised equations detailed in Section 3 can be directly solved. The solver works by first guessing what the values of the ion exchange flux are at two points in the cell model. The first value to be guessed is the flux at the first element of the anode,  $k = 1$ , denoted by  $j_1(t)$ . The second value to be guessed is the exchange flux in the first element of the cathode, denoted by  $j_{x_p}(t)$ . Note here that  $x_p$  is used as a shorthand for the grid location of the first cathode element where  $k = N_n + N_s + 1$ .

These guesses are not arbitrary values but are chosen to be higher and lower than the theoretical average value for the exchange flux in the respective electrodes. For example, when trying to guess the values for flux at the first element of the anode, the theoretical average of the anode electrode exchange flux,  $\bar{j}_n(t)$ , can be used as a useful indicator for the actual value,  $j_1(t)$ . Thus, to start the iterative solver, physics-informed initial guesses of the average exchange fluxes in the negative electrode  $\bar{j}_n(t)$  and in the positive electrode  $\bar{j}_p(t)$  are used. These guess values can be analytically calculated (see Appendix A). Hence,

guess values for the exchange fluxes in the first element of the anode and the first element of the cathode can be calculated as follows:

$$j_{1,m}(t) = \beta_{1,m} \bar{j}_n \quad (82)$$

$$j_{x_p,m}(t) = \beta_{x_p,m} \bar{j}_p \quad (83)$$

In the above equations,  $\beta_{1,m}$  is a scaling factor that represents the guess number,  $m$ , for the first element of the anode,  $k = 1$ . The same nomenclature applies to  $\beta_{x_p,m}$ . Once these guess values are determined, using the spatial boundary conditions across the cell model, the discretised P2D model equations are then solved sequentially. Thus, for a set of values  $\beta_m = [\beta_{1,m}, \beta_{x_p,m}]$ , the discretised P2D model equations are solved, producing a corresponding exchange flux distribution,  $j_m(t)$ .

As a result of this method, the values of  $\beta_m$  are critical for tuning the corresponding ion-exchange flux distribution,  $j_m(t)$ , as required. In each iteration of the solver, three estimates of  $j_m(t)$  are made, and the estimates are defined as the lower-bound estimate,  $j_1(t)$ ; the higher-bound estimate,  $j_2(t)$ ; and the intermediate estimate,  $j_3(t)$ . Note that  $m$  corresponds to the estimate number, and therefore,  $m_{max} = 3$ . Thus, three sets of  $\beta_m$  are required to produce three  $j_m(t)$  distributions. Thereafter, for each estimated distribution  $j_m(t)$ , an error value,  $err_m$ , can be calculated using the average exchange flux values, as shown below:

$$err_m = [err_{m,n}, err_{m,p}] = \left[ \frac{\bar{j}_n - \frac{1}{N_n} \sum_1^{N_n} j_{k,m}(t)}{\bar{j}_n}, \frac{\bar{j}_p - \frac{1}{N_p} \sum_{x_p}^{N_{tot}} j_{k,m}(t)}{\bar{j}_p} \right] \quad (84)$$

These error values,  $err_m$ , are estimates of how close the corresponding flux distribution estimate,  $j_m(t)$ , is to the actual flux distribution,  $j(t)$ .

The guess set,  $\beta_1$ , is selected such that it produces a flux distribution estimate,  $j_1(t)$ , that has a negative error value, i.e.,  $err_1 < 0$ . Similarly, the guess set  $\beta_2$  is selected to generate a flux distribution estimate,  $j_2(t)$ , that has a positive error value, i.e.,  $err_2 > 0$ . Thus, the intermediate guess set,  $\beta_3$ , has an error value,  $err_3$ , that is between the lower- and upper-bound error values.

The novel convergence criterion implemented in this work is intended to minimise the intermediate error,  $err_3$ , below a tolerance value  $tol$ . To the knowledge of the authors, this work is the first attempt to use the average exchange flux,  $\bar{j}_i$ , as a convergence criterion to solve P2D model equations. However, before root-finding algorithms can be used to reduce the error magnitude  $||err||$ , first, the ‘root window’ where the optimal solution lies must be found.

#### 4.1.1. Root Window Searching

In the ideal case, the condition for a root to be found can be defined as follows:

$$err_3 = f(\beta_3^{root}) = 0 \quad (85)$$

This condition means that the intermediate guess,  $\beta_3$ , is exactly at the solution of the equation at this time step. For this to be true, the upper-bound and lower-bound conditions must be on either side of the root; thus, the following condition must also be true:

$$err_1 err_2 < 0 \quad (86)$$

Condition (86) essentially means that the root lies between the upper-bound and lower-bound estimates. Thus, if condition (86) is satisfied, the ‘root window’ can be found. The solver searches for this root window using a root searching parameter,  $\mu$ , as shown by the branch on the left side of the flowchart in Figure 3b. This searching parameter can be tuned to optimise the solver’s performance.

#### 4.1.2. Hybrid Root-Finding Algorithm

Once the ‘root window’ is found, root-finding algorithms can then be applied to determine the solution for a given time step. In this work, we present a novel hybrid root-finding algorithm for the P2D model that uses a combination of the bisection and Regula Falsi (RF) root-finding methods. Both of these methods are forms of bracketed root-finding algorithms, which successively calculate smaller intervals (brackets), where a root occurs.

Initially, during this root-finding process, the guess error magnitudes,  $||err||$ , are quite large; thus, a suitable root-finding method is required that can minimise significant errors. The bisection method is effective, particularly for high-magnitude errors, as it uses a simple bracket-halving approach to reduce the interval size. The algorithm uses three guesses for  $\beta_m$ , i.e.,  $m = [1, 2, 3]$ . The algorithm selects  $\beta_1$  to have a negative error,  $err_1 < 0$ , and selects  $\beta_2$  to have a positive error,  $err_2 > 0$ . The final guess set of  $\beta_3$  is chosen such that it is the mean value of  $\beta_1$  and  $\beta_2$ , as shown below:

$$\begin{aligned} err_1 = f(\beta_1) < 0, err_2 = f(\beta_2) > 0, \\ \rightarrow err_3 = f(\beta_3) = f((\beta_1 + \beta_2)/2) \end{aligned} \quad (87)$$

Given the value of  $err_3$ ,  $\beta_m^{next}$  is chosen based on whether the root lies between  $\beta_1$  and  $\beta_3$  or  $\beta_2$  and  $\beta_3$ , as shown below:

$$\begin{aligned} if(err_1 * err_3) < 0 \\ \Rightarrow \beta_1^{next} = \beta_1, \beta_2^{next} = \beta_3, \beta_3^{next} = (\beta_1^{next} + \beta_2^{next})/2 \end{aligned} \quad (88)$$

$$\begin{aligned} if(err_2 * err_3) < 0 \\ \Rightarrow \beta_1^{next} = \beta_3, \beta_2^{next} = \beta_2, \beta_3^{next} = (\beta_1^{next} + \beta_2^{next})/2 \end{aligned} \quad (89)$$

Due to this simple bracket-halving approach, the bisection method has a slow linear convergence rate [57]. To accelerate the root-finding process, once the error magnitude,  $||err||$ , falls below a threshold value of  $thr$ , we switch the root-finding process to the Regula Falsi (RF) algorithm, which has a much faster convergence rate. The RF method can be considered a form of the gradient descent method, except applied to bracket root-finding cases where the gradient of the function is not explicitly known. The RF method chooses  $\beta_m^{next}$  based on a numerical estimate of the derivative of the root estimates, as shown below.

$$\begin{aligned} if(err_1 * err_3) < 0 \\ \Rightarrow \beta_1^{next} = \beta_1, \beta_2^{next} = \beta_3, \beta_3^{next} = \frac{\beta_1^{next} err_3 - \beta_2^{next} err_1}{err_3 - err_1} \end{aligned} \quad (90)$$

$$\begin{aligned} if(err_2 * err_3) < 0 \\ \Rightarrow \beta_1^{next} = \beta_3, \beta_2^{next} = \beta_2, \beta_3^{next} = \frac{\beta_1^{next} err_2 - \beta_2^{next} err_3}{err_2 - err_3} \end{aligned} \quad (91)$$

The hybrid root-finding algorithm tunes the scaling factors,  $\beta_m$ , with each subsequent iteration until the intermediate error value  $err_3$  goes below an absolute error tolerance threshold, which is set to  $tol = 1 \times 10^{-3}$ .

