Fast Iterative Solvers for Cahn–Hilliard Problems

Dissertation

zur Erlangung des akademischen Grades

doctor rerum naturalium
(Dr. rer. nat.)

von **M. Sc. Jessica Bosch**

geb. am **15.10.1988** in Leinefelde

genehmigt durch die Fakultät für Mathematik
der Otto-von-Guericke Universität Magdeburg

Gutachter: **Dr. Martin Stoll**

Prof. Dr. Luise Blank

eingereicht am: **21.03.2016**

Verteidigung am: **21.06.2016**

Publications

Large parts of this thesis have been published in journals.

Chapter 3 is an extended version of

> [34]: J. Bosch, M. Stoll, and P. Benner, *Fast solution of Cahn–Hilliard variational inequalities using implicit time discretization and finite elements*, J. Comput. Phys., 262 (2014), pp. 38–57.

This publication has its origins in the author's Master's thesis [30]. Hence, parts of this Master's thesis appear in Chapter 3. However, more important are the extensions developed during the author's PhD studies. In this sense, Chapter 3 is an enhanced version of both, the publication [34] and the Master's thesis [30]. Moreover, the subsequent chapters build on the substance of Chapter 3 and extend the studies therein. Therefore, it plays an important part in contributing to make a whole story out of the author's PhD projects.

Chapter 4 is an extended version of

> [33]: J. Bosch and M. Stoll, *Preconditioning for vector-valued Cahn–Hilliard equations*, SIAM J. Sci. Comput., 37 (2015), pp. S216–S243.

Moreover, we were able to improve the preconditioners proposed in [33] with some fine tuning. Chapter 5 is a combination of

> [31]: J. Bosch, D. Kay, M. Stoll, and A. J. Wathen, *Fast solvers for Cahn–Hilliard inpainting*, SIAM J. Imaging Sci., 7 (2014), pp. 67–97.

> [32]: J. Bosch and M. Stoll, *A fractional inpainting model based on the vector-valued Cahn–Hilliard equation*, SIAM J. Imaging Sci., 8 (2015), pp. 2352–2382.

Note that the latter focuses on the numerical solution via Fourier spectral methods. This is not addressed in this thesis since the central theme is the development of efficient preconditioners for the iterative solution of the large and sparse linear systems that arise from classical finite element methods. Hence, Chapter 5 presents another perspective for solving the problem discussed in [32].

Acknowledgments

This thesis would not exist without the support of the professionals I met and my family.

First and foremost, I want to thank my supervisor Martin Stoll for his constant support and guidance from my Master's thesis on. He always made great efforts to get the most out of me. I admire his insight and enthusiasm which have boosted the whole project. I am extremely grateful for all discussions with him. They have always been very fruitful and inspiring. It was a great time working with him under a friendly working atmosphere.

I would also like to thank Luise Blank for her helpful advices and for a productive cooperation. Her article on “Solving the Cahn–Hilliard variational inequality with a semi-smooth Newton method“ laid the foundation of my Master's thesis and hence of this dissertation. Therefore, I am grateful that she agreed to be one of the referees for this thesis.

I am very grateful for the friendly support and financial assistance from Peter Benner. Moreover, I want to thank all my colleagues from the research groups “Numerical Linear Algebra for Dynamical Systems“ and “Computational Methods in Systems and Control Theory“ for the pleasant cooperation. In particular, I am happy about the enjoyable moments with my office colleagues Pawan Goyal, Martin Redmann, and Hamdullah Yuecel. I would also like to acknowledge the financial support from the “International Max Planck Research School (IMPRS) for Advanced Methods in Process and Systems Engineering (Magdeburg)“.

Parts of this thesis were performed while I was visiting the Oxford Centre for Collaborative Applied Mathematics (OCCAM) at the University of Oxford. I should like to express the warmest of thanks to Andy Wathen and David Kay. Both contributed greatly to my PhD studies and my lovely time in Oxford. Thank you for your assistance and effort — not only during my time in Oxford but also afterwards. My Oxford visit was supported in part by Award No. KUK-C1-013-04, made by King Abdullah University of Science and Technology (KAUST), and I would like to acknowledge this financial support as well.

I want to give thanks to several other external professionals. First, I would like to appreciate the collaboration with Christian Kahle and his patience regarding our jointly project. Then, there are Jim Varah and Chen Greif — thank you for an exciting week in Vancouver. Thanks to Arieh Iserles for a fantastic week in Cambridge. And I am grateful to Iain Duff for a fabulous week in Toulouse.

A special thanks goes to the founding members of the Student Chapter of SIAM Magdeburg — Martin Hess, Heiko Weichelt, Kristin Simon, Norman Lang, and Constantin Kwiatkowski. Together, we have led the Chapter for about two and a half years. Two and a half years full of discussions, ideas, ups and downs, organizational stuff, networking, and fun. I want to thank you for your effort and love that you have invested in our Chapter. I am also very grateful for the faculty assistance of our Chapter by Martin Stoll, Sebastian Sager, and Peter Benner. Moreover, I would like to acknowledge the financial support from SIAM, IMPRS, as well as the Otto-von-Guericke Graduate School.

A further special thanks goes to my piano teacher. She always transported me to another world — the world of harmony and freedom. Thank you for the wonderful time we had every Wednesday afternoon. You made me forget my everyday life and made me laugh. Thanks for being such a fantastic teacher.

Abschließend möchte ich mich bei den wichtigsten Menschen in meinem Leben bedanken. Das sind an erster Stelle meine Eltern. Ihr steht hinter mir und unterstützt mich, wo Ihr nur könnt. An unseren gemeinsamen Wochenenden gabt Ihr mir stets neue Energie und erfülltet mein Herz mit großer Freude. Liebe Mutti, lieber Papa — vielen Dank für Eure Liebe, Zeit, Geduld und Unterstützung. Ich möchte mich bei meinen Großeltern bedanken für Ihre Liebe, Fürsorge und Ihren Rückhalt. Ohne Muttis und Omas Essensrationen wäre meine Zeit in Magdeburg nur halb so schmackhaft gewesen. Liebe Mutti, lieber Papa, liebe Omas, lieber Opa — Diese Doktorarbeit widme ich Euch. Ich habe Euch lieb!

Abstract

In this thesis, we study efficient numerical solution techniques for various types of Cahn–Hilliard problems. Originally, the Cahn–Hilliard equation was introduced to model phase separation in two-component alloys. In praxis, often more than two components occur, and the model has been extended to the multi-component case. The solution of these two types of Cahn–Hilliard problems form the first part of this theses. The underlying energy functional includes a potential for which different types were proposed in the past. In this thesis, we consider smooth and nonsmooth potentials with a focus on the latter. Whereas the use of smooth potentials leads to a system of parabolic partial differential equations, the nonsmooth ones result in a system of variational inequalities. Due to accuracy reasons, we propose fully implicit time discretization schemes. In the smooth case, we derive criteria for the stability and uniqueness of solutions and extend the theory to the vector-valued case. In the nonsmooth setting, we interpret the system of variational inequalities as the first-order optimality system of an optimization problem for which we derive existence and uniqueness conditions. In particular, we extend the analysis to the multi-component case. In order to deal with the variational inequalities, we apply a function space-based algorithm, which combines a Moreau–Yosida regularization technique with a semismooth Newton method. We apply classical finite element methods to discretize the problems in space. The core of our approach is the solution of the arising large and sparse fully discrete systems of linear equations. An important aim of this thesis is the development of efficient practical preconditioners for the iterative solution of the linear systems. In particular, our preconditioners are tailored to the Cahn–Hilliard problems. We present block preconditioners using effective Schur complement approximations. For the smooth systems, we derive optimal preconditioners, which are proven to be robust with respect to crucial model parameters. Even for the nonsmooth systems, extensive numerical experiments show an outstanding behavior of our developed preconditioners. In combination with an adaptive mesh refinement approach, we are able to perform three-dimensional experiments in an efficient way. As another application, we apply our preconditioner to a coupled Cahn–Hilliard/Navier–Stokes system, which governs the hydrodynamics of two-phase flows. The numerical results illustrate the effectiveness of our approach.

In the second part of this thesis, we consider an application of the Cahn–Hilliard model in image processing. In particular, we consider the image inpainting problem, which can be solved by a modified Cahn–Hilliard equation. This model was proposed in the literature and is based on the two-component Cahn–Hilliard equation equipped with a smooth potential. We extend this approach in two ways: First, we apply a nonsmooth potential, which again results in the solution of variational inequalities. The numerical results show an increase of the color intensity when the nonsmooth potential is used. Second, we generalize the black-and-white Cahn–Hilliard inpainting model to gray value images. This new model is based on the multi-component Cahn–Hilliard equation. We study efficient numerical solution techniques for the scalar and in particular for the vector-valued modified Cahn–Hilliard equation using the techniques developed in first part of this thesis. An important difference to the first part is that the modified Cahn–Hillard equation as a whole is not given by a gradient flow. Especially, the model arises as a superposition of two gradient flows. We apply the convexity splitting technique, which yields under the right conditions an unconditional gradient stable time-discrete scheme. In the case of a smooth potential, we extend the proof of consistency, unconditional stability and convergence of the time-discrete scheme to the vector-valued case. In the nonsmooth case, we apply as before the Moreau–Yosida regularization technique with a semismooth Newton method. Again, the core of our approach consists in the development of efficient practical preconditioners for the iterative solution of the large and sparse linear systems that arise from classical finite element methods. For the smooth systems, we derive the conditions for optimal preconditioners. Even for the nonsmooth systems, our developed preconditioners are shown to be reliable.

Zusammenfassung

In der vorliegenden Arbeit entwickeln wir effiziente numerische Methoden zur Lösung verschiedener Probleme, die auf der Cahn–Hilliard Gleichung basieren. Ursprünglich wurde die Cahn–Hilliard Gleichung eingeführt, um die Phasentrennung in einer Metalllegierung aus zwei Spezies zu modellieren. Da in der Praxis oft mehr als zwei Spezies vorkommen, wurde das Modell auf den mehrkomponentigen Fall erweitert. Der erste Teil dieser Arbeit beschäftigt sich mit dem Lösen dieser beiden Modelle. Das zugrunde liegende Energiefunktional besteht aus einem Potential, für welches in der Vergangenheit verschiedene Arten entwickelt wurden. In dieser Arbeit betrachten wir die beiden Typen eines glatten und nichtglatten Potentials, wobei der Fokus auf dem Letztgenannten liegt. Während die Verwendung glatter Potentiale zu einem System aus parabolischen partiellen Differentialgleichungen führt, erhalten wir bei der Verwendung nichtglatter Potentiale ein System aus Variationsungleichungen. Aus Genauigkeitsgründen verwenden ein implizites Zeitdiskretisierungsschema. Im Falle des glatten Potentials leiten wir Kriterien für die Stabilität und Eindeutigkeit der zeitdiskreten Lösung her und erweitern die Theorie auf den vektorwertigen Fall. Im nichtglatten Fall fassen wir das System aus Variationsungleichungen als Lösung eines Optimierungsproblems erster Ordnung auf. Für dieses Minimierungsproblem leiten wir Kriterien für die Existenz und Eindeutigkeit einer Lösung her. Insbesondere erweitern wir die Theorie auf den vektorwertigen Fall. Die Variationsungleichungen behandeln wir mit der Moreau–Yosida Regularisierung, welche wir mit einem halbglatten Newton-Verfahren kombinieren. Zur räumlichen Diskretisierung verwenden wir die klassische Finite Elemente Methode, was zu großen, dünnbesetzten linearen Gleichungssystemen führt. Das Lösen dieser Systeme bildet den Kern unserer Verfahren. Der Hauptschwerpunkt dieser Arbeit ist die Entwicklung effizienter, praktischer Vorkonditionierer für das iterative Lösen der linearen Gleichungssysteme. Insbesondere sind unsere Löser auf die verschiedenen Cahn–Hilliard Probleme zugeschnitten. Wir erarbeiten Blockvorkonditionierer, die auf effektiven Schurkomplement-Approximationen beruhen. Im glatten Fall zeigen wir die Optimalität unserer Vorkonditionierer im dem Sinne, dass sie robust bezüglich wesentlicher Modellparameter sind. Ausgiebige numerische Experimente zeigen auch ein vielversprechendes Verhalten unserer Vorkonditionierer im nichtglatten Fall. Zusammen mit einer adaptiven Gitterverfeinerung können wir dreidimensionale Probleme effizient lösen. Als zusätzliche Anwendung wenden wir unseren nichtglatten Vorkonditionierer auf ein gekoppeltes Cahn–Hilliard/Navier–Stokes System an. Solch ein Modell wird zur

Simulation von Zwei-Phasen Strömungen verwendet. Die numerischen Resultate zeigen die Leistungsfähigkeit unserer Vorkonditionierer.

Im zweiten Teil dieser Arbeit betrachten wir eine Anwendung des Cahn–Hilliard Modells in der Bildbearbeitung. Eine modifizierte Version der Cahn–Hilliard Gleichung kann als Tool für das Inpainting verwendet werden. Das Ausgangsmodell basiert auf der skalaren Cahn–Hilliard Gleichung mit einem glatten Potential. Dieses Modell erweitern wir in zweierlei Hinsicht: Als erstes statten wir es mit einem nichtglatten Potential aus, was wiederum das Lösen von Variationsungleichungen verlangt. Die numerischen Ergebnisse zeigen, dass diese Änderung die Farbintensität in den rekonstruierten Bildern erhöht. Zweitens erweitern wir das schwarz-weiß Cahn–Hilliard Inpainting Modell auf Graubilder. Unser entwickeltes Modell basiert auf der vektorwertigen Cahn–Hilliard Gleichung. Basierend auf den Techniken, die wir im ersten Teil dieser Arbeit erarbeitet haben, entwickeln wir effiziente numerische Methoden zur Lösung der skalaren und insbesondere vektorwertigen modifizierten Cahn–Hilliard Gleichung. Ein wichtiger Unterschied zum ersten Teil ist, dass die modifizierte Cahn–Hilliard Gleichung als Ganzes nicht als ein Gradientenfluss beschrieben werden kann, sondern als Superposition zweier Gradientenflüsse. Wir wenden die konvexe Splitting Technik an, welche unter den richtigen Bedingungen ein stabiles zeitdiskretes Schema liefert. Im Falle des glatten Potentials erweitern wir den Beweis zur Konsistenz, Stabilität und Konvergenz des zeitdiskreten Systems auf den vektorwertigen Fall. Im Falle des nichtglatten Potentials wenden wir wie im ersten Teil die Moreau–Yosida Regularisierung mit einem halbglatten Newton-Verfahren an. Wie zuvor liegt der Hauptschwerpunkt unseres Verfahrens in der Entwicklung effizienter, praktischer Vorkonditionierer für das iterative Lösen der großen, dünnbesetzten diskreten linearen Gleichungssysteme, die aus der Anwendung der klassischen Finite Elemente Methode entstehen. Im glatten Fall leiten wir Bedingungen für die Optimalität unserer Vorkonditionierer her. Im nichtglatten Fall zeigen ausgiebige numerische Experimente die Effizienz unserer Vorkonditionierer.

Contents

List of Figures XV

List of Tables XIX

List of Algorithms XXIII

List of Acronyms XXV

Notation XXVII

1 Introduction 1
1.1 Phase separation in binary alloys 1
1.2 Extensions of the Cahn–Hilliard equation 7
1.3 Outline of the thesis 10

2 Mathematical Foundations 13
2.1 Functional analysis 14
2.1.1 Normed spaces and Banach spaces 14
2.1.2 Convergence in normed spaces 16
2.1.3 Inner product spaces and Hilbert spaces 17
2.1.4 L^p and Sobolev spaces 18
2.1.5 Gradient flows 23
2.1.6 Unconditional stability, consistence and convergence 24
2.2 Basic matrix theory 25
2.2.1 Matrix properties 25
2.2.2 Spectrum of matrices 26
2.2.3 Vector and matrix norms 29
2.2.4 Block and saddle point matrices 31
2.3 Iterative solution of linear systems 34
2.3.1 Krylov subspace solvers 36
2.3.2 Preconditioning 42
2.3.3 Saddle point preconditioners 44

3 Scalar Cahn–Hilliard Equations 49
3.1 Introduction 49

3.2 Derivation . . . 51
3.2.1 Smooth systems . . . 51
3.2.2 Nonsmooth systems . . . 52
3.3 Time discretization . . . 53
3.3.1 Smooth systems . . . 54
3.3.2 Nonsmooth systems . . . 57
3.4 Moreau–Yosida regularization . . . 62
3.5 Semismooth Newton method . . . 66
3.6 Finite element approximation . . . 71
3.7 Preconditioning . . . 73
3.7.1 Smooth semi-implicit systems . . . 74
3.7.2 Smooth implicit systems . . . 79
3.7.3 Nonsmooth semi-implicit systems . . . 81
3.7.4 Nonsmooth implicit systems . . . 83
3.8 Numerical results . . . 86
3.8.1 Eigenvalue plots . . . 88
3.8.2 Robustness . . . 90
3.8.3 Mesh adaptation . . . 97
3.8.4 Implicit versus semi-implicit time discretization . . . 101
3.8.5 Long-time evolution . . . 104
3.8.6 Three-dimensional dumbbell . . . 107
3.8.7 Two-phase flows . . . 108
3.9 Existing solvers . . . 112
3.10 Conclusions . . . 113

4 Vector-Valued Cahn–Hilliard Equations 115
4.1 Introduction . . . 115
4.2 Derivation . . . 118
4.2.1 Smooth systems . . . 118
4.2.2 Nonsmooth systems . . . 121
4.3 Time discretization . . . 122
4.3.1 Smooth systems . . . 122
4.3.2 Nonsmooth systems . . . 127
4.4 Moreau–Yosida regularization . . . 133
4.5 Semismooth Newton method . . . 138
4.6 Finite element approximation . . . 139
4.7 Preconditioning . . . 142
4.7.1 Smooth systems . . . 142
4.7.2 Nonsmooth systems . . . 151
4.8 Numerical Results . . . 154
4.8.1 Robustness . . . 155
4.8.2 Mesh adaptation . . . 159
4.8.3 Long-time evolution . . . 164
4.8.4 Three-dimensional example . . . 168
4.9 Existing solvers . . . 168
4.10 Conclusions and future research perspectives . . . 170

5 Modified Cahn–Hilliard Equations 173
5.1 Introduction . . . 173

5.2 Derivation . . . 176
5.2.1 Smooth systems . . . 176
5.2.2 Nonsmooth systems . . . 177
5.3 Time discretization . . . 178
5.3.1 Smooth systems . . . 179
5.3.2 Nonsmooth systems . . . 193
5.4 Semismooth Newton method . . . 195
5.5 Finite element approximation . . . 197
5.6 Preconditioning . . . 200
5.6.1 Smooth systems . . . 200
5.6.2 Scalar nonsmooth systems . . . 204
5.6.3 Vector-valued nonsmooth systems . . . 207
5.7 Numerical results . . . 211
5.7.1 Eigenvalue plots . . . 214
5.7.2 Robustness . . . 218
5.7.3 Image inpainting . . . 230
5.7.4 Three-dimensional example . . . 237
5.7.5 Comparison with existing inpainting methods . . . 238
5.8 Conclusions and future research perspectives . . . 240

6 Conclusions and Outlook **243**

Theses **245**

Bibliography **247**

Schriftliche Ehrenerklärung **257**

List of Figures

1.1 Simulation of the phase separation and coarsening process of a binary mixture. 1

1.2 Different types of the potential function ψ. 4

1.3 Interface profiles for two different types of the potential function and varying interface parameters ε: $\varepsilon = 0.2$ (⋯⋯), $\varepsilon = 0.1$ (- - -) and $\varepsilon = 0.05$ (—). 4

1.4 A three-dimensional simulation of the phase separation and coarsening process of a binary mixture. 7

1.5 Evolution of five phases using the Cahn–Hilliard model. 7

1.6 Black-and-white Cahn–Hilliard inpainting applied to a zebra image. 8

1.7 Cahn–Hilliard inpainting applied to a gray value image. 9

1.8 Simulation of a rising bubble using a coupled Cahn–Hilliard/Navier–Stokes model. 9

3.1 Illustration of the Moreau–Yosida regularized potential for different values of the penalty parameter c. 62

3.2 Spectrum of $\hat{\boldsymbol{S}}^{-1}\boldsymbol{S}$ for the smooth semi-implicit system (3.90) with the Schur complement approximation (3.92). 91

3.3 Spectrum of $\hat{\boldsymbol{S}}^{-1}\boldsymbol{S}$ for the nonsmooth semi-implicit system (3.108) with the Schur complement approximation (3.112). 92

3.4 Spectrum of $\hat{\boldsymbol{S}}^{-1}\boldsymbol{S}$ for the nonsmooth implicit system (3.117) with the Schur complement approximation (3.119). 93

3.5 Results for the solution of the smooth semi-implicit system (3.90) with the preconditioner (3.91) and the Schur complement approximation (3.92). The x-axis shows the time t and the y-axis the average number of MINRES iterations per Newton step. 95

3.6 Results for the solution of the nonsmooth semi-implicit system (3.108) with the preconditioner (3.110) and the Schur complement approximation (3.112). The x-axis shows the time t and the y-axis the average number of BiCG iterations per SSN step. 96

3.7 Results for the solution of the nonsmooth semi-implicit and implicit system. The x-axis shows the time t and the y-axis the average number of BiCG iterations per SSN step. $h_j = 2^{-j-6}$, $\varepsilon_j = 9h_j/\pi$, $\tau_1 = 2 \cdot 10^{-5}$, $\tau_2 = 3.125 \cdot 10^{-6}$, $\tau_3 = 4 \cdot 10^{-7}$, $c_{p_{\max}} = 10^{-7}$ for $j = 1, 2, 3$. 97

3.8 Results for the solution of the nonsmooth implicit system (3.117) with the preconditioner (3.118) and the Schur complement approximation (3.119). The x-axis shows the time t and the y-axis the average number of BiCG iterations per SSN step. 98

3.9 Results for the solution of the nonsmooth implicit system (3.117) with the preconditioner (3.118) and the Schur complement approximation (3.119) using adaptive meshes. The x-axis shows the time t, the left y-axis displays the average number of BiCG iterations per SSN step, and the right y-axis illustrates the number of degrees of freedom. 102

3.10 Computations with adaptive meshes: The final phase variable with the corresponding spatial mesh for the three simulations in Figure 3.9. 103

3.11 Results for the analytical example using the nonsmooth implicit and semi-implicit Cahn–Hilliard model. 103

3.12 Results for the analytical example: Solutions of the nonsmooth semi-implicit Cahn–Hilliard model with different values of the time step size τ at time $T = 1.85 \cdot 10^{-3}$. 104

3.13 Results for the analytical example: Solutions of the smooth semi-implicit Cahn–Hilliard model with different values of the time step size τ at time $T = 1.85 \cdot 10^{-3}$. 105

3.14 Long-time evolution using the smooth (upper row) and nonsmooth (lower two rows) implicit Cahn–Hilliard model. 105

3.15 Results for the long-time evolution using the smooth (left) and nonsmooth (right) implicit Cahn–Hilliard model. The x-axis shows the time t. On the left, the y-axis displays the average number of BiCG iterations per Newton step. On the right, the left y-axis displays the average number of BiCG iterations per SSN step and the right y-axis illustrates the number of degrees of freedom. Since we use a uniform mesh for the smooth model, we do not display the number of degrees of freedom on the left. 106

3.16 Evolution of a dumbbell. 107

3.17 Results for the evolution of a dumbbell: The x-axis shows the time t, the left y-axis displays the average number of BiCG iterations per SSN step, and the right y-axis illustrates the number of degrees of freedom. 108

3.18 The average number of FGMRES iterations per SSN step for the solution of a coupled two-component Cahn–Hilliard/Navier–Stokes system. Here, $m_1^{(0)}$ and $m_2^{(0)}$ denote the initial numbers of degrees of freedom. Note that the numbers of degrees of freedom during every simulation stay in the range of $m_1^{(0)}$ and $m_2^{(0)}$, respectively. 112

4.1 Results for the solution of the smooth system (4.82) with $\boldsymbol{L} = \boldsymbol{I}$, the preconditioner (4.83), and the Schur complement approximation (4.100). The x-axis shows the time t and the y-axis the average number of BiCG iterations per Newton step. 156

4.2 Results for the solution of the smooth system (4.82) with the preconditioner (4.83) and the Schur complement approximation (4.100). The x-axis shows the time t and the y-axis the average number of BiCG iterations per Newton step. Set j: $h_j = 2^{-j-6}$, $\varepsilon_j = 9\, h_j/(4\sqrt{2} \cdot \operatorname{atanh}(0.9))$, $\tau_1 = 10^{-5}$, $\tau_2 = 4 \cdot 10^{-6}$, $\tau_3 = 8 \cdot 10^{-7}$, $N = 5$ for $j = 1, 2, 3$. 157

4.3 Results for the solution of the smooth system (4.82) with $\boldsymbol{L} = \boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T$, the preconditioner (4.83), and the Schur complement approximation (4.100). The x-axis shows the time t and the y-axis the average number of BiCG iterations per Newton step. 158

4.4 Results for the solution of the nonsmooth system (4.82) with $\boldsymbol{L} = \boldsymbol{I}$, the preconditioner (4.83), and the Schur complement approximation (4.109). The x-axis shows the time t and the y-axis the average number of BiCGstab iterations per SSN step. 160

4.5 Results for the solution of the nonsmooth system (4.82) with the preconditioner (4.83) and the Schur complement approximation (4.109). The y-axis shows the average number of BiCGstab iterations per SSN step. 161

4.6 Results for the solution of the nonsmooth system (4.82) with $\boldsymbol{L} = \boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T$, the preconditioner (4.83), and the Schur complement approximation (4.109). The x-axis shows the time t and the y-axis the average number of BiCGstab iterations per SSN step. 162

4.7 Results for the solution of the nonsmooth system (4.82) with the preconditioner (4.83) and the Schur complement approximation (4.109) using adaptive meshes. The x-axis shows the time t, the left y-axis the average number of BiCGstab iterations per SSN step and the right y-axis the number of degrees of freedom. 165

4.8 Computation with adaptive meshes: Final phase variables with the corresponding spatial mesh for the three simulations in Figure 4.7. 166

4.9 Long-time evolution using the smooth (upper row) and nonsmooth (lower row) vector-valued Cahn–Hilliard model. 167

4.10 Results for the long-time evolution using the smooth (left) and nonsmooth (right) vector-valued Cahn–Hilliard model. The x-axis shows the time t and the y-axis displays the average number of BiCG (left) and BiCGstab (right) iterations per Newton/SSN step. 167

4.11 A three-dimensional simulation of the phase separation and coarsening process of a five-component mixture. 169

4.12 Results for the three-dimensional simulation of the phase separation and coarsening process of a five-component mixture: The x-axis shows the time t, the left y-axis displays the average number of BiCGstab iterations per SSN step, and the right y-axis illustrates the number of degrees of freedom. 169

5.1 Spectrum of $\hat{\boldsymbol{S}}^{-1}\boldsymbol{S}$ for the scalar smooth system (5.63) with the Schur complement approximation (5.65). 215

5.2 Spectrum of $\hat{\boldsymbol{S}}^{-1}\boldsymbol{S}$ for the scalar nonsmooth system (5.75) with the Schur complement approximation (5.77). 216

5.3 Spectrum of $\hat{\boldsymbol{S}}^{-1}\boldsymbol{S}$ for the scalar nonsmooth system (5.75) with the Schur complement approximation (5.77). 217

5.4 Spectrum of $\hat{\boldsymbol{S}}^{-1}\boldsymbol{S}$ for the scalar nonsmooth system (5.75) with the Schur complement approximation (5.77). Set j: $h_j = 2^{-j-3}$, $\varepsilon_j = h_j$, $\tau = 1$, $C_1^{(j)} = 3\varepsilon_j^{-1}$, $\omega_0^{(1)} = 10^4$, $\omega_0^{(2)} = 5 \cdot 10^4$, $\omega_0^{(3)} = 10^5$, $C_2^{(j)} = 3\omega_0^{(j)}$, $c = 10^{-7}$ for $j = 1, 2, 3$ 218

5.5 Initial images for the robustness tests of our preconditioners developed for Cahn–Hilliard inpainting. 218

5.6 Results for the solution of the scalar smooth system (5.63) with the preconditioner (5.64) and the Schur complement approximation (5.65). The x-axis shows the time t and the y-axis the number of MINRES iterations. . 220

5.7 Results for the solution of the scalar smooth system (5.63) with the preconditioner (5.64) and the Schur complement approximation (5.65). The x-axis shows the time t and the y-axis the number of MINRES iterations. . 221

5.8 Results for the solution of the scalar nonsmooth system (5.75) with the preconditioner (5.76) and the Schur complement approximation (5.77). The x-axis shows the time t and the y-axis the average number of BiCG iterations per SSN step. 222

5.9 Results for the solution of the scalar nonsmooth system (5.75) with the preconditioner (5.76) and the Schur complement approximation (5.77). The x-axis shows the time t and the y-axis the average number of BiCG iterations per SSN step. 223

5.10 Results for the solution of the vector-valued smooth system represented in (5.59). The x-axis shows the time t and the y-axis the number of MINRES iterations. 225

