



# Model simplifications and their impact on computational complexity for an electrochemistry-based battery modeling toolbox

Z. Khalik <sup>a,\*</sup>, M.C.F. Donkers <sup>a</sup>, H.J. Bergveld <sup>a,b</sup>

<sup>a</sup> Department of Electrical Engineering, Eindhoven University of Technology, Groene Loper 19, 5612AP Eindhoven, The Netherlands

<sup>b</sup> NXP Semiconductors, High Tech Campus 46, 5656 AE Eindhoven, The Netherlands

## ARTICLE INFO

### Keywords:

Battery modeling  
Battery model implementation  
Battery simulation toolbox  
Battery model simplifications  
Battery simulation

## ABSTRACT

Using electrochemistry-based battery models in battery management systems remains challenging due to their computational complexity. In this paper, we study for the first time the impact of several types of model simplifications on the trade-off between model accuracy and computation time for the Doyle–Fuller–Newman (DFN) model. As a basis for comparison, we consider, to what we refer as, the complete DFN (CDFN) model, which is a DFN model without any simplifications, and includes the concentration-dependency of parameters that have been studied in previous literature. Furthermore, we propose a highly efficient implementation of the CDFN model that leads to a considerable decrease in computation time, and is developed into a freely downloadable toolbox. This toolbox allows the user to easily toggle between the studied simplifications to make the desired trade-off between model accuracy and computation time. We compare several simplified DFN models to the single-particle-model and the CDFN model. Here, we show that with the proposed implementation, and by selectively making the proposed simplifications, as well as selectively choosing the grid parameters, a model can be obtained that has a minor impact on model accuracy, achieving a simulation time of over 5000 times faster than real-time.

## 1. Introduction

With the current trend in rising energy densities of batteries, battery-powered electric vehicles are becoming an increasingly competitive alternative to other common technologies, such as the combustion engine. Battery models are required for battery management system functionalities, such as state-of-charge estimation [1], state-of-health estimation [2], and fast charging [3,4]. Due to their computational efficiency, equivalent-circuit models [5] are often used for this purpose [6,7], where the battery is modeled using (passive) circuit elements. However, these types of models provide limited information on the internal states of the battery, which can be important in control applications or in analysis of battery behavior. Therefore, electrochemistry-based models have gained increasing popularity in such applications [2,3]. The Doyle–Fuller–Newman (DFN) model is a widely used electrochemistry-based model, which is described by a set of partial differential equations (PDEs) [8]. While the DFN model can describe internal states of the battery, its model complexity generally leads to large computation times. This model complexity, as well as the difficulty to uniquely determine all model parameters or to uniquely determine the internal states based on the measured voltages and currents, prohibits using the DFN model in battery management system functionalities.

There are several ways to reduce the computational burden of the DFN model. One way to reduce complexity is to apply model reduction to the DFN model. A common technique is to make a polynomial approximation of the solid-phase concentration diffusion [9,10]. Model reduction is also applied on the DFN model through the use of techniques such as Galerkin projections [11] and proper orthogonal decomposition [12,13]. An overview of this type of model reduction can be found in [14]. However, the reduction is mostly in the number of equations, while the reduction in computation time is only marginal, when applied to the DFN model, as was shown in [13]. Another way to reduce complexity is by simplifying the model equations, see, e.g., [10,15–18]. A popular example is the so-called single-particle model (SPM) [1,19], in which the equations for solid-phase concentration and potentials are simplified. Another approach is to linearize the governing equations of the DFN model, see, e.g., [20]. Finally, it has been recognized that many parameters depend on concentrations [21–25], while actually ignoring these dependencies leads to a simplification of the governing model equations [26].

Besides addressing the complexity of the DFN model, another way to reduce computation time is to develop algorithms that compute the solution to the DFN model more efficiently, see e.g. [13,26,27],

\* Corresponding author.

E-mail addresses: [z.khalik@tue.nl](mailto:z.khalik@tue.nl) (Z. Khalik), [m.c.f.donkers@tue.nl](mailto:m.c.f.donkers@tue.nl) (M.C.F. Donkers), [h.j.bergveld@tue.nl](mailto:h.j.bergveld@tue.nl) (H.J. Bergveld).

<https://doi.org/10.1016/j.jpowsour.2020.229427>

Received 2 May 2020; Received in revised form 17 December 2020; Accepted 27 December 2020

Available online 12 January 2021

0378-7753/© 2021 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

and [28], where the latter proposes a computationally efficient model implementation in presence of above-discussed simplified solid-phase dynamics. In these papers, the PDEs that describe the DFN model are spatially and temporally discretized, which results in a set of nonlinear algebraic equations (AEs). Numerical methods are then developed in an attempt to solve this set of AEs as efficiently as possible.

In this paper, we study for the first time the impact of several types of model simplifications on the trade-off between model accuracy and computation time for the DFN model. As a basis for comparison, we consider, to what we will refer as, the complete DFN (CDFN) model, which is a DFN model without any simplifications, and includes the concentration-dependency of parameters that have been studied in previous literature [21–25]. The CDFN and the considered simplifications are given in Section 2. Furthermore, we propose a computationally efficient implementation of the CDFN model that leads to a significant reduction in computation time, which is presented in Section 3. After spatial and temporal discretization of the DFN model, we will show that through substitution of equations, the number of equations describing the DFN model can be considerably reduced to a small set of algebraic equations. This model implementation has been developed into a freely downloadable toolbox, which will also be presented in Section 3. This toolbox allows the user to easily toggle between the studied simplifications to make the desired trade-off between model accuracy and computation time.

We will study the impact of the proposed simplifications on model accuracy and computation time for two different sets of parameters, to show the validity of these simplifications and the trade-off between computation time and model accuracy that can be made. Furthermore, we will study the impact of varying spatial accuracy on the trade-off between model accuracy and computation time. We will validate the proposed model implementation against the implementation presented in [29], and also show that the proposed implementation is significantly faster than the model implementations presented in [29] and [13]. It is important to note here that the implementation in [29] relies on a variable-time-step solver, while the proposed implementation is based on a fixed-time-step discretization scheme. We will reflect on this choice in the results section. Finally, we will compare several simplified DFN models to the SPM and the CDFN model. The results of these studies are presented in Section 4, after which conclusions are drawn in Section 5.

## 2. Battery modeling

In this section, we briefly formulate the DFN model, and introduce several simplifications, which we will show in Section 4 to have no

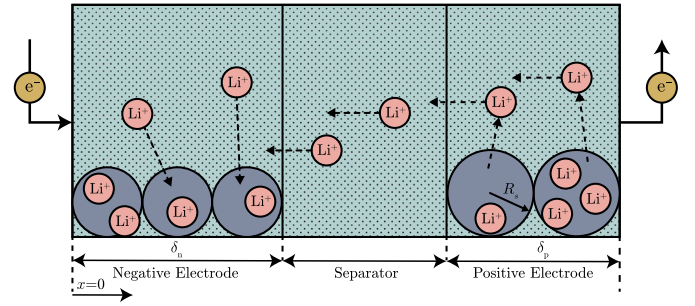


Fig. 1. DFN modeling approach for a Li-ion cell.

significant impact on the accuracy of the model, both in the input–output behavior as well as the internal states. Further simplifications on the resulting model will be applied to arrive at the so-called SPM.

### 2.1. Doyle–Fuller–Newman model

The DFN model is a widely used electrochemistry-based model introduced in [8]. Fig. 1 illustrates the modeling approach for a Li-ion cell. In the  $x$  dimension, the cell is divided into three regions, namely the negative electrode, the separator, and the positive electrode. In the electrodes, Li-ions exist essentially in two phases. In the solid phase, Li-ions are intercalated into the solid-phase material, which is represented by spheres with radius  $R_s$ . In the electrolyte phase, Li-ions exist in a dissolved state in the electrolyte. In the separator, Li-ions exist only in the electrolyte phase. During charging, as shown in Fig. 1, intercalated Li-ions exit the solid particles in the positive electrode and enter the solid particles in the negative electrode. During discharging, the opposite process happens.

The governing equations of the DFN model are summarized in Table 1 and the symbols used in this model formulation are given in Table 2. For compactness of notation, where possible, the time and space dependency of the variables given have been left out of the equations. Note that  $U$  in (5c) denotes the equilibrium potential of the electrode, which can be given by a pre-defined function typically of the solid-phase concentration at the solid–electrolyte interface ( $\bar{c}_s(x, t) = c_s(R_s, x, t)$ ). The solid-phase variables (i.e.,  $c_s(r, x, t)$ ,  $\phi_s(x, t)$ ,  $j_n(x, t)$ ,  $i_0(x, t)$ ,  $\bar{c}_s(x, t)$ ,  $\eta(x, t)$ , and  $U(x, t)$ ) are defined only in the electrodes, i.e., for  $x \in [0, \delta_n] \cup [L - \delta_p, L]$  and assumed zero for  $x \in (\delta_n, L - \delta_p)$ , while the electrolyte-phase variables (i.e.,  $c_e(x, t)$  and  $\phi_e(x, t)$ ), are defined

Table 1  
Governing equations of the DFN model.

Solid-phase Li-ion concentration	$\frac{\partial c_s}{\partial t} = \frac{D_s}{r^2} \frac{\partial}{\partial r} \left( r^2 \frac{\partial c_s}{\partial r} \right)$	(1a)
Boundary condition	$\frac{\partial c_s}{\partial r} \Big _{r=0} = 0, -D_s \frac{\partial c_s}{\partial r} \Big _{r=R_s} = j_n$	(1b)
Electrolyte-phase Li-ion concentration	$\epsilon_e \frac{\partial c_e}{\partial t} = \frac{\partial}{\partial x} \left( D_e \epsilon_e^p \frac{\partial c_e}{\partial x} \right) + \frac{3\epsilon_e(1-\nu)}{R_s} j_n$	(2a)
Boundary condition	$\frac{\partial c_e}{\partial x} \Big _{x=0} = \frac{\partial c_e}{\partial x} \Big _{x=L} = 0$	(2b)
Solid-phase potential	$\frac{\partial}{\partial x} \left( \sigma_e \frac{\partial \phi_s}{\partial x} \right) = \frac{3\epsilon_e F}{R_s} j_n$	(3a)
Boundary condition	$\sigma_e \frac{\partial \phi_s}{\partial x} \Big _{x=0} = \sigma_e \frac{\partial \phi_s}{\partial x} \Big _{x=L} = \frac{i_{app}}{A_{surf}}$	(3b)
	$\frac{\partial \phi_s}{\partial x} \Big _{x=\delta_n} = \frac{\partial \phi_s}{\partial x} \Big _{x=L-\delta_p} = 0$	(3c)
Electrolyte-phase potential	$\frac{\partial}{\partial x} \left( \kappa \epsilon_e^p \frac{\partial \phi_e}{\partial x} + \kappa \epsilon_e^p \frac{2RT}{F} \frac{\partial \ln c_e}{\partial x} \right) = -\frac{3\epsilon_e F}{R_s} j_n$	(4a)
Boundary condition	$\frac{\partial \phi_e}{\partial x} \Big _{x=0} = \frac{\partial \phi_e}{\partial x} \Big _{x=L} = 0$	(4b)
Butler–Volmer kinetics	$j_n = \frac{i_0}{F} \left( \exp \left( \frac{a_e F}{RT} \eta \right) - \exp \left( -\frac{a_e F}{RT} \eta \right) \right)$	(5a)
Exchange current density	$i_0 = k_0 c_e^{a_e} (c_{s,max})^{a_s} \bar{c}_s^{a_s'}$	(5b)
Electrode over-potential	$\eta = \phi_s - \phi_e - U$	(5c)
Terminal voltage	$V(t) = \phi_s(L, t) - \phi_s(0, t) - \frac{R_s}{A_{surf}} i_{app}(t)$	(6)

**Table 2**

Symbols used in the DFN model formulation.

