



A Segregated Approach for Modeling the Electrochemistry in the 3-D Microstructure of Li-Ion Batteries and Its Acceleration Using Block Preconditioners

Jeffery M. Allen¹ · Justin Chang¹ · Francois L. E. Usseglio-Viretta² · Peter Graf¹ · Kandler Smith²

Received: 21 January 2020 / Revised: 21 October 2020 / Accepted: 15 January 2021 /
Published online: 8 February 2021
© The Author(s) 2021

Abstract

Battery performance is strongly correlated with electrode microstructure. Electrode materials for lithium-ion batteries have complex microstructure geometries that require millions of degrees of freedom to solve the electrochemical system at the microstructure scale. A fast-iterative solver with an appropriate preconditioner is then required to simulate large representative volume in a reasonable time. In this work, a finite element electrochemical model is developed to resolve the concentration and potential within the electrode active materials and the electrolyte domains at the microstructure scale, with an emphasis on numerical stability and scaling performances. The block Gauss-Seidel (BGS) numerical method is implemented because the system of equations within the electrodes is coupled only through the nonlinear Butler–Volmer equation, which governs the electrochemical reaction at the interface between the domains. The best solution strategy found in this work consists of splitting the system into two blocks—one for the concentration and one for the potential field—and then performing block generalized minimal residual preconditioned with algebraic multigrid, using the FEniCS and the Portable, Extensible Toolkit for Scientific Computation libraries. Significant improvements in terms of time to solution (six times faster) and memory usage (halving) are achieved compared with the MULTifrontal Massively Parallel sparse direct Solver. Additionally, BGS experiences decent strong parallel scaling within the electrode domains. Last, the system of equations is modified to specifically address numerical instability induced by electrolyte depletion, which is particularly valuable for simulating fast-charge scenarios relevant for automotive application.

Keywords Lithium-ion batteries · Electrochemistry · Block preconditioners · PETSc · FEniCS

✉ Jeffery M. Allen
jeff.allen@nrel.gov

¹ Computational Science Center, NREL, Golden, CO 80401, USA

² Center for Integrated Mobility Sciences, National Renewable Energy Laboratory, Golden, CO 80401, USA

1 Introduction

With the increased interest in electric vehicles, there is a focus on increasing the performance of lithium-ion (Li-ion) batteries, especially their volumetric and gravimetric capacity and their ability to charge faster without degradation and safety issues. Electrode materials for Li-ion batteries are porous microstructures, with active material particles blended with conductive additives and binder being the typical solid matrix while an electrolyte solution fills the pores. During operation, conductivity enables electron transport through the connected solid phase of each electrode up to the current collectors, while ion migration and ion diffusion transport Li ions through the continuous electrolyte solution phase all along the cell thickness. Batteries are ideally designed to maximize migration and minimize diffusion. Electrons and Li ions react at the interface between active material and electrolyte and form Li-ion complexes that diffuse within the solid particles. The impact of the microstructure geometry, or topology, on these transport and reaction mechanisms, and on the overall cell electrochemical performances, has been considered by the battery modeling community at different scales.

A first family of Li-ion battery electrochemical models uses porous electrode theory and thus abstract microstructural heterogeneity of composite electrodes using effective macroscopic properties [14,15,23,28,40–43]. In porous electrode theory [15,16,33–35], electrodes are treated as homogeneous media with superimposed solid and electrolyte phases. This macro-homogeneous approach encompasses models that consider either uniform or nonuniform effective macroscopic properties throughout the electrode domain. Because the complex microstructure topology is not directly resolved, only a small number of degrees of freedom (DOF) is required, thus allowing fast computation. Such a method is then particularly suitable to investigate large electrode volumes—for instance, to investigate the impact of microstructural mesoscale heterogeneity or to perform sensitivity analyses that require a large number of calculations [10,32]. These models, however, are intrinsically limited by their macroscale approach. (i) First, equations are derived assuming particle morphology is spherical [14,15,23,28,40–43], which is not relevant for every electrode material, such as flake-like graphite [46]. (ii) Second, even though effective macroscopic parameter heterogeneity can be introduced in these models, the heterogeneity grid size is limited because a minimum volume is required to locally characterize and define macroscopic parameters. (iii) Third, homogenization induces a loss of information because it consists of a reduction in the microstructure DOFs. Indeed, typical macroscale models consider only four microstructure parameters [40]: (i) the porosity, which represents the available volume for the Li-ion diffusion; (ii) the electrochemically active specific surface area used to scale the reaction current; (iii) the particle diameter that controls the maximum diffusion length of the solid-phase diffusion; (iv) and, last, the factor of tortuosity, which characterizes the effect of the convoluted, tortuous path of the pores that hinders the Li-ion diffusion. This set of four parameters drastically simplifies the actual geometry and does not necessarily identify it uniquely. (iv) Fourth, determination of these microstructure parameters is challenging, especially for particle sizes for which numerical results are method-dependent [24,47] and for tortuosity factor for which the impact of the conductive additive and binder is difficult to precisely quantify [46]. Microstructure parameter errors are then propagated to the macro-homogeneous electrochemical model, resulting in increased uncertainty. In addition, even though a correct agreement can be achieved between macroscale homogeneous electrochemical model predictions and experimental data [10,46], this could be the result of an overfitting of the microstructure parameters, which could lead to an overinterpretation of the results and to an

inaccurately assessment of the validity of the model. Indeed, as for every model, discrepancy between experimental and numerical results can be reduced by either refining the model, i.e., reducing the number of model assumptions and introducing new parameters, or (over-)fitting existing parameters. (v) Fifth, the impact of local geometric features (e.g., cracks) on the electrochemical response is only partially considered because their contribution is quantified through their impact on the effective microstructure parameters, which might not be noticeable because of the volume averaging, whereas they might have a strong local impact on the electrochemical response and on degradation mechanisms.

Limitations on microstructure homogenization (limited number of microstructure parameters and partial representation of geometric features) and on microstructure homogenization heterogeneity (minimum grid size) provide a biased description of the microstructure. In-plane heterogeneity (defined as the solution variations within slices normal to the electrode thickness) and local heterogeneity (defined as the solution variations in the vicinity of geometric features) is controlled by the microstructure heterogeneity. At a low charge rate and/or for thin electrodes, a partial representation of transport-related parameters is of limited importance because solutions will be quite homogenous. Contrariwise, at high charge rate and/or for thick electrodes, both in-plane and local heterogeneities are expected to be significant and can trigger earlier degradation mechanisms. Therefore, macro-homogenous electrochemical models are particularly well suited for a low charge rate and/or for thin electrodes, but their limited heterogeneity description could lead to an underestimation of degradation mechanisms at a high charge rate and/or for thick electrodes even though they might capture the average response. The latter case corresponds to transport-limited scenarios critical for automotive application.

A second family of Li-ion battery electrochemical models has been developed recently to remedy these limitations and work at the microstructure scale. It consists of applying the governing equations on a mesh that represents the electrode microstructure geometry. This mesh is obtained through experimental observation [13,23,25,27,30,38,48,49], generally using X-ray tomography or Focus ion Beam Scanning Electron Microscopes (FIB-SEM). Some authors numerically generate the electrode microstructure either with ideal geometries [17,19,20] (such as packed spheres) or with ones that try to mimic the actual microstructure [21]. Because the microstructure geometry is used, some of the inconvenience of macro-homogenous electrochemical models are relieved: (i) no spherical assumptions; (ii) higher image resolution, typically in the order of 100 nm, which provides a much more refined description of the microstructure heterogeneity; (iii, iv) no effective microstructure parameters are required by definition; and (v) the impact of submicron geometric features on electrochemical performances can be evaluated locally. Microscale electrochemical models require extensive CPU capabilities because of the very large number of DOF, resulting in significant calculation times [20]. Therefore, to both achieve reasonable calculation times and model the full electrode thickness (essential to investigate transport-limited scenario), the authors used a relatively small in-plane section area of the electrode [23]. These trade-offs could induce inaccurate results because the investigated volumes are too small to be representative [23]. The current limitation in terms of simulated volume size is a significant bottleneck that limits microscale model predictability. Some larger volumes have been reported in the literature but at the price of an image resolution downscaling, resulting in a coarse description of the microstructure [13], which might result in the loss of small details and in a change of the surface area and roughness. In addition, even though micromodels exhibit a fine description of the active material, the scale difference between the diameter of the active material particles and those of the conductive additives and binder [50] prevents the latter from being accurately represented (i.e., meshed) at the nanoscale while preserving

a field of view large enough to be representative of the active material spatial distribution. Further, X-ray tomography typically does not distinguish the conductive additives and binder from the pores [46]. Two numerical approaches have been reported to partially capture the effect of this phase on the electrochemical response: (i) considering the electrolyte, conductive additives, and binder as a unique phase with effective transport properties [17,23]; and (ii) meshing the conductive additives and binder phase as a uniform thin layer coated at the surface of the active material particles with a high electronic conductivity [13,38]. Last, microstructure models suffer from numerical stability and meshing issues not present in macro-homogeneous models, both leading to simulation inaccuracy and divergence [23].

To provide quantitative, predictive results, microstructure-scale electrochemical models must first address computation time and numerical stability issues, even before considering additional physics. This work focuses on these two areas. Sect. 2 develops the mathematical model used for simulating Li-ion batteries, which is solved using the segregated domain scheme discussed in Sect. 3.3. Various linear solver methods are presented in Sect. 4, which are then compared in the numerical results in Sect. 5. Finally, the concluding remarks are found in Sect. 6.

2 Li-Ion Electrochemical Model

This section will describe the development of the electrochemical model used to simulate the flow of Li^+ ions. It should be noted that there are two types of simulations: full-cell and half-cell. In a full-cell simulation, both the cathode and anode are simulated along with the electrolyte. However, a half-cell simulation replaces one of the electrodes with a reference domain which provides a source (or sink) of Li ions. No simulation needs to take place in the reference domain, which makes a half-cell simulation less computationally complex. In this paper, we will only be running half-cell simulation. However, for completeness, this section will describe a full-cell simulation as that is the eventual goal of this project.

We will start by describing the physical domain we wish model followed by the system of equations and boundary conditions that describe the development of Li^+ concentration and potential fields. Next is a discussion of the interface conditions, which is vital to the coupling between concentration and potential within electrode domains as well as between the electrode and electrolyte domains. Once the model is fully developed, a nondimensionalization is presented to help with numerical performance. In order to simulate the electrochemistry properly, the appropriate initial conditions must be set up, which is described in Sect. 2.6. Finally, this section ends with a description of an optional modification that helps numerical performance when the cell is in a state called “electrolyte depletion.”

2.1 Domain Setup

In this work, we consider the following domains for a Li-ion cell, Ω , represented in Fig. 1: the positive (cathode) electrode active, Ω_c ; the negative (anode) electrode active material, Ω_a ; the electrolyte that fill the electrode pores, Ω_e ; and the separator, Ω_{sep} , a thin electron-blocking layer that prevents direct contact and short-circuits between the two electrodes, such that:

$$\Omega = \Omega_a \cup \Omega_c \cup \Omega_e \cup \Omega_{sep}.$$

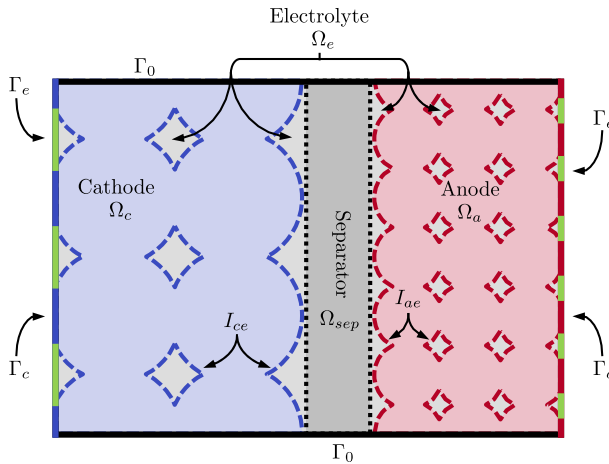


Fig. 1 Sketch of a Li-ion battery

The various additives [46] and the negative and positive current collector volumes are not represented in the domain for this model. When charging, Li ions (noted Li^+) inside of solid positive particles diffuse to the surface, where they react (de-intercalation) and transfer from the solid phase into the electrolyte phase. The positively charged ions travel through the electrolyte and the separator to the negative electrode, where they react and insert (intercalation) into solid negative particles. Electrons follow an opposite path through the electron-conductive solid particles, the current collectors, and the external circuit. More information on the construction of a Li-ion battery can be found in [36].