5.11 Results for the solution of the vector-valued smooth system represented in (5.59). The x-axis shows the time t and the y-axis the number of MINRES iterations. 226

5.12 Results for the solution of the vector-valued nonsmooth system (5.79) with the preconditioner (5.80) and the Schur complement approximation (5.83). The x-axis shows the time t and the y-axis the average number of BiCGstab iterations per SSN step. 228

5.13 Results for the solution of the vector-valued nonsmooth system (5.79) with the preconditioner (5.80) and the Schur complement approximation (5.83). The x-axis shows the time t and the y-axis the average number of BiCGstab iterations per SSN step. 229

5.14 Initial images for image inpainting. 231

5.15 Inpainted black-and-white images using the smooth and nonsmooth scalar modified Cahn–Hilliard model. 233

5.16 Inpainted gray value images using the smooth and nonsmooth vector-valued modified Cahn–Hilliard model. 234

5.17 Results for the solution of four image inpainting examples: The x-axis shows the time t. For the smooth Cahn–Hilliard inpainting models, the y-axis displays the number of MINRES iterations. For the nonsmooth Cahn–Hilliard inpainting models, the y-axis displays the average number of BiCG iterations per SSN step. 235

5.18 Results for the solution of four image inpainting examples: The x-axis shows the time t. For the smooth Cahn–Hilliard inpainting models, the y-axis displays the number of MINRES iterations. For the nonsmooth Cahn–Hilliard inpainting models, the y-axis displays the average number of BiCGstab iterations per SSN step. 236

5.19 Three-dimensional inpainting. 237

5.20 Results for the three-dimensional inpainting: The x-axis shows the time t and the y-axis the average number of BiCG iterations per SSN step. 237

5.21 Inpainted gray value image using different inpainting models. 239

List of Tables

3.1 Results for the solution of the smooth semi-implicit system (3.90) with the preconditioner (3.91) and the Schur complement approximation (3.92): The maximum and average number of Newton iterations, the maximum and average number of MINRES iterations, the average CPU time (in seconds) for MINRES, as well as the CPU time (in seconds) for the whole simulation. 99

3.2 Results for the solution of the nonsmooth semi-implicit system (3.108) with the preconditioner (3.110) and the Schur complement approximation (3.112): The maximum and average number of SSN iterations, the maximum and average number of BiCG iterations, the average CPU time (in seconds) for BiCG, and the CPU time (in seconds) for the whole simulation. 99

3.3 Results for the solution of the nonsmooth implicit system (3.117) with the preconditioner (3.118) and the Schur complement approximation (3.119): The maximum and average number of SSN iterations, the maximum and average number of BiCG iterations, the average CPU time (in seconds) for BiCG, and the CPU time (in seconds) for the whole simulation. 100

3.4 Results for the solution of the nonsmooth implicit system (3.117) with the preconditioner (3.118) and the Schur complement approximation (3.119) using adaptive meshes: The maximum and average number of SSN iterations, the maximum and average number of BiCG iterations, the average CPU time (in seconds) for BiCG, and the CPU time (in seconds) for the whole simulation. 101

3.5 Results for the long-time evolution: The maximum and average number of Newton/SSN iterations, the maximum and average number of BiCG iterations, the average CPU time (in seconds) for BiCG, and the CPU time (in seconds) for the whole simulation for the smooth and nonsmooth Cahn–Hilliard model, respectively. 106

3.6 Minimum and maximum phase values during the simulation with the smooth and nonsmooth Cahn–Hilliard model. 107

4.1 Results for the solution of the smooth system (4.82) with $\boldsymbol{L} = \boldsymbol{I}$, the preconditioner (4.83), and the Schur complement approximation (4.100): The maximum and average number of Newton iterations, the maximum and average number of BiCG iterations, the average CPU time (in seconds) for BiCG, and the CPU time (in seconds) for the whole simulation. 157

4.2 Results for the solution of the smooth system (4.82) with $\boldsymbol{L} = \boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T$, the preconditioner (4.83), and the Schur complement approximation (4.100): The maximum and average number of Newton iterations, the maximum and average number of BiCG iterations, the average CPU time (in seconds) for BiCG, and the CPU time (in seconds) for the whole simulation. 159

4.3 Results for the solution of the nonsmooth system (4.82) with $\boldsymbol{L} = \boldsymbol{I}$, the preconditioner (4.83), and the Schur complement approximation (4.109): The maximum and average number of SSN iterations, the maximum and average number of BiCGstab iterations, the average CPU time (in seconds) for BiCGstab, and the CPU time (in seconds) for the whole simulation. . . 163

4.4 Results for the solution of the nonsmooth system (4.82) with $\boldsymbol{L} = \boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T$, the preconditioner (4.83), and the Schur complement approximation (4.109): The maximum and average number of SSN iterations, the maximum and average number of BiCGstab iterations, the average CPU time (in seconds) for BiCGstab, and the CPU time (in seconds) for the whole simulation. 164

4.5 Results for the solution of the nonsmooth system (4.82) with the preconditioner (4.83) and the Schur complement approximation (4.109) using adaptive meshes: The maximum and average number of SSN iterations, the maximum and average number of BiCGstab iterations, the average CPU time (in seconds) for BiCGstab, and the CPU time (in seconds) for the whole simulation. 166

4.6 Results for the long-time evolution: The maximum and average number of Newton/SSN iterations, the maximum and average number of BiCG/BiCGstab iterations, as well as the average CPU time (in seconds) for BiCG/BiCGstab and the CPU time (in seconds) for the whole simulation for the smooth and nonsmooth Cahn–Hilliard model, respectively. . . 166

4.7 Minimum and maximum phase values during the simulation with the smooth and nonsmooth vector-valued Cahn–Hilliard model. 168

5.1 Results for the solution of the scalar smooth system (5.63) with the preconditioner (5.64) and the Schur complement approximation (5.65): The maximum and average number of MINRES iterations, the average CPU time (in seconds) for MINRES, and the CPU time (in seconds) for the whole simulation. 219

5.2 Results for the solution of the scalar nonsmooth system (5.75) with the preconditioner (5.76) and the Schur complement approximation (5.77): The maximum and average number of SSN iterations, the maximum and average number of BiCG iterations, the average CPU time (in seconds) for BiCG, and the CPU time (in seconds) for the whole simulation. 224

5.3 Results for the solution of the vector-valued smooth system represented in (5.59): The maximum and average number of MINRES iterations, the average CPU time (in seconds) for MINRES, and the CPU time (in seconds) for the whole simulation. 227

5.4 Results for the solution of the vector-valued nonsmooth system (5.79) with the preconditioner (5.80) and the Schur complement approximation (5.83): The maximum and average number of SSN iterations, the maximum and average number of BiCGstab iterations, the average CPU time (in seconds) for BiCGstab, and the CPU time (in seconds) for the whole simulation. . . . 230

5.5 Results for the solution of image inpainting examples: The maximum and average number of SSN iterations, the maximum and average number of MINRES/BiCG/BiCGstab iterations, the average CPU time (in seconds) for MINRES/BiCG/BiCGstab, and the CPU time (in seconds) for the whole simulation. 232

5.6 Results for the solution of image inpainting examples: The PSNR value as well as the minimum (min) and maximum (max) phase variable value. 232

5.7 Performance of different inpainting models: The total number of iterations (iter), CPU time (in seconds) for the whole simulation, average CPU time (in seconds) per time step ($\overline{\text{CPU}}$), PSNR value, as well as the minimum (min) and maximum (max) pixel value of the final image. . . . 239

List of Algorithms

2.1 The biconjugate gradient method 42
2.2 The preconditioned biconjugate gradient method 43

3.1 The numerical solution of the nonsmooth implicit time-discrete Cahn–Hilliard problem via an SSN method combined with a Moreau–Yosida regularization technique on a uniform mesh. 87
3.2 The numerical solution of the nonsmooth implicit time-discrete Cahn–Hilliard problem via an SSN method combined with a Moreau–Yosida regularization technique on an adaptive mesh. 89

5.1 The numerical solution of the nonsmooth (vector-valued) modified Cahn–Hilliard problem via an SSN method combined with a Moreau–Yosida regularization technique on a uniform mesh. 213

List of Acronyms

AMG	algebraic multigrid method	44
BiCG	biconjugate gradient method	41
BiCGstab	biconjugate gradient stabilized method	41
CG	conjugate gradient method	38
CPU	central processing unit	94
FEM	finite element method	5
FFT	fast Fourier transform	150
FGMRES	flexible generalized minimum residual method	111
GMRES	generalized minimum residual method	41
MG	multigrid method	43
MINRES	minimum residual method	40
PDE	partial differential equation	5
PSNR	peak signal-to-noise ratio	232
SSN	semismooth Newton	10

Notation

Ω	spatial domain
a.e.	almost everywhere
$C(\Omega)$	space of continuous functions on Ω
$C^k(\Omega)$	space of k-times continuously differentiable functions on Ω
$\mathcal{L}(\mathcal{X},\mathcal{Y})$	vector space of all bounded linear operators from a normed space $\mathcal{X}$ into a normed space $\mathcal{Y}$
$\mathcal{X}^*$	dual space of a normed space $\mathcal{X}$
$\langle\cdot,\cdot\rangle_{\mathcal{X}^*\times\mathcal{X}}$	duality pairing between a normed space $\mathcal{X}$ and its dual space $\mathcal{X}^*$
$L^p(\Omega)$	set of measurable and p-integrable functions on Ω
$L^p(\Omega)^N$	L^p space of vector-valued functions
$(\cdot,\cdot)$	$L^p(\Omega)$- and $L^p(\Omega)^N$-inner product
$\lVert\cdot\rVert$	$L^p(\Omega)$- and $L^p(\Omega)^N$-norm
$H^k(\Omega)$	Sobolev spaces
$H^k(\Omega)^N$	Sobolev spaces of vector-valued functions
$(\cdot,\cdot)_1$	$H^1(\Omega)$- and $H^1(\Omega)^N$-inner product
$\lVert\cdot\rVert_1$	$H^1(\Omega)$- and $H^1(\Omega)^N$-norm
$\lVert\cdot\rVert_*$	$H^1(\Omega)^*$-norm
$\langle\cdot,\cdot\rangle$	duality pairing between $H^1(\Omega)^*$ and $H^1(\Omega)$
$H^{-1}(\Omega)$	$:= \left\{u^* \in H^1(\Omega)^* \colon \langle u^*, 1\rangle = 0\right\}$
$(v_1, v_2)_{-1}$	$:= \int_\Omega \nabla(-\Delta)^{-1}v_1 \cdot \nabla(-\Delta)^{-1}v_2 \,\mathrm{d}\mathbf{x}$
$x_k \to x$	(strong) convergence
$x_k \rightharpoonup x$	weak convergence
$\nabla\mathbf{u}$	$:= [\nabla u_1, \dots, \nabla u_N]^T$ for $\mathbf{u} = [u_1, \dots, u_N]^T \in H^1(\Omega)^N$
$\Delta\mathbf{u}$	$:= [\Delta u_1, \dots, \Delta u_N]^T$ for $\mathbf{u} = [u_1, \dots, u_N]^T \in H^2(\Omega)^N$
$\mathbb{N}$	set of natural numbers
$\mathbb{R}, \mathbb{C}$	field of real and complex numbers
$\bar{S}$	closure of the set S
$\mathbb{R}^{n\times m}$	vector space of real $n \times m$ matrices
$\mathbb{R}^n$	$= \mathbb{R}^{n\times 1}$
$\boldsymbol{I}$	identity matrix
$\mathbf{1}$	column vector where each entry is equal to one
$\mathbf{0}$	zero matrix or zero vector
$\boldsymbol{e}_i$	column vector where the ith entry is equal to one and all remaining entries are zero

$[\boldsymbol{A}]_{ij}$	(i, j) entry of a matrix $\boldsymbol{A}$
$\boldsymbol{A}^T$	transpose of a matrix $\boldsymbol{A}$
$\boldsymbol{A}^{-1}$	inverse of a nonsingular matrix
$\det(\boldsymbol{A})$	determinant of a matrix $\boldsymbol{A}$
$\ker(\boldsymbol{A})$	kernel of a matrix $\boldsymbol{A}$
$\sigma(\boldsymbol{A})$	spectrum of a matrix $\boldsymbol{A}$
$\rho(\boldsymbol{A})$	spectral radius of a matrix $\boldsymbol{A}$
$\lambda_{\max}(\boldsymbol{A})$	maximum eigenvalue of a symmetric matrix $\boldsymbol{A}$
$\lambda_{\min}(\boldsymbol{A})$	minimum eigenvalue of a symmetric matrix $\boldsymbol{A}$
$\kappa(\boldsymbol{A})$	condition number of a matrix $\boldsymbol{A}$
$\|\boldsymbol{u}\|_p, \|\boldsymbol{A}\|_p$	p-norm of a vector $\boldsymbol{u}$ and a matrix $\boldsymbol{A}$
$\|\boldsymbol{u}\|, \|\boldsymbol{A}\|$	2-norm of a vector $\boldsymbol{u}$ and a matrix $\boldsymbol{A}$
$\mathcal{K}_l(\boldsymbol{A}, \boldsymbol{b})$	Krylov subspace spanned by $\{\boldsymbol{b}, \boldsymbol{A}\boldsymbol{b}, \ldots, \boldsymbol{A}^{l-1}\boldsymbol{b}\}$
$\otimes$	Kronecker product

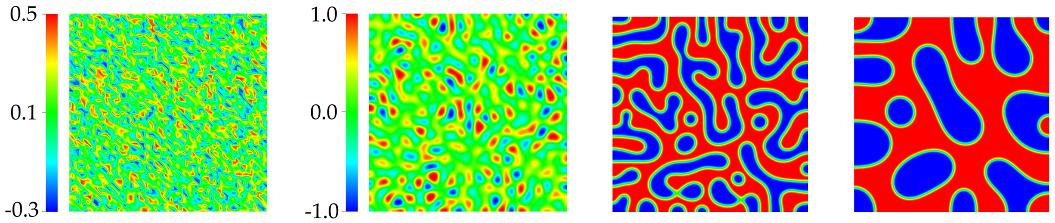
Chapter 1

Introduction

1.1 Phase separation in binary alloys

In 1958, Cahn and Hilliard [44] proposed a model which describes the phenomenon of phase separation in binary alloys. In particular, they considered a molten iron-nickel alloy and aimed to describe how it separates into its two components. Such a separation occurs if the alloy is rapidly quenched below a critical temperature. Very soon, small formations of pure iron and nickel appear. As time goes by, these formations slowly coarsen into larger ones. This coarsening process is also called aging. In principle, spatially separated areas with different physical properties develop. It is important for material scientists to understand the dynamics of the occurring processes, how fast they are, and how they influence the material properties. As laboratory experiments are time- and cost-consuming, sophisticated computer simulations are desirable.

Figure 1.1 illustrates a simulation of such a separation process. It shows the concentrations of the two components at four different times. Let us denote the two species,



(a) Mixture of two components.

(b) Phase separation starts very quickly.

(c) The two pure components have emerged to small formations in a short time.

(d) As time goes by, the formations slowly coarsen into larger ones.

Figure 1.1: Simulation of the phase separation and coarsening process of a binary mixture.

also referred to as phases, by A and B. We characterize them by their concentrations $c_A(\mathbf{x},t) \in [0,1]$ and $c_B(\mathbf{x},t) = 1 - c_A(\mathbf{x},t) \in [0,1]$. Here, $\mathbf{x}$ denotes a spatial point and t refers to the time. During the evolution, three different types of spatial regions can be characterized:

1. Regions of pure iron, i.e., the pure phase A. This case is represented by $c_A = 1$ and $c_B = 0$.

2. Regions of pure nickel, i.e., the pure phase B. This case is represented by $c_B = 1$ and $c_A = 0$.

3. Mixed regions of iron and nickel. This case is represented by $c_A \in (0,1)$ and $c_B \in (0,1)$ with $c_A + c_B = 1$.

The two characteristics c_A and c_B can be combined into one variable. This gives the phase variable[1] $u = u(\mathbf{x},t)$, which describes the difference of the local concentrations, e.g., $u(\mathbf{x},t) = c_B(\mathbf{x},t) - c_A(\mathbf{x},t) = 1 - 2c_A(\mathbf{x},t) \in [-1,1]$. Hence, if $u(\mathbf{x},t) = -1$, then only phase A (the pure phase A) is present at point $\mathbf{x}$ at time t. The case $u(\mathbf{x},t) = 1$ means only phase B (the pure phase B) exists at point $\mathbf{x}$ at time t. Values of u between -1 and 1 represent mixed regions. For the simulation in Figure 1.1, we use the square $[-1,1]^2$ as the spatial domain. The two scales appearing in this figure display the variable $u(\mathbf{x},t)$. The scale in Figure 1.1(a) only belongs to this picture. The second scale in Figure 1.1(b) holds for the Figures 1.1(b)–1.1(d). Figure 1.1(a) represents the initial state, which is built by a uniform mixture. The scale indicates that no pure phase is present at that time. Only five time steps later, phase separation occurs as can be seen in Figure 1.1(b). We observe the formation of small bubbles of pure concentrations. From now on, the pure phase A is represented by the color blue and the pure phase B by the color red. Then in a first stage, the bubbles quickly coarsen into larger ones. Figure 1.1(c) displays the system after 50 time steps. In a second stage, this process happens more slowly. Figure 1.1(d) is taken after 500 time steps.

The above simulation was done using the Cahn–Hilliard model. The Cahn–Hilliard model belongs to the class of phase field models, which are used to solve interfacial problems. The role of the interface can be understood by having a second look at Figure 1.1(c) or 1.1(d). The interface is the small boundary layer that separates the pure phases A and B from each other. Hence, it acts as a diffuse phase transition. Its width is present as a model parameter, the interfacial parameter $\varepsilon > 0$ introduced below in (1.1), and is aimed to be as small as possible. There is also the limit case $\varepsilon \downarrow 0$, which gives the sharp interface model [74, 72, 131, 42].

The theory of Cahn and Hilliard [44] is based on the Ginzburg–Landau energy

$$\mathcal{E}(u) = \int_{\Omega} \frac{\varepsilon}{2}|\nabla u|^2 + \frac{1}{\varepsilon}\psi(u)\ \mathrm{d}\mathbf{x}. \tag{1.1}$$

Here, Ω denotes the spatial domain, which contains the molten alloy. An equilibrium profile of our considered mixture minimizes the Ginzburg–Landau energy (1.1)

[1] The phase variable u is also known as the order parameter, which describes the state of the considered system at any given time.

subject to the mass conservation[2]

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_\Omega u \,\mathrm{d}\mathbf{x} = 0.$$

The parameter $\varepsilon > 0$ is proportional to the thickness of the interfacial region as mentioned above. The first part of (1.1) is large whenever u changes rapidly. Hence, its minimization gives rise to the interfacial area. The potential function ψ in (1.1) gives rise to phase separation. It has two distinct minima, one for each of the two pure phases A and B. Hence, its minimization penalizes values away from the pure phases. Different types of potential functions have been considered in the literature [27]. To explain this, we have to expand on the phase separation process. Such a separation occurs if a high-temperature mixture, existing in a state of isothermal equilibrium, is rapidly quenched to a uniform temperature θ below a critical temperature θ_c. Depending on the temperature reduction, various types of the potential function ψ have been introduced. Originally, Cahn and Hilliard [44] suggested a logarithmic potential of the form

$$\psi_{\log}(u) = \frac{\theta}{2}\left[(1+u)\ln\left(\frac{1+u}{2}\right) + (1-u)\ln\left(\frac{1-u}{2}\right)\right] + \frac{\theta_c}{2}\left(1-u^2\right) \tag{1.2}$$

for $\theta < \theta_c$; see also [43]. According to [27], the minima of $\psi_{\log}$ are $\pm\beta$, where β is the positive root of

$$2\frac{\theta_c}{\theta} = \frac{1}{\beta}\ln\left(\frac{1+\beta}{1-\beta}\right).$$

Figure 1.2(a) illustrates the logarithmic potential function (1.2) for $\theta_c = 1$ and different values of the temperature θ. For a numerical analysis of the Cahn–Hilliard equation with a logarithmic potential, we refer to [50].

When the quench $\theta < \theta_c$ is additionally shallow, i.e., θ is close to θ_c, ψ is usually approximated by a quartic polynomial like

$$\psi_{\mathrm{pol}}(u) = \frac{1}{4}(u^2-\beta^2)^2, \tag{1.3}$$

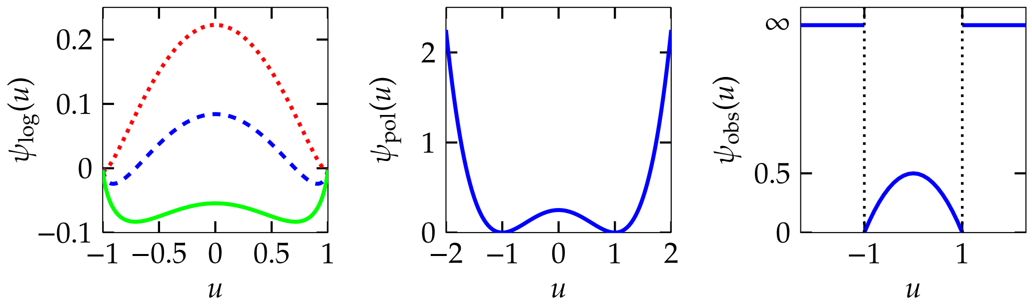
see, e.g., [127, 57]. This type of function is called double-well potential. The minima of ψ_{pol} are $\pm\beta$. Figure 1.2(b) shows the polynomial potential function (1.3) for $\beta = 1$. In contrast to the logarithmic potential, the polynomial one allows violations of $u \in [-\beta, \beta]$. Using the double-well potential, the interfacial equilibrium profile in one space dimension can be described by

$$\hat{u}(x) = \beta\tanh\left(\frac{x}{\sqrt{2}\varepsilon}\right),$$

see, e.g., [56, Section 7.9] or the note in [6, p. 374]. This profile is shown in Figure 1.3(a) for different values of the interface parameter ε. Let us describe the interface thickness as the distance between x_1 and x_2 with $u(x_1) = -0.95\,\beta$ and $u(x_2) = 0.95\,\beta$. Then, we can express the equilibrium thickness via ε by

$$0.95 = \tanh\left(\frac{x_2}{\sqrt{2}\varepsilon}\right),$$

[2] Note that the mass conservation is not a general characteristic of phase field models. Instead, it is an element of the Cahn–Hilliard model.



(a) The logarithmic potential $\psi_{\log}$ for $\theta_c = 1$ and varying temperatures θ: $\theta = 0.4\,\theta_c$ (⋯⋯), $\theta = 0.6\,\theta_c$ (- - -) and $\theta = 0.8\,\theta_c$ (——).

(b) The double-well potential ψ_{pol} for $\beta = 1$.

(c) The double-obstacle potential ψ_{obs}.

Figure 1.2: Different types of the potential function ψ.

which is equivalent to

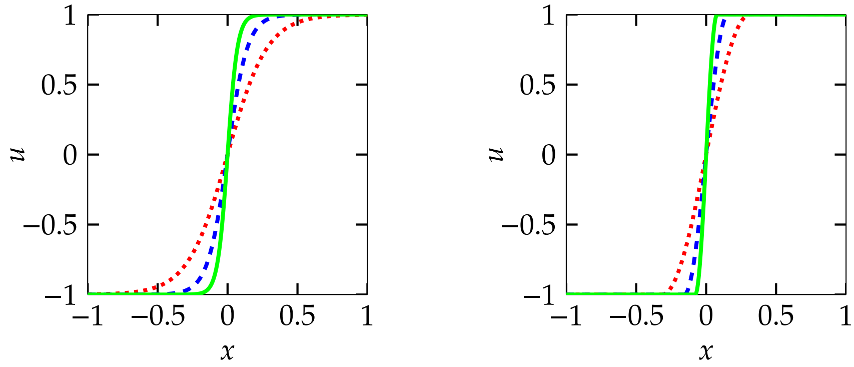
$$x_2 = \sqrt{2}\varepsilon \cdot \operatorname{atanh}(0.95).$$

Similar, we obtain

$$x_1 = -\sqrt{2}\varepsilon \cdot \operatorname{atanh}(0.95).$$

Hence, the equilibrium interfacial thickness is given by $2\sqrt{2}\varepsilon \cdot \operatorname{atanh}(0.95)$ using a polynomial potential. Experiments show that it is essential to ensure that at least eight spatial mesh points lie on the interface in order to avoid mesh effects; see also [29]. Hence, numerically we want to have eight or nine grid points across the interface transition. If we denote by h the spatial mesh size across the interface, this leads to the condition

$$h \leq \frac{2\sqrt{2}\varepsilon \cdot \operatorname{atanh}(0.95)}{9} \approx 0.5757\,\varepsilon.$$



(a) The double-well potential ψ_{pol}.

(b) The double-obstacle potential ψ_{obs}.

Figure 1.3: Interface profiles for two different types of the potential function and varying interface parameters ε: $\varepsilon = 0.2$ (⋯⋯), $\varepsilon = 0.1$ (- - -) and $\varepsilon = 0.05$ (——).

Up to here, we have presented logarithmic potential functions for the case $\theta < \theta_c$ and polynomial potential functions under the additional condition $\theta \approx \theta_c$. For the deep quench limit $\theta \to 0$, i.e., a very rapid cooling of the mixture resulting in temperatures $\theta \ll \theta_c$, Oono and Puri [128] introduced the nonsmooth double-obstacle potential

$$\psi_{\text{obs}}(u) = \begin{cases} \frac{1}{2}(1-u^2) & |u| \leq 1, \\ \infty & |u| > 1. \end{cases} \tag{1.4}$$

We refer to [27, 28] for a mathematical and numerical analysis for this setting. The minima of ψ_{obs} are attained at ±1. Figure 1.2(c) visualizes the obstacle potential function (1.4). This type of function admits a sharper interface than the polynomial potential. As stated in [42, p. 17], the equilibrium interface thickness is proportional to $\varepsilon\pi$, i.e., the interfacial equilibrium profile in one space dimension can be described by

$$\hat{u}(x) \approx \sin\left(\frac{x}{\varepsilon}\right),$$

where $x \in [-0.5\varepsilon\pi, 0.5\varepsilon\pi]$. Again, numerically we want to have eight or nine grid points across the interface transition. This leads to the condition

$$h \leq \frac{\varepsilon\pi}{9} \approx 0.3491\,\varepsilon.$$

Figure 1.3(b) displays the equilibrium interface profile for the double-obstacle potential setting.

During the rest of this thesis, we will always consider the case $\beta = 1$ when referring to the polynomial potential in (1.3). Moreover, in this thesis we consider potential functions of polynomial and obstacle type with a main focus on the latter.[3] Besides the different interfacial profiles, these two variants of the potential function exhibit another distinctive feature. The differentiable potential leads to a parabolic partial differential equation (PDE). In this thesis, we will name this formulation the *smooth system*. In contrast, the obstacle potential results in variational inequalities, which are much harder to solve. We will call this formulation the *nonsmooth system*.

Only in very simple cases it is possible to find exact solutions of the Cahn–Hilliard model. Hence, numerical techniques have to be developed for finding approximate solutions. The basic idea is to discretize the continuous problem to obtain a discrete problem with only a finite number of unknowns. In this thesis, we are focused on the classical finite element method (FEM) [144]. Moreover, due to the nonlinearity, linearization techniques have to be applied in order to end up with linear equations. To obtain the desired accuracy, the resulting discrete problems consist of huge linear systems of the general form

$$\boldsymbol{\mathcal{A}}\boldsymbol{z} = \boldsymbol{b}. \tag{1.5}$$

Here, $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{2m\times 2m}$ is the given coefficient matrix, $\boldsymbol{z} \in \mathbb{R}^{2m}$ is the unknown solution vector, and $\boldsymbol{b} \in \mathbb{R}^{2m}$ is the given right-hand side vector. By the term 'huge' we mean sizes of m in the range of millions. Moreover, the use of FEM results in a sparse matrix $\boldsymbol{\mathcal{A}}$. The coefficient matrix $\boldsymbol{\mathcal{A}}$ has a 2 × 2 block structure of the form

$$\boldsymbol{\mathcal{A}} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{B} & -\boldsymbol{C} \end{bmatrix}, \tag{1.6}$$

[3] Logarithmic potentials may be discussed in future research.

where every block itself is a huge matrix. In particular, $\mathcal{A}$ is a saddle point matrix. The unknown solution vector z contains the discrete version of the phase variable u.