Symbol	Description
$A_{surf}$	Active electrode area
$c_e$	Li-ion concentration in the electrolyte phase
$c_s$	Li-ion concentration in the solid phase
$\bar{c}_s$	Solid-phase concentration at the solid–electrolyte interface
$c_s^{max}$	Maximum solid-phase concentration
$D_e$	Li-ion diffusion coefficient in electrolyte
$D_s$	Li-ion diffusion coefficient in the solid phase
$F$	Faraday's constant
$f_{\pm}$	Mean molar salt activity coefficient
$i_{app}$	Applied current through the battery
$i_0$	Exchange current density
$j_n$	Net molar flux of Li-ions exiting the particle
$k_0$	Kinetic constant
$L$	Cell thickness
$p$	Bruggeman porosity exponent
$r$	Radial position across a spherical particle
$R$	Universal gas constant
$R_f$	Empirical film resistance
$R_s$	Radius of active material particles
$t$	Time
$T$	Absolute temperature
$t_+^0$	Transference number
$U$	Equilibrium potential of the electrode
$V_i$	Terminal voltage
$x$	Position across cell
<i>Greek</i>	
$\alpha_a$	Anodic charge-transfer coefficient
$\alpha_c$	Cathodic charge-transfer coefficient
$\delta_n$	Negative electrode thickness
$\delta_p$	Positive electrode thickness
$\epsilon_s$	Active particles volume fraction
$\epsilon_e$	Electrolyte volume fraction
$\eta$	Overpotential at the electrodes
$\kappa$	Ionic conductivity
$\nu$	Thermodynamic factor, equal to $(t_+^0 - 1) \left( 1 + \frac{d \ln f_{\pm}}{d \ln c_e} \right)$
$\sigma$	Electrical conductivity
$\phi_e$	Potential in the electrolyte phase
$\phi_s$	Potential in the solid phase

over the whole length of the cell, i.e., for  $x \in [0, L]$ . Further note that, in (4b),  $i_{app} > 0$  indicates charging.

## 2.2. Concentration-dependent parameters

Aside from physical constants, such as the universal gas constant  $R$  and Faraday's constant  $F$ , the material properties that define the parameters of the DFN model (and SPM) need to be determined. However, numerous literature, e.g. [21–25], has shown that at least some of these material properties are not necessarily constant and can significantly change depending on the Li-ion concentration level of the material. Measuring these parameters can involve destructive procedures, such as opening of the battery cell, which makes determining the dependencies difficult.

Numerous results on the measurement and use of concentration-dependent parameters have been reported in the literature, e.g., [8, 12, 21–25, 29–38]. Since the electrolyte is relatively easy to isolate, electrolyte-related parameters, and more specifically, the transport properties, such as  $\kappa$  and  $D_e$ , are more often measured than the parameters of the electrodes, such as  $D_s$ . The concentration dependency of the electrolyte transport properties have been measured and shown for various different electrolytes in e.g., [21, 22, 25]. The concentration-dependent parameters provided in these papers have been used in various literature, e.g. [8, 12, 29–33]. However, often  $f_{\pm}$  is still assumed constant, and is therefore ignored in the DFN equations, even though in [21, 22, 39],  $f_{\pm}$  is shown to significantly change over the concentration in the electrolyte. Out of all the parameters, the concentration dependency of the conductivity in the electrolyte  $\kappa$  seems to be the least

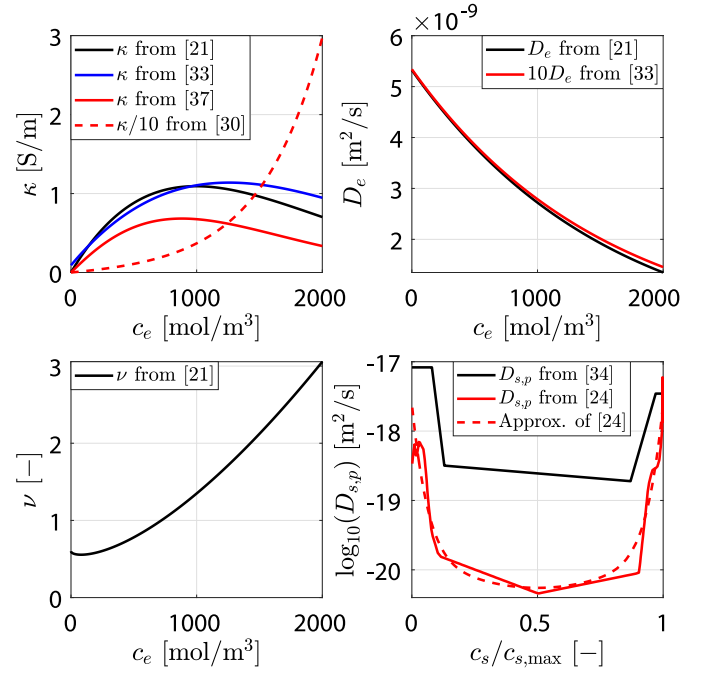


Fig. 2. Various examples of concentration-dependent parameters found in literature.

difficult to determine, as it is most commonly taken as a concentration-dependent parameter in the DFN model, as done in e.g., [8, 30]. The other electrolyte transport properties, i.e., the activity coefficient  $f_{\pm}$ , transference number  $t_+^0$ , and diffusion coefficient  $D_e$  are generally more difficult to determine. The literature on the measurement of the concentration dependency of the electrode parameters is more scarce, with the exception of the diffusion coefficient  $D_s$ , of which its concentration-dependency has been shown in e.g. [23, 24, 34–36]. However, such results have not yet been widely applied in the context of the DFN model, with [40] being the only example, to the authors' knowledge.

In Fig. 2, some examples of concentration-dependent parameters are shown. Note that there seems to be a discrepancy between the examples obtained from [30] and [37], even though the authors in [30] (indirectly) cite the paper of [37] as their source of the concentration-dependent conductivity function. We further see that the parameters can change significantly, and in the case of  $D_s$  in orders of magnitudes, over a varying concentration. This would suggest that at large currents, which would induce large concentration gradients, the parameter values can significantly change, suggesting the need to implement them as a function of concentration. In Section 4, we will show that this may not necessarily be the case for any of the parameters shown in Fig. 2.

## 2.3. Simplifications to the Doyle–Fuller–Newman model

Introducing parameter dependencies introduces additional complexity in the DFN model. Instead of increasing the complexity, we can also make several simplifications to the DFN model that aim at reducing complexity. In particular,

[S1] The rate equation (5a) can be linearized with respect to the overpotential  $\eta$  around the origin, due to the fact that  $\frac{F}{RT} \gg |\eta|$ , which also implies that as  $|\eta|$  becomes larger, the errors made due to this assumption also become larger. Note that  $\alpha_a + \alpha_c = 1$ , which allows the resulting linearized Butler–Volmer equation of (5a)–(5c) to be written as

$$j_n = \frac{i_0}{RT} (\phi_s - \phi_e - U). \quad (7)$$

[S2] A zeroth-order Taylor approximation can be made for the concentration-dependent parameters, i.e.,  $\kappa(c_e) = \kappa(c_e^*)$ ,  $D_e(c_e) = D_e(c_e^*)$ ,  $v(c_e) = v(c_e^*)$ ,  $D_s(\theta) = D_s(\theta^*)$ , where  $\theta$  is the stoichiometry  $c_s/c_{s,\max}$ ,  $c_e^*$  is the evaluation point chosen for  $c_e$ , and  $\theta^*$  is the evaluation point chosen for  $\theta$ , which can be chosen in either of the following ways.

[S2-I] The evaluation points are chosen dynamically, such that they vary over space and time. In this paper, the evaluation point is chosen after time discretization, as the values of the concentration-dependent parameters at the previous time sample  $t_{k-1}$ , i.e.,  $\kappa(c_e^*) = \kappa(c_e(t_{k-1}))$ ,  $D_e(c_e^*) = D_e(c_e(t_{k-1}))$ ,  $v(c_e^*) = v(c_e(t_{k-1}))$ ,  $D_s(\theta^*) = D_s(\theta(t_{k-1}))$ .

[S2-II] The evaluation points are chosen as a constant value, such that the concentration-dependent parameters no longer vary over space and time, i.e.,  $\kappa(c_e^*) = \kappa(\bar{c}_e)$ ,  $D_e(c_e^*) = D_e(\bar{c}_e)$ ,  $v(c_e^*) = v(\bar{c}_e)$ ,  $D_s(\theta^*) = D_s(\bar{\theta})$ . Here,  $\bar{c}_e$  is the expected average concentration in the electrolyte, which can be chosen as  $\bar{c}_e = c_{e,0}$ , and  $\bar{\theta}$  is the expected average stoichiometry, which can be chosen as  $\bar{\theta} = (\theta_{100\%} + \theta_{0\%})/2$ .

[S3] Finally, a common simplification that is made, is to make a two-parameter polynomial approximation of the solid-phase diffusion [9,41]. Here, the concentration profile within a particle is assumed to be a parabola over the radial dimension  $r$ . Then, by substituting this approximated concentration into (1a), volume-averaging the resulting expression, and evaluating the boundary conditions (1b), expressions for the bulk concentration  $c_{s,\text{bulk}}$  and surface concentration  $\bar{c}_s$  can be obtained, i.e.,

$$\frac{dc_{s,\text{bulk}}}{dt} = \frac{-3}{R_s} j_n, \quad \bar{c}_s = c_{s,\text{bulk}} - \frac{R_s}{5D_s} j_n \quad (8)$$

The main advantage of this simplification is that the diffusion equation in (1), which is the only governing equation that depends on  $r$ , simplifies to merely two equations, describing the bulk and surface concentration of the particles in the solid phase as a function of  $x$  and  $t$ .

## 2.4. Single-particle model

The SPM is defined by the main assumption that the diffusion dynamics inside the solid particles is the slowest process, and therefore dominates over the other dynamics [42]. Under this assumption, we can formulate the following corresponding simplifications [1].

[S4] The concentration in the solid phase  $c_s$  is constant over  $x$ .

[S5] The exchange current density  $i_0$  is approximated by  $\bar{i}_0 = i_0(\bar{c}_s, \bar{c}_e)$ , where  $\bar{c}_e$  is the mean electrolyte concentration over each of the electrodes.

[S6] The over-potential  $\eta$  is approximated by  $\bar{\eta} = \phi_s - \bar{\phi}_e - U$ , where  $\bar{\phi}_e$  is the mean electrolyte potential over each of the electrodes.

Note that as a consequence of [S4] and [S5],  $\bar{i}_0$  only changes over time. As a result of [S4], the flux of Li-ions at the solid–electrolyte interface  $j_n$  can be assumed to be equal for all particles in each of the electrodes, and hence  $j_n$  can be directly computed from (3), i.e.,

$$j_n(x, t) = \begin{cases} -\frac{i_{\text{app}}(t)}{A_{\text{surf}} a_s \delta_n F} & \text{for } x \in [0, \delta_n] \\ \frac{i_{\text{app}}(t)}{A_{\text{surf}} a_s \delta_p F}, & \text{for } x \in [L - \delta_p, L]. \end{cases} \quad (9)$$

The solid-phase potential  $\phi_s$  can then be computed from (5), with  $\phi_e = \bar{\phi}_e$ . Note that, as a result of (9), we have that (1), (2), and (4) are no longer coupled, which means that these equations can be solved explicitly for a given  $i_{\text{app}}$ .

