Numerical Investigation of Cahn–Hilliard-Type Phase-Field Models for Battery Active Particles

Zur Erlangung des akademischen Grades eines

DOKTORS DER NATURWISSENSCHAFTEN

von der KIT-Fakultät für Mathematik des
Karlsruher Instituts für Technologie (KIT)
genehmigte

DISSERTATION

von

Giuseppe Fabian Castelli

Tag der mündlichen Prüfung: 20.01.2021

1. Referent: Prof. Dr. Willy Dörfler

2. Referent: Prof. Dr. Christian Wieners

Danksagung

Die vorliegende Dissertation ist im Rahmen meiner Tätigkeit am Institut für Angewandte und Numerische Mathematik (IANM) am Karlsruher Institut für Technologie (KIT) in den Jahren 2016 bis 2021 entstanden. An dieser Stelle möchte ich mich bei all denjenigen bedanken, die mich auf diesem Weg unterstützt und begleitet haben.

Zuallererst möchte ich meinem Lehrer und Mentor **Prof. Dr. Willy Dörfler** danken. Ohne sein Engagement, seine Förderung und seine Geduld würde es diese Arbeit zweifelsohne nicht geben. Stets in guter Erinnerung behalten werde ich unsere gemeinsamen Konferenzteilnahmen sowie die intensiven Schreibphasen für unsere Artikel bis Mitternacht und darüber hinaus. Für die unzähligen fachlichen sowie nicht-fachlichen Gespräche zu jeder Tages- und Nachtzeit, für das Teilen seiner Erfahrungen, für die großen Freiheiten und für seine Förderung gilt ihm mein besonderer Dank!

Auch **Prof. Dr. Christian Wieners** möchte ich für seine Betreuung während meiner Arbeit und die hilfreichen Kommentare danken. Auf gemeinsam besuchten Workshops der GAMM Fachausschüsse mit geselligen Konferenz-Dinners habe ich durch ihn den interdisziplinären Austausch als einen ganz bemerkenswerten Aspekt der wissenschaftlichen Arbeit kennenlernen dürfen.

Dank gebührt natürlich ebenfalls meinen vielen aktuellen und ehemaligen Kolleginnen und Kollegen am KIT, welche mich begleitet haben und mir den Arbeitsalltag verschönert haben. Mit vielen verbindet mich mittlerweile eine enge Freundschaft.

Für eine der schönsten Konferenzerlebnisse danke ich ganz herzlich **Dr. Jan Bohn**. Meinem Kollegen **Zoltan Veszelka** gebührt großer Dank für die schöne Zeit und stets angenehme Atmosphäre in unserem gemeinsamen Büro. Ebenso großer Dank für die gemeinsame Zeit sowie den epischen Kicker-Duellen gilt **Niklas Baumgarten**, **Dr. Stefan Findeisen**, **Dr. Christine Grathwohl**, **Pascal Kraft**, **Dr. Markus Maier**, **Stephan Simonis**, **Benny Stein**, **Maria Sukhova**, **Dr. Daniel Weiß** und vielen weiteren. Dazu zählt **Raphael Schoof**, dem mein Dank auch für sein Engagement gilt, die entstandene Software dieser Arbeit nachhaltig weiterzuentwickeln. Besonders danken möchte ich **Henriette Schreiber-Schmoeger** und **Dr. Christoph Schmoeger** für ihre ganz herzliche Art sowie die kulinarischen Ausflüge in den Mittagspausen.

Als Doktorand des interdisziplinären Graduiertenkollegs SiMET möchte ich mich für den wissenschaftlichen Input aller Mitglieder, insbesondere von **Prof. Dr. Marc Kamlah** und **Prof. Dr. Arnulf Latz**, bedanken. Ebenso bedanke ich mich für die schöne Zeit mit allen Kollegiatinnen und Kollegiaten und den entstandenen Freundschaften. Hier sind vor allem **Dr. Roland Balbierer**, **Lars von Kolzenberg**, **Dr. Dieter Oehler**, **Dr. Philipp Seegert** und **Zoltan Veszelka** zu nennen. Ganz besonders bedanken möchte ich mich an dieser Stelle bei **Lars von Kolzenberg** für die langen Diskussionen über die Partikel-Modellierung und die freundschaftliche Zusammenarbeit.

Über SiMET hinaus danke ich **Dr. Daniel Schneider** und **Dr. Tao Zhang** für die Diskussionen über die Phasenfeldmodellierung.

Auf meinen besuchten Konferenzen habe ich viele Menschen kennenlernen dürfen, die mit ihrem Wissen und ihren Erfahrungen zum Gelingen dieser Arbeit beigetragen haben, wofür ich mir insbesondere bei **Prof. Dr. Timo Heister**, **Dr.-Ing. Uwe Koecher** und **Dr. Mirjam Walloth** ganz besonders bedanke.

Abseits der akademischen Welt haben meine Freunde meine Zeit als Doktorand ebenfalls wesentlich beeinflusst. Für die langanhaltenden Freundschaften und vielen schönen gemeinsamen Erinnerungen danke ich ganz herzlich **Sina Busch**, **Johanna** und **Sebastian Butzek**, **Hannah** und **Stefan Muth**, **Laura** und **Michael Ofer**, **Dr. Michael Reichenbacher**, **Christian Schellenberger** sowie **Christoph** und **Dr. Dominique Straub**.

In den vergangenen Jahren hatte ich das Glück mit meiner Tante **Carmelina Picone** und meinen Freunden **Silke Kühnrich**, **Stefanie Sauer** und **Uwe Diedrichs** denkwürdige Skiurlaube verbringen zu dürfen. Die wunderschöne Auszeit von der Arbeit mit kulinarischen Highlights, rasanten Abfahrten und schwindelerregenden Gondoletto-Fahrten werde ich nie vergessen.

Ganz herzlich möchte ich auch **Daniela Flörchinger** danken. Sie hat mich mit ihrer wundervollen Art in der abschließenden Phase meiner Promotion begleitet.

Mein tiefster Dank gebührt meinen Eltern **Alfredo** und **Antonia Castelli**. Ohne ihren Rückhalt würde diese Arbeit wahrlich nicht existieren. Sie gaben mir Kraft in dunklen Zeiten und haben mich in jeder Lebenslage ertragen, ermutigt, motiviert, unterstützt und sind einfach immer für mich da — Danke, Mama und Papa!

Fabian Castelli
Karlsruhe, im Dezember 2021

Abstract

We focus on the efficient numerical simulation of Cahn–Hilliard-type phase-field models for phase separation and large deformations in electrode particles of lithium-ion batteries. The main contribution of this thesis is the development and implementation of a space and time adaptive finite element solver for these Cahn–Hilliard-type models. In particular, for the pure phase separation problem we propose a preconditioner, which enables the matrix-free simulation.

In the most general form, accounting for anisotropic material properties, we start with the review and discussion of the phase-field model equations for phase separation in electrode particles during lithium insertion. The resulting initial boundary value problem is based on the fourth-order, nonlinear, time-dependent Cahn–Hilliard equation.

For the numerical simulation of the phase separation model we develop an adaptive matrix-free finite element solver. Thereby we employ an isoparametric C^0-continuous Lagrangian finite element method for the mixed formulation of the Cahn–Hilliard equation. We combine an error controlled variable-step, variable-order algorithm based on linear multistep methods for the time integration and a gradient recovery estimator for the adaptive mesh refinement in a space and time adaptive solution algorithm. The nonlinear systems in each time step are linearized with the Newton–Raphson method and solved with a preconditioned GMRES method. Especially for the matrix-free implementation we propose a block preconditioner respecting the nonlinearity of the Cahn–Hilliard model equation.

With the method of manufactured solutions we show the numerical efficiency and optimal order convergence of the matrix-free finite element solver. Based on the example of spinodal decomposition we demonstrate the capabilities of our solver and the mesh-independent preconditioning with our preconditioner for the Cahn–Hilliard equation with degenerated mobility.

We perform numerical experiments for phase separation in electrode particles of lithium iron phosphate under different symmetry assumptions. In the spherical symmetric case, we show the importance of adaptive methods and confirm the mesh-independent preconditioning, even for locally refined meshes. Based on an azimuthal symmetry assumption we observe phase transition morphologies for isotropic and anisotropic material properties different to the core-shell scenario.

Finally, we couple the phase separation model with a thermodynamically consistent developed mechanical theory for large deformations in a common Cahn–Hilliard-type phase-field model. For the multi-physical problem we present a finite element discretization and extend our space and time adaptive algorithm for the matrix-based solution. With numerical simulations of lithium iron phosphate electrode particles we investigate physical and numerical aspects, like the influence of the interfacial energy coefficient on the arising stress and the superior efficiency over classical implementations.

Contents

1. Introduction **1**

2. Fundamentals **5**

3. A Phase-Field Model for Phase Separation **9**
3.1. Working Principle of Lithium Ion Batteries 9
3.2. Model Equations for Phase Separation 11
3.2.1. Phase-Field Model 12
3.2.2. Boundary and Initial Conditions 15
3.2.3. Problem Statement 16
3.2.4. Alternative Phase Separation Models 16
3.3. Mathematical Issues 17

4. The Numerical Solution Algorithm **21**
4.1. The Space and Time Discretization 23
4.1.1. Discretization in Space 24
4.1.2. Discretization in Time 25
4.2. Solving the Discrete Problem 27
4.2.1. Transformation 27
4.2.2. Linearization 28
4.2.3. Linear Solver 29
4.3. The Space and Time Adaptive Solution Algorithm 34

5. Nonlinear Matrix-Free Finite Element Operator Application **39**
5.1. The Matrix-Free Framework of the deal.II Library 39
5.1.1. Motivation 39
5.1.2. Cell-Based Finite Element Operator Application 41
5.1.3. Literature Review 43
5.2. A Nonlinear Problem Solved by the Matrix-Free Method 43
5.2.1. Problem Description 44
5.2.2. Discretization with Finite Elements 44
5.2.3. Matrix-Free Numerical Linear Algebra 44
5.2.4. Numerical Results 45
5.2.5. Conclusion 47

6. Numerical Experiments **49**
6.1. Classical Cahn–Hilliard Equation 50
6.1.1. Validation with the Method of Manufactured Solutions 50
6.1.2. Study on Solver Components based on Spinodal Decomposition . 53

6.2. Phase Separation in Electrode Particles 66
6.2.1. Spherical Symmetric Solution Approach 67
6.2.2. Azimuthal Symmetric Solution Approach 73

7. Chemical-Mechanical Coupling **79**
7.1. Introduction 79
7.2. Model Development 80
7.2.1. Deformation 80
7.2.2. Phase-Field Model 81
7.3. Numerical Treatment 84
7.3.1. Mathematical Problem Formulation 84
7.3.2. Numerical Solution Method 85
7.4. Numerical Experiments 87
7.4.1. Simulation Setup 87
7.4.2. Chemomechanics 88
7.4.3. Numerical Efficiency 93
7.5. Summary and Conclusion 98

8. Conclusion **101**

A. Model Parameters **105**
A.1. Material Parameters 105
A.2. Normalization 105

B. Spherical Coordinates **107**
B.1. Coordinate Transformation 107
B.2. Calculus in Spherical Coordinates 108
B.2.1. Spherical Symmetry 108
B.2.2. Azimuthal Symmetry 109

Bibliography **111**

1. Introduction

In the past decades lithium-ion batteries (LIBs) have become a dominating energy storage technology. Their applications range from everyday mobile power devices such as laptops or cell phones to electrical vehicles and the stabilization of power grids [6, 139, 140]. Currently being the state-of-the-art technology for electrical energy storage systems, lithium-ion batteries outperformed other available battery systems already twenty years ago due to their high energy density and flexible design [140]. Nowadays the research still focuses on the improvement of the battery lifetime and performance [108].

In its generality the term *lithium-ion battery* covers the whole range of batteries where the exchanging species are lithium ions [84, Sect. 1.1]. For example the choice of the electrode materials influence the battery performance and is an active research topic [109, 119]. A particular category of cathode materials are phase separating materials like for example lithium iron phosphate Li_xFePO_4 (LFP) [59, 134, 162, 169, 170] or lithium manganese oxide $Li_xMn_2O_4$ (LMO) [85, 124, 149, 167, 169]. These materials allow the coexistence of two separated phases of low and rich lithium concentration during lithium insertion (*lithiation*) or extraction (*delithiation*) in single electrode particles and thus influence the overall battery performance. In particular, LFP for example shows strongly anisotropic material properties [134].

Especially coupled to large deformations the concentration gradients induced by phase transformations cause large stresses and therefore induce particle degradation due to fracture [170]. In order to predict the evolution of phase transitions and the arising stresses in electrode particles several mathematical models have been used. The models for phase separation are mainly divided into two classes, the sharp-interface models such as the core-shell scenario [57, 135, 165] and diffusive interface models represented by phase-field models [77, 162, 170]. However, an experimental study [101] revealed that the core-shell scenario cannot capture the structure in strongly anisotropic materials like LFP [56]. The range of mechanical models is much wider, for example, we differ between linear [85, 113, 171] and nonlinear [59, 149, 170, 173] solid mechanics theories, whereas this obviously covers only a small range of models.

In this thesis we follow the phase-field model for phase separation in electrode particles during lithium insertion presented in [84], which can be traced back to Han et al. [77] and is currently a widely used approach, see [59, 84, 124, 150, 155, 162, 163, 170, 173] among others. Later we will also couple a thermodynamically consistent theory for large deformations.

The resulting model equation for phase separation is know as the Cahn–Hilliard equation [37, 38], which is a time-dependent, fourth-order partial differential equation (PDE). Among phase-field models the Cahn–Hilliard equation is probably the most prominent one, which was developed to model phase separation in binary alloys [32]. In the past the Cahn–Hilliard equation found its application for example in two-phase flow [9, 99], wound healing and tumor growth [69, 92, 114], image inpainting [27], multicomponent systems [28, 65] and many other applications.

Hence also from the mathematical point of view the Cahn–Hilliard equation is widely studied. Global existence and uniqueness of a solution was proven for different versions of the Cahn–Hilliard equation, with constant or concentration dependent mobility, with quartic or logarithmic free energy density, see for example [62, 65, 66, 142]. For the numerical solution a vast literature is available covering spatial and temporal discretization schemes, solvers and preconditioners, adaptive algorithms, see for example [31, 32, 145, 154, 164] and the reference cited therein.

The investigation of phase transforming electrode materials for batteries is primarily carried out in the engineering literature. There, numerical experiments were performed for example with commercial software packages like COMSOL [84, 149, 155, 163, 169], ABAQUS [59], FEAP [173, 174], open source codes like FEniCS [150], MOOSE [170] or even more sophisticated implementations [51, 80].

Challenges Large effort was spent to develop numerical methods for the Cahn–Hilliard equation, compare [31] and the references cited therein. The main difference to the battery application lies in the boundary conditions. Numerical methods mostly have been developed for the Cahn–Hilliard equation with homogeneous Neumann boundary conditions (no-flux), which we denote in the following as the *classical* case. However, the lithiation of electrode particles is modeled with inhomogeneous Neumann boundary conditions. In the simplest case these are constant, uniform fluxes [84, 170] but can also be replaced by nonlinear Neumann conditions such as the Butler–Volmer kinetics for surface reactions [59,162]. However, the conservation of mass is violated and thus also the monotonous energy decay, which is an important element in the proof of global existence and uniqueness of a solution for the classical Cahn–Hilliard equation. Furthermore, the numerical methods have to be reviewed. For example methods inheriting the monotone energy decay [74] may fail.

Integrating later on also the mechanics, the simulation of such complex multi-physical problems is computationally demanding. Main challenges of the Cahn–Hilliard-type model equations are the strongly varying spatial and temporal scales during lithiation. Like in the classical case, phase separation takes place on a much faster time scale than coarsening and migration of phase transitions. Thus, during lithiation the time scale varies over several orders of magnitudes. Additionally, crucial parameters for the interfacial energy generate an almost sharp phase transition in space. Moreover, the almost sharp phase transitions between lithium poor and rich phases travel through the whole particle domain, which thus need a fine mesh resolution. In sum, when using an implicit time integration method together with a fully resolved mesh, a big number of large, badly conditioned linear systems need to be solved, typically with iterative methods. Hence, also preconditioning the Cahn–Hilliard equation becomes important, which is already a research topic itself, see [8,28,29, 31,32].

With its numerical challenges, the accurate and efficient simulation of Cahn–Hilliard-type phase-field models for phase transforming electrode materials is often limited to spheroidal particles under symmetry assumptions [170, 173] or with adapted model parameters [150, 155, 173]. Due to its high computational costs the embedding of phase separation in multiscale models is only possible with efficient numerical methods, as likewise done in [50]. On the level of parallel high performance computing the limiting factor of linear algebra operations might also be due to the access to main memory. Especially, the

open source finite element library *deal.II* [5,13] provides a matrix-free framework [96,98], which efficiently implements the finite element operator application without storing matrices. It is worth noting, that the Cahn–Hilliard equation was already solved with the functionalities of the matrix-free framework in [97,99]. Although the internal dependency on the matrix-based AMG preconditioner of the used block preconditioner may be resolved, only a constant mobility was considered. To overcome all these difficulties and workarounds leads to the research objective of this thesis.

Aim of the Thesis The main contribution of this thesis is the development and implementation of an efficient, matrix-free, space and time adaptive numerical solution algorithm for Cahn–Hilliard-type phase-field models in battery active particles under lithium insertion.

Methods As governing model equations for phase separation we consider the mixed formulation of the Cahn–Hilliard equation resulting in a set of two coupled second-order partial differential equations.

For the discretization of the mixed formulation of the Cahn–Hilliard equation [63] we employ the C^0-continuous Lagrangian finite element method of order $p \geq 1$ in space and a numerical differentiation formula (NDF) linear multistep method family from order one to five [129] in time. In an error controlled variable-step, variable-order time integration scheme [129, 130, 143] we incorporate an adaptive mesh refinement criterion based on a gradient recovery estimator [3, Chap. 4] measuring the regularity of the solution. In combination we obtain a space and time adaptive solution algorithm.

The arising nonlinear systems in each time step are linearized with the Newton–Raphson method. The linear system are solved with a preconditioned GMRES method. Hereby at the heart of our solver we propose an extended preconditioner for the mixed formulation of the Cahn–Hilliard equation, based on [31], respecting the nonlinearity of the model equations. In particular, this preconditioner is suited for matrix-free computations. Thus the Cahn–Hilliard solver is implemented in a completely matrix-free way, avoiding the storage of matrices at each level of the algorithm.

For the later incorporation of the chemical-mechanical coupling we extend this numerical solution algorithm in a straightforward way, but solve the linearized systems of the coupled problem with the LU-decomposition [52] instead.

We focused on a modular and flexible design of the solver and its components, which we implemented in `C++` based on the functionalities of the *deal.II* finite element library [5, 13, 98].

Significance The developed numerical solution algorithm for the phase-field model equations enables efficient simulations of phase separation in electrode particles during lithium insertion beyond the assumption of spherical symmetry in crucial parameter ranges. In particular, we recall the phase-field model in the most general form capable for anisotropic diffusion and anisotropic interfacial energy penalties. For the efficient numerical solution we derive an extended, matrix-free applicable preconditioner for the mixed formulation of the Cahn–Hilliard equation. The preconditioner shows optimal, mesh-independent behavior for classical Cahn–Hilliard benchmark problems and especially for the model

equations of electrode particles under lithium insertion. Additionally, the skeleton of the developed space and time adaptive solution algorithm is written in a general purpose fashion with the ability to work under some modifications also for future applications. Moreover, since identical models are also employed for the investigation of other electrode materials, like sodium iron phosphate Na_xFePO_4 (NFP) [168–170] or silicon (Si) [46, 163], new possible applications can already be seen here.

Outline The rest of this thesis is organized as follows: Next in Chapter 2 we define the basic notation used throughout this thesis. In Chapter 3 we review and discuss the Cahn–Hilliard-type phase-field model for phase separation in electrode particles of lithium-ion batteries. After this we present in Chapter 4 our numerical solution algorithm with the underlying space and time discretization, the extended preconditioner for the Cahn–Hilliard equation as well as the space and time adaptive solution algorithm. In Chapter 5 we briefly discuss the matrix-free implementation available through the *deal.II* library. Furthermore, we demonstrate the scalability of our nonlinear matrix-free finite element solver implementation as basic building block for our matrix-free Cahn–Hilliard solver. We validate our developed solver and demonstrate the efficiency in the first part of Chapter 6. In the second part of Chapter 6 we investigate with numerical experiments the performance of the solver for the model equations and the evolution of the phase transition morphology under different symmetry assumptions. Finally, we employ the generic space and time adaptive solution algorithm to the coupled chemical-mechanical phase-field model and investigate the interplay of phase separation and large deformations in Chapter 7. We conclude this thesis by summarizing the key findings in Chapter 8.

Prepublications As of the date of submission[1], parts of the results presented in this thesis are already published or submitted for publication. The whole numerical solution algorithm with underlying space and time discretization, the matrix-free preconditioner and the adaptive algorithm presented in Chapter 4 are based on our article [43], which is accepted for publication. The strong parallel scalability of our implementation of a nonlinear matrix-free solver in Chapter 5 was already presented in [41] in a slightly modified version. The numerical experiments on the solution algorithm and the preconditioner during lithium insertion into electrode particles under the spherical symmetry assumption in Chapter 6 are based on those from [43]. Finally, Chapter 7 is an adapted version of the current state of our submitted article [44] on the efficient simulation of phase separation and large deformations in electrode particles. Note, in this joint work the theoretical part, in particular the model development and the physical interpretation, was essentially contributed by Lars von Kolzenberg, who is supervised by Birger Horstmann and Arnulf Latz.

Acknowledgment The author gratefully acknowledges financial support by the German Research Foundation (DFG) through the Research Training Group 2218 SiMET – Simulation of mechano-electro-thermal processes in lithium-ion batteries, project number 281041241.

[1] 21.12.2020

2. Fundamentals

In the following we summarize the notation established in our previous works [41–44] based on [40, 107] and use it throughout this thesis.

Standard Notation We denote the *positive integers* by $\mathbb{N}$ and the *real numbers* by $\mathbb{R}$. The *spatial dimension* is $d \in \{1, 2, 3\}$. *Scalar values* are written as $s \in \mathbb{R}$ and *tensors* as $\mathbf{T} = (\mathbf{T}_{ij})_{ij} \in \mathbb{R}^{d,d}$. For a generic dimension $n \in \mathbb{N}$, we write *vectors* as $\boldsymbol{v} = (\boldsymbol{v}_i)_i \in \mathbb{R}^n$ and *matrices* as $M \in \mathbb{R}^{n,n}$. Scalar, vector, matrix and tensor-valued functions are written accordingly. For the *absolute value* as well as the *Euclidean norm* in $\mathbb{R}^n$ we write $|\cdot|$.

For two vectors $\boldsymbol{v}, \boldsymbol{w} \in \mathbb{R}^n$ the *dot product* $\boldsymbol{v} \cdot \boldsymbol{w} = \sum_{i=1}^n \boldsymbol{v}_i \boldsymbol{w}_i$ is the sum of the multiplied vector entries. For the full contraction of two tensors $\mathbf{S}, \mathbf{T} \in \mathbb{R}^{d,d}$ we use the *Frobenius inner product* $\mathbf{S} : \mathbf{T} = \sum_{i,j=1}^d \mathbf{S}_{ij} \mathbf{T}_{ij}$.

The *spatial domain* Ω is assumed to be a nonempty, open, connected and bounded subset of $\mathbb{R}^d$. The *closure* of Ω is denoted by $\overline{\Omega}$ and for the *boundary* we write $\partial\Omega$. The *outer unit normal vector* is denoted by $\boldsymbol{n}$.

Space and time-dependent functions are defined on the time-space cylinder $\mathbb{R}_{\geq 0} \times \overline{\Omega}$. The *time variable* is typically $t \in \mathbb{R}_{\geq 0}$ and the *spatial coordinate* is $\boldsymbol{x} \in \Omega$. Let $u \colon \mathbb{R}_{\geq 0} \times \overline{\Omega} \to \mathbb{R}$, $\boldsymbol{v} \colon \mathbb{R}_{\geq 0} \times \overline{\Omega} \to \mathbb{R}^d$ and $\mathbf{T} \colon \mathbb{R}_{\geq 0} \times \overline{\Omega} \to \mathbb{R}^{d,d}$ be generic smooth scalar, vector and tensor-valued functions, respectively. We denote the *partial derivatives* for each spatial direction $i \in \{1, \dots, d\}$ by $\partial_i u$. The *temporal derivative* is denoted by $\partial_t u$. The *gradient* of u is the vector of all partial derivatives $\nabla u = (\partial_i u)_i$. The *divergence* of $\boldsymbol{v}$ is denoted as the sum of the partial derivatives $\nabla \cdot \boldsymbol{v} = \sum_{i=1}^d = \partial_i \boldsymbol{v}_i$. The *Laplace operator* applied to u is defined as the sum of the second partial derivatives $\Delta u = \nabla \cdot \nabla u = \sum_{i=1}^d \partial_i^2 u$. The *gradient* of $\boldsymbol{v}$ is a matrix called the *Jacobian* $J_{\boldsymbol{v}} = \nabla \boldsymbol{v} = (\partial_j \boldsymbol{v}_i)_{ij}$. Finally, the *divergence* of $\mathbf{T}$ is defined as the vector $\nabla \cdot \mathbf{T} = \left(\sum_{j=1}^d \partial_j \mathbf{T}_{ij} \right)_i$.

We denote the *natural logarithm* by $\log \colon (0, \infty) \to \mathbb{R}$.

Function Spaces The space of $k \in \mathbb{N}_0$ times continuously differentiable functions is denoted by $C^k(\Omega)$. Functions in $C(\overline{\Omega})$ are continuous also on the boundary. For $p \in [1, \infty]$ we write $L^p(\Omega)$ for the scalar-valued *Lebesgue spaces*. The *Lebesgue norm* is $\|\cdot\|_{L^p}$. Especially for $p = 2$ the Lebesgue norm is induced by the L^2-inner product $\|\cdot\|_{L^2}^2 = (\cdot, \cdot)_{L^2}$ and when the context is clear we may omit the subscript L^2. For vector and tensor-valued functions we write $L^p(\Omega; \mathbb{R}^n)$ and $L^p(\Omega; \mathbb{R}^{d,d})$. The *Sobolev spaces* are denoted by $W^{m,p}(\Omega)$ and $W^{m,2}(\Omega) = H^m(\Omega)$. In particular, the dual space of $H^1(\Omega)$ is denoted by $H^{-1}(\Omega)$. The *Sobolev norm* is defined as $\|f\|_{W^{m,p}}^p = \sum_{|\alpha|_1 \leq m} \|D^\alpha f\|_{L^p}^p$ with the multi-index notation $\alpha \in \mathbb{N}_0^d$, $D^\alpha = D^{\alpha_1} \dots D^{\alpha_d}$. Thereby the H^1-norm reads as $\|f\|_{H^1}^2 = \|f\|_{L^2}^2 + \|\nabla f\|_{L^2}^2$.

We denote $(f, g) = \int_\Omega f\, g \,\mathrm{d}\boldsymbol{x}$ as L^2-inner product for two functions $f, g \in L^2(\Omega)$ as well as $(\boldsymbol{v}, \boldsymbol{w}) = \int_\Omega \boldsymbol{v} \cdot \boldsymbol{w} \,\mathrm{d}x$ for two vectors $\boldsymbol{v}, \boldsymbol{w} \in L^2(\Omega; \mathbb{R}^d)$ and $(\mathbf{S}, \mathbf{T}) = \int_\Omega \mathbf{S} : \mathbf{T} \,\mathrm{d}\boldsymbol{x}$

for two tensor-valued functions $\mathbf{S}, \mathbf{T} \in L^2(\Omega; \mathbb{R}^{d,d})$. We indicate boundary integrals with the subscript $\partial\Omega$.

Electrode Particle Modeling The geometry of electrode particles is encoded in the *particle domain* Ω. We denote the part of the boundary representing the *particle surface* by Γ_{ext}. *Artificial boundaries* due to symmetry assumptions are denoted by Γ_0 and some possible further subscript for the unique identification. The *volume* of the particle is denoted as $V_\Omega = |\Omega|$ and the *particle surface measure* as $S_\Omega = |\Gamma_{\text{ext}}|$. Furthermore, the *specific surface* is defined as $A_{\text{V}} = S_\Omega / V_\Omega$. For the *reference length scale* we use the symbol L_0.

The *lithium concentration* measured in $\mathrm{mol\,m^{-3}}$ is denoted by c. For the *normalized lithium concentration* we write $\overline{c} = c/c_{\text{max}}$, where c_{max} is the *maximum lithium concentration* of the host material. The *chemical potential* with unit $\mathrm{J\,mol^{-1}}$ is defined as the variational derivative of the *free energy* Ψ and denoted by the symbol μ. The free energy is the integral of *free energy density* ψ over the particle domain. In this thesis we account for *chemical* ψ_{ch}, *interfacial* ψ_{int} and *elastic* ψ_{el} contributions to the free energy. Especially, the *interfacial energy coefficient* is denoted in the isotropic version by κ and in the anisotropic case by the tensor $\mathbf{K}$. Similar, the *lithium flux* $\boldsymbol{N}$ implements the isotropic m, respective anisotropic $\mathbf{m}$, *mobility*. During phase separation the concentration profile establishes a lithium poor and a lithium rich phase with the *equilibrium concentrations* c_α, respective c_β. Starting from an *initial concentration* c_0, lithium insertion is modeled by an *applied lithium flux* N_{ext}. The intercalation takes place at the *operation temperature* T and for the universal gas constant we use the symbol $R = 8.314\,\mathrm{J\,mol^{-1}\,K^{-1}}$.

The overall amount of lithium inside an electrode particle is commonly measured by the *state of charge* (SOC), which is defined as the average of the normalized lithium concentration $\text{SOC} = 1/V_\Omega \int_\Omega \overline{c}\,\mathrm{d}\boldsymbol{x}$.

All physical quantities are stated in SI units. Except for the applied lithium flux N_{ext} we introduce a non-SI unit, the *C-rate*, which depends on the particle geometry and material. In our context, the C-rate defines the *reference cycle time* $t_{\text{cycle}} = 1/\text{C-rate}$, which it takes for completely filling an empty particle.

Taking large deformations into account we define the particle domain without any deformation Ω_0 as *reference configuration*. The deformed domain Ω is called the *current configuration*. The *deformation* of the reference domain is described by a differentiable, orientation-preserving diffeomorphism $\boldsymbol{\phi}\colon [0, t_{\text{f}}] \times \overline{\Omega}_0 \to \overline{\Omega}$ mapping any point $\boldsymbol{X}_0 \in \overline{\Omega}_0$ of the reference configuration at every time $t \in [0, t_{\text{f}}]$ onto the corresponding point $\boldsymbol{x} \in \overline{\Omega}$ in the current configuration.

With the *displacement* $\boldsymbol{u}\colon [0, t_{\text{f}}] \times \overline{\Omega}_0 \to \mathbb{R}^d$ the deformation can be expressed as $\boldsymbol{\phi}(t, \boldsymbol{X}_0) = \boldsymbol{X}_0 + \boldsymbol{u}(t, \boldsymbol{X}_0)$. Thus the *deformation gradient* $\mathbf{F}$ is a tensor, given by $\mathbf{F} = \mathbf{Id} + \nabla \boldsymbol{u}$, with the identity tensor $\mathbf{Id} \in \mathbb{R}^{d,d}$. Further, we denote the *elastic strain tensor* by $\mathbf{E}_{\text{el}}$ and the fourth-order *elasticity tensor* of the St. Venant–Kirchhoff model by $\mathbb{C}$, defined by $\mathbb{C}\mathbf{E}_{\text{el}} = 2G\mathbf{E}_{\text{el}} + \lambda\,\mathrm{tr}(\mathbf{E}_{\text{el}})\mathbf{Id}$ with the Lamé constants G, λ. For the *Cauchy stress tensor* we write σ and for the *first Piola–Kirchhoff stress tensor* $\mathbf{P}$.

Finite Elements and Time Discretization With Ω_h we denote the actual *computational domain* for finite element methods, which is a polygonal approximation to the particle domain Ω. For an admissible *mesh* of Ω_h we write $\mathcal{T}_h$. Motivated by the implementation with the finite element library *deal.II*, we assume the mesh $\mathcal{T}_h$ to be a decomposition

of the computational domain into intervals in 1D, quadrilaterals in 2D or hexahedra in 3D. The elements (1D: intervals, 2D: quadrilaterals, 3D: hexahedra) of a mesh are called *cells* $Q \in \mathcal{T}_h$. Each cell has a *diameter* denoted by $h_Q \coloneqq \operatorname{diam}(Q)$, whereby the *maximum mesh width* $h_{\max}$ is defined as the maximum cell diameter $h_{\max} \coloneqq \max_{Q \in \mathcal{T}_h} h_Q$. Simulations are performed until a *final simulation time* $t_\mathrm{f} > 0$. The *time interval* $[0, t_\mathrm{f}]$ is discretized into N_t *time steps* t_n with the *time step size* τ_n for $n = 1, \ldots, N_t$.

On a cell $Q \in \mathcal{T}_h$ we denote the set of *tensor product polynomials* of order $p \in \mathbb{N}$ by $\mathbb{Q}_p(Q)$. For a function space V we denote a *finite dimensional subspace* by $V_h \subset V$. The discrete approximation $u_h \in V_h$ of a continuous function $u \in V$ is indicated by the subscript h. In particular, for the polynomial degree $p \in \mathbb{N}$, we denote the *Lagrangian finite element space* by $\mathcal{S}^p(\mathcal{T}_h)$, which consists of continuous, piecewise p-th-order polynomial functions. The number of *degrees of freedom* (DOFs) is symbolized by N_x. The *basis functions* of the *nodal basis* $\Pi = \{\varphi_i \,:\, i = 1, \ldots, N_x\}$ satisfy for each DOF $\boldsymbol{x}_j$ the condition $\varphi_i(\boldsymbol{x}_j) = \delta_{i,j}$ with the Kronecker delta $\delta_{i,j}$. For the discrete equations we identify finite element functions $u_h \in V_h$ with its coefficient vectors $\boldsymbol{u} \in \mathbb{R}^{N_x}$ by the relation $u_h = \sum_{i=1}^{N_x} \boldsymbol{u}_i \varphi_i$.

3. A Phase-Field Model for Phase Separation

In this chapter we present the Cahn–Hilliard-type phase-field model equations for phase separation in electrode particles during lithium insertion. First, in Section 3.1 we give a brief review on the working principle of lithium-ion battery cells including the description of phase separation in electrode particles. Then in Section 3.2 we introduce the phase-field model equations and conditions. Thereby, we normalize the presented model and gather the equations and conditions in an initial boundary value problem for the further course of this work. Finally, in Section 3.3 we point out some differences to the classical Cahn–Hilliard equation.

The formulation of the explanation of the phase-field model in Section 3.2 is based on our previous articles [41, 43, 44]. Thereby the first draft of the modeling section in our collaborative work [44, Sect. 2] was written by Lars von Kolzenberg, where I was also involved in the general conception and formulation.

3.1. Working Principle of Lithium Ion Batteries

We start with a description of the basic design and working principle of lithium-ion batteries. We explain the key processes during discharge and give a description of the occurring phenomenon of phase separation in electrode particles.

Battery Cells The following brief review is based on [84, Sect. 1.2], [107, Sect. 3.1], [40, Sect. 3.1] [91, Sect. 2.1]. For an introduction into electrochemical systems we also refer to [117].

Lithium ion batteries are rechargeable electrochemical energy storage systems. In a simplified way, a lithium-ion battery cell consists of three major components: Two spatially separated *electrodes* embedded in an *electrolyte*, see Figure 3.1. Thereby both electrodes are porous solids and basically consist of the *electrode particles*, while the electrolyte is typically liquid. Common materials for the negative electrode are for example graphite C or silicon Si, whereas the positive electrode may consists of a lithium metal oxide such as lithium manganese oxide $Li_xMn_2O_4$ (LMO) or lithium iron phosphate Li_xFePO_4 (LFP). Candidates for the liquid electrolyte are for example solutions of lithium salts in solvents like lithium hexaflourophosphate $LiPF_6$ in ethylene carbonate $C_3H_4O_3$.

In this arrangement the two electrodes serve as host materials for the reversible *intercalation* and *deintercalation* of lithium. The electrolyte has the function to transport the lithium ions between the electrodes and simultaneously force the electrons to pass through the external circuit.

In order to operate a power device with the battery, it is connected to the two electrodes via an *external circuit*. Thus the potential difference of the two electrodes *discharges* the

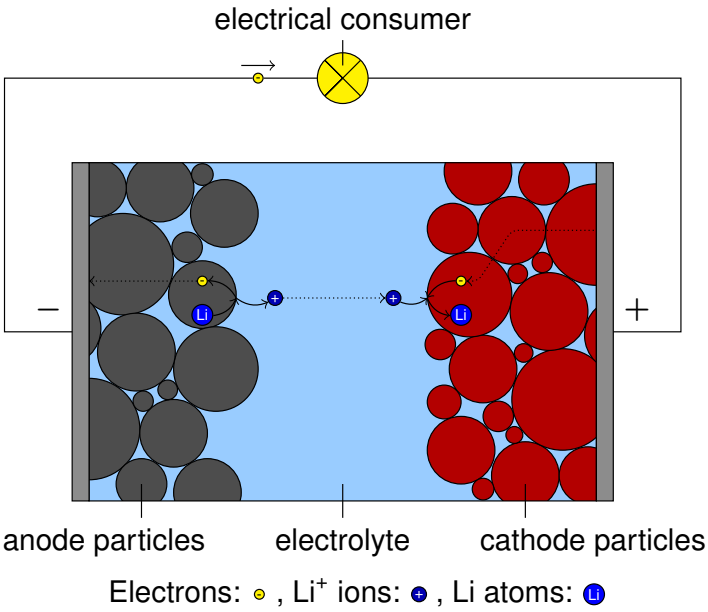

Figure 3.1.: Schematic sketch of a battery cell during discharge according to [107].

battery and provide the energy to the power device. By convention the names of the electrodes depend on the direction of the ion transfer. Hence for battery discharge the negative electrode is denoted as *anode* and the positive electrode as *cathode*. During discharge, lithium atoms *deintercalate* from the *anode* by oxidation at the particle surfaces, i. e., the lithium atoms split into negative electrons and positive lithium ions ($\mathrm{Li} \rightarrow \mathrm{Li}^+ + \mathrm{e}^-$). The released lithium ions are then transported through the electrolyte towards the cathode by diffusion and migration. Once they reached the cathode surface, the lithium ions are reduced and *intercalate* as lithium atoms into the active cathode material ($\mathrm{Li}^+ + \mathrm{e}^- \rightarrow \mathrm{Li}$). Meanwhile for each released lithium ion from the anode an electron enter the external circuit and provide energy to the power device.

As soon as the whole lithium amount is transferred from the anode to the cathode and the electrical energy has been provided to the power device, the battery is fully *discharged*. As key feature of lithium-ion batteries this process is reversible. So connecting the battery cell to a power supply instead of a power device the battery gets *charged*.

Phase Separation Of particular interest is the intercalation behavior in single cathode particles. For example LFP, as electrode material firstly investigated in [118], has emerged as promising cathode material due to its low cost, high thermal and chemical stability [35, Sect. 5.1]. Thereby in contrast to other electrode materials LFP with its chemical formula $\mathrm{Li}_x\mathrm{FePO}_4$ only exists in pure phases with $x \approx 0$ and $x \approx 1$ [134]. This ability for phase separation is also shown by experimental studies [102]. Taking LFP as an example we describe this behavior according to [35, Sect. 5.1] and visualize the process schematically in Figure 3.2.

Starting from a particle with approximately zero lithium concentration (FePO_4), lithium insertion increases the concentration in a single-phase diffusion state. As soon as a certain concentration is reached the separation process sets in and at the surface a second lithium rich (LiFePO_4) phase emerges. While the lithium insertion continues, the phase transition

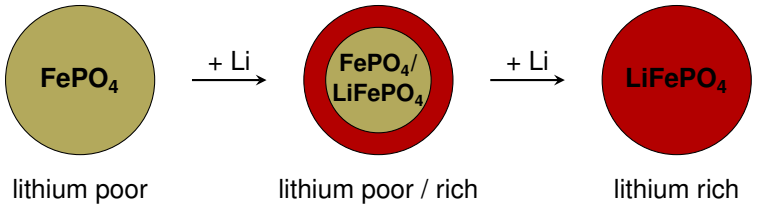

Figure 3.2.: Schematic sketch of the phase separation process during lithium insertion.

between the lithium poor ($FePO_4$) and rich ($LiFePO_4$) phases migrates through the particle. Finally, the lithium rich ($LiFePO_4$) phase occupies the whole particle, the phase transition decays and again a single diffusion state is reached. For lithium extraction the process is inverted.

Beside LFP, several other phase transforming electrode materials undergo the described intercalation process, see for example [169]. However, the crucial property that characterizes LFP is the strong anisotropy. Theoretical [115] and experimental [45, 101] studies show that lithium transport and phase separation is constrained to one-dimensional tunnels [56, 134]. Moreover, this anisotropic behavior cannot be captured by shrinking core-shell models [56, 101].

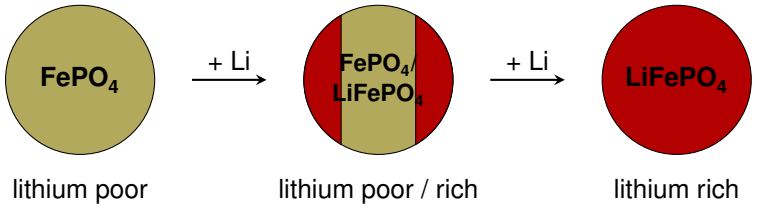

Figure 3.3.: Schematic sketch of the phase separation process during lithium insertion in anisotropic electrode materials.

Instead, the phase separation process in anisotropic electrode materials may behave like in Figure 3.3. If the particle is then lithiated and the concentration has reached the critical value after a single-phase diffusion state, instead of a uniform outer shell, two fronts are formed. Under further lithium insertion these phase fronts travel towards each other until they meet in the middle and merge. Finally, the diffusion takes place again in the single remaining lithium rich phase. For a discussion on bulk transport and surface reaction limitations, we refer to [134].

For this phase separation process in electrode particles with the transition from single-phase to two-phase and back to single-phase diffusion we follow a current modeling approach and develop an efficient numerical solution algorithm in this thesis.

3.2. Model Equations for Phase Separation

First introduced by Han et al. [77], past and recent works like [59, 134, 162, 170] model phase separation in single LFP electrode particles with a Cahn–Hilliard-type phase-field model. Since the same underlying model is used for several materials as for example for silicon [46, 113, 163], LMO [84, 149, 167] as well as sodium iron phosphate Na_xFePO_4

(NFP) [168, 170], we neglect the specific underlying material and consider a generic phase transforming electrode material.

In the upcoming section we introduce and discuss the phase-field model for phase transforming electrode materials, which we will later solve with our finite element solver from Chapter 4. Therefore we basically adapt the modeling section from our article [44, Sect. 2]. For further details on the general phase-field modeling we follow and refer to Kamlah and co-workers [84, 149, 166, 168–170]. Additional information to the incorporation of anisotropy is based on the work of Singh et al. [134] as well as Zhao et al. [174].

We first explain the phase-field model and describe the governing partial differential equations for two-phase diffusion. In particular, we discuss the most general including anisotropic diffusion and gradient energy penalties. Further, we state the boundary conditions for lithium insertion or extraction and the initial condition. This set of equations is normalized and summarized in an initial boundary value problem. We conclude this section with a brief review on alternative modeling approaches for phase separation in electrode particles.

3.2.1. Phase-Field Model

Here we introduce the phase-field model for phase transforming electrode materials under lithium insertion. The resulting well-known Cahn–Hilliard equation [37, 38] determines the evolution of the unknown *lithium concentration* distribution $c\colon [0, t_\mathrm{f}] \times \overline{\Omega} \to [0, c_\mathrm{max}]$ measured in $\mathrm{mol\,m^{-3}}$.

Geometry Electrode particles are three-dimensional objects ($d = 3$), however, from a mathematical point of view the model equations are also valid in lower spatial dimensions ($d \leq 3$). For the higher generality we write the model equations in a dimension-independent way using $d \in \{1, 2, 3\}$, where we give the units, however, only for the realistic three-dimensional case. So let $\Omega \subset \mathbb{R}^d$ describe the domain and geometry of an electrode particle with boundary $\partial\Omega$ and outer unit normal vector $\boldsymbol{n}$. Assuming an idealized particle with sufficiently smooth boundary, we define the specific surface $A_\mathrm{V} \coloneqq |\Omega|/|\partial\Omega|$ as the surface to volume ratio with the unit $\mathrm{m^3\,m^{-2}}$.

Free Energy The phase-field model for two-phase diffusion is based on a free energy density function $\psi\colon [0, 1] \times \mathbb{R}^d \to \mathbb{R}$ with the unit $\mathrm{J\,m^{-3}}$, which consists of a chemical $\psi_\mathrm{ch}\colon [0, 1] \to \mathbb{R}$ and an interfacial part $\psi_\mathrm{int}\colon \mathbb{R}^d \to \mathbb{R}$

$$\psi(z, \boldsymbol{p}) = \psi_\mathrm{ch}(z) + \psi_\mathrm{int}(\boldsymbol{p}). \tag{3.1}$$

With formal argument $z \in [0, 1]$ representing the *normalized lithium concentration* $\overline{c}\colon [0, t_\mathrm{f}] \times \overline{\Omega} \to [0, 1]$, with $\overline{c} \coloneqq c/c_\mathrm{max}$, the *homogeneous chemical free energy density* $\psi_\mathrm{ch}\colon [0, 1] \to \mathbb{R}$ has the form

$$\psi_\mathrm{ch}(z) = RTc_\mathrm{max}\left[\alpha_1 z + \frac{\alpha_2}{2}z^2 + z\log(z) + (1 - z)\log(1 - z)\right], \tag{3.2}$$

scaled by the *maximum lithium concentration* of the host material $c_\mathrm{max} \in \mathbb{R}_{>0}$, the *universal gas constant* R and the *operation temperature* T in Kelvin. The first two terms account

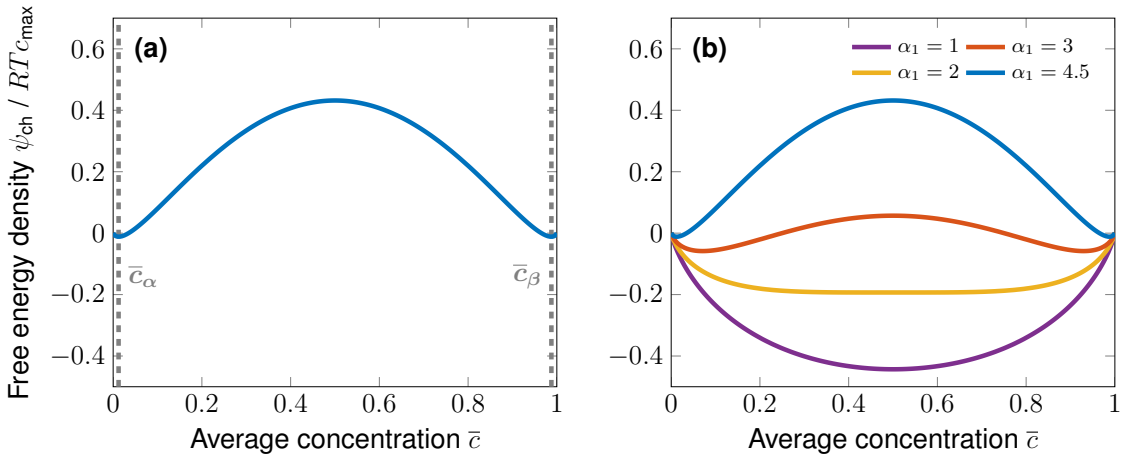

Figure 3.4.: Normalized homogeneous chemical free energy density over constant average concentration. (a) Double well shape for LFP parameters $\alpha_1 = 4.5$, $\alpha_2 = -9$ with normalized equilibrium concentrations $\bar{c}_\alpha$, $\bar{c}_\beta$. (b) Single and double well shaped potentials controlled by the material parameters α_1, α_2 under the assumption $\alpha_2 = -2\alpha_1$.

for the energetic contributions from the interaction of lithium ions with the host material ($\alpha_1 > 0$) and other lithium ions ($\alpha_2 < 0$) [77, 84, 168]. The third and fourth term describe the entropic contributions of mixing [134, 168]. For phase separating materials these interactions lead to a double well potential as illustrated for LFP parameters [168] in Figure 3.4a, whereby the normalized equilibrium concentrations of the two stable phases $\bar{c}_\alpha$, $\bar{c}_\beta$ are determined by the local minima [84, 149, 168].

Note, that we assume a constant operation temperature T equal to the reference temperature T_{ref} used in [84, Sect. 3.2.2] for the introduction of the two parameters α_1 and α_2. Thus, the homogeneous chemical free energy density ψ_{ch} is a double well if it has a zone of concavity, i. e., the set $\left\{z \in [0,1] \,:\, \partial_z^2 \psi_{\text{ch}}(z) < 0\right\}$ is nonempty. We determine an upper bound for α_2, which guarantees the double well shape of ψ_{ch}, by a simple calculation

$$\left\{z \in [0,1] \,:\, \partial_z^2 \psi_{\text{ch}}(z) < 0\right\} \neq \emptyset \iff \exists z \in [0,1] \,:\, \partial_z^2 \psi_{\text{ch}}(z) < 0 \tag{3.3}$$

$$\iff \exists z \in [0,1] \,:\, \alpha_2 + \frac{1}{z(1-z)} < 0, \tag{3.4}$$

where (3.4) is satisfied if $\alpha_2 < -4$. Under the assumption $\alpha_2 = -2\alpha_1$, valid for LFP [134, 162], we see in Figure 3.4b that the double well ($\alpha_1 > 2$) turns into a single well ($\alpha_1 \leq 2$) at $\alpha_1 = 2$. For further details particularly regarding the connection to the critical temperature for phase separation, we refer to [84, Sect. 3.2.2].

The *interfacial energy density* $\psi_{\text{int}} \colon \mathbb{R}^d \to \mathbb{R}$

$$\psi_{\text{int}}(\boldsymbol{p}) = \frac{1}{2} RT c_{\text{max}}\, \boldsymbol{p} \cdot \mathbf{K}\boldsymbol{p} \tag{3.5}$$

with the formal argument $\boldsymbol{p} \in \mathbb{R}^d$ representing the gradient of the normalized lithium concentration $\nabla \bar{c} \colon [0, t_{\text{f}}] \times \overline{\Omega} \to \mathbb{R}^d$ gives rise to a separation of these phases. In a more general fashion a symmetric, positive definite tensorial *interfacial energy coefficient* $\mathbf{K} \in \mathbb{R}^{d,d}$ determines the possibly anisotropic energy penalties for the concentration gradient [134].