Iterative methods form a general procedure for solving the system (1.5) approximately. They aim to generate a sequence of vectors $\{z^{(l)}\}_{l\in\mathbb{N}}$, which converges to the solution z. Key points regarding the efficiency of iterative methods include the rate of convergence, the computational costs per iteration, and storage requirements. The convergence is often dependent on system parameters. Various parameters are involved in $\mathcal{A}$. For the present Cahn–Hilliard problem, these are the spatial mesh size, the time step size, the interfacial parameter, and, for our treatment of the obstacle potential, a penalty parameter. Broadly speaking, the penalty parameter forces the phase variable u to stay in the desired concentration range $[-1, 1]$ as described above. The higher the penalization is, the more accurate is the fulfillment of the concentration range condition.[4] In contrast, the remaining parameters have to be chosen much smaller. Typically, for finer mesh discretizations, i.e., for larger values of m, not only the costs per iteration increase but also the the number of iterations until convergence. It would be ideal to prevent the growth of iteration numbers when the mesh is refined. This property is called robustness with respect to the mesh size. And it would be even more ideal if we have the robustness also with respect to other relevant parameters. Hence, in order to improve the performance of the iterative method we have to incorporate an accelerator, also referred to as a *preconditioner*. The basic idea is to construct a matrix $\mathcal{P} \in \mathbb{R}^{2m\times 2m}$ and replace (1.5) by

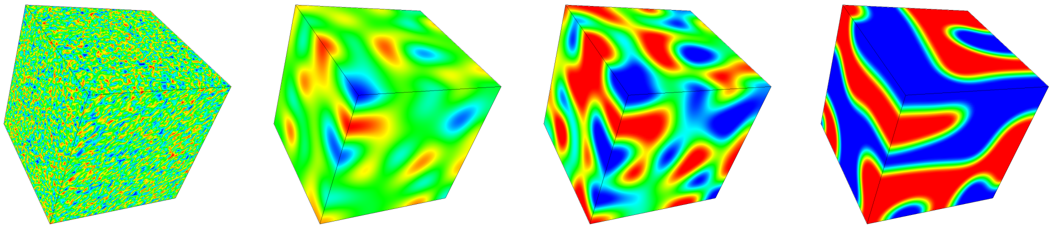
$$\mathcal{P}^{-1}\mathcal{A}z = \mathcal{P}^{-1}b. \tag{1.7}$$

Both systems, (1.5) and (1.7), have the same solution. In order to be effective, $\mathcal{P}$ should be designed such that it approximates $\mathcal{A}$ and is cheap to apply. Theoretical optimal preconditioners, which capture the block structure of $\mathcal{A}$, have been proposed (see, e.g., [124, 97]) and will be recalled in Chapter 2.3.3. In fact, the authors have proven to yield convergence of an iterative method after only a small number of iterations. However, in order to make these preconditioners also efficient in praxis, clever approximations have to be developed. Efficient approximations typically result in outstanding convergence performances.

The development of powerful preconditioners allows us to solve huge linear systems in a reasonable time. In particular, preconditioning is necessary to make the computation of very large three-dimensional problems feasible. A simulation of the phase separation and coarsening process of a binary mixture in the unit cube $[0, 1]^3$ is shown in Figure 1.4. The design, implementation, and numerical analysis of preconditioners tailored to Cahn–Hilliard problems are the focus of this thesis. In particular, the preconditioners are aimed to be robust with respect to parameter changes.

The Cahn–Hilliard model has also been used in different other contexts, e.g., in materials science [126, 72], image processing [54], or chemistry [152]. In the following, we give a flavor of the Cahn–Hilliard applications that are discussed in this thesis.

[4] Note that the smooth Cahn–Hilliard formulation does not need such a penalization. We will also discuss them in this thesis.



(a) Mixture of two components.

(b) Phase separation starts very quickly.

(c) The two pure components have emerged to small formations in a short time.

(d) As time goes by, formations slowly coarsen into larger ones.

Figure 1.4: A three-dimensional simulation of the phase separation and coarsening process of a binary mixture.

1.2 Extensions of the Cahn–Hilliard equation

Besides the phase separation and coarsening process in binary alloys, we study two more applications of the Cahn–Hilliard equation in this thesis. These are

- the phase separation and coarsening process in multi-component alloys and
- image inpainting.

All three topics form individual chapters in this thesis. Moreover, we have applied our solver from Chapter 1 to a coupled two-component Cahn–Hilliard/Navier–Stokes system, which models the hydrodynamics of two-phase flows. This is another application of scalar Cahn–Hilliard equations. We will briefly present this problem and some results in Chapter 3.8.7.

Multicomponent systems

The major part of this thesis deals with the treatment of multi-component Cahn–Hilliard problems. Instead of two phases as we have discussed so far, we consider $N > 2$ components now. Figure 1.5 shows a simulation of the phase separation and coarsening process of a five-component alloy. The numerical solution via FEM and preconditioning involves two additional difficulties. First, the coefficient matrix $\boldsymbol{\mathcal{A}}$ has the form as in (1.6), but now every of the four blocks has an additional $N \times N$ block structure. Hence, the problem size has increased manifold. Second, strong couplings between the N phases appear in $\boldsymbol{\mathcal{A}}$.

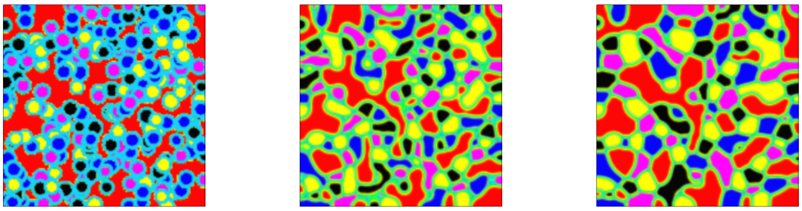
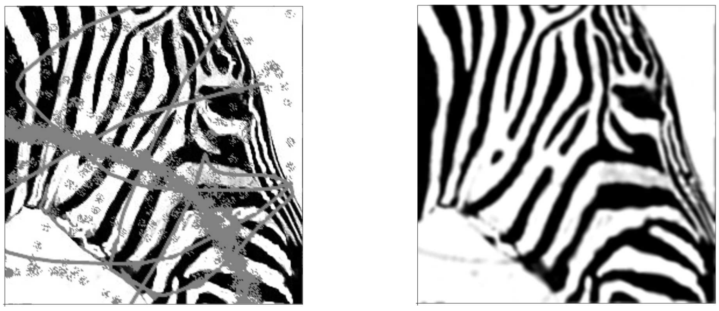


Figure 1.5: Evolution of five phases using the Cahn–Hilliard model.

Image inpainting

Modified versions of the Cahn–Hilliard equation can be used for the solution of problems in imaging science. One example is inpainting [17, 45]. Inpainting is the art of modifying parts of an image such that the resulting changes are not easily detectable by an ordinary observer. Applications include the restoration of damaged paintings and photographs, the replacement of selected objects, or the reduction of artifacts in medical images.

Black-and-white image inpainting can be done via a modification of the two-component Cahn–Hilliard model from Section 1.1. Figure 1.6(a) shows an example image, where the inpainting domain is marked by the gray color (which could be seen as cracks or scratches). Figure 1.6(b) illustrates the inpainted reconstruction using a modified Cahn–Hilliard equation.



(a) Destroyed image.

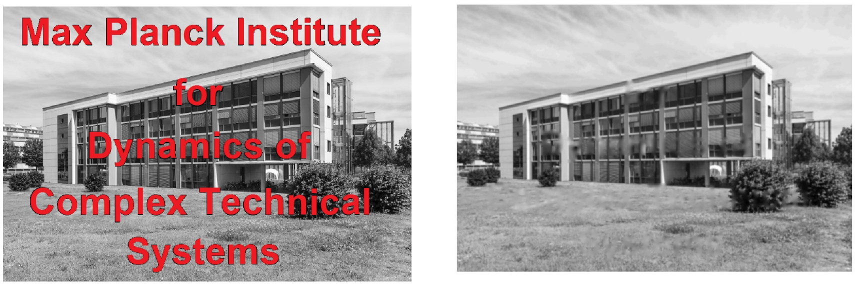
(b) Reconstructed image.

Figure 1.6: Black-and-white Cahn–Hilliard inpainting applied to a zebra image.

In Section 1.1, we have explained the underlying system in general terms of phases. Remember that we have called u the phase variable, which describes the concentration of two phases A and B. Now, we want to explain the underlying system in terms of image inpainting. In this setting, the two phases A and B represent the colors black and white. Their interface consists of gray values and forms a smooth transition. Basically, we can imagine u as a black-and-white image that evolves in time.

Imagine we have a given damaged black-and-white image as in Figure 1.6(a). In the following, we will label it by f. The parts that are going to be modified are denoted by the inpainting domain. These parts are often called missing or damaged regions since the observer (usually) does not know the original image. The task is to reconstruct the image f in this damaged region in an undetectable way. The reconstructed image is represented by our phase variable u. Without any modification of the original Cahn–Hilliard equation, the resulting image u would have nothing in common with the original image. But the reconstructed image should be (almost) identical with the given image f in the undamaged parts. This means, information about f as well as the location of the inpainting domain have to be added to the Cahn–Hilliard model.

Bertozzi, Esedoḡlu, and Gillette [19] introduced the Cahn–Hilliard inpainting approach for binary images using a smooth potential. Our main contributions are threefold. First, we extend this approach to nonsmooth potentials. Our second input is the development of effective preconditioners for both settings, the smooth and nonsmooth one. The arising coefficient matrix $\mathcal{A}$ is of the same size as the one we deal with in Section 1.1. However, new parameters appear. Our third contribution combines the multi-component framework with inpainting. We develop an inpainting model based on the multi-component Cahn–Hilliard model. This approach generalizes Bertozzi et al's binary Cahn–Hilliard inpainting model to gray value images. Figure 1.7 shows an example.



(a) Destroyed image.

(b) Reconstructed image.

Figure 1.7: Cahn–Hilliard inpainting applied to a gray value image.

When a smooth potential is used, we are aware that Fourier spectral methods provide a powerful solver for Cahn–Hilliard inpainting on simple domains; see, e.g. [32]. However, the computation of missing information on arbitrary domains is a motivation for the use of FEM and preconditioning.

Two-phase flows

Our last project arose from a collaboration with the University of Hamburg, Germany. In [73], Garcke, Hinze, and Kahle developed discretization techniques for a coupled Cahn–Hilliard/Navier–Stokes system. Such a model is used for the numerical simulation of two-phase flows. Figure 1.8 illustrates a simulation of a rising bubble. The authors were interested in three-dimensional experiments, which are not possible without having efficient iterative solution techniques. Hence, our contribution concerns the iterative solution of the arising linear systems. This is based on the preconditioning techniques we have developed for the Cahn–Hilliard model in Section 1.1 together with the methods that have been developed for the Navier–Stokes equations [99, 63].

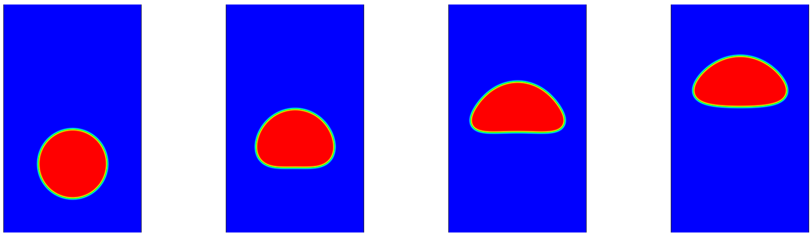


Figure 1.8: Simulation of a rising bubble using a coupled Cahn–Hilliard/Navier–Stokes model.

1.3 Outline of the thesis

We begin with a review of mathematical terms and concepts that are relevant for this thesis in Chapter 2. It consists of of two parts — a functional analysis and a numerical linear algebra part. The former briefly introduces normed spaces, Hilbert spaces, and Sobolev spaces. Moreover, we introduce the concept of gradient flows since the Cahn–Hilliard equation is one example. The first part ends with a review of unconditional stability, consistence, and convergence for PDEs, which do not arise from a gradient flow. We will need them in Chapter 5, where we utilize an unconditional time discretization scheme to a modified Cahn–Hilliard equation. Each of the Chapters 3–5 starts with the problem formulation and a discussion in function space. That is where basic functional analysis background is needed. The numerical linear algebra part of Chapter 2 starts with setting up the basics in matrix theory and notations used throughout this thesis. Then, we give an overview of iterative solution techniques for systems of linear equations. In each of the Chapters 3–5, we end up with systems of linear equations after proper discretizations. Their solution is the focus of this thesis and this is where the numerical linear algebra part is required. Further, we present the concept of preconditioning, which is needed to make iterative methods powerful.

Chapter 3 is devoted to the numerical solution of scalar Cahn–Hilliard equations. We focus on the variational inequality formulation but also study the simpler PDE version that is based on the smooth potential. Despite the inherent time step restriction, we utilize an implicit time discretization scheme. This is due to accuracy reasons, which will be discussed in the numerical examples. We prove a uniqueness and stability result for the time-discrete scheme in the smooth setting. The nonlinear time-discrete systems are solved via standard Newton methods. The time-discrete Cahn–Hilliard variational inequality is formulated as an optimal control problem with pointwise constraints on the control. Again, we prove a uniqueness result for the time-discrete problem. By applying a semismooth Newton (SSN) method combined with a Moreau–Yosida regularization technique for handling the control constraints, we show superlinear convergence in function space. For both settings, smooth and nonsmooth, we use FEM for the discretization in space. At the heart of our approach lies the solution of large and sparse linear systems for which we propose the use of preconditioned Krylov subspace solvers using effective Schur complement approximations.

Chapter 4 proceeds to the numerical solution of vector-valued Cahn–Hilliard equations. These generalize the two-component problems discussed in Chapter 3 to multi-component ones. As in Chapter 3, we discuss strategies for the two model formulations, which are on the one hand a system of PDEs based on the smooth potential, and on the other hand a system of variational inequalities based on the nonsmooth potential. Regarding the smooth setting, we extend the proof of the energy stability and uniqueness of the solution of the time-discrete scheme from the two-component to the multi-component case. The time-discrete system of Cahn–Hilliard variational inequalities is interpreted as the first-order optimality system of an optimization problem for which we have derived existence and uniqueness conditions. Motivated by Chapter 3, we apply an SSN method combined with a Moreau–Yosida regularization technique for handling the pointwise constraints. For both settings, smooth and nonsmooth, we use FEM for the discretization in space. Again, at the heart of our approach lies the solution of large and sparse linear systems. We develop effective preconditioners using efficient and cheap Schur complement approximations that are tailored to the vector-valued Cahn–Hilliard equations.

Chapter 5 studies the numerical solution of scalar and vector-valued modified Cahn–Hilliard equations. These extend the Cahn–Hilliard problems discussed in Chapter 3 and 4 to a tool in image inpainting. As before, we discuss strategies for the two model formulations based on the smooth and nonsmooth potential. An important difference to the previous two chapters is that the modified Cahn–Hillard equation as a whole is not given by a gradient flow. Especially, the model arises as a superposition of two gradient flows. We apply the convexity splitting technique, which under the right conditions yields an unconditional gradient stable time-discrete scheme. Regarding the smooth setting, we extend the proof of consistency, unconditional stability, and convergence of the time-discrete scheme from the two-component to the multi-component case. Concerning the nonsmooth framework, we follow the previous two chapters and apply an SSN method combined with a Moreau–Yosida regularization technique. For both settings, smooth and nonsmooth, we use FEM for the discretization in space. At the heart of our approach lies the solution of large and sparse linear systems. We develop effective preconditioners using efficient and cheap Schur complement approximations that are tailored to the modified Cahn–Hilliard model.

Finally, in Chapter 6, we summarize the results of this thesis and discuss possible future directions.

Chapter 2

Mathematical Foundations

In this chapter, we review mathematical terms and basic ideas that are relevant for this thesis. We start with a functional analysis part and briefly go through normed spaces, Hilbert spaces, and Sobolev spaces. All stated definitions and results are mainly collected from [109, 141] and we refer the reader to these books for the proofs and further details. We continue with a brief concept of gradient flows taken from [20]. The last section of the functional analysis part collects the principles of unconditional stability, consistence, and convergence for PDEs, which do not arise from a gradient flow. Hence, the mathematical definitions have to be adapted to these kind of problems. The presented methodology is taken from [139]. We will need it in Chapter 5, where we utilize an unconditional time discretization scheme to a problem in image processing.

The remainder of this chapter summarizes basic concepts in the field of numerical linear algebra taken mainly from [136, 80]. First, we set up basic matrix properties and notations used throughout this thesis. Then, we briefly review the concept of eigenvalues. They play the crucial role in the development of efficient iterative solvers for systems of linear equations. Next, we recall common vector and matrix norms. They are important for the numerical analysis of our developed solvers. The matrix systems in this thesis are of block form. In particular, the Kronecker product and saddle point matrices play a key role and we briefly go through these concepts. The last section gives an overview of the iterative solution techniques. First, we give an idea of classical iterative methods. Then, we go over to Krylov subspace solvers, which form the state-of-the-art iterative solvers for large and sparse linear systems. Their convergence can be enhanced by applying an accelerator, also called preconditioner. We present the idea of preconditioning and recall a class of theoretical optimal preconditioners for saddle point problems. These optimal preconditioners form the basis for the development of efficient practical preconditioners in this thesis.

2.1 Functional analysis

This section recalls the basis and main theorems from functional analysis that are relevant as background information for this thesis. Besides the definition of Banach, Hilbert, and Sobolev spaces we go through the concepts of operators, (weak) convergence and derivatives, and dual spaces. Moreover, we collect important inequalities and integral identities. The last two sections review the principles of gradient flows as well as unconditional stability, consistence, and convergence for PDEs that do not arise from a gradient flow.

2.1.1 Normed spaces and Banach spaces

In the following, we review the basic concepts of important metric spaces — normed spaces and in particular Banach spaces. These include the methodology of linear operators defined on those spaces. Further, the idea of dual spaces belongs to the theory. The presented tools are taken from the textbooks [109, 102].

A normed space is a vector space with a metric defined by a norm. If the normed space is a complete metric space, then it is called a Banach space. If $\mathcal{X}$ and $\mathcal{Y}$ are normed spaces, then a mapping from $\mathcal{X}$ into $\mathcal{Y}$ is called an operator. A functional is a special operator. It is a mapping from $\mathcal{X}$ into the scalar field $\mathbb{R}$ or $\mathbb{C}$. The set of all bounded linear functionals on $\mathcal{X}$ is called the dual space $\mathcal{X}^*$ of $\mathcal{X}$. The dual space is again a normed space.

Definition 2.1 (Linear operator [109, p. 82])**.** *A linear operator T is an operator such that*

(i) the domain $\mathcal{D}(T)$ of T is a vector space and the range $\mathcal{R}(T)$ lies in a vector space over the same field,

(ii) for all $x, y \in \mathcal{D}(T)$ and scalars α, $T(\alpha x + \beta y) = \alpha Tx + \beta Ty$.

In what follows, we use the following notations. If T is a linear operator, then $\mathcal{D}(T)$ denotes the domain of T and $\mathcal{R}(T)$ denotes the range of T.

Definition 2.2 (Bounded linear operator [109, p. 91])**.** *Let $\mathcal{X}$ and $\mathcal{Y}$ be normed spaces and $T\colon \mathcal{D}(T) \to \mathcal{Y}$ a linear operator, where $\mathcal{D}(T) \subset \mathcal{X}$. The operator T is said to be bounded if there is a real number c such that for all $x \in \mathcal{D}(T)$*

$$||Tx||_{\mathcal{Y}} \leq c||x||_{\mathcal{X}}.$$

Then, we can define the norm of T as

$$||T|| = \sup_{x\in\mathcal{D}(T),\, x\neq 0} \frac{||Tx||_{\mathcal{Y}}}{||x||_{\mathcal{X}}} = \sup_{x\in\mathcal{D}(T),\, ||x||_{\mathcal{X}}=1} ||Tx||_{\mathcal{Y}}.$$

Theorem 2.1 (Continuity and boundedness [109, p. 97])**.** *Let $\mathcal{X}$ and $\mathcal{Y}$ be normed spaces. Let $T\colon \mathcal{D}(T) \to \mathcal{Y}$ be a linear operator, where $\mathcal{D}(T) \subset \mathcal{X}$. Then*

(i) T is continuous if and only if T is bounded.

(ii) If T is continuous at a single point, it is continuous.

For two normed spaces $\mathcal{X}$ and $\mathcal{Y}$ (both real or both convex), we define the set $\mathcal{L}(\mathcal{X}, \mathcal{Y})$ of all bounded linear operators from $\mathcal{X}$ into $\mathcal{Y}$.

Theorem 2.2 (The space $\mathcal{L}(\mathcal{X}, \mathcal{Y})$ [109, p. 118])**.** *The vector space $\mathcal{L}(\mathcal{X}, \mathcal{Y})$ of all bounded linear operators from a normed space $\mathcal{X}$ into a normed space $\mathcal{Y}$ is itself a normed space with the norm defined by*

$$\|T\|_{\mathcal{L}(\mathcal{X},\mathcal{Y})} = \sup_{x\in\mathcal{X},\, x\neq 0} \frac{\|Tx\|_{\mathcal{Y}}}{\|x\|_{\mathcal{X}}} = \sup_{x\in\mathcal{X},\, \|x\|_{\mathcal{X}}=1} \|Tx\|_{\mathcal{Y}}.$$

Definition 2.3 (Linear functional [109, p. 104])**.** *A linear functional f is a linear operator with domain in a vector space $\mathcal{X}$ and range in the scalar field $\mathcal{K}$ of $\mathcal{X}$. Thus,*

$$f : \mathcal{D}(f) \to \mathcal{K},$$

where $\mathcal{K} = \mathbb{R}$ if $\mathcal{X}$ is real and $\mathcal{K} = \mathbb{C}$ if $\mathcal{X}$ is complex.

Definition 2.4 (Bounded linear functional [109, p. 104])**.** *Let $\mathcal{X}$ be a normed space. A bounded linear functional f is a bounded linear operator with range in the scalar field of $\mathcal{X}$ in which the domain $\mathcal{D}(f)$ lies. Thus, there exists a real number c such that for all $x \in \mathcal{D}(f)$*

$$|f(x)| \leq c\|x\|_{\mathcal{X}}.$$

Furthermore, the norm of f is

$$\|f\| = \sup_{x\in\mathcal{D}(f),\, x\neq 0} \frac{|f(x)|}{\|x\|_{\mathcal{X}}} = \sup_{x\in\mathcal{D}(f),\, \|x\|_{\mathcal{X}}=1} |f(x)|.$$

Theorem 2.3 (Continuity and boundedness [109, p. 104])**.** *A linear functional f with domain $\mathcal{D}(f)$ in a normed space is continuous if and only if f is bounded.*

Definition 2.5 (Dual space [109, p. 119])**.** *Let $\mathcal{X}$ be a normed space. The dual space $\mathcal{X}^*$ of $\mathcal{X}$ is the set of all bounded linear functionals on $\mathcal{X}$. It constitutes a normed space with the norm defined by*

$$\|f\|_{\mathcal{X}^*} = \sup_{x\in\mathcal{X},\, x\neq 0} \frac{|f(x)|}{\|x\|_{\mathcal{X}}} = \sup_{x\in\mathcal{X},\, \|x\|_{\mathcal{X}}=1} |f(x)|.$$

In particular, $\mathcal{X}^*$ is a Banach space.

Definition 2.6 (Duality pairing [102, p. 8])**.** *Let $\mathcal{X}, \mathcal{X}^*$ be normed spaces with $\mathcal{X}^*$ being the dual space of $\mathcal{X}$. For $x \in \mathcal{X}$ and $f \in \mathcal{X}^*$, we denote by $\langle f, x\rangle_{\mathcal{X}^*\times\mathcal{X}}$ and $\langle x, f\rangle_{\mathcal{X}\times\mathcal{X}^*}$ the duality pairing defined via*

$$\langle f, x\rangle_{\mathcal{X}^*\times\mathcal{X}} = \langle x, f\rangle_{\mathcal{X}\times\mathcal{X}^*} := f(x).$$

Theorem 2.4 (Cauchy's inequality [102, p. 8])**.** *Let $\mathcal{X}, \mathcal{X}^*$ be normed spaces with $\mathcal{X}^*$ being the dual space of $\mathcal{X}$. Let $x \in \mathcal{X}$ and $f \in \mathcal{X}^*$, then*

$$|\langle f, x\rangle_{\mathcal{X}^*\times\mathcal{X}}| \leq \|f\|_{\mathcal{X}^*}\, \|x\|_{\mathcal{X}}.$$

2.1.2 Convergence in normed spaces

In the following, we review the basic concepts of convergence in normed spaces. In particular, we summarize the ideas of strong and weak convergence. The presented tools are taken from the textbooks [109, 116, 98]. At the end, we recall the notion of Newton differentiability [92]. We will see in Chapter 3–5 that we are dealing with nonlinear operators that are not Fréchet-differentiable. Hence, we need the weaker concept of Newton differentiability.

Definition 2.7. *[Strong convergence [109, p. 256]] Let $\mathcal{X}$ be a normed space. A sequence $\{x_k\} \subset \mathcal{X}$ is said to be strongly convergent (or convergent in the norm) if there is an $x \in \mathcal{X}$ such that*

$$\lim_{k\to\infty} \|x_k - x\|_{\mathcal{X}} = 0.$$

This is written as $x_k \to x$.

Definition 2.8 (Weak convergence [109, p. 257])**.** *Let $\mathcal{X}$ be a normed space. A sequence $\{x_k\} \subset \mathcal{X}$ is said to be weakly convergent if there is an $x \in \mathcal{X}$ such that for every $f \in \mathcal{X}^*$*

$$\lim_{k\to\infty} f(x_k) = f(x).$$

This is written as $x_k \rightharpoonup x$.

Lemma 2.5 (Weak convergence [109, p. 258])**.** *Let $\mathcal{X}$ be a normed space and $\{x_k\} \subset \mathcal{X}$ a weakly convergent sequence in $\mathcal{X}$. Then:*

(i) The weak limit x of $\{x_k\}$ is unique.

(ii) Every subsequence of $\{x_k\}$ converges weakly to x.

(iii) The sequence $\{\|x_k\|\}$ is bounded.

Theorem 2.6 (Strong and weak convergence [109, p. 259])**.** *Let $\mathcal{X}$ be a normed space and $\{x_k\} \subset \mathcal{X}$ a strongly convergent sequence in $\mathcal{X}$ with $x_k \to x$. Then $x_k \rightharpoonup x$.*

Definition 2.9 ([116, p. 125])**.** *Let $\mathcal{X}$ be a normed space and $\mathcal{S} \subset \mathcal{X}$. Then, $\mathcal{S}$ is said to be weakly closed in $\mathcal{X}$ if for all weakly convergent sequences $\{x_k\} \subset \mathcal{S}$ with $x_k \rightharpoonup x$ in $\mathcal{X}$ it holds $x \in \mathcal{S}$.*

Lemma 2.7 ([116, p. 126])**.** *Let $\mathcal{X}$ be a normed space and $\mathcal{S} \subset \mathcal{X}$ be closed and convex. Then, $\mathcal{S}$ is weakly closed in $\mathcal{X}$.*

Definition 2.10 ([98, p. 89])**.** *Let $\mathcal{X}$ be a real Banach space.*

(i) A functional $F \colon \mathcal{X} \to (-\infty, \infty]$ is called convex if

$$F((1-\lambda)x_1 + \lambda x_2) \leq (1-\lambda)F(x_1) + \lambda F(x_2)$$

for all $x_1, x_2 \in \mathcal{X}$ and $0 \leq \lambda \leq 1$. It is called proper if it is not identically ∞.

(ii) A functional $F \colon \mathcal{X} \to (-\infty, \infty]$ is said to be lower semicontinuous at $x \in \mathcal{X}$ if

$$F(x) \leq \liminf_{y\to x} F(y).$$

A functional F is lower semicontinuous if it is lower semicontinuous at all $x \in \mathcal{X}$.

(iii) *A functional $F\colon \mathcal{X} \to (-\infty, \infty]$ is said to be weakly lower semicontinuous at $x \in \mathcal{X}$ if*

$$F(x) \leq \liminf_{k\to\infty} F(x_k)$$

for all sequences $\{x_k\}$ converging weakly to x. F is weakly lower semicontinuous if it is weakly lower semicontinuous at all $x \in \mathcal{X}$.

In this thesis, we need the following weaker notion of Newton differentiability; see [92, 91].

Definition 2.11 ([92, p. 866])**.** *Let $\mathcal{X}$ and $\mathcal{Y}$ be Banach spaces, $\mathcal{D} \subset \mathcal{X}$ an open subset. A mapping $F\colon \mathcal{D} \to \mathcal{Y}$ is called Newton-differentiable in the open subset $\mathcal{U} \subset \mathcal{D}$ if there exists a family of mappings $G\colon \mathcal{U} \to \mathcal{L}(\mathcal{X}, \mathcal{Y})$ such that*

$$\lim_{d\to 0} \frac{\|F(x+d) - F(x) - G(x+d)d\|_{\mathcal{Y}}}{\|d\|_{\mathcal{X}}} = 0 \quad \forall x \in \mathcal{U}.$$

The operator G is called a Newton derivative of F on $\mathcal{U}$.

For such mappings, the following convergence result for the (semismooth) Newton iteration

$$x^{(k+1)} = x^{(k)} - G(x^{(k)})^{-1}F(x^{(k)}), \quad k = 0, 1, \ldots \tag{2.1}$$

holds true.

Theorem 2.8 ([92, p. 867])**.** *Let $\mathcal{X}$ and $\mathcal{Y}$ be Banach spaces, $\mathcal{D} \subset \mathcal{X}$ an open subset. Let $F\colon \mathcal{D} \to \mathcal{Y}$ be Newton-differentiable in an open neighborhood $\mathcal{U} \subset \mathcal{D}$ containing x^* with Newton derivative $G(x)$. Suppose that x^* is a solution of $F(x) = 0$. If $G(x)$ is nonsingular for all $x \in \mathcal{U}$ and $\{\|G(x)^{-1}\|_{\mathcal{L}(\mathcal{Y},\mathcal{X})} : x \in \mathcal{U}\}$ is bounded, then the sequence $\{x^{(k)}\}$ generated by (2.1) converges superlinearly to x^*, provided that $\|x^{(0)} - x^*\|_{\mathcal{X}}$ is sufficiently small.*

2.1.3 Inner product spaces and Hilbert spaces

In this section, we refine the concepts from the previous sections to certain classes of normed spaces and Banach spaces — the inner product spaces and Hilbert spaces.