Figure 1 shows the various domains, boundaries, and interfaces that comprise a cell. Separating electrodes and electrolyte are the interfaces denoted as I_{ae} and I_{ce} . The boundary comprises four parts as well:

$$\Gamma = \Gamma_a \cup \Gamma_c \cup \Gamma_e \cup \Gamma_0,$$

where Γ_a is the boundary between the negative current collector and the anode, Γ_c is the boundary between the positive current collector and the cathode, Γ_e is where the electrolyte contacts either current collector, and Γ_0 corresponds to all the in-plane boundaries.

Note that the separator is a porous material filled with electrolyte that allows Li^+ to diffuse through but blocks electrons. In our model, the separator is homogenized. For this reason, the electrolyte and separator subdomains are treated as a single domain, $\Omega_e \cup \Omega_{sep}$, with different transport properties. This results in only three computational subdomains. For simplicity, unless otherwise specified, the electrolyte domain is considered to incorporate the separator domain.

2.2 Governing Equations

In this work, the mathematical model of a Li-ion cell is restricted to the Li^+ concentration, c , and the electric potential, ϕ , in all domains. Both the concentration and potential depend on space, x , and time t . The system of equations used to describe the flow of Li^+ is also separated into three parts. To condense notation, when used as a subscript, let $s = \{a, c\}$, which indicates a property belongs to the anode and cathode domains, respectively. Additionally, let

$i = \{a, e, c\}$ is the same as s but applies to all three computational domains. This way, instead of having to present similar sets of equations for each domain where the only difference is the use of c_a versus c_c , the notation can be simplified to c_s to refer to both equations simultaneously.

Starting with the electrode active materials, define the ionic and potential fluxes:

$$N_s = -D_s \nabla c_s, \tag{1}$$

$$j_s = -\kappa_s \nabla \phi_s, \tag{2}$$

These two depend on the diffusivity of Li^+ , D_s , and the conductivity, κ_s both of which depend on c_s . As previously noted, the subscript s is the notation to indicate that these values apply to an electrode (solid) domain. When specifically talking about the anode or cathode, the subscript will be a or c , respectively. For example, c_c would refer to the Li^+ concentration in the cathode, and c_a would refer to the concentration in the anode. Later, the subscript e will refer to the electrolyte. The flow of Li^+ in the electrodes is described by:

$$\begin{aligned} \frac{\partial c_s}{\partial t} &= -\nabla \cdot N_s, \\ 0 &= -\nabla \cdot j_s, \end{aligned}$$

for $x \in \Omega_s$ for $s = a, c$.

Next, consider the system of equations in the electrolyte domain. The main difference is in the definition of the ionic and potential fluxes:

$$N_e = -D_e \nabla c_e + \frac{t^+}{F} j_e, \tag{3}$$

$$j_e = -\kappa_e \nabla \phi_e + \kappa_D \nabla \ln(c_e), \tag{4}$$

where t^+ is the transference number for Li^+ , F is Faraday’s constant, D_e is the electrolyte ionic diffusivity, κ_e is the electrolyte ionic conductivity, and κ_D is the diffusional conductivity. The diffusional conductivity is defined as:

$$\kappa_D = (1 - t^+)(1 + A_c) \frac{2\kappa_e RT}{F},$$

where A_c is the activity coefficient, R is the gas constant, and T is temperature. The flow of Li^+ in the electrolyte is described by:

$$\begin{aligned} \frac{\partial c_e \varepsilon}{\partial t} &= -\nabla \cdot N_e, \\ 0 &= -\nabla \cdot j_e, \end{aligned}$$

for $x \in \Omega_e \cup \Omega_{sep}$, where ε is a measure of the porosity. Because of the large pore scale difference between the electrolyte domain, Ω_e , and the separator domain, Ω_{sep} , meshing both pores would result in a significant increase in DOF. To overcome this issue, the separator domain is considered as a homogeneous electrolyte domain, i.e., the separator is treated similarly with the electrolyte except for effective transport properties. To ensure mass conservation, an additional porosity term, ε , is introduced in the mass conservation equation with:

$$\varepsilon = \begin{cases} \varepsilon_{sep} & x \in \Omega_{sep} \\ 1 & x \in \Omega_e \end{cases},$$

where ε_{sep} is the bulk porosity of the separator. This form of mass conservation (with a porosity term) is used in macroscale electrochemical models [40]. As described, the present model

is then a multiscale electrochemical model because the homogenized medium (separator) and dense mediums (electrolyte and active materials) are solved concurrently.

2.3 Boundary Conditions

At the two current collectors, a non-permeability condition is applied to enforce that Li^+ cannot escape. This condition is represented as:

$$N_i \cdot \mathbf{n} = 0, \quad \text{for } x \in \Gamma_i.$$

Additionally, along the in-plane boundaries, Γ_0 , an identical no-flux boundary condition is enforced, represented as:

$$N_i \cdot \mathbf{n} = 0, \quad \text{for } x \in \Omega_i \cap \Gamma_0.$$

This condition results from the fact that the computational domain is a small portion of the full Li-ion battery.

Next, a current flux is applied to the two current collectors:

$$\begin{aligned} j_a \cdot \mathbf{n} &= g_a, & \text{for } x \in \Gamma_a, \\ j_c \cdot \mathbf{n} &= g_c, & \text{for } x \in \Gamma_c, \end{aligned}$$

where g_a and g_c are the current densities, constrained by:

$$\iint_{\Gamma_a} j_a \cdot \mathbf{n} = \iint_{\Gamma_c} j_c \cdot \mathbf{n}.$$

This constraint essentially enforces that the current flowing into the anode equals the current flowing out of the cathode. Also, notice that the sign of g_a should have the opposite sign of g_c because the outward normals are facing opposite directions. The current integral balance equation is not explicitly enforced because it is a natural extension of the physics solved with the boundary conditions applied. These boundary conditions drive the entire simulation. The final boundary condition enforces that current does not flow through the sides of the cell:

$$j_i \cdot \mathbf{n} = 0, \quad \text{for } x \in \Omega_i \cap \Gamma_0.$$

Here, $\Omega_i \cap \Gamma_0$ refers to the external boundary associated with each subdomain but excludes the current collectors. Additionally, where the electrolyte meets the current collectors:

$$j_e \cdot \mathbf{n} = 0, \quad \text{for } x \in \Gamma_e,$$

is enforced. A final boundary condition arises from a need to pin down the potential field. The only boundary conditions enforced on the potential are Neumann type, which allow for a valid solution to be shifted by a constant and remain valid. To combat this, a point within one of the electrode domains is set to a fixed potential value, thus removing the constant from the null space. This is not required for the concentration because it already depends on an initial condition.

2.4 The Interface Condition

The last aspect of the model is the interface condition. This condition communicates how the Li^+ is (de)intercalating at the electrochemically active interface between the solid particles and the electrolyte. The active interface is defined as the interface between the pore and

the solid particles, which is connected with the current collector. This condition is modeled using the Butler–Volmer equations [36]. First, define the exchange current density for each electrode:

$$i_s = Fk_s c_s^{\alpha_a} c_e^{\alpha_c} (c_s \max - c_s)^{\alpha_c},$$

where k_s is the kinetic constant, and $c_s \max$ is the maximum Li^+ concentration in the electrode. Additionally, α_a and α_c are the anodic and cathodic charge transfer exponent, respectively. Considering the symmetry factor ($\alpha_a + \alpha_c = 1$) and assuming that the reaction is equally reactant-like and product-like, then $\alpha_a = \alpha_c = 0.5$, so the exchange current density reduces to:

$$i_s = Fk_s \sqrt{c_s c_e (c_s - c_s \max)}.$$

Additionally, define the overpotential as:

$$\eta_s = \phi_s - \phi_e - U_s,$$

where U_s is the open circuit potential. Combining the exchange current density and overpotential according to the Butler–Volmer equation results in the Faraday current:

$$i_{se} = i_s \left(e^{\frac{\alpha_a F}{RT} \eta_s} - e^{-\frac{\alpha_c F}{RT} \eta_s} \right),$$

which can be simplified to:

$$= 2i_s \sinh \left(\frac{F}{2RT} \eta_s \right), \tag{5}$$

if $\alpha_a = \alpha_c = 0.5$. The interface conditions can now be represented as:

$$N_a \cdot \mathbf{n} = N_e \cdot \mathbf{n} = \frac{i_{ae}}{F}, \quad \text{and} \quad j_a \cdot \mathbf{n} = j_e \cdot \mathbf{n} = i_{ae} \quad \text{for } x \in I_{ae},$$

$$N_c \cdot \mathbf{n} = N_e \cdot \mathbf{n} = \frac{i_{ce}}{F}, \quad \text{and} \quad j_c \cdot \mathbf{n} = j_e \cdot \mathbf{n} = i_{ce} \quad \text{for } x \in I_{ce}.$$

Following is a summary of all the equations, boundary conditions, and interface conditions. System of equations:

$$-\nabla \cdot N_s = \frac{\partial c_s}{\partial t}, \quad \text{on } x \in \Omega_s, \tag{6}$$

$$-\nabla \cdot N_e = \frac{\partial c_e \varepsilon}{\partial t}, \quad \text{on } x \in \Omega_e \cup \Omega_{sep}, \tag{7}$$

$$-\nabla \cdot j_i = 0, \quad \text{on } x \in \Omega_i, \tag{8}$$

Boundary conditions:

$$j_a \cdot \mathbf{n} = g_a, \quad \text{on } x \in \Gamma_a, \tag{9}$$

$$j_c \cdot \mathbf{n} = g_c, \quad \text{on } x \in \Gamma_c, \tag{10}$$

$$j_e \cdot \mathbf{n} = 0, \quad \text{on } x \in \Gamma_e, \tag{11}$$

$$N_i \cdot \mathbf{n} = 0, \quad \text{on } x \in \Gamma_i, \tag{12}$$

$$N_i \cdot \mathbf{n} = 0, \quad \text{on } x \in \Gamma_0, \tag{13}$$

$$j_i \cdot \mathbf{n} = 0, \quad \text{on } x \in \Gamma_0, \tag{14}$$

Interface conditions:

$$N_a \cdot \mathbf{n} = N_e \cdot \mathbf{n} = \frac{i_{ae}}{F}, \quad j_a \cdot \mathbf{n} = j_e \cdot \mathbf{n} = i_{ae}, \quad \text{on } x \in I_{ae}, \quad (15)$$

$$N_c \cdot \mathbf{n} = N_e \cdot \mathbf{n} = \frac{i_{ce}}{F}, \quad j_c \cdot \mathbf{n} = j_e \cdot \mathbf{n} = i_{ce}, \quad \text{on } x \in I_{ce}, \quad (16)$$

$$\iint_{I_{ae}} i_{ae} \cdot \mathbf{n} = - \iint_{I_{ce}} i_{ce} \cdot \mathbf{n}. \quad (17)$$

The current integral balance equation, (17), is not explicitly enforced because it is a natural consequence of the physics solved with the boundary conditions applied. A table of all the parameters used in this model with a short description can be found in Table 1. Pay particular attention to which coefficients depend on concentration

2.5 Nondimensionalization

The electrochemical model includes coefficients spanning several order of magnitude and as a result will have a large condition number and will be difficult to solve efficiently and accurately. This numerical issue can be resolved by determining a set scaling coefficients so that the resulting modified system is dimensionless and its solutions are close to being unity. Problem nondimensionalization can be performed using various methods, and the scaling coefficients are not necessarily unique. This section presents one such method for the electrochemical model.