We will refer to this model, given by (1), (2), (4), (5), (6), (9), as the SPM. Note that in literature the SPM that includes electrolyte

dynamics, as presented above is usually referred to as the SPMe (Single-Particle Model with electrolyte dynamics), while SPM is usually referred to the model that is obtained under the additional assumption that electrolyte dynamics, corresponding to (2) and (4), are ignored. We further note that in our formulation of the SPM, the parameters can be concentration-dependent, and therefore Simplifications [S1]–[S3] can also be applied to the SPM.

## 3. Model implementation

The objective of this paper is to compare the computational complexity and the accuracy of several electrochemistry-based models, and to implement the DFN model such that a trade-off can be made between computational complexity and accuracy. The DFN model formulated in Section 2.1, under the assumption that the parameters considered in Section 2.2 are concentration-dependent, can be seen as the (most) complete DFN model (CDFN). From this CDFN, any of the simplifications formulated in Section 2.3 can be applied to arrive at a simplified form of the CDFN. The implementation of the CDFN and its corresponding simplifications involves several steps. Firstly, fairly standard spatial and temporal discretization is applied to arrive at a set of AEs. Secondly, the set of AEs is reduced to a smaller set of AEs through substitution, after which the resulting set of AEs can be solved using Newton's method. The discretization procedure described here is similar to the procedure presented in [13]. Therefore, in this section, we will shortly summarize this procedure to formulate the set of AEs that arise from the discretization. For further details on the discretization approach, the reader is referred to [13]. Note that a key difference between the discretization described in [13] and this paper, is that in the finite-volume method (FVM) discretization, to determine the edge parameter values of the control volumes, the harmonic mean of two neighboring control volumes is used, as described in [29]. Finally, we will present a freely downloadable toolbox to simulate the DFN model based on the proposed implementation, which can be used without any knowledge on the proposed model implementation.

### 3.1. Discretization

As a first step, spatial discretization is applied on the PDEs (1)–(4). The equation describing the diffusion of the solid-phase concentration (1) is discretized along the radial direction using a finite-difference-method (FDM), to arrive at a set of differential algebraic equations (DAEs). The other Eqs. (2)–(4) are discretized using a FVM, after which the resulting set of nonlinear DAEs can be written as

$$\frac{d}{dt} \mathbf{c}_s = A_{c_s} \mathbf{c}_s + B_{c_s} \mathbf{j}_n \quad (10a)$$

$$\frac{d}{dt} \mathbf{c}_e = A_{c_e} \mathbf{c}_e + B_{c_e} \mathbf{j}_n \quad (10b)$$

$$0 = A_{\phi_s} \boldsymbol{\phi}_s + B_{\phi_s} \mathbf{j}_n + C_{\phi_s} i_{\text{app}} \quad (10c)$$

$$0 = A_{\phi_e} \boldsymbol{\phi}_e + B_{\phi_e} \mathbf{j}_n + D_{\phi_e} \ln(\mathbf{c}_e), \quad (10d)$$

where the bold faced characters refer to their respective vector variables, which are defined as

$$\begin{aligned} \mathbf{c}_s(t) &= [c_s(x_1, r_1, t) \dots c_s(x_1, r_{n_r, n}, t) \dots c_s(x_{n_n+n_p}, r_{n_r, p}, t)]^T, \\ \mathbf{c}_e(t) &= [c_e(x_1, t) \dots c_e(x_{n_n+n_s+n_p}, t)]^T, \\ \boldsymbol{\phi}_s(t) &= [\phi_s(x_1, t) \dots \phi_s(x_{n_n}, t) \dots \phi_s(x_{n_n+n_p}, t)]^T, \end{aligned} \quad (11)$$

where  $x_i$  and  $r_i$  are the grid points of the discretization, and  $\boldsymbol{\phi}_e, \mathbf{j}_n$  are defined similarly to  $\mathbf{c}_e$  and  $\boldsymbol{\phi}_s$ , respectively. In (11),  $n_n, n_s, n_p$  are the number of elements of the FVM discretization, in the negative electrode, separator, and positive electrode, respectively. Furthermore,  $n_{r,n}, n_{r,p}$  are the number of elements of the FDM discretization, in the negative electrode, and the positive electrode, respectively. How to construct matrices  $A_i, B_i, i \in \{c_s, c_e, \phi_s, \phi_e\}$ ,  $C_{\phi_s}$ , and  $D_{\phi_e}$  is explained



in detail in [13]. The four sets of DAEs (10) are coupled by the Butler–Volmer rate equation, written as

$$\mathbf{j}_n = \text{diag} \left( \frac{\mathbf{i}_0(\bar{\mathbf{c}}_e, \bar{\mathbf{c}}_s)}{F} \right) \left( \exp \left( \frac{\alpha_a F}{RT} (\phi_s - \bar{\phi}_e - \mathbf{U}(\bar{\mathbf{c}}_s)) \right) - \exp \left( -\frac{\alpha_c F}{RT} (\phi_s - \bar{\phi}_e - \mathbf{U}(\bar{\mathbf{c}}_s)) \right) \right), \quad (12)$$

in which the  $\text{diag}(v)$  denotes a diagonal matrix with the elements of vector  $v$  on the main diagonal. Furthermore, in (12), the barred boldfaced variables  $\bar{\mathbf{c}}_s$ ,  $\bar{\mathbf{c}}_e$ , and  $\bar{\phi}_e$  refer to selected version of their non-barred boldfaced counterparts, where  $\bar{\mathbf{c}}_s$  denotes the vector of solid-phase surface concentrations, and  $\bar{\mathbf{c}}_e$  and  $\bar{\phi}_e$  denote the parts of  $\mathbf{c}_e$  and  $\phi_e$  given in the electrodes, respectively. Mathematically,  $\bar{\mathbf{c}}_s$ ,  $\bar{\mathbf{c}}_e$ ,  $\bar{\phi}_e$  and  $\mathbf{c}_s$ ,  $\mathbf{c}_e$ ,  $\phi_e$  are related, respectively, as follows

$$\bar{\mathbf{c}}_s = \bar{A}_{c_s} \mathbf{c}_s, \quad \bar{\mathbf{c}}_e = \bar{A}_{c_e} \mathbf{c}_e, \quad \bar{\phi}_e = \bar{A}_{\phi_e} \phi_e, \quad (13)$$

where

$$\bar{A}_{c_s} = \text{diag} \left( I_{n_n} \otimes [\mathbf{0}_{1 \times n_{r,n}-1}, 1], I_{n_p} \otimes [\mathbf{0}_{1 \times n_{r,p}-1}, 1] \right) \\ \bar{A}_{c_e} = \bar{A}_{\phi_e} = \begin{bmatrix} I_{n_n} & 0 & 0 \\ 0 & 0 & I_{n_p} \end{bmatrix},$$

in which  $\otimes$  is the Kronecker product. The DAEs (10a) and (10b) can further be discretized in time with sampling time  $\delta_t$  using a backward Euler scheme, to arrive at the following set of AEs

$$0 = \hat{A}_{c_s} \mathbf{c}_s(t_k) + \hat{B}_{c_s} \mathbf{j}_n(t_k) + \mathbf{c}_s(t_{k-1}) \quad (14a)$$

$$0 = \hat{A}_{c_e} \mathbf{c}_e(t_k) + \hat{B}_{c_e} \mathbf{j}_n(t_k) + \mathbf{c}_e(t_{k-1}) \quad (14b)$$

$$0 = A_{\phi_s} \phi_s(t_k) + B_{\phi_s} \mathbf{j}_n(t_k) + C_{\phi_s} i_{\text{app}}(t_k) \quad (14c)$$

$$0 = A_{\phi_e} \phi_e(t_k) + B_{\phi_e} \mathbf{j}_n(t_k) + D_{\phi_e} \ln(\mathbf{c}_e(t_k)) \quad (14d)$$

where  $t_k = k\delta_t$  for  $k \in \{1, \dots, N\}$ , in which  $N$  is the number of simulation steps. Furthermore,  $\hat{A}_{c_s} = \delta_t A_{c_s} - \mathbf{I}_{n_n n_{r,n} + n_p n_{r,p}}$ ,  $\hat{B}_{c_s} = \delta_t B_{c_s}$ ,  $\hat{A}_{c_e} = \delta_t A_{c_e} - \mathbf{I}_{n_n + n_s + n_p}$ , and  $\hat{B}_{c_e} = \delta_t B_{c_e}$ .

### 3.2. Solution method

The set of nonlinear AEs (14) obtained after discretization can be solved using any root-finding algorithm, such as Newton's method. However, due to the relatively large number of state variables, the use of such algorithms can be computationally slow. Namely, a large part of the computational effort is in the computation of the inverse of the Jacobian of the AEs (14). Therefore, in [13] a method was proposed in which Newton's method was applied sequentially to each set of equations of (14), thereby reducing the computation of the inverse of a single large Jacobian to the computation of the inverses of four smaller Jacobians. However, in doing so, some information of the large Jacobian is lost, which means that a quadratic convergence rate can no longer be achieved. Still, the computation time of the large Jacobian can be sufficiently large, such that the method proposed in [13] is still faster than solving (14) directly using Newton's method.

Rather than sequentially solving (14), we propose a solution method, which retains the full information of the Jacobian of (14), but reduces the number of AEs. This can be done by substitution of equations, such that one state variable remains. In doing so, the full information of the Jacobian of (14) is contained in a smaller Jacobian related to the equations of the remaining variable. The goal is to express  $\mathbf{j}_n$ ,  $\mathbf{c}_s$ ,  $\mathbf{c}_e$ , and  $\phi_e$  as a function of  $\phi_s$ , such that (12) is a non-linear equation that only depends on  $\phi_s$ . Hence, the set of AEs (14) will be reduced to one set of AEs in  $\phi_s$  that can be solved using Newton's method, from which  $\mathbf{c}_s$ ,  $\mathbf{c}_e$ ,  $\phi_e$  can be obtained.