An inner product space is a vector space $\mathcal{X}$ with an inner product $(\cdot, \cdot)_{\mathcal{X}}$ defined on $\mathcal{X}$. An inner product on $\mathcal{X}$ defines a norm on $\mathcal{X}$ given by

$$\|x\|_{\mathcal{X}} = \sqrt{(x, x)_{\mathcal{X}}}$$

and a metric on $\mathcal{X}$ given by

$$d(x, y)_{\mathcal{X}} = \|x - y\|_{\mathcal{X}} = \sqrt{(x - y, x - y)_{\mathcal{X}}}.$$

If the inner product space is a complete metric space, then it is called a Hilbert space.

Theorem 2.9 (Riesz's Theorem [109, p. 188])**.** *Every bounded linear functional f on a Hilbert space $\mathcal{H}$ can be represented in terms of the inner product, namely,*

$$f(x) = (x, z)_{\mathcal{H}},$$

where z depends on f, is uniquely determined by f and has norm

$$\|z\|_{\mathcal{H}} = \|f\|_{\mathcal{H}^*}.$$

Proposition 2.10 (Strong convergence [94, p. 115]). *Let $\mathcal{H}$ be a Hilbert space. A sequence $\{x_k\} \subset \mathcal{H}$ converges strongly to $x \in \mathcal{H}$ if and only if* $\limsup_{k\to\infty} \|x_k\|_{\mathcal{H}} \le \|x\|_{\mathcal{H}}$.

Definition 2.12 (Weak convergence [116, p. 121]). *Let $\mathcal{H}$ be a Hilbert space. A sequence $\{x_k\} \subset \mathcal{H}$ is said to converge weakly to $x \in \mathcal{H}$ if*

$$\lim_{k\to\infty} (x_k, y)_{\mathcal{H}} = (x, y)_{\mathcal{H}}$$

for every $y \in \mathcal{H}$.

Lemma 2.11 (Continuity of inner products [109, p. 138]). *Let $\mathcal{X}$ be an inner product space. If $x_k \to x$ and $y_k \to y$ in $\mathcal{X}$, then $(x_k, y_k)_{\mathcal{X}} \to (x, y)_{\mathcal{X}}$.*

2.1.4 L^p and Sobolev spaces

The L^p spaces are important Banach spaces in the study of PDEs. They control the regularity of functions. The Sobolev spaces $W^{k,p}$ are subspaces of L^p spaces, which control additionally the regularity of the derivatives. The presented tools are mainly taken from the textbook [141].

In the following, let $\Omega \subset \mathbb{R}^d$ be an open set. We denote the closure of Ω by $\overline{\Omega} = \Omega \cup \partial\Omega$.

Definition 2.13 (L^p spaces [141, p. 377]). *Let $1 \le p < \infty$. The space $L^p(\Omega)$ is defined as*

$$L^p(\Omega) = \{u\colon \Omega \to \mathbb{R} : u \text{ is measurable}, \|u\|_{L^p} < \infty\}$$

with the norm

$$\|u\|_{L^p(\Omega)} = \|u\|_{L^p} = \left(\int_\Omega |u(\mathbf{x})|^p \,\mathrm{d}\mathbf{x}\right)^{\frac{1}{p}}.$$

An inner product in $L^2(\Omega)$ is given by

$$(u, v)_{L^2(\Omega)} = (u, v)_{L^2} = \int_\Omega u(\mathbf{x})\, v(\mathbf{x}) \,\mathrm{d}\mathbf{x},$$

which defines the above declared norm $\|\cdot\|_{L^2}$. The space $L^\infty(\Omega)$ is defined as

$$L^\infty(\Omega) = \{u\colon \Omega \to \mathbb{R} : u \text{ is measurable}, \|u\|_{L^\infty} < \infty\}$$

with the norm

$$\|u\|_{L^\infty(\Omega)} = \|u\|_{L^\infty} = \operatorname*{ess\,sup}_{\mathbf{x}\in\Omega} |u(\mathbf{x})|.$$

The essential supremum of a function is defined as

$$\operatorname*{ess\,sup}_{\mathbf{x}\in\Omega} u(\mathbf{x}) = \inf_{Z\in\Omega, |Z|=0} \sup_{\Omega\setminus Z} u(\mathbf{x}).$$

The generalization to vector-valued functions $\mathbf{u} = [u_1, \ldots, u_N]^T\colon \Omega \to \mathbb{R}^N$ is denoted by $(L^p(\Omega))^N$, where

$$(L^p(\Omega))^N = \left\{\mathbf{u}\colon \Omega \to \mathbb{R}^N\colon u_j \in L^p(\Omega),\ \ j = 1, \ldots, N\right\}$$

for $1 \le p < \infty$, with the norm

$$\|\mathbf{u}\|_{(L^p(\Omega))^N} = \|\mathbf{u}\|_{L^p} = \left(\sum_{j=1}^{N} \|u_j\|_{L^p}^p \right)^{\frac{1}{p}} .$$

An inner product in $(L^2(\Omega))^N$ is given by

$$(\mathbf{u}, \mathbf{v})_{(L^2(\Omega))^N} = (\mathbf{u}, \mathbf{v})_{L^2} = \int_\Omega \left(\sum_{j=1}^{N} u_j(\mathbf{x})\, v_j(\mathbf{x}) \right) \mathrm{d}\mathbf{x},$$

which defines the above declared norm $\|\cdot\|_{(L^2(\Omega))^N}$.

Remark 2.1. 1. *More precisely, $L^p(\Omega)$ and $(L^p(\Omega))^N$ are spaces of equivalence classes of functions. Two functions are equivalent if they are equal almost everywhere (a.e.).*

2. *For $1 \le p \le \infty$, the spaces $L^p(\Omega)$ and $(L^p(\Omega))^N$ are Banach spaces.*

3. *The spaces $L^2(\Omega)$ and $(L^2(\Omega))^N$ are Hilbert spaces.*

In what follows, we consider scalar functions. The definitions translate to the case of vector-valued functions. The following version of Young's inequality is also called Young's inequality with α_Y (modified Young's inequality) or Cauchy's inequality with α_Y. It is an important inequality in the L^p spaces. Here, we do not formulate the general inequality but the version which we need in this thesis.

Lemma 2.12 (Young's inequality [114, p. 4])**.** *Let $\alpha_\mathrm{Y} > 0$. For any $a, b \in \mathbb{R}$, we have*

$$|ab| \le \frac{\alpha_\mathrm{Y}}{2}|a|^2 + \frac{1}{2\alpha_\mathrm{Y}}|b|^2.$$

Lemma 2.13 (Duality of L^p spaces [141, p. 386])**.** *Let $1 < p < \infty$. Then, the dual space of $L^p(\Omega)$ can be identified with $L^q(\Omega)$ where $p^{-1} + q^{-1} = 1$.*

For dealing with partial derivatives, we make use of the following compact notation.

Definition 2.14 (Multi-index [141, p. 414])**.** *Let d be the spatial dimension. A multi-index is a vector $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_d]^T \in \mathbb{N}_0^d$. By $|\boldsymbol{\alpha}| = \sum_{i=1}^{d} \alpha_i$ we denote the length of the multi-index $\boldsymbol{\alpha}$. Let $u \colon \Omega \to \mathbb{R}$ be a sufficiently often continuously differentiable function. We define the $\boldsymbol{\alpha}$th partial derivative of u by*

$$D^{\boldsymbol{\alpha}} u = \frac{\partial^{|\boldsymbol{\alpha}|} u}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2} \cdots \partial x_d^{\alpha_d}}.$$

Definition 2.15 (k-times continuously differentiable functions [41, p. 107])**.** *The space of continuous functions on Ω is denoted by $C(\Omega)$, i.e.,*

$$C(\Omega) = \{u \colon \Omega \to \mathbb{R} \colon u \text{ is continuous on } \Omega\}.$$

Let $k \in \mathbb{N}_0$. The space of k-times continuously differentiable functions on Ω is denoted by $C^k(\Omega)$, i.e.,

$$C^k(\Omega) = \{u \colon \Omega \to \mathbb{R} \colon D^{\boldsymbol{\alpha}} u \text{ exists and belongs to } C(\Omega)\ \forall |\boldsymbol{\alpha}| \le k\}.$$

Further

$$C^k(\overline{\Omega}) = \{u \in C^k(\Omega) \colon D^{\boldsymbol{\alpha}} u \text{ has a continuous extension on } \overline{\Omega} \ \forall |\boldsymbol{\alpha}| \leq k\}.$$

The case $k = 0$ gives $C^0(\Omega) = C(\Omega)$ and $C^0(\overline{\Omega}) = C(\overline{\Omega})$. For the case $k = \infty$, the space $C^\infty(\Omega)$ is defined as

$$C^\infty(\Omega) = \{u \colon \Omega \to \mathbb{R} \colon u \text{ is infinitely differentiable}\} = \cap_{k=0}^{\infty} C^k(\Omega).$$

Further

$$C^\infty(\overline{\Omega}) = \cap_{k=0}^{\infty} C^k(\overline{\Omega}).$$

Definition 2.16 (Distributions [141, p. 415])**.** *The space of distributions (infinitely smooth functions with compact support) is defined by*

$$C_0^\infty(\Omega) = \{u \in C^\infty(\Omega) \colon \operatorname{supp}(u) \subset \Omega, \operatorname{supp}(u) \text{ is compact}\}.$$

The support is given by

$$\operatorname{supp}(u) = \overline{\{\mathbf{x} \in \Omega \colon u(\mathbf{x}) \neq 0\}}.$$

The following theorem is close to the definition of weak derivatives which will be introduced afterwards.

Theorem 2.14 ([141, p. 416])**.** *Let $u \in C^k(\Omega)$ and $\boldsymbol{\alpha}$ a multi-index such that $|\boldsymbol{\alpha}| \leq k$. It holds*

$$\int_\Omega D^{\boldsymbol{\alpha}} u(\mathbf{x})\, \phi(\mathbf{x})\, \mathrm{d}\mathbf{x} = (-1)^{|\boldsymbol{\alpha}|} \int_\Omega u(\mathbf{x})\, D^{\boldsymbol{\alpha}} \phi(\mathbf{x})\, \mathrm{d}\mathbf{x} \quad \forall \phi \in C_0^\infty(\Omega).$$

Definition 2.17 (Space of locally-integrable functions [141, p. 416])**.** *Let $1 \leq p < \infty$. A function $u \colon \Omega \to \mathbb{R}$ is said to be locally p-integrable in Ω if $u \in L^p(K)$ for every compact subset $K \subset \Omega$. The space of all locally p-integrable functions in Ω is denoted by $L^p_{\mathrm{loc}}(\Omega)$.*

Theorem 2.15 (Weak derivative [141, p. 417])**.** *Let $u \in L^1_{\mathrm{loc}}(\Omega)$ and let $\boldsymbol{\alpha}$ be a multi-index. The function $D_w^{\boldsymbol{\alpha}} u \in L^1_{\mathrm{loc}}(\Omega)$ is said to be the weak $\boldsymbol{\alpha}$th derivative of u if*

$$\int_\Omega D_w^{\boldsymbol{\alpha}} u(\mathbf{x})\, \phi(\mathbf{x})\, \mathrm{d}\mathbf{x} = (-1)^{|\boldsymbol{\alpha}|} \int_\Omega u(\mathbf{x})\, D^{\boldsymbol{\alpha}} \phi(\mathbf{x})\, \mathrm{d}\mathbf{x} \quad \forall \phi \in C_0^\infty(\Omega).$$

Lemma 2.16 (Uniqueness of the weak derivative [141, p. 417])**.** *Let $u \in L^1_{\mathrm{loc}}(\Omega)$ and let $\boldsymbol{\alpha}$ be a multi-index. The weak $\boldsymbol{\alpha}$th derivative $D_w^{\boldsymbol{\alpha}} u \in L^1_{\mathrm{loc}}(\Omega)$ is defined uniquely in Ω up to a zero-measure subset of Ω.*

Lemma 2.17 (Compatibility of weak and classical derivatives [141, p. 417])**.** *Let $u \in C^k(\Omega)$ and let $\boldsymbol{\alpha}$ be a multi-index such that $|\boldsymbol{\alpha}| \leq k$. Then, the classical $\boldsymbol{\alpha}$th derivative $D^{\boldsymbol{\alpha}} u$ is identical to the weak $\boldsymbol{\alpha}$th derivative $D_w^{\boldsymbol{\alpha}} u$.*

Definition 2.18 (Sobolev spaces [141, p. 418])**.** *Let $1 \leq p \leq \infty$ and $1 \leq k \in \mathbb{N}$. The Sobolev space $W^{k,p}(\Omega)$ is defined as*

$$W^{k,p}(\Omega) = \{u \in L^p(\Omega) \colon D_w^{\boldsymbol{\alpha}} u \text{ exists and lies in } L^p(\Omega) \ \forall\, |\boldsymbol{\alpha}| \leq k\}.$$

For every $1 \leq p < \infty$, the norm $\|\cdot\|_{W^{k,p}}$ is defined as

$$\|u\|_{W^{k,p}(\Omega)} = \|u\|_{W^{k,p}} = \left(\sum_{|\boldsymbol{\alpha}| \leq k} \|D_w^{\boldsymbol{\alpha}} u\|_{L^p}^p \right)^{\frac{1}{p}}.$$

For $p = 2$, an inner product in $H^k(\Omega) := W^{k,2}(\Omega)$ is given by

$$(u, v)_{H^k(\Omega)} = (u, v)_{H^k} = \sum_{|\boldsymbol{\alpha}|\le k} \int_\Omega D_w^{\boldsymbol{\alpha}} u(\mathbf{x})\, D_w^{\alpha} v(\mathbf{x})\, \mathrm{d}\mathbf{x},$$

which defines the above declared norm $\|\cdot\|_{W^{k,2}} = \|\cdot\|_{H^k}$.

In the $W^{k,p}(\Omega)$ spaces, we use the following standard seminorm

$$|u|_{W^{k,p}(\Omega)} = |u|_{W^{k,p}} = \left(\sum_{|\boldsymbol{\alpha}|=k} \|D_w^{\boldsymbol{\alpha}} u\|_{L^p}^p \right)^{\frac{1}{p}}$$

for $1 \le p < \infty$.

Remark 2.2. 1. *For $1 \le p \le \infty$ and $k \in \mathbb{N}$, the spaces $W^{k,p}(\Omega)$ are Banach spaces.*

2. *For $k \in \mathbb{N}$, the spaces $W^{k,2}(\Omega) = H^k(\Omega)$ are Hilbert spaces.*

For the following useful integral identities, we need to specify our open set $\Omega \subset \mathbb{R}^d$ a bit further.

Definition 2.19 (Domain [141, p. 412])**.** *A subset $\Omega \subset \mathbb{R}^d$ is said to be a domain if it is nonempty, open, and connected.*

Moreover, we need the notion of a Lipschitz-continuous boundary of a domain in $\mathbb{R}^d$. Roughly speaking, it means there exists a finite covering of the boundary $\partial\Omega$ consisting of open d-dimensional rectangles such that in each rectangle $\partial\Omega$ can be expressed as a Lipschitz-continuous function of $d-1$ variables; see [141, p. 423]. In what follows, $\Omega \subset \mathbb{R}^d$ is a bounded domain with Lipschitz-continuous boundary. By $\mathbf{n}(\mathbf{x}) = [n_1, n_2, \dots, n_d]^T(\mathbf{x})$ we denote the unit outer normal vector to the boundary $\partial\Omega$ (defined a.e. on $\partial\Omega$). An important theorem related to the integration by parts is the following.

Theorem 2.18 (Gauss' theorem [141, p. 416])**.** *For every $u, v \in C^1(\Omega) \cap C(\overline{\Omega})$, we have*

$$\int_\Omega \frac{\partial u}{\partial x_i} v \,\mathrm{d}\mathbf{x} = -\int_\Omega u \frac{\partial v}{\partial x_i} \,\mathrm{d}\mathbf{x} + \int_{\partial\Omega} u\, v\, n_i \,\mathrm{d}\mathbf{s}.$$

The above theorem generalizes to the divergence of vector fields.

Theorem 2.19 (Stokes' theorem [141, p. 416])**.** *Every smooth vector field $\mathbf{w} \in [C^1(\Omega) \cap C(\overline{\Omega})]^d$ satisfies*

$$\int_\Omega \nabla \cdot \mathbf{w}(\mathbf{x}) \,\mathrm{d}\mathbf{x} = \int_{\partial\Omega} \mathbf{w}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) \,\mathrm{d}\mathbf{s}.$$

Now, we recall important standard integral identities that are used in the weak formulation of PDEs. The Theorems 2.18 and 2.19 generalize as follows.

Theorem 2.20 (Green's theorem for H^1-functions [141, p. 425])**.** *For every $u, v \in H^1(\Omega)$, it holds*

$$\int_\Omega \frac{\partial u}{\partial x_i} v \,\mathrm{d}\mathbf{x} = -\int_\Omega u \frac{\partial v}{\partial x_i} \,\mathrm{d}\mathbf{x} + \int_{\partial\Omega} u\, v\, n_i \,\mathrm{d}\mathbf{s}.$$

Green's theorem is the basis for the following two integral identities.

Lemma 2.21 ([141, p. 425]). *For all $u \in H^1(\Omega)$ and $v \in H^2(\Omega)$, it holds*

$$\int_\Omega u\,\Delta v \,\mathrm{d}\mathbf{x} = -\int_\Omega \nabla u \cdot \nabla v \,\mathrm{d}\mathbf{x} + \int_{\partial\Omega} u\,\frac{\partial v}{\partial \mathbf{n}}\,\mathrm{d}\mathbf{s},$$

where $\frac{\partial v}{\partial \mathbf{n}} = \nabla v(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x})$, $\mathbf{x} \in \partial\Omega$. For all $\mathbf{u} \in [H^1(\Omega)]^d$ and $v \in H^1(\Omega)$, it holds

$$\int_\Omega (\nabla \cdot \mathbf{u})\,v \,\mathrm{d}\mathbf{x} = -\int_\Omega \mathbf{u} \cdot \nabla v \,\mathrm{d}\mathbf{x} + \int_{\partial\Omega} (\mathbf{u}\cdot\mathbf{n})\,v\,\mathrm{d}\mathbf{s}.$$

Theorem 2.22 (Poincaré inequality [121, p. 64]). *There is a constant c_{P} such that for every $u \in H^m(\Omega)$, where $m \geq 1$,*

$$\|u\|^2_{H^m} \leq c_{\mathrm{P}} \left[\sum_{|\alpha|=m} \|D^\alpha_w u\|^2_{L^2} + \sum_{|\alpha|<m} \left(\int_\Omega D^\alpha_w u \,\mathrm{d}\mathbf{x} \right)^2 \right].$$

In the special case $m = 1$ and u has a zero average $\int_\Omega u \,\mathrm{d}\mathbf{x} = 0$, the Poincaré inequality asserts that

$$\|u\|^2_{H^1} \leq c_{\mathrm{P}}\,\|\nabla u\|^2_{L^2}.$$

In this thesis, we will mainly use the $L^2(\Omega)$- and $H^1(\Omega)$-inner product or norm. Hence during the rest of this thesis, we write $(\cdot,\cdot)$ and $\|\cdot\|$ for the $L^2(\Omega)$-inner product and $L^2(\Omega)$-norm. We denote by $(\cdot,\cdot)_1$ and $\|\cdot\|_1$ the $H^1(\Omega)$-inner product and $H^1(\Omega)$-norm. Further, we will make use of the inner product

$$(\mathbf{u},\mathbf{v}) = \int_\Omega \mathbf{u}\cdot\mathbf{v}\,\mathrm{d}\mathbf{x} = \sum_{i=1}^N (u_i, v_i)$$

in $L^2(\Omega)^N$ with the norm $\|\cdot\|$, and of the inner product

$$(\mathbf{u},\mathbf{v})_1 = (\mathbf{u},\mathbf{v}) + (\nabla\mathbf{u},\nabla\mathbf{v}) = \sum_{i=1}^N \left((u_i,v_i) + (\nabla u_i, \nabla v_i)\right) = \sum_{i=1}^N (u_i,v_i)_1$$

in $H^1(\Omega)^N$ with the norm $\|\cdot\|_1$. For vector-valued functions $\mathbf{u} \in H^1(\Omega)^N$ or $\mathbf{u} \in H^2(\Omega)^N$, we use the notation $\nabla\mathbf{u} = [\nabla u_1, \dots, \nabla u_N]^T$ and $\Delta\mathbf{u} = [\Delta u_1, \dots, \Delta u_N]^T$. Moreover, we denote by $\langle\cdot,\cdot\rangle$ the duality pairing between $H^1(\Omega)^*$ and $H^1(\Omega)$ as well as its natural extension to vector-valued functions. We define

$$H^{-1}(\Omega) = \left\{ u^* \in H^1(\Omega)^* \mid \langle u^*, 1\rangle = 0 \right\}, \tag{2.2}$$

which is equipped with the mass conserving H^{-1}-inner product

$$(v_1, v_2)_{-1} := \int_\Omega \nabla(-\Delta)^{-1} v_1 \cdot \nabla(-\Delta)^{-1} v_2 \,\mathrm{d}\mathbf{x}.$$

Here, $y = (-\Delta)^{-1} v$ is the weak solution of

$$\begin{aligned} -\Delta y &= v, \\ \nabla y \cdot \mathbf{n} &= 0 \quad \text{on } \partial\Omega. \end{aligned}$$

Note that the solution to this elliptic problem is only defined up to a constant. We always choose y such that $\int_\Omega y \,\mathrm{d}\mathbf{x} = 0$. Here, we finish the general functional analytic setting. We refer the reader to the textbooks [109, 141] for further reading. The following last two sections recall the definitions of gradient flows (taken from [20]) as well as of unconditional stability, consistence, and convergence for PDEs, which do not arise from a gradient flow (taken from [139]).

2.1.5 Gradient flows

According to [20], let $\mathcal{Z}$ be a vector space. Let $\mathcal{U} \subset \mathcal{Z}$ be an affine subspace, i.e., there exists a $\bar{u} \in \mathcal{Z}$ and a linear space $\mathcal{Y} \subset \mathcal{Z}$ such that $\mathcal{U} = \bar{u} + \mathcal{Y}$. Let $\mathcal{E} \colon \mathcal{U} \to \mathbb{R}$ be a sufficiently smooth function.

Definition 2.20 (Variational derivative [20, p. 932])**.** *The first variation of $\mathcal{E}$ at a point $u \in \mathcal{U}$ in a direction $v \in \mathcal{Y}$ is defined by*

$$\frac{\delta \mathcal{E}}{\delta u}(u)(v) := \lim_{\delta \to 0} \frac{\mathcal{E}(u + \delta v) - \mathcal{E}(u)}{\delta}.$$

There exists a gradient of $\mathcal{E}$ with respect to the inner product $(\cdot,\cdot)_{\mathcal{Z}}$ on $\mathcal{Z}$, denoted by $\mathrm{grad}_{\mathcal{Z}}\mathcal{E}(u)$, if

$$(\mathrm{grad}_{\mathcal{Z}}\mathcal{E}(u), v)_{\mathcal{Z}} = \frac{\delta \mathcal{E}}{\delta u}(u)(v) \quad \forall v \in \mathcal{Y}. \tag{2.3}$$

Now, the gradient flow of $\mathcal{E}$ with respect to the inner product $(\cdot,\cdot)_{\mathcal{Z}}$ is given as

$$\partial_t u(t) = -\mathrm{grad}_{\mathcal{Z}}\mathcal{E}(u(t)). \tag{2.4}$$

Eyre [67] analyzed semi-implicit time discretization schemes of gradient flows, which are constructed to be unconditional gradient stable. In particular, he concentrated on the case where the underlying energy functional is not convex. The approach is called convexity splitting, was originally introduced by Elliott and Stuart [61], and is often attributed to Eyre [67]. In the following, we briefly illustrate its idea based on Eyre's setting. Note that this setting is a purely discrete one. However, as remarked in [139, p. 420], the concept holds in a more general framework.

Let $\boldsymbol{u}(t) \in C^1(\mathbb{R}_+, \mathbb{R}^m)$, $E(\boldsymbol{u}) \in C^2(\mathbb{R}^m, \mathbb{R})$ and $\nabla E(\boldsymbol{u})$ be the gradient of E. Consider the initial value problem

$$\boldsymbol{u}_t = -\nabla E(\boldsymbol{u}),\ \boldsymbol{u}(0) = \boldsymbol{u}_0. \tag{2.5}$$

If E fulfills the following conditions

- $E(\boldsymbol{u}) \geq 0 \ \forall \boldsymbol{u} \in \mathbb{R}^m$,
- $E(\boldsymbol{u}) \to \infty$ as $\|\boldsymbol{u}\| \to \infty$,
- $(\mathrm{J}(\nabla E)(\boldsymbol{u})\boldsymbol{u}, \boldsymbol{u}) \geq \lambda \ \forall \boldsymbol{u} \in \mathbb{R}^m$,

then (2.5) is called a gradient system and its solutions are called gradient flows. Here, $\mathrm{J}(\nabla E)(\boldsymbol{u})$ is the Jacobian of $\nabla E(\boldsymbol{u})$ and $\lambda \in \mathbb{R}$. If $E(\boldsymbol{u})$ is not convex, then $\lambda < 0$, and multiple equilibria of (2.5) may exist. The idea of convexity splitting is to write $E(\boldsymbol{u})$ as the sum of a convex plus a concave energy functional. The convex part is then treated implicitly whilst the concave part is treated explicitly. In formulas, we write $E(\boldsymbol{u})$ as

$$E(\boldsymbol{u}) = E_c(\boldsymbol{u}) - E_e(\boldsymbol{u}), \tag{2.6}$$

where

$$E_o \in C^2(\mathbb{R}^m, \mathbb{R}) \text{ and } E_o(\boldsymbol{u}) \text{ is strictly convex for all } \boldsymbol{u} \in \mathbb{R}^m, o \in \{c, e\}. \tag{2.7}$$

The semi-implicit discretization scheme of (2.5) is then given by

$$\boldsymbol{u}^{(n)} - \boldsymbol{u}^{(n-1)} = -\tau \left(\nabla E_c(\boldsymbol{u}^{(n)}) - \nabla E_e(\boldsymbol{u}^{(n-1)}) \right), \tag{2.8}$$

where $\boldsymbol{u}^{(n)}$ approximates $\boldsymbol{u}(n\tau)$ with time step size τ for $n \in \mathbb{N}$ and $\boldsymbol{u}^{(0)} = \boldsymbol{u}_0$.

Theorem 2.23 ([67, p. 3]). *If $E_c(\boldsymbol{u})$ and $E_e(\boldsymbol{u})$ satisfy (2.6)–(2.7) and $E_e(\boldsymbol{u})$ satisfies*

$$(\mathrm{J}(\nabla E_e)(\boldsymbol{u})\boldsymbol{u}, \boldsymbol{u}) \geq -\lambda$$

when $\lambda < 0$, then for any initial condition, the numerical scheme (2.8) is consistent, gradient stable for all $\tau > 0$, and possesses a unique solution for each time step.

Next, we recall the definitions of unconditional stability, consistence, and convergence for PDEs, which do not arise from a gradient flow. The presented tools are taken from [139].

2.1.6 Unconditional stability, consistence and convergence

Let u be an element of a suitable function space $\mathcal{H}$ defined on $\Omega \times [0, T]$, with $\Omega \subset \mathbb{R}^2$ open and bounded, and $T > 0$. Consider the PDE

$$u_t = G(u, D^{\boldsymbol{\alpha}} u), \tag{2.9}$$

where G is a real valued function and $D^{\boldsymbol{\alpha}} u$ are the space derivatives with $|\boldsymbol{\alpha}| \leq 4$. In what follows, we write $u^{(n)} = u(n\tau)$ for a solution of the continuous Equation (2.9) at time $n\tau$ with time step size τ, where $n \in \mathbb{N}$. A corresponding discrete time stepping method is denoted by

$$U^{(n)} = U^{(n-1)} + \tau G^{(n-1)}(U^{(n-1)}, U^{(n)}, D^{\boldsymbol{\alpha}} U^{(n-1)}, D^{\boldsymbol{\alpha}} U^{(n)}), \tag{2.10}$$

where $G^{(n-1)}$ is a suitable approximation of G in $U^{(n-1)}$ and $U^{(n)}$. We denote by capital $U^{(n)}$ the nth solution of the time-discrete Equation (2.10). We assume that $u^{(n)}, U^{(n)} \in L^2(\Omega)$ for all $n \in \mathbb{N}_0$.