## 5. Results and Discussion

The discretised equations of the P2D model described in Section 3 and the modelling algorithm in Section 4 are implemented using MATLAB™ R2023 and are freely available for download at <https://github.com/twick07/Electrochemical-Thermal-P2D-Model-Iterative-Solver> (accessed on 1 April 2024).

To benchmark the accuracy of the proposed solver, we chose the commercial battery modelling software COMSOL™ Multiphysics 5.6. COMSOL has seen broad use in the field of battery modelling due to the software suite's flexibility in the analysis of different physics phenomena occurring in cells. This includes multiple works related to the modelling of temperature generation/dissipation [29,31,36] and ageing phenomena like SEI growth [40,47,48]. For this reason, we used COMSOL as the benchmark software to compare the accuracy and performance of our proposed solver.

Both solvers were set up with the same model parameters of a Lithium Manganese Oxide/Graphite (LMO/C6) cell (see the P2D model parameters in Table 5) and run on the same computational platform: a Windows 11 PC with an Intel Core i7-8650U@ 1.9 GHz processor and 16 GB of RAM.

As the proposed solver is designed for an electrochemical–thermal ageing model of a LiB, to validate the accuracy of the solver, we considered three different operation scenarios: (1) the standard single 1C discharge/charge cycle, (2) multiple-C-rate discharge conditions, and (3) different hybrid dynamic drive cycle conditions. In all scenarios, the solver results were compared with those of COMSOL to validate the solver's accuracy. Additionally, in Section 5.6, we present a detailed performance analysis comparing our solver, COMSOL, and other existing solver solutions in the literature. Finally, in Section 5.7, we present a brief analysis of the two SEI growth models presented in this work, along with a discussion of their advantages and disadvantages.

### 5.1. Single Discharge/Charge Cycle Validation

The first method of validation is to compare the accuracy of our solver for a single 1C discharge/charge cycle. We compare five dependent variables over this cycle for this specific battery model. These dependent variables are the cell voltage,  $V_{cell}(t)$ ; cell temperature,  $T(t)$ ; lithium-ion concentration,  $c_2(x, t)$ ; and two lithium solid concentrations:  $c_{1,ave}(x, t)$  and  $c_1(x, R_i, t)$ . Note that a single 1C rate is equivalent to an applied current,  $I_{app}(t)$ , of 17 A/m<sup>2</sup> for this LMO/C6 model. The comparison of our solver's results is benchmarked against COMSOL's results, as seen in Figure 4.

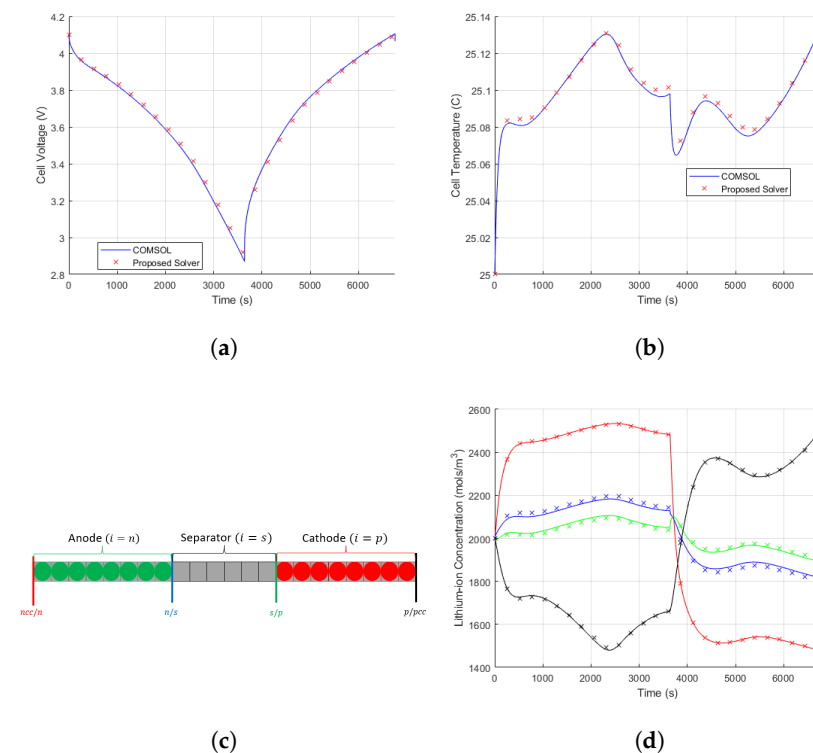
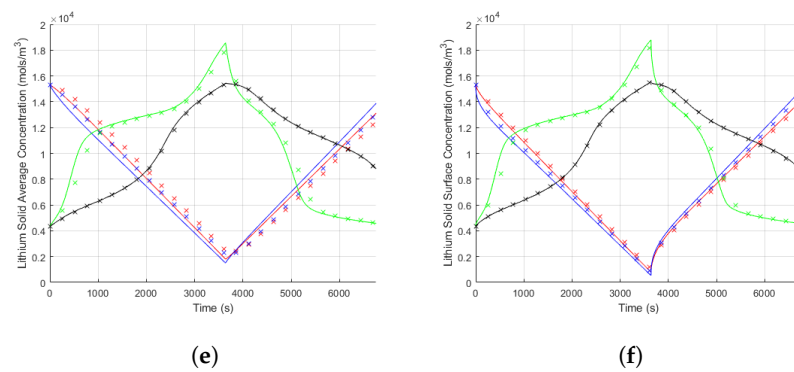


Figure 4. Cont.





**Figure 4.** Plots of 1C discharge/charge validation: [x] points are the proposed solver's results, and solid lines are COMSOL's results. (a) The cell voltage; (b) the cell temperature; (c) an illustration of the physical locations of the different lines in the concentration plots, where red lines are at ncc/n, blue lines are at n/s, green lines are at s/p, and black lines are at p/pcc; (d) the lithium-ion concentration in the electrolyte; (e) the average lithium concentration in electrodes; (f) the lithium solid surface concentration in electrodes.

For the results in Figure 4, all independent simulation variables, such as the number of mesh elements and time step, were kept the same for both solvers. Figure 4 shows clearly that the proposed solver results are in line with those of COMSOL. However, for a better assessment of accuracy, the following definition of root mean square percentage error is used for a generic dependent variable,  $\Omega$ .

$$\Omega_{error}^{\%} = \frac{1}{N_{time}} \sum_{t=t_1}^{t_{stop}} \sqrt{\frac{(\Omega_{Solver}(t) - \Omega_{COMSOL}(t))^2}{\Omega_{COMSOL}(t)}} \quad (92)$$

Note that  $N_{time}$  in Equation (92) is the total number of time steps in the modelling algorithm. Using this definition for the percentage error, the percentage errors for the results of the 1C discharge/charge simulation are shown in Table 6.

**Table 6.** Percentage errors for dependent variables in a single 1C discharge/charge cycle.

	$V_{cell}(t)$	$c_2(x, t)$	$c_1(x, R_i, t)$	$c_{1,ave}(x, t)$	$T(x, t)$
Percentage Errors	0.44%	0.43%	2.82%	2.81%	<0.01%

For this single discharge/charge cycle validation, a grid of 30 elements in  $x$  ( $N_{tot} = 30$ ) and 10 elements in  $r$  ( $N_{shell} = 10$ ) with a time step of one second ( $\Delta t = 1s$ ) was chosen for both solvers. A mesh of such size is typically considered a coarse grid, and hence, even with this low mesh resolution, errors of less than 3% for all dependent variables are achieved. The solver accuracy can be improved by using a finer mesh; fortunately, one of the key attributes of our proposed solver is that this larger mesh size does not significantly increase the computation time (see Section 5.6).