The derivation of the reduced set of AEs is as follows. First, by solving (14c) for  $\mathbf{j}_n$ ,  $\mathbf{j}_n$  can be expressed as a function of  $\phi_s$ , i.e.,

$$\mathbf{j}_n(t_k) = -B_{\phi_s}^{-1} \left( A_{\phi_s} \phi_s(t_k) + C_{\phi_s} i_{\text{app}}(t_k) \right) \quad (15)$$

This expression allows the state variables  $\bar{\mathbf{c}}_s$ ,  $\bar{\mathbf{c}}_e$  to be expressed as a function of  $\phi_s$  by first substituting  $\mathbf{j}_n$  in (15) into their respective associated equations given in (14), i.e.,

$$\hat{A}_{c_i} \mathbf{c}_i(t_k) - \hat{B}_{c_i} B_{\phi_s}^{-1} \left( A_{\phi_s} \phi_s(t_k) + C_{\phi_s} i_{\text{app}}(t_k) \right) + \mathbf{c}_i(t_{k-1}) = 0,$$

for  $i \in \{s, e\}$ , then solving this for their respective state variables and pre-multiplying by  $A_{c_s}$ ,  $A_{c_e}$ , resulting in

$$\bar{\mathbf{c}}_i(t_k) = \Gamma_{c_i} i_{\text{app}}(t_k) + \Phi_{c_i} \phi_s(t_k) + \Theta_{c_i}, \quad (16a)$$

where

$$\Gamma_{c_i} = \bar{A}_{c_i} \hat{A}_{c_i}^{-1} \hat{B}_{c_i} B_{\phi_s}^{-1} C_{\phi_s},$$

$$\Phi_{c_i} = \bar{A}_{c_i} \hat{A}_{c_i}^{-1} \hat{B}_{c_i} B_{\phi_s}^{-1} A_{\phi_s},$$

$$\Theta_{c_i} = -\bar{A}_{c_i} \hat{A}_{c_i}^{-1} \mathbf{c}_i(t_{k-1}),$$

for  $i \in \{s, e\}$ . Note the presence of the full state vectors  $\mathbf{c}_s$  and  $\mathbf{c}_e$ , which can be obtained by (16a), except without the pre-multiplication by  $\bar{A}_{c_s}$  and  $\bar{A}_{c_e}$ , respectively. Similarly,  $\bar{\phi}_e$  can be expressed as a function of  $\phi_s$  and  $\mathbf{c}_e$  by substituting (15) into (14d), i.e.,

$$A_{\phi_e} \phi_e(t_k) - B_{\phi_e} B_{\phi_s}^{-1} \left( A_{\phi_s} \phi_s(t_k) + C_{\phi_s} i_{\text{app}}(t_k) \right) + D_{\phi_e} \ln(\mathbf{c}_e(t_k)) = 0, \quad (16b)$$

then solving (14d) for  $\phi_e$  and pre-multiplying by  $\bar{A}_{\phi_e}$ , giving

$$\bar{\phi}_e(t_k) = \Gamma_{\phi_e} i_{\text{app}}(t_k) + \Phi_{\phi_e} \phi_s(t_k) + \Theta_{\phi_e} \ln(\mathbf{c}_e(t_k)) \quad (16c)$$

in which

$$\Gamma_{\phi_e} = \bar{A}_{\phi_e} A_{\phi_e}^{-1} B_{\phi_e} B_{\phi_s}^{-1} C_{\phi_s}$$

$$\Phi_{\phi_e} = \bar{A}_{\phi_e} A_{\phi_e}^{-1} B_{\phi_e} B_{\phi_s}^{-1} A_{\phi_s}$$

$$\Theta_{\phi_e} = -\bar{A}_{\phi_e} A_{\phi_e}^{-1} D_{\phi_e}$$

The above steps allow the number of AEs given in (14) to be reduced, by substituting (15), (16a), and (16c) into (12), leading to an expression of the form  $F(\phi_s(t_k)) = 0$ , which can be solved using Newton's method, i.e.,

$$\phi_s^{m+1}(t_k) = \phi_s^m(t_k) - \mathcal{J}(\phi_s^m(t_k))^{-1} F(\phi_s^m(t_k)), \quad (17)$$

where  $m \in \{1, \dots, M\}$ , in which  $M$  is the maximum number of iterations, represents the current iteration in Newton's method, and  $\mathcal{J}$  is the Jacobian of  $F$ . Note that the Jacobian of  $F$  has  $(n_n + n_p)$  rows and columns, which is considerably smaller than the Jacobian of (14), which would have  $(3 + n_{r,p} + n_{r,n})(n_n + n_p) + 2n_s$  rows and columns.

Since  $A_{c_s}$ ,  $A_{c_e}$ ,  $A_{\phi_e}$ ,  $D_{\phi_e}$  are actually concentration-dependent, these matrices, and the matrices derived from these matrices, need to be updated at every iteration  $m$ . As we will show in the next section, the computation of these matrices will be the largest bottleneck in the algorithm. Therefore, especially Simplifications [S2-I] and [S2-II] lead to a large decrease in computation time. When applying [S2-II], the matrices  $A_{c_s}$ ,  $A_{c_e}$ ,  $A_{\phi_e}$ ,  $D_{\phi_e}$  become constant, which means that these matrices only need to be computed once. When applying [S2-I], the matrices need to be updated every time step, if the evaluation points  $c_e^*$  and  $\theta^*$  are chosen as  $c_e(t_{k-1})$  and  $\theta(t_{k-1})$ , respectively. However, even after applying Simplification [S2-I] or [S2-II],  $\Theta_i$ ,  $i \in \{c_s, c_e\}$  still change at every time step, and therefore these matrices have to be updated at every time step in all cases. This summarizes the implementation of the DFN model.

### 3.3. Toolbox for fast battery simulation (TOOFAB)

The DFN model with the implementation described in the previous subsection has been coded in MATLAB and developed into a toolbox. While there are already several battery simulation toolboxes (freely) available that solve the DFN model equations, e.g. [29,33,43], our

**Table 3**  
Arguments of the DFN function.

Argument	Type	Comment
out	struct	Contains all the output variables, such as the output voltage, the concentrations and the potentials.
input_current	Scalar/array/function handle	Contains information about the current profile. This field can be provided either as a scalar representing the desired applied current from time 0 to final_time, an array which contains the current levels at each specified sample time, or as a function which takes the output voltage, current, concentration and potentials, and the parameters as input and mainly provides the current as output. The latter form is especially useful when the battery is desired to be controlled in closed-loop. Example functions for input_current are provided with the toolbox.
final_time	Scalar	Specifies the final simulation time.
init_cond	Scalar/struct	Specifies the initial condition, which can be either an initial state-of-charge, as a value between 0 and 1, an initial voltage, or a MATLAB struct where the initial condition for a non-steady-state $c_s$ , $c_e$ , and $T$ can be specified. Further details on how init_cond can be specified can be found in the documentation of the toolbox.
param	struct	Can be used to change user-configurable parameters, such as all the model parameters, and simulation parameters, e.g., the temporal and spatial grid discretization variables. Note that this field is optional, and a default set of parameters is already contained in the DFN function. Parameter files containing the parameters of [29] and [30] are also included with the toolbox.

developed toolbox provides several advantages compared to the currently available toolboxes. Firstly, with the proposed solution method described in this section, the computation times are generally significantly smaller than the currently available toolboxes, as we will show in Section 4. Secondly, because the model equations given by (10) and (12) have been solved directly, without the use of any other software or MATLAB toolboxes, the developed battery simulation toolbox does not require installing any other toolboxes. This is an advantage over the battery simulation toolbox presented in [29], which, to the authors' knowledge, is the only other published toolbox that solves the DFN model equations implemented in MATLAB, where SUNDIALS [44] and CasADi [45] are used to solve the model equations. Finally, for the same reason as given in the previous point, i.e., that the model equations are solved directly without any other toolboxes, the MATLAB code is fairly easily translatable into C-code, which can either be made into an executable for even faster simulation, or can be used to implement the model on an embedded system, which are generally programmed in C. Note that while we have not considered a thermal model in this paper, a lumped thermal model, as described in [46], has been implemented in the battery simulation toolbox. The TOOLbox for FAST Battery simulation (TOOFAB) is freely available for download at: <https://github.com/Zuan-Khalik/TOOFAB>.

TOOFAB can be interfaced with the DFN function defined as `out = DFN(input_current, final_time, init_cond, param)`, where the definition of the input and output arguments can be found in Table 3. The scripts required to reproduce the results shown in the simulation study section below are included with the toolbox. Using the configurable parameters, the toolbox allows the user to easily apply any of Simplifications [S1]–[S3], as well as varying the coarseness of the spatial and temporal discretization. This allows in making a desired trade-off between model accuracy and computation time. Several example scripts are provided with the toolbox to show how the various functions and features can be used. A more detailed user guide is also supplemented with the toolbox. In the next section, amongst other results, we will show the impact of Simplifications [S1]–[S3] to demonstrate how this trade-off can be made effectively.

#### 4. Simulation study

In this section, we will study the computational performance and accuracy of the CDFN model described in Section 2.1, and its simplified versions. Specifically, we will first study the impact of Simplifications [S1]–[S3] made in Section 2.3 on the accuracy and computational speed compared to the CDFN model. Then, the impact of discretization

of the DFN model on model accuracy and computation time is studied. This is followed by a comparison of the numerical methods presented in [29] and [13] with the numerical method described in Section 3. This comparison will also serve as a validation of the model implementation with the implementation of [29], which in turn has been validated against a DFN model implemented in COMSOL [43]. Finally, based on the observations of the impact of simplifications and discretization on accuracy and computation speed, we will define several simplified DFN (SDFN) models and SPMs, and compare these with the CDFN model in terms of model accuracy and computational performance.

The simulation results have been obtained using MATLAB R2020b on a typical desktop PC. Furthermore, unless otherwise stated, the computation times shown will be only the time it takes to compute the solutions of (17), which includes the computation of  $F$  and  $J$  in (17). The simulation studies will be done using two different sets of parameters, obtained from [30] and [29]. Interestingly, the parameter set from [30] has been parameterized with a high-power (HP) cell, while the parameter set from [29] has been parameterized with a high-energy (HE) cell. Since the original parameter sets from [30] and [29] do not consider all concentration-dependent parameters, i.e.  $\kappa$ ,  $D_e$ ,  $\nu$ ,  $D_{s,p}$ , we have extended the respective parameter sets with the missing concentration-dependent parameters. The functions used for the concentration-dependent parameters are summarized in Table 4. The time-step size has been chosen as  $\delta_t = 1$  s in all the simulation studies, and, the tolerance to terminate the Newton's method has been

**Table 4**  
Functions for the concentration-dependent parameters used in the high-power (HP) and high-energy (HE) parameter sets.

HP	
$\kappa^a$	$15.8 \times 10^{-4} c_e \exp(-0.85(c_e/1000)^{1.4})$
$D_e^b$	$0.134 \times 10^{-8.43-54/(T-229-5(c_e/1000))-0.22(c_e/1000)}$
$\nu^b$	$-(0.6-0.24(\frac{c_e}{1000})^{0.5}+0.98(1-0.0052(T-294))(\frac{c_e}{1000})^{1.5})$
$D_{s,p}^d$	$55782 \times 10^{-20.26+534.9(\theta-0.5)^8+2.263(\theta-0.5)^2}$
HE	
$\kappa^c$	$0.0413+0.5\frac{c_e}{1000}-0.47(\frac{c_e}{1000})^2+0.15(\frac{c_e}{1000})^3-0.016(\frac{c_e}{1000})^4$
$D_e^b$	$0.000233 \times 10^{-8.43-54/(T-229-5(c_e/1000))-0.22(c_e/1000)}$
$\nu^b$	$-(0.6-0.24(\frac{c_e}{1000})^{0.5}+0.98(1-0.0052(T-294))(\frac{c_e}{1000})^{1.5})$
$D_{s,p}^d$	$1315383 \times 10^{-20.26+534.9(\theta-0.5)^8+2.263(\theta-0.5)^2}$

<sup>a</sup>Obtained from [37].

<sup>b</sup>Adapted from [21].

<sup>c</sup>Obtained from [29].

<sup>d</sup>Approximated and adapted from [24], see Fig. 2.

**Table 5**

The effect of Simplification [S1]–[S3] introduced in Section 2 on accuracy and simulation time for the high-power (HP) [30] and high-energy (HE) [29] parameter sets.