To help improve solver convergence, (6)–(17) is scaled to produce a nondimensional system of equations. To find this scaling, start with the electrolyte domain because it is the most complex. Plugging the definition j_e from (4) into the potential balance equation, (8), yields:

$$0 = \kappa_e \Delta \phi_e - (1 - t_+)(1 + A_c) \frac{2\kappa_e RT}{F} \Delta \ln(c_e). \quad (18)$$

Now, for the moment, assume all coefficients are constant with respect to concentration, and define:

$$C_0 = (1 - t_+)(1 + A_c) \frac{2RT}{F},$$

and let $\phi_e = C_0 \phi_e^*$. Using these definitions, (18) becomes:

$$0 = C_0 \kappa_e \Delta \phi_e^* - C_0 \kappa_e \Delta \ln(c_e).$$

Dividing out $\kappa_e C_0$ results in:

$$0 = \Delta \phi_e^* - \Delta \ln(c_e),$$

and the potential flux simplifies to:

$$j_e = -\nabla \cdot \phi_e^* + \nabla \cdot \ln(c_e). \quad (19)$$

The next step is to plug the new representation of j_e , (19), and N_e , (3), into the concentration balance equation, (7), which yields:

$$\frac{\partial c_e}{\partial t} = D_e \Delta c_e - C_1 C_0 \kappa_e \Delta \phi_e^* - C_1 C_0 \kappa_e \Delta \ln(c_e), \quad (20)$$

Table 1 Table of symbols

| Symbol | Units | | Description |
|--------------------|--|--------|---|
| <i>Equations</i> | | | |
| N_i | $\text{mol s}^{-1} \text{m}^{-1}$ | | Ionic flux |
| j_i | A m^{-2} | | Potential flux |
| i_s | A m^{-2} | | Exchange current density |
| η_s | V | | Overpotential |
| i_{se} | A m^{-2} | | Faraday current density at I_{ae} or I_{ce} |
| <i>Variables</i> | | | |
| c_i | mol m^{-3} | | Li^+ concentration |
| ϕ_i | V | | Potential |
| t | s | | Time |
| Symbol | Units | $f(c)$ | Description |
| <i>Parameters</i> | | | |
| D_i | $\text{m}^2 \text{s}^{-1}$ | ✓ | Diffusion coefficient |
| κ_i | S m^{-1} | ✓ | Electric conductivity |
| κ_D | A m^{-1} | ✓ | Diffusional conductivity |
| t_+ | - | ✓ | Transference number of Li^+ |
| R | $\text{J K}^{-1} \text{mol}^{-1}$ | | Universal gas constant |
| T | K | | Temperature |
| F | A s mol^{-1} | | Faraday constant |
| A_c | - | ✓ | Activity coefficient |
| α_s | - | | Charge transfer exponent ($\alpha_a + \alpha_c = 1$) |
| k_s | $\text{m}^{4-3\alpha} \text{mol}^{\alpha-1} \text{s}^{-1}$ | | Reaction rate |
| $c_{s,\text{max}}$ | mol m^{-3} | | Maximum concentration in active material |
| U_s | V | ✓ | Open circuit potential |
| g_s | A m^{-2} | | Applied current density applied to Γ_a or Γ_c |
| \mathbf{n} | - | | Outward unit normal |

In the “units” column, a dash, ‘-’, indicates a unitless quantity. The $f(c)$ column indicates whether the parameter can be a function of concentration

where $C_1 = \frac{t_+}{F}$. Using, $c_e = \frac{C_1 C_0 \kappa_e}{D_e} c_e^*$, (20) becomes:

$$\frac{C_1 C_0 \kappa_e}{D_e} \frac{\partial c_e^*}{\partial t} = C_1 C_0 \kappa_e \Delta c_e^* - C_1 C_0 \kappa_e \Delta \phi_e^* - C_1 C_0 \kappa_e \Delta \ln(c_e^*),$$

which reduces to:

$$\frac{1}{D_e} \frac{\partial c_e^*}{\partial t} = \Delta c_e^* - \Delta \phi_e^* - \Delta \ln(c_e^*). \tag{21}$$

Note that:

$$\Delta \ln \left(\frac{C_1 C_0 \kappa_e}{D_e} c_e^* \right) = \Delta \ln \left(\frac{C_1 C_0 \kappa_e}{D_e} \right) + \Delta \ln(c_e^*) = \Delta \ln(c_e^*)$$

because the coefficients are assumed constant.

Next, the length scale, \mathbf{x} , is nondimensionalized using $\mathbf{x} = L\mathbf{x}^*$, where L is in the characteristic length scale (such as 1e-6 for micrometers). This results in new derivative operators defined by:

$$\nabla = \frac{1}{L}\nabla^* \quad \text{and} \quad \Delta = \frac{1}{L^2}\Delta^*.$$

Using these operators in (21) and moving the L^2 to the left-hand side yields:

$$\frac{L^2}{D_e} \frac{\partial c_e^*}{\partial t} = \Delta^* c_e^* - \Delta^* \phi_e^* - \Delta^* \ln(c_e^*). \tag{22}$$

Finally, redefine the timescale to $t = \frac{L^2}{D_e} t^*$. Differentiating this results in:

$$\frac{d}{dt} = \frac{D_e}{L^2} \frac{d}{dt^*},$$

which when used in (22) reduces to:

$$\frac{\partial c_e^*}{\partial t^*} = \Delta^* c_e^* - \Delta^* \phi_e^* - \Delta^* \ln(c_e^*).$$

The full scaled system for the electrolyte domain is:

$$\frac{\partial c_e^*}{\partial t^*} = \Delta^* c_e^* - \Delta^* \phi_e^* - \Delta^* \ln(c_e^*), \tag{23}$$

$$0 = \Delta^* \phi_e^* - \Delta^* \ln(c_e^*), \tag{24}$$

with:

$$\mathbf{x} = L\mathbf{x}^*, \quad t = \frac{L^2}{D_e} t^*, \quad c_e = \frac{C_1 C_0 \kappa_e}{D_e} c_e^*, \quad \phi_e = C_0 \phi_e^*. \tag{25}$$

Performing dimensional analysis on \mathbf{x}^* , t^* , c_e^* , and ϕ_e^* reveals that they are all unitless. These same scalings could also be used in the solid electrode domain; however, because the segregated-domain scheme solves each domain separately, a different scaling could be used that was specifically designed for the anode and cathode independently. The only guideline is that the timescale and length scale remain the same among all domains. Although this is not strictly necessary, having the same timescale among all domains keeps them from getting out of sync with each other. Additionally, keeping the same length scale makes the transfer of information between domains simpler.

Using the definitions of N_s and j_s from (1) and (2) in the concentration and potential balance equations, (6) and (8), with the new length and timescales results in the system:

$$\frac{dc_s}{dt^*} = \frac{D_s}{D_e} \Delta^* c_s, \tag{26}$$

$$0 = \Delta^* \phi_s. \tag{27}$$

Again, this assumes that all coefficients are constant. Not much can be done about the $\frac{D_s}{D_e}$, so the best scaling is to let $c_s = c_{s,max} c_s^*$, which will keep c_s^* around order one. The scaling for the potential term is in a similar situation. A decent choice would be $\phi_s = C_0 \phi_s^*$ to mimic the electrolyte potential. Another possibility would be to use the maximum voltage if known such that $\phi_s = \phi \max \phi_s^*$.

Also note that this scaling does not take the interface condition, (5), into account. This means that before applying the Butler–Volmer condition, c_i and ϕ_i need to be in their dimensional form.

2.6 Initial Conditions

To perform a simulation, a few quantities must first be calculated. The first quantity is the current flux, g_s , which will be used to drive the electrochemical reaction in the Li-ion battery. This flux is dependent on a parameter known as the C-rate. The C-rate is a measure of how fast a battery is charging or discharging relative to its total theoretical capacity. By definition, 1C corresponds to the current required to fully discharge or charge a battery in 1 hour. For example, with a 4 Ah battery, a 1C charge will take an hour and require 4 A, but a 6C charge will take 10 minutes and 24 A. Because a main focus of this research is to use the model described in Eqs. (6)–(17) to simulate the charge/discharge cycles of various batteries at several C-rates, current flux, g , needs to be written in terms of C-rate, C_{rate} :

$$g_s = \frac{c_{a,max} F V_{\Omega_a}}{s_{hr} C_{rate} A \Gamma_s},$$

where V is the volume of the corresponding domain, A is the area of the corresponding boundary, and s_{hr} is the number of seconds in an hour. This flux uses the anode to calculate the maximum lithium at full charge. To use the cathode, replace c_a and V_{Ω_a} with c_c and V_{Ω_c} .

The next set of quantities that needs to be specified to run a simulation is the initial conditions. The initial concentrations, $c_{i,0}$, are relatively easy to compute. For the two electrodes, an initial state of charge (SOC) is chosen between 0 and 1, where 0 represents fully depleted, and 1 is fully charged. Then, multiplying the SOC by the corresponding $c_{s,max}$ gives the initial concentration within the two electrode domains. For the electrolyte, the initial concentration is simply set as a constant value.

The initial potentials are a bit more difficult to calculate. Because of the high nonlinearity in the Butler–Volmer equation, (5), a good initial guess is required to get Newton’s method to converge. Luckily, with the initial concentrations already known, the Butler–Volmer equation can be inverted to calculate the required initial potentials, $\phi_{i,0}$. Recall that one of the boundary conditions requires that the potential of one point of the cathode is fixed to an arbitrary value that will provide the reference potential for the whole domain. This reference value is then subtracted (only during post-processing) for all the potentials calculated. Because of this, the initial potential in the cathode is set to this base value, ϕ_0 :

$$\phi_{c^0} = \phi_0.$$

Next, calculate the initial potential in the electrolyte using:

$$\begin{aligned} \phi_{e^0} &= \phi_{c^0} - \bar{\eta}_c - U_c, \\ \bar{\eta}_c &= \frac{2RT}{F} \operatorname{asinh} \left(\frac{i_{ce}}{2i_c} \right). \end{aligned} \tag{28}$$

Finally, the initial potential in the anode is:

$$\begin{aligned} \phi_{a^0} &= \bar{\eta}_a + \phi_{e^0} + U_a, \\ \bar{\eta}_a &= \frac{2RT}{F} \operatorname{asinh} \left(\frac{i_{ae}}{2i_a} \right). \end{aligned} \tag{29}$$

As a final note, all of the simulations in this paper are known as half-cell simulations. This is because one of the electrodes is used as a reference domain, and all of its parameters are held fixed. This means that only the electrolyte and one electrode are simulated. For these simulations, the cathode is used as the reference and is considered as a lithium plate, with $U_c = 0$ V and $i_c = 100$ A m⁻², which is simply an arbitrary high value. This method

to determine sequentially the potential from a domain to its neighbors is only valid for the initialization step because it requires all fields (concentration and potential) of all domains (solid and electrolyte) to be uniform.

2.7 Electrolyte Depletion Modification

This section describes a potential modification to Eq. 4 can help mitigate issues with numerical convergence associate with a state known as electrolyte depletion. This modification is optional and its effects are examined in Sect. 5.3.

Thick electrodes and/or fast charging induce a significant electrolyte concentration gradient all along the cell thickness that could lead to local electrolyte depletion (i.e., $c_e \rightarrow 0$) [10]. This is an undesirable situation because part of the electrode is then no more electrochemically active ($i_{os} \rightarrow 0$ when $c_e \rightarrow 0$), whereas the remaining part is overused (the integral current density is constant), which in turn triggers a degradation mechanism that will decrease the cell calendar life [10]. Thick electrodes and fast charging however, are desired features for electric automotive application because they reduce cost and charging time, respectively. Therefore, the ability to simulate electrolyte depletion is especially valuable; however, numerical instabilities arise when approaching electrolyte depletion.

The most obvious problem occurs in (4), with the $\kappa_D \nabla \ln(c_e)$, which approaches infinity when $c_e \rightarrow 0$. This singularity can actually be removed by manipulating the definition of κ_D . First, use the chain rule:

$$\begin{aligned} \kappa_D \nabla \ln(c_e) &= \kappa_D \frac{\nabla c_e}{c_e}, \\ &= \frac{\kappa_D}{c_e} \nabla c_e. \end{aligned} \tag{30}$$

Next, expand with the current definition of κ_D :

$$\begin{aligned} &= \frac{(1 - t^+)(1 + A_c) \frac{2\kappa_e RT}{F}}{c_e} \nabla c_e, \\ &= \frac{\kappa_e}{c_e} \left((1 - t^+)(1 + A_c) \frac{2RT}{F} \nabla c_e \right). \end{aligned} \tag{31}$$

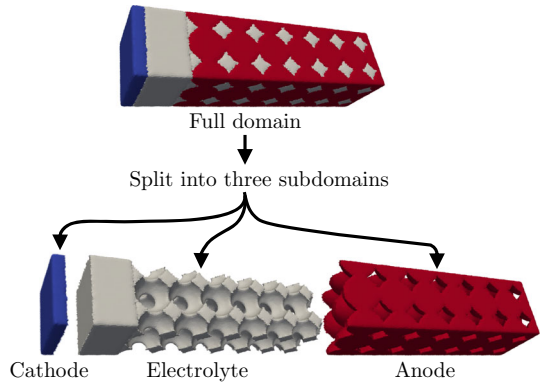
Now, take a closer look at the term $\frac{\kappa_e}{c_e}$. In this model of Li^+ transport, the conductivity in the electrolyte, κ_e , is approximated as an n th-degree polynomial:

$$\kappa_e \approx a_0 + \sum_{k=1}^n a_k c_e^k \tag{32}$$

Next, assume that the conductivity, κ_e , is zero when c_e is zero, which is the case for the electrolyte presented in this paper. This results in $a_0 = 0$ and $\frac{\kappa_e}{c_e}$ can be simplified to:

$$\begin{aligned} \frac{\kappa_e}{c_e} &\approx \frac{a_0 + \sum_{k=1}^n a_k c_e^k}{c_e}, \\ &\approx \sum_{k=1}^n a_k c_e^{k-1}. \end{aligned} \tag{33}$$

Fig. 2 Depiction of the domain splitting illustrated for a half-cell, with packed spheres for the anode



Now define:

$$\kappa_e^* = \sum_{k=1}^n a_k c_e^{k-1}, \tag{34}$$

and, as a result:

$$\kappa_D \nabla \ln(c_e) = \kappa_D^* \nabla c_e, \tag{35}$$

such that:

$$\kappa_D^* = \kappa_e^* (1 - t^+) (1 + A_c). \tag{36}$$

This modification can be used to overcome some of the numerical difficulties associated with electrolyte depletion.