## 5.2. Temperature Variation in the P2D Model

A key contribution of this work is the provision of a full-order model of temperature generation/dissipation in LiB cells; thus, as shown in Figure 5, we present a brief description of the underlying heat generation models that occur during the 1C charge/discharge simulation.

In Figure 5a, during discharge, the cell temperature is seen to gradually increase until the cell SoC falls below 50%, whereafter the cell temperature is seen to decrease. To explain why this occurs, first consider the heat generated in each of the three material domains of the cell, as shown in Figure 5b. Here, it can be seen that the heat generation of the separator

region is broadly constant throughout the entire discharge process, and the anode heat generation gradually increases over the entire discharge half cycle. The cathode region, however, undergoes dynamic heat generation, and the heat generation profile of the region broadly correlates to the temperature profile for the entire cell. This same relationship can also be observed for the charging half cycle.

To gain a deeper understanding of why cathode heat generation changes as such, consider the variation in heat generation terms for the cathode, as shown in Figure 5d. Equations (27) to (30) in Table 3 indicate that the heat generated in this model is due to the heat sources of joule heating,  $Q_{jh}(x, t)$ ; reversible heat generation,  $Q_{rev}(x, t)$ ; and irreversible heat generation,  $Q_{irrev}(x, t)$ . In Figure 5d, the total heat generation in the cathode is driven mainly by the joule heating heat source.

The joule heating sources, as shown in Equation (28) in Table 3, are related to the ionic and solid-phase currents in the material domain. These currents, in turn, are directly related to the exchange flux,  $j(x, t)$ , as shown in Equations (6) and (7) in Table 2. The exchange flux is the rate at which lithium atoms are being oxidised at the surfaces of electrode particles and is the primary cause of joule heating fluctuations in the cathode, as shown in Figure 5e. It can therefore be inferred that the variations in the redox reactions occurring in the cathode are the primary cause of the temperature change fluctuations occurring in the cell. This relationship, as will be shown in Section 5.4, is the primary cause of the temperature increase during high-C-rate discharging. Higher-C-rate discharging causes higher-magnitude exchange flux distributions in a LiB's electrodes, in turn inducing larger increases in cell temperature.

Note that the temperature modelling presented in this work only considers three heat source terms. There are more potential heat sources in a cell, such as the heat of mixing, where the variations in concentration gradients within electrode particles induce entropic changes that can generate heat in a cell [58]. One of the motivations of this work is to demonstrate how such heat generation models can be additively coupled to an iterative solver algorithm for a P2D model.

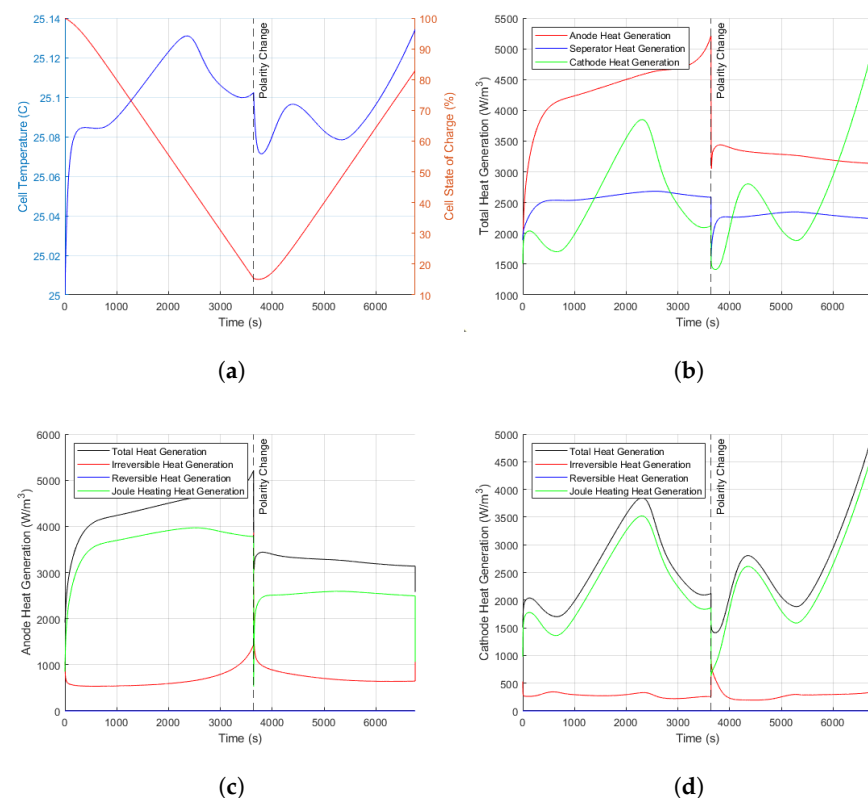
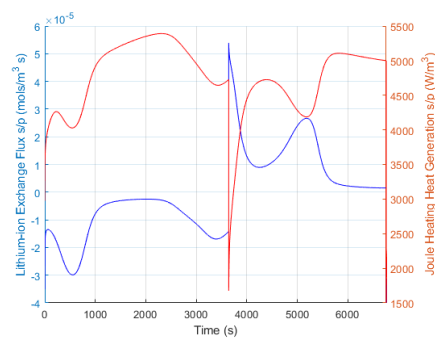


Figure 5. Cont.

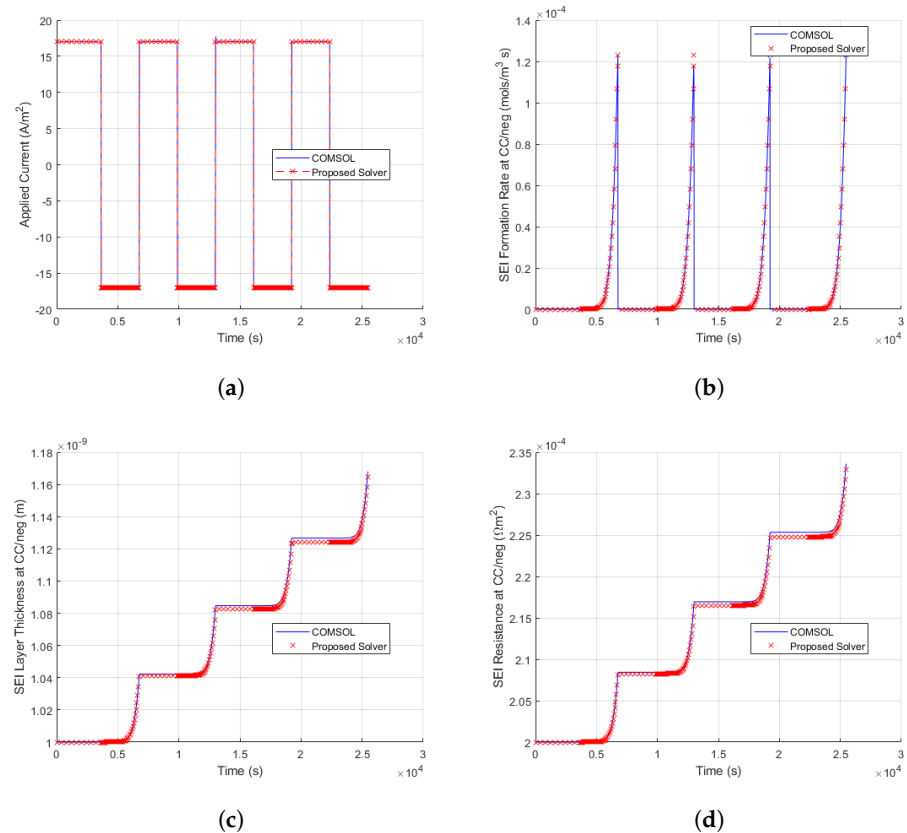


(e)

**Figure 5.** Description of heat generation during 1C discharge/charge simulations. (a) Cell temperature (blue line) vs. cell SoC change (red line); (b) comparison of average heat generation in each material domain; (c) comparison of average heat generation in anode material domain; (d) comparison of average heat generation in cathode material domain; (e) lithium-ion exchange flux (blue line) vs. joule heating heat generation (red line), at s/p boundary.