Definition 2.21 ([139, p. 421].). *The discrete time stepping method (2.10) is*

1. *unconditional stable if all solutions of (2.10) are bounded for all $\tau > 0$ and all n such that $n\tau \leq T$;*

2. *consistent if*

$$\lim_{\tau \to 0} \eta^{(n-1)}(\tau) = 0,$$

where $\eta^{(n-1)}(\tau)$ is the local truncation error of the scheme and defined as

$$\eta^{(n-1)}(\tau) = \frac{u^{(n)} - u^{(n-1)}}{\tau} - G^{(n-1)}(u^{(n-1)}, u^{(n)}, D^{\boldsymbol{\alpha}} u^{(n-1)}, D^{\boldsymbol{\alpha}} u^{(n)}). \tag{2.11}$$

Moreover, we define the global truncation error to be

$$\eta(\tau) = \max_n \|\eta^{(n)}(\tau)\|_{\mathcal{H}}.$$

A numerical scheme is said to be of order p in time if

$$\eta(\tau) = \mathcal{O}(\tau^p) \quad \text{for } \tau \to 0.$$

In this thesis, we will abbreviate $\eta^{(n-1)}$ and η for $\eta^{(n-1)}(\tau)$ and $\eta(\tau)$.

With this brief introduction of unconditional stability, consistence, and convergence, we finish the functional analytic part. Next, we change over to the fully discrete setting and the field of numerical linear algebra.

2.2 Basic matrix theory

In this section, we set up basic matrix concepts and notations used throughout this thesis. These include the theory of eigenvalues, which play the crucial role in the development of efficient iterative solvers for systems of linear equations. We recall common vector and matrix norms, which we need for the numerical analysis of our developed solvers. The matrix systems that we are considering in this thesis are of block form. In particular, the Kronecker product and saddle point matrices play a key role and are summarized at the end of this section. The presented results are mainly collected from [136, 80].

2.2.1 Matrix properties

The vector space of all $n \times m$ matrices is denoted by $\mathbb{C}^{n\times m}$. Let $\boldsymbol{A} \in \mathbb{C}^{n\times m}$ with

$$\boldsymbol{A} = (a_{ij})_{\substack{i=1,\dots,n,\\ j=1,\dots,m}} = \begin{bmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nm} \end{bmatrix}.$$

We write a_{ij} or $[\boldsymbol{A}]_{ij}$ for the (i, j) entry of $\boldsymbol{A}$. The transpose of $\boldsymbol{A}$ is given by

$$\boldsymbol{A}^T = (a_{ij})_{\substack{j=1,\dots,m,\\ i=1,\dots,n}} = \begin{bmatrix} a_{11} & \cdots & a_{n1} \\ \vdots & \ddots & \vdots \\ a_{1m} & \cdots & a_{nm} \end{bmatrix} \in \mathbb{C}^{m\times n}.$$

The transpose conjugate matrix is indicated by $\boldsymbol{A}^H = \overline{\boldsymbol{A}}^T = \overline{\boldsymbol{A}^T}$, where the bar denotes the elementwise complex conjugation. The vector space of all m-vectors is denoted by $\mathbb{C}^m$. We write row vectors $\boldsymbol{z} \in \mathbb{C}^{1\times m}$ as $\boldsymbol{z} = [z_1, \dots, z_m]$ and column vectors $\boldsymbol{z} \in \mathbb{C}^m = \mathbb{C}^{m\times 1}$ as $\boldsymbol{z} = [z_1, \dots, z_m]^T$.

A subspace of $\mathbb{C}^m$ is a subset of $\mathbb{C}^m$ that is also a complex vector space. Given the vectors $\boldsymbol{z}_1, \dots, \boldsymbol{z}_n \in \mathbb{C}^m$, then the set of all linear combinations of these vectors is a subspace called the span of $\{\boldsymbol{z}_1, \dots, \boldsymbol{z}_n\}$

$$\operatorname{span}\{\boldsymbol{z}_1, \dots, \boldsymbol{z}_n\} = \left\{ \boldsymbol{z} \in \mathbb{C}^m : \boldsymbol{z} = \sum_{j=1}^{n} \alpha_j \boldsymbol{z}_j,\ \alpha_j \in \mathbb{C}, j = 1, \dots, n \right\}.$$

If the set $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$ is linearly independent, then each vector of $\operatorname{span}\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$ admits a unique expression as a linear combination of the α_j's. Two important subspaces associated with $\boldsymbol{A} \in \mathbb{C}^{n \times m}$ are first the range of $\boldsymbol{A}$ defined by

$$\operatorname{ran}(\boldsymbol{A}) = \{\boldsymbol{A}\boldsymbol{z} \colon \boldsymbol{z} \in \mathbb{C}^m\} = \operatorname{span}\{\boldsymbol{a}_{*1}, \ldots, \boldsymbol{a}_{*m}\},$$

where $\boldsymbol{a}_{*j}$ denotes the jth column of $\boldsymbol{A}$. Second, the kernel or nullspace of $\boldsymbol{A}$ given by

$$\ker(\boldsymbol{A}) = \{\boldsymbol{z} \in \mathbb{C}^m \colon \boldsymbol{A}\boldsymbol{z} = \boldsymbol{0}\}.$$

The rank of $\boldsymbol{A}$ is defined as

$$\operatorname{rank}(\boldsymbol{A}) = \dim(\operatorname{ran}(\boldsymbol{A})).$$

For $\boldsymbol{A} \in \mathbb{C}^{n \times m}$, it holds

$$m = \dim(\ker(\boldsymbol{A})) + \operatorname{rank}(\boldsymbol{A}).$$

During the rest of this thesis, we consider real matrices $\boldsymbol{A} \in \mathbb{R}^{n \times m}$. The matrix $\boldsymbol{A}$ is square if $m = n$. In the following, we assume $\boldsymbol{A} \in \mathbb{R}^{m \times m}$ if not stated otherwise. We denote the identity matrix by

$$\boldsymbol{I} = (\delta_{ij})_{i,j=1,\ldots,m} = \begin{bmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{bmatrix},$$

where δ_{ij} is the Kronecker delta. The identity matrix satisfies $\boldsymbol{A}\boldsymbol{I} = \boldsymbol{I}\boldsymbol{A} = \boldsymbol{A}$. An important property of matrices is invertibility. $\boldsymbol{A}$ is invertible if there exists a matrix $\boldsymbol{B} \in \mathbb{R}^{m \times m}$ such that

$$\boldsymbol{B}\boldsymbol{A} = \boldsymbol{A}\boldsymbol{B} = \boldsymbol{I}.$$

If this is the case, then $\boldsymbol{B}$ is called the inverse of $\boldsymbol{A}$ and $\boldsymbol{A}$ is called invertible or nonsingular. Otherwise, we call $\boldsymbol{A}$ singular. The inverse of $\boldsymbol{A}$ is uniquely determined and we denote it by $\boldsymbol{A}^{-1}$. One criterion for determining the invertibility of $\boldsymbol{A}$ uses the determinant. The matrix $\boldsymbol{A}$ is nonsingular if and only if $\det(\boldsymbol{A}) \neq 0$. $\boldsymbol{A}$ is called orthogonal if $\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}$. Hence, the inverse of an orthogonal matrix $\boldsymbol{A}$ is given by $\boldsymbol{A}^{-1} = \boldsymbol{A}^T$.

2.2.2 Spectrum of matrices

Another important concept is built by the eigenvalues and eigenvectors of $\boldsymbol{A}$. They are in particular crucial for the development of powerful iterative solvers; see Section 2.3.1.

Definition 2.22 (Eigenvalues and eigenvectors [136, p. 3]). *$\lambda \in \mathbb{C}$ is called an eigenvalue of $\boldsymbol{A}$ if there exists a vector $\boldsymbol{0} \neq \boldsymbol{v} \in \mathbb{C}^m$ such that $\boldsymbol{A}\boldsymbol{v} = \lambda\boldsymbol{v}$. The vector $\boldsymbol{v}$ is called an eigenvector of $\boldsymbol{A}$ associated with λ. We call the pair $(\lambda, \boldsymbol{v})$ an eigenpair of $\boldsymbol{A}$. The set of all the eigenvalues of $\boldsymbol{A}$ is called the spectrum of $\boldsymbol{A}$ and is denoted by $\sigma(\boldsymbol{A})$.*

$\boldsymbol{A}$ can have at most m distinct eigenvalues. Based on Definition 2.22, we can introduce the characteristic polynomial of $\boldsymbol{A}$. It is defined by

$$c_{\boldsymbol{A}}(t) = \det(\boldsymbol{A} - t\boldsymbol{I}),$$

where $t \in \mathbb{C}$. It can be shown that $c_{\boldsymbol{A}}(t)$ is a polynomial of degree m; see [136, Exercise 1.8]. During this thesis, we will denote by Π_m the set of real polynomials of degree m. It holds that $\lambda \in \mathbb{C}$ is an eigenvalue of $\boldsymbol{A}$ if and only if $c_{\boldsymbol{A}}(\lambda) = 0$. Hence, $\boldsymbol{A}$ is nonsingular if and only if $0 \notin \sigma(\boldsymbol{A})$. The maximum modulus of the eigenvalues of $\boldsymbol{A}$ is called spectral radius and is denoted by

$$\rho(\boldsymbol{A}) = \max_{\lambda \in \sigma(\boldsymbol{A})} |\lambda|.$$

Definition 2.23 (Similarity [136, p. 15])**.** *Two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are said to be similar if there is a nonsingular matrix $\boldsymbol{C}$ such that*

$$\boldsymbol{A} = \boldsymbol{C}\boldsymbol{B}\boldsymbol{C}^{-1}.$$

The similarity of $\boldsymbol{A}$ and $\boldsymbol{B}$ implies that they have the same eigenvalues.

Definition 2.24 (Diagonalizability [95, p. 59])**.** *If $\boldsymbol{A}$ is similar to a diagonal matrix, then $\boldsymbol{A}$ is said to be diagonalizable.*

Definition 2.25 (Simultaneously diagonalizability [95, p. 61])**.** *Two matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{m\times m}$ are said to be simultaneously diagonalizable if there exists a single nonsingular $\boldsymbol{C} \in \mathbb{R}^{m\times m}$ such that $\boldsymbol{C}^{-1}\boldsymbol{A}\boldsymbol{C}$ and $\boldsymbol{C}^{-1}\boldsymbol{B}\boldsymbol{C}$ are both diagonal.*

Definition 2.26 (Commutativity [95, p. 21])**.** *$\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{m\times m}$ are said to commute if $\boldsymbol{A}\boldsymbol{B} = \boldsymbol{B}\boldsymbol{A}$.*

Theorem 2.24 ([95, p. 62])**.** *Let $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{m\times m}$ be diagonalizable. Then $\boldsymbol{A}$ and $\boldsymbol{B}$ commute if and only if they are simultaneously diagonalizable.*

The structure and properties of $\boldsymbol{A}$ plays an important role in numerical linear algebra. Examples include the development of efficient solution strategies for linear systems of the form $\boldsymbol{A}\boldsymbol{z} = \boldsymbol{b}$ or the advancement of eigenvalue solvers. For some special classes of matrices, one can specify their spectrum further. The most important type relevant for this thesis are symmetric matrices. They have the property $\boldsymbol{A}^T = \boldsymbol{A}$.

Theorem 2.25 (Spectrum of symmetric matrices [136, p. 25])**.** *The eigenvalues of a symmetric matrix $\boldsymbol{A}$ are real, i.e., $\sigma(\boldsymbol{A}) \subset \mathbb{R}$.*

The maximum and minimum eigenvalues $\lambda_{\min}(\boldsymbol{A})$ and $\lambda_{\max}(\boldsymbol{A})$ of a symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{m\times m}$ can be estimated by the use of the Rayleigh quotient.

Definition 2.27 (Rayleigh quotient)**.** *Let $\boldsymbol{A} \in \mathbb{R}^{m\times m}$ be symmetric and $\boldsymbol{0} \neq \boldsymbol{z} \in \mathbb{R}^m$. The Rayleigh quotient of $\boldsymbol{A}$ and $\boldsymbol{z}$ is defined by*

$$R_{\boldsymbol{A}}(\boldsymbol{z}) = \frac{\boldsymbol{z}^T\boldsymbol{A}\boldsymbol{z}}{\boldsymbol{z}^T\boldsymbol{z}}.$$

Note that if $\boldsymbol{z}$ is an eigenvector of $\boldsymbol{A}$, then the Rayleigh quotient is the corresponding eigenvalue.

Theorem 2.26 (Min-Max Theorem [136, p. 25])**.** *The eigenvalues of a symmetric matrix $\boldsymbol{A}$ satisfy*

$$\lambda_{\min}(\boldsymbol{A}) = \min_{\boldsymbol{z}\neq\boldsymbol{0}} R_{\boldsymbol{A}}(\boldsymbol{z}),$$
$$\lambda_{\max}(\boldsymbol{A}) = \max_{\boldsymbol{z}\neq\boldsymbol{0}} R_{\boldsymbol{A}}(\boldsymbol{z}).$$

Theorem 2.27 (Symmetric Schur decomposition [80, p. 440]). *If $\boldsymbol{A} \in \mathbb{R}^{m\times m}$ is symmetric, then there exists an orthogonal $\boldsymbol{Q} \in \mathbb{R}^{m\times m}$ such that*

$$\boldsymbol{Q}^T \boldsymbol{A} \boldsymbol{Q} = \boldsymbol{\Lambda} = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_m \end{bmatrix}.$$

Moreover, $\boldsymbol{A}\boldsymbol{q}_j = \lambda_j \boldsymbol{q}_j$ for $j = 1, \dots, m$, where $\boldsymbol{q}_j$ denotes the jth column of $\boldsymbol{Q}$.

Hence, any symmetric matrix is diagonalizable. In particular, a symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{m\times m}$ admits a set of orthonormal eigenvectors that forms a basis of $\mathbb{R}^m$.

Theorem 2.28 (Sylvester Law of Inertia [136, p. 448]). *If $\boldsymbol{A} \in \mathbb{R}^{m\times m}$ is symmetric and $\boldsymbol{C} \in \mathbb{R}^{m\times m}$ is nonsingular, then $\boldsymbol{A}$ and $\boldsymbol{C}^T\boldsymbol{A}\boldsymbol{C}$ have the same numbers of negative, zero, and positive eigenvalues.*

A symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{m\times m}$ is called

- positive definite if
$$\boldsymbol{z}^T\boldsymbol{A}\boldsymbol{z} > 0 \quad \forall \boldsymbol{0} \neq \boldsymbol{z} \in \mathbb{R}^m,$$
- positive semidefinite if
$$\boldsymbol{z}^T\boldsymbol{A}\boldsymbol{z} \geq 0 \quad \forall \boldsymbol{0} \neq \boldsymbol{z} \in \mathbb{R}^m,$$
- negative definite if
$$\boldsymbol{z}^T\boldsymbol{A}\boldsymbol{z} < 0 \quad \forall \boldsymbol{0} \neq \boldsymbol{z} \in \mathbb{R}^m, \text{and}$$
- negative semidefinite if
$$\boldsymbol{z}^T\boldsymbol{A}\boldsymbol{z} \leq 0 \quad \forall \boldsymbol{0} \neq \boldsymbol{z} \in \mathbb{R}^m.$$

A symmetric matrix, which is neither positive definite, positive semidefinite, negative definite, nor negative semidefinite, is called indefinite. Hence, $\boldsymbol{A} \in \mathbb{R}^{m\times m}$ is symmetric

- positive definite if and only if $\sigma(\boldsymbol{A}) \subset \mathbb{R}_{>0}$,
- positive semidefinite if and only if $\sigma(\boldsymbol{A}) \subset \mathbb{R}_{\geq 0}$,
- negative definite if and only if $\sigma(\boldsymbol{A}) \subset \mathbb{R}_{<0}$,
- negative semidefinite if and only if $\sigma(\boldsymbol{A}) \subset \mathbb{R}_{\leq 0}$, and
- indefinite if and only if it has both, positive and negative, eigenvalues.

Theorem 2.29. *Let $\boldsymbol{A} \in \mathbb{R}^{m\times m}$ be symmetric positive definite and $\boldsymbol{B} \in \mathbb{R}^{m\times m}$ symmetric. Then, for all $\boldsymbol{0} \neq \boldsymbol{v} \in \mathbb{R}^m$ it holds*

$$\lambda_{\min}(\boldsymbol{A}^{-1}\boldsymbol{B}) \leq \frac{\boldsymbol{v}^T\boldsymbol{B}\boldsymbol{v}}{\boldsymbol{v}^T\boldsymbol{A}\boldsymbol{v}} \leq \lambda_{\max}(\boldsymbol{A}^{-1}\boldsymbol{B}).$$

Proof. Since $\boldsymbol{A}^{-1}\boldsymbol{B}$ is similar to $\boldsymbol{A}^{\frac{1}{2}}(\boldsymbol{A}^{-1}\boldsymbol{B})\boldsymbol{A}^{-\frac{1}{2}} = \boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{B}\boldsymbol{A}^{-\frac{1}{2}}$, it holds $\sigma\left(\boldsymbol{A}^{-1}\boldsymbol{B}\right) = \sigma\left(\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{B}\boldsymbol{A}^{-\frac{1}{2}}\right)$. Since $\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{B}\boldsymbol{A}^{-\frac{1}{2}}$ is symmetric, we obtain for $\boldsymbol{0} \neq \boldsymbol{w} \in \mathbb{R}^m$

$$\begin{aligned}\lambda_{\min}(\boldsymbol{A}^{-1}\boldsymbol{B}) = \lambda_{\min}(\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{B}\boldsymbol{A}^{-\frac{1}{2}}) \leq \frac{\boldsymbol{w}^T\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{B}\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{w}}{\boldsymbol{w}^T\boldsymbol{w}} &= \frac{\boldsymbol{v}^T\boldsymbol{B}\boldsymbol{v}}{\boldsymbol{v}^T\boldsymbol{A}\boldsymbol{v}}\\ &\leq \lambda_{\max}(\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{B}\boldsymbol{A}^{-\frac{1}{2}}) = \lambda_{\max}(\boldsymbol{A}^{-1}\boldsymbol{B}),\end{aligned}$$

where $\boldsymbol{v} = \boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{w} \neq \boldsymbol{0}$. □

Lemma 2.30. *Let $\boldsymbol{A} \in \mathbb{R}^{m\times m}$ be symmetric positive definite and $\boldsymbol{B} \in \mathbb{R}^{m\times m}$ symmetric positive semidefinite. Then $\sigma(\boldsymbol{A}^{-1}\boldsymbol{B}) \subset \mathbb{R}_{\geq 0}$. If $\boldsymbol{B}$ is additionally symmetric positive definite, then $\sigma(\boldsymbol{A}^{-1}\boldsymbol{B}) \subset \mathbb{R}_{>0}$.*

Proof. Since $\boldsymbol{A}^{-1}\boldsymbol{B}$ is similar to $\boldsymbol{A}^{\frac{1}{2}}(\boldsymbol{A}^{-1}\boldsymbol{B})\boldsymbol{A}^{-\frac{1}{2}} = \boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{B}\boldsymbol{A}^{-\frac{1}{2}}$, it holds $\sigma\left(\boldsymbol{A}^{-1}\boldsymbol{B}\right) = \sigma\left(\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{B}\boldsymbol{A}^{-\frac{1}{2}}\right)$. It is clear that $\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{B}\boldsymbol{A}^{-\frac{1}{2}}$ is symmetric. Let $\boldsymbol{0} \neq \boldsymbol{z} \in \mathbb{R}^m$. Then

$$\boldsymbol{z}^T\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{B}\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{z} = (\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{z})^T\boldsymbol{B}(\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{z}) =: \boldsymbol{y}^T\boldsymbol{B}\boldsymbol{y} \geq 0.$$

□

2.2.3 Vector and matrix norms

Next, we recall important vector and matrix norms. We utilize the Euclidean inner product on $\mathbb{C}^m$. For two vectors $\boldsymbol{y} = [y_1, \ldots, y_m]^T$, $\boldsymbol{z} = [z_1, \ldots, z_m]^T \in \mathbb{C}^m$, it is defined by

$$(\boldsymbol{y}, \boldsymbol{z}) = \boldsymbol{y}^H\boldsymbol{z} = \sum_{j=1}^{m} \overline{y_j}z_j.$$

The most commonly used vector norms are the p-norms defined by

$$||\boldsymbol{z}||_p = \sqrt[p]{\sum_{j=1}^{m} |z_j|^p}.$$

The case $p = 2$ gives the Euclidean norm

$$||\boldsymbol{z}|| = \sqrt{\sum_{j=1}^{m} |z_j|^2} = \sqrt{(\boldsymbol{z}, \boldsymbol{z})},$$

which is associated with the Euclidean inner product. The vector norms $||\cdot||_p$ and $||\cdot||_q$ induce the matrix norm $||\cdot||_{pq}$, which is defined for $\boldsymbol{A} \in \mathbb{R}^{n\times m}$ as

$$||\boldsymbol{A}||_{pq} = \max_{\boldsymbol{z}\in\mathbb{R}^m, \boldsymbol{z}\neq\boldsymbol{0}} \frac{||\boldsymbol{A}\boldsymbol{z}||_p}{||\boldsymbol{z}||_q} = \max_{||\boldsymbol{z}||_q=1} ||\boldsymbol{A}\boldsymbol{z}||_p.$$

In the case $p = q$, we obtain the p-norm and write $||\cdot||_{pq} = ||\cdot||_p$. These p-norms satisfy the inequality

$$||\boldsymbol{A}\boldsymbol{B}||_p \leq ||\boldsymbol{A}||_p||\boldsymbol{B}||_p$$

for $\boldsymbol{A} \in \mathbb{R}^{n \times m}$ and $\boldsymbol{B} \in \mathbb{R}^{m \times l}$. During this thesis, we denote the 2-norm simply by $\|\cdot\|$. Important p-norms are

$$\begin{aligned}
\|\boldsymbol{A}\|_1 &= \max_{j=1,\dots,m} \sum_{i=1}^{n} |a_{ij}| \quad \text{(maximum absolute column sum norm)},\\
\|\boldsymbol{A}\|_\infty &= \max_{i=1,\dots,n} \sum_{j=1}^{m} |a_{ij}| \quad \text{(maximum absolute row sum norm)},\\
\|\boldsymbol{A}\| &= \sqrt{\rho(\boldsymbol{A}^T\boldsymbol{A})} = \sqrt{\rho(\boldsymbol{A}\boldsymbol{A}^T)} \quad \text{(spectral norm)}.
\end{aligned}$$

It holds

$$\|\boldsymbol{A}\| \le \sqrt{\|\boldsymbol{A}\|_1 \|\boldsymbol{A}\|_\infty}, \tag{2.12}$$

see [80, pp. 72–73]. Note that for a symmetric matrix $\boldsymbol{A}$, we have

$$\|\boldsymbol{A}\| = \rho(\boldsymbol{A}).$$

Theorem 2.31. *Let $\boldsymbol{A} \in \mathbb{R}^{m \times m}$. For each $k \in \mathbb{N}$, it holds*

$$\rho(\boldsymbol{A}) \le \|\boldsymbol{A}^k\|_p^{\frac{1}{k}}.$$

Proof. Let λ be an eigenvalue of $\boldsymbol{A}$ with corresponding eigenvector $\boldsymbol{v}$. It holds

$$|\lambda|^k \|\boldsymbol{v}\|_p = \|\lambda^k \boldsymbol{v}\|_p = \|\boldsymbol{A}^k \boldsymbol{v}\|_p \le \|\boldsymbol{A}^k\|_p \|\boldsymbol{v}\|_p.$$

Since $\boldsymbol{0} \ne \boldsymbol{v}$, we get $|\lambda|^k \le \|\boldsymbol{A}^k\|_p$ and therefore $|\lambda| \le \|\boldsymbol{A}^k\|_p^{\frac{1}{k}}$. Since λ is any eigenvalue of $\boldsymbol{A}$, it also holds for the maximum modulus of the eigenvalues. □

Theorem 2.32. *Let $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{m \times m}$ be symmetric. Then,*

$$\rho(\boldsymbol{A}\boldsymbol{B}) \le \rho(\boldsymbol{A})\rho(\boldsymbol{B}).$$

If $\boldsymbol{A}$ is additionally positive definite and $\boldsymbol{B}$ is additionally positive semidefinite, then

$$\max_{\lambda \in \sigma(\boldsymbol{A}\boldsymbol{B})} \lambda \le \left(\max_{\lambda \in \sigma(\boldsymbol{A})} \lambda\right)\left(\max_{\mu \in \sigma(\boldsymbol{B})} \mu\right).$$

If $\boldsymbol{B}$ is additionally positive definite, then

$$\min_{\lambda \in \sigma(\boldsymbol{A}\boldsymbol{B})} \lambda \ge \left(\min_{\lambda \in \sigma(\boldsymbol{A})} \lambda\right)\left(\min_{\mu \in \sigma(\boldsymbol{B})} \mu\right).$$

Proof. Let $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{m \times m}$ be symmetric. Then,

$$\rho(\boldsymbol{A}\boldsymbol{B}) \le \|\boldsymbol{A}\boldsymbol{B}\| \le \|\boldsymbol{A}\|\,\|\boldsymbol{B}\| = \rho(\boldsymbol{A})\rho(\boldsymbol{B}). \tag{2.13}$$

Now, let $\boldsymbol{A}$ additionally be positive definite and $\boldsymbol{B}$ additionally be positive semidefinite. Lemma 2.30 implies that $\sigma(\boldsymbol{A}\boldsymbol{B}) \subset \mathbb{R}_{\ge 0}$. Hence, (2.13) becomes

$$\max_{\lambda \in \sigma(\boldsymbol{A}\boldsymbol{B})} \lambda \le \left(\max_{\lambda \in \sigma(\boldsymbol{A})} \lambda\right)\left(\max_{\mu \in \sigma(\boldsymbol{B})} \mu\right).$$

Now, let $\boldsymbol{B}$ additionally be positive definite. Lemma 2.30 implies that $\sigma(\boldsymbol{AB}) \subset \mathbb{R}_{>0}$, i.e., the product $\boldsymbol{AB}$ is nonsingular. Hence,

$$\begin{aligned} \frac{1}{\min_{\lambda\in\sigma(\boldsymbol{AB})}\lambda} &= \rho((\boldsymbol{AB})^{-1}) = \rho(\boldsymbol{B}^{-1}\boldsymbol{A}^{-1}) \leq \|\boldsymbol{B}^{-1}\boldsymbol{A}^{-1}\| \\ &\leq \|\boldsymbol{A}^{-1}\|\,\|\boldsymbol{B}^{-1}\| = \rho(\boldsymbol{A}^{-1})\rho(\boldsymbol{B}^{-1}) = \frac{1}{\left(\min_{\lambda\in\sigma(\boldsymbol{A})}\lambda\right)\left(\min_{\mu\in\sigma(\boldsymbol{B})}\mu\right)}. \end{aligned}$$

□

An important concept in the matrix theory is the condition number of a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times m}$. This quantity depends on the chosen norm. In this thesis, we will only equip it with the 2-norm. Hence, we define the condition number as

$$\kappa(\boldsymbol{A}) := \|\boldsymbol{A}\|\,\|\boldsymbol{A}^{-1}\|.$$

In some sense, it measures the degree of singularity of a matrix. In general, if $\kappa(\boldsymbol{A})$ is large,[1], then $\boldsymbol{A}$ is said to be an ill-conditioned matrix. Hence, an almost singular matrix is usually expected to have a large condition number. We will see in Section 2.3.1 that the condition number can be used to characterize the convergence behavior of iterative solution methods. It holds $\kappa(\boldsymbol{A}) \geq 1$. Moreover, if $\boldsymbol{A}$ is symmetric, then $\kappa(\boldsymbol{A}) = |\lambda_{\max}(\boldsymbol{A})|\,|\lambda_{\min}(\boldsymbol{A})|^{-1}$, where $\lambda_{\max}(\boldsymbol{A})$ and $\lambda_{\min}(\boldsymbol{A})$ are the maximum and minimum eigenvalue of $\boldsymbol{A}$.

Finally, we make use of the following notation. If $\boldsymbol{A}$ is a symmetric positive definite matrix, we define the $\boldsymbol{A}$-norm as

$$\|z\|_{\boldsymbol{A}} = \sqrt{(\boldsymbol{z}, \boldsymbol{Az})}.$$

2.2.4 Block and saddle point matrices

Matrices can be characterized via their outer structure. A special class are circulant matrices, which appear in Chapter 4.

Definition 2.28 (Circulant matrix [80, pp. 220–222])**.** *A circulant matrix $\boldsymbol{C} \in \mathbb{C}^{m\times m}$ has the form*

$$\boldsymbol{C} = \begin{bmatrix} c_0 & c_{m-1} & \cdots & c_2 & c_1 \\ c_1 & c_0 & c_{m-1} & \cdots & c_2 \\ \vdots & c_1 & c_0 & \ddots & \vdots \\ c_{m-2} & \vdots & \ddots & \ddots & c_{m-1} \\ c_{m-1} & c_{m-2} & \cdots & c_1 & c_0 \end{bmatrix}.$$

Hence, $\boldsymbol{C}$ can be described by the vector $\boldsymbol{c} = [c_0, c_1, \ldots, c_{m-1}]^T$.

Circulant matrices are diagonalized by the discrete Fourier transform matrix.

Definition 2.29 (Discrete Fourier transform matrix [80, pp. 33–36])**.** *The discrete Fourier transform matrix $\boldsymbol{F}_m = (f_{kj})_{k,j=1,\ldots,m} \in \mathbb{C}^{m\times m}$ is defined by*

$$f_{kj} = \omega_m^{(k-1)(j-1)}$$

with $\omega_m = \exp(-2\pi i/m)$.