3 Numerical Methods

This section discusses the various numerical methods used to address the electrochemical system for Li-ion batteries. Because the system of partial differential equations in (6)–(17) is set up with respect to three separate domains, a possible approach consists in solving this system in a computationally similar way. This results in the segregated-domain scheme. Instead of solving a single system over the whole domain, Ω , the segregated-domain scheme solves three independent systems on the three subdomains: Ω_a , Ω_e , and Ω_c , which is depicted in Fig. 2.

In this paper, we discretize the set of partial differential equations within each subdomain using continuous piecewise linear Galerkin finite elements implemented through the FEniCS Project [1,29]. To solve in one subdomain independently of the others, it must be assumed that the Li^+ concentration and potential fields in the other subdomains is fixed. Notice that the only communications between subdomains occur at the interface. This results in converting the interface conditions, (15) and (16), to boundary conditions, where either (c_s, ϕ_s) or (c_e, ϕ_e) are fixed, depending on which subdomain is currently being solved. Unless the fixed values of concentration and potential are the exact solutions to (6)–(17), the solution on the independently solved subdomain will not satisfy (6)–(17). This can be resolved by performing a Picard-like iteration between the three subdomains. Picard iteration is used for resolving a nonlinear system of equations by linearizing around a fixed value and then calculating an

update. The segregated-domain scheme is similar except values on two domains are fixed while calculating an update to the third.

3.1 Variational Form

Let W_i for $i = a, e, c$ be the continuous finite element space comprising piecewise linear functions defined on each computational domain. Additionally, let $v_i \in W_i$ for $i = a, e, c$ be the test function for concentration and $q_i \in W_i$ for $i = a, e, c$ be the test functions for potential. Backward (i.e., implicit) Euler is used to handle the time discretization, where τ is the time step, k is the time index, and T is the final time. Additionally, let \hat{c}_e and $\hat{\phi}_e$ be the fixed approximations to c_e and ϕ_e from the electrolyte domain. Then, within the solid active materials, the concentration functional is:

$$F_{c_s} = \int_{\Omega_s} \frac{(c_s^{k+1} - c_s^k)}{\tau} v_s \, d\mathbf{x} - \int_{\Omega_s} N_s \cdot \nabla v \, d\mathbf{x} - \int_{\Gamma_{se}} \frac{i_{se}(c_s, \hat{c}_e, \phi_s, \hat{\phi}_e)}{F} v \, ds,$$

and the potential functional is:

$$F_{\phi_s} = \int_{\Omega_a} j_a \cdot \nabla q \, d\mathbf{x} + \int_{\Gamma_{ae}} i_{ae}(c_a, \hat{c}_e, \phi_a, \hat{\phi}_e) q \, ds - \int_{\Gamma_a} g_a q \, ds.$$

The nonlinear variational problem on the solid active material domains is to find $(c_s, \phi_s) \in W_s \times W_s$ such that:

$$F_{c_s} + F_{\phi_s} = 0, \quad \forall (v_s, q_s) \in W_s \times W_s. \tag{37}$$

The functionals for the electrolyte domain are slightly different where the concentration functional is:

$$F_{c_e} = \int_{\Omega_e} \frac{(c_e^{k+1} - c_e^k)}{\tau} v \, d\mathbf{x} - \int_{\Omega_e} N_e \cdot \nabla v \, d\mathbf{x} - \int_{\Gamma_{ae}} \frac{i_{ae}(\hat{c}_a, c_e, \hat{\phi}_a, \phi_e)}{F} v \, ds - \int_{\Gamma_{ce}} \frac{i_{ce}(\hat{c}_a, c_e, \hat{\phi}_a, \phi_e)}{F} v \, ds,$$

and the potential functional is:

$$F_{\phi_e} = \int_{\Omega_e} j_e \cdot \nabla q \, d\mathbf{x} + \int_{\Gamma_{ae}} i_{ae}(\hat{c}_a, c_e, \hat{\phi}_a, \phi_e) q \, ds + \int_{\Gamma_{ce}} i_{ce}(\hat{c}_a, c_e, \hat{\phi}_a, \phi_e) q \, ds.$$

Similarly, the nonlinear variational problem on the electrolyte is to find $(c_s, \phi_s) \in W_s \times W_s$ such that:

$$F_{c_s} + F_{\phi_s} = 0, \quad \forall (v_s, q_s) \in W_s \times W_s. \tag{38}$$

The segregated-domain scheme iterates between these variational problems using the solutions on one domain to improve the variational form on the other two. Iterating among the three domains is the consistency iteration. The domains are considered consistent when:

$$\frac{\|c_i - \hat{c}_i\|}{\|c_{i0}\|} \leq \epsilon_c \text{ and } \frac{\|\phi_i - \hat{\phi}_i\|}{\|\phi_{i0}\|} \leq \epsilon_\phi,$$

for some tolerances ϵ_c and ϵ_ϕ .

3.2 Newton’s Method

Newton’s method in each of the segregated domains can be described as:

$$\mathbf{u}^{(n+1)} = \mathbf{u}^{(n)} - \delta \mathbf{u}, \tag{39a}$$

$$\mathbf{J} \cdot \delta \mathbf{u} = \mathbf{r}^{(n)}, \tag{39b}$$

where the superscript (n) denotes the Newton step, $\mathbf{u}^{(n)}$ is a solution vector of potentials and concentrations, $\delta \mathbf{u}$ is the update vector, $\mathbf{r}^{(n)}$ is the residual vector evaluated at $\mathbf{u}^{(n)}$, and $\mathbf{J} = F'(\mathbf{u}^{(n)})[\delta \mathbf{u}]$ is the Jacobian matrix of the functional F evaluated at $\mathbf{u}^{(n)}$. The relative Newton convergence criterion is defined as:

$$\frac{\|\mathbf{r}^{(n)}\|}{\|\mathbf{r}^{(0)}\|} \leq \epsilon_n,$$

where ϵ_n is some tolerance. The linear system of equations could also be represented in block form. Let $\mathbf{u}^{(n)}$ be rewritten as a vector of potentials, $\phi^{(n)}$, and concentrations, $\mathbf{c}^{(n)}$. Thus, Eq. (39) could also be rewritten as:

$$\begin{pmatrix} \phi^{(n+1)} \\ \mathbf{c}^{(n+1)} \end{pmatrix} = \begin{pmatrix} \phi^{(n)} \\ \mathbf{c}^{(n)} \end{pmatrix} - \begin{pmatrix} \delta \phi \\ \delta \mathbf{c} \end{pmatrix}, \tag{40a}$$

$$\begin{pmatrix} \mathbf{J}_{\phi\phi} & \mathbf{J}_{\phi c} \\ \mathbf{J}_{c\phi} & \mathbf{J}_{cc} \end{pmatrix} \begin{pmatrix} \delta \phi \\ \delta \mathbf{c} \end{pmatrix} = \begin{pmatrix} \mathbf{r}_{\phi}^{(n)} \\ \mathbf{r}_{c}^{(n)} \end{pmatrix}. \tag{40b}$$

The subscripts ϕ and c found in $\mathbf{r}^{(n)}$ and \mathbf{J} correspond to the potential and concentration blocks, respectively.

3.2.1 Constant Coefficients

Assuming all coefficients listed in Table 1, the residual functions for each of the solid domains in block form are:

$$\begin{aligned} \mathbf{r}_{\phi}^{(n)} &= -\nabla \cdot \kappa_s \nabla \phi_s^{(n)} - \hat{i}_{se} \Big|_{I_{se}}, \\ \mathbf{r}_{c}^{(n)} &= \frac{c_s^{(n)} - c_{s0}}{\Delta t} - \nabla \cdot D_s \nabla c_s^{(n)} - \frac{\hat{i}_{se}}{F} \Big|_{I_{se}}, \end{aligned}$$

where c_{s0} is the initial condition or solution from the previous time step, Δt is the discrete time increment, and \hat{i}_{se} is the Butler–Volmer equation evaluated at \hat{c}_e and $\hat{\phi}_e$ along the interface I_{se} . The corresponding Jacobian blocks are:

$$\begin{aligned} \mathbf{J}_{\phi\phi} &= -\nabla \cdot \kappa_s \nabla - \frac{\partial \hat{i}_{se}}{\partial u_s} \Big|_{I_{se}}, \\ \mathbf{J}_{\phi c} &= -\frac{\partial \hat{i}_{se}}{\partial c_s} \Big|_{I_{se}}, \\ \mathbf{J}_{c\phi} &= -\left(\frac{1}{F}\right) \frac{\partial \hat{i}_{se}}{\partial u_s} \Big|_{I_{se}}, \\ \mathbf{J}_{cc} &= \frac{\mathbf{I}}{\Delta t} - \nabla \cdot D_s \nabla - \left(\frac{1}{F}\right) \frac{\partial \hat{i}_{se}}{\partial c_s} \Big|_{I_{se}}, \end{aligned}$$

where \mathbf{I} is the identity matrix. The residual functions for the electrolyte domain in block form are:

$$\mathbf{r}_\phi^{(n)} = -\nabla \cdot \kappa_e \nabla \phi_e^{(n)} + \nabla \cdot \kappa_D \nabla \ln(c_e^{(n)}) - \hat{i}_{se} \Big|_{I_{se}}, \tag{41a}$$

$$\mathbf{r}_c^{(n)} = \frac{c_e^{(n)} - c_e^0}{\Delta t} - \nabla \cdot D_e \nabla c_e^{(n)} - \nabla \cdot \frac{t^+}{F} \kappa_e \nabla \phi_e^{(n)} + \nabla \cdot \frac{t^+}{F} \kappa_D \nabla \ln(c_e^{(n)}) - \frac{\hat{i}_{se}}{F} \Big|_{I_{se}}, \tag{41b}$$

where \hat{i}_{se} is now evaluated at \hat{c}_s and $\hat{\phi}_s$ along the interface I_{se} . The corresponding Jacobian is:

$$\begin{aligned} \mathbf{J}_{\phi\phi} &= -\nabla \cdot \kappa_e \nabla - \frac{\partial \hat{i}_{se}}{\partial u_e} \Big|_{I_{se}}, \\ \mathbf{J}_{\phi c} &= \nabla \cdot \frac{\kappa_D}{c_e^{(n)}} \nabla - \nabla \cdot \frac{\kappa_D}{c_e^{(n)2}} \nabla c_e^{(n)} - \frac{\partial \hat{i}_{se}}{\partial c_e} \Big|_{I_{se}}, \\ \mathbf{J}_{c\phi} &= -\nabla \cdot \frac{t^+}{F} \kappa_e \nabla - \left(\frac{1}{F} \right) \frac{\partial \hat{i}_{se}}{\partial u_e} \Big|_{I_{se}}, \\ \mathbf{J}_{cc} &= \frac{\mathbf{I}}{\Delta t} - \nabla \cdot D_e \nabla + \nabla \cdot \frac{t^+ \kappa_D}{F c_e^{(n)}} \nabla - \nabla \cdot \frac{t^+ \kappa_D}{F c_e^{(n)2}} \nabla c_e^{(n)} - \left(\frac{1}{F} \right) \frac{\partial \hat{i}_{se}}{\partial c_e} \Big|_{I_{se}}. \end{aligned}$$

3.2.2 Nonconstant Coefficients

When any of the coefficients from Table 1 are functions of concentration, the Jacobian blocks become more complex, especially for the electrolyte domain. Assuming that both κ_s and D_s are functions of c_s , the Jacobian for the solid domains would be written as:

$$\begin{aligned} \mathbf{J}_{\phi\phi} &= -\nabla \cdot \kappa_s \nabla - \frac{\partial \hat{i}_{se}}{\partial u_s} \Big|_{I_{se}}, \\ \mathbf{J}_{\phi c} &= -\nabla \cdot \frac{\partial \kappa_s}{\partial c_s} \nabla \phi_s^{(n)} - \frac{\partial \hat{i}_{se}}{\partial c_s} \Big|_{I_{se}}, \\ \mathbf{J}_{c\phi} &= -\left(\frac{1}{F} \right) \frac{\partial \hat{i}_{se}}{\partial u_s} \Big|_{I_{se}}, \\ \mathbf{J}_{cc} &= \frac{\mathbf{I}}{\Delta t} - \nabla \cdot D_s \nabla - \nabla \cdot \frac{\partial D_s}{\partial c_s} \nabla c_s^{(n)} - \left(\frac{1}{F} \right) \frac{\partial \hat{i}_{se}}{\partial c_s} \Big|_{I_{se}}. \end{aligned}$$