With the scalar interfacial energy coefficient $\kappa > 0$ and the specialization $\mathbf{K} = \kappa\mathbf{Id}$ the isotropic case is recovered

$$\psi_{\text{int}}(\boldsymbol{p}) = \frac{1}{2} RT c_{\max} \kappa |\boldsymbol{p}|^2. \tag{3.6}$$

In particular, the scalar interfacial energy coefficient κ determines the interface thickness s [38] according to

$$s = (\overline{c}_\beta - \overline{c}_\alpha) \sqrt{\frac{\kappa}{2 \triangle \overline{\psi}_{\text{ch}}}}\,, \tag{3.7}$$

with the normalized equilibrium concentrations of the lithium poor $\overline{c}_\alpha$ and lithium rich phase $\overline{c}_\beta$. The excess free energy density $\triangle\overline{\psi}_{\text{ch}}$ is defined as the difference of the normalized chemical free energy density $\overline{\psi} = \psi / RT c_{\max}$ at $(\overline{c}_\alpha + \overline{c}_\beta)/2$ and the normalized chemical free energy density obtained as mixture of the α and β phases [173]

$$\triangle\overline{\psi}_{\text{ch}} = \overline{\psi}_{\text{ch}}\left(\frac{\overline{c}_\alpha + \overline{c}_\beta}{2}\right) - \frac{1}{2}\big(\overline{\psi}_{\text{ch}}(\overline{c}_\alpha) + \overline{\psi}_{\text{ch}}(\overline{c}_\beta)\big). \tag{3.8}$$

The total free energy of the system $\Psi\colon [0, t_{\text{f}}] \to \mathbb{R}$ measured in Joule depends on the normalized lithium concentration and is defined as

$$\Psi(t) = \Psi\big(\overline{c}(t, \cdot)\big) = \int_\Omega \psi\big(\overline{c}(t, \boldsymbol{x}), \nabla\overline{c}(t, \boldsymbol{x})\big)\, \mathrm{d}\boldsymbol{x}. \tag{3.9}$$

Based on this free energy formulation, we proceed to derive constitutive equations describing the chemistry of the model.

Chemistry The diffusion of lithium concentration in the host material is based on the conservation of mass [149]

$$\partial_t \overline{c} = -\nabla \cdot \boldsymbol{N} \qquad \text{in } (0, t_{\text{f}}) \times \Omega. \tag{3.10}$$

Inside the material a gradient in the *chemical potential* $\mu\colon [0, t_{\text{f}}] \times \overline{\Omega} \to \mathbb{R}$ drives the *lithium flux* $\boldsymbol{N}$ according to the constitutive relation

$$\boldsymbol{N} = -\mathbf{m}(\overline{c}) \nabla \mu, \tag{3.11}$$

with a symmetric, positive definite, tensor-valued *mobility* $\mathbf{m}\colon [0, 1] \to \mathbb{R}^{d,d}$ depending on the normalized lithium concentration. According to [59, 149, 168] the mobility is defined by the expression

$$\mathbf{m}(\overline{c}) = \mathbf{D} \frac{c_{\max}}{RT} \overline{c}(1 - \overline{c}), \tag{3.12}$$

with the symmetric, positive definite tensorial *diffusion coefficient* $\mathbf{D} \in \mathbb{R}^{d,d}$. In this form the concentration dependence of the mobility reflects the vanishing diffusion for pure phases [59]. Analogously to the possibly anisotropic interfacial energy coefficient $\mathbf{K}$, the isotropic mobility $m\colon [0, 1] \to \mathbb{R}$ is recovered with a scalar diffusion coefficient $D > 0$ and the assumption $\mathbf{D} = D\,\mathbf{Id}$.

Finally, the *chemical potential* with the unit $\mathrm{J\,mol^{-1}}$ is defined as variational derivative of the system's free energy Ψ (3.9) with respect to the lithium concentration c [79, 134]

$$\begin{aligned}\mu &= \partial_z \psi\big(\overline{c}, \nabla\overline{c}\big) - \nabla\cdot\partial_{\boldsymbol{p}}\psi\big(\overline{c}, \nabla\overline{c}\big)\\ &= \partial_z \psi_{\text{ch}}(\overline{c}) - \nabla\cdot\partial_{\boldsymbol{p}}\psi_{\text{int}}\big(\nabla\overline{c}\big)\\ &= RT\Big[\alpha_1 + \alpha_2\overline{c} + \log\big(\overline{c}\big) - \log\big(1-\overline{c}\big) - \nabla\overline{c}\cdot\mathbf{K}\nabla\overline{c}\Big] \qquad \text{in } (0,t_{\text{f}})\times\Omega.\end{aligned} \tag{3.13}$$

3.2.2. Boundary and Initial Conditions

So far the resulting set of partial differential equations (3.10) and (3.13) for the two-phase diffusion of lithium in electrode particles coincides with the usually studied Cahn–Hilliard equation like in [64, 74, 154]. However, with the following boundary conditions modeling lithium insertion or extraction we deviate from the standard Cahn–Hilliard modeling.

Boundary Conditions At the particle surface, we assume over time a vanishing normal derivative of the concentration

$$\nabla c\cdot\boldsymbol{n} = 0 \qquad \text{on } (0,t_{\text{f}})\times\partial\Omega. \tag{3.14}$$

This condition neglects surface wetting and ensures that the phase transition stays always orthogonal on the particle surface. An inhomogeneous Neumann condition instead influences the concentration distribution and thus the location of nucleation and the initiation of phase separation, see [162].

To model lithium insertion or extraction we actual deviate from the classical, mass conserving Cahn–Hilliard model and apply a uniform and constant *external lithium flux* $N_{\text{ext}} \in \mathbb{R}$ over time

$$\boldsymbol{N}\cdot\boldsymbol{n} = N_{\text{ext}} \qquad \text{on } (0,t_{\text{f}})\times\partial\Omega. \tag{3.15}$$

Thereby a negative value corresponds to lithium insertion, a positive sign for lithium extraction respectively. We measure this applied flux in terms of the *C-rate*, which defines the *cycle time* $t_{\text{cycle}} = 1/\text{C-rate}$ describing the hours it takes to fully charge or discharge the particle. The applied lithium flux in C is transformed to SI units by

$$|N_{\text{ext}}|\left[\mathrm{mol\,m^{-2}\,s^{-1}}\right] = \frac{1\,\mathrm{h}}{3600\,\mathrm{s}}\frac{c_{\text{max}}}{A_{\text{V}}}N_{\text{ext}}\,[\mathrm{C}] \tag{3.16}$$

with the specific surface A_{V}.

Note, that the mathematical problem formulation is not limited to uniform and constant applied lithium fluxes. In general, the problem formulation is capable for space and time-dependent as well as even nonlinear concentration and chemical potential dependent fluxes. Alternatives are studied previously in [59, 134, 162].

Initial Condition At initial time we assume a constant concentration $c_0 \in (0, c_{\text{max}})$ in the whole particle

$$c(0,\,\cdot\,) = c_0 \qquad \text{in } \Omega. \tag{3.17}$$

Thereby for lithium insertion the particle should be approximately empty $c_0 \gtrsim 0$ and for lithium extraction approximately full $c_0 \lesssim 1$.

Temporal Dependence of the State of Charge Using the C-rate and the constant normalized initial concentration $\overline{c}_0$, we link the *State of Charge* (SOC) of the particle $\mathrm{SOC}\colon [0, t_\mathrm{f}] \to [0, 1]$ to the simulation time by

$$\mathrm{SOC}(t) = \frac{1}{|\Omega|} \int_\Omega \overline{c}(t, \boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} = \overline{c}_0 + N_\mathrm{ext} \, [\mathrm{C}] \cdot t[\mathrm{h}]. \tag{3.18}$$

3.2.3. Problem Statement

Next we summarize the model equations in an initial boundary value problem. Thereby, we refer to Appendix A.2 for the normalization and assume in the following the dimensionless form of the model equations. The material parameters specifying the LFP electrode material are given in Table A.1. However, at this point we highlight, that the here presented model was recently also used for other phase transforming materials, like LMO [84, 149], silicon [46, 163] or NFP [168, 170].

For the problem formulation we introduce the chemical potential μ as additional solution variable and consider the mixed formulation of the Cahn–Hilliard equation. Thus we split the fourth-order diffusion equation into two second-order equations [46, 49, 63, 150, 154]. The dimensionless initial boundary value problem reads as:

Problem 1. Let $t_\mathrm{f} > 0$ be the final simulation time and $\Omega \subset \mathbb{R}^d$ be a bounded domain representing an electrode particle. Find the normalized lithium concentration $c\colon [0, t_\mathrm{f}] \times \overline{\Omega} \to [0, 1]$ and the chemical potential $\mu\colon [0, t_\mathrm{f}] \times \overline{\Omega} \to \mathbb{R}$ satisfying

$$\begin{cases} \partial_t c = \nabla \cdot \big(\mathbf{m}(c) \nabla \mu\big) & \text{in } (0, t_\mathrm{f}) \times \Omega, \\ \mu = \psi'_\mathrm{ch}(c) - \nabla c \cdot \mathbf{K} \nabla c & \text{in } (0, t_\mathrm{f}) \times \Omega, \\ \nabla c \cdot \boldsymbol{n} = 0 & \text{on } (0, t_\mathrm{f}) \times \partial\Omega, \\ \mathbf{m}(c) \nabla \mu \cdot \boldsymbol{n} = -N_\mathrm{ext} & \text{on } (0, t_\mathrm{f}) \times \partial\Omega, \\ c(0, \,\cdot\,) = c_0 & \text{in } \Omega. \end{cases} \tag{3.19}$$

In dimensionless form the tensor-valued mobility is given by

$$\mathbf{m}(c) = \mathbf{D} c (1 - c), \tag{3.20}$$

and we obtain for the derivative of the homogeneous chemical free energy density

$$\psi'_\mathrm{ch}(z) \coloneqq \partial_z \psi_\mathrm{ch}(z) = \alpha_1 + \alpha_2 c + \log(c) - \log(1 - c). \tag{3.21}$$

Note, since the concentration dependent mobility vanishes and the derivative of the homogeneous chemical free energy density blows up for $\overline{c} \in \{0, 1\}$, the initial concentration must be bounded away from the complete full and empty state, i. e., we assume $c_0(\boldsymbol{x}) \in (0, 1)$ for all $\boldsymbol{x} \in \overline{\Omega}$. For the application case this is in general not restrictive, since completely full or empty particles might not exists in reality.

3.2.4. Alternative Phase Separation Models

Finally, we shortly note some alternative model approaches for phase separation in electrode particles of lithium-ion batteries with possible connections to the here presented phase-field model.

In general, we distinguish between sharp and diffusive interface models. The difference is explained by their name. For sharp interface models the phase transition is represented by a discontinuity in the concentration profile. Instead, in diffusive interface models the concentration smoothly changes from one to the other phase in a small transition zone.

Sharp interface models can further be separated into shrinking core [57, 135, 137] and core-shell [132, 133, 165] scenarios. Mathematically, these models are one-phase or two-phase Stefan problems, see for example [86]. While in shrinking core models the concentration in the inner core is held constant, core-shell scenarios allow actual two-phase diffusion in both phases. For these models, tracking the sharp interface and the singularity arising due to the discontinuous jump is numerically challenging. For example, in [40, 42] we implemented and presented a finite element solver for a regularized two-phase Stefan problem as part of a microscale model of a battery half cell.

The Cahn–Hilliard-type phase-field model discussed in this thesis is in literature a well-established diffusive interface model, see for example [46, 77, 81, 84, 85, 149, 150, 162, 163, 166, 168–170] and the references cited therein. The concentration as order parameter of the phase-field allows two-phase diffusion and an interfacial energy coefficient controls the width of the diffusive interface zone. The strongly varying time scale between phase separation and phase migration makes the numerical simulation difficult.

A recent generalization of this model in the context of lithium-ion batteries is the theory of nonlocal species concentration [167], which has been similarly investigated in [59]. In this theory a nonlocal or micromorphic concentration is defined as the weighted average concentration. Depending on the new variable, penalty and variance energies with additional parameters are then introduced and replace the standard interfacial energy functional ψ_{int}. Thus the fourth-order character of the Cahn–Hilliard equation is avoided and a set of two coupled second-order equations remain. However, the Cahn–Hilliard model can be recovered from the nonlocal species concentration theory [167].

Another variant of the Cahn–Hilliard-type model for lithium-ion battery electrode particles includes viscosity, see [112, 113]. There a diffusion equation of the form $\partial_t c = M\Delta(\mu + \gamma\partial_t c)$ with the viscosity parameter γ is used for the evolution of the concentration.

If there is a parameter that controls the with of the phase transition for Cahn–Hilliard-type phase-field models, the question about the limiting case arises when the width tends to zero. According to Dreyer and Guhlke [61] the viscosity term is in fact necessary to ensure the thermodynamic consistency of the sharp limit. A matched asymptotic analysis for a model coupling phase transformation and linear elasticity was presented in [113].

3.3. Mathematical Issues

Finally, we discuss some mathematical aspects of the resulting initial boundary value problem (3.19), including existence and uniqueness of a solution. For simplicity, we assume in the following the purely isotropic case, i. e., $\mathbf{K} = \kappa\,\mathbf{Id}$ with $\kappa > 0$ in (3.5) and $\mathbf{D} = D\,\mathbf{Id}$ with $D > 0$ in (3.12).

The Classical Cahn–Hilliard Equation So far in literature [31, 64, 66, 74] many different variants of the Cahn–Hilliard equation have been considered, which differ essentially in the choice of the free energy density ψ_{ch} (quartic or logarithmic) and the mobility m

(constant or concentration dependent). Common to all of them is that they are posed with either homogeneous Neumann or periodic boundary conditions. Therefore we denote in this thesis especially the case with homogeneous Neumann boundary conditions as the *classical Cahn–Hilliard equation*.

Key properties of the classical Cahn–Hilliard equation are the conservation of mass and the monotone decrease of the energy (3.9). The major difference to the phase-field model introduced in the previous Section 3.2 is the boundary condition that models lithium insertion (3.15). Due to this condition the mass conservation and energy decay become invalid, as we can see from two simple calculations.

Firstly, for a constant and uniform external lithium flux $N_{\text{ext}} < 0$ we get

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\int_\Omega c\,\mathrm{d}\boldsymbol{x} &= \int_\Omega \partial_t c\,\mathrm{d}\boldsymbol{x}\\ &= -\int_\Omega \nabla\cdot\boldsymbol{N}\,\mathrm{d}\boldsymbol{x}\\ &= -\int_{\partial\Omega} \boldsymbol{N}\cdot\boldsymbol{n}\,\mathrm{d}\sigma(\boldsymbol{x}) = \int_{\partial\Omega} |N_{\text{ext}}|\,\mathrm{d}\sigma(\boldsymbol{x}) = |N_{\text{ext}}||\partial\Omega| > 0,\end{aligned} \tag{3.22}$$

which yields an increase of lithium concentration over time and thus implies the correlation between time and SOC (3.18).

Secondly, with $N_{\text{ext}} \neq 0$ the monotone decrease of energy is no longer guaranteed

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\Psi(t) &= \int_\Omega \partial_z\psi\big(c,\nabla c\big)\partial_t c + \partial_{\boldsymbol{p}}\psi\big(c,\nabla c\big)\nabla\partial_t c\,\mathrm{d}\boldsymbol{x}\\ &= \int_\Omega \Big(\psi'_{\text{ch}}(c) - \kappa\Delta c\Big)\partial_t c\,\mathrm{d}\boldsymbol{x}\\ &= \int_\Omega \mu\,\Big(\nabla\cdot\big(m(c)\nabla\mu\big)\Big)\,\mathrm{d}\boldsymbol{x}\\ &= -\int_\Omega m(c)\nabla\mu\cdot\nabla\mu\,\mathrm{d}\boldsymbol{x} - \int_{\partial\Omega}\mu\,N_{\text{ext}}\,\mathrm{d}\sigma(\boldsymbol{x}) \not\leq 0.\end{aligned} \tag{3.23}$$

As we will see later in the numerical experiments, the energy actually increases during lithiation.

Existence of a Solution The monotone energy decrease plays a central role in existing proofs for a priori estimates for solution of the Cahn–Hilliard equation, see for example [54, 62, 65, 66], and is therefore necessary for the proof of the global existence and uniqueness of solutions. For the classical Cahn–Hilliard equation with degenerated mobility and logarithmic free energy, however, Elliott and Garcke [64] only proved the global existence of a weak solution. The uniqueness in the nondegenerated case was later proven in [15].

Theorem 3.1 (Compare Thm. 1 of [64])**.** *If either $\partial\Omega \in C^{1,1}$ or Ω is convex, $c_0 \in H^1(\Omega)$ with $|c_0| \leq 1$ almost everywhere and further assumptions are fulfilled, there exists a*

pair $(c, \boldsymbol{N})$ *with*

$$
\begin{aligned}
&c \in L^2\big(0, t_f; H^2(\Omega)\big) \cap L^\infty\big(0, t_f; H^1(\Omega)\big) \cap C\big([0, t_f]; L^2(\Omega)\big),\\
&\partial_t c \in L^2\big(0, t_f; H^{-1}(\Omega)\big),\\
&c(0) = c_0 \text{ and } \nabla c \cdot \boldsymbol{n} = 0 \text{ on } (0, t_f) \times \partial\Omega,\\
&|c| \leq 1 \text{ almost everywhere in } (0, t_f) \times \Omega,\\
&\boldsymbol{N} \in L^2\big((0, t_f) \times \Omega, \mathbb{R}^d\big),
\end{aligned}
$$

satisfying $\partial_t c = -\nabla \cdot \boldsymbol{N}$ *in* $L^2\big(0, t_f; H^{-1}(\Omega)\big)$*, i. e.,*

$$
\int_0^{t_f} \big\langle \chi(t), \partial_t c(t) \big\rangle_{H^1, H^{-1}} \,\mathrm{d}t = \int_{(0,t_f)\times\Omega} \boldsymbol{N} \cdot \nabla \chi \,\mathrm{d}(t, \boldsymbol{x}),
$$

for all $\chi \in L^2\big(0, t_f; H^1(\Omega)\big)$ *and*

$$
\boldsymbol{N} = -m(c) \nabla \big(\psi'_{ch}(c) - \kappa \Delta c\big),
$$

in the following weak sense

$$
\int_{(0,t_f)\times\Omega} \boldsymbol{N} \cdot \boldsymbol{\eta} \,\mathrm{d}(t, \boldsymbol{x}) = -\int_{(0,t_f)\times\Omega} \Big[\kappa \Delta c \nabla \cdot \big(m(c)\boldsymbol{\eta}\big) + m(c)\psi''_{ch}(c) \nabla c \cdot \boldsymbol{\eta}\Big] \,\mathrm{d}(t, \boldsymbol{x}),
$$

for all $\boldsymbol{\eta} \in L^2\big(0, t_f; H^1(\Omega, \mathbb{R}^d)\big) \cap L^\infty\big((0, t_f) \times \Omega, \mathbb{R}^d\big)$ *which fulfill* $\boldsymbol{\eta} \cdot \boldsymbol{n} = 0$ *on* $(0, t_f) \times \partial\Omega$*.*

The idea of the proof is to first show the existence of a solution for the nondegenerated case [64]. This is used to show the existence of approximate solutions for the degenerated case [64]. Finally, energy estimates are used to pass to the limit in the approximate solutions, which gives the existence of a weak solution [64].

Note, that in [64] a modified logarithmic free energy density is used, such that the local minima are located in ± 1. This, however, is only a matter of an appropriate scaling.

In particular, the theorem states, that the solution is always bounded in the range of the local minima, which supports the fact that the concentration in the above phase-field model should not exceed the physical meaningful range $[0, c_{\text{max}}]$. Instead, if one uses quartic polynomials the solution might under or overshoots this range.

Modeling Issues Causing Mathematical Problems In the phase-field model lithium insertion is modeled with an inhomogeneous Neumann boundary condition, for which case the global existence and uniqueness of a solution is not covered by the above Theorem 3.1. In fact, we cannot even expect global existence and uniqueness of a solution for a constant and uniform lithium flux. Because, as we have seen above, the lithium concentration is increasing monotonously and without a stopping criterion the maximum concentration capability $|\Omega| c_{\text{max}}$ of the particle is exceeded at some time. Hence, we can at most expect only local existence and uniqueness of a solution.

Essentially the same issue occurred in the mathematical analysis of a microscale model for lithium-ion batteries in [107]. There local in time existence of weak solutions and for $d \leq 3$ also uniqueness of the weak solutions were shown. In fact, from a physical point of view, this is the most we can expect, because a constant and uniform insertion rate leads

at some point to a completely filled battery and then further insertion is no longer physical meaningful.

It should be noted that this issue may also depend on the choice of the external lithium flux N_{ext}, which itself depends on the modeling and the application. The constant and uniform flux at a low rate $N_{\text{ext}} = 0.001\,\text{C}$ is used for example in [149, 168, 170] to observe equilibrium states under quasi static insertion rates. Differently for example in [59, 162], where time-dependent or Butler–Volmer-type reaction kinetics are used to model lithium insertion.

Finally, it is worth pointing out that a strong anisotropy in (3.5) or (3.12) can lead to an ill-posed problem [153].

4. The Numerical Solution Algorithm

After the introduction of the model equations for phase separation in electrode particles under lithium insertion, we present in this chapter the main development of this thesis—an adaptive matrix-free finite element solver for the Cahn–Hilliard equation.

In the subsequent sections we explain the three major components of the solver. First, in Section 4.1 we derive a fully discrete problem formulation for the model equations from Chapter 3, where we discretize the mixed formulation of the Cahn–Hilliard equation in space with a finite element method and employ linear multistep methods for the time integration. Second, in Section 4.2, we explain the solution of the nonlinear algebraic system with Newton's method. In particular, for the solution of the linearized systems with the GMRES method we propose a preconditioner, which is suitable for the matrix-free application. Finally, in Section 4.3 we present and discuss our space and time adaptive solution algorithm, based on a variable-step, variable-order time integration scheme and an a posteriori refinement criterion for local mesh refinement.

Basically, this chapter is an extension of our article [43], where we presented a first version of the finite element solver, the matrix-free preconditioner and the adaptive algorithm. However, here we discuss an improved version of the methods in more detail. Although our submitted work [44] considers the coupled chemical-mechanical problem, the basic idea of the solver is the same and thus the collaborative writing of the manuscript also influenced the formulation and wording of this chapter.

Before we explain the numerical methods, we give a short review on numerical methods for the Cahn–Hilliard equation. Note, since the phase-field model from Chapter 3 generalizes the classical Cahn–Hilliard equation with homogeneous Neumann boundary conditions, the presented finite element solver is applicable for this case as well.

Overview on Numerical Methods For an introduction into the theory of finite element methods we refer to the textbooks [30, 33, 95]. An introduction into time integration algorithms can be found in [75, 76, 120, 128] and in the context of parabolic partial differential equations also in [95].

Spatial Discretization The numerical solution of the Cahn–Hilliard equation involves the spatial discretization of a fourth-order differential operator. Established methods from literature are mainly divided into two types, which both have advantages and drawbacks. On the one hand, *direct approaches* attack the Cahn–Hilliard equation as fourth-order equation

$$\partial_t c = \Delta\big(f'(c) - \Delta c\big), \tag{4.1}$$

while on the other hand, *mixed formulations* introduce an auxiliary variable μ and split the

fourth-order equation into a set of two coupled equations

$$\begin{cases} \partial_t c = \Delta \mu, \\ \mu = f'(c) - \Delta c. \end{cases} \tag{4.2}$$

Considering the direct approach (4.1), we have only one solution variable and thus save the amount of degrees of freedom (DOFs) for an additional variable. However, due to the fourth-order differential operator, the discrete solution has to fulfill additional regularity requirements. Unfortunately only a limited number of finite element methods fulfill the regularity requirements [154], like the C^1-continuous Hermite elements used in [164]. Alternatively to finite element methods, an isogeometric analysis technique based on C^1-continuous non-uniform rational B-splines (NURBS) was proposed by Gomez et al. [73]. Different to that a discontinuous Galerkin method was used in [151]. Especially the isogeometric analysis was already used for electrode particle simulations in [173, 174].

Instead, mixed formulations have the double amount of DOFs due to the auxiliary solution variable, but can be solved without additional regularity requirements. For example the mixed formulation was already solved with methods based on finite differences [153], finite volumes [80], isogeometric analysis NURBS [74] or most commonly with C^0-continuous Lagrangian finite elements [16, 31, 49, 63, 154, 164]. In the battery context the phase-field model for phase separation in electrode particles was solved for example with a finite volume scheme in [50] or finite elements [168, 169].

For a comparison of accuracy and efficiency of different methods we refer for example to [89, 90, 164]. Further references on numerical methods are given for example in [31, 55].

With the chemical potential (3.13) as additional solution variable we consider the Cahn–Hilliard equation in this thesis in form of the mixed formulation (4.2) and employ a finite element method. Existence and uniqueness of finite element approximations to the mixed formulation and convergence results were proven for different versions of the Cahn–Hilliard equation. For example Elliott et al. [63] showed optimal error bounds of a semi-discrete finite element method for the Cahn–Hilliard equation with quartic free energy and constant mobility. Copetti and Elliott [49] proved the existence and uniqueness of the numerical solution of a fully discrete finite element method as well as convergence for the Cahn–Hilliard equation with logarithmic free energy and constant mobility. Later Barrett and Blowey [15] assumed a logarithmic free energy and a nondegenerated, concentration dependent mobility and showed the existence and uniqueness of a numerical solution for a fully discrete piecewise linear finite element method in one and two space dimensions. In particular, Barrett et al. [16] showed for the Cahn–Hilliard equation with degenerated mobility well posedness, stability and proved convergence of a fully practical finite element approximation in one space dimension.

Time Integration Methods The efficient time integration of the Cahn–Hilliard equation is a difficult task due to the strongly varying time scales between the fast phase separation and the slow coarsening process [145, 154]. Thus much effort was spent to investigate time integrators for the classical Cahn–Hilliard equation.

Generally, due to the high stiffness of the Cahn–Hilliard equation explicit methods are infeasible [154]. Implicit methods, like the backward Euler scheme used for example in [49], are unconditionally stable and allow larger time step sizes, however, need to solve a nonlinear system in each step [154]. Alternatively, semi-implicit schemes use a mix of implicitly and explicitly tracked terms, see for example [55, 145]. A well-established

semi-implicit class of schemes traces back to Eyre [68], which splits the free energy in an implicitly tracked convex and an explicit tracked nonconvex part [145, 154]. The key property of this method is the unconditionally energy stability, i. e., the monotonous energy decay of the continuous problem is inherited to the discrete problem [145].

In the classical Cahn–Hilliard theory the energy stability is an important property. In particular, considering our model equations (3.19) with logarithmic free energy and degenerated mobility in the classical case, Gomez and Hughes [74] presented a provably, second-order convergent and energy stable time integrator. But, energy stable schemes may fail in the context of our model problem (3.19), since the energy might not be monotonously decreasing, see (3.23).

More advanced higher-order, diagonal implicit Runge–Kutta methods in an adaptive algorithm are used and compared in [154]. For a review on different time integration schemes sorted according their properties, such as convergence order, energy stability, unique solvability and (non-)linearity, we refer to [145].

Discrete Solvers For the semi-implicit as well as the fully implicit time integration, the solution of linear systems becomes necessary in the end. In order to solve the arising systems, iterative methods like GMRES are usually used, for which efficient preconditioners are necessary, since the stiffness of the fourth-order Cahn–Hilliard equation results in a high condition number of the resulting matrices. Preconditioners for the mixed formulation of the Cahn–Hilliard equation were for example studied in [8, 28, 29, 31, 32]. In a wider context, preconditioners for a more general class of fourth-order partial differential equations including the Cahn–Hilliard equation are discussed for example in [14, 175]. Alternatively, multigrid methods were used so far, see [153, 176].

Adaptive Schemes The multiscale nature in time and the almost sharp, moving phase transition cause the main challenges of the Cahn–Hilliard equation. Thus adaptive methods are inevitable for the efficient numerical solution.

In order to resolve the spatial profile accurately the mesh should clearly be refined locally in the phase transition zone, where the concentration varies in the range between the equilibrium concentrations. In regions away from the phase transition a locally coarsened mesh might be sufficient. Local refinement criteria like a modified flux jump indicator based on the biharmonic operator [136] or recovery estimators [176] were used beside a posteriori error estimators [19, 20, 23, 24, 71, 161].

A common heuristic strategy for the adaptive change of the time step size is based on the number of iteration steps for the internal nonlinear solver [154], see for example [81]. More convenient is the adaption of the time step size based on hierarchical error estimation [74, 154].

For the most efficient simulation spatial and temporal adaptivity should be coupled, see [17, 136] and as we did already in [43, 44].

4.1. The Space and Time Discretization

In this section we present the space and time discretization of the mixed formulation of the Cahn–Hilliard-type phase-field model (3.19). For the spatial discretization we employ a p-th-order isoparametric Lagrangian finite element method. Thereby we bring the resulting semi-discrete problem in the form of a generic differential algebraic equation (DAE). For

the time integration we use the linear multistep method (LMM) family of Klopfenstein–Shampine numerical differentiation formulas (NDFs) [129]. These time integration methods are based on the well-known backward differentiation formulas (BDFs) [75, 76, 120, 128] and are particularly suited for stiff problems.

4.1.1. Discretization in Space

We start deriving the weak formulation of the initial boundary value Problem 1. Due to physical limitations inside electrode particles, the lithium concentration must never exceed the physical meaningful range $c \in [0, c_{\text{max}}]$ (in dimensionless form $c \in [0, 1]$). For our numerical experiments we control the initial concentration, the applied lithium flux, and the final time, such that the concentration is bounded away from zero and one. Thus, by assumption, the concentration dependent mobility never gets degenerated.

So for the weak formulation we follow [15], multiply the partial differential equations (3.19) with test functions and integrate over the domain Ω. Integration by parts yields the weak formulation: Find the concentration $c\colon [0, t_{\text{f}}] \times \overline{\Omega} \to [0, 1]$ and the chemical potential $\mu\colon [0, t_{\text{f}}] \times \overline{\Omega} \to \mathbb{R}$ such that $c(0, \cdot) = c_0$ and for $t \in [0, t_{\text{f}}]$ satisfying

$$\begin{cases} \big(v, \partial_t c\big) = -\big(\nabla v, \mathbf{m}(c)\nabla\mu\big) - \big(v, N_{\text{ext}}\big)_{\partial\Omega} & \text{for all } v \in V \coloneqq H^1(\Omega), \\ \big(w, \mu\big) = \quad \big(w, \psi'_{\text{ch}}(c)\big) + \big(\nabla w, \mathbf{K}\nabla c\big) & \text{for all } w \in V. \end{cases} \tag{4.3}$$

For the application of the isoparametric Lagrangian finite element method [30, 33] we replace the domain Ω by a computational domain Ω_h, which we assume to be a polytop with appropriate boundary approximation of curved boundaries. Next we chose an admissible mesh $\mathcal{T}_h$ of the computational domain Ω_h and define the finite dimensional subspace $V_h \subset V$ by the usual continuous piecewise p-th-order polynomial function space

$$V_h \coloneqq \mathcal{S}^p(\mathcal{T}_h) = \Big\{ v \in C(\overline{\Omega}_h) \,:\, v|_Q \in \mathbb{Q}_p(Q) \text{ for all } Q \in \mathcal{T}_h \Big\}, \tag{4.4}$$

with its nodal basis $\Pi \coloneqq \big\{\varphi_i \in V_h \,:\, i = 1, \dots, N_x\big\}$. Now for $t \in [0, t_{\text{f}}]$ we seek $c_h(t, \cdot) \in V_h \cap \big\{c_h \in [0, 1]\big\}$ and $\mu_h(t, \cdot) \in V_h$ to be solutions of the spatial discrete system

$$\begin{cases} \big(\varphi_i, \partial_t c_h\big) = -\big(\nabla\varphi_i, \mathbf{m}(c_h)\nabla\mu_h\big) - \big(\varphi_i, N_{\text{ext}}\big)_{\partial\Omega} & \text{for all } i = 1, \dots, N_x, \\ 0 = -\big(\varphi_i, \mu_h\big) + \big(\varphi_i, \psi'_{\text{ch}}(c_h)\big) + \big(\nabla\varphi_i, \mathbf{K}\nabla c_h\big) & \text{for all } i = 1, \dots, N_x. \end{cases} \tag{4.5}$$

In this set of equations we insert the basis representation for the discrete functions

$$c_h(t, x) = \sum_{j=1}^{N_x} c_j(t)\varphi_j(x), \qquad \mu_h(t, x) = \sum_{j=1}^{N_x} \mu_j(t)\varphi_j(x). \tag{4.6}$$

Now we identify the spatially discrete function $c_h(t, \cdot)$ with the vector-valued function $\boldsymbol{c}(t) = [c_j(t)]_j \in [0, 1]^{N_x}$ and $\mu_h(t, \cdot)$ with $\boldsymbol{\mu}(t) = [\mu_j(t)]_j \in \mathbb{R}^{N_x}$. Gathering the solution variables for the concentration and the chemical potential in a vector-valued solution variable

$$\boldsymbol{y}\colon [0, t_{\text{f}}] \to \mathbb{R}^{2N_x},\ t \mapsto \boldsymbol{y}(t) = \begin{pmatrix} \boldsymbol{c}(t) \\ \boldsymbol{\mu}(t) \end{pmatrix}, \tag{4.7}$$

we arrive at the nonlinear DAE for the spatially discretized system: Find $\boldsymbol{y}\colon [0,t_\mathrm{f}] \to \mathbb{R}^{2N_x}$ satisfying:

$$\mathbf{M}\partial_t \boldsymbol{y} = \boldsymbol{f}(t,\boldsymbol{y}) \quad \text{for } t \in (0,t_\mathrm{f}], \qquad \boldsymbol{y}(0) = \boldsymbol{y}^0. \tag{4.8}$$

The system mass matrix $\mathbf{M} \in \mathbb{R}^{2N_x \times 2N_x}$ on the left hand side of the DAE is singular and its block entry $\mathbf{M}_{11} \coloneqq M \coloneqq \big[(\varphi_i,\varphi_j)\big]_{ij}$ denotes the mass matrix of our finite element space V_h. The right hand side function is defined according to the weak formulation (4.5): If $\boldsymbol{y}$ is related to c_h, μ_h as described, we have $\boldsymbol{f}\colon [0,t_\mathrm{f}] \times \mathbb{R}^{2N_x} \to \mathbb{R}^{2N_x}$ with

$$(t,\boldsymbol{y}) \mapsto \boldsymbol{f}(t,\boldsymbol{y}) \coloneqq \begin{pmatrix} \Big(-\big(\nabla\varphi_i, \mathbf{m}(c_h)\nabla\mu_h\big) - \big(\varphi_i, N_\mathrm{ext}\big)_{\partial\Omega}\Big)_{i=1,\dots,N_x} \\ \Big(-\big(\varphi_i,\mu_h\big) + \big(\varphi_i, \psi'_\mathrm{ch}(c_h)\big) + \big(\nabla\varphi_i, \mathbf{K}\nabla c_h\big)\Big)_{i=1,\dots,N_x} \end{pmatrix}. \tag{4.9}$$

Defining the matrices

$$K_\mathbf{m}(\boldsymbol{y}_1) \coloneqq \Big[\big(\nabla\varphi_i, \mathbf{m}(c_h)\nabla\varphi_j\big)\Big]_{ij}, \qquad K_\mathbf{K} \coloneqq \Big[\big(\nabla\varphi_i, \mathbf{K}\nabla\varphi_j\big)\Big]_{ij}, \tag{4.10}$$

the vector for the boundary condition and the nonlinearity

$$\boldsymbol{J} \coloneqq \Big[\big(\varphi_i, N_\mathrm{ext}\big)_{\partial\Omega}\Big]_i, \qquad \boldsymbol{\Psi}_\mathrm{ch}(\boldsymbol{y}_1) \coloneqq \Big[\big(\varphi_i, \psi'_\mathrm{ch}(c_h)\big)\Big]_i, \tag{4.11}$$

we can rewrite $\boldsymbol{f}$ as

$$\boldsymbol{f}(t,\boldsymbol{y}) = \begin{pmatrix} \boldsymbol{f}_1(t,\boldsymbol{y}_1,\boldsymbol{y}_2) \\ \boldsymbol{f}_2(t,\boldsymbol{y}_1,\boldsymbol{y}_2) \end{pmatrix} \coloneqq \begin{pmatrix} -K_\mathbf{m}(\boldsymbol{y}_1)\boldsymbol{y}_2 - \boldsymbol{J} \\ -M\boldsymbol{y}_2 + \boldsymbol{\Psi}_\mathrm{ch}(\boldsymbol{y}_1) + K_\mathrm{K}\boldsymbol{y}_1 \end{pmatrix}. \tag{4.12}$$

Note, that an explicit dependence of $\boldsymbol{f}$ on the time t will only occur through a time-dependent Neumann boundary condition $N_\mathrm{ext}\colon [0,t_\mathrm{f}] \times \partial\Omega \to \mathbb{R}$.

In practice, when we use a p-th-order Lagrangian finite element method, we evaluate the integrals through a Gauß–Legendre quadrature formula with $p+1$ quadrature points in spatial direction. Note, that the assumption of a mesh consisting of quadrilaterals in 2D or hexahedra in 3D is motivated by the implementation of the methods with *deal.II*.

4.1.2. Discretization in Time

The Klopfenstein–Shampine numerical differentiation formulas (NDFs) [129] are based on the backward differentiation formulas (BDFs) [75, 76, 120, 128]. Thereby the NDF family was initially considered by Klopfenstein [94] in order to increase the stability of the higher order BDF methods on the cost of reduced efficiency [129]. Later Shampine and Reichelt [129] used this approach to minimize the leading term of the truncation error, while the stability is only slightly reduced, resulting in the Klopfenstein–Shampine NDFs [129]. Writing about NDFs in the following we will always mean the Klopfenstein–Shampine NDFs. For completeness of this thesis we briefly review the method explained in [129].

In the previous subsection, we formulated the spatially discrete problem (4.5) as generic DAE (4.8), which can be treated similarly to a stiff ordinary differential equation. According to [129], the constant-step NDF(k) method for the DAE (4.8) advance the solution $\boldsymbol{y}^n$

at time $t_n \in [0, t_\mathrm{f} - \tau]$ with time step size $\tau > 0$ to $\boldsymbol{y}^{n+1}$ at time $t_{n+1} = t_n + \tau$ by solving

$$\mathbf{M} \sum_{m=1}^{k} \frac{1}{m} \triangle^m \boldsymbol{y}^{n+1} - \tau \boldsymbol{f}(t_{n+1}, \boldsymbol{y}^{n+1}) - \kappa \gamma_k \mathbf{M} \Big(\boldsymbol{y}^{n+1} - \boldsymbol{y}^{(0),n+1} \Big) = \mathbf{0}. \tag{4.13}$$

Here $\triangle^m$ denotes the recursively defined m-th backward difference

$$\triangle^m \boldsymbol{y}^{n+1} = \triangle^{m-1} (\boldsymbol{y}^{n+1} - \boldsymbol{y}^n), \tag{4.14}$$

for $m = 1, \ldots, k$, with $\triangle^0 \coloneqq Id$. Furthermore, in (4.13) we use the predictor

$$\boldsymbol{y}^{(0),n+1} = \sum_{m=0}^{k} \triangle^m \boldsymbol{y}^n, \tag{4.15}$$

as well as the BDF(k) coefficients $\gamma_k = \sum_{j=1}^{k} 1/j$ and the NDF(k) parameters κ [129], both given in Table 4.1.

Table 4.1.: BDF coefficients [120, 129] and NDF parameters [129] for the NDF(k) method from order one to five.

Order k	BDF γ_k	NDF κ
1	1	-0.1850
2	$3/2$	$-1/9$
3	$11/6$	-0.0823
4	$25/12$	-0.0415
5	$137/60$	0

In the formulation of (4.13) we identify the first two terms as in the BDF(k) method. The third term shows its meaning if we use the identity [129]

$$\boldsymbol{y}^{n+1} - \boldsymbol{y}^{(0),n+1} = \triangle^{k+1} \boldsymbol{y}^{n+1}. \tag{4.16}$$

As the leading term of the truncation error of BDF(k) is approximated by [129]

$$\frac{1}{k+1} \tau^{k+1} \partial_t^{k+1} \boldsymbol{y} \approx \frac{1}{k+1} \triangle^{k+1} \boldsymbol{y}^{n+1}, \tag{4.17}$$

the leading term of the truncation error for NDF(k) becomes [129]

$$\left(\kappa \gamma_k + \frac{1}{k+1} \right) \tau^{k+1} \partial_t^{k+1} \boldsymbol{y}. \tag{4.18}$$

Thus for negative κ, as given in Table 4.1, the error constant for the Klopfenstein–Shampine NDFs is decreased and so the accuracy gets enhanced. Note, that for $\kappa = 0$ we recover the BDF(k) methods. In particular, for the NDF(5) method, $\kappa = 0$ was chosen in [129] so that it coincides with the BDF(5) method.

Rearranging the terms according to [129, Sect. 2.3] and multiplying the mass matrix M in the first component back to the left hand side, we obtain with the original right hand side function $\boldsymbol{f}$, defined in (4.12), the fully discrete system for the NDF(k) methods of order one to five as

$$\mathbf{M}\left((1-\kappa)\gamma_k\left(\boldsymbol{y}^{n+1}-\boldsymbol{y}^{(0),n+1}\right)+\sum_{m=1}^{k}\gamma_m\triangle^m\boldsymbol{y}^n\right)-\tau\boldsymbol{f}(t_{n+1},\boldsymbol{y}^{n+1})=\mathbf{0}. \tag{4.19}$$

With the summarized quantity

$$\boldsymbol{\Phi}_k^n \coloneqq \boldsymbol{y}^{(0),n+1}-\frac{1}{(1-\kappa)\gamma_k}\sum_{m=1}^{k}\gamma_m\triangle^m\boldsymbol{y}^n, \tag{4.20}$$

depending only on some previous time steps $\boldsymbol{y}^n,\dots,\boldsymbol{y}^{n-k}$, and the *generalized time step size*

$$\xi_k \coloneqq \frac{\tau}{(1-\kappa)\gamma_k}, \tag{4.21}$$

we simplify the general NDF formulation (4.19) and thus obtain the fully discrete system

$$\mathbf{M}\big(\boldsymbol{y}^{n+1}-\boldsymbol{\Phi}_k^n\big)-\xi_k\boldsymbol{f}\big(t_{n+1},\boldsymbol{y}^{n+1}\big)=\mathbf{0}. \tag{4.22}$$

Note the similarity to the implicit Euler scheme of this notation.

4.2. Solving the Discrete Problem

During the time integration of the Cahn–Hilliard equation, we solve in each time step t_{n+1} the nonlinear system of equations (4.22) for the approximate solution $\boldsymbol{y}^{n+1}\approx\boldsymbol{y}(t_{n+1})$. Hereby, we transform the discrete system (4.22) for the later applied preconditioner and employ Newton's method. The linearized systems are then solved with a preconditioned GMRES method. In particular, based on the previous work of Brenner et al. [31], we propose our block preconditioner, which can be applied matrix-free, i. e., without assembling and storing matrices. As an extension, our preconditioner now additionally accounts for the nonlinear mobility (3.12) of the Cahn–Hilliard model equations (3.19).

We recall the spatially discrete problem (4.5) as the origin of the fully discrete equation (4.22). Hence, the components of solution vector $\boldsymbol{y}^n$ correspond to the finite element functions for the concentration $\boldsymbol{y}_1^n\cong c_h^n\approx c(t_n,\,\cdot\,)$ and the chemical potential $\boldsymbol{y}_2^n\cong\mu_h^n\approx\mu(t_n,\,\cdot\,)$ at the discrete time steps t_n.

4.2.1. Transformation

The fully discrete equation (4.22) consists of the two algebraic equations

$$M\big(\boldsymbol{y}_1^{n+1}-\boldsymbol{\Phi}_{k,1}^n\big)+\xi_k\big(K_{\mathbf{m}}(\boldsymbol{y}_1^{n+1})\boldsymbol{y}_2^{n+1}+\boldsymbol{J}\big)=\mathbf{0}, \tag{4.23}$$

$$M\boldsymbol{y}_2^{n+1}-\boldsymbol{\Psi}_{\mathrm{ch}}(\boldsymbol{y}_1^{n+1})-K_{\mathbf{K}}\boldsymbol{y}_1^{n+1}=\mathbf{0}. \tag{4.24}$$

Based on the transformation used in Brenner et al. [31] we switch the ordering of vector components of $\boldsymbol{y}^{n+1} = \left[\boldsymbol{y}_1^{n+1}, \boldsymbol{y}_2^{n+1}\right]^\mathsf{T}$ and rescale the blocks with a fraction of the generalized time step size (4.21)

$$\boldsymbol{z}_1^{n+1} \coloneqq \xi_k^{1/4}\boldsymbol{y}_2^{n+1} \qquad \boldsymbol{z}_2^{n+1} \coloneqq \xi_k^{-1/4}\boldsymbol{y}_1^{n+1} \qquad \boldsymbol{\Upsilon}_{k,2}^n \coloneqq \xi_k^{-1/4}\boldsymbol{\Phi}_{k,1}^n. \tag{4.25}$$

Note, that from the data vector $\boldsymbol{\Phi}_k^n$, we only need the first component due to the DAE character of (4.22). Inserting these transformed variables into (4.23) and (4.24), the multiplication of the first equation (4.23) with $\xi_k^{-1/4}$ and of the second equation (4.24) with $\xi_k^{1/4}$ yields the transformed discrete algebraic equations

$$M\big(\boldsymbol{z}_2^{n+1} - \boldsymbol{\Upsilon}_{k,2}^n\big) + \xi_k^{1/2} K_{\mathbf{m}}\big(\xi_k^{1/4}\boldsymbol{z}_2^{n+1}\big)\boldsymbol{z}_1^{n+1} + \xi_k^{3/4}\boldsymbol{J} = \mathbf{0}, \tag{4.26}$$

$$M\boldsymbol{z}_1^{n+1} - \xi_k^{1/4}\boldsymbol{\Psi}_{\mathrm{ch}}\big(\xi_k^{1/4}\boldsymbol{z}_2^{n+1}\big) - \xi_k^{1/2} K_{\mathbf{K}}\boldsymbol{z}_2^{n+1} = \mathbf{0}. \tag{4.27}$$

According to these two equations, we define for a fixed time, e. g. t_{n+1}, the nonlinear function $\boldsymbol{F}\colon \mathbb{R}^{2N_x} \to \mathbb{R}^{2N_x}$

$$\boldsymbol{F}\Big(\left[\boldsymbol{v}_1, \boldsymbol{v}_2\right]^\mathsf{T}\Big) \coloneqq \begin{pmatrix} M\big(\boldsymbol{v}_2 - \boldsymbol{\Upsilon}_{k,2}^n\big) + \xi_k^{1/2} K_{\mathbf{m}}\big(\xi_k^{1/4}\boldsymbol{v}_2\big)\boldsymbol{v}_1 + \xi_k^{3/4}\boldsymbol{J} \\ M\boldsymbol{v}_1 - \xi_k^{1/4}\boldsymbol{\Psi}_{\mathrm{ch}}\big(\xi_k^{1/4}\boldsymbol{v}_2\big) - \xi_k^{1/2} K_{\mathbf{K}}\boldsymbol{v}_2 \end{pmatrix}, \tag{4.28}$$

such that the root of this function corresponds to the discrete solution $\boldsymbol{y}^{n+1}$ at the time step t_{n+1} via the transformed variables (4.25).

4.2.2. Linearization

We employ Newton's method for the solution of the root problem

$$\boldsymbol{F}(\boldsymbol{v}) \stackrel{!}{=} \mathbf{0}. \tag{4.29}$$

The residual is defined by (4.28) and the Jacobian $J_{\boldsymbol{F}}(\boldsymbol{v})\colon \mathbb{R}^{2N_x} \to \mathbb{R}^{2N_x,2N_x}$ is given as follows

$$J_{\boldsymbol{F}}\Big(\left[\boldsymbol{v}_1, \boldsymbol{v}_2\right]^\mathsf{T}\Big) = \begin{pmatrix} \xi_k^{1/2} K_{\mathbf{m}}\big(\xi_k^{1/4}\boldsymbol{v}_2\big) & M + \xi_k^{3/4} K'_{\mathbf{m}}\big(\xi_k^{1/4}\boldsymbol{v}_2, \boldsymbol{v}_1\big) \\ M & -\xi_k^{1/2}\boldsymbol{\Psi}'_{\mathrm{ch}}\big(\xi_k^{1/4}\boldsymbol{v}_2\big) - \xi_k^{1/2} K_{\mathbf{K}} \end{pmatrix}, \tag{4.30}$$

where we use the abbreviations

$$K'_{\mathbf{m}}\big(\xi_k^{1/4}\boldsymbol{v}_2, \boldsymbol{v}_1\big) \coloneqq \left[\Big(\nabla\varphi_i, \mathbf{m}\big(\xi_k^{1/4} v_h^{(2)}\big)\nabla v_h^{(1)}\varphi_j\Big)\right]_{ij}, \tag{4.31}$$

$$\boldsymbol{\Psi}'_{\mathrm{ch}}\big(\xi_k^{1/4}\boldsymbol{v}_2\big) \coloneqq \left[\big(\varphi_i, \psi''_{\mathrm{ch}}\big(\xi_k^{1/4} v_h^{(2)}\big)\varphi_j\big)\right]_{ij}. \tag{4.32}$$

Hereby we identified the vector components with the finite element functions $\boldsymbol{v}_i \cong v_h^{(i)}$ for $i \in \{1, 2\}$ and inserted those into the arguments of the nonlinearities $\mathbf{m}$, ψ''_{ch}. Especially the expression $\xi_k^{1/4}\boldsymbol{v}_2$ represents the concentration c_h.

In order to solve for the root $\boldsymbol{z}^{n+1}$ of $\boldsymbol{F}$, which corresponds to the approximate solution $\boldsymbol{y}^{n+1}$, Newton's method takes an initial guess $\boldsymbol{z}^{(0)} \in \mathbb{R}^{2N_x}$ and iterates over the following two steps

1. Solve for $\boldsymbol{d}^{(i)} \in \mathbb{R}^{2N_x}$

$$J_{\boldsymbol{F}}\big(\boldsymbol{z}^{(i)}\big)\boldsymbol{d}^{(i)} = -\boldsymbol{F}\big(\boldsymbol{z}^{(i)}\big), \tag{4.33}$$

2. Update

$$\boldsymbol{z}^{(i+1)} = \boldsymbol{z}^{(i)} + \boldsymbol{d}^{(i)}, \tag{4.34}$$

until either the residual $\left|\boldsymbol{F}\big(\boldsymbol{z}^{(i+1)}\big)\right| \leq \mathrm{TOL}_f$ or the update $\left|\boldsymbol{d}^{(i)}\right| \leq \mathrm{TOL}_x$ fulfill the given tolerances.

4.2.3. Linear Solver

For the solution of the arising linear systems (4.33) we use the GMRES method and a right preconditioner. Based on the derivation in [31] our extended preconditioner reads as

$$P\big(\boldsymbol{z}^{(i)}\big) = \begin{pmatrix} \xi_k^{1/2} K_{\mathbf{m}}\big(\xi_k^{1/4}\boldsymbol{z}_2^{(i)}\big) + M & 0 \\ 0 & \xi_k^{1/2} K_{\mathbf{K}} + M \end{pmatrix}, \tag{4.35}$$

and depends nonlinearly on the Newton step $\boldsymbol{z}_2^{(i)}$. The key advantages of our preconditioner are the incorporation of nonlinear mobility and the usability for matrix-free computations.

As we will observe later in our numerical experiments in Chapter 6, the inclusion of the nonlinear mobility has a significant impact on the maximum number of GMRES steps.