[1] Of course, this depends on the definition of 'large'; see, e.g., [80, Chapter 3.5].

Theorem 2.33 ([80, p. 222]). *Let $\boldsymbol{C} \in \mathbb{C}^{m \times m}$ be a circulant matrix with the corresponding vector $\boldsymbol{c}$ as described in Definition 2.28. If $\boldsymbol{\lambda} = \overline{\boldsymbol{F}}_m \boldsymbol{c}$, then*

$$\boldsymbol{F}_m^{-1} \boldsymbol{C} \boldsymbol{F}_m = \operatorname{diag}(\lambda_1, \ldots, \lambda_m).$$

Let $\boldsymbol{A} \in \mathbb{R}^{m \times m}$. Further important types of matrices relevant for this thesis are:

- diagonal matrices: $a_{ij} = 0$ for $j \neq i$, $\boldsymbol{A} = \operatorname{diag}(a_{ii})_{i=1,\ldots,m}$,
- upper triangular matrices: $a_{ij} = 0$ for $i > j$,
- lower triangular matrices: $a_{ij} = 0$ for $i < j$, and
- block matrices, which generalize the matrices by replacing each entry by a matrix.

When we are coming to vector-valued Cahn–Hilliard problems in Chapter 4 and 5, we make use of the notion of the Kronecker product. If $\boldsymbol{B} \in \mathbb{R}^{n_1 \times m_1}$ and $\boldsymbol{C} \in \mathbb{R}^{n_2 \times m_2}$ such that

$$\boldsymbol{B} = \begin{bmatrix} b_{11} & \cdots & b_{1m_1} \\ \vdots & \ddots & \vdots \\ b_{n_1 1} & \cdots & b_{n_1 m_1} \end{bmatrix},$$

then their Kronecker product $\boldsymbol{B} \otimes \boldsymbol{C} \in \mathbb{R}^{n_1 n_2 \times m_1 m_2}$ is given by

$$\boldsymbol{B} \otimes \boldsymbol{C} = \begin{bmatrix} b_{11}\boldsymbol{C} & \cdots & b_{1m_1}\boldsymbol{C} \\ \vdots & \ddots & \vdots \\ b_{n_1 1}\boldsymbol{C} & \cdots & b_{n_1 m_1}\boldsymbol{C} \end{bmatrix}.$$

Hence, $\boldsymbol{B} \otimes \boldsymbol{C}$ can also be seen as a $n_1 \times m_1$ block matrix whose (i, j) block is the $n_2 \times m_2$ matrix $b_{ij}\boldsymbol{C}$. Important Kronecker properties include (see [80, p. 27]):

$$(\boldsymbol{B} \otimes \boldsymbol{C})^T = \boldsymbol{B}^T \otimes \boldsymbol{C}^T, \tag{2.14}$$

$$(\boldsymbol{B} \otimes \boldsymbol{C})(\boldsymbol{D} \otimes \boldsymbol{F}) = \boldsymbol{B}\boldsymbol{D} \otimes \boldsymbol{C}\boldsymbol{F}, \tag{2.15}$$

$$(\boldsymbol{B} \otimes \boldsymbol{C})^{-1} = \boldsymbol{B}^{-1} \otimes \boldsymbol{C}^{-1}, \tag{2.16}$$

$$\boldsymbol{B} \otimes (\boldsymbol{C} \otimes \boldsymbol{D}) = (\boldsymbol{B} \otimes \boldsymbol{C}) \otimes \boldsymbol{D}.$$

Note that the matrix sizes in (2.15) must be compatible, i.e., $\boldsymbol{B} \in \mathbb{R}^{n_1 \times m_1}$, $\boldsymbol{C} \in \mathbb{R}^{n_2 \times m_2}$, $\boldsymbol{D} \in \mathbb{R}^{m_1 \times p_1}$, $\boldsymbol{F} \in \mathbb{R}^{m_2 \times p_2}$. Moreover, $\boldsymbol{B}$ and $\boldsymbol{C}$ in (2.16) must be nonsingular. Another property is the following:

$$\text{If } \boldsymbol{B} \text{ is } \left\{ \begin{array}{l} \text{diagonal} \\ \text{lower triangular} \\ \text{upper triangular} \end{array} \right\}, \text{ then } \boldsymbol{B} \otimes \boldsymbol{C} \text{ is } \left\{ \begin{array}{l} \text{block diagonal} \\ \text{block lower triangular} \\ \text{block upper triangular} \end{array} \right\}.$$

Theorem 2.34 ([142, p. 78]). *Let $\boldsymbol{B} \in \mathbb{R}^{n \times n}$ with eigenvalues $\lambda_1, \ldots, \lambda_n$ and corresponding eigenvectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$. Let $\boldsymbol{C} \in \mathbb{R}^{m \times m}$ with eigenvalues $\mu_1, \ldots, \mu_m$ and corresponding eigenvectors $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_m$. Then, the matrix $\boldsymbol{B} \otimes \boldsymbol{C}$ has the eigenvalues $\lambda_j \mu_k$ with the corresponding eigenvectors $\boldsymbol{v}_j \otimes \boldsymbol{w}_k$, where $1 \leq j \leq n$ and $1 \leq k \leq m$.*

A major class of block matrices are 2×2 block matrices arising from saddle point problems. Applications include computational fluid dynamics [79] or constrained optimization [78]. In particular, such kind of matrices form the core of this thesis. The discretization of Cahn–Hilliard type problems leads to linear systems of saddle point type. The following results are collected from [16].

Consider the 2×2 block matrix

$$\mathcal{A} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}_1^T \\ \boldsymbol{B}_2 & -\boldsymbol{C} \end{bmatrix}, \tag{2.17}$$

with $\boldsymbol{A} \in \mathbb{R}^{m \times m}$, $\boldsymbol{B}_1, \boldsymbol{B}_2 \in \mathbb{R}^{p \times m}$, $\boldsymbol{C} \in \mathbb{R}^{p \times p}$ and $m \geq p$. When (2.17) describes a saddle point problem, its blocks satisfy one or more of the following conditions according to [16, p. 3]:

- $\boldsymbol{A}$ is symmetric,
- the symmetric part of $\boldsymbol{A}$, i.e., $\frac{1}{2}(\boldsymbol{A} + \boldsymbol{A}^T)$ is positive semidefinite,
- $\boldsymbol{B}_1 = \boldsymbol{B}_2 = \boldsymbol{B}$,
- $\boldsymbol{C}$ is symmetric positive semidefinite,
- $\boldsymbol{C} = 0$.

Note that the saddle point problems in Chapter 3–5 have the property $m = p$. If $\boldsymbol{A}$ is nonsingular, the saddle point matrix $\mathcal{A}$ admits the following block triangular factorization:

$$\mathcal{A} = \begin{bmatrix} \boldsymbol{I} & 0 \\ \boldsymbol{B}_2\boldsymbol{A}^{-1} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{A} & 0 \\ 0 & \boldsymbol{S} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{A}^{-1}\boldsymbol{B}_1^T \\ 0 & \boldsymbol{I} \end{bmatrix}, \tag{2.18}$$

where $\boldsymbol{S} = -(\boldsymbol{C} + \boldsymbol{B}_2\boldsymbol{A}^{-1}\boldsymbol{B}_1^T)$ is the Schur complement of $\mathcal{A}$ in $\boldsymbol{A}$. It follows from (2.18) that $\mathcal{A}$ is nonsingular if and only if $\boldsymbol{S}$ is. Equivalent factorizations to (2.18) are

$$\mathcal{A} = \begin{bmatrix} \boldsymbol{A} & 0 \\ \boldsymbol{B}_2 & \boldsymbol{S} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{A}^{-1}\boldsymbol{B}_1^T \\ 0 & \boldsymbol{I} \end{bmatrix}, \tag{2.19}$$

$$\mathcal{A} = \begin{bmatrix} \boldsymbol{I} & 0 \\ \boldsymbol{B}_2\boldsymbol{A}^{-1} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}_1^T \\ 0 & \boldsymbol{S} \end{bmatrix}. \tag{2.20}$$

Hence, we can formulate the determinant of $\mathcal{A}$ as

$$\det(\mathcal{A}) = \det(\boldsymbol{A})\det(\boldsymbol{S}). \tag{2.21}$$

Similar to (2.18), we can write under the assumption that $\boldsymbol{C}$ is nonsingular

$$\mathcal{A} = \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{B}_1^T\boldsymbol{C}^{-1} \\ 0 & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{T} & 0 \\ 0 & -\boldsymbol{C} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & 0 \\ -\boldsymbol{C}^{-1}\boldsymbol{B}_2 & \boldsymbol{I} \end{bmatrix}, \tag{2.22}$$

where $\boldsymbol{T} = \boldsymbol{A} + \boldsymbol{B}_1^T\boldsymbol{C}^{-1}\boldsymbol{B}_2$ is the Schur complement of $\mathcal{A}$ in $-\boldsymbol{C}$. It follows from (2.22) that $\mathcal{A}$ is nonsingular if and only if $\boldsymbol{T}$ is. The three factorizations (2.18)–(2.20) will be used later in Section 2.3.2 when we discuss optimal solution techniques for problems of the form $\mathcal{A}\boldsymbol{z} = \boldsymbol{b}$. In order for $\mathcal{A}\boldsymbol{z} = \boldsymbol{b}$ to have a unique solution, $\mathcal{A}$ has to be nonsingular.

Theorem 2.35 ([16, pp. 15–16]). *Assume $\boldsymbol{A}$ is symmetric positive definite, $\boldsymbol{B}_1 = \boldsymbol{B}_2 = \boldsymbol{B}$, and $\boldsymbol{C}$ is symmetric positive semidefinite. If $\ker(\boldsymbol{C}) \cap \ker(\boldsymbol{B}^T) = \{\boldsymbol{0}\}$, then the saddle point matrix $\boldsymbol{\mathcal{A}}$ is nonsingular. In particular, $\boldsymbol{\mathcal{A}}$ is invertible if $\boldsymbol{B}$ has full rank.*

Remark 2.3. *Assume that $\boldsymbol{A}$ is symmetric positive definite, $\boldsymbol{B}_1 = \boldsymbol{B}_2 = \boldsymbol{B}$ has full rank, and $\boldsymbol{C}$ is symmetric positive semidefinite. Then, $\boldsymbol{S}$ is symmetric negative definite. It follows from the Sylvester Law of Inertia (Theorem 2.28) that $\boldsymbol{\mathcal{A}}$ is indefinite, with m positive and p negative eigenvalues.*

Assume that $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{A}$ are nonsingular. Then, according to (2.18)

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}_1^T \\ \boldsymbol{B}_2 & -\boldsymbol{C} \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{A}^{-1} + \boldsymbol{A}^{-1}\boldsymbol{B}_1^T\boldsymbol{S}^{-1}\boldsymbol{B}_2\boldsymbol{A}^{-1} & -\boldsymbol{A}^{-1}\boldsymbol{B}_1^T\boldsymbol{S}^{-1} \\ -\boldsymbol{S}^{-1}\boldsymbol{B}_2\boldsymbol{A}^{-1} & \boldsymbol{S}^{-1} \end{bmatrix}. \tag{2.23}$$

Similar, assume that $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{C}$ are nonsingular. Then, according to (2.22)

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}_1^T \\ \boldsymbol{B}_2 & -\boldsymbol{C} \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{T}^{-1} & \boldsymbol{T}^{-1}\boldsymbol{B}_1^T\boldsymbol{C}^{-1} \\ \boldsymbol{C}^{-1}\boldsymbol{B}_2\boldsymbol{T}^{-1} & -\boldsymbol{C}^{-1} + \boldsymbol{C}^{-1}\boldsymbol{B}_2\boldsymbol{T}^{-1}\boldsymbol{B}_1^T\boldsymbol{C}^{-1} \end{bmatrix}. \tag{2.24}$$

Next, assuming that $\boldsymbol{\mathcal{A}}$ is nonsingular, we want to discuss the numerical solution of systems of the form $\boldsymbol{\mathcal{A}}\boldsymbol{z} = \boldsymbol{b}$. We begin with a general matrix $\boldsymbol{\mathcal{A}}$ and come back to saddle point matrices in Section 2.3.3.

2.3 Iterative solution of linear systems

This section is devoted to the solution of linear systems of the general form

$$\boldsymbol{A}\boldsymbol{z} = \boldsymbol{b}. \tag{2.25}$$

Here, $\boldsymbol{A} \in \mathbb{R}^{m\times m}$ is the given coefficient matrix, $\boldsymbol{z} \in \mathbb{R}^m$ is the unknown solution vector, and $\boldsymbol{b} \in \mathbb{R}^m$ is the given right-hand side vector. In order for the Equation (2.25) to have a unique solution, we assume that $\boldsymbol{A}$ is nonsingular. Systems of the form (2.25) arise after the discretization of a continuous problem like a system of PDEs. The linear systems are usually of very large dimension in order to obtain an acceptable quality in the approximate solutions. Moreover, discretizations coming from FEM result in a sparse matrix $\boldsymbol{A}$, i.e., most of its entries are zero. There are two classes of approaches, which solve the linear system (2.25) — direct methods and iterative methods.

Direct methods [80, Chapter 11] are based on Gaussian elimination. Without careful modifications they suffer from the fill-in during the factorization process. Hence, for large and sparse matrices they demand high storage requirements. Effective versions like sparse elimination methods, which control the fill-in, build an own field of research; see, e.g., [77, 55, 51, 52]. The effectiveness and applicability of those methods depends on the matrix structure. But in general, direct methods are competitive for moderate system sizes. However, for much larger problems and in particular in combination with three-dimensional experiments the application of direct solvers becomes infeasible.

This is where iterative methods come in. Iterative methods form a general procedure for solving the system (2.25). Starting with an initial guess $\boldsymbol{z}^{(0)} \in \mathbb{R}^m$, they aim to generate a sequence of vectors $\{\boldsymbol{z}^{(l)}\}_{l\in\mathbb{N}}$, which converges to the solution $\boldsymbol{z}$ of (2.25). This section starts with an illustration of the strategy of iterative solvers on the basis of the Jacobi iteration. The Jacobi iteration is a well-known classic among the iterative approaches. This part is followed by an introduction into state-of-the-art iterative solvers. These are called Krylov subspace solvers. Such methods are usually only efficient in combination with an accelerator, which is called a preconditioner. We recall a class of theoretical optimal preconditioners for saddle point systems on which we want to build on. In fact, in theory the application of those preconditioners yield outstanding performances. A class of Krylov subspace solvers are proven to converge after only a small number of iterations when optimal preconditioners are included. These theoretical preconditioners form the basis for the development of practical approximations for the Cahn–Hilliard type problems discussed in this thesis. For an overview of iterative solvers and preconditioning techniques, we refer to [70, 86, 136, 63, 16, 2, 15, 149].

Classical iterative methods

Classical approaches, such as the Jacobi or Gauss-Seidel iteration, have proved their worth in the past. This is due to their easy derivation and implementation. Nowadays however, they often do not meet the desired demands of the rate of convergence. Instead of using these techniques as standalone solvers, they are nowadays mostly used as successful accelerators. This belongs to the concept of preconditioners, which is briefly touched at the end of this section. Section 2.3.2 provides a more detailed introduction into preconditioning techniques. We refer the reader to [136, Chapter 4] to get familiar with the classical iterative methods. We will only briefly discuss the Jacobi iteration as a representative. The following results are collected from [136, Chapter 4].

We begin with the splitting

$$\boldsymbol{A} = \boldsymbol{D} - \boldsymbol{L} - \boldsymbol{U},$$

such that (2.25) becomes

$$(\boldsymbol{D} - \boldsymbol{L} - \boldsymbol{U})\boldsymbol{z} = \boldsymbol{b}. \tag{2.26}$$

Here, $\boldsymbol{D}$ is the diagonal matrix of $\boldsymbol{A}$, which is assumed to be nonsingular. Further, $-\boldsymbol{L}$ is the strict lower triangular matrix of $\boldsymbol{A}$ and $-\boldsymbol{U}$ is the strict upper triangular matrix of $\boldsymbol{A}$. Since $\boldsymbol{D}$ is nonsingular, we can write (2.26) as

$$\boldsymbol{z} - \boldsymbol{D}^{-1}(\boldsymbol{L} + \boldsymbol{U})\boldsymbol{z} = \boldsymbol{D}^{-1}\boldsymbol{b}. \tag{2.27}$$

This can be formulated as a stationary iteration

$$\boldsymbol{z}^{(l+1)} = \boldsymbol{D}^{-1}(\boldsymbol{L} + \boldsymbol{U})\boldsymbol{z}^{(l)} + \boldsymbol{D}^{-1}\boldsymbol{b} =: \boldsymbol{G}\boldsymbol{z}^{(l)} + \boldsymbol{f} \tag{2.28}$$

for $l \in \mathbb{N}_0$. Let us denote the ith component of $\boldsymbol{z}^{(l)}$ by $z_i^{(l)}$ and the ith component of $\boldsymbol{b}$ by b_i. Then, (2.28) reads componentwise as

$$z_i^{(l+1)} = a_{ii}^{-1}\left(b_i - \sum_{j=1, j\neq i}^{m} a_{ij} z_j^{(l)}\right).$$

The following theorem answers the question of convergence.

Theorem 2.36 (Convergence of stationary iterations [136, p. 115])**.** *The stationary iteration (2.28) converges to the solution $\boldsymbol{z}$ of (2.25) if and only if* $\rho(\boldsymbol{G}) < 1$.

From (2.28), we can realize the efficiency regarding the computational costs per iteration step: We need to perform one matrix vector multiplication and one cheap solve with the favorable matrix $\boldsymbol{D}$. Moreover, we can bring out the connection to preconditioning. Since $\boldsymbol{G} = \boldsymbol{I} - \boldsymbol{D}^{-1}\boldsymbol{A}$, Equation (2.27) is equivalent to

$$\boldsymbol{D}^{-1}\boldsymbol{A}\boldsymbol{z} = \boldsymbol{D}^{-1}\boldsymbol{b}.$$

This is a preconditioned system of the original problem $\boldsymbol{A}\boldsymbol{z} = \boldsymbol{b}$ in (2.25). Both systems have the same solution. The matrix $\boldsymbol{D}$ is called preconditioner. In general, $\boldsymbol{D}$ can be any nonsingular matrix. In the case of the Jacobi iteration, $\boldsymbol{D}$ is constructed as the diagonal matrix of $\boldsymbol{A}$ as derived above.

Here, we finish our discussion about classical iterative solvers. In the following, we change over to the today's techniques.

2.3.1 Krylov subspace solvers

This section gives a brief review of Krylov subspace solvers, which form the state-of-the-art iterative solvers for large and sparse linear systems. The following results are mainly collected from [136, 63].

As stated in the last section, classical iterative methods nowadays often do not meet the desired demands of the speed of convergence. More efficient iterative methods are projection methods. The idea of projection techniques is to extract the sequence $\{\boldsymbol{z}^{(l)}\}$ of approximate solutions from an affine subspace $\boldsymbol{z}^{(0)} + \mathcal{K}_l$ of $\mathbb{R}^m$ of dimension l. This is called the search subspace. The uniqueness of $\boldsymbol{z}^{(l)}$ is typically realized via the imposition of l orthogonality conditions. We denote by

$$\boldsymbol{r}^{(l)} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{z}^{(l)}$$

the residuum in the lth step of the iterative method. It is a measure for the quality of the approximate solution $\boldsymbol{z}^{(l)}$. The l orthogonality conditions consist of constraining the lth residuum $\boldsymbol{r}^{(l)}$ to be orthogonal to l linearly independent vectors, i.e.,

$$\boldsymbol{r}^{(l)} \perp \mathcal{L}_l.$$

Here, $\mathcal{L}_l$ is another subspace of $\mathbb{R}^m$ of dimension l. It is called the subspace of constraints. This framework is commonly known as the Petrov-Galerkin conditions in diverse areas of mathematics, e.g., the FEM. The case $\mathcal{L}_l = \mathcal{K}_l$ leads to the Galerkin conditions. Let us write $\boldsymbol{z}^{(l)} = \boldsymbol{z}^{(0)} + \boldsymbol{k}^{(l)}$, where $\boldsymbol{k}^{(l)} \in \mathcal{K}_l$. Then

$$\boldsymbol{r}^{(l)} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{z}^{(l)} = \boldsymbol{b} - \boldsymbol{A}(\boldsymbol{z}^{(0)} + \boldsymbol{k}^{(l)}) = \boldsymbol{r}^{(0)} - \boldsymbol{A}\boldsymbol{k}^{(l)}. \tag{2.29}$$

Hence, the approximate solution $\boldsymbol{z}^{(l)}$ can be obtained from

$$\boldsymbol{z}^{(l)} = \boldsymbol{z}^{(0)} + \boldsymbol{k}^{(l)}, \quad \boldsymbol{k}^{(l)} \in \mathcal{K}_l, \tag{2.30}$$

$$\left(\boldsymbol{r}^{(0)} - \boldsymbol{A}\boldsymbol{k}^{(l)}, \boldsymbol{w}\right) = 0 \quad \forall \boldsymbol{w} \in \mathcal{L}_l. \tag{2.31}$$

The Krylov subspace

$$\mathcal{K}_l = \mathcal{K}_l(\boldsymbol{A}, \boldsymbol{r}^{(0)}) = \operatorname{span}\left\{\boldsymbol{r}^{(0)}, \boldsymbol{A}\boldsymbol{r}^{(0)}, \boldsymbol{A}^2\boldsymbol{r}^{(0)}, \dots, \boldsymbol{A}^{l-1}\boldsymbol{r}^{(0)}\right\} \tag{2.32}$$

has proven to be efficient. It makes use of the sparsity of $\boldsymbol{A}$ in which case matrix vector products are cheap to apply. According to [16, p. 50], it is known that the Krylov subspaces form a nested sequence. In particular,

$$\mathcal{K}_1(\boldsymbol{A}, \boldsymbol{r}^{(0)}) \subset \dots \subset \mathcal{K}_{d_\mathrm{K}}(\boldsymbol{A}, \boldsymbol{r}^{(0)}) = \dots = \mathcal{K}_m(\boldsymbol{A}, \boldsymbol{r}^{(0)}),$$

where $d_\mathrm{K} = \dim\left(\mathcal{K}_m(\boldsymbol{A}, \boldsymbol{r}^{(0)})\right) \leq m$. Moreover, it holds $\dim\left(\mathcal{K}_l(\boldsymbol{A}, \boldsymbol{r}^{(0)})\right) = l$ for each $l \leq d_\mathrm{K}$. This choice of $\mathcal{K}_l$ forms the basis of Krylov subspace methods. Depending on the properties of $\boldsymbol{A}$, different constraint subspaces $\mathcal{L}_l$ can be constructed which lead to unique approximates $\boldsymbol{z}^{(l)}, l = 1, 2, \dots$. The following theorem states two common examples.

Theorem 2.37 ([16, pp. 50-51],[136, p. 136])**.** *Suppose that* $\dim\left(\mathcal{K}_l(\boldsymbol{A}, \boldsymbol{r}^{(0)})\right) = l$. *If*

- $\boldsymbol{A}$ *is symmetric positive definite and* $\mathcal{L}_l = \mathcal{K}_l(\boldsymbol{A}, \boldsymbol{r}^{(0)})$ *($\rightsquigarrow$ conjugate gradient method), or*
- $\boldsymbol{A}$ *is nonsingular and* $\mathcal{L}_l = \boldsymbol{A}\mathcal{K}_l(\boldsymbol{A}, \boldsymbol{r}^{(0)})$ *($\rightsquigarrow$ minimum residual method),*

then there exists a uniquely defined iterate $\boldsymbol{z}^{(l)}$ *of the form (2.30) for which the residual* $\boldsymbol{r}^{(l)}$ *satisfies (2.31).*

Various Krylov subspace methods arose from different possibilities of the subspace $\mathcal{L}_l$ as well as from the way of preconditioning. We are going to recall the most common Krylov subspace approaches further below. But first, we want to give an explanation for the particular construction of the Krylov subspace in (2.32). In particular, one might ask why do we build the Krylov subspace with the vector $\boldsymbol{r}^{(0)}$.

For this purpose, let us consider for a moment the Krylov subspace

$$\mathcal{K}_l(\boldsymbol{A}, \boldsymbol{v}) = \operatorname{span}\left\{\boldsymbol{v}, \boldsymbol{A}\boldsymbol{v}, \boldsymbol{A}^2\boldsymbol{v}, \dots, \boldsymbol{A}^{l-1}\boldsymbol{v}\right\}$$

with an arbitrary given vector $\boldsymbol{v} \in \mathbb{R}^m$. For $\boldsymbol{k}^{(l)} \in \mathcal{K}_l(\boldsymbol{A}, \boldsymbol{v})$, we can write

$$\boldsymbol{k}^{(l)} = \sum_{j=0}^{l-1} \alpha_j \boldsymbol{A}^j \boldsymbol{v} = q_{l-1}(\boldsymbol{A})\boldsymbol{v}, \tag{2.33}$$

where $\alpha_j \in \mathbb{R}$, $j = 0, \dots, l-1$, and $q_{l-1}(t) = \sum_{j=0}^{l-1} \alpha_j t^j$ is a real polynomial of degree $l-1$. Now we make use of the Cayley-Hamilton theorem, which states that every square matrix satisfies its own characteristic equation. Formally, this means $c_{\boldsymbol{A}}(\boldsymbol{A}) = 0$, where $c_{\boldsymbol{A}}(t)$ is the characteristic polynomial of $\boldsymbol{A}$. One can show that

$$c_{\boldsymbol{A}}(\boldsymbol{A}) = (-1)^m \boldsymbol{A}^m + \dots - \left(\sum_{i=1}^{m} \prod_{j=1, j\neq i}^{m} \lambda_j\right) \boldsymbol{A} + \det(\boldsymbol{A})\boldsymbol{I}, \tag{2.34}$$

where λ_j, $j = 1, \dots, m$, are the eigenvalues of $\boldsymbol{A}$. The nonsingularity of $\boldsymbol{A}$ implies $\det(\boldsymbol{A}) \neq 0$. Hence, we can multiply (2.34) by $\det(\boldsymbol{A})^{-1}\boldsymbol{A}^{-1}$ and rearrangements yield

$$\det(\boldsymbol{A})^{-1}\left[(-1)^{m+1}\boldsymbol{A}^{m-1} + \ldots + \left(\sum_{i=1}^{m} \prod_{j=1, j\neq i}^{m} \lambda_j\right)\boldsymbol{I}\right] = \boldsymbol{A}^{-1}.$$

Thus, we have

$$\boldsymbol{A}^{-1} = q_{m-1}(\boldsymbol{A})$$

and the exact solution $\boldsymbol{z}$ of $\boldsymbol{A}\boldsymbol{z} = \boldsymbol{b}$ can be written as

$$\boldsymbol{z} = \boldsymbol{A}^{-1}\boldsymbol{b} = q_{m-1}(\boldsymbol{A})\boldsymbol{b}.$$

In general, we include an initial guess $\boldsymbol{z}^{(0)}$ as stated in (2.30). Then,

$$\boldsymbol{z} - \boldsymbol{z}^{(0)} = \boldsymbol{A}^{-1}\boldsymbol{b} - \boldsymbol{A}^{-1}(\boldsymbol{b} - \boldsymbol{r}^{(0)}) = \boldsymbol{A}^{-1}\boldsymbol{r}^{(0)} = q_{m-1}(\boldsymbol{A})\boldsymbol{r}^{(0)},$$

and a comparison with (2.33) shows that $\boldsymbol{v} = \boldsymbol{r}^{(0)}$ is a natural choice. This answers the question about the choice of the vector $\boldsymbol{r}^{(0)}$ in the Krylov subspace (2.32). During the explanation for this issue, another question might arise: How do we choose the sequence of polynomials $\{q_l\}_{l\in\mathbb{N}_0}$ such that the approximations

$$\boldsymbol{z}^{(l)} = \boldsymbol{z}^{(0)} + q_{l-1}(\boldsymbol{A})\boldsymbol{r}^{(0)} \in \boldsymbol{z}^{(0)} + \mathcal{K}_l(\boldsymbol{A}, \boldsymbol{r}^{(0)})$$

are successively closer to the exact solution $\boldsymbol{z}$? Will will discuss this in a moment. But first, let us express the residuum $\boldsymbol{r}^{(l)}$ as well as the error $\boldsymbol{e}^{(l)} = \boldsymbol{z} - \boldsymbol{z}^{(l)}$ in terms of the polynomial $q_l(\boldsymbol{A})$. The residuum in (2.29) can be written as

$$\boldsymbol{r}^{(l)} = \boldsymbol{r}^{(0)} - \boldsymbol{A}q_{l-1}(\boldsymbol{A})\boldsymbol{r}^{(0)} = (\boldsymbol{I} - \boldsymbol{A}q_{l-1}(\boldsymbol{A}))\boldsymbol{r}^{(0)} = p_l(\boldsymbol{A})\boldsymbol{r}^{(0)},$$

where $p_l(t) = 1 - tq_{l-1}(t)$ is a polynomial of degree l with $p_l(0) = 1$. Regarding the error, we first rewrite (2.29) in another way:

$$\boldsymbol{r}^{(0)} = \boldsymbol{A}\boldsymbol{z} - \boldsymbol{A}\boldsymbol{z}^{(0)} = \boldsymbol{A}\boldsymbol{e}^{(0)}. \tag{2.35}$$

Now, the error can be formulated as

$$\begin{aligned}\boldsymbol{e}^{(l)} &= \boldsymbol{z} - \boldsymbol{z}^{(0)} - (\boldsymbol{z}^{(l)} - \boldsymbol{z}^{(0)}) = \boldsymbol{e}^{(0)} - q_{l-1}(\boldsymbol{A})\boldsymbol{r}^{(0)} \overset{(2.35)}{=} \boldsymbol{e}^{(0)} - \boldsymbol{A}q_{l-1}(\boldsymbol{A})\boldsymbol{e}^{(0)}\\ &= (\boldsymbol{I} - \boldsymbol{A}q_{l-1}(\boldsymbol{A}))\boldsymbol{e}^{(0)} = p_l(\boldsymbol{A})\boldsymbol{e}^{(0)}.\end{aligned}$$

The first class of Krylov subspace solvers that we are going to briefly summarize are the minimizing solvers. This means, they either minimize the error or the residuum in a certain norm. In other words, the polynomials $\{q_l\}_{l\in\mathbb{N}_0}$ are constructed such that these quantities will be minimized.