For the electrolyte, under the assumptions outlined in Sect. 2.7, Eq. (41) can be simplified to:

$$\begin{aligned} \mathbf{r}_\phi^{(n)} &= -\nabla \cdot \kappa_e \nabla \phi_e^{(n)} + \nabla \cdot \kappa_D^* \nabla c_e^{(n)} - \hat{i}_{se} \Big|_{I_{se}}, \\ \mathbf{r}_c^{(n)} &= \frac{c_e^{(n)} - c_e^0}{\Delta t} - \nabla \cdot D_e \nabla c_e^{(n)} - \nabla \cdot \frac{t^+}{F} \kappa_e \nabla \phi_e^{(n)} + \nabla \cdot \frac{t^+}{F} \kappa_D^* \nabla c_e^{(n)} - \frac{\hat{i}_{se}}{F} \Big|_{I_{se}}. \end{aligned}$$

Now, assuming that κ_e , κ_D^* , and D_e are all functions of c_e , the corresponding Jacobian becomes:

$$\mathbf{J}_{\phi\phi} = -\nabla \cdot \kappa_e \nabla - \frac{\partial \hat{i}_{se}}{\partial u_e} \Big|_{I_{se}},$$

$$\begin{aligned}
 \mathbf{J}_{\phi c} &= -\nabla \cdot \frac{\partial \kappa_e}{\partial c_e} \nabla \phi_e^{(n)} + \nabla \cdot \kappa_D^* \nabla + \nabla \cdot \frac{\partial \kappa_D^*}{\partial c_e} \nabla c_e^{(n)} - \frac{\partial \hat{i}_{se}}{\partial c_e} \Big|_{I_{se}}, \\
 \mathbf{J}_{c\phi} &= -\nabla \cdot \frac{t^+}{F} \kappa_e \nabla - \left(\frac{1}{F} \right) \frac{\partial \hat{i}_{se}}{\partial u_e} \Big|_{I_{se}}, \\
 \mathbf{J}_{cc} &= \frac{\mathbf{I}}{\Delta t} - \nabla \cdot D_e \nabla - \nabla \cdot \frac{\partial D_e}{\partial c_e} \nabla c_e^{(n)} - \nabla \cdot \frac{t^+}{F} \kappa_D^* \nabla - \nabla \cdot \left(\frac{t^+}{F} \right) \frac{\partial \kappa_D^*}{\partial c_e} \nabla c_e^{(n)} \\
 &\quad - \nabla \cdot \left(\frac{\kappa_D^*}{F} \right) \frac{\partial t^+}{\partial c_e} \nabla c_e^{(n)} - \left(\frac{1}{F} \right) \frac{\partial \hat{i}_{se}}{\partial c_e} \Big|_{I_{se}}.
 \end{aligned}$$

3.3 Segregated-Domain Scheme Algorithm

Now that all the parts are in place, we present the algorithm used for the segregated-domain scheme displayed in Algorithm 1.

Algorithm 1: Segregated-Domain Scheme

```

t ← 0;
while t ≤ T do
  while ||c_i - ĉ_i|| / ||c_i,0|| ≥ ε_c and ||φ_i - φ̂_i|| / ||φ_i,0|| ≥ ε_c do // Consistency Iteration
    for i in (a, e, c) do
      (ĉ_j, φ̂_j) ← (c_j, φ_j) for j ≠ i;
      F ← F_{c_i} + F_{φ_i};
      u^0 ← (c_i, φ_i);
      n ← 0;
      while ||r^{(n)}|| / ||r^{(0)}|| ≥ ε_n do // Newton Iteration
        J ← F'(u^{(n)})[δu];
        r^{(n)} ← F(u^{(n)});
        solve // Linear Solve Iteration
          | J · δu = r^{(n)}; // Using MUMPS or BGS
        end
        u^{(n+1)} ← u^{(n)} - δu;
        n ← n + 1
      end
    end
  end
  t ← t + τ;
end

```

Within each time step, the segregated-domain scheme starts by solving for concentration, c_a , and potential, ϕ_a , within the anode domain while assuming c_e and ϕ_e are fixed, which results in enforcing the interface conditions as the boundary condition depending on:

$$i_{ae}(c_a, \hat{c}_e, \phi_a, \hat{\phi}_e) = i_a(c_a, \hat{c}_e) \left(e^{\frac{\alpha_a F}{RT} \eta_a(\phi_a, \hat{\phi}_e)} - e^{-\frac{\alpha_c F}{RT} \eta_a(\phi_a, \hat{\phi}_e)} \right), \tag{42}$$

where:

$$\begin{aligned}
 i_a(c_a, \hat{c}_e) &= k_a \sqrt{c_a \hat{c}_e (c_a - c_a \max)}, \\
 \eta_a(\phi_a, \hat{\phi}_e) &= \phi_a - \hat{\phi}_e - U_a,
 \end{aligned}$$

and $\hat{c}_e, \hat{\phi}_e$ are the fixed approximations to c_e and ϕ_e in the electrolyte. Once Newton's method has converged and produced approximations to c_a and ϕ_a , it is time to update the electrolyte fields based on fixed values from the anode and cathode using an equation similar to (42). Finally, the fields in the cathode are updated. One cycle of solves on the anode, electrolyte, cathode is referred to as a consistency iteration and is labeled as such in Algorithm 1. The consistency iteration is repeated until the difference between the newly computed values, (c, ϕ) , and the fixed values, $(\hat{c}, \hat{\phi})$, reach some tolerance.

Because the coupling between domains only occurs at the interfaces, I_{ae} and I_{ce} , the fixed values, $(\hat{c}, \hat{\phi})$, from the previous domain's solve only needs to be interpolated to the next domain at the common interface. In FEniCS, this is achieved by first constructing the mesh to that the DOFs at the interface are shared between the relevant domains (the solid mesh and the electrolyte mesh are then conforming meshes). In other words, the anodes DOFs that lie in I_{ae} are duplicated in the electrolyte's copy of I_{ae} . Then, during the setup phase of the simulation, a map can be constructed that takes the values at the interface from the anode's function space to the electrolyte's functions space. This allows for communication between domains without the need for interpolation.

For the tests within this paper, the relative tolerance for the consistency iteration, ϵ_c , was set to 5×10^{-4} and the relative tolerance for the Newton iteration, ϵ_n was set to 1×10^{-7} . As for the linear solver, When using block Gauss-Seidel, relative tolerance was 1×10^{-7} . A possible way to reduced the solve time for the segregated-domain scheme would be to loosen the tolerances for the Newton and linear solve iterations. However, the linear solver's tolerance was chosen to reach a similar level of accuracy as the direct solver. Further simulation would be needed to find optimal values for these tolerances that balance speed and accuracy.

One advantage to using the segregated-domain scheme is that the solve on each subdomain requires less memory than solving a single large system. This means that for a fixed memory limit, the segregated-domain scheme can be applied to larger meshes.

The trade-off is the consistency iteration loop. If there is a large change in concentration or potential between times steps, more consistency iteration is expected. These large changes tend to occur at the beginning of the simulation or where the charge rate varies. Note that proof of the convergence of the consistency iteration is an open question, but in practice we find it converges reliably in 3–10 iterations depending on the time step size.

4 Linear Solvers

Solving the linear system from Eq. (39b) is computationally the most expensive step in our proposed segregated framework. The key to accelerating the overall time to solution is to solve the linear systems of equations in a scalable and efficient way. For notational convenience, let us rewrite Eq. (39b), which solves a linear system of N equations in N variables as:

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \quad (43)$$

where $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the coefficient matrix, $\mathbf{b} \in \mathbb{R}^N$ is the right-hand side vector, and $\mathbf{x} \in \mathbb{R}^N$ is the vector of variable unknowns. If all N equations are linearly independent, then there exists a unique solution $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$. Instead of explicitly computing and storing the inverted \mathbf{A} matrix, one can apply the action of \mathbf{A}^{-1} through either direct or iterative solvers.

4.1 Background on Direct and Iterative Solvers

One way to directly solve the system of equations is to first perform a lower-upper (LU) decomposition of A . That is:

$$A = LU,$$

where L and U are the lower and upper triangular matrices, respectively. The diagonal elements should be all ones for one matrix and all zeros for the other. Using this decomposition, Eq. (43) can be rewritten as:

$$LUx = b,$$

which is solved by solving these two systems of equations consecutively:

$$Ly = b,$$

$$Ux = y.$$

These systems are solved using forward and backward sweeps, respectively, and have an algorithmic complexity proportional to the number of nonzeros in A ; however, the L and U matrices are computationally expensive to analyze and factorize; the algorithmic complexity can be as high as $\mathcal{O}(N^3)$ [45]. Several research efforts over the years have attempted to mitigate the cost by exploiting the sparsity patterns through techniques such as matrix reordering, but even with these improvements, direct approaches still do not scale well with N in terms of both algorithmic complexity and memory consumption [5,11,44].

An alternative is to use an iterative approach to compute the action of A^{-1} . An initial guess, \mathbf{x}_0 , is used to calculate an eventual sequence of iterates $\{\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \dots\}$. The idea of an iterative scheme is that at some iterate k , the solution \mathbf{x}_k will be close enough to the true solution to be used as the approximate solution. To determine whether the approximate solution at iterate k is accurate enough, we evaluate the distance in error between the approximate and exact solutions with the residual vector:

$$\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k.$$

The iterative approach employed in this work is the Krylov subspace (KSP) method. The KSP of dimension k belonging to A and \mathbf{r}_0 is defined as:

$$\mathcal{K}_k(A, \mathbf{r}_k) = \text{span} \left\{ \mathbf{r}_0, A\mathbf{r}_0, \dots, A^{k-1}\mathbf{r}_0 \right\}.$$

KSP methods generate iterates in the following way:

$$\mathbf{x}_k \in \mathbf{x}_0 + \mathcal{K}_k(A, \mathbf{r}_k).$$

For example, the simplest form of the KSP solver can be written as:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{r}_k,$$

but in this work, we use the more complex generalized minimum residual (GMRES) method [39], which generates orthonormal vectors $\{\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \dots\}$ through the Gram-Schmidt process to form a basis for \mathcal{K}_k . Each iteration of GMRES requires the following matrix-vector product:

$$\mathbf{w} = A\mathbf{v}_k, \tag{44}$$

where \mathbf{w} is the resulting vector used to update the solution. Krylov methods such as GMRES are said to have the optimality property if in each iteration the computed iterate is the best

possible approximation of the solution within the current KSP; however, the convergence rate of the iterates produced by GMRES is limited by the Krylov subspace itself. That is, if the matrix A is either ill-conditioned or very large and complex, GMRES will converge slowly or in some cases not converge at all.

To improve the convergence rate of the iterative solvers, we need to supply a preconditioner matrix P to modify the KSP. In this work, we consider only left-preconditioning. The linear system of equations in (43) with left-preconditioning becomes:

$$P^{-1}A = P^{-1}b.$$

The preconditioned residual for this is:

$$r_k = P^{-1}(b - Ax_k).$$

The KSP for this new linear system of equations is:

$$\mathcal{K}_k(P^{-1}A, P^{-1}r_k).$$

In the preconditioned GMRES approach, the action of P^{-1} alters the spectrum of the linear system by transforming w from Eq. (44) into a new vector z . That is:

$$z = P^{-1}w. \tag{45}$$

Choosing the right preconditioner such that $P^{-1}A \approx I$ is the task at hand. For elliptic PDEs such as the diffusion or heat equations, the algebraic multigrid (AMG) approach is a good way to construct and apply the action of P^{-1} .

4.2 Block Preconditioning

If Eq. (43) contains multiple physics or domains, applying GMRES with algebraic multigrid preconditioning to the entire linear system will not always yield a convergent solution. Such systems require careful preconditioning techniques where the preconditioning matrix is decomposed into individual blocks. First, let us rewrite Eq. (45) in block form:

$$\begin{pmatrix} P_{\phi\phi} & P_{\phi c} \\ P_{c\phi} & P_{cc} \end{pmatrix} \begin{pmatrix} z_\phi \\ z_c \end{pmatrix} = \begin{pmatrix} w_\phi \\ w_c \end{pmatrix}, \tag{46}$$

where the subscripts ϕ and c correspond to the potential and concentration blocks in Eq. (40b), respectively. We now highlight two common block preconditioning approaches for Eq. (46):

1. *Block Jacobi (BJacobi)* In the BJacobi approach, we set all the blocks in P equal to their corresponding blocks in J . We set the initial guess for both z_ϕ and z_c to 0 so their solutions under this approach are obtained by solving these two inner linear systems of equations:

$$\begin{aligned} P_{\phi\phi}z_\phi &= w_\phi, \\ P_{cc}z_c &= w_c. \end{aligned}$$

The inverses of the diagonal Jacobian blocks are approximated using GMRES coupled with algebraic multigrid preconditioning. Alternatively, one could choose to apply only a single sweep of algebraic multigrid preconditioning to these two equations. This will save computational costs associated with fully solving the two inner systems of equations, but more outer GMRES iterations might be required.