To apply the preconditioner, only the two blocks have to be inverted. As the lower block and, if the mobility is not degenerated, e. g. by assumption $c \in (0, 1)$, also the upper block are symmetric and positive definite, the inverse application of the blocks can be calculated by a preconditioned CG method, compare [31]. If we use a matrix-free preconditioner for the inverse application of the blocks by the CG method, neither the outer GMRES nor the inner CG methods require stored matrices. This allows us to use the matrix-free finite element operator application framework within *deal.II* for the efficient solution of the linearized problems. In practice, we use for the inner CG methods a geometric multigrid preconditioner based on a Chebyshev smoother, like presented in *deal.II* tutorial `step-37` [5, 98] and used for nonlinear problems in Section 5.2 of this thesis. However, especially for the upper block with the nonlinear mobility, we observe that the CG method, preconditioned with multigrid, may exceeds 50 iteration steps. In such cases we stop the CG solver and restart it with a damped Jacobi preconditioner, which can also be used matrix-free. We also observe that the number of iteration steps of the multigrid preconditioned CG method is reduced when we evaluate the mobility with the constant concentration value 0.5.

Optimal Preconditioning Although the mathematical analysis of the Cahn-Hilliard equation with logarithmic free energy and degenerate mobility is complicated, see for instance [16, 64], we want to extend the ideas of Brenner et al. [31, 32] at least by a constant mobility $m \geq 1$. In this case we show generalized eigenvalue estimates for a variation of our proposed preconditioner (4.35) in the sense of Brenner et al. [31, 32].

As Brenner et al. [31], we consider the classical Cahn–Hilliard equation with quartic free energy density and extend the problem setting by a constant mobility.

Problem 2. Let $\Omega \subset \mathbb{R}^d$ be a polytop. Find $c\colon \mathbb{R}_{\geq 0} \times \overline{\Omega} \to \mathbb{R}$ and $\mu\colon \mathbb{R}_{\geq 0} \times \overline{\Omega} \to \mathbb{R}$ satisfying

$$\begin{cases} \partial_t c = m\Delta\mu & \text{in } \mathbb{R}_{\geq 0} \times \Omega, \\ \mu = f'(c) - \kappa\Delta c & \text{in } \mathbb{R}_{\geq 0} \times \Omega, \\ \nabla c \cdot \boldsymbol{n} = 0 & \text{on } \mathbb{R}_{\geq 0} \times \partial\Omega, \\ \nabla \mu \cdot \boldsymbol{n} = 0 & \text{on } \mathbb{R}_{\geq 0} \times \partial\Omega, \\ c(0, \cdot\,) = c_0 & \text{in } \Omega, \end{cases} \tag{4.36}$$

with the constant mobility $m \geq 1$, the quartic free energy density

$$f(c) = \frac{1}{4}(c^2 - 1)^2, \qquad f'(c) = c^3 - c, \tag{4.37}$$

the interfacial parameter $\kappa > 0$ and a given initial condition $c_0\colon \overline{\Omega} \to \mathbb{R}$, consistent with the boundary conditions.

We follow the course of Brenner et al. [31] for the discretization in space and time with the linear finite element method and the convex splitting scheme based on Eyre [68]. Further, the linearization with Newton's method, the transformation of the variables and the workaround of the mean value constraint leads us in each Newton step of every time step to the following system matrix, compare [31, Eq. 3.7]

$$S = \begin{pmatrix} \tau^{1/2} m\big(K + \boldsymbol{c}\boldsymbol{c}^{\mathsf{T}}\big) & M \\ M & -\tau^{1/2} J(c_{h,j}^m) - \tau^{1/2}\kappa\big(K + \boldsymbol{c}\boldsymbol{c}^{\mathsf{T}}\big) \end{pmatrix}, \tag{4.38}$$

where τ is the time step size, $\boldsymbol{c} \coloneqq \big[(\phi_i, 1)\big]_i$ is the vector introduced due to the mean value constraint, M and K are the mass and stiffness matrices of the finite element space and

$$J(c_{h,j}^m) \coloneqq \Big[\big(\varphi_i, 3(c_{h,j}^m)^2 \varphi_j\big)\Big]_{ij}, \tag{4.39}$$

is the block matrix corresponding to the second derivative of the convex part of the free energy density $f_{\text{convex}}(c) = 1/4\, c^4 + 1$, which depends on the outcome of the last Newton step for the concentration $c_{h,j}^m$, here the j-th Newton step for the m-th time step.

In regions away from the narrow phase transition zone, the concentration is approximately constant and approximately equal to one of the two equilibrium values c_α, c_β, which are characterized by the local minima of the free energy density ($f'(c_\alpha) = f'(c_\beta) = 0$). According to [31] we approximate the nonlinear part $J(c_{h,j}^m)$ of the matrix (4.38), however, for axis symmetric free energy densities, we allow a more general approximation through $f_{\text{eq}} \coloneqq f''_{\text{convex}}(c_\alpha) = f''_{\text{convex}}(c_\beta)$

$$J(c_{h,j}^m) = \Big[\big(\varphi_i, f''_{\text{convex}}(c_{h,j}^m)\varphi_j\big)\Big]_{ij} \approx f_{\text{eq}} M. \tag{4.40}$$

Note, that the equilibrium concentrations c_α, c_β depend on the underlying free energy density. For example for the quartic expression (4.37) these equilibrium values are given by $c_{\alpha,\beta} = \pm 1$.

Now we can formulate a generalized version of [31, Thm. 3.2].

Theorem 4.1. *Assume* $1 \leq e \leq f_{\mathrm{eq}}$ *and* $m \geq 1$. *Further, let* $0 < \tau \leq 1$ *and* $0 < \kappa \leq 1$. *Let the matrices* B *and* P *be defined by*

$$B = \begin{pmatrix} \tau^{1/2} m(K + \boldsymbol{c}\boldsymbol{c}^{\mathsf{T}}) & M \\ M & -\tau^{1/2} f_{\mathrm{eq}} M - \tau^{1/2}\kappa(K + \boldsymbol{c}\boldsymbol{c}^{\mathsf{T}}) \end{pmatrix}, \tag{4.41}$$

$$P = \begin{pmatrix} \tau^{1/2} m(K + \boldsymbol{c}\boldsymbol{c}^{\mathsf{T}}) + M & 0 \\ 0 & eM + \tau^{1/2}\kappa(K + \boldsymbol{c}\boldsymbol{c}^{\mathsf{T}}) \end{pmatrix}. \tag{4.42}$$

Then for any eigenvalue λ *of* $P^{-1}B$ *it holds*

$$\frac{1}{2(e + f_{\mathrm{eq}})} \max\left\{ \frac{\sqrt{\kappa}}{\sqrt{em}}, \sqrt{\tau} \right\} \leq |\lambda| \leq 1 + \frac{f_{\mathrm{eq}}}{e}. \tag{4.43}$$

Proof. With minor modifications, the proof essentially follows the one in [31].

With the definition $\widetilde{K} = M^{-1}(K + \boldsymbol{c}\boldsymbol{c}^{\mathsf{T}})$ and Id for the $N_x \times N_x$ identity matrix, we calculate

$$\begin{aligned} P^{-1}B = & \left[\begin{pmatrix} M & 0 \\ 0 & M \end{pmatrix} \begin{pmatrix} \tau^{1/2} m\widetilde{K} + Id & 0 \\ 0 & eId + \tau^{1/2}\kappa\widetilde{K} \end{pmatrix} \right]^{-1} \\ & \times \left[\begin{pmatrix} M & 0 \\ 0 & M \end{pmatrix} \begin{pmatrix} \tau^{1/2} m\widetilde{K} & Id \\ Id & -\tau^{1/2} f_{\mathrm{eq}} Id - \tau^{1/2}\kappa\widetilde{K} \end{pmatrix} \right] \\ = & \begin{pmatrix} \tau^{1/2} m\widetilde{K} + Id & 0 \\ 0 & eId + \tau^{1/2}\kappa\widetilde{K} \end{pmatrix}^{-1} \\ & \times \begin{pmatrix} \tau^{1/2} m\widetilde{K} & Id \\ Id & -\tau^{1/2} f_{\mathrm{eq}} Id - \tau^{1/2}\kappa\widetilde{K} \end{pmatrix}. \end{aligned}$$

From the generalized eigenvalue problem $(K + \boldsymbol{c}\boldsymbol{c}^{\mathsf{T}})\boldsymbol{v} = \mu M\boldsymbol{v}$ we obtain N_x positive eigenvalues $\mu_j > 0$ and corresponding eigenvectors $\boldsymbol{v}_j \in \mathbb{R}^{N_x}$, satisfying for all $j \in \{1, \ldots, N_x\}$

$$\widetilde{K}\boldsymbol{v}_j = \mu_j \boldsymbol{v}_j,$$

and are orthogonal with respect to the mass matrix

$$\boldsymbol{v}_i^{\mathsf{T}} M \boldsymbol{v}_j = \delta_{i,j}.$$

Further, we observe that for an eigenvector $\boldsymbol{v}_j$ the two-dimensional subspace spanned by $[\boldsymbol{v}_j, 0]^{\mathsf{T}}$ and $[0, \boldsymbol{v}_j]^{\mathsf{T}}$ is invariant under $P^{-1}B$. More precisely for α, β, γ, $\delta \in \mathbb{R}$ it holds

$$B\left[\alpha \begin{pmatrix} \boldsymbol{v}_j \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ \boldsymbol{v}_j \end{pmatrix}\right] = P\left[\gamma \begin{pmatrix} \boldsymbol{v}_j \\ 0 \end{pmatrix} + \delta \begin{pmatrix} 0 \\ \boldsymbol{v}_j \end{pmatrix}\right],$$

which leads to the relation

$$\begin{pmatrix} \tau^{1/2} m\mu_j & 1 \\ 1 & -\tau^{1/2} f_{\mathrm{eq}} - \tau^{1/2}\kappa\mu_j \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \tau^{1/2} m\mu_j + 1 & 0 \\ 0 & \tau^{1/2}\kappa\mu_j + e \end{pmatrix} \begin{pmatrix} \gamma \\ \delta \end{pmatrix}.$$

Accordingly, the eigenvalues λ of $P^{-1}B$ are identical to the eigenvalues of the 2×2 matrices

$$\begin{pmatrix} \tau^{1/2} m\mu_j + 1 & 0 \\ 0 & \tau^{1/2}\kappa\mu_j + e \end{pmatrix}^{-1} \times \begin{pmatrix} \tau^{1/2} m\mu_j & 1 \\ 1 & -\tau^{1/2} f_{\mathrm{eq}} - \tau^{1/2}\kappa\mu_j \end{pmatrix} = \begin{pmatrix} \frac{\tau^{1/2} m\mu_j}{\tau^{1/2} m\mu_j + 1} & \frac{1}{\tau^{1/2} m\mu_j + 1} \\ \frac{1}{\tau^{1/2}\kappa\mu_j + e} & -\frac{\tau^{1/2} f_{\mathrm{eq}} + \tau^{1/2}\kappa\mu_j}{\tau^{1/2}\kappa\mu_j + e} \end{pmatrix},$$

for each $j \in \{1, \dots, N_x\}$. For better readability we define for each eigenvalue μ_j of the matrix $\widetilde{K}$ the positive value $\omega := \tau^{1/2}\mu_j$ and the corresponding matrix

$$C_j = \begin{pmatrix} \frac{m\omega}{m\omega+1} & \frac{1}{m\omega+1} \\ \frac{1}{\omega\kappa+e} & -\frac{\tau^{1/2} f_{\mathrm{eq}} + \kappa\omega}{\omega\kappa+e} \end{pmatrix}.$$

It is now sufficient to estimate the eigenvalues $\lambda_{j,1}$, $\lambda_{j,2}$ of these matrices C_j. This is done using the following facts:

- Let $w_{j,k} \in \mathbb{R}^2 \setminus \{0\}$ be the corresponding eigenvector to the eigenvalue $\lambda_{j,k}$ for $k \in \{1, 2\}$ in context to the matrices C_j. For each induced matrix norm $\|\cdot\|$ it holds

  $$|\lambda_{j,k}| \|w_{j,k}\| = \|\lambda_{j,k} w_{j,k}\| = \|C_j w_{j,k}\| \leq \|C_j\| \|w_{j,k}\|,$$

  and hence $|\lambda_{j,k}| \leq \|C_j\|$.

- Further, it holds

  $$|\det(C_j)| = |\lambda_{j,1}\lambda_{j,2}|.$$

Now we start estimating: With the $\|\cdot\|_\infty$ norm we get

$$\|C_j\|_\infty \leq \left\{ \begin{array}{l} 1 \\ \frac{1+\tau^{1/2} f_{\mathrm{eq}} + \kappa\omega}{\omega\kappa+e} \leq \frac{e + f_{\mathrm{eq}} + \kappa\omega}{\omega\kappa+e} \leq \frac{f_{\mathrm{eq}}}{e} + 1 \end{array} \right\} \leq 1 + \frac{f_{\mathrm{eq}}}{e}.$$

Thus, we directly determined the upper bound for the eigenvalues. Next, for the lower bound, we estimate the absolute value of the determinant

$$\begin{aligned} |\det(C_j)| &= \frac{m\omega(\tau^{1/2} f_{\mathrm{eq}} + \omega\kappa) + 1}{(1 + m\omega)(e + \kappa\omega)} \\ &= \frac{1 + m\omega\tau^{1/2} f_{\mathrm{eq}} + m\kappa\omega^2}{e + (\kappa + me)\omega + m\kappa\omega^2} \\ &\geq \frac{1 + m\omega\tau^{1/2} f_{\mathrm{eq}} + m\kappa\omega^2}{e + 2me\omega + m\kappa\omega^2}. \qquad (\kappa < 1 \leq me) \end{aligned}$$

For this we first estimate the numerator of the fraction as follows

$$\begin{aligned} 1 + m\omega\tau^{1/2} f_{\mathrm{eq}} + m\kappa\omega^2 &= \frac{\tau^{1/2}}{2e}\left(2e\tau^{-1/2} + 2em\omega f_{\mathrm{eq}} + 2e\tau^{-1/2} m\kappa\omega^2\right) \\ &\geq \frac{\tau^{1/2}}{2e}\left(e + 2em\omega + m\kappa\omega^2\right), \qquad (\tau^{-1/2}, f_{\mathrm{eq}} \geq 1) \end{aligned}$$

which gives us $|\det(C_j)| \geq \tau^{1/2}/2e$. Alternatively, we estimate the numerator of the fraction as follows

$$\begin{aligned}
1 + m\omega\tau^{1/2} f_{\text{eq}} + m\kappa\omega^2 &= \frac{1}{e}\left(e + emf_{\text{eq}}\tau^{1/2} + em\kappa\omega^2\right) \\
&\geq \frac{1}{e}\left(e + m\kappa\omega^2\right) \\
&\geq \frac{1}{2e}\left(e + m\kappa\omega^2 + 2\sqrt{em\kappa}\omega\right) && (2ab \leq a^2 + b^2) \\
&= \frac{1}{2e}\left(e + m\kappa\omega^2 + 2\frac{em\sqrt{\kappa}\omega}{\sqrt{em}}\right) \\
&\geq \frac{1}{2e}\left(\frac{\sqrt{\kappa}}{\sqrt{em}}e + \frac{\sqrt{\kappa}}{\sqrt{em}}m\kappa\omega^2 + 2\frac{em\sqrt{\kappa}\omega}{\sqrt{em}}\right) && (\sqrt{\kappa/em} < 1) \\
&\geq \frac{\sqrt{\kappa}}{2e\sqrt{em}}\left(e + 2em\omega + m\kappa\omega^2\right),
\end{aligned}$$

which leads us to $|\det(C_j)| \geq \sqrt{\kappa}/\left(2e\sqrt{em}\right)$. Together, we finally estimate the absolute value of the determinant by

$$|\det(C_j)| \geq \frac{1}{2e}\max\left\{\frac{\sqrt{\kappa}}{\sqrt{em}}, \sqrt{\tau}\right\},$$

and with the above upper bound for the eigenvalues we obtain for $k \in \{1, 2\}$

$$\frac{1}{2e}\max\left\{\frac{\sqrt{\kappa}}{\sqrt{em}}, \sqrt{\tau}\right\} \leq |\det(C_j)| = |\lambda_{j,1}\lambda_{j,2}| \leq |\lambda_{j,k}|\left(1 + \frac{f_{\text{eq}}}{e}\right),$$

leading to the lower bound for the eigenvalues

$$\frac{1}{2(e + f_{\text{eq}})}\max\left\{\frac{\sqrt{\kappa}}{\sqrt{em}}, \sqrt{\tau}\right\} \leq |\lambda_{j,k}|.$$

This brings us finally to the estimation of the eigenvalues λ of $P^{-1}B$

$$\frac{1}{2(e + f_{\text{eq}})}\max\left\{\frac{\sqrt{\kappa}}{\sqrt{em}}, \sqrt{\tau}\right\} \leq |\lambda| \leq 1 + \frac{f_{\text{eq}}}{e}.$$

□

Remark 4.2. The parameter $e \in [1, f_{\text{eq}}]$ gives the additional freedom to finetune the preconditioner. We recover the original case of Brenner et al. [31] if we set the mobility $m = 1$ and use $f_{\text{eq}} = 3$ and $e = 1$.

Remark 4.3. As noted by Brenner et al. [31], one can also replace the precondition matrix P (4.42) by a spectrally equivalent matrix P_* and obtain a similar eigenvalue estimate

$$C_1 \max\left\{\frac{\sqrt{\kappa}}{\sqrt{m}}, \sqrt{\tau}\right\} \leq |\lambda| \leq C_2\,, \tag{4.44}$$

with two constants C_1, $C_2 > 0$ independent of τ, h, κ and m. In particular, our proposed preconditioner (4.35) with constant mobility would be such a spectrally equivalent matrix

in this context. In the scalar case and with time integration based on NDFs, our preconditioner (4.35) simplifies to

$$P_\star = \begin{pmatrix} \xi_k^{1/2} mK + M & 0 \\ 0 & \xi_k^{1/2} \kappa K + M \end{pmatrix}. \tag{4.45}$$

Remark 4.4. When we consider the Cahn–Hilliard equation with a concentration dependent mobility, like $m(c) = Dc(1-c)$ with $D > 0$ as in the scalar version of (3.12), the incorporation of this nonlinear term in the preconditioner has a significant impact on the maximum number of GMRES steps per time step. For this, we will compare the two preconditioners (4.35) and (4.45) in the case of concentration dependent mobility in our numerical experiments in Chapter 6.

4.3. The Space and Time Adaptive Solution Algorithm

Finally, in this section, we present our space and time adaptive solution algorithm. The core of the algorithm is based on the variable-step, variable-order time integration scheme, which is in *MATLAB* known as `ode15s`, see [129, 130, 143]. For our problem we have adapted this algorithm by adding the spatial adaptivity and implemented it in `C++` based on the functionalities of the open source finite element library *deal.II* [5, 13].

In the following we describe our adaptive Algorithm 1 and explain some details.

Description We initialize the time integration algorithm based on multistep methods by starting from order one and computing the first three steps with the BDF(1) method (=implicit Euler) without adaptivity and a small time step size. Then, for the further description of the algorithm, we assume that the current mesh $\mathcal{T}_n$, the numerical solutions at previous time steps $\boldsymbol{y}^n, \ldots, \boldsymbol{y}^{n-k_n}$ and candidates for the time steps size τ_n and the order k_n are given.

In Line 2 we solve the nonlinear system of the fully discrete problem (4.22) with a Newton–Raphson method, as described in Section 4.2. The solution of the linearized systems is a critical issue addressed in Subsection 4.2.3. For small problems with less than 10^4 DOFs we can also solve the linear systems with the LU-decomposition [52], however, for an increasing problem size this becomes infeasible due to the large memory requirements. Therefore we employ the GMRES method [123] with our proposed preconditioner (4.35) and an explicitly zero start vector. Note, however, that the GMRES method includes the restart after a certain number of iterations. This number is controlled by the size of the Arnoldi basis. While in *deal.II* [5] the default value of 30 is suggested, we use 300 instead. The influence of the size of the Arnoldi basis on the number of GMRES steps is discussed later in Subsection 6.1.2. It should be mentioned here that in the matrix-free framework, the large number of basis vectors to be stored is not critical, because we avoid the more memory-intensive storage of matrices. Note further, that the time step size is reduced [129, 143] if the Newton method does not converge within four steps. The tolerances for the outer GMRES and inner CG methods are set at 10^{-12}, while the convergence of the Newton method is measured using the tactics from `ode15s` [129, 130, 143].

Algorithm 1 Space and time adaptive solution algorithm.

1: Given $\mathcal{T}_n$, τ_n, k_n and $\boldsymbol{y}^n, \ldots, \boldsymbol{y}^{n-k_n}$
2: Solve for $\boldsymbol{y}^{n+1}$
3: Estimate time error err_t and spatial regularity est_x
4: **if** $\omega_t \, \mathrm{err}_t < \mathrm{RelTol}_t$ **and** $\omega_x \, \mathrm{est}_x < \mathrm{RelTol}_x$ **then**
5: Continue with Line 15
6: **else**
7: **if** $\omega_t \, \mathrm{err}_t > \mathrm{RelTol}_t$ **then**
8: Adapt time step size τ_n and order k_n
9: **end if**
10: **if** $\omega_x \, \mathrm{est}_x > \mathrm{RelTol}_x$ **then**
11: Mark cells and refine mesh $\mathcal{T}_n$
12: **end if**
13: Go to Line 2
14: **end if**
15: Update solution $\boldsymbol{y}^{n+1} \to \boldsymbol{y}^n$ and time $t_{n+1} = t_n + \tau_n \to t_n$
16: **if** a sufficient number of time steps were accepted try coarsening **then**
17: Adapt time step size τ_n and order k_n
18: Mark cells and coarsen mesh $\mathcal{T}_n$
19: **end if**
20: Advance time step

The estimation of the temporal error in Line 3 is based on the leading term of the local truncation error of the NDF(k) method [129, 130, 143]. According to [129] we recall the identity

$$\triangle^{k+1} \boldsymbol{y}^{n+1} = \boldsymbol{y}^{n+1} - \boldsymbol{y}^{(0),n+1}, \tag{4.46}$$

and together with the approximation (4.17) we obtain an approximation of the leading term of the truncation error of the NDF(k) method (4.18)

$$\mathrm{err}_t \coloneqq \left(\kappa \gamma_k + \frac{1}{k+1} \right) \triangle^{k+1} \boldsymbol{y}^{n+1} \approx \left(\kappa \gamma_k + \frac{1}{k+1} \right) \tau^{k+1} \partial_t^{k+1} \boldsymbol{y}, \tag{4.47}$$

which we use as temporal error estimate. As spatial refinement criterion we use a gradient recovery estimator, which we explain in the subsequent paragraph in more detail.

Note, in the Lines 4, 7 and 10 we realize a mixed error control with relative (RelTol_t, RelTol_x) and absolute (AbsTol_t, AbsTol_x) tolerances, compare [128, Chap. 1.4], by multiplying the temporal error and the spatial regularity estimates of Line 3 with the weights

$$\omega_t = 1 / \max\Big\{ \max\big\{ |\boldsymbol{y}^n|, |\boldsymbol{y}^{(0),n+1}| \big\}, \mathrm{AbsTol}_t \,/\, \mathrm{RelTol}_t \Big\}, \tag{4.48}$$

$$\omega_x = 1 / \max\Big\{ \max\big\{ |\boldsymbol{y}^n|, |\boldsymbol{y}^{(0),n+1}| \big\}, \mathrm{AbsTol}_x \,/\, \mathrm{RelTol}_x \Big\}, \tag{4.49}$$

according to [143, `ode15s`]. To illustrate the basic idea, we consider the simplified weight $\widetilde{\omega}_t = 1 / \max\big\{ |\boldsymbol{y}^n|, \mathrm{AbsTol}_t \,/\, \mathrm{RelTol}_t \big\}$, which yields the estimate

$$\mathrm{err}_t < \frac{1}{\widetilde{\omega}_t} \mathrm{RelTol}_t = \max\Big\{ \mathrm{RelTol}_t \, |\boldsymbol{y}^n|, \mathrm{AbsTol}_t \Big\} < \mathrm{RelTol}_t \, |\boldsymbol{y}^n| + \mathrm{AbsTol}_t, \tag{4.50}$$

for the success condition of the temporal error in Line 4.

In Line 8 and 17 we change the time step size τ_n and order k_n adaptively according to the variable-step, variable-order time integration algorithm from [129, 130, 143]. For example, if we reject the solution computed with the current time step size τ_n and order $k_n > 1$, because the tolerance in Line 7 is not satisfied, we determine a new time step size and order based on the temporal error estimates for the current order $\mathrm{err}_t^{k_n}$ and for the lower order $\mathrm{err}_t^{k_n-1}$. In the following we explain the key steps of the adaption of the time step size and the order. We first compute the proposed new time step sizes of the current and the lower order method, which are given by

$$\tau_{\text{new}}^{k_n} = \tau_n \max\left\{0.1, \frac{1}{1.2}\left(\frac{\mathrm{RelTol}_t}{\omega_t \,\mathrm{err}_t^{k_n}}\right)^{\frac{1}{k_n+1}}\right\}, \tag{4.51}$$

$$\tau_{\text{new}}^{k_n-1} = \tau_n \max\left\{0.1, \frac{1}{1.3}\left(\frac{\mathrm{RelTol}_t}{\omega_t \,\mathrm{err}_t^{k_n-1}}\right)^{\frac{1}{k_n}}\right\}. \tag{4.52}$$

Then, if $\tau_{\text{new}}^{k_n-1} > \tau_{\text{new}}^{k_n}$, i. e., the proposed time step size of the lower order method is larger than the one of the current order method, we decrease the order $k_n \to k_n - 1$ and set the candidate of the new time step size to $\tau_{\text{new}} = \tau_{\text{new}}^{k_n-1}$. Finally, if the new candidate is not larger than the currently used time step size we decrease the time step size, i. e., $\tau_n \to \min\{\tau_n, \tau_{\text{new}}\}$. On the other hand, if $\tau_{\text{new}}^{k_n-1} \leq \tau_{\text{new}}^{k_n}$ we keep the current order k_n and set $\tau_n \to \tau_{\text{new}}^{k_n}$. In the case $k_n = 1$, when no lower order method is available, we directly set $\tau_n \to \tau_{\text{new}}^{k_n}$.

In a similar way, we try to enlarge time step size and adapt the order in Line 17 if the algorithm has accepted $k_n + 2$ time steps in a row. If the current order is $k_n \in \{2, 3, 4\}$, we compute the proposed new time step sizes for the current k_n, the lower $k_n - 1$ and the higher $k_n + 1$ order method according to

$$\tau_{\text{new}}^{k_n+1} = \tau_n \min\left\{10, \frac{1}{1.4}\left(\frac{\mathrm{RelTol}_t}{\omega_t \,\mathrm{err}_t^{k_n+1}}\right)^{\frac{1}{k_n+2}}\right\}, \tag{4.53}$$

$$\tau_{\text{new}}^{k_n} = \tau_n \min\left\{10, \frac{1}{1.2}\left(\frac{\mathrm{RelTol}_t}{\omega_t \,\mathrm{err}_t^{k_n}}\right)^{\frac{1}{k_n+1}}\right\}, \tag{4.54}$$

$$\tau_{\text{new}}^{k_n-1} = \tau_n \min\left\{10, \frac{1}{1.3}\left(\frac{\mathrm{RelTol}_t}{\omega_t \,\mathrm{err}_t^{k_n-1}}\right)^{\frac{1}{k_n}}\right\}. \tag{4.55}$$

For more details on the adaption procedure of the time step size and the order we refer to [143] and [4, Sect. 2], which we have followed both for the above description.

For the spatial marking in Line 11 and 18 we use a maximum strategy according to [23]. Thereby, we introduce two parameters θ_{c} and θ_{r} controlling the amount of cells that are locally coarsened or refined. Moreover, to stabilize the adaptive algorithm we implement an upper bound for the refinement level, the number of adaptive cycles (for space and time) and minimal/maximal time step sizes.

Note on the Gradient Recovery Estimator The gradient recovery estimator measures the spatial regularity and yields an indicator for local refinement, see for example [3,

Chap. 4]. For a general finite element function $v_h \in V_h$ (here c_h, μ_h), which approximates an unknown exact solution $v \in V$, we recover the gradient $G(v_h) \in (V_h)^d$ in the finite element space of v_h approximating the exact gradient ∇v. Thus, for the error in the H^1-seminorm we can substitute the gradient of the exact solution ∇v by the recovered gradient $G(u_h) \in (V_h)^d$ and obtain the approximation

$$|v - v_h|_{H^1}^2 = \int_\Omega |\nabla v - \nabla v_h|^2 \,\mathrm{d}x \approx \int_\Omega |G(v_h) - \nabla v_h|^2 \,\mathrm{d}x. \tag{4.56}$$

Accordingly we define the local estimates for each cell Q of the mesh $\mathcal{T}_n$ by

$$\eta_Q^2(v_h) \coloneqq \int_Q |G(v_h) - \nabla v_h|^2 \,\mathrm{d}x, \tag{4.57}$$

and for our model equations we define the global estimate

$$\mathrm{est}_x \coloneqq \left(\sum_{Q \in \mathcal{T}_n} \eta_Q^2 \right)^{1/2} \qquad \text{with} \qquad \eta_Q^2 \coloneqq \eta_Q^2(c_h) + \eta_Q^2(\mu_h). \tag{4.58}$$

In order to obtain an asymptotically exact a posteriori error estimator according to [3, Thm 4.4] the recovery operator has to satisfy certain conditions, see [3, Sect. 4.2], and a superconvergence property has to be fulfilled, see [3, Sect. 4.3]. The deciding difficulty is the dependence of the superconvergence property on restrictive assumptions on the mesh and the regularity of the solution [3, Sect. 4.4]. Since we use locally refined meshes with hanging nodes and higher order boundary mappings for curved boundaries we cannot expect to obtain an asymptotically exact a posteriori error estimator. Thus we circumvent the implementation of a sophisticated recovery process, like the averaging procedure proposed by Zienkiewicz and Zhu [177], and recover the gradient as component-wise L^2-projection instead. This means for each component $j \in \{1, \ldots, d\}$ we solve for $(G)_j \coloneqq \big(G(u_h)\big)_j \in V_h$ satisfying

$$\big(\varphi_i, (G)_j\big) = \big(\varphi_i, (\nabla u_h)_j\big), \tag{4.59}$$

for all basis functions $\varphi_i \in V_h$ for $i = 1, \ldots, N_x$. The resulting mass matrix is inverted by applying the CG method together with a Jacobi preconditioner. This approach seems computationally expensive since an additional mass matrix has to be inverted, but in the matrix-free framework this can be done efficiently.

Finally, for an analysis of averaging techniques, which yield error estimators, we also refer to Bartels and Carstensen [18, 39].

Implementation Details The described algorithm is implemented in `C++` based on the functionalities the library *deal.II* [5, 13]. Thereby, the efficiency of our implementation is based on the usage of the matrix-free framework [98], which allows direct parallelization with MPI, threads and CPUs vector units. We tried to implement the algorithm as general as possible to allow individual components to be easily replaced. For example, we can exchange the use of a different local refinement strategy, a different spatial estimator or even the model equations to be solved. However, for a different set of model equations we have to adapt the linear solver as well. In particular, the gradient recovery estimator, as we discussed above, is implemented in a general way like the `KellyErrorEstimator` of *deal.II*, so that it can be applied to a broad class of problems.

5. Nonlinear Matrix-Free Finite Element Operator Application

The main workload of our finite element solver from Chapter 4 remains in the solution of the linearized system (4.33) in each Newton step for each time step. For small problems with less than 10^4 DOFs we solve it with the LU-decomposition [52]. However, for an increasing problem size this becomes infeasible due to the large memory requirements, so we employ the GMRES method [123] with our proposed preconditioner (4.35). In particular, for the efficient solution we use the matrix-free framework of the finite element library *deal.II*—an innovative implementation of matrix-vector multiplications representing finite element operator applications.

The aim of this chapter is to first present the matrix-free framework within the *deal.II* library and and its key ideas in Section 5.1. Furthermore, in Section 5.2 we demonstrate the efficiency and strong parallel scalability of our implementation approach for a matrix-free Newton solver based on geometric multigrid preconditioned iterative solvers for the linearized systems. The presented nonlinear matrix-free operator layout defines the basis for the later application in our Cahn–Hilliard finite element solver.

5.1. The Matrix-Free Framework of the deal.II Library

In this subsection we motivate the usage of the matrix-free framework implemented in the *deal.II* library [5, 13] based on the work of Kronbichler and Kormann [96, 98]. Furthermore, we describe the key idea of cell-based finite element operator application reviewing the article [98] as well as the documentation of the *deal.II* library and the corresponding tutorial steps [5].

The whole section is understood as review and summary and therefore consists partially of literally and analogously adopted contents of the works [5, 96, 98]. For further details we refer the reader to these references and the references cited therein.

5.1.1. Motivation

Let us consider a plain Poisson problem owing to homogeneous Dirichlet boundary conditions on the d-dimensional unit cube $\Omega = (0,1)^d$: For a given smooth function $f\colon \Omega \to \mathbb{R}$, find a function $u\colon \Omega \to \mathbb{R}$ satisfying

$$\begin{cases} -\Delta u = f & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega. \end{cases} \tag{5.1}$$

By the theory of elliptic partial differential equations [30, 33] this problem has for every $f \in L^2(\Omega)$ a unique weak solution $u \in H_0^1(\Omega)$.

The basic steps to solve this problem with a standard Lagrangian finite element method [30, 33] are summarized in Algorithm 2.

Algorithm 2 Solving a boundary value problem with the finite element method.

1: Discretize the domain with a mesh $\mathcal{T}_h$
2: Distribute the DOFs and setup the finite element space
3: Assemble the system matrix A_h
4: Assemble the right hand side F_h
5: Solve the linear system $A_h\, u_h = F_h$
6: Postprocess the solution

For simplicity we assume a uniform quadrilateral mesh in 2D or hexahedral mesh in 3D $\mathcal{T}_h$ without hanging nodes and distribute the DOFs according to the p-th-order Lagrangian finite element space respecting the homogeneous Dirichlet boundary conditions

$$\mathcal{S}_0^p(\mathcal{T}_h) \coloneqq \left\{ v \in C(\overline{\Omega}) \,:\, v|_Q \in \mathbb{Q}_p(Q) \text{ for all } Q \in \mathcal{T}_h \right\} \cap H_0^1(\Omega). \tag{5.2}$$

With the nodal basis $\Pi = \{\varphi_i \,:\, i = 1, \ldots, N_x\}$ we obtain the discretized problem: For $f \in L^2(\Omega)$ find a function $u_h \in \mathcal{S}_0^p(\mathcal{T}_h)$ satisfying

$$(\nabla \varphi_i, \nabla u_h) = (\varphi_i, f), \tag{5.3}$$

for all $i = 1, \ldots, N_x$. According to the theory of finite elements [30, 33] this problem has a unique solution.

Using the basis representation of the solution $u_h = \sum_{j=1}^{N_x} \varphi_j U_j$, $\boldsymbol{U} \in \mathbb{R}^{N_x}$, we translate the discrete problem into a linear system to solve: Find $\boldsymbol{U} \in \mathbb{R}^{N_x}$ satisfying

$$\underbrace{\big((\nabla \varphi_i, \nabla \varphi_j)\big)_{ij}}_{A_h \coloneqq} \boldsymbol{U} = \underbrace{\big((\varphi_i, f)\big)_i}_{F_h \coloneqq}. \tag{5.4}$$

In general the most time consuming parts are the assembly of the system matrix A_h and moreover the solution of the linear system $A_h\, u_h = F_h$. In addition, memory requirements are also a limiting factor for the overall performance, even for sparse matrices that are stored for example in *compressed row storage* (CRS) format. The matrix-free framework in *deal.II* attacks these issues to reduce the total wall-clock time. Described in the introduction of tutorial `step-37` of the *deal.II* library, the main motivation for matrix-free implementations is hereby the observation that on modern CPUs the access to main memory for objects that do not fit into the cache has become the bottleneck of the solvers [5].

This limitation becomes particularly prominent for example during the application of a Krylov subspace method like CG to solve the linear system (5.4) [5, 98]. Note, that Krylov subspace methods are mainly based on matrix-vector multiplication. For large problems ($\gtrsim 10^4$) the recomputation of matrix entries may become faster than loading those entries from main memory [5, 98]. Thus the idea is to provide a function that evaluates the matrix-vector multiplication in an efficient way [5, 98].

Generic finite element software packages like *deal.II* [5, 13], *DUNE* [21, 22], *FEniCS* [104, 105] and others provide linear algebra implementations or interfaces to external libraries such as *PETSc* [10, 11] or *Trilinos* [78, 141] for the solution of the discrete

problems [98, p. 1]. However, in the context of finite element methods, matrix-vector multiplications of system matrices represent the action of the discrete differential operator on finite element functions [98]. For pure linear algebra implementations this information on the finite element origin of the matrices is lost [98].

The cell-based finite element operator application interface of Kronbichler and Kormann [96,98] reviews this issue and provides a highly efficient implementation for matrix-free computations within the finite element library *deal.II*. The matrix-free framework is memory-efficient and able to use three levels of parallelization: MPI over nodes, threads within nodes and CPUs vector units [98]. Moreover, for our nonlinear, time-dependent phase-field model (3.19) we benefit from the flexible approach, since the system matrices are updated several times in each time step and thus would need to be reassembled [98].

5.1.2. Cell-Based Finite Element Operator Application

In the following we briefly explain the key ideas for the efficient finite element operator application $A_h\, u_h$ of the matrix-free framework.

Like the cell-wise assembly of the system matrix in classical finite element codes, we first realize that the action of a global finite element operator can be split into cell-wise operator applications [5, 98]. Neglecting hanging nodes it is

$$A_h\, u_h = \sum_{k=1}^{n_{\text{cells}}} P_k^{\mathsf{T}} A_k P_k u_h, \tag{5.5}$$

where the rectangular matrix P_k defines the mapping of global to local DOFs and the square matrix A_k represents the local cell-related application of A_h [5, 98]. The operator application is thus evaluated by looping over all cells, extracting the local vector entries u_k, evaluating the cell operation $A_k\, u_k$ and finally distribute the local contributions into the global result [98, Alg. 2.1]. Hereby the main work is the evaluation of the cell operation $A_k\, u_k$. In order to avoid the a priori computation and storage of the cell matrices A_k, the matrix-vector product $A_k\, u_k$ is directly evaluated by quadrature [98].

Cell-Wise Quadrature Approach For a cell $Q_k \in \mathcal{T}_h$ we denote $F_k \colon \widehat{Q} \to Q_k$ as the mapping from the reference cell $\widehat{Q} = (0,1)^d$ to the real cell Q_k with its Jacobian $J_k = \nabla F_k$. We denote the cell-related basis functions of a p-th-order finite element method with the local DOFs $i = 1, \dots, (p+1)^d$ with $(\varphi_i)_i$ and the corresponding reference cell shape functions with $(\widehat{\varphi}_i)_i$. Furthermore, for the quadrature on the reference cell we have the quadrature weights w_q and the quadrature points $\widehat{x}_q$ mapped to the real cell points via $F_k(\widehat{x}_q) = x_q$ [5, 98].

With this notation the local operator application for our example (5.1) on cell $Q_k \in \mathcal{T}_h$

$$\left(A_k\, u_k\right)_i = \int_{Q_k} \nabla\varphi_i \cdot \nabla u_h \,\mathrm{d}x \tag{5.6}$$

for $i = 1, \dots, (p+1)^d$, is approximated by the transformation to the reference cell and

using a quadrature rule [98]

$$\big(A_k\, u_k\big)_i = \int_{Q_k} \nabla\varphi_i \cdot \nabla u_h \,\mathrm{d}x \tag{5.7}$$

$$= \int_{\widehat{Q}} \Big(\big((\nabla\varphi_i) \circ F_k\big) \cdot \big((\nabla u_h) \circ F_k\big) \Big) \big|\det J_k(\widehat{x})\big| \,\mathrm{d}\widehat{x} \tag{5.8}$$

$$\approx \sum_q \Big(\big(\nabla\widehat{\varphi}_i(\widehat{x}_q)\big)^{\mathsf{T}} J_k^{-1}(\widehat{x}_q) \nabla u_h(x_q) \Big) w_q \big|\det\big(J_k(\widehat{x}_q)\big)\big|, \tag{5.9}$$

for $i = 1, \ldots, (p+1)^d$. To save arithmetic operations the gradients $\nabla u_k(x_q)$ are evaluated in all quadrature points on the reference cell and afterwards multiplied with the inverse transposed Jacobian for the mapping to the real cell [98]

$$\nabla u_h(x_q) = \sum_{i=1}^{(p+1)^d} J_k^{-\mathsf{T}}(\widehat{x}_q) \nabla\widehat{\varphi}_i(\widehat{x}_q) u_k^{(i)} = J_k^{-\mathsf{T}}(\widehat{x}_q) \sum_{i=1}^{(p+1)^d} \nabla\widehat{\varphi}_i(\widehat{x}_q) u_k^{(i)}. \tag{5.10}$$

With this formula, the whole computation of $A_k\, u_k$ is performed in such a way that a significant amount of arithmetic operations is saved, see [98, Alg. 2.2]. The trick is the ordering of the multiplications and summations of quadrature data (weights, Jacobians) as well as the reuse of previous results (summations).

Sum-Factorization Another important technique is the sum-factorization known from spectral element methods [34, 98].

Let us consider the three-dimensional case $d = 3$. For tensor-product based finite element methods the basis functions can be expressed as tensor-product of one-dimensional basis functions, for example

$$\varphi_i(x_1, x_2, x_3) = \varphi_{i_1}(x_1)\varphi_{i_3}(x_2)\varphi_{i_3}(x_3), \tag{5.11}$$

with a multi-index $i \cong (i_1, i_2, i_3)$. The partial derivative in the second direction evaluated at a quadrature point x_q can be expressed as

$$\partial_2 u_h(x_q) = \sum_{i_3=1}^{p+1} \varphi_{i_3}(x_{q,3}) \sum_{i_2=1}^{p+1} \partial_2\varphi_{i_2}(x_{q,2}) \sum_{i_1=1}^{p+1} \varphi_{i_1}(x_{q,1}) u_h^{(i_3,i_2,i_1)}. \tag{5.12}$$

The partial derivative in the third direction evaluated at the same quadrature point x_q is analogously

$$\partial_3 u_h(x_q) = \sum_{i_3=1}^{p+1} \partial_3\varphi_{i_3}(x_{q,3}) \sum_{i_2=1}^{p+1} \varphi_{i_2}(x_{q,2}) \sum_{i_1=1}^{p+1} \varphi_{i_1}(x_{q,1}) u_h^{(i_3,i_2,i_1)}. \tag{5.13}$$

The key is to notice that the innermost sum $\sum_{i_1=1}^{p+1} \varphi_{i_1}(x_{q,1}) u_h^{(i_3,i_2,i_1)}$ is equal in both cases. Hence it can be computed once and then reused [98]. In particular, the same sum can also be reused for the evaluation of the function value [98]. Moreover, in the matrix-free framework the directional sums are evaluated at all quadrature points at once and then used for the summation over the next direction [98]. For a discussion of the computational complexity we refer the reader to [98, Sect. 2.4].

5.1.3. Literature Review

In the wide range of finite element software packages *deal.II* is not the only one providing matrix-free implementations. Alternative finite element libraries are for example NGSolve [126], Firedrake [93, 121] and others.

In particular, for phase-field models the `C++` software package PRISMS-PF [58] is based on the functionalities of the *deal.II* library using the matrix-free framework. However, only explicit time integration schemes are considered in PRISMS-PF [58]. As we observe the large computational savings due to the implicit adaptive time integration, this solution is not suited for our application problem.

Another Cahn–Hilliard finite element solver implemented with the matrix-free framework of *deal.II* from [99] is part of adaflo[1] [97]. Although the used block preconditioner from [8] is suitable for matrix-free computation, the implementation still relies on a matrix-based application of an AMG preconditioner provided through the interface to the Trilinos library [78, 141]. Furthermore, nonlinear mobility is not considered. Instead, the Cahn–Hilliard solver discussed in this thesis completely avoids the assembly and storage of matrices in the entire solution algorithm on every level.

Beside Cahn–Hilliard-type phase-field models the matrix-free framework of *deal.II* was recently also used for phase-field fracture simulations [87, 88]. Among others, the matrix-free framework was also used for problems in fluid dynamics [70,97], solid mechanics [53] and acoustic wave propagation [127].

The development of matrix-free finite element operator application framework with discontinuous finite element methods is a project within the priority program *SPEXXA* funded by the German Research Foundation (DFG). For an overview see the recently published article [7] and the references cited therein.

In addition, the matrix-free implementation is a promising technique for GPU computing [100, 103, 110], because recomputing matrix elements may be even more faster compared to CPUs. In particular, the *deal.II* tutorial `step-64` shows how to solve a Helmholtz equation with the matrix-free method on GPUs. However, GPU computing requires a different type of hardware and a different compiler.

5.2. A Nonlinear Problem Solved by the Matrix-Free Method

The implicit time integration of the Cahn–Hilliard equation requires for each time step a nonlinear problem to solve. As a first step, in this section we demonstrate the efficiency and strong parallel scalability of a matrix-free Newton solver. For the linearized Gelfand problem we implement a parallel matrix-free geometric multigrid solver and compare the wall-clock times for a matrix-based and a matrix-free implementation. The design of the nonlinear matrix-free operator is used in our Cahn–Hilliard solver as a basic building block.

We studied a previous version of this implementation for a slightly different problem setting already in [41]. The code has been revised and, initiated by Timo Heister, it is currently under review in pull request #8229 at the GitHub repository of the *deal.II* library[2].

[1] `https://github.com/kronbichler/adaflo` (accessed at 15.11.2020)
[2] `https://github.com/dealii/dealii/pull/8229` (accessed at 15.11.2020)

A final version will be part of the *deal.II* library as tutorial `step-66`. Note, the following description of the problem, the discretization and the implementation is based on the contributed documentation to the proposed tutorial.

5.2.1. Problem Description

On the d-dimensional unit ball $\Omega = \left\{\boldsymbol{x} \in \mathbb{R}^d : |\boldsymbol{x}| \leq 1\right\}$ we consider the following nonlinear elliptic boundary value problem subject to a homogeneous Dirichlet boundary condition: Find a function $u\colon \Omega \to \mathbb{R}$ satisfying

$$\begin{cases} -\Delta u = \exp(u) & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega. \end{cases} \tag{5.14}$$

This problem is known as the *Gelfand problem* and is a typical example for problems from combustion theory, see for example in the book of Bebernes and Eberly [26].

5.2.2. Discretization with Finite Elements

We discretize the domain with a mesh $\mathcal{T}_h$ using an isoparametric boundary mapping and distribute the DOFs according to the p-th-order Lagrangian finite element space $\mathcal{S}_0^p(\mathcal{T}_h)$ as for the plain Poisson problem (5.1) above. Multiplication with basis functions, integration by parts and the rearrangement of terms yields the discrete nonlinear problem: Find $u_h \in \mathcal{S}_0^p(\mathcal{T}_h)$ satisfying

$$F(u_h) \coloneqq \big(\nabla\varphi_i, \nabla u_h\big) - \big(\varphi_i, \exp(u_h)\big) = 0 \tag{5.15}$$

for all $i = 1, \ldots, N_x$, with the nonlinear residual $F\colon \mathbb{R}^{N_x} \to \mathbb{R}^{N_x}$. We solve this nonlinear problem with Newton's method. Given an initial guess $u_h^0 \in \mathcal{S}_0^p(\mathcal{T}_h)$ we compute a sequence of Newton steps $\big(u_h^n\big)_{n\in\mathbb{N}} \subset \mathcal{S}_0^p(\mathcal{T}_h)$ by successively applying the following scheme

$$\text{Solve for } s_h^n \in \mathcal{S}_0^p(\mathcal{T}_h): \qquad F'(u_h^n)[s_h^n] = -F(u_h^n), \tag{5.16}$$

$$\text{Update:} \qquad u_h^{n+1} = u_h^n + s_h^n. \tag{5.17}$$

We accept the current Newton step u_h^n as solution if either $|F(u_h^n)|_2 \leq 10^{-12}$ or $|s_h^n|_2 \leq 10^{-10}$ is satisfied. The implementation of the Newton method was partially inspired by the *deal.II* tutorial `step-15` [5] and the NGLBib [60].

5.2.3. Matrix-Free Numerical Linear Algebra

For each Newton step $n \in \mathbb{N}$ we solve the linear system $F'(u_h^n)[s_h^n] = -F(u_h^n)$, with the Jacobian $F'(u_h^n)[\,\cdot\,]\colon \mathbb{R}^{N_x} \to \mathbb{R}^{N_x}$ corresponding to the discrete finite element operator $A(u_h^n)$

$$A(u_h^n)\, s_h^n \coloneqq F'(u_h^n)[s_h^n] = \big(\nabla\varphi_i, \nabla s_h^n\big) - \big(\varphi_i, \exp(u_h^n) s_h^n\big), \tag{5.18}$$

and the negative residual $-F(u_h^n) \in \mathbb{R}^{N_x}$ as right hand side.

Note, how the operator A depends on the previous Newton step u_h^n. In a classical implementation with a priori computed system matrix we would gather the function values using the member function `get_function_values` of the `FEValues` class during assembly. In the matrix-free framework instead we implement a class `JacobianOperator` derived from the `MatrixFreeOperator::Base` class. Inspired by the implementation of the matrix-free Laplace operator with variable coefficient in *deal.II* tutorial `step-37` [5, 98], we implement in the sense of the `evaluate_coefficient` function a public member function `evaluate_newton_step`, which evaluates the provided vector of the old Newton step once before the application. In this way we store the function values in the `JacobianOperator` internally and access them during each application of the operator. So we need to update the function values only once before we solve for the next Newton step. Furthermore, we implement the evaluation of the residual with the `FEEvaluation` class and the `cell_loop` of the `MatrixFree` class.

For the solution of the linear system we employ a geometric multigrid (GMG) preconditioned CG method. In order to apply the operators on the different levels for the GMG preconditioner we pass the last Newton step to all level operators. The interpolation of the finite element function u_h^n onto all levels is done by the member function `interpolate_to_mg` of the `MGTransferMatrixFree` class. Note, that we cannot use typical smoothers like SOR or preconditioners like ILU, since in the matrix-free framework we have no access on matrix entries, except on the diagonal [5, 98]. Instead, we setup the GMG preconditioned according to *deal.II* tutorial `step-37` [5, 98]. We employ a fourth-order Chebyshev iteration based on the Jacobi preconditioner for smoothing the eigenvalue range $[1.2\lambda_{\text{max}}/15, 1.2\lambda_{\text{max}}]$. Hereby we compute an estimate for the largest eigenvalue λ_{max} by ten steps of the unpreconditioned CG algorithm. On the lowest level we employ the Chebyshev iteration as coarse grid solver with relative tolerance 10^{-3}. For more details on the Chebyshev iteration as smoother we refer to [1] and for details on the algorithm implemented in *deal.II* we refer to [148, Sect. 5.1] [5]. The computation is accelerated by performing the preconditioning steps only in single precision and the CG method reduces the residual for the Newton updates below 10^{-12} [5, 98].

A quite similar implementation for the evaluation of the old Newton step was already used in the adaflo solver [97]. There the function values of the old Newton step are evaluated and stored internally while computing the residual. These stored values can then be accessed during the application of the nonlinear matrix-free operator. Hence a separated evaluation function is not necessary. However, the strategy in adaflo is not sufficient in our case, because we use a multigrid solver and for its level operators we have to evaluate the old Newton step on all levels.

5.2.4. Numerical Results

We show the efficiency of the matrix-free framework in the nonlinear case by comparing it first to a matrix-based implementation with the LU-decomposition and second measure the strong parallel scalability. The experiments were performed on the *Skylake* partition of the parallel computer *Horst*. For the software and hardware specifications we refer to the introduction of the following Chapter 6.