### 5.3. Validation of Kinetic SEI Growth Model

As one of the key contributions of this work is the provision of an SEI growth model, along with a P2D modelling suite, we briefly study the accuracy of the kinetic-limited SEI growth model. The kinetic growth model described in Section 2.2 was implemented in both our solver and COMSOL, and the results for a four-cycle galvanostatic discharge/charge process are presented in Figure 6.



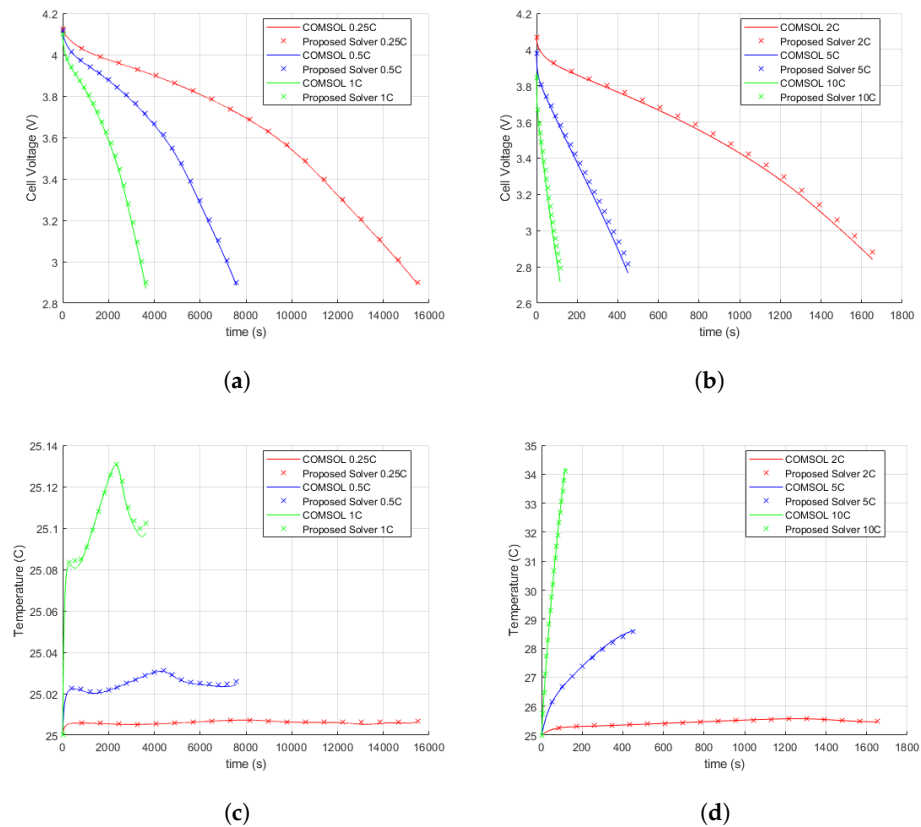
**Figure 6.** SEI growth validation plots at ncc/n boundary. (a) Applied current; (b) SEI formation rate; (c) SEI thickness; (d) SEI resistance.

The key variable is the SEI growth rate ( $j_{SEI}(x, t)$ ), shown in Figure 6b, as this variable indicates the speed at which the SEI layer thickness grows. As seen in Equations (41)–(43),

$j_{SEI}(x, t)$  is used to determine SEI growth parameters such as the SEI thickness (Figure 6c) and SEI resistance (Figure 6d). The accuracy of the key parameter  $j_{SEI}(x, t)$  is found to be an error of 4.5% in comparison to COMSOL, and as with the other variables, the accuracy of the solver for SEI growth can be increased by using a finer mesh in the simulation.

#### 5.4. Multi-C-Rate Discharge Validation

To further validate the accuracy of the proposed solver, we present an analysis of the solver accuracy for a range of different constant current discharge scenarios. The LMO/C6 cell model was subjected to C-rates ranging from 0.25C to 10C, and the proposed solver results are compared to COMSOL's in Figure 7.



**Figure 7.** Multi-C-rate validation plots. (a) Cell voltage for low C-rates; (b) cell voltage for high C-rates; (c) cell temperature for low C-rates; (d) cell temperature for high C-rates.

Using the definition for error percentage in Equation (92), the cell voltage,  $V_{cell}(t)$ , and cell temperature,  $T(t)$ , for the different discharge rates are compared in Table 7.

**Table 7.** Multi-C-rate error comparison.

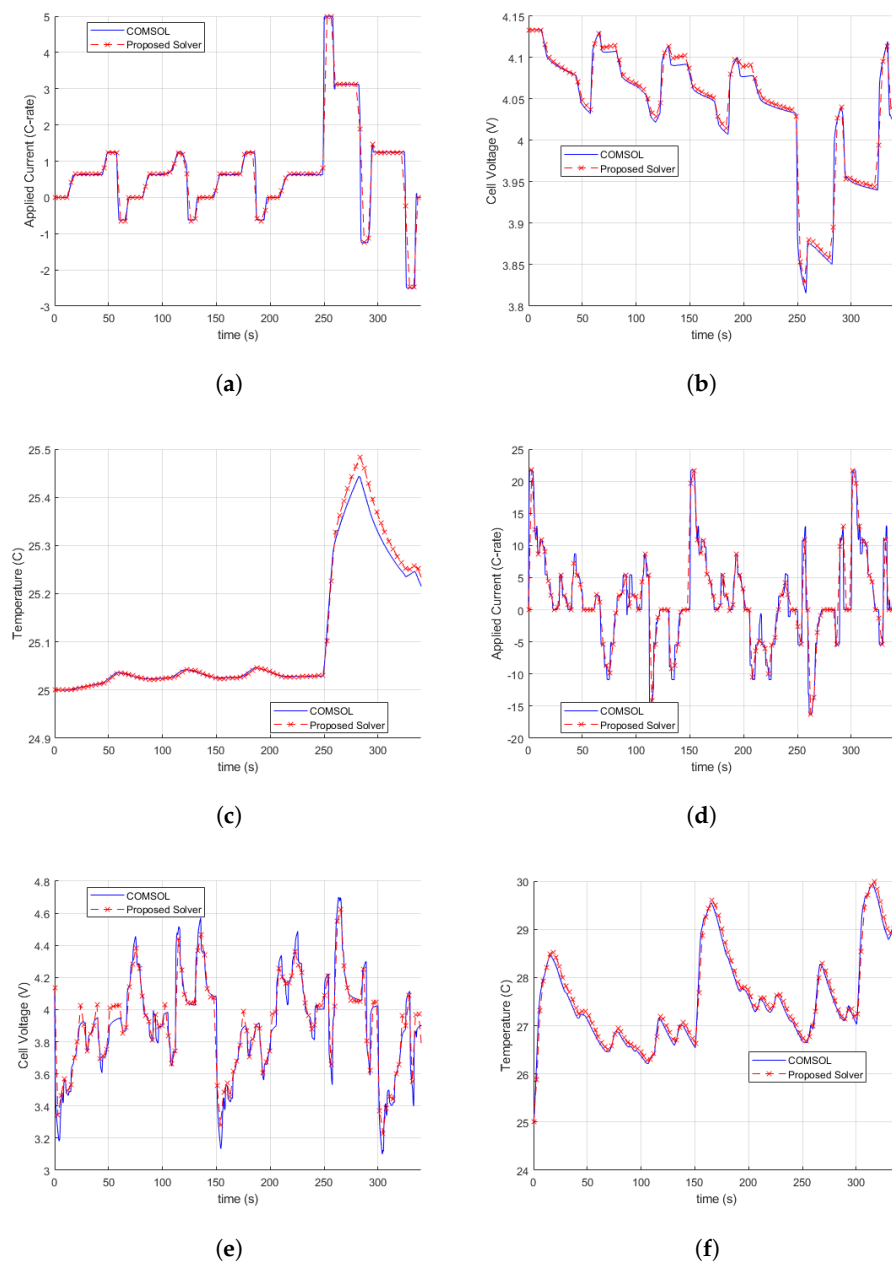
	0.25C	0.5C	1.0C	2.0C	5.0C	10.0C
$V_{cell}(t)$	0.13%	0.25%	0.44%	0.71%	0.98%	1.7%
$T(t)$	<0.01%	<0.01%	<0.01%	0.02%	0.15%	0.3%

According to these error results, the cell voltage error tends to increase with increasing C-rates. However, even for an extreme C-rate of 10C, the solver accuracy is kept to less than 2% error. In addition to this, the solver shows high accuracy for temperature modelling over all the C-rates.