2. *Block Gauss-Seidel (BGS)* The BGS approach is the method of successive displacement. That is, the block equations for \mathbf{z}_ϕ and \mathbf{z}_c are solved sequentially. The solution vector obtained from the first block solve is used to obtain the second solution vector. We again set all the blocks in \mathbf{P} equal to their corresponding blocks in \mathbf{J} and set the initial guesses to 0, resulting in the following two inner linear systems of equations:

$$\begin{aligned} \mathbf{P}_{\phi\phi}\mathbf{z}_\phi &= \mathbf{w}_\phi, \\ \mathbf{P}_{cc}\mathbf{z}_c &= \mathbf{w}_c - \mathbf{P}_{c\phi}\mathbf{z}_\phi. \end{aligned}$$

These two blocks can also be solved for either through GMRES with algebraic multigrid preconditioning or through a single algebraic multigrid preconditioning sweep. One can alternatively rearrange the block system so that the concentration block is solved first. That is:

$$\begin{aligned} \mathbf{P}_{cc}\mathbf{z}_c &= \mathbf{w}_c, \\ \mathbf{P}_{\phi\phi}\mathbf{z}_\phi &= \mathbf{w}_\phi - \mathbf{P}_{\phi c}\mathbf{z}_c. \end{aligned}$$

In this paper, we employ the BGS approach to solve the linear system of equations associated with all three domains. Both the concentration and potential blocks are approximated using GMRES that is preconditioned with AMG. This setup was chosen because the system of equation within the solid domains, Ω_s . If the interface condition is ignored, then (6) and (8) are completely decoupled. These two independent systems are similar to a nonlinear heat equation and a nonlinear Poisson equation. Both of these systems are easily handled by GMRES that is preconditioned with AMG. As for the electrolyte domain, we are still investigating potentially better solvers. To this end, part of our future work is to derive a Schur complement approximation to handle nonlinear coupling within the electrolyte domain.

4.3 Parallel Implementation

To accelerate our proposed block preconditioning approach even further, we need to use parallelism. Specifically, we leverage the Portable, Extensible Toolkit for Scientific Computation (PETSc) library [3,4,12], a numerical linear algebra library built on top of the message passing interface (MPI). It not only has scalable nonlinear and optimization-based solvers (see [7,8]), but it can interface to other important math libraries such as HYPRE [18]. To handle the block system in Eq. (40b), we use PETSc's composable linear solver capabilities (referred to as PCFieldSplit) to approximate the inverse of the $\mathbf{J}_{\phi\phi}$ and \mathbf{J}_{cc} blocks individually using the BGS approach discussed previously. This is achieved by setting up a multiplicative FieldSplit within PETSc. The solve is decomposed into two levels of iterations. The outer iteration is simply GMRES, which relies on the inner iteration consisting of a single application of HYPRE's algebraic multigrid BoomerAMG [22] solver on the separate potential and concentration blocks. In the next section, we compare the performance of the BGS solver with that of a popular parallel LU solver, the MULTifrontal Massively Parallel sparse direct Solver (MUMPS) package [2]. All numerical experiments are carried out on Dual Intel Xeon Gold Skylake 6154 (3.0-GHz, 18-core) server nodes with 96, 192, or 768 GB of memory

5 Numerical Results

This section discusses the numerical results of two simulated batteries to explore how the block solver performs. First, a simple generated battery domain is used to compare the solver

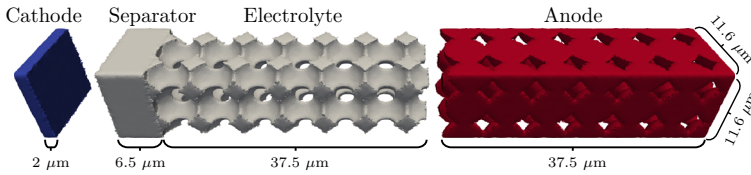


Fig. 3 The 7-particle test domain for examining the performance of BGS

performance of BGS and MUMPS as well as investigate the parallel performance of BGS. Second, a more complex and real battery domain is used to investigate how BGS performs under electrolyte depletion conditions.

5.1 Solver Comparison

To compare MUMPS and BGS, a simple battery geometry was generated. This domain consisted of seven particles in line with the x -direction (electrode thickness) and repeated in the y - and z -directions, which is represented in Fig. 3. The mesh was created using the MATLAB-based mesh generation `ISO2mesh` toolbox [37]. The initial concentrations for this problem were:

$$\begin{aligned} c_{0a} &= 0.05c_{a,\max}, \\ c_{0e} &= 1200 \text{ mol m}^{-3}, \\ c_{0c} &= 0.5c_{c,\max}, \end{aligned}$$

with:

$$\begin{aligned} c_{a,\max} &= 30000.0 \text{ mol m}^{-3} \\ c_{c,\max} &= 9 \times 10^9 \text{ mol m}^{-3} \end{aligned}$$

For this simulation, the cathode is used as the reference domain, so no actual calculation will occur within it. A reference voltage, ϕ_0 , of 5 V is arbitrarily chosen and set as the cathode potential for the whole calculation. The initial concentration of the cathode is of no importance because its overcharge potential (OCP) is set constant equal to 0 V, and its exchange current density, i_{oc} , is set to a high constant value (100 A m^{-2}) because the cathode is considered to be a lithium plate. The remaining potentials were calculated using Eqs. (28) and (29). The simulation persisted until the intercalation fraction in the anode was 0.95.

The first series of tests performed involve increasing the number of DOFs while recording memory requirements and time to solution. Because these are 3-D simulations, uniform refinement of an initial mesh would increase the number of DOFs by a factor of around 8. In other words, it would take only a few uniform refinements before the meshes grew beyond the limitations of the hardware. To prevent this while still having comparable results, three initial meshes were chosen that resulted in approximately 0.5, 1.0, 2.0 million DOFs for the entire domain. These meshes were then uniformly refined three times to provide nine experimental meshes. All of these tests were performed on a single node with 16 processes.

For each mesh, the simulation was performed to completion or until it ran out of allocated time with a maximum of 10 days. During a simulation, anytime a solve occurred in the electrolyte or anode, the maximum memory and wall clock time were recorded. Keep in mind that a single solve on a domain will happen multiple times because the Newton and

Table 2 The raw data used to compare MUMPS and BGS on the Anode solve

| Test | DOFs in | Memory usage (GB) | | | Solve time (s) | | |
|--------------|----------|-------------------|--------|-------|----------------|--------|---------|
| Index | Millions | MUMPS | BGS | Ratio | MUMPS | BGS | Speedup |
| <i>Anode</i> | | | | | | | |
| 1 | 0.26 | 9.99 | 7.41 | 1.35 | 2.66 | 0.53 | 5.00 |
| 2 | 0.51 | 13.46 | 10.24 | 1.31 | 5.65 | 1.06 | 5.32 |
| 3 | 1.02 | 23.33 | 16.02 | 1.46 | 12.62 | 2.22 | 5.68 |
| 4 | 1.74 | 37.75 | 24.31 | 1.55 | 24.74 | 3.48 | 7.10 |
| 5 | 3.52 | 75.83 | 44.39 | 1.71 | 58.20 | 7.50 | 7.76 |
| 6 | 7.15 | 163.65 | 78.95 | 2.07 | 143.64 | 15.49 | 9.27 |
| 7 | 12.58 | 323.84 | 130.76 | 2.48 | 333.40 | 29.20 | 11.42 |
| 8 | 25.76 | 730.90 | 263.83 | 2.77 | 1042.77 | 61.42 | 16.98 |
| 9 | 52.90 | OOM | 492.61 | – | OOM | 171.21 | – |

All simulations use the 7-particle domain on a single node with 16 processors. The mesh from the italic row was used to perform the strong scaling study in Fig. 9

Table 3 The raw data used to compare MUMPS and BGS on the Electrolyte solve

| Test | DOFs in | Memory usage (GB) | | | Solve time (s) | | |
|--------------------|----------|-------------------|--------|-------|----------------|--------|---------|
| Index | Millions | MUMPS | BGS | Ratio | MUMPS | BGS | Speedup |
| <i>Electrolyte</i> | | | | | | | |
| 1 | 0.22 | 10.02 | 7.46 | 1.34 | 2.51 | 0.88 | 2.85 |
| 2 | 0.45 | 13.91 | 10.35 | 1.34 | 5.75 | 1.81 | 3.17 |
| 3 | 0.90 | 25.83 | 16.26 | 1.59 | 14.21 | 4.03 | 3.53 |
| 4 | 1.53 | 37.96 | 24.50 | 1.54 | 25.42 | 6.15 | 4.13 |
| 5 | 3.14 | 85.54 | 44.83 | 1.91 | 69.68 | 13.95 | 4.99 |
| 6 | 6.46 | 201.23 | 79.39 | 2.53 | 214.12 | 33.57 | 6.38 |
| 7 | 11.22 | 341.71 | 130.48 | 2.62 | 394.13 | 65.72 | 6.00 |
| 8 | 23.26 | OOM | 263.37 | – | OOM | 152.36 | – |
| 9 | 48.33 | OOM | 490.31 | – | OOM | 340.25 | – |

All simulations use the 7-particle domain on a single node with 16 processors. The mesh from the italic row was used to perform the strong scaling study in Fig. 9

Consistency loops discussed in Algorithm 1. The average memory usage and wall clock time for each domain and all meshes are found in Tables 2 and 3. Some simulations ran out of memory, which is denoted as OOM. As a note, The “Test Index” indicates which meshes were being used in the same simulation. For example, Test 4 contained the electrolyte mesh with 1.53 M DOFs and the anode mesh with 1.74 M DOFs. The full simulation for Test 4 involved solving on those two meshes independently and sharing information across their interface, Γ_{ae} .

The data in Tables 2 and 3 is also presented in Figs. 4 and 5. Figure 4 show how the time to complete a single solve on a specific domain grows with increasing DOFs. The iterative solver, BGS, is between 5.00 to 16.98 times faster than MUMPS when solving in the anode domain, and between 2.85 to 6.38 times faster than MUMPS in the electrolyte domain. The speed advantage of BGS is expected to increase as the number of DOFs increases. The

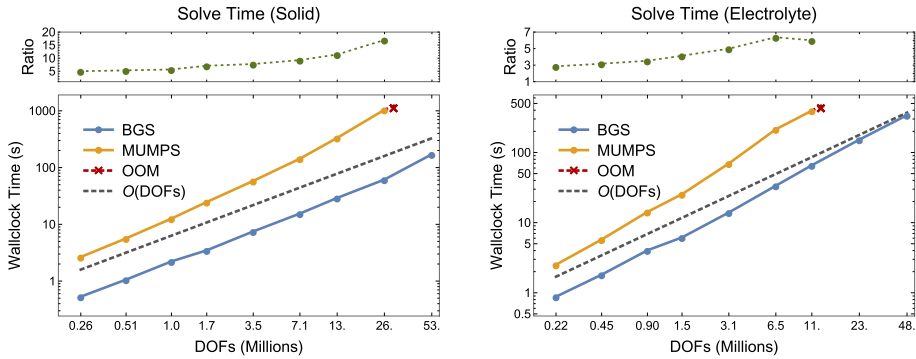


Fig. 4 The bottom set of plots are a comparison of the time to solution for one linear solve of each solver method: MUMPS or BGS. The timing data was collected by taking the average of every domain specific solve for an entire simulation. Additionally, the top set of plots show the ratio of MUMPS computational time to BGS computational time

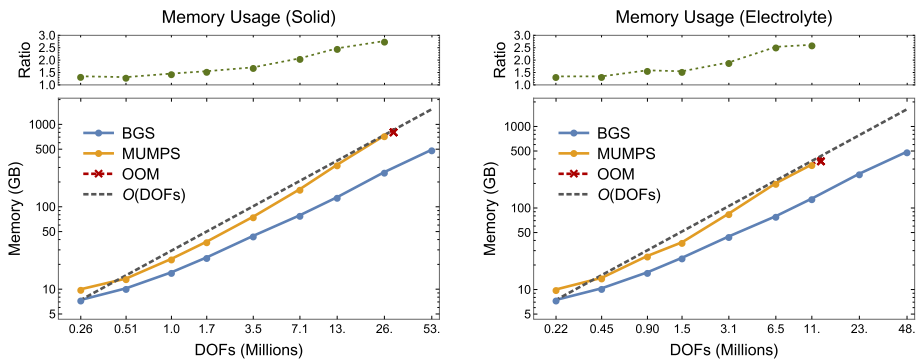


Fig. 5 The bottom set of plots are a comparison of the peak memory usage used by each solver method: MUMPS or BGS. This data was collected using the Python package `memory_profiler`. Additionally, the top set of plots show the ratio of MUMPS memory usage to BGS memory usage

smaller speedup in the electrolyte domain is expected because the system is more difficult because of the additional off-diagonal terms. It is expected that a better preconditioner using Schur complements could be constructed to increase the performance within the electrolyte domain. This Schur complement method is still in development.