NRMSE [ $10^{-3}$ ]	$V$		$\phi_e$		$c_e$		$\bar{c}_s$		$j_n$		Sim. time [s]	
	HP	HE	HP	HE	HP	HE	HP	HE	HP	HE	HP	HE
Full model	–	–	–	–	–	–	–	–	–	–	5.60	9.88
[S1]	0.0019	4.0	0.019	2.9	0.0094	1.1	0.0022	0.73	0.17	3.1	5.52	6.41
[S2-I]-all	0.059	0.041	0.036	0.044	0.017	0.069	0.026	0.014	0.24	0.042	3.19	3.78
[S2-II]- $\kappa$	0.0056	3.8	0.30	6.1	0.020	3.1	0.017	3.3	0.075	1.2	5.04	8.55
[S2-II]- $D_e$	0.0068	1.0	0.33	1.5	0.58	12	0.020	0.94	0.064	0.36	4.84	8.35
[S2-II]- $\nu$	0.0093	1.1	0.46	1.7	0.027	0.72	0.027	1.1	0.19	0.40	5.62	9.71
[S2-II]- $D_{s,p}$	16	7.6	0.27	4.6	0.14	2.3	6.9	1.9	0.74	3.8	2.41	4.54
[S2-II]-all	17	9.0	1.2	10	0.62	10	6.9	5.3	0.74	4.2	0.59	1.29
[S3]	–	1.7	–	1.3	–	0.36	–	1.1	–	0.94	–	4.98

chosen, unless otherwise stated, as,  $2 \times 10^{-3}$  and  $10^{-2}$ , for the HP cell and HE cell, respectively, as these values were found to strike a good trade-off between accuracy and computation time. Furthermore, all the reported computation times in this section have been obtained from an average of 10 repetitions. Some striking differences between the two parameter sets will be observed below.

#### 4.1. Impact of model simplifications

In order to investigate the impact of Simplifications [S1]–[S3] described in Section 2.3 on model accuracy and computational speed, the CDFN model has been compared to a DFN model with varying simplifications for two parameter sets. The current profile used to produce the simulation results can be seen in Fig. 3 and the grid parameters have been chosen as  $n_n = n_s = n_p = n_{r,n} = n_{r,p} = 10$ . To express the difference in accuracy of the various simplifications, we will use a normalized root-mean-square error (NRMSE), defined as

$$\text{NRMSE}(p, q) = \frac{\sqrt{\frac{1}{N} \sum_{k=1}^N (p_k - q_k)^2}}{\max_k \left\{ \frac{1}{2}(p_k + q_k) \right\} - \min_k \left\{ \frac{1}{2}(p_k + q_k) \right\}}, \quad (18)$$

for some vectors  $p$  and  $q$ . For the comparison of the output voltage between different models,  $p$  and  $q$  are of length  $N$ , while for the comparison of internal states,  $p$  and  $q$  are stacked vectors of the vectors as defined in (11) over the sample times.

In Table 5 the NRMSE of various quantities and simulation times of the full and simplified models are shown for the HP and HE parameter sets. We can first observe that the errors made due to applying Simplification [S2-I] are negligibly small for both parameter sets. We can further observe that, generally, the errors made due to simplifications are smaller with the HP parameter set than the HE parameter set. This

can be explained by the fact that the current density levels in the HP cell are much smaller than in the HE cell, which in turn leads to smaller over-potentials, and therefore smaller concentration gradients, particularly in the electrolyte. Simplification [S1], therefore leads to very small errors in the HP case, while in the HE case the errors are significantly larger. This can also be seen in Fig. 3, where we can see that at some points the output voltage deviates somewhat from the CDFN model. It should be noted, however, that throughout most of the simulation, the over-potentials do stay relatively small, which means that overall, Simplification [S1] can still be considered reasonable, even for the HE cell. Even when (small) voltage differences do occur, e.g. at 3320 s, the differences in the internal states are still small.

The conductivity  $\kappa$ , diffusivity  $D_e$  and activity coefficient  $\nu$  do not deviate significantly from their nominal values in the HP cell, making the simplification that they are constant (Simplification [S2-II]) quite reasonable. This small deviation can be explained by the fact that the electrolyte concentration gradients are very small under the considered applied current profile. However, as the diffusion coefficient  $D_s$  in the electrodes is two orders of magnitude smaller in the HP cell than in the HE cell, even with the smaller over-potentials, there is still a large concentration gradient within the radius of the particles. This in turn makes the simplification for a constant  $D_s$  less reasonable in the HP case. Finally, we see that Simplification [S3] is reasonable for the HE cell, while with the HP cell, the errors made were so large, that the simulation would not converge. This can again be explained by the fact that diffusion gradients in the radius of the electrodes in the HP cell is much larger than the HE cell. This means that the diffusion dynamics in the HP cell becomes more important, while in the HE cell, the two-parameter approximation is sufficient. These results show that making parameters concentration-dependent does not always lead to a significant change in the model, since even the relatively (to the other

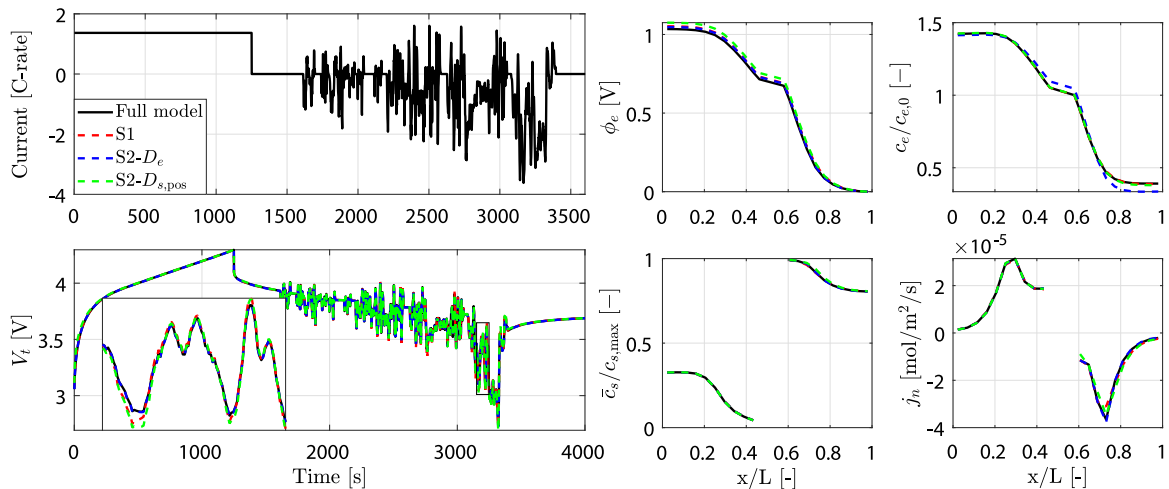


Fig. 3. Visualization of the errors made using Simplifications S1 and S2-II with the high-energy parameter set [29].

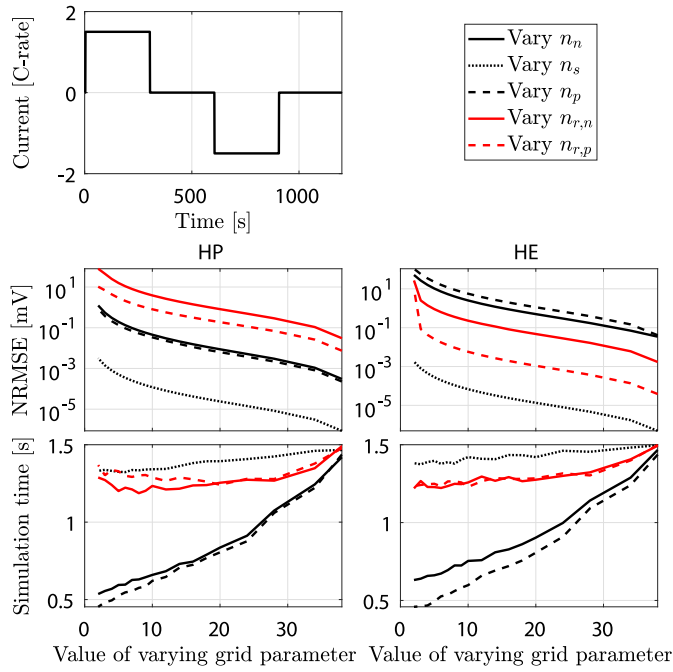


Fig. 4. NRMSE of the output voltage and computation time for varying grid parameters for the high-power (HP) [30] and the high-energy (HE) [29] parameter set.

simplifications) large errors observed can still considered to be small as far as usual modeling errors go.

When looking at the simulation times in Table 5, we can observe that in both the HE cell and HP cell, simplifications on parameter  $D_{s,p}$  lead to the largest reduction in simulation time. The reason for this is that  $D_{s,p}$  affects the matrix  $A_{cs}$ , which is generally the largest matrix, and takes substantially more computational effort to compute than the other matrices. Note, however, that considering the simplifications individually, does not paint the full picture. When applying [S2-II] on all the considered concentration-dependent parameters, an implementation can be obtained that is much more computationally efficient than the full model. This is because with the proposed implementation presented in Section 3, computation of the matrices  $A_{cs}$ ,  $A_{\phi_e}$ ,  $D_{\phi_e}$ ,  $A_{ce}$  becomes the bottleneck in the computation of the solution. If these matrices can be pre-computed, as is the case with Simplification [S2-II], then the computation of the solution can be made much faster. However, we also observe that there is some sacrifice in accuracy.

#### 4.2. Impact of discretization

In the discretization approach described in Section 3, there is freedom in selecting the number of volume elements  $n_n, n_s, n_p$  of the FVM discretization and the number of elements  $n_{r,n}, n_{r,p}$  of the FDM discretization. Choosing these parameters large enough leads to a better approximation of the diffusion dynamics in the battery. However, a finer discretization also leads to an increased computation time. To analyze this trade-off, a study has been done where the DFN model has been simulated with various sets of grid parameters  $n_n, n_s, n_p, n_{r,n}, n_{r,p}$ . Specifically, we will take a DFN model with Simplifications [S1] and [S2-II] applied. As a first case, a certain set of base grid parameters  $n_n, n_s, n_p, n_{r,n}, n_{r,p} = 40$  has been chosen as a baseline. Then, each grid parameter has been varied between 1 and 40, for which the NRMSE of the output voltage (with respect to the simulation with the base grid parameters) and simulation time have been computed.

In Fig. 4, the results of this first case study are shown. We see that, generally, in both parameter sets, as the value of the grid parameter increases, the NRMSE decreases, while computation time increases.