### 5.5. Drive Cycle Validation

The final validation method used is dynamic drive cycles that are commonly experienced by electric vehicle battery packs. In this work, we chose two dynamic drive cycles to compare accuracy.

The first drive cycle used is the Dynamic Stress Test (DST) for a rechargeable battery. The DST drive profile was first proposed by the Lawrence Livermore National Laboratory and is commonly used to evaluate various battery models and SoC estimation algorithms [59]. The DST profile consists of a sequence of power steps with seven discrete power levels, and here, the maximum power level is considered a 5C applied current. The comparison solver results are presented in Figure 8a–c.



**Figure 8.** Dynamic drive cycle plots. (a) Applied current for DST drive cycle; (b) cell voltage response for DST drive cycle; (c) cell temperature response for DST drive cycle; (d) applied current for COMSOL dynamic drive cycle; (e) cell voltage response for COMSOL dynamic drive cycle; (f) cell temperature response for COMSOL dynamic drive cycle.

The second drive cycle is adopted from the COMSOL dynamic drive cycle experiment [60]. In this drive cycle, the cell is subject to large changes in applied current, ranging from +21C discharge to −17C charge. The comparison solver results are presented in Figure 8d–f.

As previously mentioned, to assess the accuracy, Equation (92) is used to compare the cell voltage,  $V_{cell}(t)$ , and cell temperature,  $T(t)$ . The solver errors with respect to COMSOL are shown in Table 8.

**Table 8.** Dynamic drive cycle error comparison.

Parameters	Dynamic Stress Test	COMSOL Dynamic Drive Cycle
$V_{cell}(t)$	0.4%	1%
$T(t)$	0.05%	0.02%

As is evident, the solver can estimate the cell voltage with a maximum error of 1% for both drive cycles while also having a very high accuracy for temperature modelling.

### 5.6. Solver Performance

In the results presented thus far, we have shown that the proposed solver can accurately represent the P2D electrochemical–thermal ageing model for multiple different operation scenarios. Here, we compare the performance of our solver to other state-of-the-art P2D solvers.

Since the previous works on P2D solvers have not considered the modelling of temperature generation/dissipation and ageing, we use the speed of solving a 1C isothermal discharge process as a measure for the performance comparison. Furthermore, the existing works have not tested their solver performance using a large range of different mesh sizes. In Table 9, we summarise the comparison between the proposed solver and some of the important existing solver solutions.

**Table 9.** Simulation speeds of different P2D solvers for 1C isothermal discharge.

Solver	Discretisation Method	Number of Elements in $x$								
		10	20	30	40	50	60	100	200	300
<sup>1</sup> Chen et al. [25]	FDM (ROM)	-	-	-	-	-	10.7 s	* 12.98 s	* 20.04 s	34.1 s
<sup>1</sup> Geng et al. [30]	FDM (ROM)	-	-	-	-	-	8 s	-	-	-
<sup>1</sup> R.Han et al. [31]	FDM	4.09 s	4.24 s	3.98 s	4.15 s	4.49 s	-	-	-	-
<sup>1</sup> Torchio et al. [31,36]	FVM	7.46 s	9.85 s	15.48 s	26.80 s	54.60 s	-	-	-	-
<sup>1</sup> Lee et al. [33]	FDM	* 1.28 s	* 2.1 s	-	-	40.25 s	-	* 50.8 s	-	-
<sup>1</sup> Doyle et al. [11,31]	FDM	28 s	69 s	97 s	137 s	185 s	-	-	-	-
<sup>2</sup> COMSOL	FEM	9 s	11 s	17 s	23 s	25 s	28.4 s	35 s	-	-
<sup>2</sup> Proposed solver	FVM	1.6 s	2.2 s	3.1 s	3.3 s	4.3 s	4.5 s	7.75 s	15.3 s	24.5 s

<sup>1</sup> The results of other solver works in the literature run on different parameter sets with different computational platforms. \* Simulation speeds are interpolated based on known speeds. <sup>2</sup> Solvers run with the parameter set of an LMO/C6 cell (see Table 5) on the same computational platform. The performance of the proposed solver for other chemistries, such as LCO/C6 and NMC111/C6 batteries, was found to have similar computation time to the results shown above.

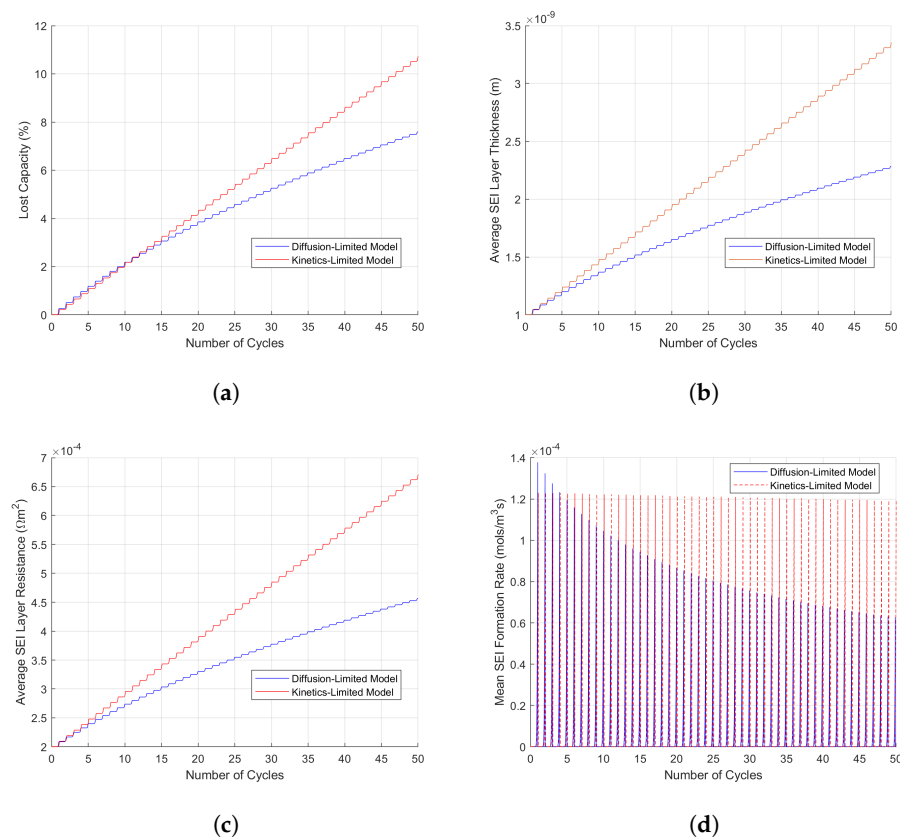
As shown in Table 9, our solver shows extremely fast solving speeds for the P2D model, with speeds of 1.6s for an extremely coarse  $N_{tot} = 10$  grid and speeds of 24.5 s for an extremely fine grid of  $N_{tot} = 300$ . Hence, our P2D solver can show significant performance even with extremely fine grids. To describe this further, we use the fine grid size of  $N_{tot} = 100$ , which is the finest grid resolution in COMSOL. Compared to other works at this resolution, our solver speed is much faster compared to that of COMSOL, and in fact, our solver has 4.5 times the solver speed of COMSOL for this grid resolution.