The memory usage information is summarized in Figs. 5 and 6. On average, a solve using BGS takes about half the memory required by MUMPS; however, the ratio of memory required by MUMPS vs. BGS continues to increase as the number of DOFs increases. It is interesting to note that even though the anode domain generally has more DOFs, it requires less memory to solve. This is because the coefficient matrix for the electrolyte system contains more nonzeros resulting from the off-diagonal terms. This is why Test 8 ran out of memory while solving the electrolyte but succeeded in solving the anode. In short, the BGS method is much better suited for solving larger scale problems than MUMPS. This is not only because BGS is faster; it also uses less memory. From looking at the number of bytes per DOF in Fig. 6, the clear trend is that MUMPS becomes less efficient at larger problems sizes, while BGS become more efficient

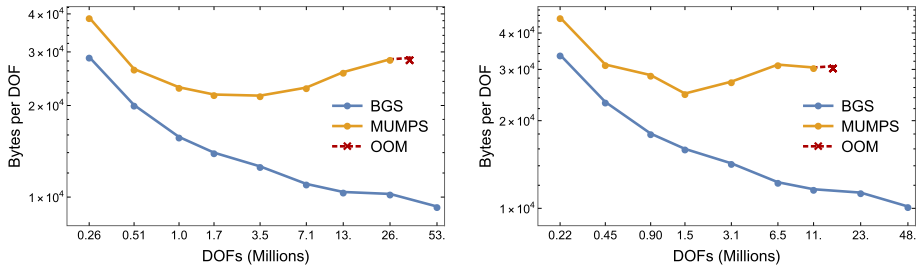
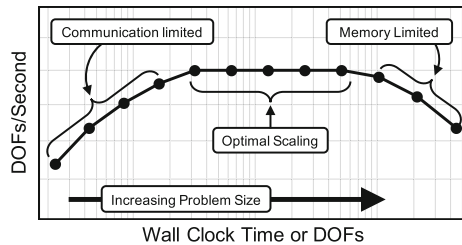


Fig. 6 Comparison of the bytes used per DOF for each solver method: MUMPS or BGS

Table 4 A summary of the various methods for measuring parallel performance

| Scaling | Type problem size | Number of processes |
|---------|-------------------|---------------------|
| Strong | Fixed | Increasing |
| Weak | Increasing | Increasing |
| Static | Increasing | Fixed |

Fig. 7 An example of ideal static scaling



5.2 Parallel Scaling

In the previous section, it was shown that the BGS method is more computationally efficient than the direct solver, MUMPS; however these tests were performed on a single node with 16 processes. This section examines the BGS method’s parallel scalability. Traditional methods for testing an algorithm’s parallel performance involve strong and weak scaling studies. In a strong scaling study, the problem size is held constant and the number of processes increases. A weak scaling study attempts to keep the ratio of problem size and processes constant by increasing both DOFs and processes at the same rate using a DOF per process constraint. A third option, however, holds the number of processes constant while increasing the problem size. This method is known as static scaling and is used to assess the ideal problem size for a given parallel framework. A summary of all the scaling metrics is found in Table 4.

An example of ideal static scaling is found in Fig. 7. In this figure, there are three dominant regions: communication limited, optimal scaling, memory limited. In the communication limited region, the problem size is very small compared to the number of processes. This results in the solve slowing down because of the extensive communication overhead. The memory limited region results from the problem size growing out so large that the number of processes is insufficient, resulting in a slowdown because of transferring information in memory from suboptimal locations. In the final region, the problem size is a perfect match for the number of processes, which minimizes solve time and results in optimal scaling. More about static scaling can be found in [6,9,26,31].

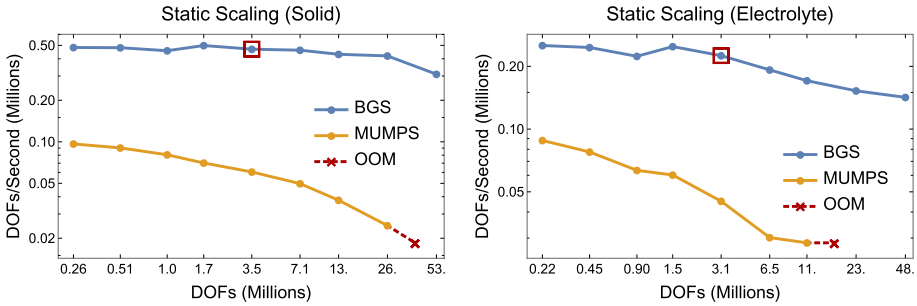


Fig. 8 Static scaling for both MUMPS and BGS. All simulations use the 7-particle domain on a single node with 16 processors. The boxed point is the problem chosen in the strong scaling study in Fig. 9

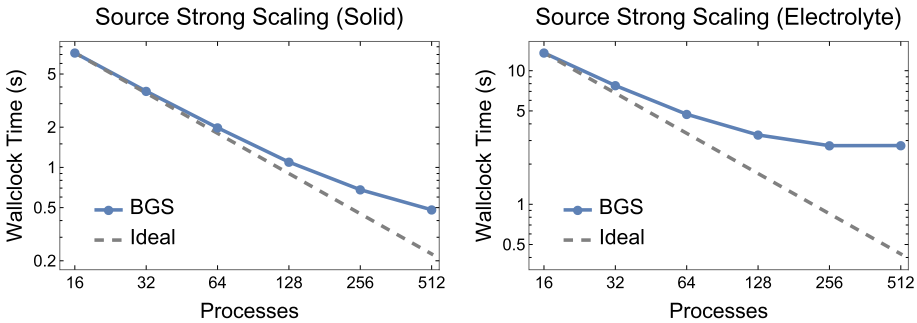


Fig. 9 Strong scaling plot generated using the 3.52M DOF mesh for the electrode domain and the 3.14M DOF mesh for the electrolyte domain and 16 processes per node. These meshes correspond to the highlighted rows in Tables 2 and 3

The static scaling result for BGS and MUMPS on the system in (16)–(17) is found in Fig. 8. As expected, this figure shows that the BGS method can process more DOFs/s than MUMPS, which is only a different way of presenting the information from Fig. 4. Additionally, for both solution methods the anode problem is more efficient, which is expected because of the difficulty in solving the electrolyte problem. The main point of this figure is to show that the BGS method remains within the optimal scaling region longer than MUMPS. This means the BGS method should exhibit good scaling performance.

The strong scaling study is found in Fig. 9. The initial mesh used for this set of experiments comes from the gray rows in Tables 2 and 3, which also corresponds to the boxed points in Fig. 8. This series of tests used 16 processes per node and double the number of nodes with each successive test. Using BGS to solve the anode problem results in nearly ideal scaling; however, the solve within the electrolyte domain starts to see diminishing returns after using 4 nodes (64 processes).

One possible cause for the suboptimal scaling within the electrolyte is an increase of Krylov iterations required to solve the problem as the number of MPI processes increases. As shown in Fig. 10, however, the average number of Krylov iterations remains relatively constant. Another possible cause has to do with the sparsity patterns of the two domains. Unlike in the solid domains, the electrolyte problem contains coupling on more than just the interface boundary. This could lead to a degradation in parallel performance because sparse matrix-vector multiplications have very low arithmetic intensities and can present as serial bottlenecks at the memory levels of each server node (see [9] for further discussion).

Fig. 10 The average number of Krylov iterations required to solve each domain for the specified number of nodes

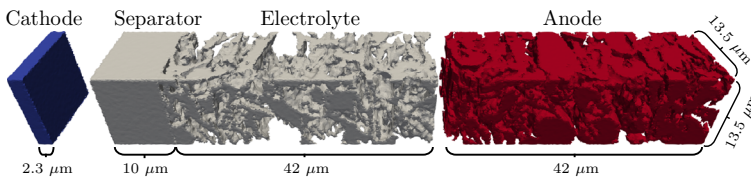
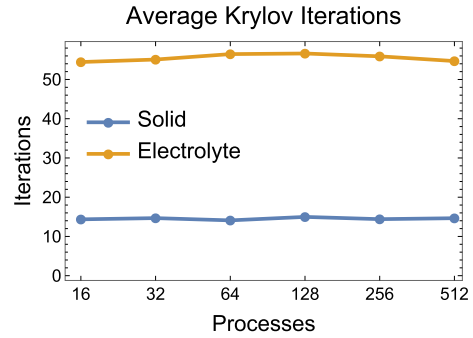


Fig. 11 The tomography-based mesh generated from a sample of SLC1506T Round 2 electrode used for testing electrolyte depletion. This electrode has a porosity of 37.4%

Currently, the inverse of each block, $\mathbf{J}_{\phi\phi}$ and \mathbf{J}_{cc} , is approximated using algebraic multigrid; however, a better approximation to \mathbf{J}_{cc}^{-1} could be constructed using the Schur complement. This is the topic of future research.

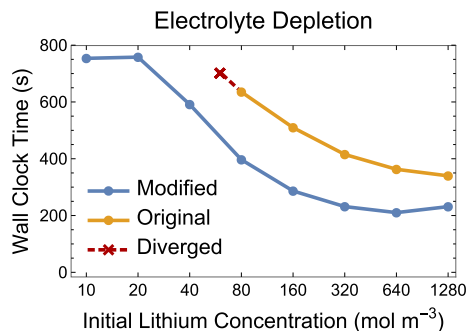
5.3 Electrolyte Depletion

The final experiment involves the modification discussed in Sect. 2.7. When charging at high C-rates, real Li-ion batteries can experience electrolyte depletion, which makes solving the electrolyte problem significantly more difficult. The mesh used to test electrolyte depletion was generated from a real SLC1506T Round 2 Electrode fabricated by the Cell Analysis, Modeling and Prototyping facility at Argonne National Laboratory and imaged by University College London with nano-CT, which is shown in Fig. 11. The complex geometry of this mesh will make diffusing Li^+ through the electrolyte more difficult. Further, to artificially induce electrolyte depletion, several simulations are performed by decreasing the initial concentration of Li^+ within the electrolyte.

Figure 12 shows the performance of BGS with the original and modified version of the system in Eqs. (6)–(17). Because electrolyte depletion occurs as the concentration gradient develops over time within the electrode, the time required to increase the SOC of the anode by 0.003555 was used instead of the time to perform an individual BGS solve. This took about 5 full time steps or about 2.88 simulated minutes. Including all five time steps incorporates the effects of electrolyte depletion on the Newton and consistency iterations as well as the linear solve.

In general, as the initial Li^+ concentration decreases, the time to solution increases, which is expected; however, without the modification, BGS begins to diverge as the concentration decreases, to the point where the simulation crashes with an initial concentration of 40 mol m^{-3} . The modified version was able to simulate down to 10 mol m^{-3} . In addition to this, the modified version is generally faster. This has to do with the fact that the unmodified case is

Fig. 12 The effect of electrolyte depletion on solve time. This data was collected by summing the time to completion of the first five time steps. The modified system refers to the modification discussed in Sect. 2.7. In a typical Li-ion battery, the electrolyte concentration is around 1200 mol m^{-3}



closer to singular when concentration approaches zero, which requires more linear iterations to get to the same level of accuracy as the modified case. On average, the modified case was 1.7 times faster than the unmodified problem.

6 Conclusion

In this paper, we presented a segregated-domain scheme to solve the electrochemistry of the Li-ion battery with FEniCS and a block preconditioning approach that significantly accelerates the linear solver. By simply leveraging the state-of-the-art preconditioned Krylov solvers available in the PETSc library, we achieved a nearly 6 times speedup over traditionally used direct solvers, and we expect this speedup to increase when we examine larger domains. Further, our block solver approach is scalable both in the algorithmic and parallel sense. Additionally, with a small modification to the off-diagonal term within the electrolyte, the solver was able to handle simulations with significantly smaller Li^+ concentrations. Continuing research will include the development of a better Schur-based block preconditioner for the electrolyte to overcome the issue of off-diagonal dominance associated with electrolyte depletion.