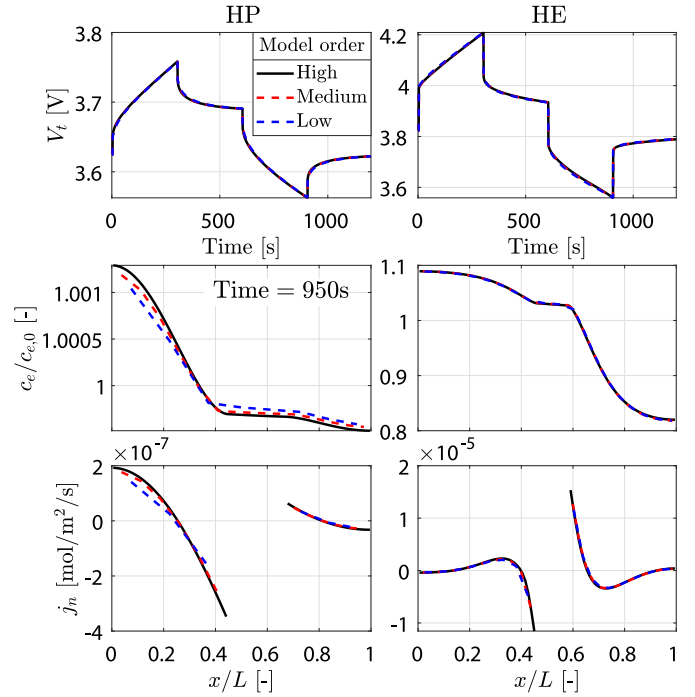


Fig. 5. Output voltage and various internal variables for varying degrees of model order corresponding to Table 6 for the high-power (HP) [30] and the high-energy (HE) [29] parameter set.

However, some grid parameters seem to have a larger impact on the RMS error and computation time than others. With the HP parameter set, the grid parameters that relate to the solid-phase diffusion  $n_{r,n}$  and  $n_{r,p}$  have the largest impact on the NRMSE, while the impact of  $n_n, n_s, n_p$  is at least an order of magnitude smaller than that of  $n_{r,n}$  and  $n_{r,p}$ . This indicates that diffusion dynamics in the solid phase largely dominates diffusion dynamics in the electrolyte phase, as, apparently, diffusion along the length of the cell does not need a high discretization. Thus, for the HP parameter set, to retain model accuracy and limit computation time,  $n_{r,n}$  and  $n_{r,p}$  need to be relatively large, while  $n_n, n_s, n_p$  can be small. For the HE parameter set, the grid parameters related to the electrodes have the most impact on the NRMSE. Here, to retain model accuracy and computation time,  $n_n$  and  $n_p$  need to be chosen relatively large. Furthermore, we can see that the effect of the grid parameters on the computation time is similar for both parameter sets, which is to be expected.

In Fig. 5, the output voltage and normalized electrolyte concentration are shown, where the considered DFN model has been simulated with varying degrees of model order for the two parameter sets. The selected model orders are based on the analysis above, to show the trade-off that can be made between model accuracy and computation time. The specific grid parameters of each model order, together with their respective computation time and NRMSE, are shown in Table 6.

Table 6

Selected grid parameters of the varying degrees of model orders shown in Fig. 5.

	Model order	Grid param. <sup>a</sup>	NRMSE [ $10^{-3}$ ]	Comp. time [s]
HP [30]	Low	[3, 2, 3, 9, 9]	5.3	0.12
	Medium	[5, 5, 5, 16, 14]	2.1	0.14
	High	[10, 5, 10, 20, 20]	0.97	0.20
HE [29]	Low	[9, 2, 12, 3, 3]	5.7	0.19
	Medium	[12, 8, 15, 3, 3]	3.0	0.23
	High	[22, 10, 23, 3, 7]	1.0	0.44

<sup>a</sup>The order of the grid parameters is  $[n_n, n_s, n_p, n_{r,n}, n_{r,p}]$ .



The reported NRMSEs have been computed in relation to the model with the base set of grid parameters. The low, medium, and high order models have been chosen such that the NRMSE is at most  $6 \times 10^{-3}$ ,  $3 \times 10^{-3}$ , and  $10^{-3}$ , respectively, and that there is a somewhat reasonable, good, and very good agreement of the internal states with the baseline model, respectively. We observe that output voltage can be modeled well with a relatively coarse discretization over the cell length, especially in the HP case. However, in order to model the internal states well, both in the HP and HE case,  $n_s$ ,  $n_p$  are important, as they relate to diffusion across the cell.

#### 4.3. Comparison of numerical methods

In order to validate the model implementation presented in Section 3, the proposed numerical method of this paper will be compared to the numerical methods proposed in [29] and [13]. The numerical method of [29] in turn has been validated against a DFN model validated in COMSOL [43]. To ensure that the model equations are the same in all cases, the parameter set used in this paper has been adapted exactly to that used in [29] under isothermal conditions. This amounts to choosing  $\nu = 0$ , and applying [S2-II] on the parameters  $D_e$  and  $D_{s,p}$ . Thus,  $\kappa$  is the only remaining concentration-dependent parameter. The results for the numerical method of [29] have been obtained using the LIONSIMBA toolbox presented in the aforementioned paper, while the numerical method of [13] has been implemented according to the description of the method, with the same discretization as presented in this paper, in order to obtain the respective results. Since chosen tolerances to terminate the algorithms can have a significant effect on computation times, to ensure a fair comparison, the tolerances for the methods have been chosen such that for each method, the NRMSEs of the considered state variables in Table 5 between the model with a largest possible chosen tolerance and a model with a tolerance of  $10^{-6}$  do not exceed  $10^{-4}$ . From this requirement, the tolerances in LIONSIMBA [29] have been set to  $2 \times 10^{-5}$ , the tolerance in the method of [13] has been set to 0.009, and the tolerance of the proposed method has been set to 0.03.

A simulation of the DFN model using the various numerical methods with the current profile that can be seen in Fig. 3 is shown in Fig. 6. We observe that both in the output voltage and internal variables, the results are visually identical. The computation times and NRMSEs for the results shown in Fig. 6 can be seen in Table 7. The total

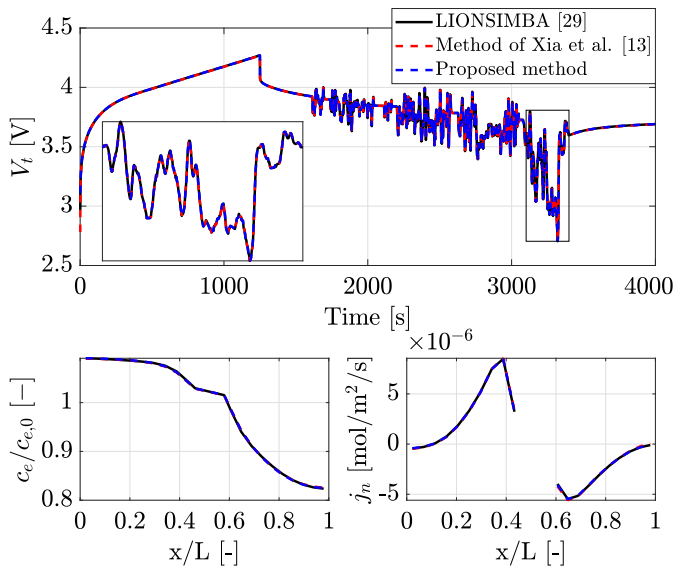


Fig. 6. Comparison of output voltage and several internal variables between various model implementations.

Table 7

Computation times and NRMSEs of various numerical methods for the results shown in Fig. 6.

	Computation time [s]		NRMSE [ $10^{-3}$ ]	
	Total	Solutions	$V_t$	Internal states <sup>a</sup>
LIONSIMBA [29]	1106	137.0	–	–
Method of Xia et al. [13]	129.3	129.2	1.6	0.46
Proposed method	2.78	2.71	1.6	0.46

<sup>a</sup>Taken as the mean value of the NRMSE values of  $\phi_e$ ,  $c_e$ ,  $\bar{c}_s$ , and  $\eta$ .

computation time represents the total time taken to simulate the model, while the computation time of the solutions only accounts for the actual computations required to compute the solution itself, without overhead in computing additional variables or pre-computation of matrices. From Table 7, we can first observe that in accordance to our earlier observation of Fig. 6, the NRMSE values are sufficiently small to consider the numerical methods to be equivalent. The computation times, however, vary significantly between the different numerical methods, in particular when the total computation time is considered, where the computation time of the method proposed in this paper is almost 400 times smaller than the computation time of the method of [29]. However, comparing this to the computation time of the solutions, we observe that, apparently, there is a large amount of overhead in the LIONSIMBA toolbox. Even so, the method proposed in this paper is over 50 times faster than the method of [29] and over 45 times faster than the method of [13]. It should be noted that the method of [29] uses a variable time-step solver, which becomes especially slow when dynamic current profiles are applied, while in the method of this paper the time-step size is constant (set to 1 s). However, we report that with constant currents, i.e., the first 1600 s of the applied current profile shown in Fig. 3, it takes 1.4 s and 0.63 s to compute the solution with the method of [29] and the method proposed in this paper, respectively. Thus, even with constant current profiles, the method proposed in this paper is still faster than the method of [29]. We should further note that as a consequence of having a fixed-time-step solver, discretization errors can be larger than when using a variable-time-step solver, particularly when simulating over a very large time frame. However, as can also be observed from the results in Table 7, in the considered simulations, the discretization errors were not significantly large.

#### 4.4. Comparison of models

With the numerical method proposed in this paper validated, and with the observations made in Sections 4.1 and 4.2, a selection of SDFN models and SPMs can be made and compared to the CDFN model to show how a general trade-off can be made between accuracy and computational speed with the numerical method proposed in this paper. Based on the analysis of the trade-off that can be made between model accuracy and computation time using Simplifications [S1]–[S3] and the choice of grid parameters, for each parameter set, high-fidelity (HIFI) and low-fidelity (LOFI) models have been defined. The choices for the simplifications and grid parameters for these models are shown in Table 8. Here we note that since Simplification [S2-I] had a negligible effect on the model accuracy, the CDFN model is considered with this simplification applied. We further note that [S3] is not additionally applied for the LOFI models in the HE case, since it does not lead to any decrease in computation time if used in combination with [S2]- $D_{s,p}$ . Similar to the SDFN models, HIFI and LOFI versions are considered for the SPM, with the same choice of simplifications and grid parameters. The so-called common DFN model is additionally included, to show how the considered models compare to the DFN models typically found in literature, where usually either  $\kappa$  and/or  $D_e$  are considered as concentration-dependent parameter, e.g. [29,30,43]. The selected grid

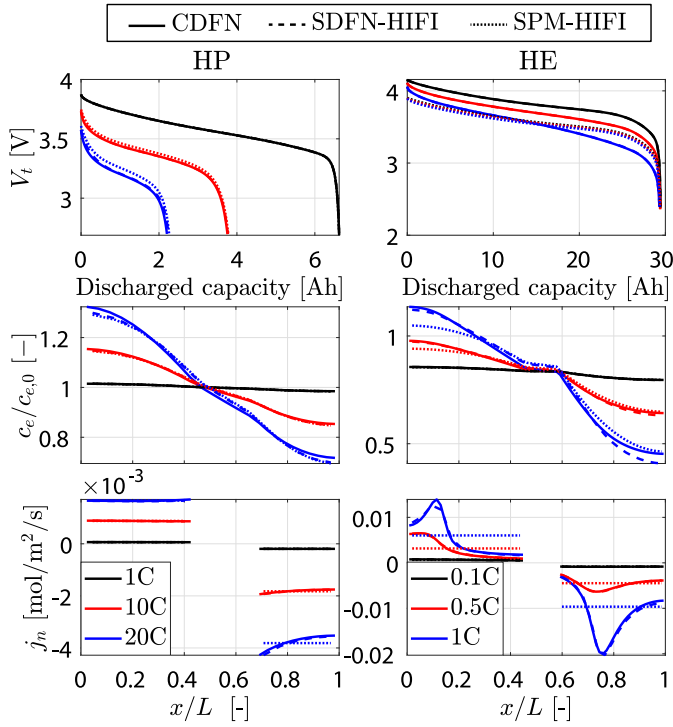


Fig. 7. Comparison of discharge curves of various models under several C-rates for the high-power (HP) [30] and high-energy (HE) [29] parameter sets. The internal variables are shown at the point corresponding to 80% of the discharged capacity.

parameters for the common DFN model are based on those used in the LIONSIMBA toolbox [29].