### 5.7. SEI Growth Model Comparison

In the proposed P2D model, we implement the two most used SEI growth models to consider ageing [44]. As mentioned in Section 2.2, these two models are the kinetic-limited SEI growth model and the diffusion-limited SEI growth model. To illustrate the difference between these two models, we simulated SEI growth in a LiB for both models over a 50-cycle duration. The parameters for the respective SEI growth models can be found in Table 5, and the results of these cycling simulations are shown in Figure 9.

As shown in Figure 9a, both models display at least an 8% loss in capacity after 50 cycles. This amount of capacity loss is far faster than that practically observed in LMO/C6 LiBs. The reason for such a significant capacity loss in this work is due to the parameters of the kinetic-limited and diffusion-limited SEI growth models being tuned to demonstrate a clear illustration of the difference between the two models. For a more practical parameter set for SEI growth models, we refer the reader to the works mentioned in Section 2.2.1 for the kinetic-limited model and the works mentioned in Section 2.2.2 for the diffusion-limited model.



**Figure 9.** Comparison of different SEI growth parameters for kinetic-limited model (red lines) and diffusion-limited model (blue lines). (a) Percentage lost capacity, (b) SEI layer resistance, (c) SEI thickness, (d) SEI reaction rate comparison.

In Figure 9a, the kinetic-limited model shows a linear rate of capacity loss with each cycle, which eventually results in a 10% capacity loss after 50 cycles. The rate of capacity loss in the kinetic model can be controlled by simply tuning the  $i_{0,SEI}$  term in Table 5; hence, a lower value for  $i_{0,SEI}$  is used to reduce the capacity loss in each cycle. This is one of the clear benefits of the kinetic-limited model, i.e., the simplicity of implementation and the ease of tuning the model for the BMS's practical application.

The diffusion-limited model displays a more complex behaviour than the kinetic-limited model. As seen in Figure 9, initially, when the SEI layer is thin, the diffusion-limited model displays a linear SEI growth behaviour like that of the kinetic model. However, as

the SEI layer grows thicker, the rate of growth of the SEI layer is reduced with each cycle. To further examine this, consider the comparison of the SEI growth rate,  $j_{SEI}(x, t)$ , between the two models in Figure 9d.

In Figure 9d, in the initial few cycles, the SEI growth rates are similar between the two models. However, with increasing cycles, the diffusion-limited model shows a steady reduction in  $j_{SEI}(t)$ . The primary cause of this reduction can be explained clearly via Equation (40) for the diffusion growth model. The equation is repeated here for the ease of the reader:

$$c_{sol}(x, R_n, t + \Delta t) = \epsilon_{SEI} c_{sol,2} + \frac{L_{SEI}}{D_{sol}} j_{SEI}(x, t) \quad (93)$$

The solvent concentration at the interface between the graphite anode particle and SEI layer,  $c_{sol}(x, R_n, t + \Delta t)$ , is a key parameter in determining the rate of SEI growth. As the diffusion of solvent particles across the SEI layer is very slow, i.e., as  $D_{sol}$  is very small, when the SEI layer grows thicker, the rate at which solvent particles diffuse across the layer becomes slower, meaning that as the SEI layer becomes thicker, the rate of SEI growth slows down. The 'self-limiting' side reaction effect is the reason why the diffusion model's SEI growth rate decreases with an increasing number of cycles.

The phenomenon of a 'self-limiting' side reaction for SEI growth on graphite anode particles is widely described in the literature [61]. For this reason, the diffusion-limited SEI growth model can be considered more accurate for the real-world phenomena of self-limiting SEI growth on graphite anode particles. The diffusion-limited model does, however, come with some added complexity for modelling and more parameters to be tuned in the model. However, the most complexity is in the implementation of the PDAEs. In this work, order reduction assumptions are made to implement the diffusion-limited model: there is a linear diffusion profile across the SEI layer, no SEI growth occurred during discharge, and when  $c_{sol}(x, R_n, t + \Delta t)$  reaches very low levels, negligible SEI growth occurs at that point in the anode.

To accurately implement this model, a full-order solver for the PDAE is required, which can introduce more complexity to the solver. Furthermore, a model for solvent diffusion in the SEI layer during discharge also needs to be defined.

Therefore, considering the merits and limitations of the different models, the choice of which SEI growth model to use should be decided on a case-by-case basis. In cases where a baseline decay of capacity is required for model fitting, the kinetic-limited model may be simpler to apply and tune. In cases where accurate 'self-limiting' SEI growth models are required, the diffusion-limited case is more useful, which may come with the added complexity of accurately tuning the diffusion-limited model parameters to fit the desired behaviour. As the diffusion-limited model has a nonlinear decay behaviour, its model parameters can be quite sensitive, and thus, tuning the diffusion-limited model is a non-trivial task.

## 6. Conclusions

In this work, we propose an iterative solver for the implementation of a physics-based P2D model, which includes the additional models of temperature generation/dissipation and the ageing mechanism of Solid Electrolyte Interphase (SEI) growth. We discretise the P2D model equations using the Finite Volume Method and the Verlet Integration Method. The proposed solver uses physics-based approximate guesses to start the solver, using a convergence criterion based on the average exchange flux expected in the model and a unique bracketing root-finding approach to solve the discretised equations of the P2D model. As a result, the solver can achieve high accuracy with significant performance improvement.

The proposed solver's accuracy is validated against other commercially available solvers and shows high accuracy in different battery operating conditions, namely, the 1C discharge/charge cycle, multiple-C-rate discharge conditions, and multiple dynamic drive cycles. The solver also shows much faster solving speeds compared to commercially

available solvers. We further include the two most used SEI growth models (kinetic-limited and diffusion-limited growth models) in the proposed P2D model to represent ageing and capacity fade in LiBs, and we studied how the proposed solver can accurately represent ageing scenarios.

Considering all the merits, the proposed solver is a better choice for battery researchers to implement a P2D model of a LiB in more realistic scenarios, e.g., the real-time BMS operation of an EV. Our work can still be improved by adding more degradation mechanisms for battery diagnostic applications.

**Author Contributions:** T.W.: Conceptualisation, Methodology, Software, Validation, Analysis, Writing. M.J.: Review and Editing. K.M.: Review and Editing, Supervision. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Appendix A

To derive the average exchange flux in an electrode, the following definition is used:

$$\bar{j}_i(t) = \frac{1}{L_i} \int_x^{L_i+x} j(x, t) \cdot dx \quad (\text{A1})$$

The exchange flux can be related to the ionic current using Equation (7) in Table 2, repeated here for clarity:

$$\frac{\partial i_2(x, t)}{\partial x} = a_{1,i} F j(x, t) \quad (\text{A2})$$

Substituting the spatial derivative of the ionic current into the definition in Equation (A1) yields

$$\bar{j}_i(t) = \frac{1}{a_{1,i} F L_i} \int_x^{L_i+x} \frac{\partial i_2(x, t)}{\partial x} \cdot dx = \frac{1}{a_{1,i} F L_i} \int_x^{L_i+x} \partial i_2(x, t) \quad (\text{A3})$$

This general expression for the average exchange flux can be expressed for the anode and cathode, respectively, as follows:

$$\bar{j}_n(t) = \frac{1}{a_{1,n} F L_n} \int_0^{L_n} \partial i_2(x, t) = \frac{1}{a_{1,n} F L_n} [i_2(L_n, t) - i_2(0, t)] \quad (\text{A4})$$

$$\bar{j}_p(t) = \frac{1}{a_{1,p} F L_p} \int_{L_n+L_s}^{L_{tot}} \partial i_2(x, t) = \frac{1}{a_{1,p} F L_p} [i_2(L_{tot}, t) - i_2(L_n + L_s, t)] \quad (\text{A5})$$

As the value for the ionic current is known at the different material boundaries, the above equations yield the following values:

$$\bar{j}_n(t) = \frac{I_{app}(t)}{a_{1,n} F L_n} \quad (\text{A6})$$

$$\bar{j}_p(t) = \frac{-I_{app}(t)}{a_{1,p} F L_p} \quad (\text{A7})$$

## References

1. Grey, C.P.; Hall, D.S. Prospects for lithium-ion batteries and beyond—A 2030 vision. *Nat. Commun.* **2020**, *11*, 6279. [[CrossRef](#)] [[PubMed](#)]
2. Lu, L.; Han, X.; Li, J.; Hua, J.; Ouyang, M. A review on the key issues for lithium-ion battery management in electric vehicles. *J. Power Sources* **2013**, *226*, 272–288. [[CrossRef](#)]