Comparison of the Matrix-Based and Matrix-Free Method To show the efficiency over matrix-based methods we compare the wall-clock times for the solution of the three-dimensional problem (5.14) using Newton's method. The measured times for different finite element degrees and different refined meshes are listed in Table 5.1. Note, that these times include the setup of the GMG preconditioner and for the matrix-based implementation also the assembly the system matrices. Both cases are executed with shared memory parallelization through eight CPUs. For the matrix-free computation we use the partition/partition scheme [96] and take advantage of using the $256\,\mathrm{bit}$ AVX vector units. The matrix-based implementation assembles the system matrices and the residuals with shared memory parallelization through threads and solves the linear systems with the LU-decomposition provided by UMFPACK [52].

Table 5.1.: Wall-clock times (in seconds) of the Newton solver for the three-dimensional Gelfand problem. Comparison of matrix-free (MF) and matrix-based (MB) implementation with different finite element degrees.

	2 levels (448 cells)		3 levels (3584 cells)	
FE degree	MF time	MB time	MF time	MB time
1	0.025	0.095	0.093	0.46
2	0.065	0.46	0.30	8.38
3	0.17	3.07	0.79	87.01
4	0.40	12.6	1.73	470.9

Over all finite element degrees and both levels the matrix-free implementation outperforms the matrix-based LU-decomposition. We particularly benefit from the direct application of the nonlinear finite element operator without the necessity of reassembling the system matrix in each Newton step. As we increase the problem size this advantage pays off more significantly. The evaluation of the residuals is also significantly faster due to the efficient implementation within the matrix-free framework. For a comparison of the matrix-free and the matrix-based matrix-vector multiplication for the Laplace operator we refer to [98]. Note also, that higher order boundary mappings influence the performance of the matrix-free implementation [98, Sect. 5.1.1].

Parallel Scalability We solve the Gelfand problem for two different spatial discretization with the third-order isoparametric Lagrangian finite element method in two and three spatial dimensions. Enabling distributed memory parallelization via MPI and using the $256\,\mathrm{bit}$ AVX vector units we perform a strong parallel scalability analysis [106, Def. 2.4] of the computational wall-clock time for the Newton–Raphson iteration without the initial setup. Thereby we measure indirectly the scalability of the evaluation of the nonlinearity in each Newton step, since the setup of the GMG preconditioner is part of the `compute_update` function. In Figure 5.1 we plot the wall-clock times in seconds over the number of parallel used CPUs for different problem sizes.

In both spatial dimensions we observe nearly optimal scalability if the number of cells is sufficiently large. For the cases with less cells the offset of the reference line for ideal scaling is more pronounced and indicate the beginning of scaling saturation by approximately

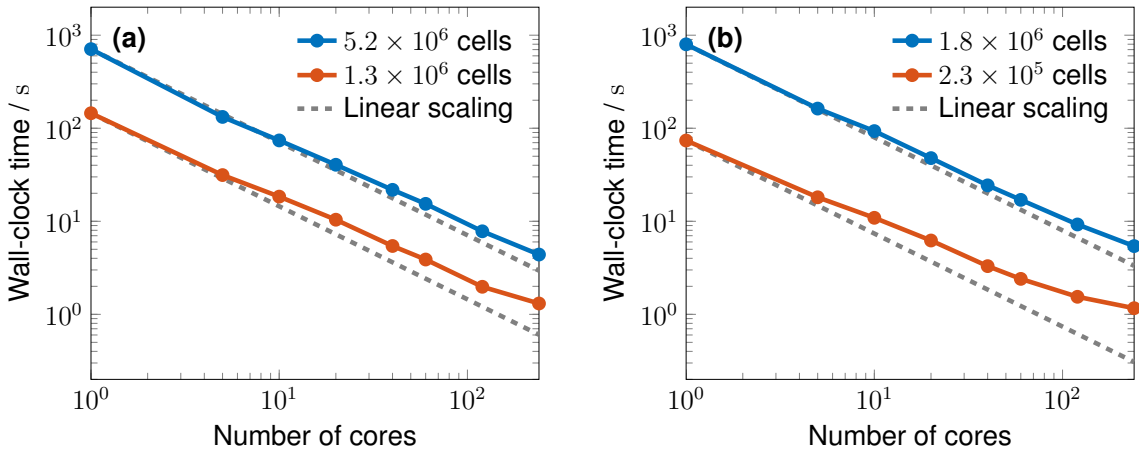

Figure 5.1.: Strong parallel scalability analysis of the computational wall-clock time (in seconds) for the solution of the (a) two- and (b) three-dimensional Gelfand problem with the matrix-free Newton iteration.

one second. This reduction in the scalability may have its reason in the communication and evaluation of the old Newton step for Jacobian operator.

5.2.5. Conclusion

We have analyzed the layout for an implementation of a nonlinear matrix-free finite element operator suitable for the application of geometric multigrid based linear solvers. This layout is especially intended for Jacobian operators that are used in the Newton–Raphson method for solving nonlinear partial differential equations with finite element methods.

Derived from the `MatrixFreeOperator::Base` class, we therefore implemented the `JacobianOperator` class with an additional public member function for the evaluation and internally storage of the previous Newton step. In this way the nonlinearities are evaluated only once before the linearized system is solved.

The performance of the linear solver for the Newton updates clearly depends on the underlying problem. For symmetric and positive definite Jacobians, as for the considered Gelfand problem, a GMG preconditioned CG method shows superior numerical efficiency compared to the naive attempt with a matrix-based LU-decomposition. In particular we demonstrated the nearly optimal strong parallel scalability including the communication and evaluation of the old Newton step and the GMG preconditioner setup in each Newton step for sufficiently large problem sizes.

As proposed in the extension of the *deal.II* tutorial `step-48`, further speedup can be achieved by the vectorized evaluation of the nonlinearities [5]. For a comparison of the cell-based matrix-free finite element operator application against the SpMV case we refer to Kronbichler and Kormann [98] as well as the *deal.II* tutorial `step-37` [5].

Together with the space and time discretization as well as the extended preconditioner we have all ingredients for our completely matrix-free finite element solver for the Cahn–Hilliard equation. A special key is the preconditioner (4.35) from Subsection 4.2.3, which is applicable for matrix-free computations. The evaluation of the old Newton step is necessary at two major points. First in the global Jacobian of the mixed finite element formulation and second in the nonlinear preconditioner block with the concentration dependent

mobility (3.12). We demonstrate the efficiency of our matrix-free Cahn–Hilliard solver in the next chapter.

6. Numerical Experiments

In this chapter we present and discuss the results of the numerical experiments performed with our matrix-free adaptive finite element solver from Chapter 4. Preliminarily, we define the simulation setup and the default parameters of the adaptive solution algorithm. The presentation of the results is then divided into two parts. First, in Section 6.1 we validate the developed solver with the method of manufactured solutions and study its performance based on the typical Cahn–Hilliard example of spinodal decomposition. Then in Section 6.2 we switch to the application case of phase separation in electrode particles. Based on the Cahn–Hilliard phase-field model from Chapter 3 we simulate lithium insertion in phase transforming LFP electrode particles. We validate the solver numerically and analyze its performance for the application problem. Furthermore, we show the evolution of the phase transition morphology under different symmetry assumptions and, in particular, for anisotropic material properties.

Note, that the experiments and discussion for the spherical symmetric case, presented in Subsection 6.2.1, are based on our article [43]. However, here we use an improved version of the solver. Additionally, the presentation of the numerical experiments contains formulations from our article [44].

Software and Hardware We implement the presented numerical solution algorithm from Chapter 4 as C++ code based on the functionalities of the finite element library *deal.II* in version 9.0 [5, 13] using the matrix-free framework [96, 98]. The mesh is distributed among MPI ranks through the interface to *p4est* [12, 36].

All simulations are performed on one of the following three hardware architectures. Small problems are solved on a *laptop* pc with $4\,\mathrm{GB}$ RAM and an Intel i5-4200U CPU with $1.6\,\mathrm{GHz}$. Computationally intensive simulations are submitted to the parallel computer *Horst* with two different partitions. The *IvyBridge* partition has 16 nodes with $64\,\mathrm{GB}$ RAM and 16 Intel Xeon E5-2650 v2 CPUs with $2.6\,\mathrm{GHz}$. The more modern *Skylake* partition has 12 nodes with $96\,\mathrm{GB}$ RAM and 20 Intel Xeon Silver 4114 CPUs with $2.2\,\mathrm{GHz}$. Matrix-free computations are performed on the *laptop* with the $128\,\mathrm{bit}$ SSE2 register, whereas on the parallel computer *Horst* the $256\,\mathrm{bit}$ AVX register is supported. When we solve a problem in parallel using MPI we disable shared memory parallelization.

During the description of the numerical experiments we thus only note the architecture and the number of MPI ranks used for solving a specific problem.

The graphical postprocessing here and in the later Section 7.4 was done with *ParaView* [2] and PGFPLOTS.

Adaptive Solution Algorithm If not stated otherwise we employ for all numerical experiments fourth-order isoparametric Lagrangian finite elements ($p = 4$) and use the variable-step, variable-order time integration scheme with the tolerances $\mathrm{RelTol}_t = 10^{-5}$ and $\mathrm{AbsTol}_t = 10^{-8}$. When we allow spatial adaptivity, we refine and coarsen the mesh

locally with the maximum strategy according to [23]. Hereby we use the parameters for refinement $\theta_\mathrm{r} = 0.5$ and coarsening $\theta_\mathrm{c} = 0.05$ as well as the tolerances $\mathrm{RelTol}_x = 10^{-5}$ and $\mathrm{AbsTol}_x = 10^{-8}$.

However, in general we only employ spatial adaptivity for one-dimensional electrode particle simulations under the assumption of spherical symmetry. In the higher-dimensional case further improvements of the spatial estimation process have to be done. A further issue is the bounded scalability of the GMG method on locally refined meshes in the matrix-free framework, see the remark in *deal.II* tutorial `step-37` [5].

6.1. Classical Cahn–Hilliard Equation

In this section we validate our finite element solver, presented in Chapter 4, and study its performance based on classical Cahn–Hilliard problem setups for spinodal decomposition from literature. First, we numerically show the convergence and the efficiency of our finite element solver with the method of manufactured solutions. Based on three examples of spinodal decomposition we then analyze several numerical aspects, like the space and time adaptivity and, in particular, the preconditioning.

Problem Formulation For the rest of this section we consider the classical Cahn–Hilliard initial boundary value problem, which is a simplified version of Problem 1.

Problem 3. Let $t_\mathrm{f} > 0$ be the final simulation time and $\Omega \subset \mathbb{R}^d$, $d \in \{1, 2, 3\}$, an open, bounded domain. We seek two functions $c\colon [0, t_\mathrm{f}] \times \overline{\Omega} \to \mathbb{R}$ and $\mu\colon [0, t_\mathrm{f}] \times \overline{\Omega} \to \mathbb{R}$ satisfying

$$\begin{cases} \partial_t c = \nabla \cdot \big(m(c) \nabla \mu\big) & \text{in } (0, t_\mathrm{f}) \times \Omega, \\ \mu = f'(c) - \kappa \Delta c & \text{in } (0, t_\mathrm{f}) \times \Omega, \\ \nabla c \cdot \boldsymbol{n} = 0 & \text{on } (0, t_\mathrm{f}) \times \partial\Omega, \\ m(c) \nabla \mu \cdot \boldsymbol{n} = 0 & \text{on } (0, t_\mathrm{f}) \times \partial\Omega, \\ c(0, \,\cdot\,) = c_0 & \text{in } \Omega, \end{cases} \tag{6.1}$$

with the scalar mobility $m\colon \mathbb{R} \to \mathbb{R}$, the homogeneous free energy density $f\colon \mathbb{R} \to \mathbb{R}$, the interfacial energy coefficient $\kappa > 0$ and the given initial condition $c_0\colon \overline{\Omega} \to \mathbb{R}$, which will be specified below in the respective subsections.

6.1.1. Validation with the Method of Manufactured Solutions

We validate our matrix-free finite element solver with the method of manufactured solutions [122] and show convergence with optimal order. Therefore we follow the quantitative comparison of Zhang et al. [164] for C^0- and C^1-continuous finite elements, which was similarly done by Kästner et al. [90] for an isogeometric analysis (IGA) and by Kaessmair and Steinmann [89] additionally for a natural element analysis (NEA). Moreover, we compare the efficiency of our solver for increasing finite element degree, as in [164].

According to the method of manufactured solutions we allow an additional residual term $F\colon [0, 1] \times [0, 1]^d \to \mathbb{R}$ to the diffusion equation of (6.1) in the formulation of Problem 3.

For $d \in \{2, 3\}$, find $c\colon [0,1] \times [0,1]^d \to \mathbb{R}$ and $\mu\colon [0,1] \times [0,1]^d \to \mathbb{R}$ satisfying

$$\begin{cases} \partial_t c = \Delta \mu + F & \text{in } (0,1) \times (0,1)^d, \\ \mu = f'(c) - \Delta c & \text{in } (0,1) \times (0,1)^d, \end{cases} \tag{6.2}$$

where we assume the quartic free energy density $f(z) = 1/4\, z^2(z^2 - 2)$. The mobility and the interfacial energy coefficient are set to one. In order to directly satisfy the homogeneous Neumann boundary conditions, $\nabla c \cdot \boldsymbol{n} = \nabla \mu \cdot \boldsymbol{n} = 0$, we adapt the proposed manufactured solution from [164] and use with $\alpha = 2$ the solution

$$c(t, x) = (t + 1) \cos(\alpha \pi x_1), \tag{6.3}$$

which serves, evaluated at $t = 0$, also as initial condition. We insert this manufactured solution in the Cahn–Hilliard equation (6.2) and compute the residual

$$\begin{aligned} F(t, x) &= \partial_t c - \Delta f'(c) + \Delta^2 c \\ &= \cos(\alpha \pi x_1) \\ &\quad - 6(\alpha\pi)^2 (t+1)^3 \sin^2(\alpha \pi x_1) \cos(\alpha \pi x_1) + 3(\alpha\pi)^2 (t+1)^3 \cos^3(\alpha \pi x_1) \\ &\quad - (\alpha\pi)^2 (t+1) \cos(\alpha \pi x_1) + (\alpha\pi)^4 (t+1) \cos(\alpha \pi x_1). \end{aligned} \tag{6.4}$$

Note, that this manufactured solution only depends on the x_1-direction and validates the code whether a one-dimensional solution profile is recovered in higher dimensions.

For the comparison with [164] we solve one time step ($\tau = 1$) of the Cahn–Hilliard equation with our matrix-free finite element solver of increasing finite element degree in parallel with 20 MPI ranks on the *Skylake* partition of *Horst*, compute the errors

$$\mathrm{err}_\diamond := \left(\| c(1, \cdot) - c_h(1, \cdot) \|_\diamond^2 + \| \mu(1, \cdot) - \mu_h(1, \cdot) \|_\diamond^2 \right)^{1/2}, \tag{6.5}$$

for $\diamond \in \{L^2, H^1\}$, and measure the computational time of the Newton–Raphson method (excluding the setup time) for the efficiency analysis.

In Figure 6.1 we plot the errors in the two- and three-dimensional case for different finite element degrees over the number of DOFs. For both spatial dimensions and both errors we observe the optimal order of convergence according to the theory of the Lagrangian finite element method [30, 33].

The comparison of the computational wall-clock times for solving one time step (without the setup time) depends on whether we compare it to the number of DOFs or the number of cells. In Figure 6.2 we plot the computational wall-clock times over the number of DOFs. While the higher order methods ($p \geq 2$) basically show the same performance, we observe a larger computational time for linear finite elements. This observation reflects the performance of the matrix-free framework for different finite element degrees. In contrast, comparing the wall-clock time against the number of cells per dimension as in [164], higher order methods are computationally more expensive. However, knowing that higher order methods distribute more DOFs per cell, also the number of arithmetic operations increases and thus more computational time per cell is necessary. Furthermore, the steeper increase of the wall-clock time in the two-dimensional case between 10^4 and 10^5 DOFs may reflect the imbalance of the parallel distribution of the cells, since we solve all cases with 20 MPI ranks. In the three-dimensional case already a small refinement level generates enough cells to keep all MPI ranks busy.

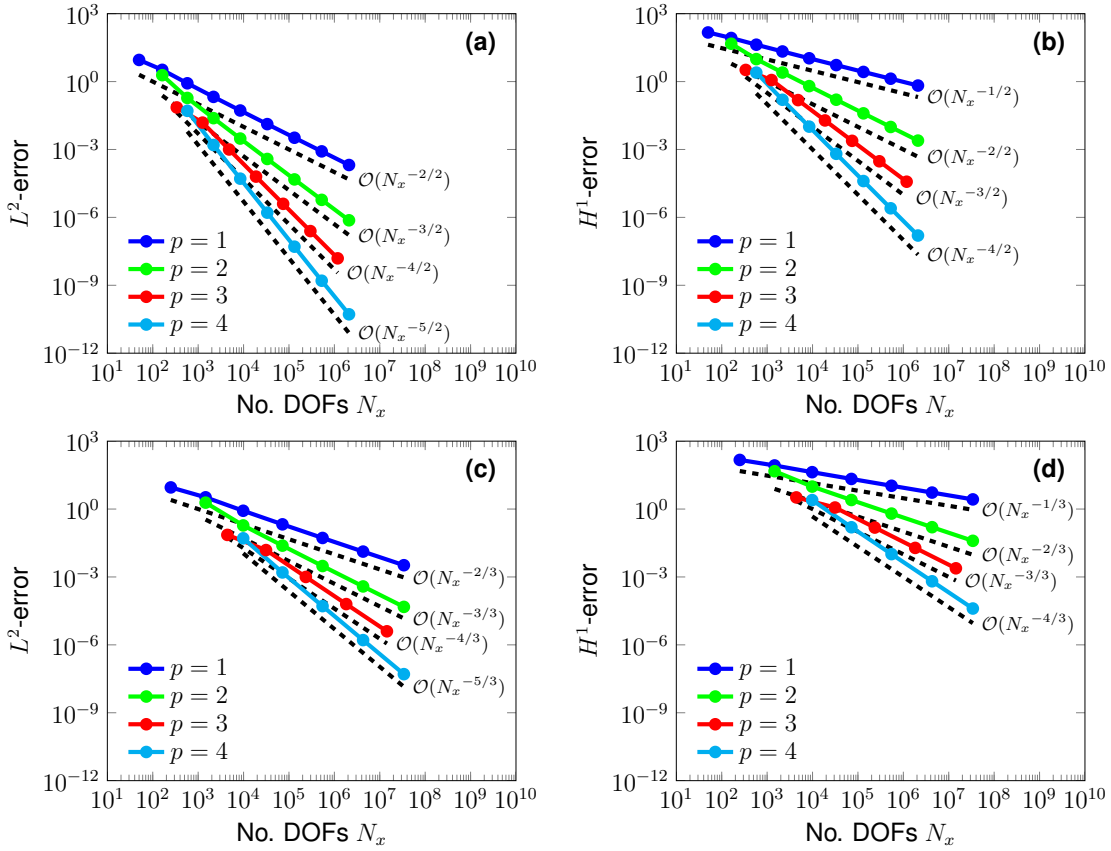

Figure 6.1.: Convergence analysis for the (a,b) two- and (c,d) three-dimensional problem. In both cases the (a,c) L^2- and (b,d) H^1-errors are plotted against the number of DOFs. Dashed reference lines indicate the optimal orders.

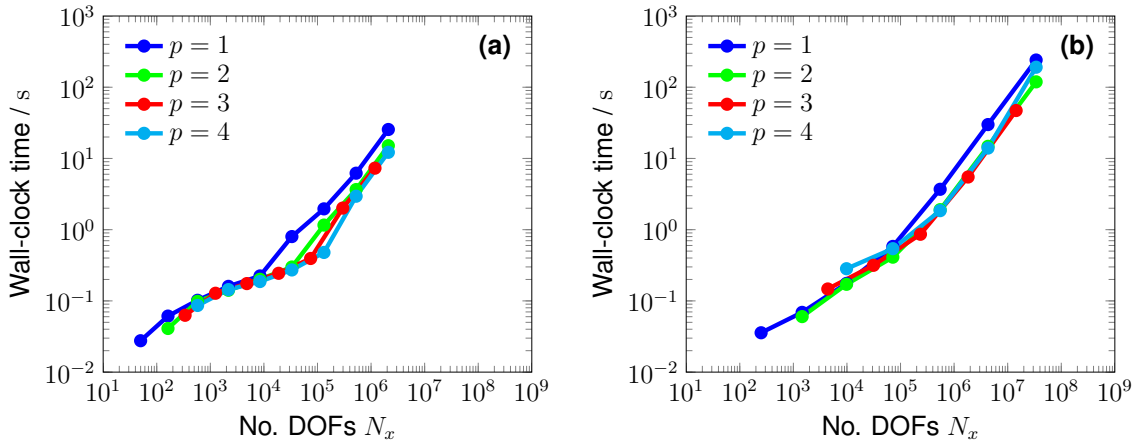

Figure 6.2.: Computational wall-clock time for the solution of one time step for the (a) two- and (b) three-dimensional manufactured solution for different finite element degrees.

We compare the numerical efficiency of the finite element methods by plotting the L^2-errors against the computational wall-clock times in Figure 6.3. Independent of the spatial dimension, for a higher finite element degree the computational time to achieve a certain

error is clearly reduced and thus higher order methods are much more efficient. We observe the same for the H^1-errors not plotted here. This efficiency is mainly due to the high-performance implementation of the matrix-free framework, which explains our choice of higher order finite element method for the upcoming simulations.

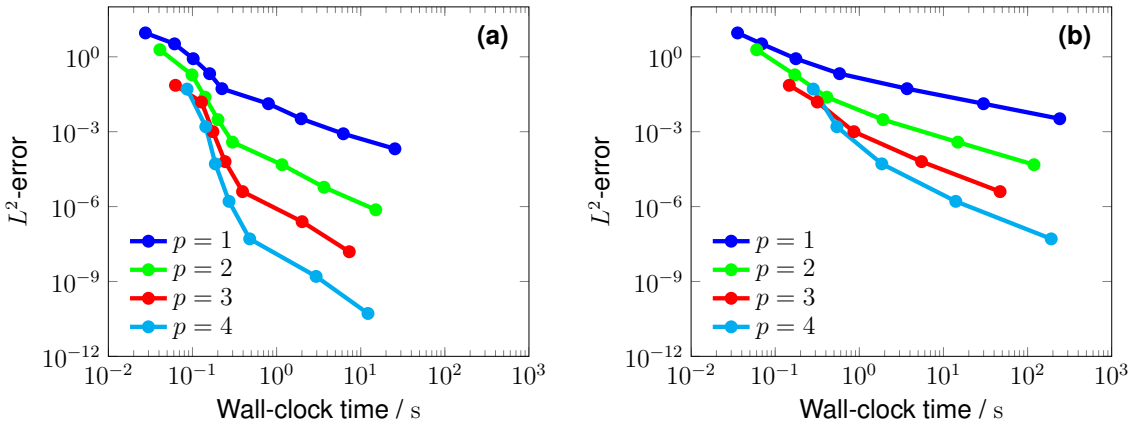

Figure 6.3.: Numerical efficiency of L^2-error over the computational wall-clock time for the (a) two- and (b) three-dimensional case.

Comparing this convergence and efficiency analysis with examples from literature [89, 164] the parallel matrix-free finite element solver outperforms existing implementations of C^0- and C^1-continuous methods. However, the performance always depends on the used software and hardware. In particular, only a clearly defined benchmark problem, where all parameters are available, can serve for a concrete efficiency analysis and comparison.

6.1.2. Study on Solver Components based on Spinodal Decomposition

Next we analyze the functionality of our matrix-free finite element solver based on the example of spinodal decomposition. Therefore we consider in the following three different settings of the classical Cahn–Hilliard equation.

1. Random initial data, constant mobility and quartic homogeneous free energy,
2. Random initial data, degenerated mobility and logarithmic homogeneous free energy,
3. Smooth initial data, constant mobility and quartic homogeneous free energy.

Based on these cases we validate the numerical solution, demonstrate the efficiency of the adaptive matrix-free solver and show the discretization-independent preconditioning with our proposed preconditioner (4.35).

Spinodal decomposition is the most commonly studied application of the classical Cahn–Hilliard equation, see for example [17, 20, 73, 74, 153, 154]. For this application the Cahn–Hilliard equation is considered with an initial condition of the form

$$c_0 = \bar{c}_0 + X, \tag{6.6}$$

where $X\colon \overline{\Omega} \to [-\varepsilon, \varepsilon]$ introduces a small perturbation $\varepsilon > 0$ to the average concentration $\overline{c}_0 \in \mathbb{R}$. Thus when the average concentration $\overline{c}_0$ is in the spinodal region, i. e., in the range between the two inflection points ($f''(\overline{c}_\pm) = 0$) of the homogeneous free energy density f, the process of phase separation is initiated, which is followed by the coarsening process. Note, that the range of the concentration depends on the homogeneous free energy density f. While commonly used quartic double wells with local minima in 0 and 1 or ± 1 allow concentration values in $\mathbb{R}$, logarithmic free energies, as in our case (3.2), allow concentration values only in the range $[0, 1]$.

In the following we realize a random perturbation of the average concentration by a white noise per DOF. More precisely, we define the spatially discrete random initial condition DOF-wise by

$$c_{0,h}(\boldsymbol{x}_i) = \overline{c}_0 + X_i, \tag{6.7}$$

for each DOF $\boldsymbol{x}_i$, $i \in \{1, \dots, N_x\}$, where $X_i \sim \mathcal{U}(-\varepsilon, \varepsilon)$ are uniformly distributed random variables with values in the range $[-\varepsilon, \varepsilon]$. In Figure 6.4 we depict an example of such a random initial condition with $\overline{c}_0 = 0.3$ and $\varepsilon = 0.01$, where the spatial domain is discretized with 4225 DOFs, which are distributed according to the fourth-order Lagrangian finite element method on 256 cells among four MPI ranks.

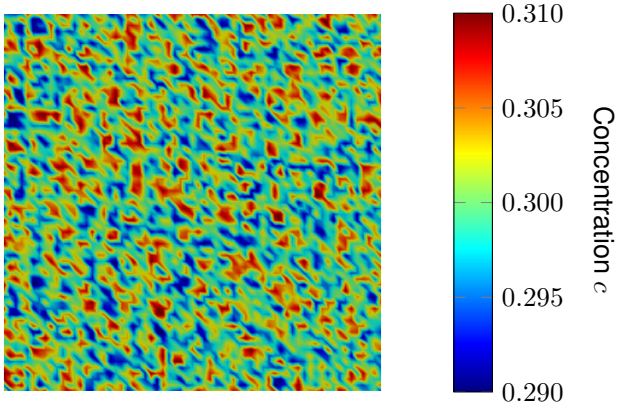

Figure 6.4.: Example of a nodal random initial condition with the average concentration $\overline{c}_0 = 0.3$ and the amplitude $\varepsilon = 0.01$.

Example 1: Random Initial Data

In the first part we assume a random perturbation of the average concentration $\overline{c}_0 = 0.3$ with the amplitude $\varepsilon = 0.01$ and solve the classical Cahn–Hilliard equation (6.1) with constant mobility $m = 1$, the interfacial energy coefficient $\kappa = 0.01$ and a quartic homogeneous free energy density $f(z) = 25(z^2 - 1)^2$ on the unit square $\Omega = (0, 1)^2$ until the final time $t_\mathrm{f} = 1$, as studied in [17].

Especially for this example an analytical expression approximating the one-dimensional phase transition profile in the equilibrium state is given by

$$\widetilde{c}(x) = \tanh\bigl(100\, x/\sqrt{2}\bigr), \tag{6.8}$$

which satisfies the boundary condition $\widetilde{c}(x) \to \pm 1$ for $x \to \pm\infty$ and is determined as the minimizer of the free energy functional Ψ by solving

$$\mu(c) = f'(c) - \kappa \Delta c \stackrel{!}{=} 0, \tag{6.9}$$

compare [17, 154]. With this analytical expression we validate the equilibrium profile of the numerical solution. Furthermore, we demonstrate the large computational savings due to the variable-step, variable-order time integration. Moreover, we reveal the size of the Arnoldi basis to be a critical parameter controlling the maximum number of iterations for the preconditioned GMRES method. For the better visualization of the time step data (energy, time step size, number of GMRES steps) over time we cut off the initial stage of smoothing the random initial data by starting the time axis at 10^{-6}.

Numerical Validation We discretize the domain with $n_{\text{cells}} = 2^7 \times 2^7 = 16\,384$ cells, distribute fourth-order finite elements ($N_x = 526\,338$) and perform the time integration with the described variable-step, variable-order algorithm for the numerical solution. Distributed over 16 MPI ranks on the *IvyBridge* partition, our matrix-free finite element solver without spatial adaptivity takes less than three hours. For the reproducibility of the simulation we initialize the random number generator with the default seed plus the number of the MPI rank.

In Figure 6.5a–i we depict a series of snapshots of the concentration distribution, where we see the evolution of the typical spinodal decomposition process, compare [17, 154]. Starting the simulation with the initial time step size $\tau = 10^{-12}$ the adaptive time stepping uses 4436 steps to reach the final time. After 422 steps, the spontaneous phase separation process has formed small droplets (Figure 6.5a), which are merged and absorbed during the following coarsening process (Figure 6.5a–h) until the equilibrium state is reached (Figure 6.5i).

We validate the numerical approximation of the equilibrium solution (Figure 6.5i) with the known approximation of the equilibrium profile (6.8). In Figure 6.6 we plot the concentration profile over the indicated diagonal line and observe the accordance to the analytical expression. Note, that in order to compare the equilibrium profile to the analytical expression we have to introduce a shift by the radius of the equilibrium droplet.

Additionally in Figure 6.7 we plot the free energy

$$\Psi(t) \coloneqq \int_\Omega \psi\big(c(t,\boldsymbol{x}), \nabla c(t,\boldsymbol{x})\big)\,\mathrm{d}\boldsymbol{x} = \underbrace{\int_\Omega \psi_{\text{ch}}\big(c(t,\boldsymbol{x})\big)\,\mathrm{d}\boldsymbol{x}}_{=:\Psi_{\text{ch}}(t)} + \underbrace{\int_\Omega \psi_{\text{int}}\big(\nabla c(t,\boldsymbol{x})\big)\,\mathrm{d}\boldsymbol{x}}_{=:\Psi_{\text{int}}(t)}, \tag{6.10}$$

over time and observe that our solver satisfies the monotonous energy decay property of the Cahn–Hilliard equation, compare (3.23), where we obtain with $N_{\text{ext}} = 0$ and $m(c) \equiv 1$

$$\frac{\mathrm{d}}{\mathrm{d}t}\Psi(t) = \int_\Omega \partial_z \psi\big(c, \nabla c\big)\partial_t c + \partial_{\boldsymbol{p}} \psi\big(c, \nabla c\big)\nabla \partial_t c\,\mathrm{d}\boldsymbol{x} = -\int_\Omega |\nabla \mu|^2\,\mathrm{d}\boldsymbol{x} \leq 0. \tag{6.11}$$

We also note the correlation of the drops in the energy plot (Figure 6.7) to the adaptive change of the time step size, plotted in Figure 6.8.

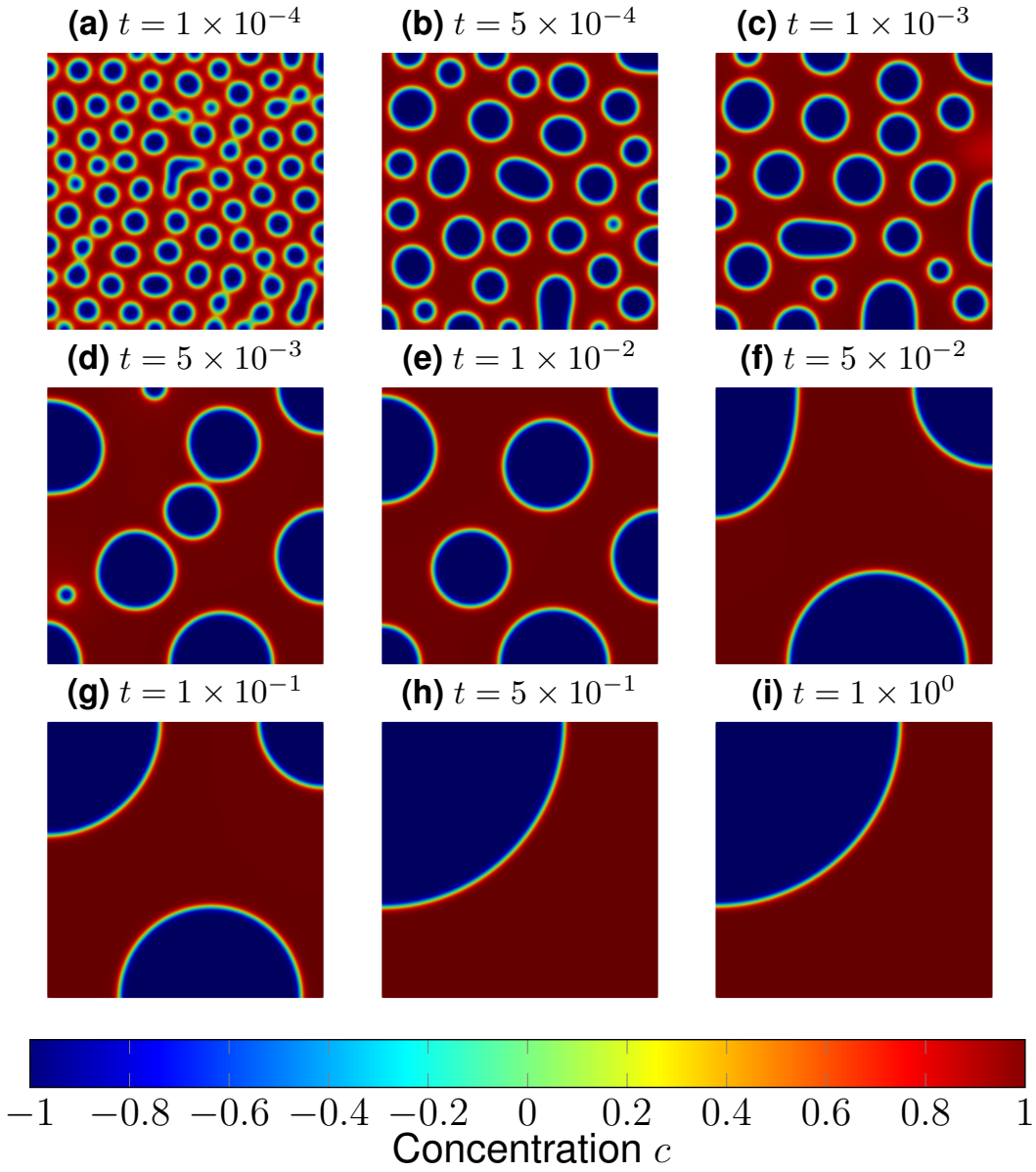

Figure 6.5.: Numerical solution of the Cahn–Hilliard equation with constant mobility and quartic free energy density. (a–i) Snapshots of the spinodal decomposition process at different time instances.

Moreover, our solver conserves the total mass $\int_\Omega c(t, \boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$, which is another key property of the Cahn–Hilliard equation, compare (3.22) with $N_{\text{ext}} = 0$

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_\Omega c\,\mathrm{d}\boldsymbol{x} = -\int_\Omega \nabla\cdot\boldsymbol{N}\,\mathrm{d}\boldsymbol{x} = -\int_{\partial\Omega} \boldsymbol{N}\cdot\boldsymbol{n}\,\mathrm{d}\sigma(\boldsymbol{x}) = 0. \tag{6.12}$$

Variable-Step, Variable-Order Time Integration Next we investigate the efficiency of the variable-step, variable-order time integration scheme.

In Figure 6.8 we plot the time step size and the order of the used multistep method for each step over time to the simulation visualized in Figure 6.5. After the fast separation process, before $t = 10^{-6}$, the time step size stabilizes but then starts to vary strongly in

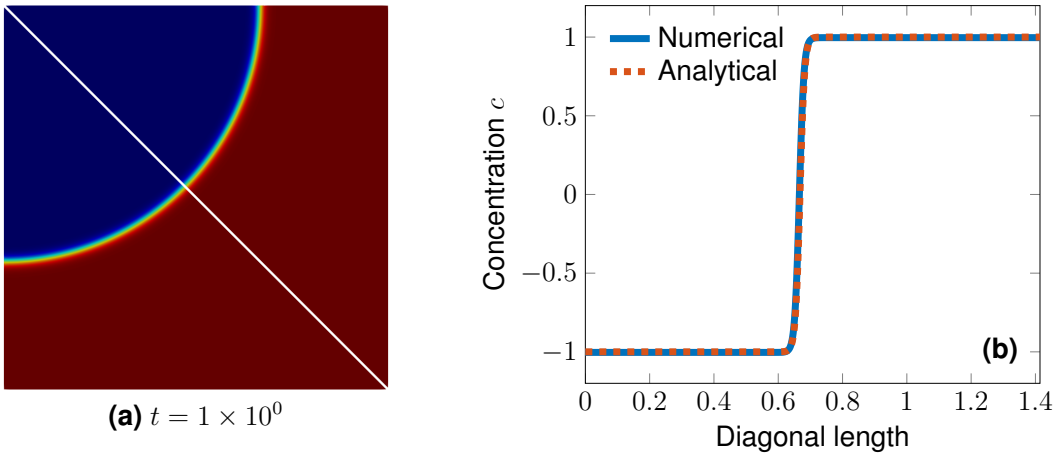

Figure 6.6.: Numerical validation of the finite element approximation. (a) Equilibrium concentration distribution with white diagonal line for the comparison in (b) of the concentration profile plotted over the diagonal line to the analytical expression (6.8).

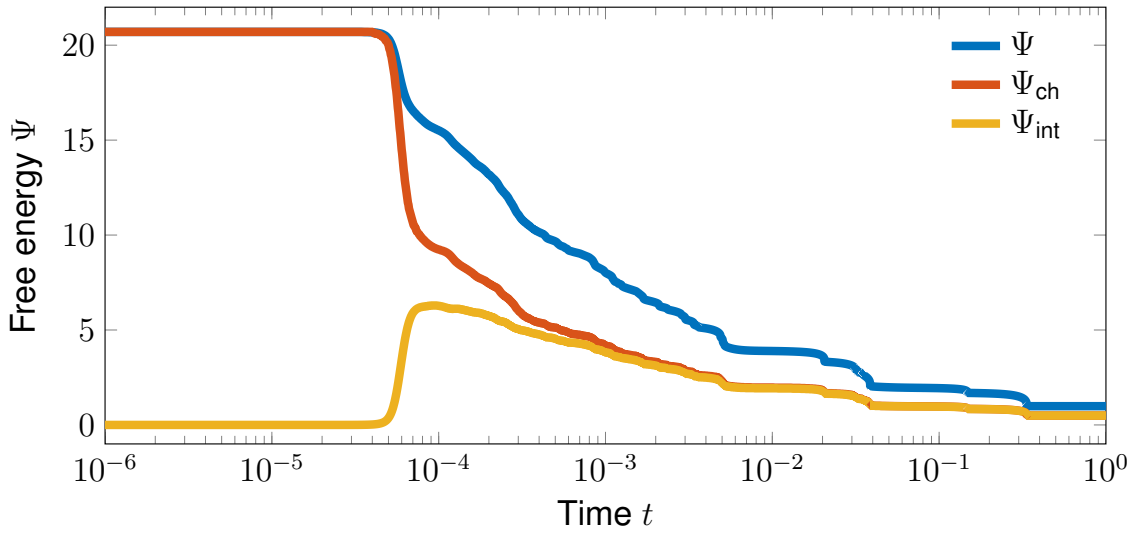

Figure 6.7.: Monotonous decrease of the free energy Ψ over time. Additionally the chemical Ψ_{ch} and the interfacial Ψ_{int} contributions are plotted.

the time interval $[10^{-4}, 5 \times 10^{-3}]$. This stage is correlated to the fast merging and absorption during the coarsening process, compare Figure 6.5a–d, and each jump of the time step size corresponds to a topological change in the solution profile. As the time further evolves, the phase separation dynamic relaxes, however, the time step size jumps over an increasing range of magnitudes to capture the individual phase separation events, compare Figure 6.5e–i.

Obviously the adaptive change of the time step size dramatically reduces the number of time steps compared to a constant-step method. A small calculation illustrates the computational saving through the adaptive time stepping: The smallest time scale of the coarsening process is approximately 4×10^{-7}. If we would solve the Cahn–Hilliard equation in the

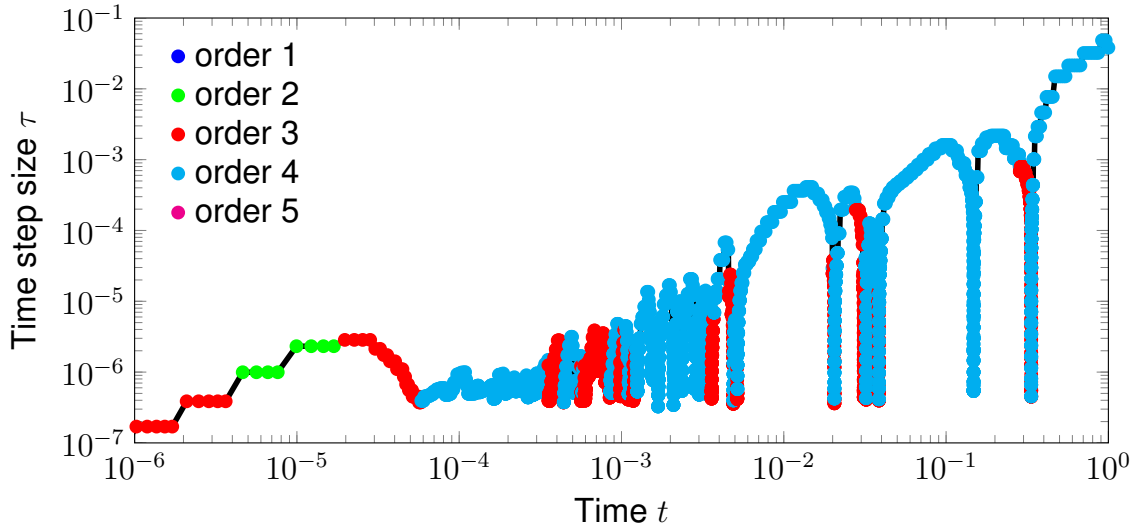

Figure 6.8.: Adaptive time step size and used order of the variable-step, variable-order time integration scheme over time.

time interval $[0, 1]$ with this constant time step, we would need to solve 2.5×10^6 time steps. Instead, with the variable-step, variable-order method, we used 4436 steps only. Thus we reduced the number of time steps at least by the factor 500.

Table 6.1.: Comparison of variable-step, variable-order BDF and NDF time integration statistics (Number of time steps N_t, maximum time step size $\tau_{\max}$) for different maximum orders $k_{\max}$. Wall-clock time in hours for the whole solution process (without graphical output).

	BDF(k)			NDF(k)		
$k_{\max}$	N_t	$\tau_{\max}$	Wall time / h	N_t	$\tau_{\max}$	Wall time / h
5	4580	4.0×10^{-2}	2.1	4436	4.8×10^{-2}	2.2
4	5304	2.2×10^{-2}	2.6	4289	4.2×10^{-2}	2.1
3	5868	3.1×10^{-2}	2.7	4511	4.0×10^{-2}	2.4
2	9529	2.7×10^{-2}	4.4	8259	3.7×10^{-2}	3.9
1	39576	4.1×10^{-3}	20.0	28813	1.5×10^{-2}	14.9

The colored dots in Figure 6.8 indicate that the time integration is mainly done with the NDF(3) and NDF(4) method. This leads us to the investigation of the influence of the maximum order of the time integration scheme on the overall performance. In Table 6.1 we list the number of time steps, the maximum time step size as well as the wall-clock time in hours depending on the maximum order of the variable-step, variable-order time integration scheme. Additionally we give the same data for the solution with the variable-step, variable-order algorithm based on BDF(k) methods. Note, that these values only represent the results of a single run. Due to the random initial data the comparison of averaged values would lead to a more reliable quantitative comparison. As above we obtain these val-

ues by solving the Cahn–Hilliard equation with 16 MPI ranks on the *IvyBridge* partition. The comparison of the time step statistics demonstrates the additional computational savings due to the variation of the order and illustrates the advantage of the NDF(k) methods over the BDF(k) methods. In particular, the variable-step, variable-order algorithm based on NDF(k) methods allows the usage of larger time step sizes and needs less time steps compared to the algorithm based on BDF(k) methods. Further, the usage of higher order methods drastically reduces the number of time steps and thus the computational time. For example the variable-step, variable-order algorithm based on NDF(k) methods up to order five compared to the variable-step BDF(1) method reduces the computational time approximately by the factor ten.

We conclude that the employed variable-step, variable-order algorithm is able to capture the phase separation dynamics. Moreover, due to the strongly varying time scales, the adaptive time integration is inevitable for the efficient solution of the Cahn–Hilliard equation and, in particular, the variation of the order generates additional computational savings.

Size of the Arnoldi Basis For the solution of the linearized system (4.33) in each Newton step we employ the GMRES method, which incorporates a restart as soon as the number of iteration steps exceeds the size of the Arnoldi basis. For the Cahn–Hilliard equation we observe a significant impact of the size of the Arnoldi basis on the maximum number of GMRES steps per time step.

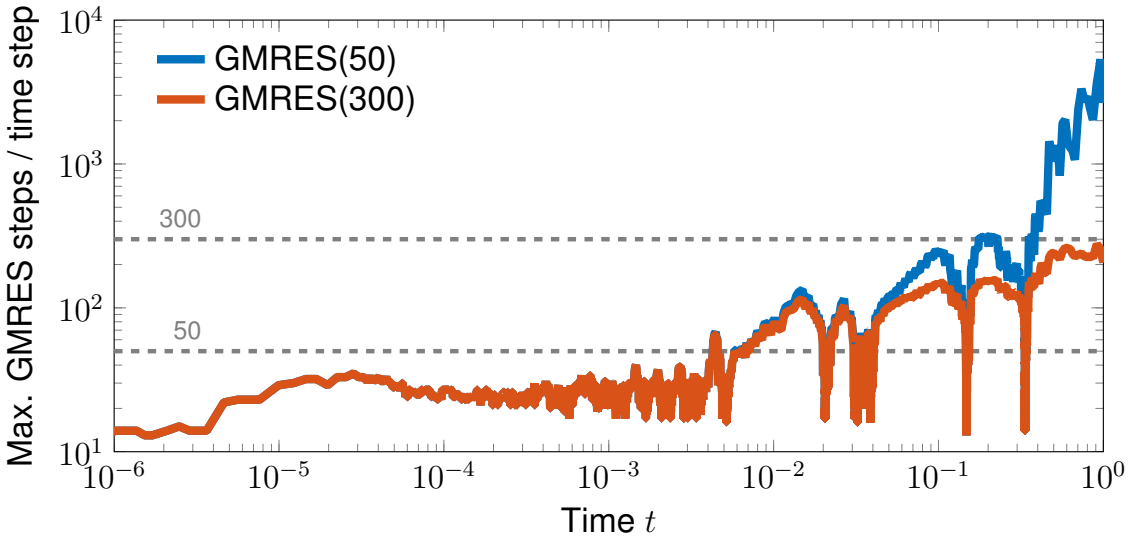

Figure 6.9.: Maximum number of GMRES steps per time step over time for different sizes of the Arnoldi basis.

In Figure 6.9 we plot the maximum number of GMRES steps per time step for two different sizes of the Arnoldi basis over time. If we use the GMRES method with restart after 50 iterations, the maximum number of GMRES steps exceeds the size of the Arnoldi basis in the late coarsening stage $t \gtrsim 10^{-2}$ and even overshoots the mark of 1000 steps for the equilibrium state. Varying the size of the basis, we find that for approximately 300 basis vectors the GMRES method does not need to restart at all. In particular, for the final equilibrium state the number of GMRES steps is drastically reduced.

The storage of 300 basis vector seems to be highly inefficient in terms of memory consumption. However, in fact the storage of these basis vectors is less memory consuming compared to the storage of the sparse system matrix. In particular, with the matrix-free framework we create computational savings on two levels. First, the evaluation of matrix-vector products is significantly faster compared to the matrix-based approach and second the reduced memory requirement due to the obsolete system matrix allows the usage of the GMRES method with larger Arnoldi basis, which can reduce the number of GMRES steps dramatically.

Note, the size 300 is a rather arbitrary choice, which might be fitted to a specific problem setting. However, this basis size is also suited for the later electrode particle simulations in Section 6.2. In fact, it is a compromise between the memory consumption and the history of the GMRES method.

Example 2: Random Initial Data

In the second part of this subsection we again assume a random initial perturbation, but here with average concentration $\bar{c}_0 = 0.7$ and amplitude $\varepsilon = 0.05$. With this initial condition we solve the classical Cahn–Hilliard equation (6.1) with degenerated mobility $m(c) = 600c(1-c)$, the interfacial energy coefficient $\kappa = 1/600$ and a logarithmic homogeneous free energy density

$$f(z) = 3z(1-z) + z\log(z) + (1-z)\log(1-z), \tag{6.13}$$

on the unit square $\Omega = (0,1)^2$ until the final time $t_\mathrm{f} = 1$, as studied in [74].

With this example we analyze the performance of our proposed preconditioner (4.35) for the classical Cahn–Hilliard equation. In particular, we investigate the influence of the incorporation of the nonlinear mobility in our preconditioner and compare the results to the preconditioner with constant mobility (4.45). Finally, we demonstrate the capabilities of our parallel matrix-free solver and simulate a three-dimensional spinodal decomposition. As above we cut off the initial stage in the visualization of the time step data (energy, time step size, number of GMRES steps) over time by starting the time axis at 10^{-6}.

Numerical Solution We discretize the domain with $n_\mathrm{cells} = 2^6 \times 2^6 = 4096$ cells, distribute quadratic ($p = 2$) finite elements ($N_x = 33\,282$) and perform the time integration with the variable-step, variable-order algorithm starting with the initial time step size $\tau = 10^{-12}$. Without spatial adaptivity our matrix-free finite element solver computes 1308 time steps until the final time in less than one hour with 16 MPI ranks on the *IvyBridge* partition.

In Figure 6.10a–i we visualize a series of snapshots of the concentration distribution, where we observe the spinodal decomposition process, compare [73, 74]. The basic stages of spinodal decomposition are similar to the previous example with constant mobility and quartic homogeneous free energy density shown in Figure 6.5a–i. However, the degenerated mobility limits the transport by diffusion to the phase transition zones. Further, due to the logarithmic free energy density, the range of the concentration is bounded to the range $[0, 1]$.