A first comparison of some of the models is made in Fig. 7, where the discharge curves for the CDFN model, SDFN-HIFI model and SPM-HIFI are shown under several C-rates for both of the parameter sets. With the HP cell, until 10C, the SPM models have a good agreement with the CDFN model, while at 20C the agreement becomes worse. The SDFN-HIFI model, on the other hand, has a good agreement at all C-rates, except for some observed difference in the normalized electrolyte concentration. This difference is mainly due to Simplification [S2-II]- $D_e$ , since, in the HP case, at very high C-rates the electrolyte concentration gradient becomes sufficiently large for the effect of the concentration-dependency to show. For the HE cell, we can observe that now the SPM fails to represent the output voltage at all C-rates,

Table 8

Choice of simplifications and grid parameters of the presented models.

		Simplifications <sup>a</sup>	Grid param. <sup>b</sup>
CDFN	HP	[S2-I]-all	[10, 5, 10, 20, 20]
	HE	[S2-I]-all	[22, 10, 23, 3, 7]
Common DFN	HP	[S2-II]- $v$ , $D_{s,p}$	[10, 10, 10, 10, 10]
	HE	[S2-II]- $v$ , $D_{s,p}$	[10, 10, 10, 10, 10]
HIFI models	HP	[S1],[S2-II]- $\kappa$ , $D_e$ , $v$	[5, 5, 5, 16, 14]
	HE	[S3],[S2-II]- $\kappa$ , $D_e$ , $v$	[12, 8, 15, 3, 3]
LOFI models	HP	[S1],[S2-II]-all	[3, 2, 3, 9, 9]
	HE	[S1],[S2-II]-all	[9, 2, 12, 3, 3]

<sup>a</sup>Unless stated otherwise, the concentration-dependent parameters have been simplified according to [S2-I].

<sup>b</sup>The order of the grid parameters is  $[n_s, n_p, n_{r,n}, n_{r,p}]$ .

while the effect on the normalized concentration is somewhat less pronounced. In terms of output voltage and ionic flux  $j_n$ , the SDFN-HIFI model has still a very good agreement with the CDFN model, while for similar reasons as in the HP case, some discrepancy can be observed in the electrolyte concentration.

In Fig. 8, the selected models have been simulated with the current profile shown in Fig. 3 for both parameter sets. Here we see again that the SPM agrees well with the CDFN model, especially so the SPM-HIFI model, since the solid-phase diffusion coefficient is still varying with solid-phase concentration. The Simplification [S2-II]- $D_{s,p}$ , therefore, seems to have the largest effect on the perceived differences between the LOFI models and the CDFN model. However, it should be noted that the errors are still small in this case. When observing the results of the HE cell, we see that the errors made with the SPM are, again, considerably large, and show entirely different dynamics than the CDFN model. The SDFN models, on the other hand, both show a good agreement with the CDFN model.

The computation times and NRMSE values for the results shown in Fig. 8 can be seen in Table 9. As expected, the computation times of the SPM are the lowest, since the SPM has the lowest complexity, although with the HE parameter set, the NRMSE is unacceptable. The SDFN models, on the other hand, have low errors in all cases for both parameter sets, while the computation times are not much higher than the SPM. We can also see how a trade-off can be made between accuracy and computation time, where, e.g., in the HE case, the computation time of the SDFN-HIFI model is approximately twice that of the SDFN-LOFI model, whereas the NRMSE is approximately twice as low. However, the NRMSE values of the SDFN-LOFI model can actually still be considered to be highly accurate, since the common

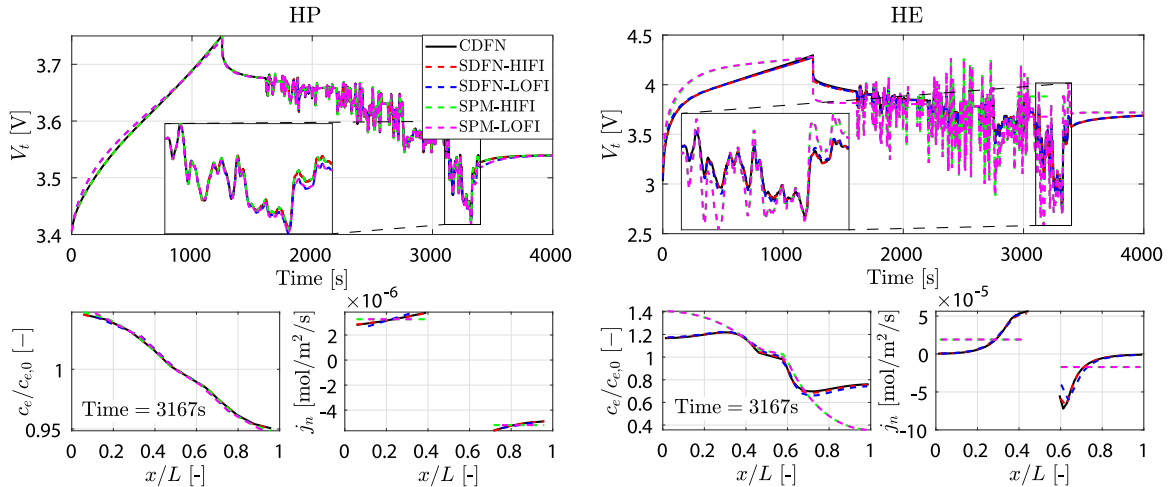


Fig. 8. Comparison of output voltage and internal states of various models for the high-power (HP) [30] and high-energy (HE) [29] parameter sets.

**Table 9**

Computation times and NRMSEs of the selected models for the results shown in Fig. 8.

	Computation time [s]		NRMSE $V_i$ [ $10^{-3}$ ]	
	HP [30]	HE [29]	HP [30]	HE [29]
CDFN	6.0	8.6	–	–
Common DFN	1.6	2.3	17	9.3
SDFN-HIFI	1.4	1.8	0.99	5.6
SDFN-LOFI	0.40	0.72	17	10
SPM-HIFI	0.30	0.30	0.65	89
SPM-LOFI	0.13	0.14	17	88

DFN model has similar error for both parameter sets. This suggests that most of the errors made originate from Simplification [S2-II]- $D_{s,p}$ , which, to the authors' knowledge, is a parameter that has been considered to be concentration-dependent only in [40]. Therefore, the usual considered concentration-dependent parameters, i.e.,  $\kappa$  and  $D_e$  can be assumed constant with practically no loss in model accuracy of the output. These results show that with the proposed implementation, and by selectively making the proposed simplifications, as well as selectively choosing the grid parameters, a model can be obtained that has a small impact on model accuracy, while the computation time can be drastically decreased, to achieve a simulation time of over 5000 times faster than real-time.

## 5. Conclusions

In this paper, we have studied the impact of several types of model simplifications on the trade-off between model accuracy and computation time for the DFN model. Furthermore, we have proposed a computationally efficient implementation of the CDFN model that has lead to a significant reduction in computation time. The proposed model implementation has been developed into a freely downloadable toolbox, which has also been presented in this paper. In the validation of the model simplifications, we have shown that linearizing the Butler–Volmer equation (5) (Simplification [S1]) and simplifying concentration-dependent parameters to constant parameters (Simplification [S2-II]) has a small effect on the model dynamics in most cases, while leading to a large decrease in computation time. Furthermore, we have studied the impact of the coarseness of the spatial discretization, where we have shown that the discretization grid parameters can be chosen differently depending on the modeled cell characteristics to make a good trade-off between model accuracy and computation time. To validate the proposed numerical method, a comparison has been made with the numerical methods of [29] and [13], where we have also shown that the numerical method proposed in this paper is substantially faster than the aforementioned numerical methods. Finally, we have compared several simplified DFN models to the SPM and the CDFN model. Here we have shown that with the proposed implementation, and by selectively making the proposed simplifications, as well as selectively choosing the grid parameters, a model can be obtained that has a small impact on model accuracy in terms of battery voltage and internal states, while the computation time can be drastically decreased, to achieve a simulation time of over 5000 times faster than real-time.

## CRedit authorship contribution statement

**Z. Khalik:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing - original draft. **M.C.F. Donkers:** Conceptualization, Methodology, Resources, Writing - review & editing, Supervision, Funding acquisition. **H.J. Bergveld:** conceptualization, Methodology, resources, Writing - review & editing, Supervision, Funding acquisition.

## Declaration of competing interest

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

## Acknowledgments

This work has received financial support from the Horizon 2020 programme of the European Union under the grants 'Electric Vehicle Enhanced Range, Lifetime And Safety Through INGenious battery management' (EVERLASTING-713771) and 'Advancing fail-aware, fail-safe, and fail-operational electronic components, systems, and architectures for fully automated driving to make future mobility safer, affordable, and end-user acceptable' (AutoDrive-737469).