3. Abada, S.; Marlair, G.; Lecocq, A.; Petit, M.; Sauvant-Moynot, V.; Huet, F. Safety focused modeling of lithium-ion batteries: A review. *J. Power Sources* **2016**, *306*, 178–192. [\[CrossRef\]](#)
4. Liu, W.; Placke, T.; Chau, K.T. Overview of batteries and battery management for electric vehicles. *Energy Rep.* **2022**, *8*, 4058–4084. [\[CrossRef\]](#)
5. Plett, G.L. *Battery Management Systems, Volume I: Battery Modeling*; Artech House: Minto, Australia, 2015; Volume 1.
6. Ramadesigan, V.; Northrop, P.W.C.; De, S.; Santhanagopalan, S.; Braatz, R.D.; Subramanian, V.R. Modeling and Simulation of Lithium-Ion Batteries from a Systems Engineering Perspective. *J. Electrochem. Soc.* **2012**, *159*, R31–R45. [\[CrossRef\]](#)
7. Zhang, C.; Li, K.; McLoone, S.; Yang, Z. Battery modelling methods for electric vehicles—A review. In Proceedings of the 2014 European Control Conference, ECC 2014, Strasbourg, France, 24–27 June 2014; pp. 2673–2678. [\[CrossRef\]](#)
8. Liaw, B.Y.; Nagasubramanian, G.; Jungst, R.G.; Doughty, D.H. Modeling of lithium ion cells—A simple equivalent-circuit model approach. *Solid State Ionics* **2004**, *175*, 835–839. [\[CrossRef\]](#)
9. Seaman, A.; Dao, T.S.; McPhee, J. A survey of mathematics-based equivalent-circuit and electrochemical battery models for hybrid and electric vehicle simulation. *J. Power Sources* **2014**, *256*, 410–423. [\[CrossRef\]](#)
10. Jokar, A.; Rajabloo, B.; Désilets, M.; Lacroix, M. Review of simplified Pseudo-two-Dimensional models of lithium-ion batteries. *J. Power Sources* **2016**, *327*, 44–55. [\[CrossRef\]](#)
11. Doyle, M.; Fuller, T.F.; Newman, J. Modeling of Galvanostatic Charge and Discharge of the Lithium/Polymer/Insertion Cell. *J. Electrochem. Soc.* **1993**, *140*, 1526–1533. [\[CrossRef\]](#)
12. Fuller, T.F.; Doyle, M.; Newman, J. Simulation and Optimization of the Dual Lithium Ion Insertion Cell. *J. Electrochem. Soc.* **1994**, *141*, 1. [\[CrossRef\]](#)
13. Rao, L.; Newman, J. Heat-Generation Rate and General Energy Balance for Insertion Battery Systems. *J. Electrochem. Soc.* **1997**, *144*, 2697–2704. [\[CrossRef\]](#)
14. Farag, M.; Sweity, H.; Fleckenstein, M.; Habibi, S. Combined electrochemical, heat generation, and thermal model for large prismatic lithium-ion batteries in real-time applications. *J. Power Sources* **2017**, *360*, 618–633. [\[CrossRef\]](#)
15. Ramadass, P.; Haran, B.; Gomadam, P.M.; White, R.; Popov, B.N. Development of First Principles Capacity Fade Model for Li-Ion Cells. *J. Electrochem. Soc.* **2004**, *151*, A196. [\[CrossRef\]](#)
16. Pinson, M.B.; Bazant, M.Z. Theory of SEI Formation in Rechargeable Batteries: Capacity Fade, Accelerated Aging and Lifetime Prediction. *J. Electrochem. Soc.* **2012**, *160*, A243–A250. [\[CrossRef\]](#)
17. Ramos, A.M. On the well-posedness of a mathematical model for Lithium-ion batteries. *Appl. Math. Model.* **2015**, *40*, 115–125. [\[CrossRef\]](#)
18. Bermejo, R. Numerical analysis of a finite element formulation of the P2D model for Lithium-ion cells. *Numer. Math.* **2021**, *149*, 463–505. [\[CrossRef\]](#)
19. Zhang, D.; Popov, B.N.; White, R.E. Modeling Lithium Intercalation of a Single Spinel Particle under Potentiodynamic Control. *J. Electrochem. Soc.* **2000**, *147*, 831. [\[CrossRef\]](#)
20. Luo, W.; Lyu, C.; Wang, L.; Zhang, L. An approximate solution for electrolyte concentration distribution in physics-based lithium-ion cell models. *Microelectron. Reliab.* **2013**, *53*, 797–804. [\[CrossRef\]](#)
21. Kim, G.H.; Smith, K.; Lawrence-Simon, J.; Yang, C. Efficient and Extensible Quasi-Explicit Modular Nonlinear Multiscale Battery Model: GH-MSMD. *J. Electrochem. Soc.* **2017**, *164*, A1076–A1088. [\[CrossRef\]](#)
22. Ma, Y.; Yin, M.; Ying, Z.; Chen, H. Establishment and simulation of an electrode averaged model for a lithium-ion battery based on kinetic reactions. *RSC Adv.* **2016**, *6*, 25435–25443. [\[CrossRef\]](#)
23. Perez, H.E.; Hu, X.; Moura, S.J. Optimal charging of batteries via a single particle model with electrolyte and thermal dynamics. In Proceedings of the 2016 American Control Conference (ACC), Boston, MA, USA, 6–8 July 2016; pp. 4000–4005. [\[CrossRef\]](#)
24. Yin, X.; Zhang, D. batP2dFoam: An Efficient Segregated Solver for the Pseudo-2-Dimensional (P2D) Model of Li-Ion Batteries. *J. Electrochem. Soc.* **2023**, *170*, 030521. [\[CrossRef\]](#)
25. Chen, Q.; Chen, X.S.; Li, Z. A fast numerical method with non-iterative source term for pseudo-two-dimension lithium-ion battery model. *J. Power Sources* **2023**, *577*, 233258. [\[CrossRef\]](#)
26. Ai, W.; Liu, Y. Improving the convergence rate of Newman’s battery model using 2nd order finite element method. *J. Energy Storage* **2023**, *67*, 107512. [\[CrossRef\]](#)
27. Chayambuka, K.; Mulder, G.; Danilov, D.L.; Notten, P.H. Physics-based modeling of sodium-ion batteries part II. Model and validation. *Electrochim. Acta* **2022**, *404*, 139764. [\[CrossRef\]](#)
28. Jiang, Y.; Zhang, L.; Offer, G.; Wang, H. A user-friendly lithium battery simulator based on open-source CFD. *Digit. Chem. Eng.* **2022**, *5*, 100055. [\[CrossRef\]](#)
29. Han, S.; Tang, Y.; Rahimian, S.K. A numerically efficient method of solving the full-order pseudo-2-dimensional (P2D) Li-ion cell model. *J. Power Sources* **2021**, *490*, 229571. [\[CrossRef\]](#)
30. Geng, Z.; Wang, S.; Lacey, M.J.; Brandell, D.; Thiringer, T. Bridging physics-based and equivalent circuit models for lithium-ion batteries. *Electrochim. Acta* **2021**, *372*, 137829. [\[CrossRef\]](#)
31. Han, R.; Macdonald, C.; Wetton, B. A fast solver for the pseudo-two-dimensional model of lithium-ion batteries. *arXiv* **2021**, arXiv:2111.09251.