Preconditioning As the considered Cahn–Hilliard equation of this example includes the degenerated mobility, we can study the influence of the incorporation of the nonlinear

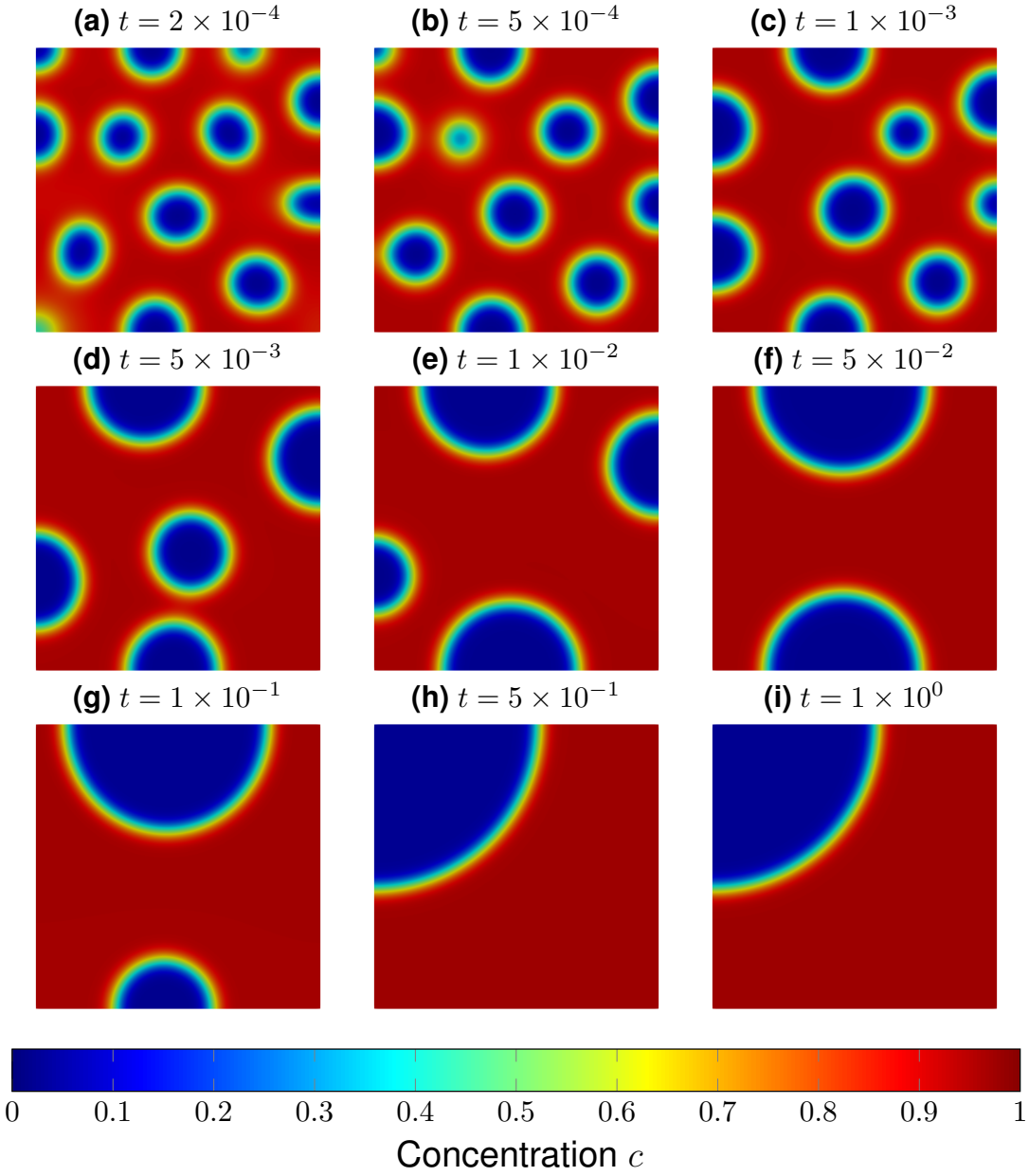

Figure 6.10.: Numerical solution of the Cahn–Hilliard equation with degenerated mobility and logarithmic free energy density. (a–i) Snapshots of the spinodal decomposition process at different time instances.

mobility in our proposed preconditioner. Therefore we solve this example with linear finite elements on a series of successively uniform refined meshes and for finite element methods of increasing degree holding the number of DOFs approximately constant. Additionally, we solve the example for all cases again but replace our proposed preconditioner (4.35) by the preconditioner with constant mobility (4.45).

In Figure 6.11 we plot the maximum number of GMRES steps per time step over time. For both preconditioners we observe the independence from the mesh refinement (Figure 6.11a) and the finite element degree (Figure 6.11b). Moreover, the experiment shows, that the incorporation of the nonlinear mobility in the preconditioner significantly reduces

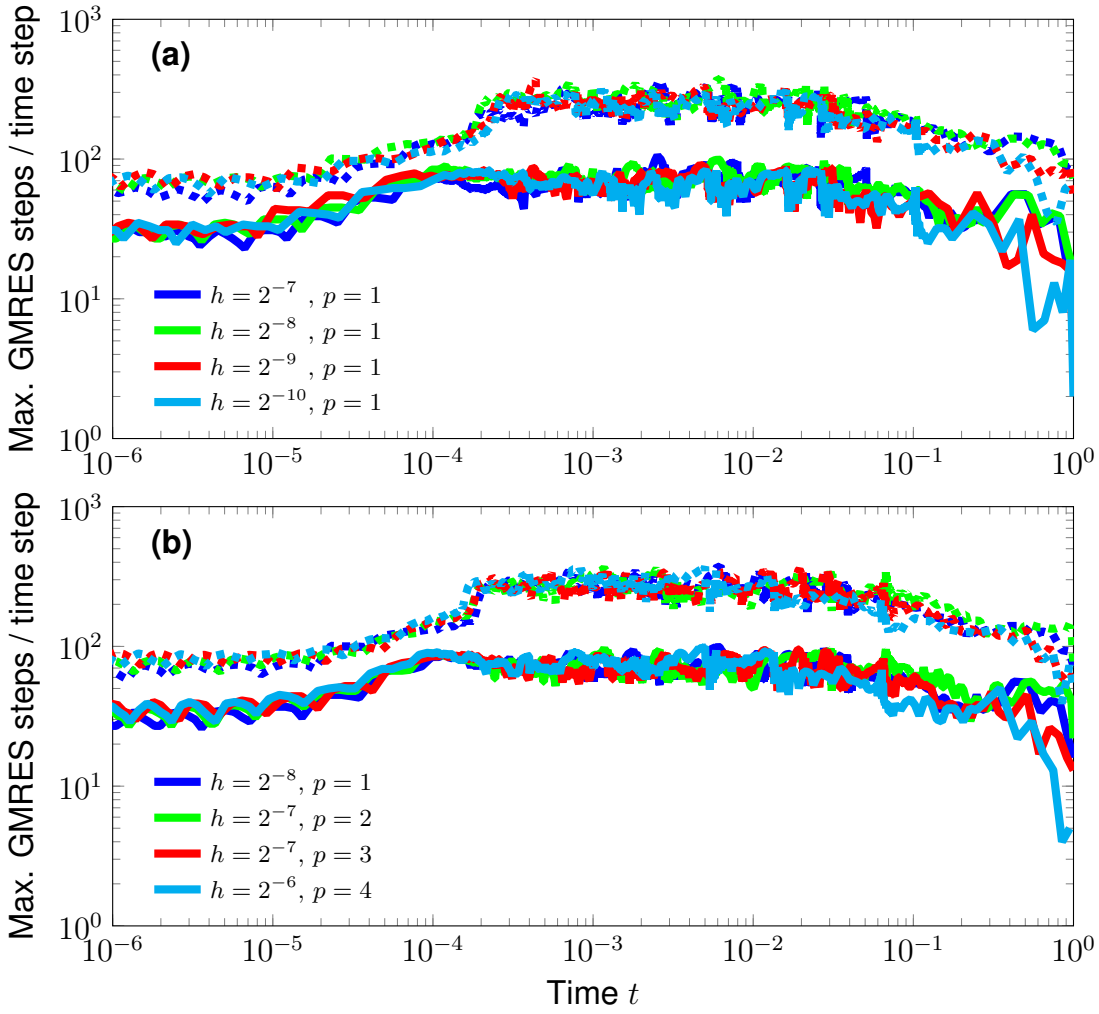

Figure 6.11.: Maximum number of GMRES steps per time step over time for (a) linear finite elements and a uniform refined mesh and (b) for an increasing finite element degree while keeping the number of DOFs equal. Solid lines represents the proposed preconditioner (4.35) with nonlinear mobility, dashed lines the preconditioner with constant mobility (4.45).

the number of iteration steps. Note, that we observe minor fluctuations in the number of GMRES steps per time step, because we did not use a random initial condition, which is unique for all meshes and distributions of DOFs according to finite element methods of different order.

In addition, we compare the average of the maximum number of GMRES steps per time step and the computational wall-clock times in Table 6.2. The comparison shows that our proposed preconditioner with nonlinear mobility also reduces the overall computational time.

Three-Dimensional Spinodal Decomposition Finally, we demonstrate the large scale capabilities of our parallel matrix-free finite element solver for the Cahn–Hilliard equation by simulating the three-dimensional spinodal decomposition. Therefore we consider the Cahn–Hilliard equation with degenerated mobility and logarithmic free energy

Table 6.2.: Comparison of preconditioning with constant and nonlinear mobility (Number of linear systems to solve n_{solve}, average maximum number of GMRES steps per time step $\overline{n}_{\text{GMRES}}$) for different spatial discretizations. Wall-clock time in hours for the whole solution process (without graphical output).

		$P_\star$ (4.45)			P (4.35)		
h	p	n_{solve}	$\overline{n}_{\text{GMRES}}$	Wall time / h	n_{solve}	$\overline{n}_{\text{GMRES}}$	Wall time / h
2^{-7}	1	1092	157	0.8	1553	47	0.5
2^{-8}	1	2031	191	3.2	1941	54	2.0
2^{-9}	1	1510	179	7.5	1445	50	4.6
2^{-10}	1	1677	166	33.6	1733	45	20.2

density, as above, on the unit cube $\Omega = (0,1)^3$ and solve the equation with quadratic finite elements on a mesh with $262\,144$ cells, resulting in $4\,293\,378$ DOFs. With 40 MPI ranks on the *Skylake* partition our matrix-free solver takes approximately one day for this simulation. Visualized in Figure 6.12 we observe the usual spinodal decomposition behavior, solved with 1871 time steps starting from the initial time step size $\tau = 10^{-12}$.

In addition to the numerical study of our preconditioner in the previous paragraph, we plot in Figure 6.13 the maximum number of GMRES steps per time step for the three-dimensional example in comparison to the two-dimensional case. Up to small fluctuations, which are due to the different initial conditions, we observe the same number of iterations for both dimensions. This comparison supports the mesh independence of our preconditioner.

Example 3: Smooth Initial Data

According to [55, Sect. 5.4] we replace in the last part of this subsection the random initial data by a smooth perturbation of the average concentration $\bar{c}_0 = 0.5$ based on sine and cosine terms

$$c_0(x,y) = 0.5 + 0.17\cos(\pi x)\cos(2.0\pi y) + 0.2\cos(3.0\pi x)\cos(\pi y), \tag{6.14}$$

and solve the classical Cahn–Hilliard equation (6.1) with constant mobility $m = 0.25$, the interfacial energy coefficient $\kappa = 10^{-4}$ and a quartic homogeneous free energy density $f(z) = 1/4\, z^2(1-z)^2$ on the unit square $\Omega = (0,1)^2$ until the final time $t_{\text{f}} = 10$.

Note, that we presented the numerical solution of this example already in [41], where we employed a uniform discretization with finite elements and used the adaptive time integration scheme from [74]. However, in this thesis we demonstrate with this example the functionality of our coupled space and time adaptive solution algorithm presented in Section 4.3.

Space and Time Adaptive Solution Starting from a uniform mesh with $n_{\text{cells}} = 2^7 \times 2^7 = 16\,384$ cells and $N_x = 526\,338$ DOFs distributed according to the fourth-order finite element method, we solve this example with our space and time adaptive solution algorithm with 16 MPI ranks on the *IvyBridge* partition. For the validation of the fully

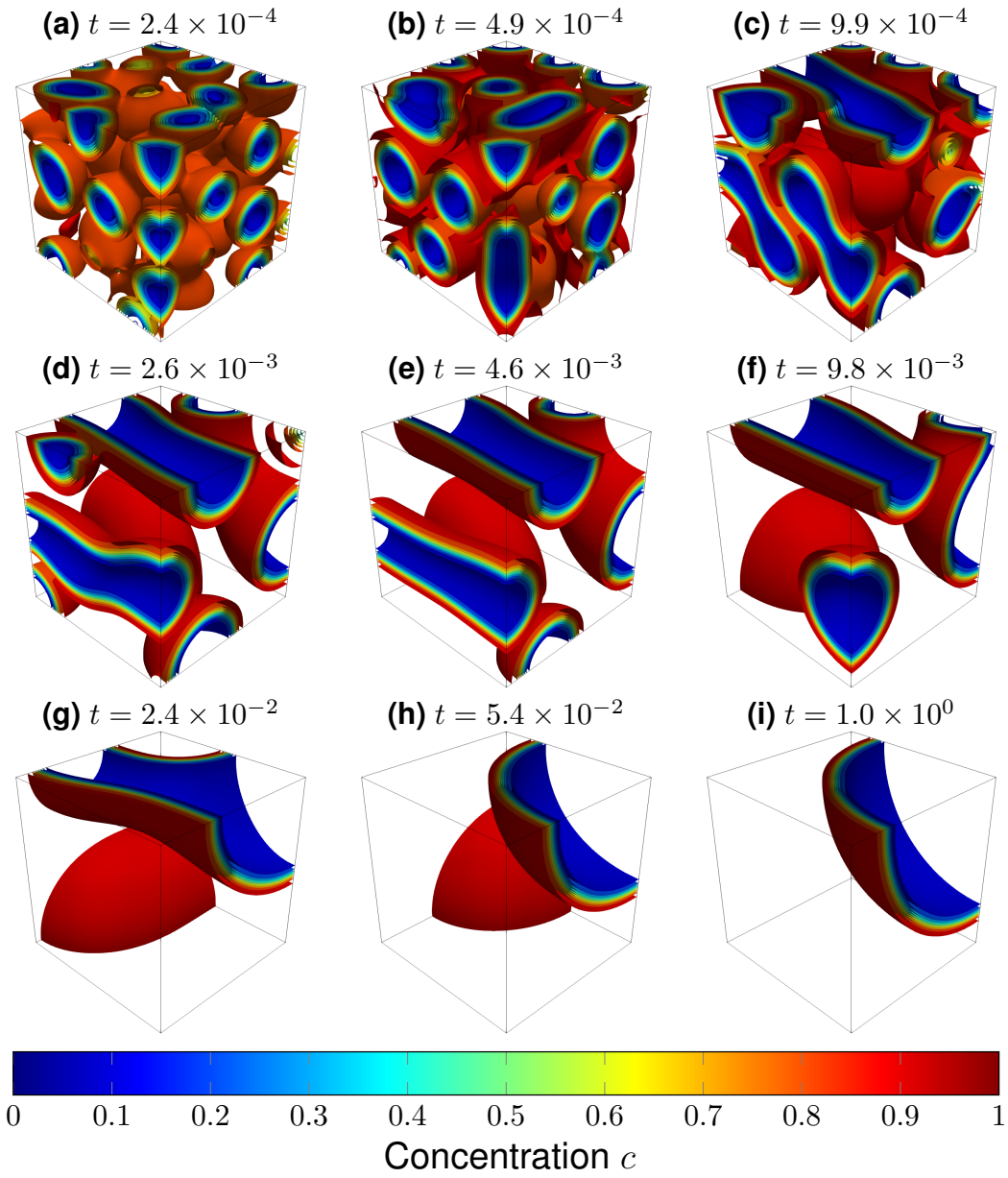

Figure 6.12.: Numerical solution of the Cahn–Hilliard equation with degenerated mobility and logarithmic free energy density. (a–i) Snapshots of contour plot of the three-dimensional spinodal decomposition process at different time instances.

adaptively computed solution, we solve this example also on the uniform initial mesh with disabled spatial adaptivity. We start the algorithm in both cases with the initial time step size $\tau = 10^{-6}$ and solve 264 (263) time steps in the spatially uniform (adaptive) case.

In Figure 6.14 we depict the numerical approximation of the concentration profile at the initial and final time with the underlying locally refined mesh. The uniform initial mesh is fine enough to distribute a sufficient number of DOFs for the accurate resolution of the phase transition. When we enable the spatial adaptivity based on the gradient recovery estimator as refinement criterion, we start with the uniform mesh and as time evolves the

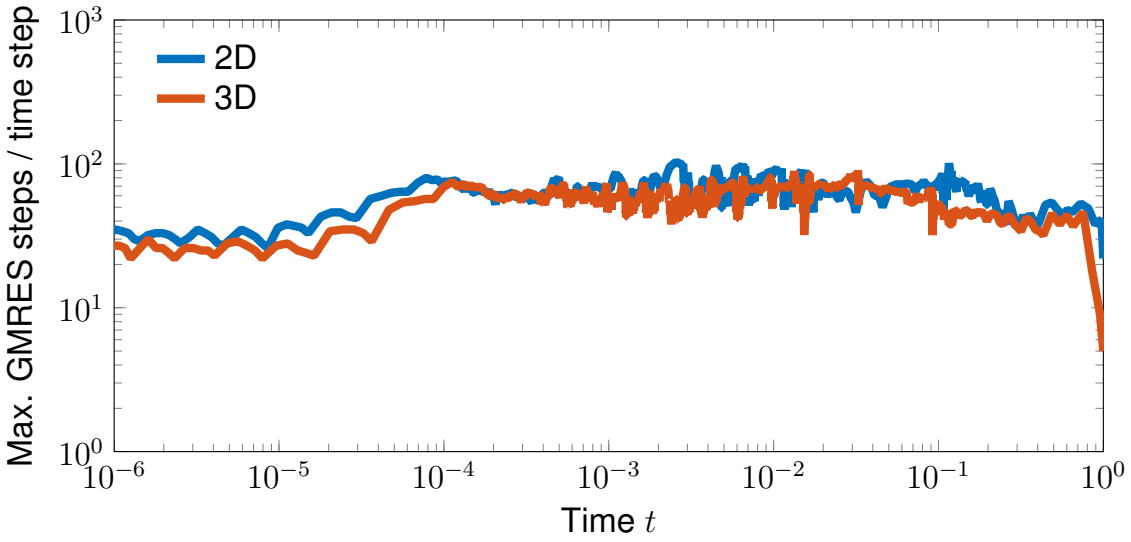

Figure 6.13.: Maximum number of GMRES steps per time step over time for the two- and three-dimensional spinodal decomposition solved with the full preconditioner (4.35).

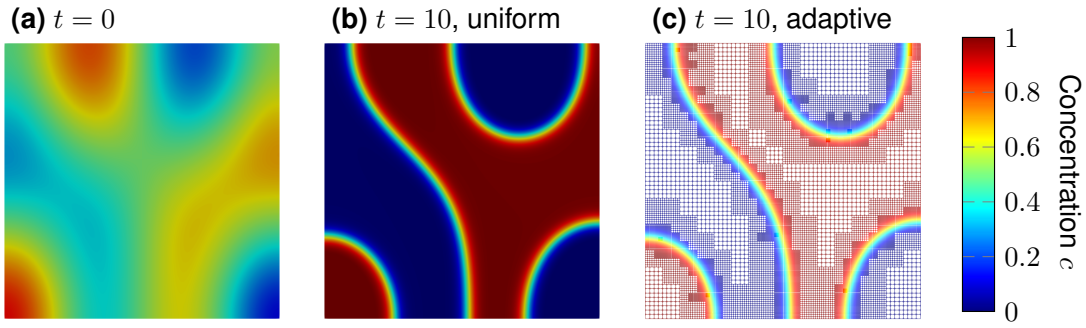

Figure 6.14.: Numerical solution of the concentration profile at (a) initial and (b/c) final time solved on (b) a uniform and (c) a locally refined mesh based on the gradient recovery criterion.

mesh is locally coarsened and refined. In Figure 6.14c we see that the phase transition zone is accurately resolved with a locally refined mesh, whereas the regions of pure phases are locally coarsened. In particular, the numerical solution of the concentration, solved with our space and time adaptive solution algorithm in Figure 6.14c, is in accordance with the spatially uniform solved concentration profile in Figure 6.14b.

Finally, we investigate the influence of the spatial adaptivity on the number of DOFs, the time step size, the maximum number of GMRES steps per time step and the free energy. Therefore we plot the temporal evolution of these quantities in Figure 6.15. As the phase separation starts approximately at time $t = 8 \times 10^{-2}$ we start the time axis at $t = 10^{-2}$. In Figure 6.15a we first show the number of DOFs over time. When we use a uniform mesh the number of DOFs is obviously constant, but in the spatially adaptive case the initially uniform mesh is immediately coarsened after some initial time steps in the early smoothing stage. In the following phase separation and coarsening stage the number of DOFs is by a

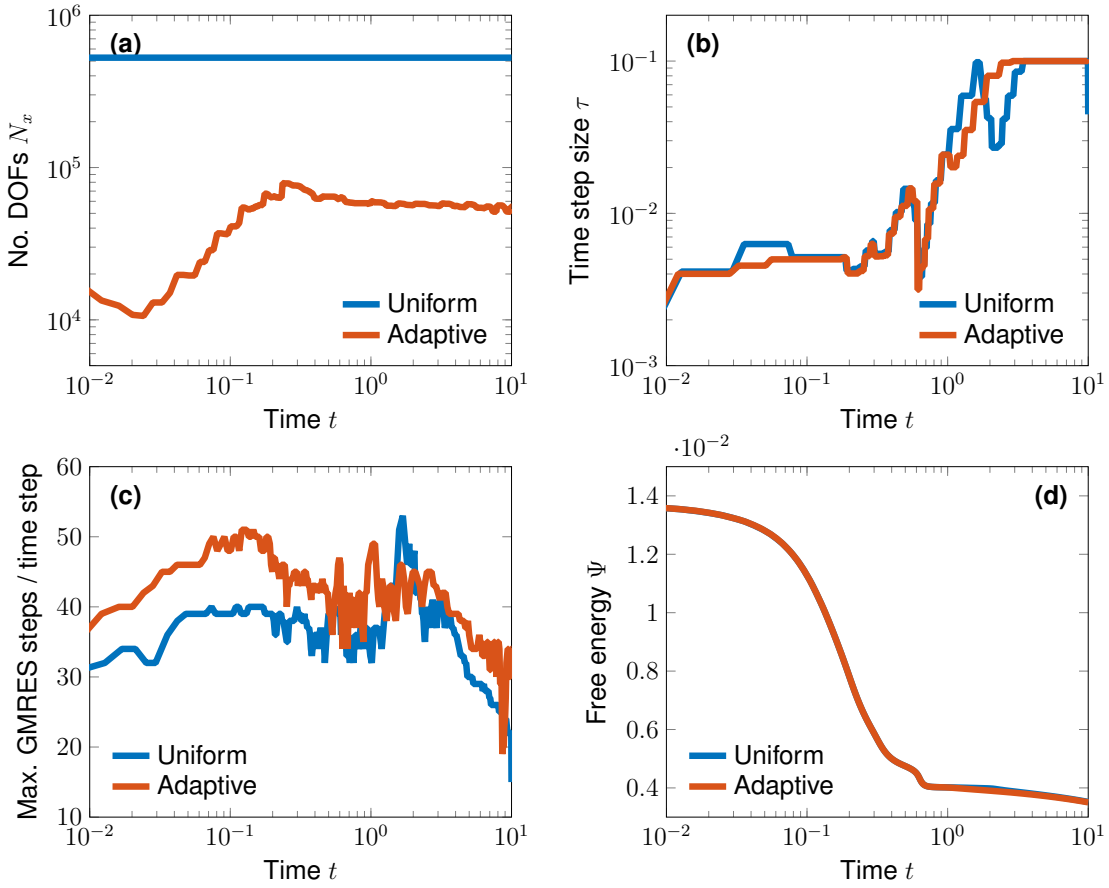

Figure 6.15.: (a) Number of DOFs, (b) time step size, (c) maximum number of GMRES steps per time step and (d) free energy over time during the numerical solution with uniform and locally refined mesh.

factor ten smaller than in the uniform case. In general, the evolution of the time step sizes (Figure 6.15b) and the free energy (Figure 6.15d) over time is in agreement with the one in the spatially uniform case. In contrast, the maximum number of GMRES steps per time step (Figure 6.15c) in the adaptive case is a bit larger than in the uniform case. Further, however, we observe a deviation of the time step size, the maximum iteration number and the free energy around the time $t = 2$, which indicates that the coarsening process is tracked differently in the adaptive case.

We have seen at this example that our space and time adaptive solution algorithm works in principle and creates computational savings due to the reduced number of DOFs. In general the adaptive mesh refinement criteria and their parameters and tolerances need always to be chosen carefully, but the numerical solution of the Cahn–Hilliard equation is extremely sensitive not only to local mesh refinement. Already small differences in the computational setup can ultimately lead to totally different phase transition morphologies during the coarsening stage. The robustness of our adaptive finite element solver is an open issue and might be fixed by using more reliable a posteriori error estimators for the Cahn–Hilliard equation, as for example studied in [19, 20, 23, 24, 71, 161].

6.2. Phase Separation in Electrode Particles

So far we have validated our developed solver and analyzed the components of the solution algorithm based on examples of the classical Cahn–Hilliard equation. In the following section we switch from the classical Cahn–Hilliard equation (6.1) to our model problem (3.19) and simulate the phase separation in LFP electrode particles during lithium insertion. Firstly, in Subsection 6.2.1 we consider the model problem under the common assumption of spherical symmetry and investigate the performance of our solver. Secondly, in Subsection 6.2.2 we study our solver and the phase transition morphology for isotropic and anisotropic material properties in a two-dimensional computational domain under the azimuthal symmetry assumption, discussed in Appendix B.2.2.

Model Parameters Throughout the following experiments we use the material parameters for LFP electrode particles, taken from [168, 169] and listed in Table A.1. However, we perform our numerical experiments with normalized model parameters. In particular, if not stated otherwise, we assume at the initial time $t = 0$ a constant normalized concentration $\bar{c}_0 = 0.01$ and apply an external lithium flux N_{ext} corresponding to the C-rate equal to one. Note, that the concrete value of the boundary condition depends on the particle geometry according to Equation (3.16). The set of normalized model parameters for $N_{\text{ext}} = 1\,\text{C}$ is given in Table A.3.

Mesh Width For the accurate resolution of the phase transition we distribute at least ten DOFs in the interfacial zone. According to Equation (3.7) the width of the phase transition is proportional to $\sqrt{\kappa}$. Thus we obtain the estimate $h < p\sqrt{\kappa}/10$ for the minimal mesh width, knowing that for a p-th-order Lagrangian finite element method the approximate distance of two DOFs is about h/p.

6.2.1. Spherical Symmetric Solution Approach

Under the assumption of a spherical symmetric solution we solve the model problem (3.19) for phase separation during the lithiation of a LFP electrode particle. In this subsection we validate our solver by determining the experimental order of convergence, demonstrating that we can efficiently solve the model equations with our space and time adaptive solution algorithm and showing numerically the optimality of our preconditioner, even for locally refined meshes.

Since the computational domain reduces to a one-dimensional interval, which can currently in *deal.II*-v9.0 not be distributed among different MPI ranks with *p4est*, we perform these experiments with a matrix-based implementation. Thereby we replace the inner GMG preconditioners for the inverse application of the two preconditioner blocks by a Jacobi preconditioner. However, we emphasize, that we do not introduce a dependence on any matrix. In this sense the solver is still matrix-free.

Note, that the numerical study in this subsection is an improved and extended version of our previous work [43], with discussions and formulations adopted from our submitted article [44].

Spherical Symmetry Assuming a spherical symmetric solution we reduce the computational domain to the one-dimensional unit interval $\Omega = (0, 1)$, representing the radial line from the particle center $\Gamma_0 = \{0\}$ to the particle surface $\Gamma_{\text{ext}} = \{1\}$, see Figure B.1. To preserve the symmetry we impose homogeneous Neumann boundary conditions at the artificial boundary in the particle center

$$\nabla c \cdot \boldsymbol{n} = \boldsymbol{N} \cdot \boldsymbol{n} = 0 \quad \text{on } (0, t_{\text{f}}) \times \Gamma_0. \tag{6.15}$$

We translate the spherical symmetry into the discrete finite element formulation by using the modified quadrature weight $\mathrm{d}\boldsymbol{x} = 4\pi r^2 \, \mathrm{d}r$, compare Appendix B.2.1.

Numerical Solution First, we solve the model equations numerically with fourth-order finite elements and our space and time adaptive solution algorithm, as explained in Section 4.3. The initial mesh consists of 128 cells and 1026 DOFs and the time integration takes 1194 steps, beginning with the initial time step size $\tau = 10^{-6}$. In Figure 6.16 we plot snapshots of the numerical solution for the concentration and the chemical potential at three characteristic stages.

From the constant initial concentration, the inhomogeneous boundary condition modeling lithium insertion leads to an initially uniform increase of the concentration in the whole particle. Once the concentration at the particle surface reaches the critical value, which is related to the location of the inflection point of the homogeneous chemical free energy density ($\psi''_{\text{ch}}(z) = 0$), phase separation is initiated (Figure 6.16a). At this point a phase transition emerges and the concentrations in the pure phases tend to their equilibrium values $\overline{c}_\alpha$, $\overline{c}_\beta$. In the next stage, further lithium insertion drives the phase transition through the particle towards the center (Figure 6.16b). Finally, the phase transition reaches the particle center and the phase transition decays because the amount of lithium is so high that the lithium rich phase will occupy the complete particle (Figure 6.16c). Keeping in mind that this is a spherical symmetric solution, we recognize the core-shell scenario during lithiation. During phase separation, a lithium rich shell forms around the lithium poor core, which is then occupied by the growing shell.

In addition, we observe that the chemical potential is initially constant with values according to the derivative of the homogeneous chemical free energy density ψ'_{ch}. As phase separation is initiated the chemical potential at the surface drops (Figure 6.16d). During the migration of the phase transition the chemical potential is then again approximately constant at a small value below zero (Figure 6.16e). Finally, the peak in the chemical potential correlates to the decay of the phase transition in the particle center (Figure 6.16f).

The numerically solved concentration profiles and the evolution of the free energy in Figure 6.17 are in qualitative accordance with the results reported in [169] for a slightly different particle size. In Figure 6.17 we observe the discussed nonmonotonous energy evolution, compare (3.23). Especially we note the energy increase in the initial stage for single-phase diffusion until SOC ≈ 0.127, when the phase separation is initiated. Additionally, we note that the SOC of our numerical solution is linearly increasing according to the relation (3.18).

Convergence Analysis We validate our solver for the model problem numerically by measuring the experimental order of convergence. Because no exact solution is known

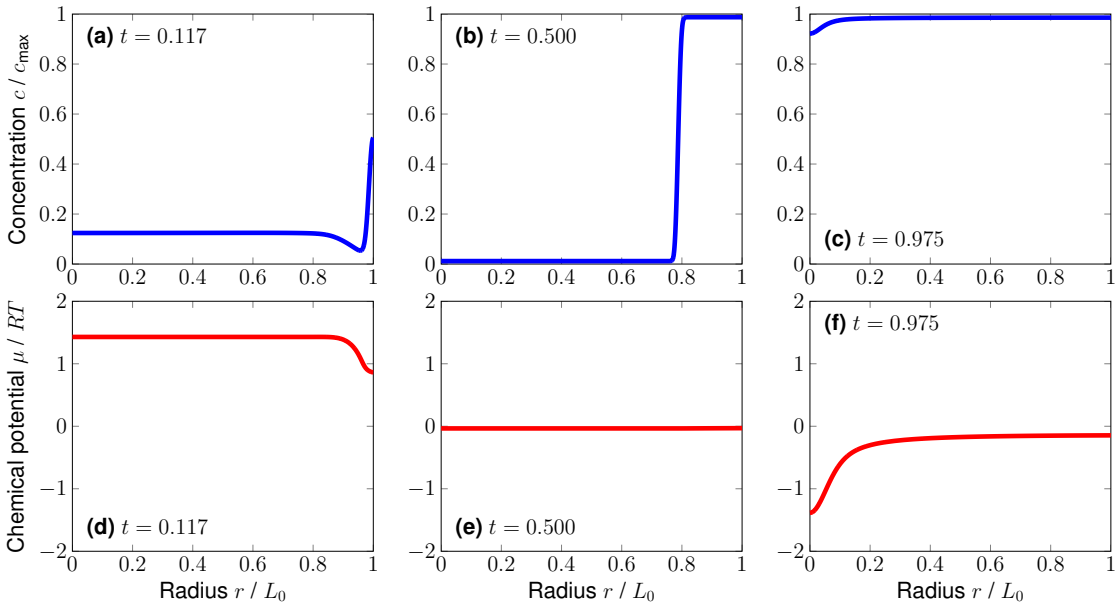

Figure 6.16.: (a/b/c) Concentration and (d/e/f) chemical potential inside the particle at three characteristic times for lithium insertion. (a/d) Initiation of phase separation. (b/e) Migration of the phase transition. (c/f) Vanishing of the phase transition.

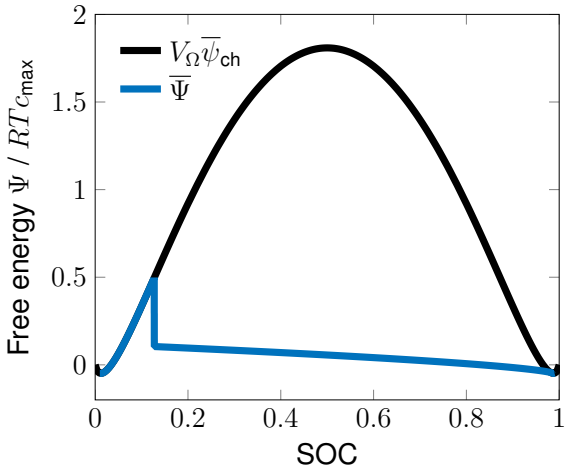

Figure 6.17.: Evaluated free energy ($\overline{\Psi} = \Psi(c_h)/RTc_{\max}$) over the SOC compared to the homogeneous free energy.

for this problem, we consider the L^2- and H^1-errors of the solution $y_h = [c_h, \mu_h]^\mathsf{T}$ to a reference solution y_r at time $t_\mathrm{f} = 0.29$ (SOC $= 0.3$), when a phase transition is present

$$\mathrm{err} := \left(\|c_h(t_\mathrm{f}, \cdot) - c_\mathrm{r}(t_\mathrm{f}, \cdot)\|^2 + \|\mu_h(t_\mathrm{f}, \cdot) - \mu_\mathrm{r}(t_\mathrm{f}, \cdot)\|^2 \right)^{1/2}. \tag{6.16}$$

The reference solution y_r is assumed to approximate the unknown exact solution y_ex more accurately than y_h and is computed once with fourth-order finite elements and the tolerances $\mathrm{RelTol}_t = 10^{-8}$, $\mathrm{AbsTol}_t = 10^{-11}$ and $\mathrm{RelTol}_x = 10^{-20}$, $\mathrm{AbsTol}_x = 10^{-10}$.

For the computation of the experimental order of convergence of the spatial error we fix the tolerances of an accurate time integration ($\mathrm{RelTol}_t = 10^{-7}$, $\mathrm{AbsTol}_t = 10^{-10}$)

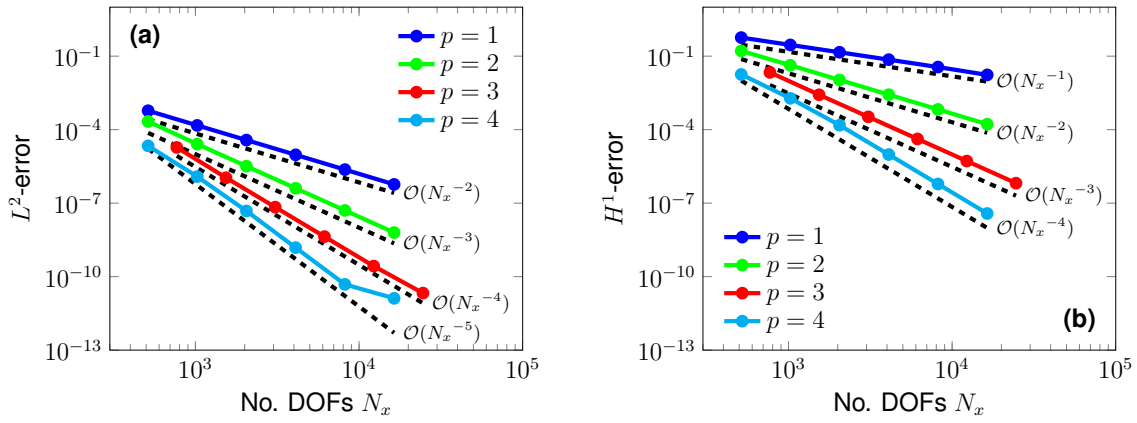

Figure 6.18.: Convergence analysis of the (a) L^2- and (b) H^1-error for uniform spatial refinement with fixed tolerances for the adaptive time integration.

and compute the errors of the numerical solution on a series of successively uniformly refined meshes. In Figure 6.18 we plot the L^2- and the H^1-errors over the number of DOFs. According to the theory of finite elements [30], our implementation shows the optimal order of convergence for the finite element degrees one to four. Note, that the L^2-errors for fourth-order elements at the finest level already reach a value of 10^{-11} when the saturation sets in. A convergence analysis of the time integration with fixed spatial tolerances ($\mathrm{RelTol}_x = 10^{-10}$, $\mathrm{AbsTol}_x = 10^{-12}$) for fourth-order finite elements yields errors of order 10^{-8}.

With the convergence analysis based on the fully adaptively computed reference solution we validated simultaneously the spatial adaptivity component of our solution algorithm, because we demonstrated that the numerical solutions, computed on uniform meshes, converge to the spatially adaptively computed reference solution.

To reduce the computational time of this convergence analysis, we solved the model equations with a matrix-based implementation, where we solely replaced the GMRES method by the LU-decomposition for the solution of the linearized systems (4.33).

Space and Time Adaptivity The high demand on spatial and temporal adaptive methods and large computational savings in terms of the number of DOFs N_x and time steps N_t for this application problem becomes immediately clear when we look into Figure 6.19 and 6.20.

In Figure 6.19a we plot the concentration profile in the phase separated state at time $t = 0.5$. Because the concentration is approximately constant except for the phase transition, a locally high resolution of the transition is advantageous. Accordingly, our solution algorithm distributes the DOFs adaptively considering the gradient recovery estimates with respect to both solution variables (c_h, μ_h). In addition, the black circles in Figure 6.19a indicate the refinement level $n \in \mathbb{N}$ related to the cell diameter by $h = 2^{-n}$. We observe that the mesh is locally refined precisely in the phase transition zone. Over time the phase transition migrates through the particle and the mesh is adapted. In Figure 6.19b we plot the number of DOFs over time, for the solution of the model equations with different finite element degrees. We observe the three states of single and two-phase diffusion. In particular, we find that higher order methods are beneficial for saving DOFs, which is due to

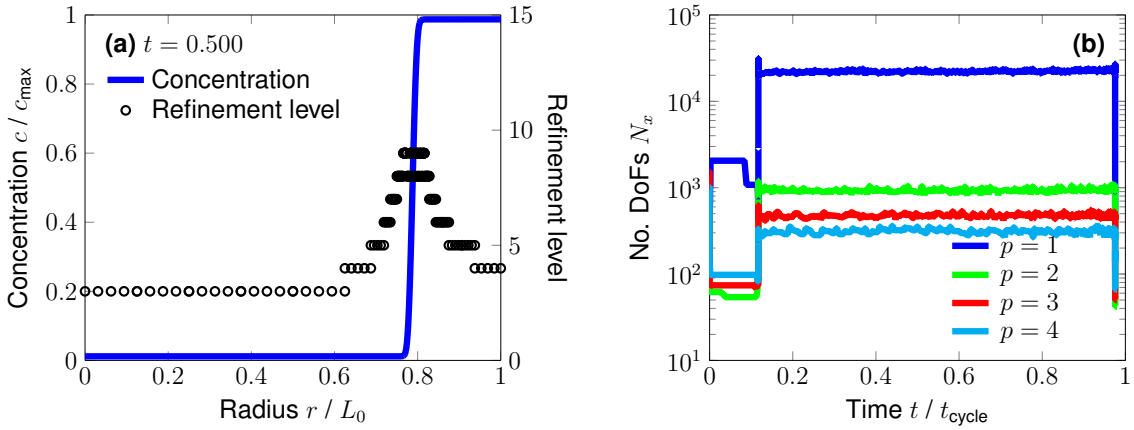

Figure 6.19.: (a) Concentration profile at the time $t = 0.5$ with refinement level and (b) number of DOFs over time for the solution with different finite element degrees.

better approximation properties of higher order derivatives in the solution profiles, as for example in the chemical potential profile.

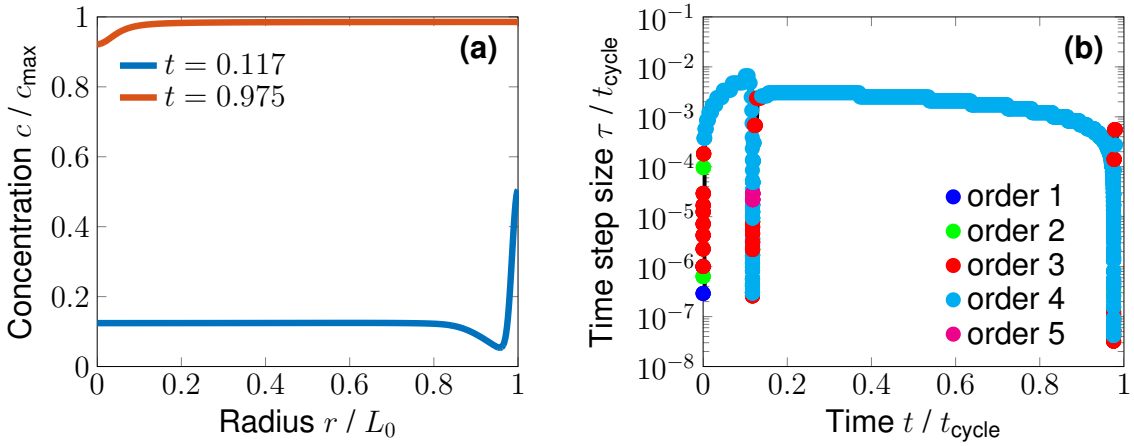

Figure 6.20.: (a) Concentration profile at initiation and decay of the phase transition. (b) Time step size with used orders (markers) over time.

The variable-step, variable-order time integration scheme is an important key for the efficient solution. Thereby we particularly benefit from higher order methods, which allow even larger time step sizes at the same error tolerance, while the essential gain of efficiency is due to the large time step sizes during the two-phase diffusion state. In Figure 6.20b we plot the time step size and for each step the order of the used multistep method over time. We observe that the time integration is primarily done by the fourth-order method and, as before, identify the three regimes of single and two-phase diffusion. Moreover, the jumps of the time step size over several orders of magnitude are related to the initiation and decay of the phase transition, see Figure 6.20a. During two-phase diffusion up to the decay of the phase transition we observe a decreasing time step size. This reflects the observation that in the spherical symmetric case the phase transition migrates with increasing velocity towards the particle center. Thus, to keep the tolerances of the time integration, smaller

time step sizes are necessary for the accelerating process.

Preconditioning To show the performance of our proposed preconditioner (4.35) in the application case, compared to the constant mobility version (4.45), we study the maximum number of GMRES steps needed to solve a time step over time.

For this analysis we first solve the model equation (3.19) with linear finite elements for a series of successively uniformly refined meshes, see Figure 6.21. Once the phase transition is fully resolved, the number of GMRES steps for our proposed preconditioner (4.35) with nonlinear mobility is practically independent of further mesh refinement, see Figure 6.21a. In Figure 6.21b, for the constant mobility preconditioner (4.45), we also observe the saturation of the maximum number of GMRES steps per time step but at a value, which is one order of magnitude larger compared to our nonlinear mobility preconditioner. In particular, over all considered uniform meshes the iteration number is higher than with our proposed preconditioner (4.35).

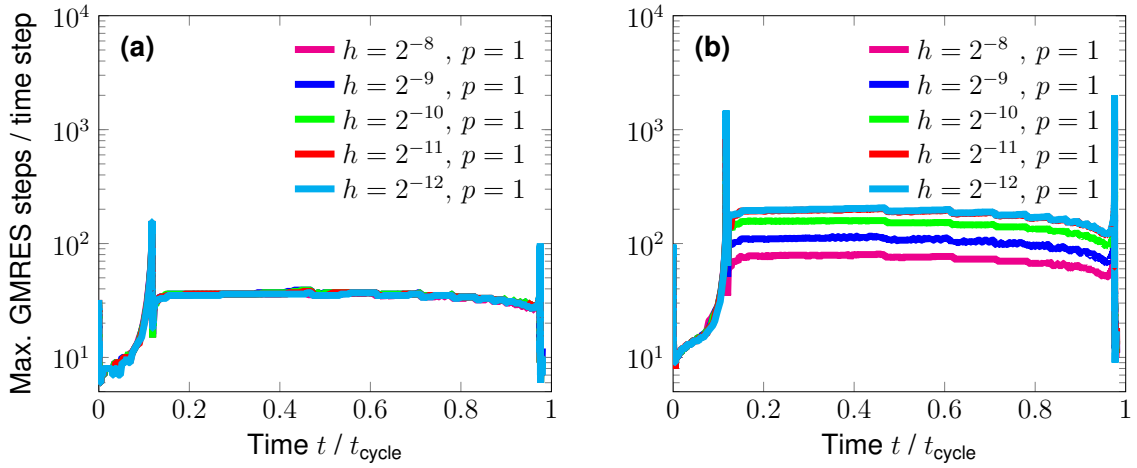

Figure 6.21.: Maximum number of GMRES steps per time step over time during the solution with linear finite elements on uniformly refined meshes using (a) the proposed preconditioner (4.35) and (b) the constant mobility preconditioner (4.45).

In Figure 6.22a we see that the variation of the finite element degree has no significant influence on the number of iteration steps for our proposed preconditioner (4.35). In particular, the maximum iteration number in the two-phase diffusion state is the same as for different uniform refinements in Figure 6.21a. The constant mobility preconditioner (4.45) is practically also independent on the finite element degree. However, we observe for third-order finite elements a slightly higher number of iteration steps. This can be explained by the fact, that the discretization with third-order finite elements distributes 1538 DOFs, whereas the linear, quadratic and fourth-order finite elements distribute 1026 DOFs. Thus the higher number of DOFs leads to an increased number of GMRES steps, as the above mentioned saturation level is not yet reached.

Finally, we allow local mesh refinement with our adaptive solution algorithm from Section 4.3 and visualize the iteration numbers for the solution with different finite element degrees in Figure 6.23. Since our proposed preconditioner (4.35) shows in Figure 6.23a the same performance as in the uniform refined case, we conclude the mesh independence of

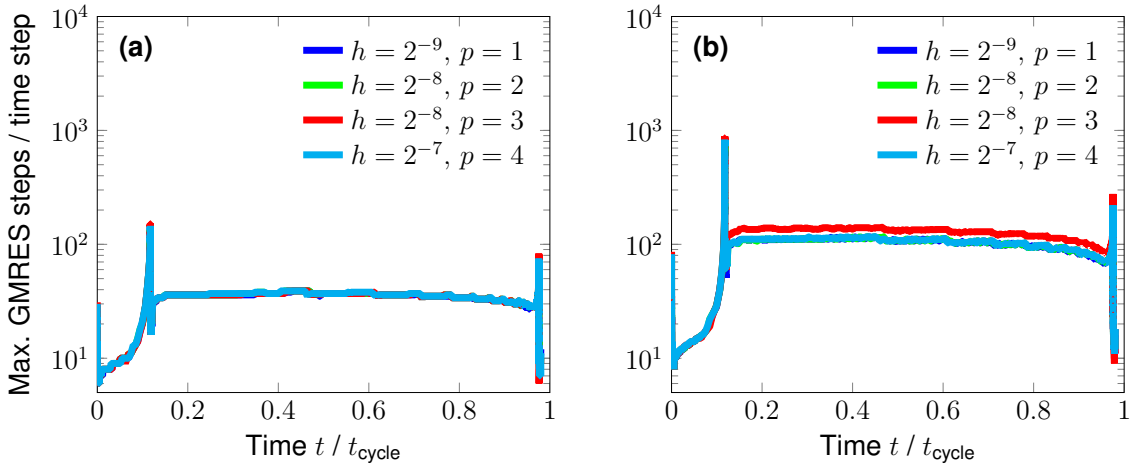

Figure 6.22.: Maximum number of GMRES steps per time step over time during the solution with different finite element degrees at the approximately same number of DOFs using (a) the proposed preconditioner (4.35) and (b) the constant mobility preconditioner (4.45).

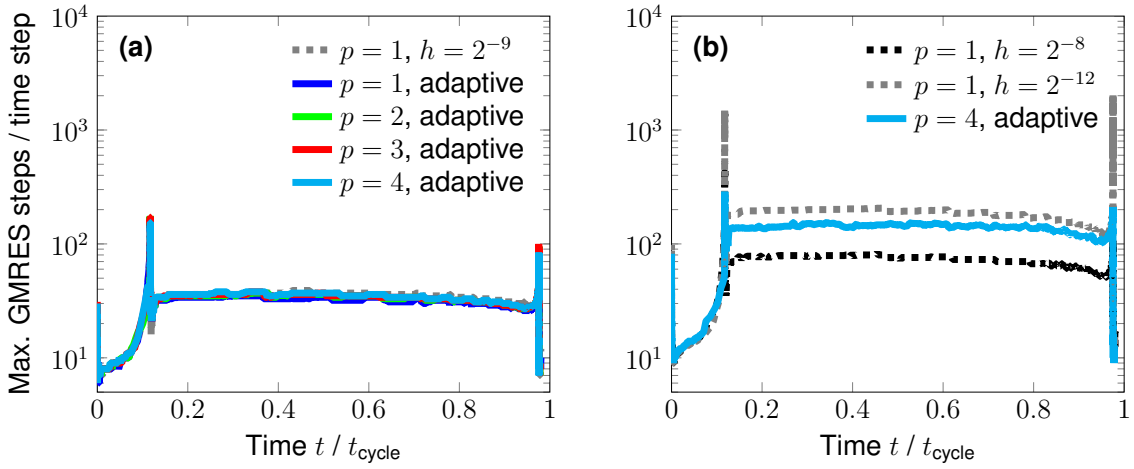

Figure 6.23.: Maximum number of GMRES steps per time step over time during the solution with different finite element degrees and adaptive mesh refinement using (a) the proposed preconditioner (4.35) and (b) the constant mobility preconditioner (4.45).

our preconditioner even for locally refined meshes. Instead, for the constant mobility preconditioner (4.45) solely the adaptive fourth-order finite element method solved the model problem until the final simulation time, with the maximum iteration number less than the saturation level. While for the adaptive solver based on second- and third-order elements the GMRES method failed to converge at the initiation of phase separation, the solution with the adaptive linear finite element method exceeded a reasonable wall-clock time.

These observations are in agreement with the numerical study of the preconditioner for the classical Cahn–Hilliard equation in Subsection 6.1.2. In particular, we note that for our proposed preconditioner (4.35) the maximum number of GMRES steps (< 200) is smaller than the chosen size of the Arnoldi basis ($= 300$) such that we avoid the restart of the

GMRES method.

6.2.2. Azimuthal Symmetric Solution Approach

The spherical symmetry assumption was previously used to reduce the computational effort for the numerical solution of the phase-field model (3.19). However, under this assumption characteristic material properties, like the strong anisotropy of LFP, cannot be captured. Therefore we assume in this subsection an azimuthal symmetric solution to reduce the computational effort, while allowing the incorporation of anisotropic material properties. Finally, we study the functionality of our parallel matrix-free finite element solver for the solution of the model equations particularly with anisotropic mobility and interfacial energy coefficient.

Note, for the following experiments we increase the interfacial energy coefficient κ by the factor ten, such that the minimal mesh width enlarges. The modification of the material parameters is a critical issue, however, in literature the value of the interfacial energy coefficient is varying, see for example [59, 150, 155]. Moreover, in this thesis we focus on the functionality of our matrix-free finite element solver, which can in future be employed for the simulation of more realistic application cases.