## References

- [1] S.J. Moura, F.B. Argomedo, R. Klein, A. Mirtabatabaei, M. Krstic, Battery state estimation for a single particle model with electrolyte dynamics, *Trans. Control Syst. Technol.* 25 (2) (2017) 453–468, <http://dx.doi.org/10.1109/TCST.2016.2571663>.
- [2] C. Zou, A.G. Kallapur, C. Manzie, D. Nesic, PDE battery model simplification for SOC and SOH estimator design, in: *Conf. Decis. & Control*, 2015, pp. 1328–1333, <http://dx.doi.org/10.1109/CDC.2015.7402395>.
- [3] R. Klein, N.A. Chaturvedi, J. Christensen, J. Ahmed, R. Findeisen, A. Kojic, Optimal charging strategies in lithium-ion battery, in: *Am. Control Conf.*, 2011, pp. 382–387, <http://dx.doi.org/10.1109/ACC.2011.5991497>.
- [4] M. Torchio, N.A. Wolff, D.M. Raimondo, L. Magni, U. Kreuer, R.B. Gopaluni, J.A. Paulson, R.D. Braatz, Real-time model predictive control for the optimal charging of a lithium-ion battery, in: *Am. Control Conf.*, 2015, p. 4536, <http://dx.doi.org/10.1109/ACC.2015.7172043>.
- [5] B. Yann Liaw, G. Nagasubramanian, R.G. Jungst, D.H. Doughty, Modeling of lithium ion cells—A simple equivalent-circuit model approach, *Solid State Ion.* 175 (1) (2004) 835–839, <http://dx.doi.org/10.1016/j.ssi.2004.09.049>.
- [6] H.E. Perez, X. Hu, S. Dey, S.J. Moura, Optimal charging of Li-Ion batteries with coupled electro-thermal-aging dynamics, *Trans. Veh. Technol.* 66 (9) (2017) 7761–7770, <http://dx.doi.org/10.1109/TVT.2017.2676044>.
- [7] Shuo Pang, J. Farrell, Jie Du, M. Barth, Battery state-of-charge estimation, in: *Am. Control Conf.*, Vol. 2, 2001, pp. 1644–1649, <http://dx.doi.org/10.1109/ACC.2001.945964>.
- [8] M. Doyle, T.F. Fuller, J. Newman, Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell, *J. Electrochem. Soc.* 140 (6) (1993) 1526–1533, <http://dx.doi.org/10.1149/1.2221597>.
- [9] V.R. Subramanian, V.D. Diwakar, D. Tapriyal, Efficient macro-micro scale coupled modeling of batteries, *J. Electrochem. Soc.* 152 (10) (2005) A2002, <http://dx.doi.org/10.1149/1.2032427>.
- [10] T.-S. Dao, C.P. Vyasarayani, J. McPhee, Simplification and order reduction of lithium-ion battery model based on porous-electrode theory, *J. Power Sources* 198 (2012) 329–337, <http://dx.doi.org/10.1016/j.jpowsour.2011.09.034>.
- [11] M.A. Kehs, M.D. Beeney, H.K. Fathy, Computational efficiency of solving the DFN battery model using descriptor form with Legendre polynomials and Galerkin projections, in: *Am. Control Conf.*, 2014, pp. 260–267, <http://dx.doi.org/10.1109/ACC.2014.6858858>.
- [12] L. Cai, R.E. White, Reduction of model order based on proper orthogonal decomposition for Lithium-Ion battery simulations, *J. Electrochem. Soc.* 156 (3) (2009) A154, <http://dx.doi.org/10.1149/1.3049347>.
- [13] L. Xia, E. Najafi, Z. Li, H. Bergveld, M. Donkers, A computationally efficient implementation of a full and reduced-order electrochemistry-based model for Li-ion batteries, *Appl. Energy* 208 (2017) 1285–1296, <http://dx.doi.org/10.1016/j.apenergy.2017.09.025>.
- [14] G. Fan, K. Pan, M. Canova, A comparison of model order reduction techniques for electrochemical characterization of Lithium-ion batteries, in: *Conf. Decis. & Control*, 2015, pp. 3922–3931, <http://dx.doi.org/10.1109/CDC.2015.7402829>.
- [15] C. Zou, C. Manzie, D. Nesic, A framework for simplification of PDE-based Lithium-Ion battery models, *Trans. Control Syst. Technol.* 24 (5) (2016) 1594–1609, <http://dx.doi.org/10.1109/TCST.2015.2502899>.
- [16] X. Han, M. Ouyang, L. Lu, J. Li, Simplification of physics-based electrochemical model for lithium ion battery on electric vehicle. Part I: Diffusion simplification and single particle model, *J. Power Sources* 278 (2015) 802–813, <http://dx.doi.org/10.1016/j.jpowsour.2014.12.101>.
- [17] X. Han, M. Ouyang, L. Lu, J. Li, Simplification of physics-based electrochemical model for lithium ion battery on electric vehicle. Part II: Pseudo-two-dimensional model simplification and state of charge estimation, *J. Power Sources* 278 (2015) 814–825, <http://dx.doi.org/10.1016/j.jpowsour.2014.08.089>.

- [18] P. Kemper, S.E. Li, D. Kum, Simplification of pseudo two dimensional battery model using dynamic profile of lithium concentration, *J. Power Sources* 286 (2015) 510–525, <http://dx.doi.org/10.1016/j.jpowsour.2015.03.134>.
- [19] S. Khaleghi Rahimian, S. Rayman, R.E. White, Extension of physics-based single particle model for higher charge–discharge rates, *J. Power Sources* 224 (2013) 180–194, <http://dx.doi.org/10.1016/j.jpowsour.2012.09.084>.
- [20] X. Li, G. Fan, G. Rizzoni, M. Canova, C. Zhu, G. Wei, A simplified multi-particle model for lithium ion batteries via a predictor–corrector strategy and quasi-linearization, *Energy* 116 (2016) 154–169, <http://dx.doi.org/10.1016/j.energy.2016.09.099>.
- [21] L.O. Valen, J.N. Reimers, Transport properties of LiPF<sub>6</sub>-based Li-ion battery electrolytes, *J. Electrochem. Soc.* 152 (5) (2005) A882–A891, <http://dx.doi.org/10.1149/1.1872737>.
- [22] P. Georén, G. Lindbergh, Characterisation and modelling of the transport properties in lithium battery gel electrolytes: Part I. The binary electrolyte PC/LiClO<sub>4</sub>, *Electrochim. Acta* 49 (21) (2004) 3497–3505, <http://dx.doi.org/10.1016/j.electacta.2004.03.020>.
- [23] C. Delacourt, M. Ati, J.M. Tarascon, Measurement of lithium diffusion coefficient in Li<sub>y</sub> FeSO<sub>4</sub>f, *J. Electrochem. Soc.* 158 (6) (2011) A741–A749, <http://dx.doi.org/10.1149/1.3581087>.
- [24] M. Safari, C. Delacourt, Mathematical modeling of Lithium Iron Phosphate electrode: Galvanostatic charge/discharge and path dependence, *J. Electrochem. Soc.* 158 (2) (2011) A63–A73, <http://dx.doi.org/10.1149/1.3515902>.
- [25] M.M. Doeff, L. Edman, S.E. Sloop, J. Kerr, L.C. De Jonghe, Transport properties of binary salt polymer electrolytes, *J. Power Sources* 89 (2) (2000) 227–231, [http://dx.doi.org/10.1016/S0378-7753\(00\)00433](http://dx.doi.org/10.1016/S0378-7753(00)00433).
- [26] Z. Khalik, H.J. Bergveld, M.C.F. Donkers, On trade-offs between computational complexity and accuracy of electrochemistry-based battery models, in: *Conf. Decis. & Control*, 2019, pp. 7740–7745, <http://dx.doi.org/10.1109/CDC40024.2019.9029977>.
- [27] S. Mazumder, J. Lu, Faster-than-real-time simulation of Lithium Ion batteries with full spatial and temporal resolution, *Int. J. Electrochem. Soc.* 13 (2013) 1–10, <http://dx.doi.org/10.1155/2013/268747>.
- [28] V.R. Subramanian, V. Boovaragavan, V. Ramadesigan, M. Arabandi, Mathematical model reformulation for Lithium-Ion battery simulations: Galvanostatic boundary conditions, *J. Electrochem. Soc.* 156 (4) (2009) A260, <http://dx.doi.org/10.1149/1.3065083>.
- [29] M. Torchio, L. Magni, R.B. Gopaluni, R.D. Braatz, D.M. Raimondo, LIONSIMBA: A matlab framework based on a finite volume model suitable for Li-ion battery design, simulation, and control, *J. Electrochem. Soc.* 163 (7) (2016) A1192–A1205, <http://dx.doi.org/10.1149/2.0291607>.
- [30] K.A. Smith, C.D. Rahn, C.-Y. Wang, Control oriented 1d electrochemical model of lithium ion battery, *Energy Convers. Manag.* 48 (9) (2007) 2565–2578, <http://dx.doi.org/10.1016/j.enconman.2007.03.015>.
- [31] P.W. Northrop, V. Ramadesigan, S. De, V.R. Subramanian, Coordinate transformation orthogonal collocation model reformulation and simulation of electrochemical-thermal behavior of lithium-ion battery stacks, *J. Electrochem. Soc.* 158 (12) (2011) A1461–A1477, <http://dx.doi.org/10.1149/2.058112>.
- [32] R. Klein, N.A. Chaturvedi, J. Christensen, J. Ahmed, R. Findeisen, A. Kojic, Electrochemical model based observer design for a Lithium-Ion battery, *Trans. Control Syst. Technol.* 21 (2) (2013) 289–301, <http://dx.doi.org/10.1109/TCST.2011.2178604>.
- [33] Fortran programs for the simulation of electrochemical systems, 2020, <http://www.cchem.berkeley.edu/jsngrp/fortran.html> (accessed November 10, 2020).
- [34] M. Farkhondeh, C. Delacourt, Mathematical modeling of commercial LiFePO<sub>4</sub> electrodes based on variable solid-state diffusivity, *J. Electrochem. Soc.* 159 (2) (2011) A177–A192, <http://dx.doi.org/10.1149/2.073202>.
- [35] X.-C. Tang, L.-X. Li, Q.-L. Lai, X.-W. Song, L.-H. Jiang, Investigation on diffusion behavior of Li<sup>+</sup> in LiFePO<sub>4</sub> by capacity intermittent titration technique (CITT), *Electrochim. Acta* 54 (8) (2009) 2329–2334, <http://dx.doi.org/10.1016/j.electacta.2008.10.065>.
- [36] P.P. Prossini, M. Lisi, D. Zane, M. Pasquali, Determination of the chemical diffusion coefficient of lithium in LiFePO<sub>4</sub>, *Solid State Ion.* 148 (1) (2002) 45–51, [http://dx.doi.org/10.1016/S0167-2738\(02\)00134-0](http://dx.doi.org/10.1016/S0167-2738(02)00134-0).
- [37] M. Doyle, Y. Fuentes, Computer simulations of a Lithium-Ion polymer battery and implications for higher capacity next-generation battery designs, *J. Electrochem. Soc.* 150 (6) (2003) A706, <http://dx.doi.org/10.1149/1.1569478>.
- [38] M. Ecker, T.K.D. Tran, P. Dechent, S. Käbitz, A. Warnecke, D.U. Sauer, Parameterization of a physico-chemical model of a Lithium-Ion battery I. Determination of parameters, *J. Electrochem. Soc.* 162 (9) (2015) A1836–A1848, <http://dx.doi.org/10.1149/2.0551509>.
- [39] Y. Ma, M. Doyle, T.F. Fuller, M.M. Doeff, L.C.D. Jonghe, J. Newman, The measurement of a complete set of transport properties for a concentrated solid polymer electrolyte solution, *J. Electrochem. Soc.* 142 (6) (1995) 1859–1868, <http://dx.doi.org/10.1149/1.2044206>.
- [40] M. Ecker, S. Käbitz, I. Laresgoiti, D.U. Sauer, Parameterization of a physico-chemical model of a Lithium-Ion battery II. Model validation, *J. Electrochem. Soc.* 162 (9) (2015) A1849–A1857, <http://dx.doi.org/10.1149/2.0541509>.
- [41] C.Y. Wang, W.B. Gu, B.Y. Liaw, Micro-macroscopic coupled modeling of batteries and fuel cells I. Model development, *J. Electrochem. Soc.* 145 (10) (1998) 3407–3417, <http://dx.doi.org/10.1149/1.1838820>.
- [42] G.L. Plett, *Battery management systems*, in: *Volume I: Battery Modeling*, 2015.
- [43] L. Cai, R.E. White, Mathematical modeling of a lithium ion battery with thermal effects in COMSOL Inc. Multiphysics (MP) software, *J. Power Sources* 196 (14) (2011) 5985–5989, <http://dx.doi.org/10.1016/j.jpowsour.2011.03.017>.
- [44] A.C. Hindmarsh, P.N. Brown, K.E. Grant, S.L. Lee, R. Serban, D.E. Shumaker, C.S. Woodward, SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers, *Trans. Math. Softw.* 31 (3) (2005) 363–396, <http://dx.doi.org/10.1145/1089014.1089020>.
- [45] J.A.E. Andersson, J. Gillis, G. Horn, J.B. Rawlings, M. Diehl, Casadi – a software framework for nonlinear optimization and optimal control, *Math. Program. Comput.* 11 (1) (2019) 1–36, <http://dx.doi.org/10.1007/s12532-018-0139-4>.
- [46] L. Rao, J. Newman, Heatgeneration rate and general energy balance for insertion battery systems, *J. Electrochem. Soc.* 144 (8) (1997) 2697–2704, <http://dx.doi.org/10.1149/1.1837884>.