32. Noor, A.S.; Kadhim, A.H.; Ali, M.A.; Zülke, A.; Korotkin, I. Dandelion v1: An Extremely Fast Solver for the Newman Model of Lithium-Ion Battery (Dis)charge. *J. Electrochem. Soc.* **2021**, *168*, 060544. [\[CrossRef\]](#)
33. Lee, S.B.; Onori, S. A Robust and Sleek Electrochemical Battery Model Implementation: A MATLAB® Framework. *J. Electrochem. Soc.* **2021**, *168*, 090527. [\[CrossRef\]](#)
34. Esfahanian, V.; Chaychizadeh, F.; Dehghandorost, H.; Shokouhmand, H. An Efficient Thermal-Electrochemical Simulation of Lithium-Ion Battery Using Proper Mathematical-Physical CFD Schemes. *J. Electrochem. Soc.* **2019**, *166*, A1520–A1534. [\[CrossRef\]](#)
35. Guo, M.; Jin, X.; White, R.E. Nonlinear State-Variable Method for Solving Physics-Based Li-Ion Cell Model with High-Frequency Inputs. *J. Electrochem. Soc.* **2017**, *164*, E3001–E3015. [\[CrossRef\]](#)
36. Torchio, M.; Magni, L.; Gopaluni, R.B.; Braatz, R.D.; Raimondo, D.M. LIONSIMBA: A Matlab Framework Based on a Finite Volume Model Suitable for Li-Ion Battery Design, Simulation, and Control. *J. Electrochem. Soc.* **2016**, *163*, A1192–A1205. [\[CrossRef\]](#)
37. Tulsyan, A.; Tsai, Y.; Gopaluni, R.B.; Braatz, R.D. State-of-charge estimation in lithium-ion batteries: A particle filter approach. *J. Power Sources* **2016**, *331*, 208–223. [\[CrossRef\]](#)
38. Wickramanayake, T.; Javadipour, M.; Mehran, K. A Novel Root-Finding Algorithm to Solve the Pseudo-2D Model of a Lithium-ion Battery. In Proceedings of the 2023 IEEE International Conference on Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles and International Transportation Electrification Conference, ESARS-ITEC 2023, Venice, Italy, 29–31 March 2023. [\[CrossRef\]](#)
39. Mai, W.; Colclasure, A.; Smith, K. A Reformulation of the Pseudo2D Battery Model Coupling Large Electrochemical-Mechanical Deformations at Particle and Electrode Levels. *J. Electrochem. Soc.* **2019**, *166*, A1330–A1339. [\[CrossRef\]](#)
40. Liu, L.; Park, J.; Lin, X.; Sastry, A.M.; Lu, W. A thermal-electrochemical model that gives spatial-dependent growth of solid electrolyte interphase in a Li-ion battery. *J. Power Sources* **2014**, *268*, 482–490. [\[CrossRef\]](#)
41. Birkl, C.R.; Roberts, M.R.; McTurk, E.; Bruce, P.G.; Howey, D.A. Degradation diagnostics for lithium ion cells. *J. Power Sources* **2017**, *341*, 373–386. [\[CrossRef\]](#)
42. Edge, J.S.; O’Kane, S.; Prosser, R.; Kirkaldy, N.D.; Patel, A.N.; Hales, A.; Ghosh, A.; Ai, W.; Chen, J.; Yang, J.; et al. Lithium ion battery degradation: What you need to know. *Phys. Chem. Chem. Phys.* **2021**, *23*, 8200–8221. [\[CrossRef\]](#)
43. O’Kane, S.E.; Ai, W.; Madabattula, G.; Alonso-Alvarez, D.; Timms, R.; Sulzer, V.; Edge, J.S.; Wu, B.; Offer, G.J.; Marinescu, M. Lithium-ion battery degradation: How to model it. *Phys. Chem. Chem. Phys.* **2022**, *24*, 7909–7922. [\[CrossRef\]](#) [\[PubMed\]](#)
44. Reniers, J.M.; Mulder, G.; Howey, D.A. Review and Performance Comparison of Mechanical-Chemical Degradation Models for Lithium-Ion Batteries. *J. Electrochem. Soc.* **2019**, *166*, A3189–A3200. [\[CrossRef\]](#)
45. Lamorgese, A.; Mauri, R.; Tellini, B. Electrochemical-thermal P2D aging model of a LiCoO<sub>2</sub>/graphite cell: Capacity fade simulations. *J. Energy Storage* **2018**, *20*, 289–297. [\[CrossRef\]](#)
46. Awarke, A.; Pischinger, S.; Ogrzewalla, J. Pseudo 3D Modeling and Analysis of the SEI Growth Distribution in Large Format Li-Ion Polymer Pouch Cells. *J. Electrochem. Soc.* **2013**, *160*, A172–A181. [\[CrossRef\]](#)
47. Ashwin, T.R.; Chung, Y.M.; Wang, J. Capacity fade modelling of lithium-ion battery under cyclic loading conditions. *J. Power Sources* **2016**, *328*, 586–598. [\[CrossRef\]](#)
48. Kamyab, N.; Weidner, J.W.; White, R.E. Mixed Mode Growth Model for the Solid Electrolyte Interface (SEI). *J. Electrochem. Soc.* **2019**, *166*, A334–A341. [\[CrossRef\]](#)
49. Das, S.; Attia, P.M.; Chueh, W.C.; Bazant, M.Z. Electrochemical Kinetics of SEI Growth on Carbon Black: Part II. Modeling. *J. Electrochem. Soc.* **2019**, *166*, E107–E118. [\[CrossRef\]](#)
50. Cheng, H.; Sun, Q.; Li, L.; Zou, Y.; Wang, Y.; Cai, T.; Zhao, F.; Liu, G.; Ma, Z.; Wahyudi, W.; et al. Emerging Era of Electrolyte Solvation Structure and Interfacial Model in Batteries. *ACS Energy Lett.* **2022**, *7*, 490–513. [\[CrossRef\]](#)
51. Botte, G.G.; Ritter, J.A.; White, R.E. Comparison of finite difference and control volume methods for solving differential equations. *Comput. Chem. Eng.* **2000**, *24*, 2633–2654. [\[CrossRef\]](#)
52. Kong, X.R.; Wetton, B.; Gopaluni, B. Assessment of Simplifications to a Pseudo-2D Electrochemical Model of Li-ion Batteries. *IFAC-PapersOnLine* **2019**, *52*, 946–951. [\[CrossRef\]](#)
53. Zwillinger, D. *CRC Standard Mathematical Tables and Formulae*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2003; p. 910.
54. Butcher, J.C. *Numerical Methods for Ordinary Differential Equations*; John Wiley & Sons: Hoboken, NJ, USA, 2016; pp. 1–513. [\[CrossRef\]](#)
55. Hairer, E.; Lubich, C.; Wanner, G. Geometric numerical integration illustrated by the Störmer-Verlet method. *Acta Numer.* **2000**, *12*, 399–450. [\[CrossRef\]](#)
56. Hosseinzadeh, E.; Marco, J.; Jennings, P. Electrochemical-Thermal Modelling and Optimisation of Lithium-Ion Battery Design Parameters Using Analysis of Variance. *Energies* **2017**, *10*, 1278. [\[CrossRef\]](#)
57. Dahlquist, G.; Björck, A. *Numerical Methods*; Anderson, N., Translator; Prentice-Hall Series in Automatic Computation; Prentice-Hall: Englewood Cliffs, NJ, USA, 1974.
58. Chalise, D.; Lu, W.; Srinivasan, V.; Prasher, R. Heat of Mixing During Fast Charge/Discharge of a Li-Ion Cell: A Study on NMC523 Cathode. *J. Electrochem. Soc.* **2020**, *167*, 090560. [\[CrossRef\]](#)
59. Mayer, S. *Electric Vehicle Dynamic-Stress-Test Cycling Performance of Lithium-Ion Cells*; Lawrence Livermore National Lab. (LLNL): Livermore, CA, USA, 1994. [\[CrossRef\]](#)

- 
60. COMSOL Inc. *1D Lithium-Ion Battery Drive-Cycle Monitoring*; COMSOL: Stockholm, Sweden, 2023.
  61. Agubra, V.A.; Fergus, J.W. The formation and stability of the solid electrolyte interface on the graphite anode. *J. Power Sources* **2014**, *268*, 153–162. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.