Azimuthal Symmetry Under the assumption of an azimuthal symmetric solution, we reduce the computational domain to the two-dimensional half circle $\Omega = \left\{\boldsymbol{x} \in \mathbb{R}^2 \,:\, |\boldsymbol{x}| \leq 1 \text{ and } \boldsymbol{x}_1 \geq 0\right\}$, see Figure B.2, where we create additionally to the curved boundary, representing the particle surface $\Gamma_{\text{ext}} = \left\{\boldsymbol{x} \in \mathbb{R}^2 \,:\, |\boldsymbol{x}| = 1 \text{ and } \boldsymbol{x}_1 > 0\right\}$, an artificial boundary in the pole axis $\Gamma_0 = \left\{\boldsymbol{x} \in \mathbb{R}^2 \,:\, |\boldsymbol{x}| \leq 1 \text{ and } \boldsymbol{x}_1 = 0\right\}$. The rotation of the computational domain around the pole axis links it to the original three-dimensional particle domain. To preserve the azimuthal symmetry we impose homogeneous Neumann boundary conditions at the artificial boundary on the pole axis

$$\nabla c \cdot \boldsymbol{n} = \boldsymbol{N} \cdot \boldsymbol{n} = 0 \quad \text{on } (0, t_{\text{f}}) \times \Gamma_0. \tag{6.17}$$

We translate the azimuthal symmetry into the discrete finite element formulation by using a modified quadrature weight as discussed in Appendix B.2.2.

Isotropic Material Properties In the first experiment under the assumption of azimuthal symmetry we consider the isotropic case and solved the model equations numerically on a uniform mesh ($n_{\text{cells}} = 4096$) with fourth-order finite elements ($N_x = 131\,842$) and the variable-step, variable-order time integration scheme within two days with 20 MPI ranks on the *Skylake* partition. We start from the constant initial concentration $\overline{c}_0 = 0.1$ and simulate the lithiation of an electrode particle until the particle is approximately filled at the final time $t_{\text{f}} = 0.89$. Further, to obtain a more structured solution, we increase the applied lithium flux to $N_{\text{ext}} = 10\,\text{C}$.

In Figure 6.24 we depict snapshots of the normalized concentration during the lithium insertion. Starting from the constant initial concentration (Figure 6.24a), the phase separation is initiated by the formation of several *lithium rich bubbles* at the particle surface (Figure 6.24b). During a fast rearrangement stage the pure phases at the equilibrium concentrations $\overline{c}_\alpha$ and $\overline{c}_\beta$ are evolved and two lithium rich bubbles at the north and the south pole remain. As further lithium enters, the phase transitions are migrating (Figure 6.24c) until

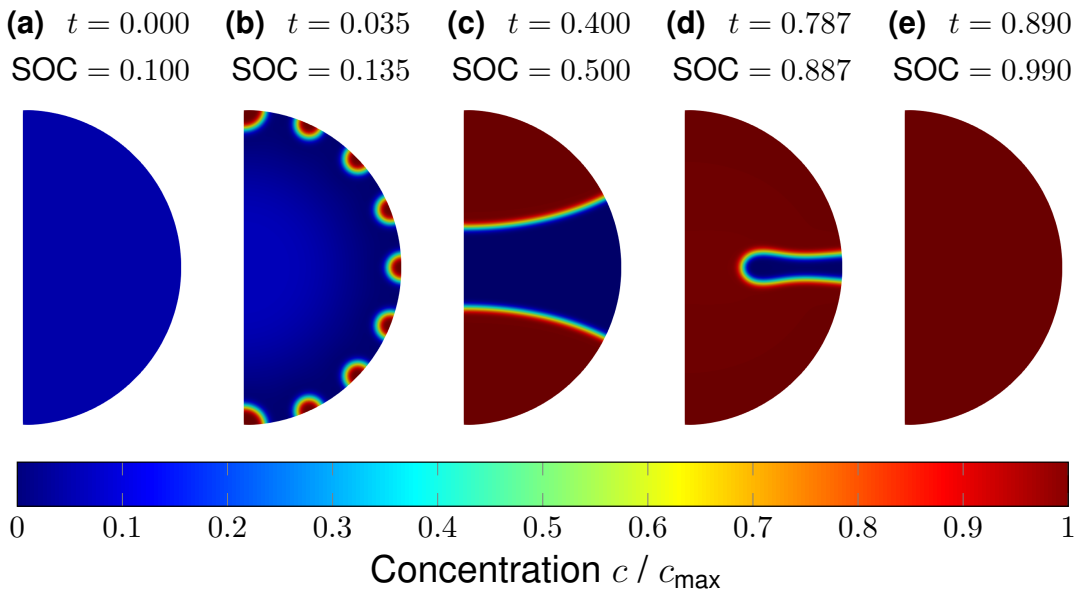

Figure 6.24.: Normalized lithium concentration in an isotropic electrode particle at characteristic time steps during lithium insertion. (a) Initial state. (b) Initiation of phase separation. (c) Half filled particle. (d) Vanishing of lithium poor phase. (e) Completely filled particle.

they meet at the equator and merge (Figure 6.24d). Finally, the lithium poor phase gets occupied by the lithium rich phase and the phase transition decays. In the end, the particle is completely filled by a homogeneous lithium rich phase (Figure 6.24e).

The occurrence of the lithium rich bubbles during phase separation is totally in contrast to expected core-shell scenario in the spherical symmetric case. However, the break up of the core-shell structure is in agreement with results reported by Zhang and Kamlah [170].

As mentioned in [173], larger lithium fluxes might recover the core-shell scenario. Instead, for lower lithium fluxes $N_{\text{ext}} < 10\,\text{C}$ we observe a random behavior during the phase separation stage. Initially also several lithium rich bubbles are formed at the surface, but then some of the bubbles absorb the others until only a single bubble remains. In particular, this behavior might be enforced by the usage of spatial adaptivity, which is the reason for us to disable this component of our adaptive solution algorithm for these simulations.

In addition to the numerical solution we plot in Figure 6.25 the time step sizes with the used orders and the maximum number of GMRES steps per time step.

In contrast to the spherical symmetric case, the time step size is varying strongly in phase separation stage to accurately capture the formation and rearrangement of the lithium rich bubbles. During the two-phase diffusion state we create large computational savings by a large time step size. In the end the adaptive time integration scheme reduces the time step size for the merging of the two bubbles and the final decay of the phase transition. For the computation of the initial steps we have used the time step size $\tau = 10^{-10}$ and solved the whole insertion process with 1420 time steps.

Further, our proposed matrix-free preconditioner (4.35) is also suited for the solution of the phase-field model equations (3.19) in the higher dimensional case. For the maximum number of GMRES steps per time step we observe in Figure 6.25b the relation to the varying time steps size and the discussed evolution of the phase transition morphology. In

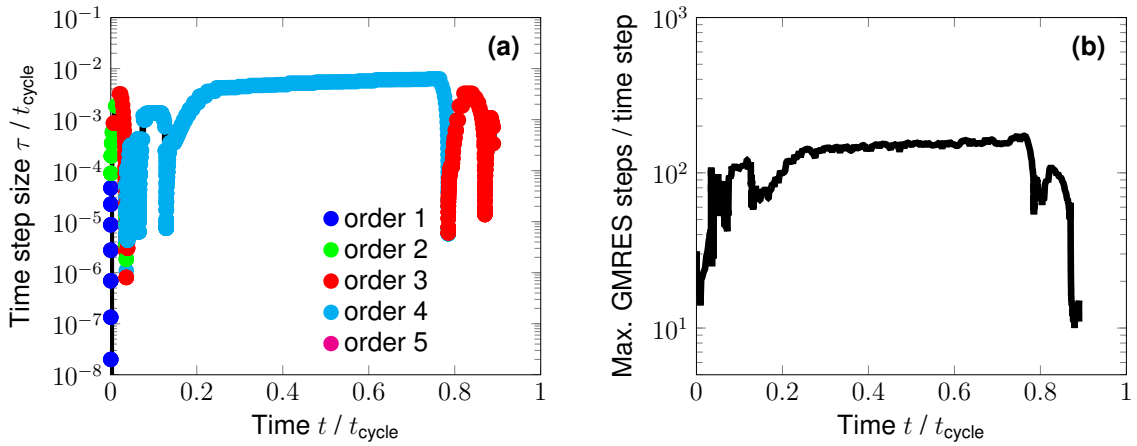

Figure 6.25.: (a) Time step size with used order and (b) maximum number of GMRES iterations per time step over time for the azimuthal symmetric, isotropic case.

particular, we note that the maximum number of iterations does not exceed the size of the Arnoldi basis, so there is no need to restart the GMRES method. However, different to the spherical symmetric case, the overall maximum number of iterations is reached during the two-phase diffusion state.

Anisotropic Material Properties Next we come the simulation of phase separation during lithium insertion into electrode particles with anisotropic material properties. The model equations, discussed in Chapter 3, include already the necessary terms, such that we only need to specify the tensor-valued mobility $\mathbf{D}$ and interfacial energy coefficient $\mathbf{K}$. Note, however, that we assume an azimuthal symmetric solution and thus cannot describe the correct anisotropy of LFP with one-dimensional tunnels. Instead, we follow the general form of the three-dimensional tensors from [134, 174] and adapt it to our azimuthal symmetry assumption. With the scalar diffusion coefficient $D > 0$ and the increased interfacial energy coefficient $\kappa > 0$ as for the previous isotropic case, we use the following two-dimensional tensors for anisotropy

$$\widehat{\mathbf{D}} = D \begin{pmatrix} 1 & 0 \\ 0 & 100 \end{pmatrix}, \qquad \widehat{\mathbf{K}} = \kappa \begin{pmatrix} 1 & 0 \\ 0 & 100 \end{pmatrix}, \tag{6.18}$$

and replace the arising integrals according to

$$\int_{\Omega} \nabla v \cdot \mathbf{K} \nabla w \, \mathrm{d}\boldsymbol{x} \quad \rightsquigarrow \quad 2\pi \int_{\Omega_c} \left(\widehat{\mathbf{K}}_{1,1} \partial_x v \partial_x w + \widehat{\mathbf{K}}_{2,2} \partial_z v \partial_z w \right) r \sin(\theta) \, \mathrm{d}\boldsymbol{x}, \tag{6.19}$$

with $v, w \colon \Omega \subset \mathbb{R}^3 \to \mathbb{R}$, compare Appendix B.2.2.

Although this numerical experiment does not incorporate the anisotropy of LFP correctly, it is a proof of concept and demonstrates the capabilities of our matrix-free finite element solver for the solution of the model equations (3.19) with anisotropic parameters.

As for the isotropic experiment we start from the constant initial concentration $\overline{c}_0 = 0.1$ and simulate the lithium insertion with the applied uniform flux $N_{\text{ext}} = 10\,\mathrm{C}$ until the time $t_{\text{f}} = 0.89$. With the same spatial discretization our solver takes one day with 20 MPI ranks on the *Skylake* partition.

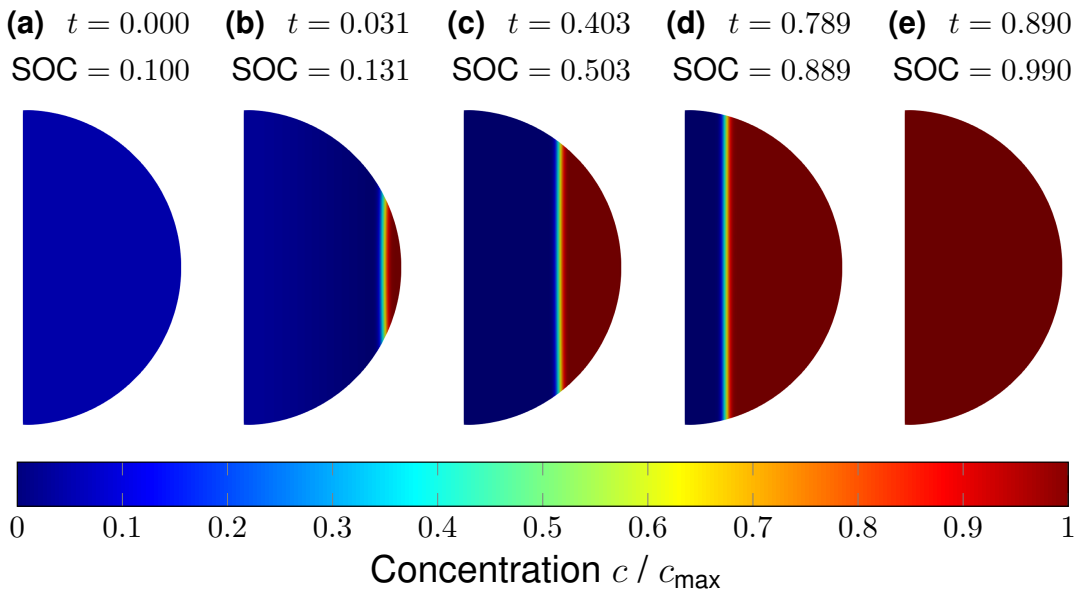

Figure 6.26.: Normalized lithium concentration in an anisotropic electrode particle at characteristic time steps during lithium insertion. (a) Initial state. (b) Initiation of phase separation. (c) Half filled particle. (d) Vanishing of lithium poor phase. (e) Completely filled particle.

The snapshots of the normalized lithium concentration, depicted in Figure 6.26, show the evolution of the phase transformation during lithium insertion. Opposed to the isotropic case, the initiation of phase separation is triggered at the particle surface in the point, where the outer normal vector points in the direction of the smallest mobility. As we assume $\widehat{\mathbf{D}}_{1,1} \ll \widehat{\mathbf{D}}_{2,2}$, the phase transition emerges as orthogonal front to the x-axis (Figure 6.26b). The further lithiation drives the pure phases to their equilibrium concentrations and push the planar phase transition front towards the center (Figure 6.26c–d) until the phase transition decays and the particle is completely filled (Figure 6.26e).

In contrast to the isotropic case, the formation and movement of the phase transition is governed by the strong anisotropy of the model parameters. The observation of an aligned phase transition, which migrates into the direction of the smallest mobility, is in qualitative agreement with the numerical experiment in Zhao et al. [174].

Furthermore, we observe that the variation of the externally applied lithium flux $N_{\text{ext}} \in \{1, 10, 100, 1000\}$ does not influence the general phase transition morphology.

Finally, we inspect the time step sizes with the used orders and the maximum number of GMRES steps per time step in Figure 6.27.

Compared to the isotropic case with the complex phase separation stage, the separation process for anisotropic parameters is simplified by the triggered evolution of the phase transition. Thus only the initiation and the decay have to be tracked with sufficiently small time step sizes, see Figure 6.27a. The simplified phase transformation is also reflected in the number of time steps. While the isotropic simulation was solved with 1420 steps, the adaptive time integration uses for the anisotropic simulation 501 steps only. Thereby both simulations used the same initial time step size $\tau = 10^{-10}$. In particular, we note the similarity of the time step size profile to the spherical symmetric case.

Additionally, in Figure 6.27b we observe the relation of the maximum iteration numbers

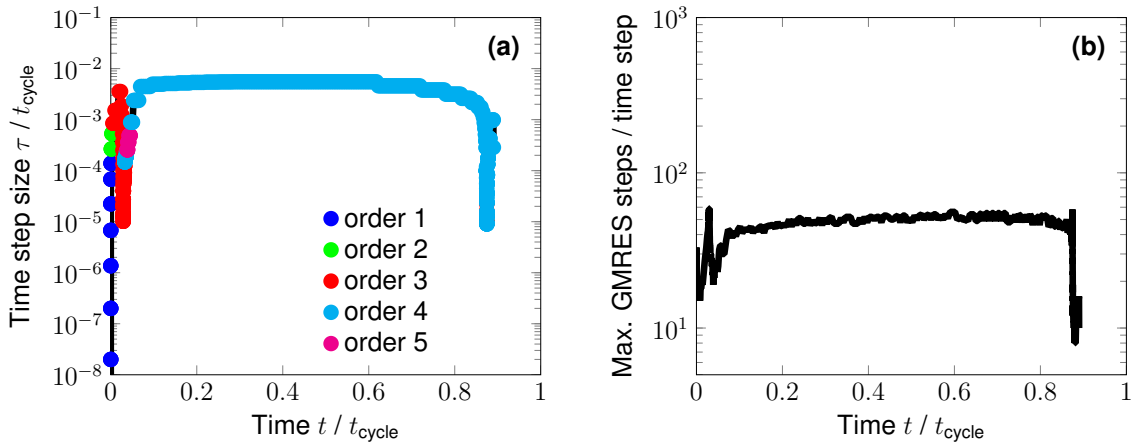

Figure 6.27.: (a) Time step size with used order and (b) maximum number of GMRES iterations per time step over time for the azimuthal symmetric, anisotropic case.

per time step to the varying time step size during the initiation and the decay of the phase transition. Further, the maximum number is even smaller compared to the isotropic case. Hence, we conclude for our proposed matrix-free preconditioner (4.35) the functionality, even with anisotropic mobility.

7. Chemical-Mechanical Coupling

The content of this chapter represents an adapted version of our article [44].

After the introduction from our article [44, Sect. 1] in Section 7.1, we present the coupled chemical-mechanical model in a form adapted to this thesis in Section 7.2. Following, in Section 7.3 we focus on the spatial discretization of the model equations and describe the differences to the previously explained adaptive solution algorithm from Section 4.3. Then in Section 7.4 we give an extended discussion of the simulation results, including the additional Figures 7.8, 7.12 and 7.13. Finally, in Section 7.5 we summarize the key findings as stated in the conclusion of our article [44, Sect. 5].

7.1. Introduction

The decarbonization of society requires efficient storage of energy for portable, mobile, and stationary applications, e. g., stabilization of power grids based on fluctuating renewable energy sources. A promising approach for this challenge are electrochemical storage systems in the form of batteries. For mobile applications, lithium-ion batteries are the state of the art technology as they comprise high energy density and good cyclability [6]. Improving the performance of lithium-ion batteries is thus key towards a green future.

A critical issue limiting the performance and lifetime of lithium-ion batteries is the coupling of chemistry and mechanics inside the battery active particles. The lithium uptake of the host material leads to volume mismatch inside the particles. This causes mechanical stress, which eventually leads to particle fracture and thereby capacity loss [111, 158]. In turn, the mechanical stress influences the diffusion of lithium inside the host [47, 131, 157,158,160]. These effects are especially pronounced for phase separating materials, e. g., lithium manganese oxide spinel $Li_xMn_2O_4$ (LMO), lithium iron phosphate Li_xFePO_4 (LFP) and group IV elements (silicon, germanium and tin), which form a lithium poor and a lithium rich phase during battery operation [56, 144, 146, 147, 149, 159, 172].

Physical models extend our understanding of the chemical-mechanical coupling and the phase separation within the electrode particles. The Cahn–Hilliard theory [37, 38] provides a framework to consistently describe diffusion and phase separation based on a free energy functional. By extending this free energy functional for mechanical contributions, a consistent description of the chemical-mechanical coupling emerges. Several groups implemented these models to describe the lithiation of silicon [46, 112, 113, 163], $Li_xMn_2O_4$ [85, 149, 167, 169], Li_xFePO_4 [48, 59, 116, 138, 150, 168–170] or the sodiation of Na_xFePO_4 [168–170].

However, phase-field models are computationally demanding, which impedes their application in simulations of full cells. The geometric nonlinearity of large deformations and the higher order differential operators impose a numerical challenge. Additionally, the quasi instantaneous formation and comparable slow migration of an almost sharp phase transition causes a strong variation of the spatial and temporal scales. Thus, the accurate

solution of the model equations requires a high spatial and temporal resolution and many large linear systems to solve.

In literature similar phase-field models coupling chemical-mechanical intercalation processes were already discussed [46,59,84,85,112,113,149,150,156,167–170,173,174]. To overcome the numerical challenge caused by the higher order differential operators, oftentimes a mixed formulation was considered, introducing a new solution variable and splitting the equations into two second-order equations [46, 59, 84, 85, 112, 113, 149, 150, 167–170]. In contrast [156, 173, 174] treated the fourth-order derivative straightforward using the isogeometric analysis. However, only marginally advanced numerical techniques like adaptivity were used to overcome the heterogeneities in spatial and temporal scale [170].

In this work, we develop a phase-field model for phase separating electrode materials coupling chemistry, mechanics and phase separation consistently with a common free energy functional. We consider a mixed formulation of the model equations and present a convergent, space and time adaptive, higher order finite element solver with a fully implicit, variable-step, variable-order time integration scheme. Due to the adaptive solution algorithm we create large computational savings, which allows us to perform numerical experiments in crucial parameter regimes efficiently.

7.2. Model Development

In this section, we derive a phase-field model to describe the chemical-mechanical coupling in phase transforming materials. First, we discuss the different deformations of the particle. Afterwards, we state a phase-field model accounting for the phase separation of the material. From this phase-field model, we consistently derive equations to describe mechanics and chemistry.

Note, in this chapter we add the mechanical coupling to the already discussed phase separation model form Chapter 3. For completeness of the model derivation we recall here some equations, which we already introduced for the phase separation model in Section 3.2.

7.2.1. Deformation

During battery operation, the particle deforms from its initial (reference) configuration Ω_0 to the current configuration Ω as shown in Figure 7.1. We describe this process with the deformation gradient $\mathbf{F} = \partial \boldsymbol{x} / \partial \boldsymbol{X}_0$, which corresponds to the mapping of a point $\boldsymbol{X}_0 \in \Omega_0$ from the reference configuration to a point $\boldsymbol{x} \in \Omega$ in the current configuration. The displacement $\boldsymbol{u} \colon \mathbb{R}_{\geq 0} \times \Omega_0 \to \mathbb{R}^d$ is incorporated in this mapping $\mathbf{F} \colon \mathbb{R}^{d,d} \to \mathbb{R}^{d,d}$ according to

$$\mathbf{F}(\nabla \boldsymbol{u}) = \mathbf{Id} + \nabla \boldsymbol{u}, \tag{7.1}$$

with the identity matrix $\mathbf{Id}$ [30, 82]. We multiplicatively split the deformation according to

$$\mathbf{F} = \mathbf{F}_{\text{el}} \mathbf{F}_{\text{ch}}, \tag{7.2}$$

into a chemical part $\mathbf{F}_{\text{ch}}$, caused by lithium uptake of the host material, and an elastic part $\mathbf{F}_{\text{el}}$, caused by mechanical stress. We assume an isotropic and linear chemical expansion

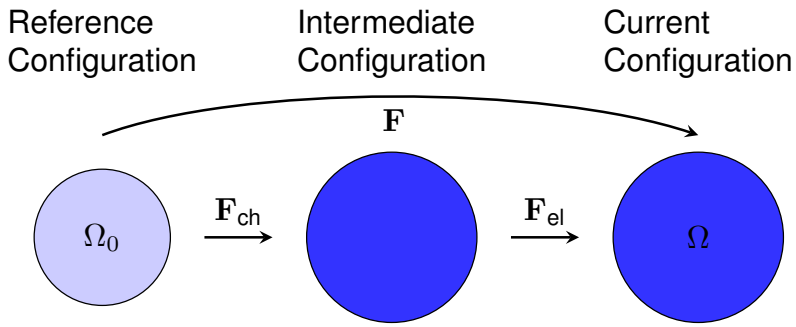

Figure 7.1.: The particle first deforms chemically $\mathbf{F}_{\text{ch}}$ then elastically $\mathbf{F}_{\text{el}}$, leading to the overall deformation $\mathbf{F}$.

of the host material with rising lithium content according to

$$\mathbf{F}_{\text{ch}}(\overline{c}) = \lambda_{\text{ch}}(\overline{c})\mathbf{Id} \qquad \text{with} \qquad \lambda_{\text{ch}}(\overline{c}) = \sqrt[3]{1 + v c_{\max} \overline{c}}\,. \tag{7.3}$$

Here, $v > 0$ denotes the *partial molar volume* of lithium inside the host material and $\overline{c} = c/c_{\max}$ the normalized lithium concentration, with the maximum lithium concentration of the host material $c_{\max}$. In the following subsections, we derive the balance equations in the nondeforming reference configuration Ω_0.

7.2.2. Phase-Field Model

Free Energy Density

We derive a thermodynamically consistent model based on a free energy density $\psi\colon [0,1] \times \mathbb{R}^d \times \mathbb{R}^{d,d} \to \mathbb{R}$, which guarantees a strictly positive entropy production [125]. The free energy density consists of three parts, the chemical $\psi_{\text{ch}}\colon [0,1] \to \mathbb{R}$ and interfacial part $\psi_{\text{int}}\colon \mathbb{R}^d \to \mathbb{R}$, already introduced in (3.2) and (3.5), and an additional part for the elastic free energy $\psi_{\text{el}}\colon [0,1] \times \mathbb{R}^{d,d} \to \mathbb{R}$ [46, 59, 149, 168]

$$\psi(z, \boldsymbol{p}, \mathbf{S}) = \psi_{\text{ch}}(z) + \psi_{\text{int}}(\boldsymbol{p}) + \psi_{\text{el}}(z, \mathbf{S}), \tag{7.4}$$

where the formal arguments $z \in [0,1]$, $\boldsymbol{p} \in \mathbb{R}^d$ and $\mathbf{S} \in \mathbb{R}^{d,d}$ represent respectively the normalized concentration $\overline{c}$, the gradient of the normalized concentration $\nabla\overline{c}$ and the *Jacobian of the displacement* $\nabla\boldsymbol{u}\colon \mathbb{R}_{\geq 0} \times \Omega_0 \to \mathbb{R}^{d,d}$.

The chemical free energy density ψ_{ch} has the form

$$\psi_{\text{ch}}(\overline{c}) = RTc_{\max}\left[\alpha_1\overline{c} + \frac{\alpha_2}{2}\overline{c}^2 + \overline{c}\log(\overline{c}) + (1-\overline{c})\log(1-\overline{c})\right], \tag{7.5}$$

equal to (3.2), depending on the normalized lithium concentration $\overline{c} = c/c_{\max}$ with the maximum lithium concentration of the host material $c_{\max}$, the universal gas constant R and the operation temperature T in Kelvin. The first two terms account for the energetic contributions from the interaction of lithium ions with the host material (α_1) and other lithium ions (α_2) [77, 84, 168]. The third and fourth term describes the entropic contributions based on a solid solution approach. These interactions lead to a double well potential as illustrated in Figure 7.2 with two stable phases α and β [84, 149, 168].

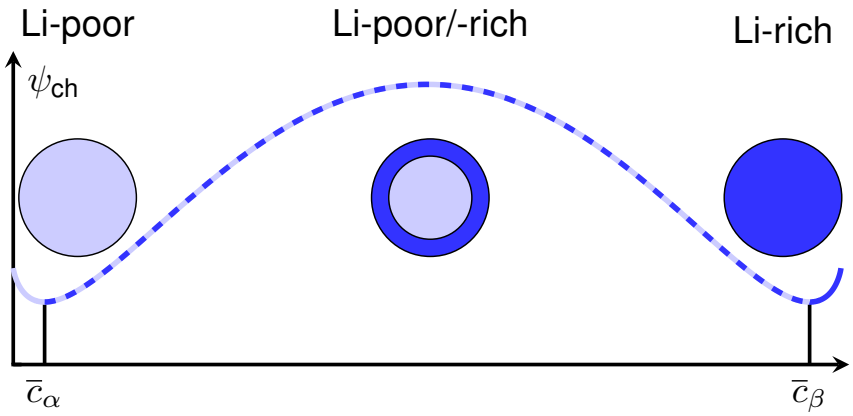

Figure 7.2.: Upon lithium uptake, the particle separates into a lithium poor α-phase and a lithium rich β-phase. The normalized lithium concentrations $\overline{c}_\alpha$ and $\overline{c}_\beta$ of the respective phase is set by the minima of the double well potential.

The interfacial energy functional

$$\psi_{\text{int}}(\nabla \overline{c}) = \frac{1}{2} RT c_{\text{max}} \kappa |\nabla \overline{c}|^2, \tag{7.6}$$

according to the isotropic version of (3.5), gives rise to a separation of these phases. We recall (3.7), that the interfacial energy coefficient κ determines the magnitude of the interfacial energy and thereby the interface thickness s [38] according to

$$s = (\overline{c}_\beta - \overline{c}_\alpha) \sqrt{\frac{\kappa}{2 \triangle \overline{\psi}_{\text{ch}}}}, \tag{7.7}$$

with the normalized equilibrium concentrations of the lithium poor $\overline{c}_\alpha$ and lithium rich phase $\overline{c}_\beta$. The excess free energy density $\triangle \overline{\psi}_{\text{ch}}$ is defined as the difference of the normalized chemical free energy density $\overline{\psi} = \psi / RT c_{\text{max}}$ at $(\overline{c}_\alpha + \overline{c}_\beta)/2$ and the normalized chemical free energy density obtained as mixture of the α and β phases [173]

$$\triangle \overline{\psi}_{\text{ch}} = \overline{\psi}_{\text{ch}} \left(\frac{\overline{c}_\alpha + \overline{c}_\beta}{2} \right) - \frac{1}{2} \big(\overline{\psi}_{\text{ch}}(\overline{c}_\alpha) + \overline{\psi}_{\text{ch}}(\overline{c}_\beta) \big). \tag{7.8}$$

We describe the elastic energy inside the host material with an elastic approach [30, 82]

$$\psi_{\text{el}}(\overline{c}, \nabla \boldsymbol{u}) = \frac{1}{2} \mathbf{E}_{\text{el}}(\overline{c}, \nabla \boldsymbol{u}) : \mathbb{C}\mathbf{E}_{\text{el}}(\overline{c}, \nabla \boldsymbol{u}) \quad \text{with} \quad \mathbb{C}\mathbf{E}_{\text{el}} = \lambda \operatorname{tr}(\mathbf{E}_{\text{el}}) \mathbf{Id} + 2G \mathbf{E}_{\text{el}}, \tag{7.9}$$

where $\lambda = 2G\nu/(1-2\nu)$ and $G = \mathrm{E}_{\mathrm{H}}/\big(2(1+\nu)\big)$ are the Lamé constants, which depend on the elastic modulus E_{H} and Poisson's ratio ν of the host material.

We define the elastic strain tensor $\mathbf{E}_{\text{el}} \colon [0,1] \times \mathbb{R}^{d,d} \to \mathbb{R}^{d,d}$ as the difference of total and stress-free chemical strain

$$\mathbf{E}_{\text{el}} = \frac{1}{2} \left(\mathbf{C} - \mathbf{C}_{\text{ch}} \right) \qquad \text{with} \qquad \mathbf{C} = \mathbf{F}^{\mathsf{T}} \mathbf{F} \qquad \text{and} \qquad \mathbf{C}_{\text{ch}} = \mathbf{F}_{\text{ch}}^{\mathsf{T}} \mathbf{F}_{\text{ch}}. \tag{7.10}$$

Based on this free energy density, we proceed to derive constitutive equations to describe the coupling of chemistry and mechanics.

Chemistry

The lithium concentration inside the host changes over time due to a gradient in the lithium flux $\boldsymbol{N}$

$$\partial_t \overline{c} = -\nabla \cdot \boldsymbol{N}. \tag{7.11}$$

Note, that we assume the lithium transport to be driven by a flux $\boldsymbol{N}$, which is defined in the reference configuration. This assumption might be justified for materials like LFP for which our simulations show a volume expansion of about six percent. However, this assumption might not be valid for a more realistic description of materials like silicon, which can undergo a volume expansion of about 300 percent, compare [46, 163]. In such a case the lithium transport has to be considered in the current configuration, which leads to a modified lithium transport equation, as for example in [163, 174].

At the particle boundary, we apply a uniform and constant external flux N_{ext}. As described in (3.16), we measure this flux in terms of the C-rate, which describes the hours it takes to fully charge the particle, $t_{\text{cycle}} = 1/\text{C-rate}$. The external flux in C is transformed to SI units by

$$|N_{\text{ext}}| \left[\mathrm{mol\,m^{-2}\,s^{-1}}\right] = \frac{1\,\mathrm{h}}{3600\,\mathrm{s}} \frac{c_{\max}}{A_{\mathrm{V}}} N_{\text{ext}}\,[\mathrm{C}] \tag{7.12}$$

with the specific surface $A_{\mathrm{V}} = S/V$ in $\mathrm{m^2\,m^{-3}}$. Using the C-rate, we obtain the *State of Charge* (SOC) of the particle as in (3.18) by

$$\text{SOC} = \frac{1}{V} \int_{\Omega_0} \overline{c}\,\mathrm{d}\boldsymbol{X}_0 = \overline{c}_0 + N_{\text{ext}}\,[\mathrm{C}] \cdot t[\mathrm{h}], \tag{7.13}$$

with the normalized initial concentration $\overline{c}_0$.

Inside the particle, a gradient in the chemical potential μ drives the lithium flux according to

$$\boldsymbol{N} = -m(\overline{c}) \nabla \mu, \tag{7.14}$$

with the isotropic mobility $m\colon [0,1] \to \mathbb{R}$ of the lithium atoms inside the host material. Following [59], we define the mobility as

$$m(\overline{c}) = \frac{D c_{\max}}{RT} \overline{c}(1-\overline{c}), \tag{7.15}$$

with the diffusion coefficient $D > 0$ of lithium atoms inside the host material. This isotropic and scalar version of the mobility is also covered by the more general expression (3.12).

We determine the chemical potential as variational derivative of the system's free energy $\Psi = \int_{\Omega_0} \psi\,\mathrm{d}\boldsymbol{X}_0$ with respect to the lithium concentration c [79]

$$\begin{aligned} \mu &= \partial_z \psi\big(\overline{c}, \nabla\overline{c}, \nabla\boldsymbol{u}\big) - \nabla \cdot \partial_{\boldsymbol{p}} \psi\big(\overline{c}, \nabla\overline{c}, \nabla\boldsymbol{u}\big) \\ &= \partial_z \psi_{\text{ch}}(\overline{c}) + \partial_z \psi_{\text{el}}\big(\overline{c}, \nabla\boldsymbol{u}\big) - \nabla \cdot \partial_{\boldsymbol{p}} \psi_{\text{int}}\big(\nabla\overline{c}\big) \\ &= RT \left[\alpha_1 + \alpha_2 \overline{c} + \log\left(\frac{\overline{c}}{1-\overline{c}}\right) - \kappa \Delta \overline{c}\right] - \frac{v}{3\lambda_{\text{ch}}} \operatorname{tr}(\mathbb{C}\mathbf{E}_{\text{el}}), \end{aligned} \tag{7.16}$$

where the first part coincides with the isotropic version of the chemical potential (3.13) for the phase-field model without elasticity.

Mechanics

Mechanical stresses arise inside the host material due to large deformations during lithiation. We describe the mechanics with the momentum balance in the reference configuration Ω_0

$$\nabla \cdot \mathbf{P}(\bar{c}, \nabla \boldsymbol{u}) = 0, \tag{7.17}$$

which neglects body and inertial forces. Here $\mathbf{P} = \det(\mathbf{F})\sigma\mathbf{F}^{-\mathsf{T}}$ denotes the *first Piola–Kirchhoff stress tensor*, which is the transformation of the *Cauchy stress* $\sigma\colon \mathbb{R}_{\geq 0} \times \Omega_0 \to \mathbb{R}^{d,d}$ to the reference configuration. We determine the first Piola–Kirchhoff stress tensor $\mathbf{P}\colon [0,1] \times \mathbb{R}^{d,d} \to \mathbb{R}^{d,d}$ thermodynamically consistent as the partial derivative of the free energy with respect to the deformation gradient $\mathbf{F}$ [82]

$$\mathbf{P} = \partial_{\mathbf{F}}\psi = \mathbf{F}\mathbb{C}\mathbf{E}_{\text{el}}. \tag{7.18}$$

7.3. Numerical Treatment

In this section we first state the mathematical problem by summarizing all normalized equations of the phase-field model in an initial boundary value problem. Afterwards, we describe the discretization of the coupled chemical-mechanical model equations from Section 7.2 with finite elements. Finally, we point out the modifications of our adaptive solution algorithm from Section 4.3.

7.3.1. Mathematical Problem Formulation

For the statement of the mathematical problem we assume in the following the dimensionless form of all model equations from Section 7.2. Thereby we refer to Appendix A.2 for the precise normalization of all quantities.

The phase-field model consists of the mixed formulation of the Cahn–Hilliard equation, which is coupled to the momentum balance equation (7.17) for the first Piola–Kirchhoff stress tensor through the free energy density (7.4). To simplify the model equations, we express the stress and strain tensors $\mathbf{P}$, $\mathbf{E}_{\text{el}}$ as well as the deformation gradient $\mathbf{F}$ in terms of the displacement $\boldsymbol{u}$. We thus rely on the normalized concentration c, the chemical potential μ and the displacement $\boldsymbol{u}$ as variables and recover the stress in a postprocessing step.

The resulting dimensionless initial boundary value problem then reads as:

Problem 4. Let $t_{\text{f}} > 0$ be the final simulation time and $\Omega_0 \subset \mathbb{R}^3$ be the bounded domain representing the reference configuration of an electrode particle. Find the normalized lithium concentration $c\colon [0, t_{\text{f}}] \times \overline{\Omega}_0 \to [0,1]$, the chemical potential $\mu\colon [0, t_{\text{f}}] \times \overline{\Omega}_0 \to \mathbb{R}$

and the displacement $\boldsymbol{u}\colon [0,t_\mathrm{f}]\times\overline{\Omega}_0\to\mathbb{R}^3$, up to rigid body motions, satisfying

$$\begin{cases} \partial_t c = \nabla\cdot\big(m(c)\nabla\mu\big) & \text{in } (0,t_\mathrm{f})\times\Omega_0,\\ \mu = \partial_z\psi_\mathrm{ch}(c)+\partial_z\psi_\mathrm{el}(c,\nabla\boldsymbol{u})-\kappa\Delta c & \text{in } (0,t_\mathrm{f})\times\Omega_0,\\ \nabla\cdot\mathbf{P}(c,\nabla\boldsymbol{u})=0 & \text{in } (0,t_\mathrm{f})\times\Omega_0,\\ \nabla c\cdot\boldsymbol{n}=0 & \text{on } (0,t_\mathrm{f})\times\partial\Omega_0,\\ m(c)\nabla\mu\cdot\boldsymbol{n}=-N_\mathrm{ext} & \text{on } (0,t_\mathrm{f})\times\partial\Omega_0,\\ \mathbf{P}(c,\nabla\boldsymbol{u})\cdot\boldsymbol{n}=0 & \text{on } (0,t_\mathrm{f})\times\partial\Omega_0,\\ c(0,\,\cdot\,)=c_0 & \text{in } \Omega_0, \end{cases} \tag{7.19}$$

where we assume an initial condition c_0 consistent with the boundary conditions.

7.3.2. Numerical Solution Method

We solve the initial boundary value problem (7.19) with a straightforwardly extended version of our previously discussed discretization scheme, which we presented in Chapter 4 of this thesis. Thus, we discuss here only the differences to the numerical solution algorithm from Chapter 4.

Space and Time Discretization

To derive the spatial discrete formulation, we first multiply the partial differential equations of (7.19) with test functions and integrate over the reference domain Ω_0. Integration by parts yields the weak formulation with the solutions c, $\mu\in V:=H^1(\Omega_0)$ and $\boldsymbol{u}\in V^*:=H^1_*(\Omega_0;\mathbb{R}^3)$. The space V^* incorporates appropriate displacement constraints, which will be specified later for the precise application case in Subsection 7.4.1.

For the application of the isoparametric Lagrangian finite element method [30] we replace the domain Ω_0 of the reference configuration by a computational domain Ω_h, which we assume to be a polytop with appropriate boundary approximation of curved boundaries. Next, we choose an admissible mesh $\mathcal{T}_h$ of the computational domain Ω_h and assume $V_h\subset V$ and $V_h^*\subset V^*$ to be finite dimensional Lagrangian finite element spaces with bases $\big\{\varphi_i\in V_h\,:\,i=1,\dots,N_x\big\}$ and $\big\{\boldsymbol{\xi}_i\in V_h^*\,:\,i=1,\dots,3N_x\big\}$. Now for $t\in[0,t_\mathrm{f}]$ we seek $c_h(t,\,\cdot\,)\in V_h\cap\big\{c_h\in[0,1]\big\}$, $\mu_h(t,\,\cdot\,)\in V_h$ and $\boldsymbol{u}_h(t,\,\cdot\,)\in V_h^*$ to be solutions of the spatial discrete system

$$\begin{cases} \big(\varphi_i,\partial_t c_h\big) = -\big(m(c_h)\nabla\varphi_i,\nabla\mu_h\big)-\big(\varphi_i,N_\mathrm{ext}\big)_{\partial\Omega_0},\\ 0 = -\big(\varphi_j,\mu_h\big)+\big(\varphi_j,\partial_z\psi_\mathrm{ch}(c_h)+\partial_z\psi_\mathrm{el}(c_h,\nabla\boldsymbol{u}_h)\big)+\kappa\big(\nabla\varphi_j,\nabla c_h\big),\\ 0 = -\big(\nabla\boldsymbol{\xi}_k,\mathbf{P}(c_h,\nabla\boldsymbol{u}_h)\big), \end{cases} \tag{7.20}$$

for all $i,j=1,\dots,N_x$ and $k=1,\dots,3N_x$. We insert the finite element ansatz, namely the basis representation of the solution variables

$$c_h(t,x)=\sum_{i=1}^{N_x}c_i(t)\varphi_i(x),\quad \mu_h(t,x)=\sum_{i=1}^{N_x}\mu_i(t)\varphi_i(x),\quad \boldsymbol{u}_h(t,x)=\sum_{i=1}^{3N_x}u_i(t)\boldsymbol{\xi}_i(x), \tag{7.21}$$

and gather the time-dependent coefficients in a common vector-valued function

$$\boldsymbol{y}\colon [0,t_\mathrm{f}] \to \mathbb{R}^{(2+3)N_x},\ t \mapsto \boldsymbol{y}(t) = \left[(c_i(t))_i, (\mu_i(t))_i, (u_i(t))_i\right]^\mathsf{T}, \tag{7.22}$$

such that we can reformulate the spatially discrete problem (7.20) as general nonlinear differential algebraic equation (DAE): Find $\boldsymbol{y}\colon [0,t_\mathrm{f}] \to \mathbb{R}^{5N_x}$ satisfying

$$\mathbf{M}\partial_t \boldsymbol{y} = \boldsymbol{f}(t,\boldsymbol{y}) \quad \text{for } t \in (0,t_\mathrm{f}], \qquad \boldsymbol{y}(0) = \boldsymbol{y}^0. \tag{7.23}$$

The mass matrix $\mathbf{M}$ of the DAE is singular with its only nonzero block entry $\mathbf{M}_{11} = \left[(\varphi_i,\varphi_j)\right]_{ij}$ denoting the mass matrix of the finite element space V_h. The right hand side is defined according to the weak formulation (7.20): With $\boldsymbol{y}$ related to c_h, μ_h, $\boldsymbol{u}_h$ as described, we have $\boldsymbol{f}\colon [0,t_\mathrm{f}] \times \mathbb{R}^{5N_x} \to \mathbb{R}^{5N_x}$

$$\boldsymbol{f}(t,\boldsymbol{y}) \coloneqq \begin{pmatrix} -K_m(c_h)\mu_h - \boldsymbol{N}_\mathrm{ext} \\ -M\mu_h + \boldsymbol{\Psi}_\mathrm{ch}(c_h) + \boldsymbol{\Psi}_\mathrm{el}(c_h,\nabla\boldsymbol{u}_h) + \kappa K_1 c_h \\ -\boldsymbol{\Sigma}(c_h,\nabla\boldsymbol{u}_h) \end{pmatrix}, \tag{7.24}$$

with the matrices

$$M = \left[(\varphi_i,\varphi_j)\right]_{ij}, \quad K_1 = \left[\left(\nabla\varphi_i,\nabla\varphi_j\right)\right]_{ij}, \quad K_m(c_h) = \left[\left(m(c_h)\nabla\varphi_i,\nabla\varphi_j\right)\right]_{ij}, \tag{7.25}$$

and the vectors for the nonlinearities and the boundary condition

$$\boldsymbol{\Psi}_\mathrm{ch}(c_h) = \left[\left(\varphi_i,\partial_z\psi_\mathrm{ch}(c_h)\right)\right]_i, \qquad \boldsymbol{\Psi}_\mathrm{el}(c_h,\nabla\boldsymbol{u}_h) = \left[\left(\varphi_i,\partial_z\psi_\mathrm{el}(c_h,\nabla\boldsymbol{u}_h)\right)\right]_i, \tag{7.26}$$

$$\boldsymbol{\Sigma}(c_h,\nabla\boldsymbol{u}_h) = \left[\left(\nabla\boldsymbol{\xi}_i,\mathbf{P}(c_h,\nabla\boldsymbol{u}_h)\right)\right]_i, \qquad \boldsymbol{N}_\mathrm{ext} = \left[\left(\varphi_i,N_\mathrm{ext}\right)_{\partial\Omega_0}\right]_i. \tag{7.27}$$

The temporal discretization of the resulting DAE (7.23) is then performed with the NDF(k) methods, as discussed in Subsection 4.1.2.

Space and Time Adaptive Algorithm

For the numerical solution of the chemical-mechanical problem (7.19) we employ an adaptive solution algorithm, which is essentially identical to the Algorithm 1 discussed in Section 4.3, however, it differs in two steps.

Firstly, in Line 2, we solve the fully discrete problem for the discrete solution $\boldsymbol{y}^{n+1}$. For the solution of the nonlinear algebraic system we employ the Newton–Raphson method, but we solve the linearized systems with the LU-decomposition instead of a preconditioned GMRES method. We replace the iterative solver by the direct solver, because we have no suitable preconditioner for the coupled problem. Consequently, to use the LU-decomposition we have to rely on the matrix-based implementation. However, the recent application of the *deal.II* matrix-free framework to solid mechanics problems, for example in [53], suggests to develop a matrix-free solver for the coupled chemical-mechanical problem in future.

Secondly, in Line 3, we measure the spatial regularity using a gradient recovery estimator. As we solve the coupled problem (7.19) additionally for the displacement, we add the local estimates for the displacement to the global estimate

$$\mathrm{est}_x \coloneqq \left(\sum_{Q\in\mathcal{T}_h} \eta_Q^2\right)^{1/2} \qquad \text{with} \qquad \eta_Q^2 \coloneqq \eta_Q^2(c_h) + \eta_Q^2(\mu_h) + \eta_Q^2(\boldsymbol{u}_h). \tag{7.28}$$

7.4. Numerical Experiments

In the following section we present and discuss the results of our numerical experiments obtained with the adaptive finite element solver from Section 7.3 for the model developed in Section 7.2. Therefore we first specify the simulation setup in Subsection 7.4.1. The discussion of the results is then split into a physical analysis in Subsection 7.4.2 and a numerical analysis of the efficiency in Subsection 7.4.3.

7.4.1. Simulation Setup

Throughout the numerical experiments we apply the so far generally derived formalism to the lithiation of a LFP electrode particle [48,59,116,138,150,168–170] with the normalized model parameters given in Table A.3.

As an exemplary particle geometry we consider a spherical electrode particle and assume a spherical symmetric solution. Thus the three-dimensional geometry reduces to the radial direction. A final two-dimensional example will demonstrate the capabilities and superior efficiency of our numerical solution method.

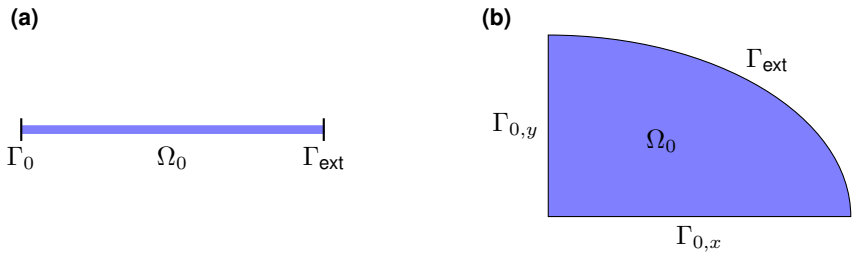

Figure 7.3.: (a) One-dimensional radial particle domain and (b) two-dimensional quarter domain of the cross-section of a spheroidal particle.

In the following we define the additional boundary conditions for both geometries based on symmetry arguments and briefly discuss some further implementation details.

Spherical Symmetry The computational domain becomes the one-dimensional unit interval $\Omega_0 = (0, 1)$, representing a radial line from the particle center $\Gamma_0 = \{0\}$ to the surface $\Gamma_{\text{ext}} = \{1\}$, see Figure 7.3a. At the artificial boundary in the particle center we assume a natural boundary condition for the concentration, a no-flux condition for the lithium flux and fix the radial displacement:

$$\nabla c \cdot \boldsymbol{n} = 0, \qquad \boldsymbol{N} \cdot \boldsymbol{n} = 0, \qquad \boldsymbol{u} = \boldsymbol{0} \qquad \text{on } (0, t_{\text{f}}) \times \Gamma_0. \tag{7.29}$$

We translate the spherical symmetry into the discrete finite element formulation by using the modified quadrature weight $\mathrm{d}\boldsymbol{X}_0 = 4\pi r^2 \,\mathrm{d}r$, compare Appendix B.2.1. Assuming spherical symmetry and a constant initial concentration c_0, an appropriate starting value for the Newton–Raphson method at the first time step is given by c_0, the constant chemical potential $\mu_0 = \partial_z \psi_{\text{ch}}(c_0)$ and the stress-free radial displacement $\boldsymbol{u}_0(r) = r\big(\lambda_{\text{ch}}(c_0) - 1\big)$.

Spheroidal Electrode Particle We consider the quarter of the cross-section of a spheroidal particle (quarter of an ellipse) with a fraction of the semi-axes $1 : 0.6$, see Figure 7.3b, inspired by Di Leo et al. [59]. We generate two flat boundaries $\Gamma_{0,x}$ and $\Gamma_{0,y}$ at the semi-axes and one curved boundary Γ_{ext} representing the particle surface, which we approximate with an isoparametric mapping. On the semi-axes we assume natural boundary conditions for the concentration and a vanishing lithium flux. The displacement on the semi-axes is only allowed in the radial direction. Summing up we have the following artificial boundary conditions

$$\begin{aligned} \nabla c \cdot \boldsymbol{n} &= 0 \quad \text{on } (0, t_{\text{f}}) \times \Gamma_0, & \boldsymbol{u}_y &= 0 \quad \text{on } (0, t_{\text{f}}) \times \Gamma_{0,x}, \\ \boldsymbol{N} \cdot \boldsymbol{n} &= 0, \quad \text{on } (0, t_{\text{f}}) \times \Gamma_0, & \boldsymbol{u}_x &= 0 \quad \text{on } (0, t_{\text{f}}) \times \Gamma_{0,y}. \end{aligned} \tag{7.30}$$

For this geometry we initialize the Newton–Raphson method at the first time step with the given constant initial concentration c_0 along with $\mu_0 = 0$ and $\boldsymbol{u}_0 = \boldsymbol{0}$.

Mesh Width For the accurate resolution of the phase transition we distribute at least ten degrees of freedom (DOFs) in the interfacial zone. Following Equation (7.7) we determine the dimensionless interface thickness $s \approx 0.02$ for the normalized LFP parameters from Table A.3. Thus we obtain the estimate $h < 0.02\, p/10$ for the minimal mesh width, knowing that for a p-th order Lagrangian finite element method the approximate distance of two DOFs is about h/p.