Acknowledgements This work was authored by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by the U.S. Department of Energy Office of Vehicle Technology Computer-Aided Engineering of Batteries (CAEBAT) program, program manager Brian Cunningham. A portion of the research was performed using computational resources sponsored by the Department of Energy's Office of Energy Efficiency and Renewable Energy and located at the National Renewable Energy Laboratory. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

Data Availability Statement The data used to support the findings of this study are available from the corresponding author upon request.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Alnæs, M.S., Blechta, J., Hake, J., Johansson, A., Kehlet, B., Logg, A., Richardson, C., Ring, J., Rognes, M.E., Wells, G.N.: The fenics project version 1.5. Arch. Numer. Softw. (2015). <https://doi.org/10.11588/ans.2015.100.20553>
2. Amestoy, P.R., Duff, I.S., L'Excellent, J.Y., Koster, J.: Mumps: a general purpose distributed memory sparse solver. In: Sørveik, T., Manne, F., Gebremedhin, A.H., Moe, R. (eds.) Applied Parallel Computing. New Paradigms for HPC in Industry and Academia, pp. 121–130. Springer, Berlin (2001)
3. Balay, S., Abhyankar, S., Adams, M.F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Eijkhout, V., Gropp, W.D., Kaushik, D., Knepley, M.G., May, D.A., McInnes, L.C., Mills, R.T., Munson, T., Rupp, K., Sanan, P., Smith, B.F., Zampini, S., Zhang, H., Zhang, H.: PETSc Web page (2018)
4. Balay, S., Abhyankar, S., Adams, M.F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Eijkhout, V., Gropp, W.D., Kaushik, D., Knepley, M.G., McInnes, L.C., Rupp, K., Smith, B.F., Zampini, S., Zhang, H., Zhang, H.: PETSc users manual. Tech. Rep. ANL-95/11—Revision 3.8, Argonne National Laboratory (2017)
5. Bunch, J.R., Hopcroft, J.E.: Triangular factorization and inversion by fast matrix multiplication. Math. Comput. **28**(125), 231–236 (1974)
6. Chang, J., Fabien, M.S., Knepley, M.G., Mills, R.T.: Comparative study of finite element methods using the Time-Accuracy-Size (TAS) spectrum analysis. SIAM J. Sci. Comput. (under revision) **40**, 137–163 (2018)
7. Chang, J., Karra, S., Nakshatrala, K.B.: Large-scale optimization-based non-negative computational framework for diffusion equations: parallel implementation and performance studies. J. Sci. Comput. **70**, 243–271 (2017)
8. Chang, J., Nakshatrala, K.B.: Variational inequality approach to enforce the non-negative constraint for advection-diffusion equations. Comput. Methods Appl. Mech. Eng. **320**, 287–334 (2017)
9. Chang, J., Nakshatrala, K.B., Knepley, M.G., Johnson, L.: A performance spectrum for parallel computational frameworks that solve PDEs. Concurr. Comput. Pract. Exp. **30**, e4401 (2018)
10. Colclasure, A.M., Dunlop, A.R., Trask, S.E., Polzin, B.J., Jansen, A.N., Smith, K.: Requirements for enabling extreme fast charging of high energy density Li-ion cells while avoiding lithium plating. J. Electrochem. Soc. **166**(8), A1412–A1424 (2019)
11. Coppersmith, D., Winograd, S.: Matrix multiplication via arithmetic progressions. J. Symb. Comput. **9**(3), 251–280 (1990)
12. Dalcin, L.D., Paz, R.R., Kler, P.A., Cosimo, A.: Parallel distributed computing using Python. Adv. Water Resour. **34**(9), 1124–1139 (2011). **(New computational methods and software tools)**
13. Danner, T., Singh, M., Hein, S., Kaiser, J., Hahn, H., Latz, A.: Thick electrodes for li-ion batteries: a model based analysis. J. Power Sources **334**, 191–201 (2016)
14. Dees, D., Gunen, E., Abraham, D., Jansen, A., Prakash, J.: Electrochemical modeling of lithium-ion positive electrodes during hybrid pulse power characterization tests. J. Electrochem. Soc. **155**(8), A603–A613 (2008)
15. Doyle, M., Fuller, T.F., Newman, J.: Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell. J. Electrochem. Soc. **140**(6), 1526–1533 (1993)
16. Doyle, M., Newman, J.: The use of mathematical modeling in the design of lithium/polymer battery systems. Electrochim. Acta **40**(13), 2191–2196 (1995)
17. Du, W., Xue, N., Shyy, W., Martins, J.R.R.A.: A surrogate-based multi-scale model for mass transport and electrochemical kinetics in lithium-ion battery electrodes. J. Electrochem. Soc. **161**(8), E3086–E3096 (2014)
18. Falgout, R.D., Yang, U.M.: HYPRE: a library of high performance preconditioners. In: International Conference on Computational Science. Springer, pp 632–641 (2002)
19. García, R.E., Chiang, Y.M., Craig Carter, W., Limthongkul, P., Bishop, C.M.: Microstructural modeling and design of rechargeable lithium-ion batteries. J. Electrochem. Soc. **152**(1), A255–A263 (2005)
20. Goldin, G.M., Colclasure, A.M., Wiedemann, A.H., Kee, R.J.: Three-dimensional particle-resolved models of li-ion batteries to assist the evaluation of empirical parameters in one-dimensional models. Electrochim. Acta **64**, 118–129 (2012)
21. Hein, S., Latz, A.: Influence of local lithium metal deposition in 3d microstructures on local and global behavior of lithium-ion batteries. Electrochim. Acta **201**, 354–365 (2016)
22. Henson, V.E., Yang, U.M.: Boomeramg: a parallel algebraic multigrid solver and preconditioner. Appl. Numer. Math. **41**(1), 155–177 (2002). [https://doi.org/10.1016/S0168-9274\(01\)00115-5](https://doi.org/10.1016/S0168-9274(01)00115-5). **(Developments and trends in iterative methods for large systems of equations—in memoriam Rudiger Weiss)**

23. Higa, K., Wu, S.L., Parkinson, D.Y., Fu, Y., Ferreira, S., Battaglia, V., Srinivasan, V.: Comparing macroscale and microscale simulations of porous battery electrodes. *J. Electrochem. Soc.* **164**(11), E3473–E3488 (2017)
24. Holzer, L., Münch, B., Iwanschitz, B., Cantoni, M., Hocker, T., Graule, T.: Quantitative relationships between composition, particle size, triple phase boundary length and surface area in nickel-cermet anodes for solid oxide fuel cells. *J. Power Sources* **196**(17), 7076–7089 (2011)
25. Hutzenlaub, T., Thiele, S., Paust, N., Spotnitz, R., Zengerle, R., Walchshofer, C.: Three-dimensional electrochemical Li-ion battery modelling featuring a focused ion-beam/scanning electron microscopy based three-phase reconstruction of a LiCO₂ cathode. *Electrochim. Acta* **115**, 131–139 (2014)
26. Joshaghani, M.S., Chang, J., Nakshatrala, K.B., Knepley, M.G.: Composable block solvers for the four-field double porosity/permeability model. *J. Comput. Phys.* **386**, 428–466 (2019)
27. Kashkooli, A.G., Farhad, S., Lee, D.U., Feng, K., Litster, S., Babu, S.K., Zhu, L., Chen, Z.: Multiscale modeling of lithium-ion battery electrodes based on nano-scale X-ray computed tomography. *J. Power Sources* **307**, 496–509 (2016)
28. Kim, G.H., Smith, K., Lee, K.J., Santhanagopalan, S., Pesarani, A.: Multi-domain modeling of lithium-ion batteries encompassing multi-physics in varied length scales. *J. Electrochem. Soc.* **158**(8), A955–A969 (2011)
29. Logg, A., Mardal, K.A., Wells, G.N., et al.: Automated Solution of Differential Equations by the Finite Element Method. Springer, Berlin (2012). <https://doi.org/10.1007/978-3-642-23099-8>
30. Malavé, V., Berger, J., Zhu, H., Kee, R.J.: A computational model of the mechanical behavior within reconstructed Li_xCoO₂ Li-ion battery cathode particles. *Electrochim. Acta* **130**, 707–717 (2014). <https://doi.org/10.1016/j.electacta.2014.03.113>
31. Mapakshi, N.K., Chang, J., Nakshatrala, K.B.: A scalable variational inequality approach for flow through porous media models with pressure-dependent viscosity. *J. Comput. Phys.* **359**, 137–163 (2018)
32. Mistry, A.N., Smith, K., Mukherjee, P.P.: Secondary-phase stochastics in lithium-ion battery electrodes. *ACS Appl. Mater. Interfaces* **10**(7), 6317–6326 (2018). <https://doi.org/10.1021/acsami.7b17771>
33. Newman, J., Thomas-Alyea, K.E.: *Electrochemical Systems*, 3rd edn. Wiley-Interscience, Hoboken (2004)
34. Newman, J., Tiedemann, W.: Porous-electrode theory with battery applications. *AIChE J.* **21**(1), 25–41 (1975)
35. Newman, J.S., Tobias, C.W.: Theoretical analysis of current distribution in porous electrodes. *J. Electrochem. Soc.* **109**(12), 1183–1191 (1962)
36. Plett, G.L.: *Battery Management Systems, Volume I: Battery Modeling*. Artech House Publishers, Norwood (2015)
37. Qianqian, F., Boas, D.A.: Tetrahedral mesh generation from volumetric binary and grayscale images. In: 2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro, pp. 1142–1145 (2009). <https://doi.org/10.1109/ISBI.2009.5193259>
38. Roberts, S.A., Mendoza, H., Brunini, V.E., Trembacki, B.L., Noble, D.R., Grillet, A.M.: Insights into lithium-ion battery degradation and safety mechanisms from mesoscale simulations using experimentally reconstructed mesostructures. *J. Electrochem. Energy Convers. Storage* **13**(3), 031005-1 (2016). <https://doi.org/10.1115/1.4034410>
39. Saad, Y., Schultz, M.H.: GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.* **7**(3), 856–869 (1986)
40. Smith, K., Wang, C.Y.: Power and thermal characterization of a lithium-ion battery pack for hybrid-electric vehicles. *J. Power Sources* **160**(1), 662–673 (2006)
41. Smith, K.A., Rahn, C.D., Wang, C.: Model-based electrochemical estimation of lithium-ion batteries. In: 2008 IEEE International Conference on Control Applications, pp. 714–719 (2008). <https://doi.org/10.1109/CCA.2008.4629589>
42. Smith, K.A., Rahn, C.D., Wang, C.Y.: Control oriented 1d electrochemical model of lithium ion battery. *Energy Convers. Manag.* **48**(9), 2565–2578 (2007)
43. Stephenson, D.E., Hartman, E.M., Harb, J.N., Wheeler, D.R.: Modeling of particle-particle interactions in porous cathodes for lithium-ion batteries. *J. Electrochem. Soc.* **154**(12), A1146–A1155 (2007)
44. Strassen, V.: Gaussian elimination is not optimal. *Numer. Math.* **13**(4), 354–356 (1969)
45. Trefethen, L.N., Bau, D., for Industrial, S., Mathematics, A.: *Numerical linear algebra*. Society for Industrial and Applied Mathematics, Philadelphia (1997)
46. Usseglio-Viretta, F.L.E., Colclasure, A., Mistry, A.N., Claver, K.P.Y., Pouraghajan, F., Finegan, D.P., Heenan, T.M.M., Abraham, D., Mukherjee, P.P., Wheeler, D., Shearing, P., Cooper, S.J., Smith, K.: Resolving the discrepancy in tortuosity factor estimation for li-ion battery electrodes through micro-macro modeling and experiment. *J. Electrochem. Soc.* **165**(14), A3403–A3426 (2018)

47. Usseglio-Viretta, F.L.E., Smith, K.: Quantitative microstructure characterization of a NMC electrode. *ECS Trans.* **77**(11), 1095–1118 (2017)
48. Wiedemann, A.H., Goldin, G.M., Barnett, S.A., Zhu, H., Kee, R.J.: Effects of three-dimensional cathode microstructure on the performance of lithium-ion battery cathodes. *Electrochim. Acta* **88**, 580–588 (2013)
49. Yan, B., Lim, C., Yin, L., Zhu, L.: Three dimensional simulation of galvanostatic discharge of LiCoO₂ cathode based on X-ray nano-CT images. *J. Electrochem. Soc.* **159**(10), A1604–A1614 (2012)
50. Zielke, L., Hutzenlaub, T., Wheeler, D.R., Chao, C.W., Manke, I., Hilger, A., Paust, N., Zengerle, R., Thiele, S.: Three-phase multiscale modeling of a LiCoO₂ cathode: combining the advantages of FIB-SEM imaging and X-ray tomography. *Adv. Energy Mater.* **5**(5), 1401612 (2015)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.