We start with the conjugate gradient method (CG), which was developed independently and in different versions by Lanczos [115] and Hestenes and Stiefel [90] in 1952. This is the state-of-the-art solver for symmetric positive definite matrices $\boldsymbol{A}$. It is based on the subspace of constraints $\mathcal{L}_l = \mathcal{K}_l(\boldsymbol{A}, \boldsymbol{r}^{(0)})$. CG minimizes the $\boldsymbol{A}$-norm of the error $\boldsymbol{e}^{(l)}$, i.e.,

$$||\boldsymbol{e}^{(l)}||_{\boldsymbol{A}} = \min_{p_l\in\Pi_l, p_l(0)=1} ||p_l(\boldsymbol{A})\boldsymbol{e}^{(0)}||_{\boldsymbol{A}}. \tag{2.36}$$

This expression can be simplified by utilizing the fact that $\boldsymbol{A}$ is symmetric. This implies that $\mathbb{R}^m$ has a basis of orthonormal eigenvectors of $\boldsymbol{A}$. Thus, we can write the error in terms of this basis:

$$\boldsymbol{e}^{(0)} = \sum_{j=1}^{m} \beta_j \boldsymbol{v}_j, \tag{2.37}$$

where $\beta_j \in \mathbb{R}$ and $\boldsymbol{A}\boldsymbol{v}_j = \lambda_j \boldsymbol{v}_j$. Thus, we can simplify (2.36) to

$$\begin{aligned}
||\boldsymbol{e}^{(l)}||_{\boldsymbol{A}} &= \min_{p_l \in \Pi_l,\, p_l(0)=1} \left\| \sum_{j=1}^{m} \beta_j p_l(\boldsymbol{A}) \boldsymbol{v}_j \right\|_{\boldsymbol{A}} \\
&= \min_{\alpha_i \in \mathbb{R},\, i=0,\ldots,l-1} \left\| \sum_{j=1}^{m} \beta_j \left(\boldsymbol{I} - \boldsymbol{A} \sum_{i=0}^{l-1} \alpha_i \boldsymbol{A}^i \right) \boldsymbol{v}_j \right\|_{\boldsymbol{A}} \\
&= \min_{\alpha_i \in \mathbb{R},\, i=0,\ldots,l-1} \left\| \sum_{j=1}^{m} \beta_j \left(\boldsymbol{v}_j - \sum_{i=0}^{l-1} \alpha_i \boldsymbol{A}^{i+1} \boldsymbol{v}_j \right) \right\|_{\boldsymbol{A}} \\
&= \min_{\alpha_i \in \mathbb{R},\, i=0,\ldots,l-1} \left\| \sum_{j=1}^{m} \beta_j \left(\boldsymbol{v}_j - \sum_{i=0}^{l-1} \alpha_i \lambda_j^{i+1} \boldsymbol{v}_j \right) \right\|_{\boldsymbol{A}} \\
&= \min_{\alpha_i \in \mathbb{R},\, i=0,\ldots,l-1} \left\| \sum_{j=1}^{m} \beta_j \left(\boldsymbol{v}_j - \lambda_j \sum_{i=0}^{l-1} \alpha_i \lambda_j^{i} \boldsymbol{v}_j \right) \right\|_{\boldsymbol{A}} \\
&= \min_{p_l \in \Pi_l,\, p_l(0)=1} \left\| \sum_{j=1}^{m} \beta_j p_l(\lambda_j) \boldsymbol{v}_j \right\|_{\boldsymbol{A}} \\
&\leq \min_{p_l \in \Pi_l,\, p_l(0)=1} \left\| \max_j |p_l(\lambda_j)| \sum_{j=1}^{m} \beta_j \boldsymbol{v}_j \right\|_{\boldsymbol{A}} \\
&= \min_{p_l \in \Pi_l,\, p_l(0)=1} \max_j |p_l(\lambda_j)| \, ||e_0||_{\boldsymbol{A}}.
\end{aligned} \tag{2.38}$$

If we know eigenvalue bounds of the form $\lambda_j \in [a, b]$, $j = 1, \ldots, m$, with $a > 0$, then the bound (2.38) becomes

$$\frac{||\boldsymbol{e}^{(l)}||_{\boldsymbol{A}}}{||\boldsymbol{e}^{(0)}||_{\boldsymbol{A}}} \leq \min_{p_l \in \Pi_l,\, p_l(0)=1} \max_{z \in [a,b]} |p_l(z)|. \tag{2.39}$$

The polynomial, which achieves this minimization, is the scaled and shifted Chebyshev polynomial. We will not discuss this polynomial further and refer the reader to [133]. Of course, the best estimate is given if we know exactly the minimum and maximum eigenvalue of $\boldsymbol{A}$. The bound (2.38) gives a nice intuition about sufficient conditions for fast convergence rates. Imagine a symmetric positive definite matrix $\boldsymbol{A} \in \mathbb{R}^{m \times m}$ which has only two distinct eigenvalues a and b. Then, we can construct a polynomial $p_2(t)$ of degree two that satisfies $p_2(a) = 0, p_2(b) = 0$ and $p_2(0) = 1$. Then, the bound (2.38) implies that CG will converge to the exact solution in at most two iterations. This gives us an idea when we can expect CG to be a powerful solver: If $\boldsymbol{A}$ is symmetric positive definite and has only few number of distinct eigenvalues, then the approximations $\boldsymbol{z}^{(l)}$ converge quickly to $\boldsymbol{z}$. In praxis, we usually do not have knowledge about the exact eigenvalues of a matrix. One could compute them but this requires in general much more work than solving the linear system $\boldsymbol{Az} = \boldsymbol{b}$. In praxis, we can replace the knowledge about the location of exact eigenvalues by eigenvalue clusters. Hence, if $\boldsymbol{A}$ is symmetric positive definite and has only few number of eigenvalue clusters, then the approximations $\boldsymbol{z}^{(l)}$ converge quickly to $\boldsymbol{z}$. An important note is that the concept of eigenvalue clustering forms the basic methodology of Krylov subspace solvers to converge fast. Further below, we will recall some other Krylov subspace solvers. Their applicability differs for instance in the assumptions on $\boldsymbol{A}$ or in the choice of the subspace $\mathcal{L}_l$. Moreover, some of them are not constructed based on a minimization criterion of the form (2.38). However in praxis, a small number of eigenvalue clusters usually results in fast convergence rates for any Krylov subspace method.

Another criterion for the convergence behavior makes use of the condition number of $\boldsymbol{A}$.

Theorem 2.38 ([63, p. 75])**.** *After l steps of the CG method, the iteration error $\boldsymbol{e}^{(l)}$ satisfies the bound*

$$\frac{\|\boldsymbol{e}^{(l)}\|_{\boldsymbol{A}}}{\|\boldsymbol{e}^{(0)}\|_{\boldsymbol{A}}} \leq 2\left(\frac{\sqrt{\kappa(\boldsymbol{A})}-1}{\sqrt{\kappa(\boldsymbol{A})}+1}\right)^{l}.$$

Hence, if $\boldsymbol{A}$ has a small condition number, then CG will converge quickly. The converse does not have to be true. For instance, imagine a matrix which has a large condition number but only a few distinct eigenvalues. Then, the tighter eigenvalue bound (2.38) implies convergence after a few iterations.

For an algorithm of CG, we refer to [63, p. 72]. The costs of CG per iteration add up to two inner products, three vector updates, and one matrix vector multiplication. CG requires to store only four vectors. We close the short discussion about CG here. CG assumes a symmetric positive definite coefficient matrix. For many applications, in particular in most of the problems of this thesis, this is not given. Hence, we want to briefly touch more general Krylov subspace solvers. The minimum residual method (MINRES) was introduced by Paige and Saunders [129] in 1975. It uses $\mathcal{L}_l = \boldsymbol{A}\mathcal{K}_l(\boldsymbol{A}, \boldsymbol{r}^{(0)})$ as subspace of constraints. In contrast to CG, MINRES only assumes a symmetric coefficient matrix $\boldsymbol{A}$. Moreover, it does not minimize the error but the 2-norm of the residuum:

$$\frac{\|\boldsymbol{r}^{(l)}\|}{\|\boldsymbol{r}^{(0)}\|} \leq \min_{p_l \in \Pi_l,\, p_l(0)=1} \max_{z \in [a,b]} |p_l(z)|, \tag{2.40}$$

where the eigenvalues λ_j of $\boldsymbol{A}$ satisfy $\lambda_j \in [a,b]$ for all $j = 1, \ldots, m$. For an algorithm of MINRES, we refer to [63, p. 87]. The costs of MINRES per iteration add up to two inner products, five vector updates, and one matrix vector multiplication. MINRES requires to store only six vectors.

Up to this point, we have considered Krylov subspace methods which assume a symmetric coefficient matrix. Now, let us generalize this further. In the following, $\boldsymbol{A} \in \mathbb{R}^{m \times m}$ is an arbitrary nonsingular matrix. For this general case, there are two kinds of Krylov subspace solvers: First, we have minimizing solvers, which satisfy a generalized version of the optimality condition (2.40). However, they suffer from increasing costs per iteration. The second class requires a fixed amount of computational work per iteration. However, this class forgoes the optimality condition. The generalized minimum residual method (GMRES) and biconjugate gradient method (BiCG) are the two analogous methods to MINRES and CG for nonsymmetric matrices. GMRES was developed by Saad and Schultz [137] in 1986. It uses $\mathcal{L}_l = \boldsymbol{A}\mathcal{K}_l(\boldsymbol{A}, \boldsymbol{r}^{(0)})$ as subspace of constraints. Similar as MINRES, the bounds on the norm of the residuals are derived from the optimality condition.

Theorem 2.39 ([63, p. 169])**.** *Let $\boldsymbol{z}^{(l)}$ denote the lth iterate generated after l steps of GMRES with residual $\boldsymbol{r}^{(l)}$. If $\boldsymbol{A}$ is diagonalizable, that is, $\boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{-1}$, where $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \dots, \lambda_m)$ is the diagonal matrix of eigenvalues of $\boldsymbol{A}$, and $\boldsymbol{V}$ is the matrix whose columns are the eigenvectors, then*

$$\frac{\|\boldsymbol{r}^{(l)}\|}{\|\boldsymbol{r}^{(0)}\|} \leq \kappa(\boldsymbol{V}) \min_{p_l \in \Pi_l,\, p_l(0)=1} \max_j |p_l(\lambda_j)|.$$

If in addition S is any set that contains the eigenvalues of $\boldsymbol{A}$, then

$$\frac{\|\boldsymbol{r}^{(l)}\|}{\|\boldsymbol{r}^{(0)}\|} \leq \kappa(\boldsymbol{V}) \min_{p_l \in \Pi_l,\, p_l(0)=1} \max_{\lambda \in S} |p_l(\lambda)|.$$

Compared to the case of a symmetric matrix in (2.39), the eigenvectors of $\boldsymbol{A}$ enter the bounds in the nonsymmetric case. However, the eigenvalue clustering still plays the crucial role for fast convergence rates. One drawback of GMRES are the increased work and storage requirements, which grow like $\mathcal{O}(lm)$. This can be improved for example by restarting techniques; see, e.g., [122]. However, it is usually not clear when to restart. For an algorithm of GMRES, we refer to [63, p. 168]. The costs of GMRES per iteration add up to one matrix vector multiplication but many inner products. As already hinted, we have to store the whole basis for the search subspace $\mathcal{K}_l$.

BiCG was originally introduced in 1952 by Lanczos [115] and later in 1976 rediscovered by Fletcher [69]. It uses $\mathcal{L}_l = \mathcal{K}_l(\boldsymbol{A}^T, \boldsymbol{r}^{(0)})$ as subspace of constraints. BiCG belongs to the class of Krylov subspace solvers which requires a fixed amount of computational work per iteration. Moreover, it has fixed storage requirements. BiCG requires two matrix vector products (one with $\boldsymbol{A}$ and one with $\boldsymbol{A}^T$), two inner products, and five vector updates. As a drawback, there is essentially no convergence analysis available. However, in praxis the eigenvalue clustering still plays the crucial role as we will see throughout this thesis. Another drawback is the possibility of breakdowns, i.e., it may happen that no iterate $\boldsymbol{z}^{(l)}$ satisfying both, (2.30) and (2.31), exists. Variants of BiCG have been developed, e.g., the biconjugate gradient stabilized method (BiCGstab) of van der Vorst [147]. In praxis such breakdowns are rare. However, near-breakdowns may cause irregular convergence. In our experience with the computational results in this thesis, we have not encountered any problems due to breakdowns. Hence, we choose BiCG or BiCGstab as the numerical solver for the nonsymmetric linear systems in this thesis. The procedure taken from [63, p. 173] is illustrated in Algorithm 2.1.

We finish the presentation about the variants of Krylov subspace solvers. For a deeper insight, we refer the reader to [136] and references therein. The important

Algorithm 2.1: The biconjugate gradient method

```
Choose $\boldsymbol{z}^{(0)}$, compute $\boldsymbol{r}^{(0)} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{z}^{(0)}$, set $\boldsymbol{p}^{(0)} = \boldsymbol{r}^{(0)}$
Choose $\hat{\boldsymbol{r}}^{(0)}$ such that $(\boldsymbol{r}^{(0)}, \hat{\boldsymbol{r}}^{(0)}) \neq 0$
Set $\hat{\boldsymbol{p}}^{(0)} = \hat{\boldsymbol{r}}^{(0)}$
for $l = 0, 1, 2, \dots$ do
    $\alpha_l = \frac{(\hat{\boldsymbol{r}}^{(l)}, \boldsymbol{r}^{(l)})}{(\hat{\boldsymbol{p}}^{(l)}, \boldsymbol{A}\boldsymbol{p}^{(l)})}$
    $\boldsymbol{z}^{(l+1)} = \boldsymbol{z}^{(l)} + \alpha_l \boldsymbol{p}^{(l)}$
    $\boldsymbol{r}^{(l+1)} = \boldsymbol{r}^{(l)} - \alpha_l \boldsymbol{A}\boldsymbol{p}^{(l)}$
    $\hat{\boldsymbol{r}}^{(l+1)} = \hat{\boldsymbol{r}}^{(l)} - \alpha_l \boldsymbol{A}^T \hat{\boldsymbol{p}}^{(l)}$
    <Test for convergence >
    $\beta_{(l)} = \frac{(\hat{\boldsymbol{r}}^{(l+1)}, \boldsymbol{r}^{(l+1)})}{(\hat{\boldsymbol{r}}^{(l)}, \boldsymbol{r}^{(l)})}$
    $\boldsymbol{p}^{(l+1)} = \boldsymbol{r}^{(l+1)} + \beta_l \boldsymbol{p}^{(l)}$
    $\hat{\boldsymbol{p}}^{(l+1)} = \hat{\boldsymbol{r}}^{(l+1)} + \beta_l \hat{\boldsymbol{p}}^{(l)}$
end
```

message from the part above is the eigenvalue clustering, which typically results in a fast convergence rate for any Krylov subspace solver. Of course, the matrices coming from PDEs and applications usually do not have neither a nice eigenvalue structure nor an acceptable small condition number. Hence, solving the resulting linear systems with a Krylov subspace solver without any acceleration results in bad convergence behaviors. This is where preconditioning techniques come in.

2.3.2 Preconditioning

We have already motivated the idea of preconditioning in the introduction part of Section 2.3. We have seen that classical iterative methods like the Jacobi iteration can be used as a preconditioner. The aim of a preconditioner is to enhance the convergence of the iterative solver. In our case, we want to accelerate the speed of convergence of Krylov subspace solvers. The basic idea is to construct a nonsingular matrix $\boldsymbol{P} \in \mathbb{R}^{m \times m}$ and solve

$$\boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{z} = \boldsymbol{P}^{-1}\boldsymbol{b} \tag{2.41}$$

instead of $\boldsymbol{A}\boldsymbol{z} = \boldsymbol{b}$. In order for $\boldsymbol{P}$ to be efficient, it should satisfy the following conditions:

- $\boldsymbol{P}$ should approximate $\boldsymbol{A}$ and
- the action of $\boldsymbol{P}^{-1}$ should require little work.

The construction process of $\boldsymbol{P}$ should incorporate the goal of eigenvalue clustering. That means, $\boldsymbol{P}^{-1}\boldsymbol{A}$ is aimed to have a few number of eigenvalues or eigenvalue clusters. This typically results in outstanding performances of Krylov subspace solvers. The algorithms of Krylov subspace solvers have been modified to incorporate a preconditioner. CG and MINRES need a symmetric positive definite preconditioner in order to guarantee the inherent symmetry. GMRES and BiCG impose, besides the nonsingularity, no additional requirements on $\boldsymbol{P}$. As an example, Algorithm 2.2 depicts the preconditioned BiCG. The preconditioned system in (2.41) is called

Algorithm 2.2: The preconditioned biconjugate gradient method

```
Choose $\boldsymbol{z}^{(0)}$, compute $\boldsymbol{r}^{(0)} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{z}^{(0)}$
Choose $\hat{\boldsymbol{r}}^{(0)}$ such that $(\boldsymbol{r}^{(0)}, \hat{\boldsymbol{r}}^{(0)}) \neq 0$
Solve $\boldsymbol{P}\boldsymbol{y}^{(0)} = \boldsymbol{r}^{(0)}$ and $\boldsymbol{P}^T\hat{\boldsymbol{y}}^{(0)} = \hat{\boldsymbol{r}}^{(0)}$
Set $\boldsymbol{p}^{(0)} = \boldsymbol{y}^{(0)}$ and $\hat{\boldsymbol{p}}^{(0)} = \hat{\boldsymbol{z}}^{(0)}$
for $l = 0, 1, 2, \ldots$ do
    $\alpha_l = \frac{(\hat{\boldsymbol{r}}^{(l)}, \boldsymbol{y}^{(l)})}{(\hat{\boldsymbol{p}}^{(l)}, \boldsymbol{A}\boldsymbol{p}^{(l)})}$
    $\boldsymbol{z}^{(l+1)} = \boldsymbol{z}^{(l)} + \alpha_l \boldsymbol{p}^{(l)}$
    $\boldsymbol{r}^{(l+1)} = \boldsymbol{r}^{(l)} - \alpha_l \boldsymbol{A}\boldsymbol{p}^{(l)}$
    $\hat{\boldsymbol{r}}^{(l+1)} = \hat{\boldsymbol{r}}^{(l)} - \alpha_l \boldsymbol{A}^T\hat{\boldsymbol{p}}^{(l)}$
    Solve $\boldsymbol{P}\boldsymbol{y}^{(l+1)} = \boldsymbol{r}^{(l+1)}$ and $\boldsymbol{P}^T\hat{\boldsymbol{y}}^{(l+1)} = \hat{\boldsymbol{r}}^{(l+1)}$
    <Test for convergence >
    $\beta_{(l)} = \frac{(\hat{\boldsymbol{r}}^{(l+1)}, \boldsymbol{y}^{(l+1)})}{(\hat{\boldsymbol{r}}^{(l)}, \boldsymbol{y}^{(l)})}$
    $\boldsymbol{p}^{(l+1)} = \boldsymbol{y}^{(l+1)} + \beta_l \boldsymbol{p}^{(l)}$
    $\hat{\boldsymbol{p}}^{(l+1)} = \hat{\boldsymbol{y}}^{(l+1)} + \beta_l \hat{\boldsymbol{p}}^{(l)}$
end
```

left-preconditioned system. There is also the right-preconditioned system

$$\boldsymbol{A}\boldsymbol{P}^{-1}\hat{\boldsymbol{z}} = \boldsymbol{b}, \quad \hat{\boldsymbol{z}} = \boldsymbol{P}\boldsymbol{z},$$

and the centrally preconditioned system

$$\boldsymbol{P}_1^{-1}\boldsymbol{A}\boldsymbol{P}_2^{-1}\hat{\boldsymbol{z}} = \boldsymbol{P}_1^{-1}\boldsymbol{b}, \quad \hat{\boldsymbol{z}} = \boldsymbol{P}_2\boldsymbol{z}.$$

In this thesis, we will apply left preconditioning. The development of efficient preconditioners often depends on the structure and properties of $\boldsymbol{A}$, which in turn depends on the underlying PDE or problem. This is in particular the case for saddle point systems. If we only consider the theoretical side of the construction, there are preconditioners which are optimal for a wide range of problems. Optimal means that they accelerate the convergence behavior of Krylov subspace solvers to a small number of iterations. However, these theoretically optimal preconditioners have crucial drawbacks in praxis. They are usually built on inverses of large and dense matrices. Hence, they typically cannot be explicitly used and clever practical approximations have to be designed. Nevertheless, these theoretical preconditioners are of high importance and form the basis for the development of practical solvers. Before we are going to introduce a special class of such theoretical optimal preconditioners, we briefly want to give another example of a preconditioner. We have already learned about the Jacobi iteration as one example of a preconditioner. A powerful technique, in particular for elliptic problems, is the multigrid method (MG).

Multigrid methods

It is known that classical iterative solvers like the Jacobi or Gauss-Seidel iteration can damp high frequency errors rapidly. However, after the error is smoothed out their convergence rates decelerate. Typically, they have difficulties in damping the low frequency errors without special relaxation techniques. But the low frequency errors can be generally seen as high frequency errors on a coarser mesh. Hence, if we would have a proper mapping from the current mesh to a coarser mesh, we could further effectively damp the error via a standard iterative solver. This procedure could be repeated again and again until we reach a satisfied coarse grid on which we are able to efficiently solve a reduced linear system. What remains is to interpolate the calculated solution back to the starting grid and update the approximate solution. MGs exploit this knowledge and employ grids of different mesh sizes. Under a careful construction, their convergence rates are independent of the problem size m. In this sense, they provide an optimal solver. Basically, MGs can be divided into two classes. Geometric multigrid methods [151, 88] require information of the underlying spatial mesh of the discretized problem. The algebraic multigrid method (AMG) [134, 68] works in a purely algebraic framework. That means, they only use the knowledge of the underlying matrix. Hence, they work well even for complicated geometries and meshes. For a detailed description to MG and its origins, we refer to [40]. MG is a fast and efficient method to solve many classes of problems. However, its implementation can be quite complex. In this thesis, we will use AMG as part of our developed preconditioners.

Now, we want to turn to a special class of theoretical optimal preconditioners which form the basis for this thesis.

2.3.3 Saddle point preconditioners

In Section 2.2.4, we have introduced saddle point type matrices since they form the core theme of this thesis. This section considers saddle point type linear systems of the form

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}_1^T \\ \boldsymbol{B}_2 & -\boldsymbol{C} \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{w} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{g} \end{bmatrix}, \tag{2.42}$$

shortly written as $\boldsymbol{\mathcal{A}}\boldsymbol{z} = \boldsymbol{b}$, with $\boldsymbol{A} \in \mathbb{R}^{m\times m}$, $\boldsymbol{B}_1, \boldsymbol{B}_2 \in \mathbb{R}^{p\times m}$, $\boldsymbol{C} \in \mathbb{R}^{p\times p}$, and $m \geq p$. In particular, we recall theoretical optimal preconditioners for such kind of problems. The focus lies on preconditioners involving the Schur complement $\boldsymbol{S} = -(\boldsymbol{C} + \boldsymbol{B}_2\boldsymbol{A}^{-1}\boldsymbol{B}_1)$. In the following, we assume that $\boldsymbol{\mathcal{A}}$ as well as $\boldsymbol{A}$ and $\boldsymbol{S}$ are nonsingular.

Block diagonal preconditioners

The basic block diagonal preconditioner is given by

$$\boldsymbol{\mathcal{P}}_D = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{S} \end{bmatrix}. \tag{2.43}$$

Bank, Welfert, and Yserentant [9], Kuznetsov [113], as well as Murphy, Golub, and Wathen [124] proposed it for the case $\boldsymbol{C} = \boldsymbol{0}$. In that case, left preconditioning of $\boldsymbol{\mathcal{A}}$ with $\boldsymbol{\mathcal{P}}_D$ results in the matrix

$$\boldsymbol{\mathcal{T}}_D = \boldsymbol{\mathcal{P}}_D^{-1}\boldsymbol{\mathcal{A}} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{A}^{-1}\boldsymbol{B}_1^T \\ -\boldsymbol{S}^{-1}\boldsymbol{B}_2 & \boldsymbol{0} \end{bmatrix}. \tag{2.44}$$

The matrix $\boldsymbol{\mathcal{T}}_D$ is nonsingular by assumption, diagonalizable, and has only three distinct eigenvalues, which are 1, $\frac{1}{2}(1+\sqrt{5})$, and $\frac{1}{2}(1-\sqrt{5})$. That means that any Krylov subspace method with an optimality or Galerkin property, e.g., GMRES [137] or BiCG [69], will converge in at most three iterations. Further, Ipsen [97] generalized these results to the case $\boldsymbol{C} \neq \mathbf{0}$.

Block triangular preconditioners
Block triangular preconditioners of the form

$$\mathcal{P}_T = \begin{bmatrix} \boldsymbol{A} & \mathbf{0} \\ \boldsymbol{B}_2 & -\boldsymbol{S} \end{bmatrix} \tag{2.45}$$

were first considered by Bramble and Pasciak [39]. They studied saddle point systems with $\boldsymbol{A}$ being positive definite, $\boldsymbol{C}$ being positive semidefinite, and $\boldsymbol{B}_1 = \boldsymbol{B}_2 = \boldsymbol{B}$.

The block diagonal preconditioner $\mathcal{P}_D$ can be presented in block triangular form in two other ways:

$$\mathcal{P}_{-T} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}_1^T \\ \mathbf{0} & -\boldsymbol{S} \end{bmatrix} \quad \text{and} \quad \mathcal{P}_{+T} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}_1^T \\ \mathbf{0} & +\boldsymbol{S} \end{bmatrix}. \tag{2.46}$$

In the following, we use the notation

$$\mathcal{P}_{\pm T} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}_1^T \\ \mathbf{0} & \pm\boldsymbol{S} \end{bmatrix} \tag{2.47}$$

that combines both block tridiagonal preconditioners into one expression. Murphy, Golub, and Wathen [124] proposed them for the case $\boldsymbol{C} = \mathbf{0}$. In that case, right preconditioning of $\mathcal{A}$ with $\mathcal{P}_{\pm T}$ results in the matrix

$$\mathcal{T}_{\pm T} = \mathcal{A}\mathcal{P}_{\pm T}^{-1} = \begin{bmatrix} \boldsymbol{I} & \mathbf{0} \\ \boldsymbol{B}_2\boldsymbol{A}^{-1} & \pm\boldsymbol{I} \end{bmatrix}. \tag{2.48}$$

The matrices $\mathcal{T}_{\pm T}$ are nonsingular by assumption. The preconditioned matrix $\mathcal{T}_{-T}$ is diagonalizable and has only the two distinct eigenvalues ±1. In contrast, $\mathcal{T}_{+T}$ has only a single eigenvalue of 1, but is not diagonalizable. The use of $\mathcal{T}_{\pm T}$ rather than $\mathcal{T}_D$ requires only one additional multiplication per iteration of a vector by $\boldsymbol{B}_1^T$. As in the diagonal case, Ipsen [97] generalized these results to the case $\boldsymbol{C} \neq \mathbf{0}$. Further, the same results apply for left preconditioning $\mathcal{T}_{\pm T} = \mathcal{P}_{\pm T}^{-1}\mathcal{A}$. Moreover, the results spread to the lower block triangular preconditioners

$$\begin{bmatrix} \boldsymbol{A} & \mathbf{0} \\ \boldsymbol{B}_2 & -\boldsymbol{S} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \boldsymbol{A} & \mathbf{0} \\ \boldsymbol{B}_2 & +\boldsymbol{S} \end{bmatrix}.$$

Block preconditioners
Axelsson and Neytcheva [4, Proposition 5.1] considered saddle point matrices of the form

$$\mathcal{A} = \begin{bmatrix} \boldsymbol{A} & -\alpha^2\boldsymbol{B}^T \\ \beta^2\boldsymbol{B} & \gamma^2\boldsymbol{A} \end{bmatrix}, \tag{2.49}$$

where $\boldsymbol{A} \in \mathbb{R}^{m\times m}$ is symmetric positive definite, the symmetric part $\frac{1}{2}(\boldsymbol{B} + \boldsymbol{B}^T)$ of $\boldsymbol{B} \in \mathbb{R}^{m\times m}$ is positive semidefinite, and α, β, and γ are some real constants. They derived an optimal preconditioner

$$\mathcal{P} = \begin{bmatrix} \boldsymbol{A} & -\alpha^2\boldsymbol{B}^T \\ \beta^2\boldsymbol{B} & \gamma^2\boldsymbol{A} + \alpha\beta\gamma(\boldsymbol{B} + \boldsymbol{B}^T) \end{bmatrix}. \tag{2.50}$$

The eigenvalues λ of the generalized eigenvalue problem

$$\mathcal{A}\begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \end{bmatrix} = \lambda \mathcal{P}\begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \end{bmatrix}$$

satisfy

$$\lambda \in \begin{cases} [0.5, 1], \\ \{1\} \text{ for } \boldsymbol{v}_2 \in \ker(\boldsymbol{B} + \boldsymbol{B}^T). \end{cases}$$

This preconditioner has been used for smooth Cahn–Hilliard equations by Boyanova et al. [35, 37, 36, 3, 38]. Moreover, they showed in [35, p. 21] its connection to the block diagonal and block triangular preconditioner discussed above. Their implementations turn out to contain the same steps and the computational amount of work is almost the same.