Solver Parameters We employ fourth-order isoparametric Lagrangian finite elements for all numerical experiments. In addition we use for the adaptive space and time algorithm the tolerances $\text{RelTol}_t = \text{RelTol}_x = 10^{-5}$, $\text{AbsTol}_t = \text{AbsTol}_x = 10^{-8}$ and the marking parameters for local coarsening $\theta_{\text{c}} = 0.05$ and refinement $\theta_{\text{r}} = 0.5$. Solely for the simulation to Figure 7.5 we deviate and use a uniform mesh ($h = 2^{-7}$), because the strongly varying C-rates would require different tolerances for spatial adaptivity.

Implementation We implement the numerical methods described in Section 7.3 as C++ code based on the functionalities of the finite element library *deal.II* [13] and the LU-decomposition of [52]. We execute the simulations on a laptop pc with $4\,\text{GB}$ RAM and an Intel i5-4200U CPU with $1.6\,\text{GHz}$. For the two-dimensional simulation, we instead use a node of a cluster with $96\,\text{GB}$ RAM and 20 Intel Xeon Silver 4114 CPUs with $2.2\,\text{GHz}$. To speedup the computational time, we enable shared memory parallelization for the assembly of the system matrices, residuals and spatial estimates.

7.4.2. Chemomechanics

In this subsection, we analyze the relationship between phase separation and mechanical stress. We start by investigating the stress evolution during lithiation. Afterwards, we look at the development of the maximum stress inside the particle for different charging rates. We then analyze the influence of the interface properties on the mechanical stress and conclude with the investigation of the chemical potential during charging and discharging.

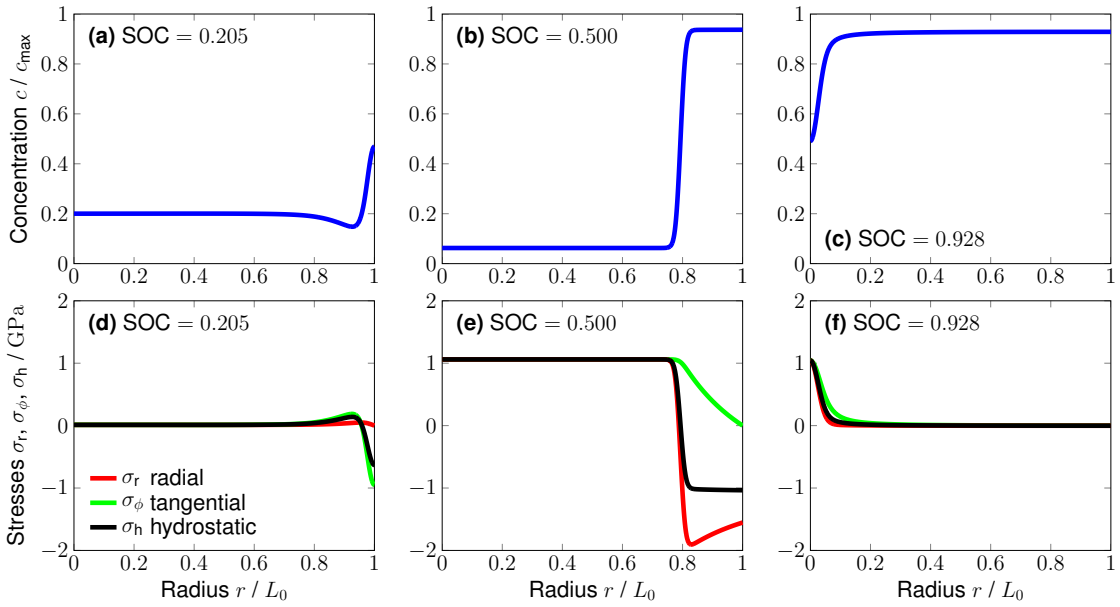

Figure 7.4.: Concentration (top) and radial σ_r, tangential σ_ϕ, and hydrostatic stresses σ_h (bottom) inside the particle at three characterizing SOCs for lithium insertion. (a/d) Initiation of phase separation. (b/e) Migration of the phase transition. (c/f) Vanishing of the phase transition.

Lithium Insertion

In Figure 7.4, we see the normalized concentration c and the radial σ_r, tangential σ_ϕ as well as hydrostatic stress $\sigma_h = 1/3\,\sigma_r + 2/3\,\sigma_\phi$ inside the particle during lithium insertion with $N_{ext} = 1\,\mathrm{C}$. We observe three different states throughout lithiation. First, for SOC $\lesssim 0.2$, the particle exhibits only a single phase and the stress magnitude is low. Second, for $0.2 \lesssim$ SOC $\lesssim 0.9$, the particle separates into a lithium poor phase and a lithium rich phase. Figure 7.4a shows the emergence of both phases, which develop to the concentration distribution shown in Figure 7.4b. Along the phase boundary, a sharp transition of the concentration and the hydrostatic stress from tensile (lithium poor phase) to compressive (lithium rich phase) occurs. Third, if SOC $\gtrsim 0.9$, the lithiation of the lithium poor phase sets in and the phase boundary vanishes, as shown in Figure 7.4c. The stress in the lithium poor part remains tensile whereas the stress in the lithium rich part vanishes.

We observe a tensile stress in the lithium rich phase and a compressive stress in the lithium poor phase, because the different degrees of lithiation cause a volume mismatch inside the particle. The elastic deformation compensates this trend by compressing the lithium rich phase and expanding the lithium poor phase. This leads to a sharp drop of the tangential stress σ_ϕ from tensile to compressive at the interface. The radial stress σ_r, in contrast, shows a continuous transition across the interface to fulfill the mechanical equilibrium.

Comparing our results for the mechanical stress with literature, we see that the radial and the tangential stresses are in excellent qualitative agreement with Walk et al. [149]. Likewise, the hydrostatic stress shows the same qualitative behavior as in Zhang and Kamlah [168–170].

Maximum Stress

Variation of the C-rate The maximum arising stress is of interest to analyze plasticity and fracture inside the particle. In Figure 7.5 we thus present the maximum stress for different SOC and charging rates. Figure 7.5a shows the absolute value of the maximum

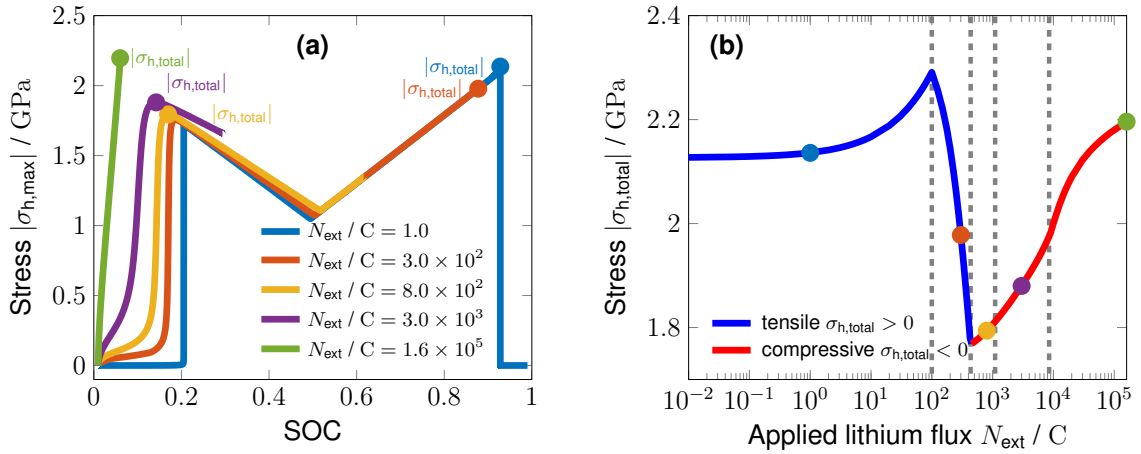

Figure 7.5.: (a) Maximal hydrostatic stress over SOC for different lithium insertion rates and (b) maximal hydrostatic stress in the whole particle over applied C-rate coded with the same colors in both figures.

hydrostatic stress $|\sigma_{h,max}|$ inside the particle during charging with various C-rates. Again, we identify the three distinct states (SOC $\lesssim 0.2$, $0.2 \lesssim$ SOC $\lesssim 0.9$, $0.9 \lesssim$ SOC). For low C-rates, the stress vanishes in the first and third state, which correspond to a purely lithium poor, respective rich phase. In the second state, however, a large hydrostatic stress emerges with two maxima at the single phase separation (SOC ≈ 0.2), respective formation (SOC ≈ 0.9), and a minimum in between. For higher C-rates, the end point of the simulation is shifted towards lower SOC. Moreover, the first phase (SOC $\lesssim 0.2$) is less homogeneous and exhibits high stresses. In Figure 7.5b, we plot the maximum stress over the applied C-rate. We see, that the stress approaches a constant value of $|\sigma_{h,total}| \approx 2.13\,\mathrm{GPa}$ for quasi-static charging, i. e., a vanishing C-rate. This stress is largely constant for low C-rates, but increases to a maximum of $|\sigma_{h,total}| \approx 2.3\,\mathrm{GPa}$ at $N_{ext} = 100\,\mathrm{C}$. For higher C-rates, we first see a decrease to $|\sigma_{h,total}| \approx 1.7\,\mathrm{GPa}$ at $N_{ext} = 500\,\mathrm{C}$, followed by a continuous increase with rising N_{ext}.

As already discussed in Figure 7.4, inhomogeneous lithiation causes volume mismatches and thereby mechanical stress. For single phases, the concentration is distributed evenly for C-rates up to 100 C, because Fo $\gg 1$. When the C-rate increases further, the lithium concentration becomes less homogeneous and thus gives rise to a hydrostatic stress, which increases with the C-rate. Upon the initial phase separation (SOC ≈ 0.2), a large compressive stress arises in the outer lithium rich phase. This stress decreases with increasing SOC, because the size of the lithium rich phase approaches that of the lithium poor phase. Thereby, each element of the lithium poor phase is stretched further, while each element of the lithium rich phase is compressed less. At SOC ≈ 0.5, compressive and tensile stress are equally large, leading to the observed minimum in Figure 7.5a. For higher SOC, the tensile stress in the particle center is the maximum absolute stress. This stress further increases with decreasing size of the lithium poor phase, until the phase ultimately vanishes

at SOC ≈ 0.9.

The charging rate affects the maximum stress only weakly below $N_{\text{ext}} = 1\,\text{C}$ (Figure 7.5b), because the phases are homogeneously lithiated due to Fo $\gg 1$. The sharp decrease in maximum stress for $100\,\text{C} < N_{\text{ext}} < 500\,\text{C}$ comes from the premature end of the simulation. Thereby, the stress no longer reaches its maximum at SOC $= 0.9$ and thus decreases until the compressive stress at the phase separation, SOC ≈ 0.2, is the maximum stress. Higher charging currents $N_{\text{ext}} > 500\,\text{C}$ cause larger concentration gradients in the lithium poor phase (SOC < 0.2) causing a high hydrostatic stress in this regime.

Our results for the maximum absolute stress over SOC are in good qualitative accordance with the results of Zhang and Kamlah [168]. Likewise, the dependence of the maximum absolute stress on the charging rate accords to the work of Kamlah and co-workers [85, 149]. However, we see that our solver exits prematurely above $100\,\text{C}$, as soon as the concentration reaches its maximum at the particle surface. Although $100\,\text{C}$ are well above typically realized C-rates, we could resolve this issue by employing Butler–Volmer boundary conditions instead of a constant charging rate [59, 162].

Variation of the interfacial energy coefficient Next, we analyze how the interfacial energy parameter κ influences the hydrostatic stress at the phase boundary. In Figure 7.6a

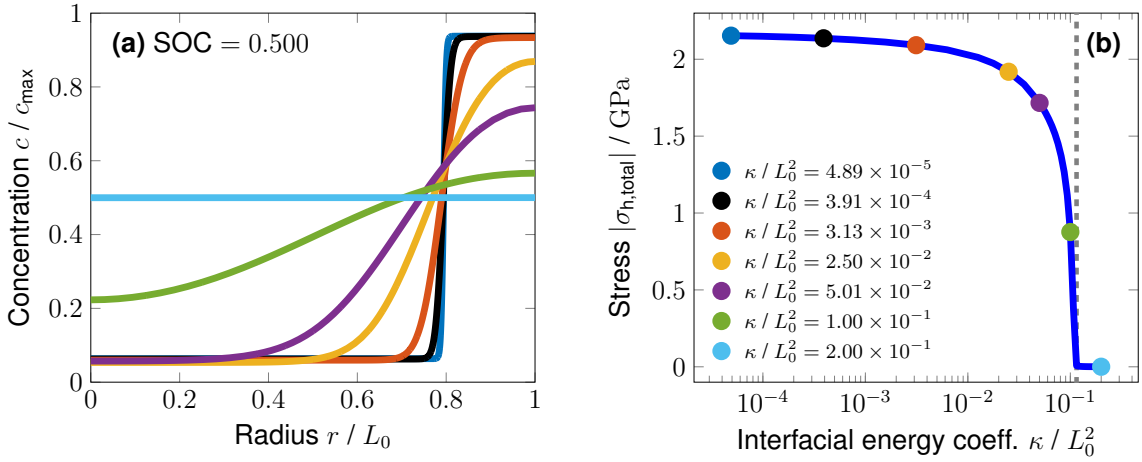

Figure 7.6.: (a) Phase transition and (b) maximal arising hydrostatic stresses for varying phase-field parameter κ coded with the same colors in both figures.

we see the dependence of the concentration profile on the interfacial energy quantified by the parameter κ for a SOC of 0.5. We observe that the phase separation becomes more diffuse with increasing κ up to a completely homogeneous concentration at $\kappa/L_0^2 = 2 \times 10^{-1}$. In Figure 7.6b we plot the maximum resulting stress as a function of κ. We see that the arising stress is constant for several orders of magnitude, $1 \times 10^{-5} < \kappa/L_0^2 < 5 \times 10^{-2}$. For larger κ, the maximum stress sharply drops until it reaches approximately zero for $\kappa/L_0^2 = 0.115$.

The interface thickness depends on the interfacial energy coefficient according to $s \sim \sqrt{\kappa}$, see Equation (7.7). The total stress $|\sigma_{\text{h,total}}|$ decreases with increasing interface thickness s, because the volume mismatches at more diffuse interfaces are weaker. For $\kappa/L_0^2 = 0.115$, the phase separation is completely suppressed, because the energy penalty $\sim \kappa|\nabla \bar{c}|^2$ is higher than the energy gain of the separation into two phases. In this regime, the free

energy follows the profile shown in Figure 7.2 without undergoing the spinodal decomposition [85, 167].

Our concentration profiles are in good accordance with the results of Kamlah and coworkers [85, 167] and show that the interface parameter κ is critical to determine the stress inside the particle. However, the values for κ vary throughout literature as shown in Table 7.1.

Three-dimensional particle simulations can so far only be performed if the setting allows less computational expenses. Thus, larger values for κ are used to increase the interface thickness and thereby the mesh width of the underlying mesh [59, 150, 173, 174]. Based on Figure 7.6b, we argue that this approach underestimates the resulting stress at the interface.

Table 7.1.: Interfacial energy coefficients for different materials.

Material	**Source**	**Coeff.** κ / m^2	**Particle scale** L_0 / m	**Interface thickness** s / L_0
LMO	[149]	7.0×10^{-18}	1×10^{-6}	6.4×10^{-3}
LMO	[173]	4.2×10^{-14}	1×10^{-6}	5.4×10^{-1}
LFP	[168]	8.8×10^{-18}	150×10^{-9}	2.0×10^{-2}
LFP	[162]	8.8×10^{-18}	100×10^{-9}	3.1×10^{-2}
LFP	[59]	2.5×10^{-16}	500×10^{-9}	5.7×10^{-2}
LFP	[150]	2.7×10^{-17}	40×10^{-9}	1.0×10^{-1}

Hysteresis

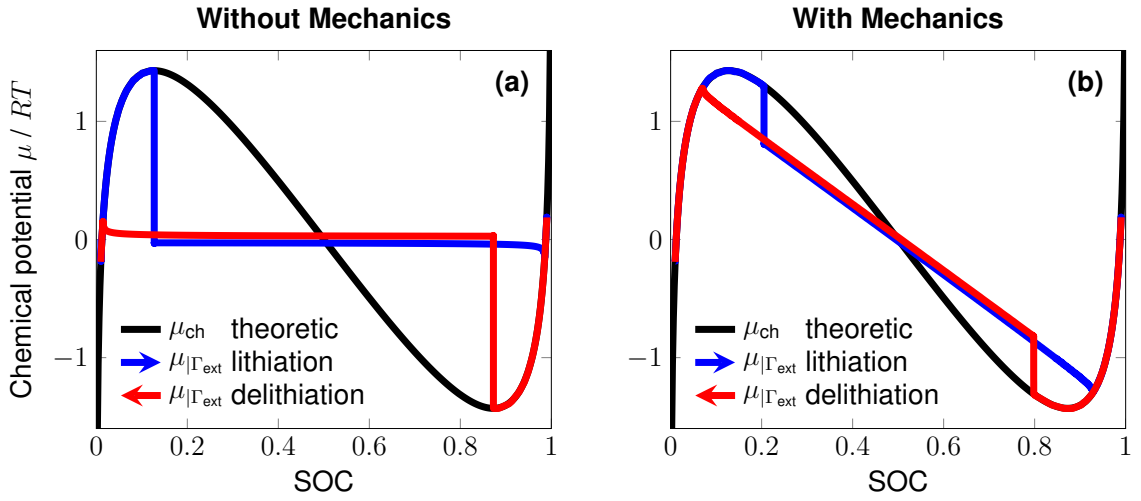

Figure 7.7.: Hysteresis showing different chemical potentials at the particle surface ($r = L_0$) during lithiation and delithiation versus the state of charge. (a) Without mechanics and (b) with mechanics.

Now, we investigate how mechanics influences the phase separation. In Figure 7.7, we illustrate the chemical part of the chemical potential, $\mu_{\mathrm{ch}} = \alpha_1 + \alpha_2 \bar{c} + \log\big(\bar{c}/(1-\bar{c})\big)$ in black and the simulated chemical potential during lithiation and delithiation in red and blue, respectively. In Figure 7.7a, we plot the hysteresis of the chemical potential without the

mechanical effects. During lithiation, the chemical potential follows the black curve at low states of charge and then sharply drops to a negative value at the maximum of the chemical potential. During delithiation, the black and the red curve coincide for high states of charge and a sharp drop towards positive values occurs at the minimum of the chemical potential. In Figure 7.7b, we see the effects of mechanics on the chemical potential. The sharp decline of the chemical potential is smaller and shifted behind the maximum (lithiation) respective minimum (delithiation). Moreover, the chemical potential decreases linearly over the state of charge instead of remaining constant.

To understand the hysteresis, we rely on the Gibbs stability criterion resulting from the second law of thermodynamics [25, 72, 83]

$$\frac{\partial \mu}{\partial c} > 0. \tag{7.31}$$

First, the particle lithiates to a SOC between the lithium poor phase and the lithium rich phase, which fulfills Equation (7.31). Once the chemical potential reaches the unstable region $\partial\mu/\partial c < 0$, the spinodal decomposition sets in and the particle spontaneously separates into a lithium poor and a lithium rich phase [168]. For delithiation, the process is inverted, because $-c$ instead of c increases over time. By including mechanics into this process, we extend the chemical potential by the elastic contributions

$$\mu_{\text{el}} = -\frac{v}{3\lambda_{\text{ch}}} \operatorname{tr}(\mathbb{C}\mathbf{E}_{\text{el}}). \tag{7.32}$$

Deriving Equation (7.32) with respect to the concentration yields

$$\frac{\partial \mu_{\text{el}}}{\partial c} = \left(\frac{v}{3\lambda_{\text{ch}}^2}\right)^2 \operatorname{tr}\big(\mathbb{C}(\mathbf{E}_{\text{el}} + 2\mathbf{C}_{\text{ch}})\big). \tag{7.33}$$

The derivative is positive, if the elastic deformation is smaller than the chemical deformation, which in our case is always fulfilled. Hence, the Gibbs stability criterion (7.31) is valid for higher SOC, so that the spinodal decomposition is shifted to higher SOC (lithiation), respective lower SOC (delithiation). After the spinodal decomposition, the elastic part of the chemical potential, Equation (7.32) keeps rising according to Equation (7.33). Thus, we observe a linear decline of the chemical part of the chemical potential, so that the overall chemical potential remains constant.

7.4.3. Numerical Efficiency

So far we discussed the results of the numerical experiments from a physical point of view. In the rest of this section we discuss the numerical aspects of the solution algorithm presented in Section 7.3. We first validate our implemented solver numerically. Then we compare the efficiency of the adaptive algorithm to standard methods. Finally, we employ our adaptive solver to the two-dimensional geometry.

Solver Validation

We validate our implementation by measuring the convergence of the numerical solution for spatial and temporal refinement, respectively. Because no exact solution is known for

this problem, we consider the L^2- and H^1-errors of the solution $y_h = [c_h, \mu_h, \boldsymbol{u}_h]^\mathsf{T}$ to a reference solution y_r at time $t_\mathrm{f} = 0.29$ (SOC $= 0.3$), when a phase transition is present

$$\begin{aligned}\mathrm{err} \coloneqq \Big(&\|c_h(t_\mathrm{f}, \cdot) - c_\mathrm{r}(t_\mathrm{f}, \cdot)\|^2 \\ &+ \|\mu_h(t_\mathrm{f}, \cdot) - \mu_\mathrm{r}(t_\mathrm{f}, \cdot)\|^2 + \|\boldsymbol{u}_h(t_\mathrm{f}, \cdot) - \boldsymbol{u}_\mathrm{r}(t_\mathrm{f}, \cdot)\|^2 \Big)^{1/2}.\end{aligned} \tag{7.34}$$

The reference solution y_r is assumed to approximate the unknown exact solution y_ex more accurately than y_h and is computed once with fourth-order finite elements and $\mathrm{RelTol}_t = 10^{-8}$, $\mathrm{AbsTol}_t = 10^{-11}$ and $\mathrm{RelTol}_x = 10^{-20}$, $\mathrm{AbsTol}_x = 10^{-11}$.

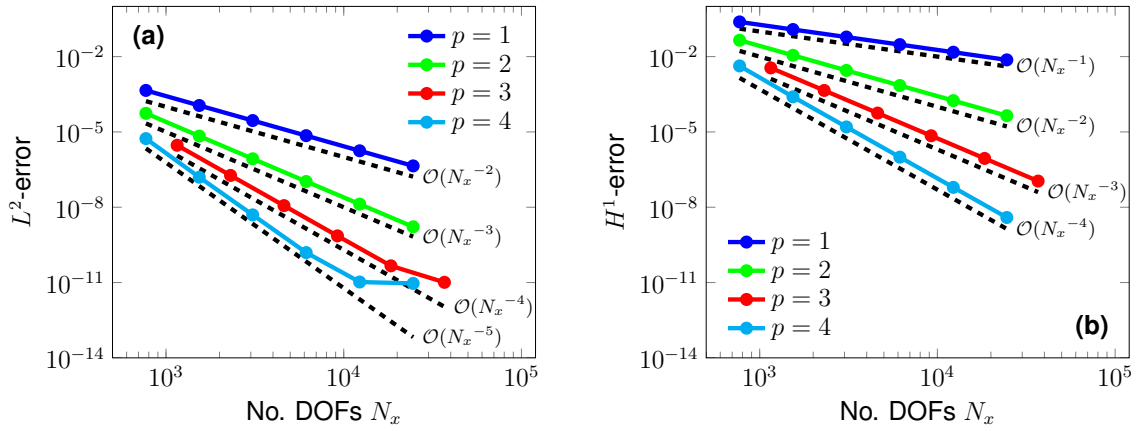

Figure 7.8.: Convergence analysis of the (a) L^2-error and (b) H^1-error for uniform spatial refinement with fixed tolerances for the time integration.

For the experimental order of convergence of the spatial error we fix the tolerances of an accurate time integration ($\mathrm{RelTol}_t = 10^{-7}$, $\mathrm{AbsTol}_t = 10^{-10}$) and compute the errors of the numerical solution on a series of successively uniformly refined meshes. In Figure 7.8 we plot the L^2- and the H^1-errors over the number of DOFs. According to the theory of finite elements [30], our implementation shows the optimal order of convergence for the finite element degrees one to four. Note, that the L^2-errors for third- and fourth-order elements at the finest level already reach a value of 10^{-11} when the saturation sets in. A convergence analysis of the time integration with fixed spatial tolerances ($\mathrm{RelTol}_x = 10^{-20}$, $\mathrm{AbsTol}_x = 10^{-9}$) for fourth-order finite elements yields L^2- and H^1-errors below 10^{-8}. This validates our implementation.

Adaptivity

Next we discuss the efficiency of our adaptive solution algorithm by analyzing the computational savings through spatial and temporal adaptivity for the simulation of lithium insertion as in Subsection 7.4.2.

Spatial Adaptivity In Figure 7.9a we illustrate the concentration profile in a phase separated state. Because the concentration profile is approximately constant except for the phase transition, a locally high resolution of the transition is advantageous. Accordingly, our algorithm distributes the DOFs adaptively considering the regularity estimator with

respect to all solution variables (c, μ, $\boldsymbol{u}$). The black dots in Figure 7.9a show the refinement level of the cells to reach a certain regularity tolerance. We observe that precisely the phase transition zone is resolved with additional DOFs.

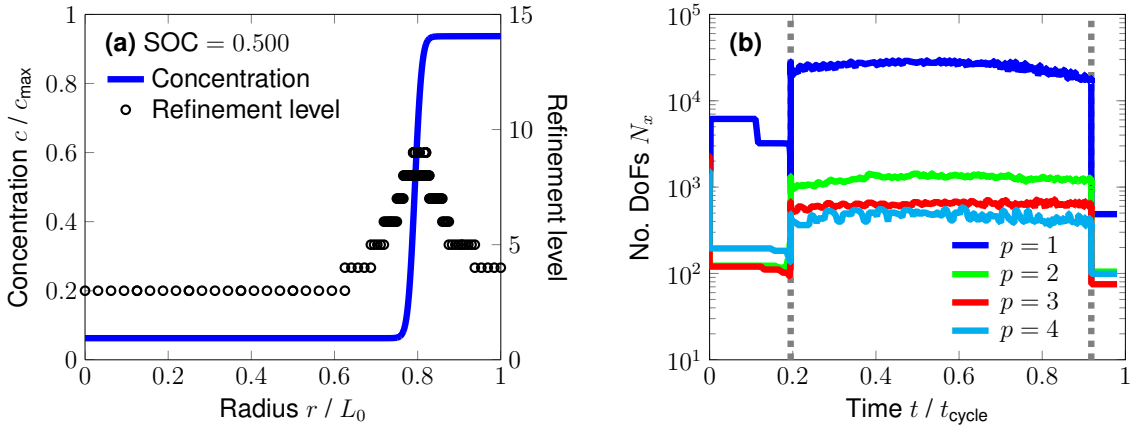

Figure 7.9.: (a) Concentration profile at SOC $= 0.5$ with adaptive refinement level and (b) number of DOFs over time for different finite element method degrees.

Over time the phase transition migrates through the particle and the mesh is adapted. In Figure 7.9b we plot the number of DOFs over time for different finite element orders. We again observe the three states of single and two-phase diffusion. In particular, we see that higher order methods are beneficial for saving DOFs. This is due to better approximation properties of higher order curvatures in the solution profiles, as for example in the chemical potential profile.

We emphasize the computational savings by a comparison of the number of DOFs. For the adaptive fourth-order method the minimal mesh width was 2^{-9} and 1539 DOFs were distributed at maximum. In contrast, a linear finite element method would require at least a uniform mesh mesh width 2^{-11} and $3 \times (2^{11} + 1) = 6147$ DOFs to distribute the same number of DOFs in the phase transition zone. We already reduced the maximum size of the linear systems by at least a factor of four. Thus spatial adaptivity is one key for highly efficient solution algorithms.

Temporal Adaptivity Another important key for the efficient solution is the adaptive time integration. In particular, we use an error controlled adaptive change of the time step size and the used order. Thereby we benefit from higher order methods, which allow even larger time step sizes at the same error tolerance. In Figure 7.10a we plot the time step size and the used order over time. We identify the three regimes of single and two-phase diffusion as before. The time step size varies over several orders of magnitude to capture the initiation and decay of the phase transition at the given tolerances. During two-phase diffusion up to the decay of the phase transition we observe a decreasing time step size. This reflects the observation that in the spherical symmetric case the phase transition migrates with increasing velocity towards the particle center. Thus, to keep the tolerances of the time integration, smaller time step sizes are necessary for the accelerating process.

The essential benefit from the temporal adaptivity is the use of large time step sizes during the two-phase diffusion. This advantage is particularly reflected in the number of time steps and the computational time. In Figure 7.10b we plot the computational times

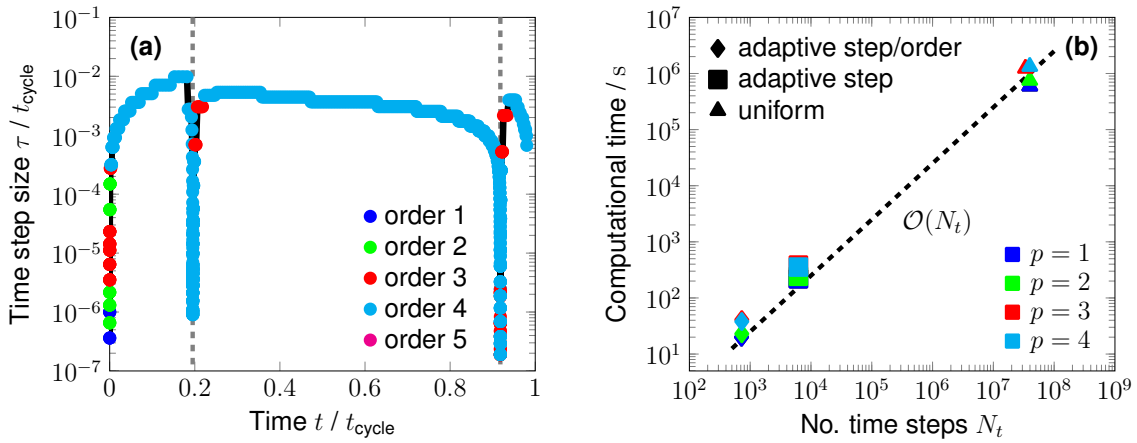

Figure 7.10.: (a) Time step size with used orders (markers) over time and (b) computational wall time over the number of time steps for different time integration schemes and finite element orders.

for the whole solution process over the number of time steps for different strategies of time integration and finite element orders. To neglect the influence of spatial adaptivity we used a uniform spatial discretization in all cases such that the phase transition is always resolved with at least ten unknowns ($h = 2^{\lfloor \log_2(p\,s/10) \rfloor}$). In terms of computational time and number of time steps, the most efficient strategy is the fully adaptive algorithm changing the time step sizes and the orders adaptively (♦). Using the backward Euler method with an adaptive time step size (■) the computational times already increase by a factor of ten. If we would use the minimum time step size for the backward Euler method reaching the given tolerances ($\tau = 2.5 \times 10^{-8}$) as uniform step size (▲), the number of time steps and the estimated computational time would increase by a factor of $\sim 10^4$, compared to the fully adaptive strategy (♦). Hereby *estimated* means, that we determined the computational time of one time step as average over 1000 steps and extrapolated this value to the total number of time steps.

Spheroidal Electrode Particle

As proof of concept, we demonstrate the capabilities and efficiency of our adaptive solution algorithm at the two-dimensional geometry described in Subsection 7.4.1. Even though the theory was derived assuming three-dimensional geometries, the mathematical problem is still valid in lower dimensions. So neglecting any additional three-dimensional symmetry we solve the model equations in Cartesian coordinates and postpone the further study of two- and three-dimensional particle geometries to future work.

In Figure 7.11 we see the concentration profile of the ellipse in the current configuration at three characteristic states during lithium insertion. In accordance with Santoki et al. [124] we observe in Figure 7.11a the initiation of the phase separation at surfaces with high curvature. The insertion process is then followed by the migration of two traveling fronts until they merge in the particle center. During lithiation we observe that unstable nucleation can occur forming small domains of lithium rich phases. However, these domains vanish after a short time and only the two stable fronts remain, see Figure 7.13b–7.13e.

In addition Figure 7.11 and 7.13 visualize the adaptive mesh. We see clearly that the

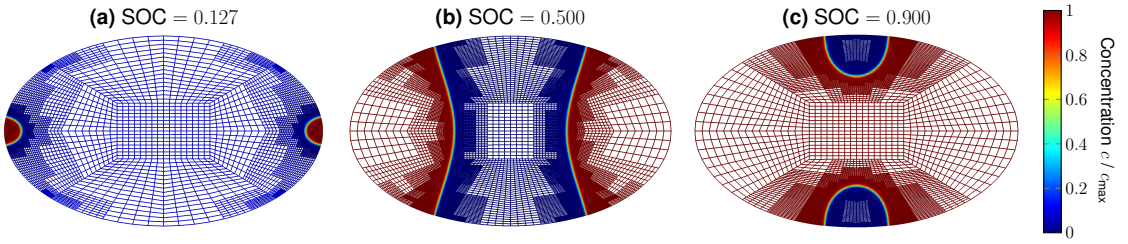

Figure 7.11.: Concentration profile in a two-dimensional ellipsoidal particle domain at different SOCs with adaptive meshes.

mesh is only resolved at the phase transition, which migrates through the particle's domain over time. In the lithium poor and rich bulk phases, a coarser resolution can be applied. In Figure 7.12 we depict the number of DOFs and the time step sizes with the used order over time. During the simulation we distributed between 2.5×10^3 and 1.7×10^5 DOFs. In contrast, a uniform mesh with the same maximum level as in the adaptive case would have approximately 2.1×10^7 DOFs. The time step size of the $\sim$2400 adaptive time steps varies between 10^{-8} and 10^{-2}. A uniform time discretization with the time step size 10^{-8} would take approximately 10^8 steps. Thus we reduced the number of DOFs by a factor of at least 10^2 and the number of time steps by a factor of at least 10^4. With the space and time adaptive algorithm we solve this example in less than 6 hours on the cluster specified in Section 7.4.1.

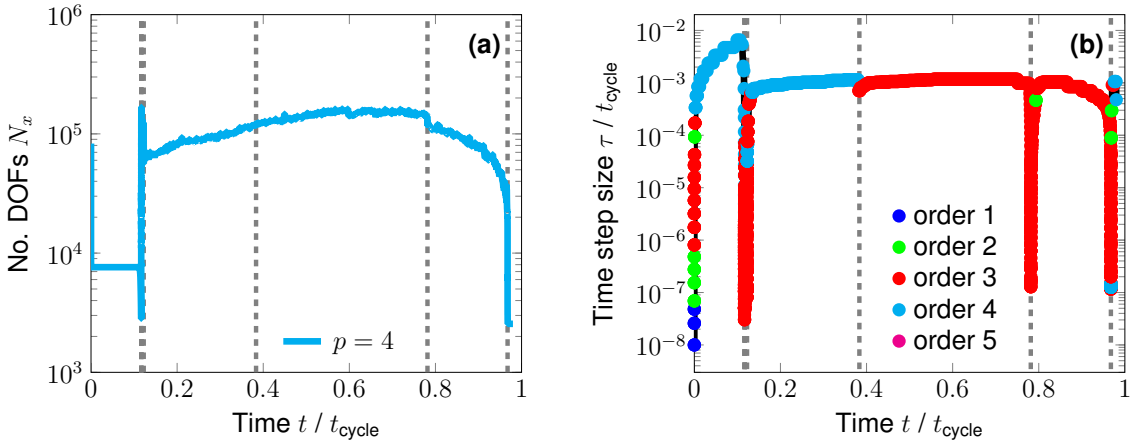

Figure 7.12.: (a) Number of DOFs and (b) time step size with used orders (markers) over time for the two-dimensional geometry.

Finally, in Figure 7.12 we indicate the time instances with locally small time step sizes by gray dashed lines. Except of the state related to Figure 7.13f at time $t/t_{\text{cycle}} \approx 0.38$, we can identify for each highlighted instance a specific event such as nucleation (Figure 7.13a–c), decay of phase transitions (Figure 7.13d–e,h) and other topological changes (Figure 7.13g). Note, that the first five events happens on such a small time scale, that the gray dashed lines in Figure 7.12 are not distinguishable. The reduction of the time step size at time $t/t_{\text{cycle}} \approx 0.38$ cannot be explained by a specific event due to phase transformation. It is more likely due to the increasing time step size of the adaptive algorithm until time $t/t_{\text{cycle}} \approx 0.38$. In order to fulfill the tolerances the time step size drops and the order is reduced. We

emphasize here again that the purpose of this simulation is only a proof of concept for the application of the space and time adaptive algorithm to the coupled multi-physical problem.

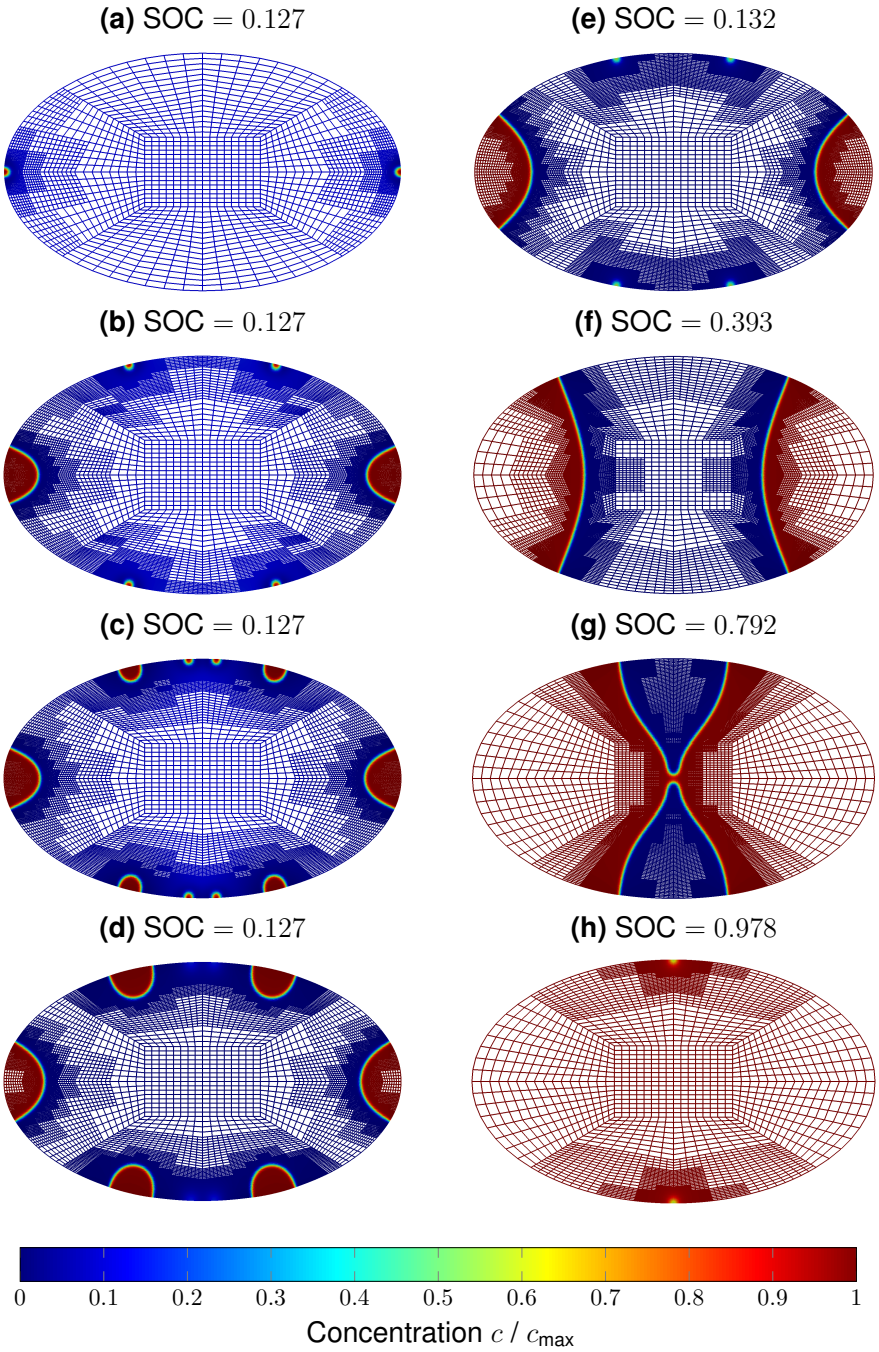

Figure 7.13.: Concentration profile in a two-dimensional ellipsoidal particle domain at SOCs with smallest adaptive time step sizes.

7.5. Summary and Conclusion

Summing up, we have developed a thermodynamically consistent phase-field model coupling Cahn–Hilliard-type phase separation and finite deformations during the lithiation

of electrode particles in lithium-ion batteries. Additionally, we have presented our implemented space and time adaptive, higher order finite element solver for the numerical simulation of the resulting model equations. For the example of lithium iron phosphate as electrode material we have performed several simulations to investigate physical and numerical aspects of the model and the solver.

A comparison of our simulation results with literature [46, 59, 85, 149, 167–170, 174] showed excellent qualitative accordance. In particular, we revealed that the interfacial energy coefficient κ is critical to determine the maximum occurring stress inside an electrode particle. However, three-dimensional phase-field simulations, as for example in [59, 173, 174], often rely on larger values of κ than for one-dimensional simulations [168–170] to reduce the computational costs, eventually leading to underestimated stress magnitudes.

Comparing the computational time with our problem-tailored solution algorithm to a standard implementation based on a constant step size backward Euler scheme together with a linear finite element method implementation revealed an estimated speedup of $\sim 10^4$. We achieve this speedup, by using a space and time adaptive, higher order finite element solver. The time adaptivity greatly decreases the simulation time by selectively resolving the emergence and decay of the phase separation. The spatial adaptivity reduces the computational costs by only locally refining the mesh at the phase transition. The advantages of adaptive methods are especially beneficial for the two-dimensional example of a generic spheroidal shaped electrode particle. With this analysis we hope to have demonstrated the advantage, that adaptive methods are a powerful tool for efficient simulations.

Future works will rely on the numerical efficiency of our implementation. Thereby, it will be possible to calculate three-dimensional phase-field problems with almost sharp interfaces leading to better predictions for the arising stress. Moreover, low computational costs of our single particle model make an implementation in full cell simulations feasible. As our methodology is very general, we can also easily analyze different phase-transforming materials and geometries.

8. Conclusion

Summing up, we considered the phenomenon of phase separation in electrode particles of lithium-ion batteries during lithium insertion and developed a matrix-free, parallel, adaptive finite element solver for the efficient numerical solution of the resulting Cahn–Hilliard-type phase-field model equations.

In the beginning of this thesis we explained the basic working principle of lithium-ion batteries and described the phenomenon of phase separation in possibly anisotropic electrode materials. For the modeling of phase separation in electrode particles under lithium insertion we followed the currently widely used Cahn–Hilliard-type phase-field approach from literature and discussed the model equations in a most general formulation accounting for anisotropic mobilities and interfacial energy coefficients. In contrast to the classical Cahn–Hilliard equation, we showed the violation of the mass conservation due to the inhomogeneous Neumann boundary condition modeling the lithium insertion. For the same reason we found that the monotonous energy decrease is no longer guaranteed. In fact, under lithium insertion our numerical experiments even show an increase of the system's free energy.

For the numerical solution of the Cahn–Hilliard model equations we developed a space and time adaptive solution algorithm based on a variable-step, variable-order time integration scheme and a gradient recovery estimator as criterion for the local mesh refinement. Thereby we employed a p-th-order isoparametric Lagrangian finite element method for the spatial discretization and used the NDF family of linear multistep methods for the discrete time stepping. The transformed nonlinear algebraic systems in each time step are linearized with Newton's method and solved in each Newton step with a preconditioned GMRES method. At the heart of the solver is our derived preconditioner, which incorporates the nonlinear mobility of the Cahn–Hilliard model. In a simplified problem setting with some additional approximations we proved mesh-independent eigenvalue estimates for the preconditioned system in the constant mobility case. This result indicates the optimality of our proposed preconditioner in the simplified setting. Furthermore, the preconditioner is applicable for matrix-free computations, because the inverse application of the two blocks can be computed by a CG method. In particular, the new aspect of our solver is the fact, that it is implementable completely without storing matrices. For the implementation in `C++` we used the functionalities of the *deal.II* library and benefit particularly from the efficiency of the matrix-free framework.

In a separated chapter we reviewed the key ideas of the matrix-free framework within the *deal.II* library. With an example of a nonlinear elliptic problem we demonstrated the strong parallel scalability of the parallel, matrix-free Newton solver based on a geometric multigrid preconditioned CG solver for the linearized systems. The layout of the nonlinear operator class served as basic building block for our Cahn–Hilliard solver.

Next we performed an exhaustive study on the performance and efficiency of the presented finite element solver and its components. First, we validated the solver based on the method of manufactured solutions, where we observed the optimal orders of convergence

for uniform spatial refinement. In terms of error over computational wall-clock time we observed the increasing efficiency of higher order finite element methods, which is supported by the implementation with the matrix-free framework. Based on three examples of spinodal decomposition we then studied the performance of the solver and its single components. For a special free energy density the numerically recovered equilibrium profile of the phase transition is in accordance with the given analytical expression. Further, we demonstrated, that the variable-step, variable-order time integration reduced the number of time steps dramatically and is thus inevitable for the efficient solution. In particular, we revealed the impact of the size of the Arnoldi basis on the number of GMRES steps. Although the storage of a larger history of the GMRES method is expensive in terms of memory consumption, the number of GMRES steps was significantly reduced. This shows a further benefit of the matrix-free framework, which avoids the storage of matrices and thus allows the storage of more Arnoldi basis vectors. Furthermore, at the example of the classical Cahn–Hilliard equation with degenerated mobility and logarithmic free energy we found numerically that with our proposed preconditioner the number of GMRES steps does not depend on the size of the uniform mesh and the finite element degree. It turned out that the incorporation of the nonlinear mobility is an essential improvement over the constant mobility preconditioner in terms of the number of iterations and the computational wall-clock time. A large scale three-dimensional simulation of spinodal decomposition demonstrated the parallel capability of our solver and supported the optimality of our proposed preconditioner. Based on an example of spinodal decomposition with smooth initial data, we demonstrated the functionality of the spatial adaptivity. However, we note, that the numerical solution of the Cahn–Hilliard equation is extremely sensitive to numerical instabilities. Already small perturbations can lead to qualitatively different phase transition morphologies during the coarsening process. Since the matrix-free framework allows the simulation of highly uniformly resolved domains, we disabled the spatial adaptivity in a first attempt.

Finally, we employed the developed solver for the simulation of phase separation in LFP electrode particles under lithium insertion. In the spherical symmetric case we measured the optimal orders of convergence by comparing the numerical solutions to an adaptively computed reference solution. We demonstrated the high demand and large computational savings through spatial and temporal adaptive methods. As for the classical Cahn–Hilliard equation we showed numerically the optimality of our proposed preconditioner. Especially for the spherical symmetric application problem we even showed the optimality on locally refined meshes. Next we assumed an azimuthal symmetry and simulated the phase separation during lithium insertion on a two-dimensional computational domain. We demonstrated the functionality in the two-dimensional case and particularly showed, that our solver can handle anisotropic mobilities and interfacial energy coefficients. In this case we observed a phase transition morphology, which is different to the core-shell scenario indicated by the spherical symmetric solution. In contrast, the anisotropic material parameters triggered the phase separation process in an aligned form. Note, that an imbalance of the small insertion rate compared and the large diffusion coefficient leads to instabilities in the numerical solutions at the curved particle surface boundary. Thus even for uniform refined meshes we observed a phase separation dynamic with randomly increasing lithium rich bubbles. To avoid an additional effect of the mesh we disabled the adaptive mesh refinement for the two-dimensional examples.

In the last chapter of this thesis we presented our currently submitted collaborative work,

where we combined a thermodynamically consistent mechanics theory for large deformations with the phase separation model in a common phase-field model. For this multiphysical problem we extended our adaptive solution algorithm and obtained an efficient matrix-based finite element solver. In our numerical experiments we discussed physical as well as numerical aspects. Thereby we revealed the dependence of the maximal arising hydrostatic stress on the interfacial energy coefficient. Especially, larger values of the interfacial energy coefficient leads to underestimated stresses. Under the assumption of spherical symmetry we observed for our finite element solver of the coupled problem the optimal orders of convergence. The comparison of different time integration strategies showed that the variable-step, variable-order time integration algorithm outperforms standard methods, like variable- and constant-step first-order time stepping schemes. In addition, the adaptive mesh refinement reduced the number of DOFs significantly. Finally, a two-dimensional example demonstrated the functionality and savings of the adaptive solver for the multiphysical problem. Different to the azimuthal symmetric simulations, the adaptive mesh refinement works well and reduced the amount of DOFs dramatically. The key difference is the geometry. It is known, that the phase separation is initiated at surfaces with high curvature. Thus in the quarter ellipsoidal domain the initiation of phase separation is triggered in the corner with highest curvature of the surface, whereas in the half circle domain the curvature is constant over the whole particle surface boundary.

Another outcome of this thesis is an implementation of the adaptive solution algorithm as a modular `C++` solver package, ready for future applications. In particular, the gradient recovery estimator and the subroutines for the convergence analysis based on a finite element reference solution are implemented in the style of the *deal.II* library, so that we could contribute these codes to the *deal.II* library in future. We also highlight the so far prepared draft of the *deal.II* tutorial `step-66` about a nonlinear matrix-free Newton solver, which is currently under review.

Based on this thesis a couple of future research objectives can be identified.

First of all, one could review the existence and uniqueness of a solution for the Cahn–Hilliard-type phase-field model starting from simplified versions with constant mobility and quartic free energy density. As pointed out, the inhomogeneous boundary condition, modeling lithium insertion, destroys the mass-conservation and the monotonous energy decay, which are missing key properties in the existing proofs. In particular, depending on the boundary condition, only local existence could be expected. Further improvements of the adaptive solution algorithm should replace the gradient recovery estimator by a reliable a posteriori error estimator. Additional optimizations of the implementation could increase the overall robustness of the solver. An interesting extension would also be the development of a matrix-free preconditioner for the coupled chemical-mechanical problem.

The capabilities of an improved version of our solver could be exploited for future electrode particle simulations. Thereby the influence of anisotropic materials and parameters in crucial ranges on the phase separation dynamics in more realistic, three-dimensional particle geometries could be studied. As we considered the chemical-mechanical coupling, a more realistic model could include a reviewed diffusion equation in the deformed configuration and for large deformations also plasticity should be considered. Finally, the efficient solution of the phase-field model allows the embedding of phase separation models in multiscale models describing the performance of battery cells.

A. Model Parameters

In this appendix we provide the material parameters for the electrode material lithium iron phosphate Li_xFePO_4 (LFP) and describe their normalization based on our article [44].

A.1. Material Parameters

In Table A.1 we summarize the material parameters for LFP particles based on [168, 169] and the references cited therein.

Table A.1.: Material parameters for LFP electrode particles.

Description	Symbol	Value	Unit
Particle length scale	L_0	150×10^{-9}	m
Diffusion coefficient	D	1×10^{-14}	$\mathrm{m^2\,s^{-1}}$
Coefficient for ψ_{ch}	α_1	4.5	$-$
Coefficient for ψ_{ch}	α_2	-9	$-$
Coefficient for ψ_{int}	κ	8.8×10^{-18}	$\mathrm{m^2}$
Young's modulus	$\mathrm{E_H}$	124.5×10^9	Pa
Poisson ratio	ν	0.25	$-$
Partial molar volume	v	2.9×10^{-6}	$\mathrm{m^3\,mol^{-1}}$
Maximal concentration	c_{max}	2.29×10^4	$\mathrm{mol\,m^{-3}}$

A.2. Normalization

We define the particle length scale in the reference configuration L_0 as reference for the spatial scale, the cycle time t_{cycle} for the time scale and the maximum lithium concentration c_{max} as reference concentration. With the operation temperature T equal to the room temperature T_{ref} we normalize the energy density using RTc_{max}. Table A.2 summarizes the arising dimensionless quantities.

The normalization yields three dimensionless parameters characterizing the coupled chemical-mechanical intercalation process: The *Fourier number* of mass transport Fo relates the diffusion time scale to the process time scale. The numbers $\widetilde{\mathrm{E}}_{\mathrm{H}}$ and $\widetilde{\kappa}$ relate the mechanical and the interfacial energy scale to the chemical energy scale.

Note, that the temporal reference value t_{cycle}, the cycle time, is defined according to the C-rate of the applied lithium flux N_{ext}, as discussed in Subsection 3.2.2 and 7.2.2. Thus,

Table A.2.: Normalization of basic model parameters.

$\widetilde{t} = t/t_{\text{cycle}}$	$\widetilde{\boldsymbol{X}}_0 = \boldsymbol{X}_0/L_0$	$\widetilde{\boldsymbol{u}} = \boldsymbol{u}/L_0$	$\widetilde{c} = c/c_{\text{max}}$	$\widetilde{T} = T/T_{\text{ref}}$
$\widetilde{v} = vc_{\text{max}}$	$\widetilde{\psi} = \psi/RTc_{\text{max}}$	$\widetilde{\mu} = \mu/RT$	$\widetilde{N}_{\text{ext}} = N_{\text{ext}}t_{\text{cycle}}/L_0c_{\text{max}}$	
$\text{Fo} = Dt_{\text{cycle}}/L_0^2$	$\widetilde{\kappa} = \kappa/L_0^2$	$\widetilde{\text{E}}_{\text{H}} = \text{E}_{\text{H}}/RTc_{\text{max}}$		

we finally summarize the normalized model parameters exemplary for an external lithium flux N_{ext} corresponding to the C-rate equal to one in Table A.3. For better readability we suppress the accentuation $\sim$.

Table A.3.: Normalized model parameters for LFP electrode particles.

Description	Symbol	Dimensionless Value
Operation temperature	T	1
Particle length scale	L_0	1
Diffusion coefficient	D	1.6×10^3
Coefficient for ψ_{ch}	α_1	4.5
Coefficient for ψ_{ch}	α_2	-9
Coefficient for ψ_{int}	κ	3.91×10^{-4}
Young's modulus	E_{H}	2.19×10^3
Poisson ratio	ν	0.25
Partial molar volume	v	6.64×10^{-2}
Maximal concentration	c_{max}	1
Initial concentration (chosen)	c_0	1×10^{-2}

B. Spherical Coordinates

Spherical coordinates are used in our electrode particle simulations. With symmetry assumptions we can reduce the dimension of the computational domain. For completeness we include the transformation of spherical coordinates we use throughout the thesis and derive the generic form of integrals for stiffness matrices in the spherical and azimuthal symmetric case.

B.1. Coordinate Transformation

In literature different definitions of spherical coordinates are available, which depend on the underlying discipline. We follow [152, Sect. IV.8], which reflects the physical convention. With spherical coordinates a point $\boldsymbol{p} \in \mathbb{R}^3$ is represented by the triple $(r, \theta, \varphi) \in \mathbb{R}^3$, which consists of the radius $r \in [0, \infty)$, the polar angle $\theta \in [0, \pi]$ and the azimuthal angle $\varphi \in [0, 2\pi]$. The spherical coordinates are mapped onto the Cartesian coordinates by

$$x = r\sin(\theta)\cos(\varphi), \qquad y = r\sin(\theta)\sin(\varphi), \qquad z = r\cos(\theta).$$

Note, that this mapping of $\mathbb{R}^3$ onto itself is not injective (consider for example the origin). For the application of the transformation theorem for integrals [67, 152] we obtain a C^1-diffeomorphism between spherical and Cartesian coordinates $g\colon B \to \mathbb{R}^3$, $(r, \theta, \varphi) \mapsto (x, y, z)$ on the open set $B = (0, \infty) \times (0, \pi) \times (0, 2\pi)$ [152, Sect. IV.8].

In spherical coordinates the space $\mathbb{R}^3$ is spanned by the three unit vectors

$$\boldsymbol{e}_r = \begin{pmatrix} \sin(\theta)\cos(\varphi) \\ \sin(\theta)\sin(\varphi) \\ \cos(\theta) \end{pmatrix} \qquad \boldsymbol{e}_\theta = \begin{pmatrix} \cos(\theta)\cos(\varphi) \\ \cos(\theta)\sin(\varphi) \\ -\sin(\theta) \end{pmatrix} \qquad \boldsymbol{e}_\varphi = \begin{pmatrix} -\sin(\varphi) \\ \cos(\varphi) \\ 0 \end{pmatrix}.$$

The Jacobian of the coordinate transformation is given by

$$\nabla g = \begin{pmatrix} \sin(\theta)\cos(\varphi) & r\cos(\theta)\cos(\varphi) & -r\sin(\theta)\sin(\varphi) \\ \sin(\theta)\sin(\varphi) & r\cos(\theta)\sin(\varphi) & r\sin(\theta)\cos(\varphi) \\ \cos(\theta) & -r\sin(\theta) & 0 \end{pmatrix} = \left(\boldsymbol{e}_r\,,\, r\,\boldsymbol{e}_\theta\,,\, r\sin(\theta)\,\boldsymbol{e}_\varphi\right),$$

with its determinant $J = \det(\nabla g)$ representing the volume element for spherical coordinates

$$\frac{\mathrm{d}(x, y, z)}{\mathrm{d}(r, \theta, \varphi)} = J(r, \theta, \varphi) = r^2\sin(\theta).$$

B.2. Calculus in Spherical Coordinates

Next we give formulas for the action of standard differential operators in spherical coordinates.

Let $u\colon \mathbb{R}^3 \to \mathbb{R}$ be a smooth scalar-valued function. The gradient of u in spherical coordinates is

$$\nabla u = \begin{pmatrix} \partial_r u \\ \frac{1}{r}\partial_\theta u \\ \frac{1}{r\sin(\theta)}\partial_\varphi u \end{pmatrix}.$$

Let $\boldsymbol{v}\colon \mathbb{R}^3 \to \mathbb{R}^3$ be a smooth vector-valued function with components $\boldsymbol{v} = (v_r, v_\theta, v_\varphi)^\mathsf{T}$. The divergence of $\boldsymbol{v}$ is given by

$$\nabla\cdot\boldsymbol{v} = \frac{1}{r^2}\partial_r\big(r^2 v_r\big) + \frac{1}{r\sin(\theta)}\partial_\theta\big(\sin(\theta)v_\theta\big) + \frac{1}{r\sin(\theta)}\partial_\varphi v_\varphi.$$

The Jacobian of $\boldsymbol{v}$ is

$$\nabla\boldsymbol{v} = \begin{pmatrix} \partial_r v_r & \frac{1}{r}\partial_\theta v_r - \frac{1}{r}v_\theta & \frac{1}{r\sin(\theta)}\partial_\varphi v_r - \frac{1}{r}v_\varphi \\ \partial_r v_\theta & \frac{1}{r}\partial_\theta v_\theta + \frac{1}{r}v_r & \frac{1}{r\sin(\theta)}\partial_\varphi v_\theta - \frac{\cot(\theta)}{r}v_\varphi \\ \partial_r v_\varphi & \frac{1}{r}\partial_\theta v_\varphi & \frac{1}{r\sin(\theta)}\partial_\varphi v_\varphi + \frac{\cot(\theta)}{r}v_\theta + \frac{1}{r}v_r \end{pmatrix}.$$

In the following we derive the differential operators for special kinds of symmetries. While spherical symmetry, also known as radial symmetry, is rather standard, the reduction to a two-dimensional formulation can be done with various assumptions. An alternative to the here presented approach can be found in [84, 170].

B.2.1. Spherical Symmetry

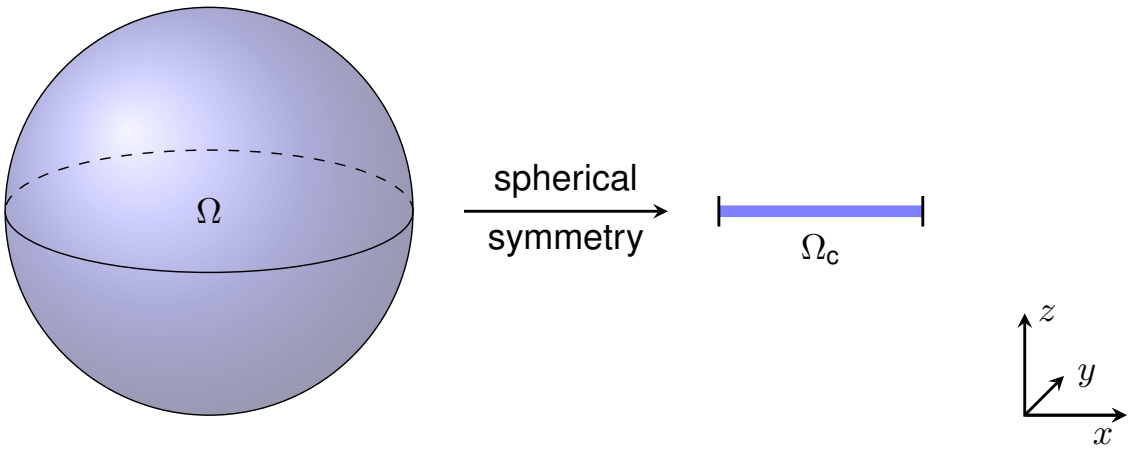

Figure B.1.: Dimension reduction of the three-dimensional unit ball to the one-dimensional unit interval assuming spherical symmetry.

In the spherical symmetric case we assume no dependence in both angular directions, i. e., fluxes and derivatives in $\boldsymbol{e}_\theta$- and $\boldsymbol{e}_\varphi$-direction are assumed to be zero ($v_\theta = v_\varphi = 0$, $\partial_\theta = \partial_\varphi = 0$). Thus the above differential operators get simplified to

$$\nabla u = \begin{pmatrix} \partial_r u \\ 0 \\ 0 \end{pmatrix}, \qquad \nabla \cdot \boldsymbol{v} = \frac{1}{r^2} \partial_r \left(r^2 v_r \right), \qquad \nabla \boldsymbol{v} = \begin{pmatrix} \partial_r v_r & 0 & 0 \\ 0 & \frac{1}{r} v_r & 0 \\ 0 & 0 & \frac{1}{r} v_r \end{pmatrix}.$$

Let $B = \{\boldsymbol{x} \in \mathbb{R}^3 \,:\, |\boldsymbol{x}| \leq 1\}$ be the three-dimensional unit ball and $v, w \colon B \to \mathbb{R}$ two generic scalar-valued functions. For the assembly of a generic stiffness matrix we compute the integral

$$\int_B \nabla v \cdot \nabla w \, \mathrm{d}\boldsymbol{x}.$$

Transforming the integral into spherical coordinates and using the simplified gradient, we obtain

$$\begin{aligned} \int_B \nabla v \cdot \nabla w \, \mathrm{d}\boldsymbol{x} &= \int_0^1 \int_0^\pi \int_0^{2\pi} \partial_r v \partial_r w \, r^2 \sin(\theta) \, \mathrm{d}r \, \mathrm{d}\theta \, \mathrm{d}\varphi \\ &= 4\pi \int_0^1 \partial_r v \partial_r w \, r^2 \, \mathrm{d}r. \end{aligned}$$

Thus the three-dimensional domain of the unit ball is reduced to the one-dimensional unit interval, see Figure B.1. To implement the spherical symmetric coordinate transformation we modify in the code the quadrature weight by the factor $4\pi r^2$.

For the chemical-mechanical coupled problem we also have to assemble integrals of the form

$$\int_B \nabla \boldsymbol{v} : \nabla \boldsymbol{w} \, \mathrm{d}\boldsymbol{x},$$

here with two generic vector-valued functions $\boldsymbol{v}, \boldsymbol{w} \colon B \to \mathbb{R}^3$. As above we transform the integral, insert the Jacobian in spherical coordinates and obtain

$$\begin{aligned} \int_B \nabla \boldsymbol{v} : \nabla \boldsymbol{w} \, \mathrm{d}\boldsymbol{x} &= \int_0^1 \int_0^\pi \int_0^{2\pi} \left(\partial_r v_r \partial_r w_r + \frac{2}{r^2} v_r w_r \right) r^2 \sin(\theta) \, \mathrm{d}r \, \mathrm{d}\theta \, \mathrm{d}\varphi \\ &= 4\pi \int_0^1 \left(\partial_r v_r \partial_r w_r + \frac{2}{r^2} v_r w_r \right) r^2 \, \mathrm{d}r. \end{aligned}$$

B.2.2. Azimuthal Symmetry

In the azimuthal symmetric case we assume no dependence only in the azimuthal angular direction, i. e., fluxes and derivatives in $\boldsymbol{e}_\varphi$-direction are assumed to be zero ($v_\varphi = 0$, $\partial_\varphi = 0$). In other words, the geometry and the solution are invariant under rotations around the z-axis. Thus the gradient operator gets simplified to

$$\nabla u = \begin{pmatrix} \partial_r u \\ \frac{1}{r} \partial_\theta u \\ 0 \end{pmatrix}.$$

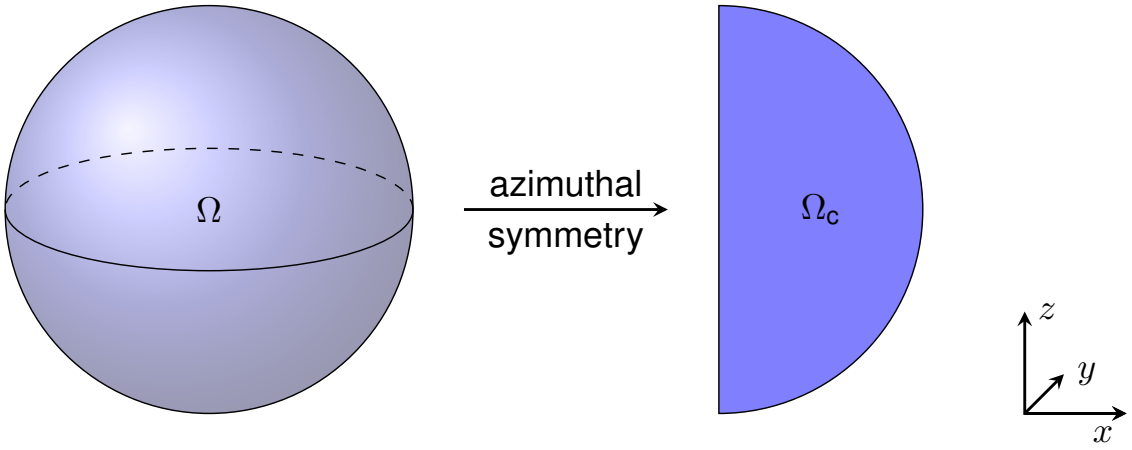

Figure B.2.: Dimension reduction of the three-dimensional unit ball to the two-dimensional unit half circle assuming azimuthal symmetry.

Let $B = \{\boldsymbol{x} \in \mathbb{R}^3 \,:\, |\boldsymbol{x}| \leq 1\}$ be the three-dimensional unit ball and $v, w \colon B \to \mathbb{R}$ two generic scalar-valued functions. For the assembly of a generic stiffness matrix we have to compute the integral

$$\int_B \nabla v \cdot \nabla w \,\mathrm{d}\boldsymbol{x}.$$

Transforming the integral into spherical coordinates and using the simplified gradient, we obtain

$$\begin{aligned}\int_B \nabla v \cdot \nabla w \,\mathrm{d}\boldsymbol{x} &= \int_0^1 \int_0^\pi \int_0^{2\pi} \left(\partial_r v \partial_r w + \frac{1}{r^2} \partial_\theta v \partial_\theta w\right) r^2 \sin(\theta) \,\mathrm{d}r \,\mathrm{d}\theta \,\mathrm{d}\varphi \\ &= 2\pi \int_0^1 \int_0^\pi \underbrace{\left(\partial_r v \partial_r w + \frac{1}{r^2} \partial_\theta v \partial_\theta w\right)}_{\nabla_{(r,\theta)} v \cdot \nabla_{(r,\theta)} w} r^2 \sin(\theta) \,\mathrm{d}r \,\mathrm{d}\theta.\end{aligned}$$

We identify the integrand as the scalar product of the two-dimensional polar coordinate gradients $\nabla_{(r,\theta)} v = \left(\partial_r v, \partial_\theta v / r\right)^\mathsf{T}$ and thus can apply the inverse transformation onto the two-dimensional unit circle $B_2 = \{\boldsymbol{x} \in \mathbb{R}^2 \,:\, |x| \leq 1\}$ in Cartesian coordinates

$$\int_B \nabla v \cdot \nabla w \,\mathrm{d}\boldsymbol{x} = \cdots = 2\pi \int_{B_2} \left(\partial_x v \partial_x w + \partial_z v \partial_z w\right) r \sin(\theta) \,\mathrm{d}\boldsymbol{x},$$

where $r = (x^2 + z^2)^{1/2}$ represents the radius and $\theta = \arctan(z/x)$ the polar angle. Note, that the inverse transformation from polar coordinates onto Cartesian coordinates eliminates a factor r of the original integration weight for spherical coordinates.

Thus the three-dimensional unit ball is reduced to the two-dimensional half circle, see Figure B.2. To implement the azimuthal symmetric coordinate transformation, we modify the quadrature weight in the code by a factor of $2\pi r \sin(\theta)$.

Since we use the azimuthal symmetry assumption only for simulations of phase separation in Section 6.2.2, we do not need the formulas for the divergence and the Jacobian of vector-valued functions under this assumption and can neglect them here.

Bibliography

[1] M. Adams *et al.*: *Parallel multigrid smoothing: Polynomial versus Gauss–Seidel*. J. Comput. Phys., 188(2):593–610, 2003.

[2] J. Ahrens, B. Geveci, and C. Law: *36 – ParaView: An end-user tool for large-data visualization*. In C. D. Hansen and C. R. Johnson (editors): *Visualization Handbook*, pages 717–731. Butterworth-Heinemann, Burlington, 2005.

[3] M. Ainsworth and J. T. Oden: *A Posteriori Error Estimation in Finite Element Analysis*. Pure and Applied Mathematics. John Wiley & Sons, Inc., New York, 2000.

[4] E. Alberdi Celaya, J. J. Anza Aquirrezabala, and P. Chatzipantelidis: *Implementation of an adaptive BDF2 formula and comparison with the MATLAB ode15s*. Proc. Comput. Sci., 29:1014–1026, 2014.

[5] G. Alzetta *et al.*: *The deal.II library, version 9.0*. J. Numer. Math., 26(4):173–183, 2018.

[6] M. Armand and J. M. Tarascon: *Building better batteries*. Nature, 451(7179):652–657, 2008.

[7] D. Arndt *et al.*: *ExaDG: Higher-order discontinuous galerkin for the exa-scale*. In H. J. Bungartz *et al.* (editors): *Software for the Exascale Computing - SPPEXA 2016–2019*, volume 136 of *Lect. Notes Comput. Sci. Eng.*, pages 189–225. Springer, Cham, 2020.

[8] O. Axelsson *et al.*: *Numerical and computational efficiency of solvers for two-phase problems*. Comput. Math. Appl., 65(3):301–314, 2013.

[9] V. E. Badalassi, H. D. Ceniceros, and S. Banerjee: *Computation of multiphase systems with phase field models*. J. Comput. Phys., 190(2):371–397, 2003.

[10] S. Balay *et al.*: *PETSc Web page*. https://www.mcs.anl.gov/petsc, 2019. `https://www.mcs.anl.gov/petsc`.

[11] S. Balay *et al.*: *PETSc users manual*. Technical Report ANL-95/11 - Revision 3.14, Argonne National Laboratory, 2020. `https://www.mcs.anl.gov/petsc`.

[12] W. Bangerth *et al.*: *Algorithms and data structures for massively parallel generic adaptive finite element codes*. ACM Trans. Math. Software, 38(2):Art. 14, 28, 2011.

[13] W. Bangerth, R. Hartmann, and G. Kanschat: *deal.II—a general-purpose object-oriented finite element library*. ACM Trans. Math. Software, 33(4):Art. 24, 27, 2007.

[14] E. Bänsch, P. Morin, and R. H. Nochetto: *Preconditioning a class of fourth order problems by operator splitting*. Numer. Math., 118(2):197–228, 2011.

[15] J. W. Barrett and J. F. Blowey: *Finite element approximation of the Cahn–Hilliard equation with concentration dependent mobility*. Math. Comp., 68(226):487–517, 1999.

[16] J. W. Barrett, J. F. Blowey, and H. Garcke: *Finite element approximation of the Cahn–Hilliard equation with degenerate mobility*. SIAM J. Numer. Anal., 37(1):286–318, 1999.

[17] G. F. Barros: *A space-time adaptivity scheme for the Cahn–Hilliard equation*. Master's thesis, Universidade Federal do Rio de Janeiro, 2019.

[18] S. Bartels and C. Carstensen: *Each averaging technique yields reliable a posteriori error control in FEM on unstructured grids. II. Higher order FEM*. Math. Comp., 71(239):971–994, 2002.

[19] S. Bartels and R. Müller: *A posteriori error controlled local resolution of evolving interfaces for generalized Cahn–Hilliard equations*. Interfaces Free Bound., 12(1):45–73, 2010.

[20] S. Bartels and R. Müller: *Error control for the approximation of Allen–Cahn and Cahn–Hilliard equations with a logarithmic potential*. Numer. Math., 119(3):409–435, 2011.

[21] P. Bastian *et al.*: *A generic grid interface for parallel and adaptive scientific computing. II. Implementation and tests in DUNE*. Computing, 82(2–3):121–138, 2008.

[22] P. Bastian *et al.*: *A generic grid interface for parallel and adaptive scientific computing. I. Abstract framework*. Computing, 82(2-3):103–119, 2008.

[23] L. Baňas and R. Nürnberg: *Adaptive finite element methods for Cahn–Hilliard equations*. J. Comput. Appl. Math., 218(1):2–11, 2008.

[24] L. Baňas and R. Nürnberg: *A posteriori estimates for the Cahn–Hilliard equation with obstacle free energy*. M2AN Math. Model. Numer. Anal., 43(5):1003–1026, 2009.

[25] M. Z. Bazant: *Thermodynamic stability of driven open systems and control of phase separation by electro-autocatalysis*. Faraday Discuss., 199:423–463, 2017.

[26] J. Bebernes and D. Eberly: *Mathematical problems from combustion theory*, volume 83 of *Applied Mathematical Sciences*. Springer-Verlag, New York, NY, 1989.

[27] A. L. Bertozzi, S. Esedoḡlu, and A. Gillette: *Inpainting of binary images using the Cahn–Hilliard equation*. IEEE Trans. Image Process., 16(1):285–291, 2007.

[28] J. Bosch and M. Stoll: *Preconditioning for vector-valued Cahn–Hilliard equations*. SIAM J. Sci. Comput., 37(5):S216–S243, 2015.

[29] P. Boyanova and M. Neytcheva: *Efficient numerical solution of discrete multi-component Cahn–Hilliard systems*. Comput. Math. Appl., 67(1):106–121, 2014.

[30] D. Braess: *Finite Elements*. Cambridge University Press, Cambridge, third edition, 2007.

[31] S. C. Brenner, A. E. Diegel, and L. Y. Sung: *A robust solver for a mixed finite element method for the Cahn–Hilliard equation*. J. Sci. Comput., 77(2):1234–1249, 2018.

[32] S. C. Brenner, A. E. Diegel, and L. Y. Sung: *A robust solver for a second order mixed finite element method for the Cahn–Hilliard equation*. J. Comput. Appl. Math., 364:112322, 12, 2020.

[33] S. C. Brenner and L. R. Scott: *The Mathematical Theory of Finite Element Methods*, volume 15 of *Texts in Applied Mathematics*. Springer, New York, NY, 3rd edition, 2008.

[34] A. Bressan and S. Takacs: *Sum factorization techniques in isogeometric analysis*. Comput. Methods Appl. Mech. Engrg., 352:437–460, 2019.

[35] P. G. Bruce, B. Scrosati, and J. M. Tarascon: *Nanomaterials for rechargeable lithium batteries*. Angew. Chem. Int. Ed., 47(16):2930–2946, 2008.

[36] C. Burstedde, L. C. Wilcox, and O. Ghattas: *p4est: scalable algorithms for parallel adaptive mesh refinement on forests of octrees*. SIAM J. Sci. Comput., 33(3):1103–1133, 2011.

[37] J. W. Cahn: *On spinodal decomposition*. Acta Metall., 9(9):795–801, 1961.

[38] J. W. Cahn and J. E. Hilliard: *Free energy of a nonuniform system. I. Interfacial free energy*. J. Chem. Phys., 28(2):258–267, 1958.

[39] C. Carstensen and S. Bartels: *Each averaging technique yields reliable a posteriori error control in FEM on unstructured grids. I. Low order conforming, nonconforming, and mixed FEM*. Math. Comp., 71(239):945–969, 2002.

[40] G. F. Castelli: *Die Numerische Simulation eines Mikroskalenmodells für Li-Ionen-Batterien*. Master's thesis, Karlsruhe Institute of Technology (KIT), 2016.

[41] G. F. Castelli and W. Dörfler: *An efficient matrix-free finite element solver for the Cahn–Hilliard equation*. In T. Gleim and S. Lange (editors): *Proceedings of the 8th GACM Colloquium on Computational Mechanics*, pages 441–444, 2019.

[42] G. F. Castelli and W. Dörfler: *The numerical study of a microscale model for lithium-ion batteries*. Comput. Math. Appl., 77(6):1527–1540, 2019.

[43] G. F. Castelli and W. Dörfler: *Study on an adaptive finite element solver for the Cahn–Hilliard equation*. In F. J. Vermolen and C. Vuik (editors): *Numerical Mathematics and Advanced Applications ENUMATH 2019*, volume 139 of *Lect. Notes Comput. Sci. Eng.*, pages 245–253. Springer, Cham, 2021.

[44] G. F. Castelli *et al.*: *Efficient simulation of chemical-mechanical coupling in battery active particles*. Energy Technol., 9(6):2000835, 2021.

[45] G. Chen, X. Song, and T. J. Richardson: *Electron microscopy study of the $LiFePO_4$ to $FePO_4$ phase transition*. Electrochem. Solid-State Lett., 9(6):A295–A298, 2006.

[46] L. Chen *et al.*: *A phase-field model coupled with large elasto-plastic deformation: Application to lithiated silicon electrodes*. J. Electrochem. Soc., 161(11):F3164–F3172, 2014.

[47] J. Christensen and J. Newman: *Stress generation and fracture in lithium insertion materials*. J. Solid State Electrochem., 10(5):293–319, 2006.

[48] D. A. Cogswell and M. Z. Bazant: *Coherency strain and the kinetics of phase separation in $LiFePO_4$ nanoparticles*. ACS Nano, 6(3):2215–2225, 2012.

[49] M. I. M. Copetti and C. M. Elliott: *Numerical analysis of the Cahn–Hilliard equation with a logarithmic free energy*. Numer. Math., 63(1):39–65, 1992.

[50] S. Dargaville and T. W. Farrell: *A comparison of mathematical models for phase-change in high-rate $LiFePO_4$ cathodes*. Electrochim. Acta, 111:474–490, 2013.

[51] S. Dargaville and T. W. Farrell: *A least square based finite volume method for the Cahn–Hilliard and Cahn–Hilliard-reaction equations*. J. Comput. Appl. Math., 273:225–244, 2015.

[52] T. A. Davis: *Algorithm 832: UMFPACK V4.3—an unsymmetric-pattern multifrontal method*. ACM Trans. Math. Software, 30(2):196–199, 2004.

[53] D. Davydov *et al.*: *A matrix-free approach for finite-strain hyperelastic problems using geometric multigrid*. Internat. J. Numer. Methods Engrg., 121(13):2874–2895, 2020.

[54] A. Debussche and L. Dettori: *On the Cahn–Hilliard equation with a logarithmic free energy*. Nonlinear Anal., 24(10):1491–1514, 1995.

[55] M. Dehghan and V. Mohammadi: *The numerical solution of Cahn–Hilliard (CH) equation in one, two and three-dimensions via globally radial basis functions (GRBFs) and RBFs-differential quadrature (RBFs-DQ) methods*. Eng. Anal. Bound. Elem., 51:74–100, 2015.

[56] C. Delmas *et al.*: *Lithium deintercalation in $LiFePO_4$ nanoparticles via a domino-cascade model*. Nat. Mater., 7(8):665–671, 2008.

[57] R. Deshpande *et al.*: *Diffusion induced stresses and strain energy in a phase-transforming spherical electrode particle*. J. Electrochem. Soc., 158(6):A718–A724, 2011.

[58] S. DeWitt *et al.*: *PRISMS-PF: A general framework for phase-field modeling with a matrix-free finite element method*. npj Comput. Mater., 6(1):1–12, 2020.

[59] C. V. Di Leo, E. Rejovitzky, and L. Anand: *A Cahn–Hilliard-type phase-field theory for species diffusion coupled with large elastic deformations: Application to phase-separating Li-ion electrode materials*. J. Mech. Phys. Solids, 70:1–29, 2014.

[60] W. Dörfler: *Eine FORTRAN-Bibliothek zur Lösung nichtlinearer Gleichungssysteme*, 1996.

[61] W. Dreyer and C. Guhlke: *Sharp limit of the viscous Cahn–Hilliard equation and thermodynamic consistency*. Contin. Mech. Thermodyn., 29(4):913–934, 2017.

[62] C. M. Elliott: *The Cahn–Hilliard model for the kinetics of phase separation*. In *Mathematical models for phase change problems (Óbidos, 1988)*, volume 88 of *Internat. Ser. Numer. Math.*, pages 35–73. Birkhäuser, Basel, 1989.

[63] C. M. Elliott, D. A. French, and F. A. Milner: *A second order splitting method for the Cahn–Hilliard equation*. Numer. Math., 54(5):575–590, 1989.

[64] C. M. Elliott and H. Garcke: *On the Cahn–Hilliard equation with degenerate mobility*. SIAM J. Math. Anal., 27(2):404–423, 1996.

[65] C. M. Elliott and S. Luckhaus: *A generalised diffusion equation for phase separation of multi-component mixture with interfacial free energy*. IMA (Minnesota) Preprint #887, 1991.

[66] C. M. Elliott and Z. Songmu: *On the Cahn–Hilliard equation*. Arch. Rational Mech. Anal., 96(4):339–357, 1986.

[67] J. Elstrod: *Maß- und Integrationstheorie*. Springer-Lehrbuch. Springer, Berlin, Heidelberg, 2011.

[68] D. J. Eyre: *Unconditionally gradient stable time marching the Cahn–Hilliard equation*. In *Computational and mathematical models of microstructural evolution (San Francisco, CA, 1998)*, volume 529 of *Mater. Res. Soc. Sympos. Proc.*, pages 39–46. MRS, Warrendale, PA, 1998.

[69] H. Fakih: *Asymptotic behavior of a generalized Cahn–Hilliard equation with a mass source*. Appl. Anal., 96(2):324–348, 2017.

[70] N. Fehn, W. A. Wall, and M. Kronbichler: *A matrix-free high-order discontinuous Galerkin compressible Navier–Stokes solver: a performance comparison of compressible and incompressible formulations for turbulent incompressible flows*. Internat. J. Numer. Methods Fluids, 89(3):71–102, 2019.

[71] X. Feng and H. Wu: *A posteriori error estimates for finite element approximations of the Cahn–Hilliard equation and the Hele-Shaw flow*. J. Comput. Math., 26(6):767–796, 2008.

[72] J. W. Gibbs: *The collected works of J. Willard Gibbs: In two volumes*. New York: Longmans, Green and Co., 1928.

[73] H. Gómez *et al.*: *Isogeometric analysis of the Cahn–Hilliard phase-field model*. Comput. Methods Appl. Mech. Engrg., 197(49-50):4333–4352, 2008.

[74] H. Gómez and T. J. R. Hughes: *Provably unconditionally stable, second-order time-accurate, mixed variational methods for phase-field models*. J. Comput. Phys., 230(13):5310–5327, 2011.

[75] E. Hairer, S. P. Nørsett, and G. Wanner: *Solving ordinary differential equations. I*, volume 8 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 1993.

[76] E. Hairer and G. Wanner: *Solving ordinary differential equations. II*, volume 14 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2010.

[77] B. C. Han *et al.*: *Electrochemical modeling of intercalation processes with phase field models*. Electrochim. Acta, 49(26):4691–4699, 2004.

[78] M. A. Heroux *et al.*: *An overview of the Trilinos Project*. ACM Trans. Math. Software, 31(3):397–423, 2005.

[79] V. Hoffmann *et al.*: *Influence of a silver salt on the nanostructure of a Au(111)/ionic liquid interface: An atomic force microscopy study and theoretical concepts*. Phys. Chem. Chem. Phys., 20(7):4760–4771, 2018.

[80] T. Hofmann, R. Müller, and H. Andrä: *A fast immersed interface method for the Cahn–Hilliard equation with arbitrary boundary conditions in complex domains*. Comput. Mater. Sci., 140:22–31, 2017.

[81] T. Hofmann *et al.*: *Numerical simulation of phase separation in cathode materials of lithium ion batteries*. Int. J. Solids Struct., 100–101:456–469, 2016.

[82] G. A. Holzapfel: *Nonlinear Solid Mechanics*. John Wiley & Sons, Ltd., Chichester, 2000.

[83] B. Horstmann *et al.*: *Rate-dependent morphology of Li_2O_2 growth in Li-O_2 batteries*. J. Phys. Chem. Lett., 4(24):4217–4222, 2013.

[84] M. Huttin: *Phase-field modeling of the influence of mechanical stresses on charging and discharging processes in lithium ion batteries*. PhD thesis, Karlsruhe Institute of Technology (KIT), 2014.

[85] M. Huttin and M. Kamlah: *Phase-field modeling of stress generation in electrode particles of lithium ion batteries*. Appl. Phys. Lett., 101(13):133902–1–133902–4, 2012.

[86] E. Javierre *et al.*: *A comparison of numerical models for one-dimensional Stefan problems*. J. Comput. Appl. Math., 192(2):445–459, 2006.

[87] D. Jodlbauer, U. Langer, and T. Wick: *Matrix-free multigrid solvers for phase-field fracture problems*. Comput. Methods Appl. Mech. Engrg., 372:113431, 2020.

[88] D. Jodlbauer, U. Langer, and T. Wick: *Parallel matrix-free higher-order finite element solvers for phase-field fracture problems*. Math. Comput. Appl., 25(3):40, 2020.

[89] S. Kaessmair and P. Steinmann: *Comparative computational analysis of the Cahn–Hilliard equation with emphasis on C^1-continuous methods*. J. Comput. Phys., 322:783–803, 2016.

[90] M. Kästner, P. Metsch, and R. de Borst: *Isogeometric analysis of the Cahn–Hilliard equation—a convergence study*. J. Comput. Phys., 305:360–371, 2016.

[91] M. A. Kespe: *Simulation von Transportvorgängen in Lithium-Ionen Batterien auf der partikulären Ebene*. PhD thesis, Karlsruhe Institute of Technology (KIT), 2019.

[92] E. Khain and L. M. Sander: *Generalized Cahn–Hilliard equation for biological applications*. Phys. Rev. E, 77(5):051129, 2008.

[93] R. C. Kirby and L. Mitchell: *Solver composition across the PDE/linear algebra barrier*. SIAM J. Sci. Comput., 40(1):C76–C98, 2018.

[94] R. W. Klopfenstein: *Numerical differentiation formulas for stiff systems of ordinary differential equations*. RCA Rev., 32:447–462, 1971.

[95] P. Knabner and L. Angermann: *Numerical Methods for Elliptic and Parabolic Partial Differential Equations*, volume 44 of *Texts in Applied Mathematics*. Springer, New York, NY, 2003.

[96] K. Kormann and M. Kronbichler: *Parallel finite element operator application: Graph partitioning and coloring*. In *Proceedings of the 7th IEEE International Conference on e-Science*, 2011.

[97] M. Kronbichler, A. Diagne, and H. Holmgren: *A fast massively parallel two-phase flow solver for microfluidic chip simulation*. Int. J. High Perf. Comput. Appl., 32(2):266–287, 2018.

[98] M. Kronbichler and K. Kormann: *A generic interface for parallel cell-based finite element operator application*. Comput. Fluids, 63:135–147, 2012.

[99] M. Kronbichler and G. Kreiss: *A phase-field microscale enhancement for macro models of capillary-driven contact point dynamics*. J. Comput. Multiph. Flows, 9(3):114–126, 2017.

[100] M. Kronbichler and K. Ljungkvist: *Multigrid for matrix-free high-order finite element computations on graphics processors*. ACM Trans. Parallel Comput., 6(1):Art. 2, 32, 2019.

[101] L. Laffont *et al.*: *Study of the $LiFePO_4/FePO_4$ two-phase system by high-resolution electron energy loss spectroscopy*. Chem. Mater., 18(23):5520–5529, 2006.

[102] D. Li and H. Zhou: *Two-phase transition of Li-intercalation compounds in Li-ion batteries*. materialstoday, 17(9):451–463, 2014.

[103] K. Ljungkvist: *Matrix-free finite-element operator application on graphics processing units*. In L. Lopes *et al.* (editors): *Euro-Par 2014: Parallel Processing Workshops*, volume 8806 of *Lecture Notes in Computer Science*. Springer, Cham, 2014.

[104] A. Logg: *Automating the finite element method*. Arch. Comput. Methods Eng., 14(2):93–138, 2007.

[105] A. Logg and G. N. Wells: *DOLFIN: Automated finite element computing*. ACM Trans. Math. Software, 37(2):Art. 20, 28, 2010.

[106] F. Magoulés, F. X. Roux, and G. Houzeaux: *Parallel Scientific Computing*. John Wiley & Sons, Inc., 2016.

[107] M. Maier: *The Mathematical Analysis of a Micro Scale Model for Lithium-Ion Batteries*. PhD thesis, Karlsruhe Institute of Technology (KIT), Karlsruhe, 2016.

[108] A. Manthiram: *An outlook on lithium ion battery technology*. ACS Cent. Sci., 3(10):1063–1069, 2017.

[109] A. Manthiram: *A reflection on lithium-ion battery cathode chemistry*. Nat. Commun., 11:1550, 2020.

[110] J. Martínez-Frutos and D. Herrero-Pérez: *Efficient matrix-free GPU implementation of Fixed Grid Finite Element Analysis*. Finite Elem. Anal. Des., 104:61–71, 2015.

[111] M. T. McDowell *et al.*: *Studying the kinetics of crystalline silicon nanoparticle lithiation with in situ transmission electron microscopy*. Adv. Mater., 24(45):6034–6041, 2012.

[112] E. Meca, A. Münch, and B. Wagner: *Thin-film electrodes for high-capacity lithium-ion batteries: Influence of phase transformations on stress*. Proc. R. Soc. A., 472(2193):20160093, 2016.

[113] E. Meca, A. Münch, and B. Wagner: *Sharp-interface formation during lithium intercalation into silicon*. Eur. J. Appl. Math., 29(1):118–145, 2018.

[114] A. Miranville: *Asymptotic behaviour of a generalized Cahn–Hilliard equation with a proliferation term*. Appl. Anal., 92(6):1308–1321, 2013.

[115] D. Morgan, A. Van der Ven, and G. Ceder: *Li conductivity in Li_xMPO_4 (M=Mn,Fe,Co,Ni) olivine materials*. Electrochem. Solid State Lett., 7:A30–A32, 2004.

[116] N. Nadkarni *et al.*: *Interplay of phase boundary anisotropy and electro-autocatalytic surface reactions on the lithium intercalation dynamics in Li_xFePO_4 plateletlike nanoparticles*. Phys. Rev. Materials, 2(8):085406, 2018.

[117] J. Newman and K. E. Thomas-Alyea: *Electrochemical Systems*. Electrochemical Society Series. John Wiley & Sons, Inc., Hoboken, NJ, 3rd edition, 2004.

[118] A. K. Padhi, K. S. Nanjundaswamy, and J. B. Goodenough: *Phospho-olivines as positive-electrode materials for rechargeable lithium batteries*. J. Electrochem. Soc., 144(4):1188–1194, 1997.

[119] M. Park *et al.*: *A review of conduction phenomena in Li-ion batteries*. J. Power Sources, 195:7904–7929, 2010.

[120] R. Plato: *Numerische Mathematik kompakt*. Vieweg+Teubner, Wiesbaden, 2010.

[121] F. Rathgeber *et al.*: *Firedrake: Automating the finite element method by composing abstractions*. ACM Trans. Math. Software, 43(3):Art. 24, 27, 2017.

[122] P. J. Roache: *Code verification by the method of manufactured solutions*. J. Fluids Eng., 124(1):4–10, 2002.

[123] Y. Saad: *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, PA, second edition, 2003.

[124] J. Santoki *et al.*: *Phase-field study of surface irregularities of a cathode particle during intercalation*. Modelling Simul. Mater. Sci. Eng., 26(6):065013, 2018.

[125] M. Schammer, B. Horstmann, and A. Latz: *Theory of transport in highly concentrated electrolytes*. J. Electrochem. Soc., 168(2):026511, 2021.

[126] J. Schöberl: *C++11 implementation of finite elements in NGSolve*, 2014. ASC Report No. 30/2014, Institute for Analysis and Scientific Computing, Vienna University of Technology.

[127] S. Schoeder *et al.*: *Efficient explicit time stepping of high order discontinuous Galerkin schemes for waves*. SIAM J. Sci. Comput., 40(6):C803–C826, 2018.

[128] L. F. Shampine, I. Gladwell, and S. Thompson: *Solving ODEs with MATLAB*. Cambridge University Press, Cambridge, 2003.

[129] L. F. Shampine and M. W. Reichelt: *The MATLAB ODE suite*. SIAM J. Sci. Comput., 18(1):1–22, 1997.

[130] L. F. Shampine, M. W. Reichelt, and J. A. Kierzenka: *Solving index-1 DAEs in MATLAB and Simulink*. SIAM Rev., 41(3):538–552, 1999.

[131] B. W. Sheldon *et al.*: *Stress contributions to solution thermodynamics in Li-Si alloys*. Electrochem. Solid-State Lett., 15(1):A9, 2012.

[132] H. C. Shin and S. I. Pyun: *An investigation of the electrochemical intercalation of lithium into a $Li_{1-\delta}CoO_2$ electrode based upon numerical analysis of potentiostatic current transients*. Electrochim. Acta, 44:2235–2244, 1999.

[133] H. C. Shin and S. I. Pyun: *The kinetics of lithium transport through $Li_{1-\delta}CoO_2$ by theoretical analysis of current transient*. Electrochim. Acta, 45:489–501, 1999.

[134] G. K. Singh, G. Ceder, and M. Z. Bazant: *Intercalation dynamics in rechargeable battery materials: General theory and phase-transformation waves in $LiFePO_4$*. Electrochim. Acta, 53:7599–7613, 2008.

[135] V. Srinivasan and J. Newman: *Discharge model for the lithium iron-phosphate electrode*. J. Electrochem. Soc., 151(10):A1517–A1529, 2004.

[136] R. H. Stogner, G. F. Carey, and B. T. Murray: *Approximation of Cahn–Hilliard diffuse interface models using parallel adaptive mesh refinement and coarsening with C^1 elements*. Internat. J. Numer. Methods Engrg., 76(5):636–661, 2008.

[137] V. R. Subramanian, H. J. Ploehn, and R. E. White: *Shrinking core model for the discharge of a metal hydride electrode*. J. Electrochem. Soc., 147(8):2868–2873, 2000.

[138] M. Tang, J. F. Belak, and M. R. Dorr: *Anisotropic phase boundary morphology in nanoscale olivine electrode particles*. J. Phys. Chem. C, 115(11):4922–4926, 2011.

[139] J. M. Tarascon: *Key challenges in future Li-battery research*. Phil. Trans. R. Soc. A, 368:3227–3241, 2010.

[140] J. M. Tarascon and M. Armand: *Issues and challenges facing rechargeable lithium batteries*. Nature, 414(6861):359–367, 2001.

[141] The Trilinos Project Team: *The Trilinos Project Website*, 2020. `https://trilinos.github.io`.

[142] R. Temam: *Infinite-Dimensional Dynamical Systems in Mechanics and Physics*, volume 68 of *Applied Mathematical Sciences*. Springer-Verlag, New York, NY, 1988.

[143] The MathWorks, Inc.: *Matlab*. `http://www.mathworks.com`.

[144] H. Tian *et al.*: *High capacity group-IV elements (Si, Ge, Sn) based anodes for lithium-ion batteries*. J. Materiomics, 1(3):153–169, 2015.

[145] G. Tierra and F. Guillén-González: *Numerical methods for solving the Cahn–Hilliard equation and its applicability to related energy-based models*. Arch. Comput. Methods Eng., 22(2):269–289, 2015.

[146] A. Van Der Ven, J. Bhattacharya, and A. A. Belak: *Understanding Li diffusion in Li-intercalation compounds*. Acc. Chem. Res., 46(5):1216–1225, 2013.

[147] A. Van Der Ven *et al.*: *Phase transformations and volume changes in spinel* $Li_xMn_2O_4$. Solid State Ion., 135(1–4):21–32, 2000.

[148] R. S. Varga: *Matrix Iterative Analysis*, volume 27 of *Springer Series in Computational Mathematics*. Springer, Berlin, Heidelberg, 2000.

[149] A. C. Walk, M. Huttin, and M. Kamlah: *Comparison of a phase-field model for intercalation induced stresses in electrode particles of lithium ion batteries for small and finite deformation theory*. Eur. J. Mech. A Solids, 48:74–82, 2014.

[150] M. J. Welland *et al.*: *Miscibility gap closure, interface morphology, and phase microstructure of 3D* Li_xFePO_4 *nanoparticles from surface wetting and coherency strain*. ACS Nano, 9(10):9757–9771, 2015.

[151] G. N. Wells, E. Kuhl, and K. Garikipati: *A discontinuous Galerkin method for the Cahn–Hilliard equation*. J. Comput. Phys., 218(2):860–877, 2006.

[152] D. Werner: *Einführung in die höhere Analysis*. Springer-Lehrbuch. Springer-Verlag, Berlin, 2009.

[153] S. Wise, J. Kim, and J. Lowengrub: *Solving the regularized, strong anisotropic Cahn–Hilliard equation by an adaptive nonlinear multigrid method*. J. Comput. Phys., 226(1):414–446, 2007.

[154] O. Wodo and B. Ganapathysubramanian: *Computationally efficient solution to the Cahn–Hilliard equation: Adaptive implicit time schemes, mesh sensitivity analysis and the 3D isoperimetric problem*. J. Comput. Phys., 230(15):6037–6060, 2011.

[155] L. Wu *et al.*: *Phase field modeling of coupled phase separation and diffusion-induced stress in lithium iron phosphate particles reconstructed from synchrotron nano X-ray tomography*. ASME J. Electrochem. En. Conv. Stor., 16(4):041006, 2019.

[156] B. X. Xu, Y. Zhao, and P. Stein: *Phase field modeling of electrochemically induced fracture in Li-ion battery with large deformation and phase segregation*. GAMM-Mitt., 39(1):92–109, 2016.

[157] R. Xu, L. Scalco De Vasconcelos, and K. Zhao: *Computational analysis of chemo-mechanical behaviors of composite electrodes in Li-ion batteries*. J. Mater. Res., 31(18):2715–2727, 2016.

[158] R. Xu and K. Zhao: *Electrochemomechanics of electrodes in Li-ion batteries: A review*. J. Electrochem. En. Conv. Stor., 13(3):030803, 2016.

[159] A. Yamada *et al.*: *Room-temperature miscibility gap in Li_xFePO_4*. Nat. Mater., 5(5):357–360, 2006.

[160] H. Yang *et al.*: *Strong kinetics-stress coupling in lithiation of Si and Ge anodes*. Extreme Mech. Lett., 2:1–6, 2015.

[161] K. G. van der Zee *et al.*: *Goal-oriented error estimation for Cahn–Hilliard models of binary phase transition*. Numer. Methods Partial Differential Equations, 27(1):160–196, 2011.

[162] Y. Zeng and M. Z. Bazant: *Phase separation dynamics in isotropic ion-intercalation particles*. SIAM J. Appl. Math., 74(4):980–1004, 2014.

[163] K. Zhang *et al.*: *A phase-field study of the effect of local deformation velocity on lithiation-induced stress in wire-like structures*. J. Phys. D: Appl. Phys., 52:145501, 2019.

[164] L. Zhang *et al.*: *A quantitative comparison between C^0 and C^1 elements for solving the Cahn–Hilliard equation*. J. Comput. Phys., 236:74–80, 2013.

[165] Q. Zhang and R. E. White: *Moving boundary model for the discharge of a $LiCoO_2$ electrode*. J. Electrochem. Soc., 154(6):A587–A596, 2007.

[166] T. Zhang: *Phase-field Modeling of Phase Changes and Mechanical Stresses in Electrode Particles of Secondary Batteries*. PhD thesis, Karlsruhe Institute of Technology (KIT), 2019.

[167] T. Zhang and M. Kamlah: *A nonlocal species concentration theory for diffusion and phase changes in electrode particles of lithium ion batteries*. Contin. Mech. Thermodyn., 30(3):553–572, 2018.

[168] T. Zhang and M. Kamlah: *Sodium ion batteries particles: Phase-field modeling with coupling of Cahn–Hilliard equation and finite deformation elasticity*. J. Electrochem. Soc., 165(10):A1997–A2007, 2018.

[169] T. Zhang and M. Kamlah: *Phase-field modeling of the particle size and average concentration dependent miscibility gap in nanoparticles of $Li_xMn_2O_4$, Li_xFePO_4, and Na_xFePO_4 during insertion*. Electrochim. Acta, 298:31–42, 2019.

[170] T. Zhang and M. Kamlah: *Mechanically coupled phase-field modeling of microstructure evolution in sodium ion batteries particles of Na_xFePO_4*. J. Electrochem. Soc., 167(2):020508, 2020.

[171] X. Zhang, W. Shyy, and A. M. Sastry: *Numerical simulation of intercalation-induced stress in Li-ion battery electrode particles*. J. Electrochem. Soc., 154(10):A910–A916, 2007.

[172] Y. Zhao *et al.*: *A review on modeling of electro-chemo-mechanics in lithium-ion batteries*. J. Power Sources, 413:259–283, 2019.

[173] Y. Zhao, P. Stein, and B. X. Xu: *Isogeometric analysis of mechanically coupled Cahn–Hilliard phase segregation in hyperelastic electrodes of Li-ion batteries*. Comput. Methods Appl. Mech. Engrg., 297:325–347, 2015.

[174] Y. Zhao *et al.*: *Phase-field study of electrochemical reactions at exterior and interior interfaces in Li-ion battery electrode particles*. Comput. Methods Appl. Mech. Engrg., 312:428–446, 2016.

[175] B. Zheng *et al.*: *Fast multilevel solvers for a class of discrete fourth order parabolic problems*. J. Sci. Comput., 69(1):201–226, 2016.

[176] J. Zhou *et al.*: *An efficient two-grid scheme for the Cahn–Hilliard equation*. Commun. Comput. Phys., 17(1):127–145, 2015.

[177] O. C. Zienkiewicz and J. Z. Zhu: *A simple error estimator and adaptive procedure for practical engineering analysis*. Internat. J. Numer. Methods Engrg., 24(2):337–357, 1987.