The basis for most of the problems in this thesis is the lower block triangular preconditioner $\mathcal{P}_T$ in (2.45). In some simpler cases, we make use of its block diagonal version. Both types need the application of the inverse of two matrices. These are the $(1, 1)$ block $\boldsymbol{A}$ as well as the Schur complement $\boldsymbol{S} = -(\boldsymbol{C} + \boldsymbol{B}_2\boldsymbol{A}^{-1}\boldsymbol{B}_1^T)$. The crucial part is the inversion of $\boldsymbol{S}$, since it is usually large and dense. Hence, it typically cannot be explicitly used and clever practical approximations have to be designed. Depending on the properties of $\boldsymbol{A}$, it might also not be an easy task to apply the action of its inverse. Hence, it is our task to develop practical nonsingular approximations $\boldsymbol{A}_0 \approx \boldsymbol{A}$ and $\boldsymbol{S}_0 \approx \boldsymbol{S}$. This results in the two preconditioners

$$\mathcal{P}_{D_0} = \begin{bmatrix} \boldsymbol{A}_0 & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{S}_0 \end{bmatrix} \quad \text{and} \quad \mathcal{P}_{T_0} = \begin{bmatrix} \boldsymbol{A}_0 & \boldsymbol{0} \\ \boldsymbol{B}_2 & -\boldsymbol{S}_0 \end{bmatrix} \tag{2.51}$$

that we are going to design in this thesis. The implementation that performs the action of the inverse of $\mathcal{P}_{D_0}$ requires the following two steps. The solution of the problem $\mathcal{P}_{D_0}[\boldsymbol{y}_1^T, \boldsymbol{y}_2^T]^T = [\boldsymbol{g}_1^T, \boldsymbol{g}_2^T]^T$ requires:

1. Solve $\boldsymbol{A}_0\boldsymbol{y}_1 = \boldsymbol{g}_1$.

2. Solve $-\boldsymbol{S}_0\boldsymbol{y}_2 = \boldsymbol{g}_2$.

The solution of the problem $\mathcal{P}_{T_0}[\boldsymbol{y}_1^T, \boldsymbol{y}_2^T]^T = [\boldsymbol{g}_1^T, \boldsymbol{g}_2^T]^T$ requires one additional matrix vector multiplication:

1. Solve $\boldsymbol{A}_0\boldsymbol{y}_1 = \boldsymbol{g}_1$.

2. Solve $\boldsymbol{S}_0\boldsymbol{y}_2 = \boldsymbol{B}_2\boldsymbol{y}_1 - \boldsymbol{g}_2$.

In many problems discussed in this thesis, the $(1, 1)$ block $\boldsymbol{A}$ is formed either by diagonal, symmetric positive matrices or by discretizations of elliptic operators. Hence, in the former case we apply the action of the inverse of $\boldsymbol{A}$ by simple elementwise multiplications. For the latter case, we already know that AMG provides a powerful solver. Note that some other special cases for $\boldsymbol{A}$ arise, which need additional consideration as we will see later in the corresponding chapters. Our general procedure for the approximation of the Schur complement $\boldsymbol{S}$ originates in the work of Pearson and Wathen [130], who developed preconditioners for PDE-constrained optimization. Their matching strategy is the following: Construct a preconditioner of the form $\hat{\boldsymbol{S}} = \boldsymbol{S}_1\hat{\boldsymbol{A}}^{-1}\boldsymbol{S}_2$, which captures the exact Schur complement $\boldsymbol{S}$ as close as possible. Note that we need $\hat{\boldsymbol{S}}$ to be nonsingular. The block $\hat{\boldsymbol{A}}$ is usually a symmetric positive definite approximation of $\boldsymbol{A}$ if $\boldsymbol{A}$ does not already have these properties. For symmetric positive definite matrices $\boldsymbol{S}_1, \boldsymbol{S}_2$, an approximate inverse of $\boldsymbol{S}$ is given by $\hat{\boldsymbol{S}}^{-1} = \boldsymbol{S}_2^{-1}\hat{\boldsymbol{A}}\boldsymbol{S}_1^{-1}$. Suitable constructions of $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ are typically problem dependent as we will see in the course of this thesis. Finally, $\boldsymbol{S}_0$ is constructed as an approximation of $\hat{\boldsymbol{S}}$ which uses AMGs for the approximations of $\boldsymbol{S}_1^{-1}$ and $\boldsymbol{S}_2^{-1}$, i.e., $\boldsymbol{S}_0 = \mathrm{AMG}(\boldsymbol{S}_1)\hat{\boldsymbol{A}}^{-1}\mathrm{AMG}(\boldsymbol{S}_2)$.

The construction of a Schur complement approximation of the form $\boldsymbol{S}_1\hat{\boldsymbol{A}}^{-1}\boldsymbol{S}_2$ has been derived in another context by Bänsch, Morin, and Nochetto [10]. They developed operator preconditioners for a class of fourth-order problems. These include the smooth scalar Cahn–Hilliard equation for special time discretization schemes. In particular, they provided theoretical results on the spectrum of their preconditioners. In contrast, we derive our preconditioners for smooth Cahn–Hilliard problems from a different direction of view. This includes another theoretical perspective and a different way to prove their optimality. Moreover, in this thesis we extend this preconditioning strategy to nonsmooth Cahn–Hilliard problems and other Cahn–Hilliard applications.

The design, implementation, and numerical analysis of preconditioners of the form (2.51) tailored for Cahn–Hilliard problems are the focus of this thesis. In particular, the preconditioners are aimed to be robust with respect to parameter changes. We finish here the mathematical background setting. The next three chapters present three different Cahn–Hilliard problems. Each chapter starts with the problem formulation and a brief function space discussion. This is where we have to keep the functional analysis in mind. After proper discretizations, we end up with systems of linear equations where the numerical linear algebra part is required. In the following chapter, we start with the scalar Cahn–Hilliard equation.

Chapter 3

Scalar Cahn–Hilliard Equations

3.1 Introduction

We start with the study of two-component systems as introduced in Chapter 1.1. Imagine a molten binary alloy like iron and nickel inside the spatial domain $\Omega \subset \mathbb{R}^d$ with $d \in \{1, 2, 3\}$. The two pure phases are denoted by A (pure iron) and B (pure nickel). We are interested in the evolution of the two components or their mixture in the period $(0, T)$ with a fixed time $T > 0$. Hence, we have to take the time $t \in (0, T)$ and the spatial point $\mathbf{x} \in \Omega$ into account. Remember the phase variable $u = u(\mathbf{x}, t)$, which describes the difference of the local concentrations of both components. If $u(\mathbf{x}, t) \approx 1$, then only phase A (the pure phase A) is present at point $\mathbf{x}$ at time t. The case $u(\mathbf{x}, t) \approx -1$ means only phase B (the pure phase B) exists at point $\mathbf{x}$ at time t. Values of u between -1 and 1 represent mixed regions. In particular, these regions include the interfacial area. The interface is a small boundary layer that separates the pure phases A and B from each other. Hence, it acts as a diffuse phase transition. We can control its width via the model parameter $\varepsilon > 0$, which is again introduced in Equation (3.1) below.

The theory of Cahn and Hilliard [44] is based on the Ginzburg–Landau energy

$$\mathcal{E}(u) = \int_\Omega \frac{\varepsilon}{2}|\nabla u|^2 + \frac{1}{\varepsilon}\psi(u)\ \mathrm{d}\mathbf{x}. \tag{3.1}$$

An equilibrium profile of our considered mixture minimizes the Ginzburg–Landau energy (3.1) subject to the mass conservation

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_\Omega u\ \mathrm{d}\mathbf{x} = 0.$$

The parameter $\varepsilon > 0$ is proportional to the thickness of the interfacial region as mentioned above. The first part of (3.1) is large whenever u changes rapidly. Hence, its minimization gives rise to the interfacial area. The potential function $\psi \colon \mathbb{R} \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ in (3.1) gives rise to phase separation. It has two distinct minima, one for each of the two pure phases A and B. Hence, its minimization penalizes values away from the pure phases. As mentioned in Chapter 1, we consider potential functions of polynomial and obstacle type with a main focus on the latter. The former is the smooth double-well potential and is given as

$$\psi_{\text{pol}}(u) = \frac{1}{4}(u^2 - 1)^2. \tag{3.2}$$

The latter is the nonsmooth double-obstacle potential

$$\psi_{\text{obs}}(u) = \begin{cases} \frac{1}{2}(1-u^2) & |u| \leq 1, \\ \infty & |u| > 1. \end{cases} \tag{3.3}$$

As we will show in the course of this chapter, the solution of linear systems of the form $\boldsymbol{\mathcal{A}}\boldsymbol{z} = \boldsymbol{b}$ with a large, sparse, and symmetric matrix $\boldsymbol{\mathcal{A}}$ is at the heart of our method. They have the following saddle point structure

$$\boldsymbol{\mathcal{A}} = \begin{bmatrix} -\boldsymbol{A} & \boldsymbol{M} \\ \boldsymbol{M} & \boldsymbol{K} \end{bmatrix}$$

with $\boldsymbol{M} \in \mathbb{R}^{m \times m}$ being symmetric positive definite, $\boldsymbol{K} \in \mathbb{R}^{m \times m}$ being symmetric positive semidefinite, and $\boldsymbol{A} \in \mathbb{R}^{m \times m}$ being symmetric and possibly indefinite. Due to the possible indefiniteness of $\boldsymbol{A}$, a nonsymmetric Krylov subspace solver is our method of choice. The crucial parameters represented in $\boldsymbol{\mathcal{A}}$ are the spatial mesh size h, the time step size τ, the interface parameter ε, as well as the Moreau–Yosida regularization parameter c. We develop efficient preconditioners $\boldsymbol{\mathcal{P}}$ for the solution of the linear systems above. This is based on effective Schur complement approximations as well as (algebraic) multigrid solvers developed for elliptic systems [68, 136, 134]. In particular, our preconditioners behave independent of all crucial parameters. Moreover, we state a theoretical robustness proof for the smooth setting.

The structure of the chapter is as follows. The Cahn–Hilliard model is derived in Section 3.2. We first consider the smooth double-well potential (3.2), which leads to a fourth-order PDE. Then, we study the nonsmooth double-obstacle potential (3.3), which yields a variational inequality. Both formulations are discretized in time in Section 3.3. We focus on a fully implicit time-discrete scheme. This is due to accuracy reasons and its motivation is investigated later in the numerical results. Regarding the smooth setting, we proof the energy stability and uniqueness of the solution of our time discretization scheme under reasonable assumptions. Concerning the nonsmooth framework, we follow Hintermüller et al. [91] and extend the analysis therein. Note that Hintermüller et al. developed the mathematical theory for a semi-implicit time-discrete scheme for the Cahn–Hilliard variational inequality. In contrast, we focus on a fully implicit time-discrete scheme. We show that the time-discrete problem is equivalent to an optimal control problem with pointwise constraints on the control. In Section 3.4–3.5, we apply Hintermüller et al's function space-based algorithm. It combines a Moreau–Yosida regularization technique with an SSN method. The former handles the control constraints while the latter solves the optimality systems of the resulting subproblems. This method allows for a convergence analysis in function space [92, 146] for which one expects a mesh-independent behavior of the algorithm [93]. We derive the linear systems arising from the discretization using finite elements in Section 3.6. In Section 3.7, we analyze the linear systems and propose preconditioning strategies for the saddle point problems. We additionally introduce preconditioners for a smooth semi-implicit scheme derived in [67] and for a nonsmooth semi-implicit scheme derived in [91]. Section 3.8 illustrates the robustness of our preconditioners for both problem setups. In Section 3.9, we discuss alternative approaches. Section 3.10 summarizes our findings.

3.2 Derivation

There are two ways of deriving the Cahn–Hilliard equation. First, it can be derived as the H^{-1}-gradient flow of the Ginzburg-Landau energy (3.1); see, e.g. [20]. The second kind comes from the mass balance law; see, e.g. [57, 126]. We consider the latter case and briefly review the derivation of the Cahn–Hilliard equation. First of all, the smooth double-well potential (3.2) setting is used. Then, we go over to the nonsmooth double-obstacle potential (3.3) setting.

3.2.1 Smooth systems

In the following, we focus on the smooth double-well potential (3.2). We briefly derive the Cahn–Hilliard equation in the framework of [57]. For a more detailed and physical study, we refer to [53, 127]. We assume that the considered system is isothermal. The law of mass conservation is given as

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_R u\,\mathrm{d}\mathbf{x} = -\int_{\partial R} \mathbf{J}\cdot\mathbf{n}\,\mathrm{d}\mathbf{s}$$

for any subregion $R \subset \Omega$. Here, $\mathbf{J}$ denotes the mass flux. Due to Lemma 2.21, we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_R u\,\mathrm{d}\mathbf{x} = -\int_R \nabla\cdot\mathbf{J}\,\mathrm{d}\mathbf{x}.$$

Since R is fixed and arbitrary, we can derive $\partial_t u = -\nabla\cdot\mathbf{J}$. The mass flux is defined as $\mathbf{J} = -L(u)\nabla w$, where $L(u) \geq 0$ ($L(u) \not\equiv 0$) denotes the mobility function and w the chemical potential difference between the species A and B. The latter is defined via the variational derivative of $\mathcal{E}$ with respect to u (see Definition 2.20), whereby $\mathcal{U} = \mathcal{Y} = H^1(\Omega)$:

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}\eta}\mathcal{E}(u+\eta d) &= \lim_{\eta\to 0}\frac{\mathcal{E}(u+\eta d)-\mathcal{E}(u)}{\eta}\\
&= \lim_{\eta\to 0}\frac{1}{\eta}\int_\Omega \frac{\varepsilon}{2}|\nabla(u+\eta d)|^2 + \frac{1}{4\varepsilon}((u+\eta d)^2-1)^2 - \frac{\varepsilon}{2}|\nabla u|^2 - \frac{1}{4\varepsilon}(u^2-1)^2\,\mathrm{d}\mathbf{x}\\
&= \int_\Omega \frac{1}{\varepsilon}\left(u^3-u\right)d\,\mathrm{d}\mathbf{x} + \varepsilon\int_\Omega \nabla u\cdot\nabla d\,\mathrm{d}\mathbf{x}\\
&= \int_\Omega \left(\frac{1}{\varepsilon}\psi'_{\mathrm{pol}}(u) - \varepsilon\Delta u\right) d\,\mathrm{d}\mathbf{x} \qquad (3.4)\\
&= \int_\Omega w\,d\,\mathrm{d}\mathbf{x}.
\end{aligned}$$

The identity in (3.4) is supplemented with Lemma 2.21 together with the natural zero Neumann boundary condition $\nabla u\cdot\mathbf{n} = 0$ on $\partial\Omega$. Here, $\mathbf{n}$ is the unit normal vector to $\partial\Omega$ pointing outwards from Ω. Finally, we impose the mass conserving boundary condition

$$L(u)\nabla w\cdot\mathbf{n} = 0 \quad \text{on } \partial\Omega. \qquad (3.5)$$

In conclusion, we obtain the Cahn–Hilliard equation:

$$\partial_t u = \nabla \cdot (L(u)\nabla w), \tag{3.6}$$

$$w = -\varepsilon \Delta u + \varepsilon^{-1} \psi'_{\mathrm{pol}}(u), \tag{3.7}$$

$$\nabla u \cdot \mathbf{n} = L(u)\nabla w \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega. \tag{3.8}$$

This formulation states the Cahn–Hilliard problem in form of two second-order PDEs. Substituting Equation (3.7) into (3.6), results in the fourth-order formulation. Note also the high nonlinearity presented by the potential ψ_{pol}. Differentiating the energy $\mathcal{E}$ in (3.1) with respect to the time yields

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{E}(u) &= \int_\Omega \varepsilon \nabla u \cdot \nabla \partial_t u + \frac{1}{\varepsilon}\psi'_{\mathrm{pol}}(u)\,\partial_t u \,\,\mathrm{d}\mathbf{x} = \int_\Omega w\,\partial_t u \,\mathrm{d}\mathbf{x} = \int_\Omega w\,\nabla \cdot (L(u)\nabla w)\,\mathrm{d}\mathbf{x} \\ &= \int_{\partial\Omega} w\,L(u)\nabla w \cdot \mathbf{n}\,\mathrm{d}\mathbf{s} - \int_\Omega L(u)\nabla w \cdot \nabla w \,\mathrm{d}\mathbf{x} \overset{(3.5)}{=} - \int_\Omega L(u)|\nabla w|^2 \,\mathrm{d}\mathbf{x}.\end{aligned}$$

Therefore, the total energy is non-increasing in time. Differentiating the total mass $\int_\Omega u \,\mathrm{d}\mathbf{x}$ with respect to the time, gives

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_\Omega u \,\mathrm{d}\mathbf{x} = \int_\Omega \partial_t u \,\mathrm{d}\mathbf{x} = \int_\Omega \nabla \cdot (L(u)\nabla w)\,\mathrm{d}\mathbf{x} = \int_{\partial\Omega} L(u)\nabla w \cdot \mathbf{n}\,\mathrm{d}\mathbf{x} \overset{(3.5)}{=} 0.$$

Hence, the total mass is conserved.

In this paper, we assume a constant mobility and without loss of generality we use $L(u) \equiv 1$. Concerning well-posedness and regularity of a solution of (3.6)–(3.8) with $L(u) \equiv 1$ we refer to [62, 125]. Concentration dependent mobilities, and in particular degenerate ones, are required for many applications; see, e.g. [58, 13, 117]. This is for example the case if the mobility in the interface is larger than in the pure phases. This motivates us to consider concentration dependent mobilities in the future in order to model other physical situations.

3.2.2 Nonsmooth systems

In the last section, we focused on the smooth potential ψ_{pol}. We could easily calculate the derivative of the smooth potential with respect to u. Now, we turn to the nonsmooth potential ψ_{obs} given in (3.3). It can be written via the indicator function

$$\mathcal{I}_{[-1,1]}(u) = \begin{cases} 0 & u \in [-1,1], \\ \infty & \text{otherwise} \end{cases}$$

as

$$\psi_{\mathrm{obs}}(u) = \psi_0(u) + \mathcal{I}_{[-1,1]}(u),$$

where $\psi_0(u) = \frac{1}{2}(1-u^2)$. Differentiating $\mathcal{I}_{[-1,1]}$ in the sense of subdifferentials [65, p. 523], we get the following Cahn–Hilliard system

$$\partial_t u = \Delta w, \tag{3.9}$$

$$w = -\varepsilon \Delta u + \varepsilon^{-1}\left(\psi'_0(u) + \mu\right), \tag{3.10}$$

$$\mu \in \partial \mathcal{I}_{[-1,1]}(u), \tag{3.11}$$

$$|u| \leq 1, \tag{3.12}$$

$$\nabla u \cdot \mathbf{n} = \nabla w \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega, \tag{3.13}$$

where $\partial\mathcal{I}_{[-1,1]}(u)$ is the subdifferential of $\mathcal{I}_{[-1,1]}$ at u, the nonsmooth part of the energy $\mathcal{E}$. We refer to, e.g., [20], for more details about the derivation. This system can be formulated as a variational inequality

$$\langle \partial_t u, v\rangle + (\nabla w, \nabla v) = 0 \quad \forall v \in H^1(\Omega), \tag{3.14}$$

$$(w, v-u) \le \varepsilon(\nabla u, \nabla(v-u)) + \varepsilon^{-1}(\psi_0'(u), v-u) \quad \forall v \in H^1(\Omega),\ |v| \le 1, \tag{3.15}$$

$$|u| \le 1 \quad \text{a.e. in } \Omega, \tag{3.16}$$

which is supplemented by the initial condition $u_0 \in H^1(\Omega)$ with $|u_0| \le 1$ in Ω. The existence, uniqueness, and regularity of a solution of (3.14)–(3.16) was shown in [27].

Due to the variational inequality, the nonsmooth Cahn–Hilliard problem poses a harder challenge compared to the smooth one from the last section. This holds for both, the mathematical as well as numerical analysis, as we shall see during the following sections.

After the derivation of the constitutive Cahn–Hilliard equation or inequality, we are going to study their discretizations in order to be able to solve them numerically. We start with the discretization in time in the next section.

3.3 Time discretization

In this section, we focus on a fully implicit time discretization scheme. Indeed, this results in a time step restriction for both, the smooth and nonsmooth setting, as we will see below. This is not the case if a proper semi-implicit scheme is used. However, as noted, e.g., in [140], semi-implicit schemes usually have larger truncation errors and require smaller time steps than fully implicit schemes. Another motivation arises in numerical simulations for the solution of Allen–Cahn or Cahn–Hilliard variational inequalities [23, 25, 42, 138]: The experiments performed with a semi-implicit scheme yield highly inaccurate results for large time steps. More precisely, these schemes are not able to capture the evolution of the sharp interface limit anymore when the time step is too large.

These arguments motivate us to apply a fully implicit time-discrete scheme.[1] This means, we use the backward Euler discretization for the time derivative $\partial_t u$ and treat all the other terms implicitly. In particular, we treat the potential function implicitly. Let $\tau > 0$ denote the time step size and $t_{n-1} = (n-1)\tau$, $n \in \mathbb{N}$, discrete times. We denote by $u^{(n-1)} \in H^1(\Omega)$ the time-discrete solution at time step t_{n-1}. Further, $u^{(n)}$, $w^{(n)} \in H^1(\Omega)$ form the time-discrete solution at time step $t_n = t_{n-1} + \tau$. In order to ease the notation, from now on we write u^{old}, u, and w instead of $u^{(n-1)}$, $u^{(n)}$, and $w^{(n)}$, respectively.

Again, we start with the smooth setting and consider the weak formulation of (3.6)–(3.8). We discretize this problem in time and give stability and uniqueness conditions. Afterwards, we go over to the nonsmooth setting (3.14)–(3.16). We formulate this problem as the first-order optimality system of an optimization problem. This allows us to specify the conditions for a unique solution.

[1] Note that we draw on a semi-implicit scheme in the preconditioning part in Section 3.7.

3.3.1 Smooth systems

Let us focus on the smooth setting and the corresponding Cahn–Hilliard Equation (3.6)–(3.8). We consider its weak formulation and utilize the implicit Euler scheme. Then, (u, w) solves the following problem: Find u, $w \in H^1(\Omega)$ such that

$$\left(u - u^{\text{old}}, v\right) + \tau\left(\nabla w, \nabla v\right) = 0 \quad \forall v \in H^1(\Omega), \tag{3.17}$$

$$-\left(w, v\right) + \varepsilon\left(\nabla u, \nabla v\right) + \varepsilon^{-1}\left(\psi'_{\text{pol}}(u), v\right) = 0 \quad \forall v \in H^1(\Omega). \tag{3.18}$$

Choosing $v = 1$ in (3.17), we obtain the conservation of mass, i.e., $(u, 1) = (u^{\text{old}}, 1)$, the specific feature of the Cahn–Hilliard model.

Now, we want to give stability and uniqueness conditions for the time step. However, the quartic growth of $\psi_{\text{pol}}(u)$ at infinity introduces various technical difficulties in the analysis. Therefore, we consider a truncated double-well potential. To be more precise, we restrict the growth of $\psi_{\text{pol}}(u)$ to be quadratic for $|u| \geq M$ for a given constant M. This can be done by replacing $\psi_{\text{pol}}(u)$ with

$$\tilde{\psi}(u) := \begin{cases} \frac{3M^2-1}{2}u^2 - 2M^3 u + \frac{1}{4}\left(3M^4 + 1\right) & u > M, \\ \frac{1}{4}\left(u^2 - 1\right)^2 & u \in [-M, M], \\ \frac{3M^2-1}{2}u^2 + 2M^3 u + \frac{1}{4}\left(3M^4 + 1\right) & u < -M, \end{cases}$$

as done, e.g., in [140]. This gives

$$\tilde{\psi}''(u) = \begin{cases} 3M^2 - 1 & u > M, \\ 3u^2 - 1 & u \in [-M, M], \\ 3M^2 - 1 & u < -M, \end{cases}$$

and hence the following condition is satisfied: There exists a constant S such that

$$\max_{u \in \mathbb{R}} |\tilde{\psi}''(u)| \leq S. \tag{3.19}$$

Truncating the polynomial potential is in fact a common practice; see, e.g. [140] and references therein. In particular, the authors of [140] have proven stability conditions for Allen–Cahn and Cahn–Hilliard equations with a truncated double-well potential. They considered various numerical schemes except for the fully implicit one that we are looking at. In the following, we state a uniqueness and stability result for our time-discrete scheme.

Theorem 3.1. *The solution of (3.17)–(3.18) is unique provided that $\tau < \frac{4\varepsilon^3}{S^2}$ and $\psi = \psi_{\text{pol}}$ is replaced by $\tilde{\psi}$.*

Proof. Assume there exist two solutions (u, w) and $(\tilde{u}, \tilde{w})$ of (3.17)–(3.18). Then, we get

$$\left(u - \tilde{u}, v\right) + \tau\left(\nabla(w - \tilde{w}), \nabla v\right) = 0 \quad \forall v \in H^1(\Omega), \tag{3.20}$$

$$-\left(w - \tilde{w}, v\right) + \varepsilon\left(\nabla(u - \tilde{u}), \nabla v\right) + \varepsilon^{-1}\left(\psi'(u) - \psi'(\tilde{u}), v\right) = 0 \quad \forall v \in H^1(\Omega). \tag{3.21}$$

Choosing $v = w - \tilde{w}$ in (3.20) gives

$$0 = \left(u - \tilde{u}, w - \tilde{w}\right) + \tau\left\|\nabla(w - \tilde{w})\right\|^2. \tag{3.22}$$

Choosing $v = u - \tilde{u}$ in (3.21) gives

$$0 = -(u - \tilde{u}, w - \tilde{w}) + \varepsilon \|\nabla(u - \tilde{u})\|^2 + \varepsilon^{-1} (\psi'(u) - \psi'(\tilde{u}), u - \tilde{u}) . \tag{3.23}$$

The last term in (3.23) can be reformulated using the Taylor expansion of the potential

$$\psi(u) = \psi(\tilde{u} + u - \tilde{u}) = \psi(\tilde{u}) + (u - \tilde{u})\, \psi'(\tilde{u}) + \frac{1}{2}(u - \tilde{u})^2\, \psi''(s),$$
$$\psi(\tilde{u}) = \psi(u + \tilde{u} - u) = \psi(u) + (\tilde{u} - u)\, \psi'(u) + \frac{1}{2}(u - \tilde{u})^2\, \psi''(\tilde{s}),$$

where s and $\tilde{s}$ lie between u and $\tilde{u}$. Adding these two equations, gives

$$(\psi'(u) - \psi'(\tilde{u}))\, (u - \tilde{u}) = \frac{1}{2}(u - \tilde{u})^2\, (\psi''(s) + \psi''(\tilde{s})) \overset{(3.19)}{\geq} -S(u - \tilde{u})^2.$$

Therefore, we obtain in (3.23)

$$0 \geq -(u - \tilde{u}, w - \tilde{w}) + \varepsilon \|\nabla(u - \tilde{u})\|^2 - \frac{S}{\varepsilon} \|u - \tilde{u}\|^2 . \tag{3.24}$$

For the last equation, we multiply (3.20) by $\frac{S}{\varepsilon}$ and choose $v = u - \tilde{u}$:

$$\begin{aligned} 0 &= \frac{S}{\varepsilon} \|u - \tilde{u}\|^2 + \frac{\tau S}{\varepsilon} (\nabla(w - \tilde{w}), \nabla(u - \tilde{u})) \\ &= \frac{S}{\varepsilon} \|u - \tilde{u}\|^2 + \left(\frac{\tau S}{\varepsilon \sqrt{2\varepsilon}} \nabla(w - \tilde{w}),\ \sqrt{2\varepsilon} \nabla(u - \tilde{u}) \right) \\ &\geq \frac{S}{\varepsilon} \|u - \tilde{u}\|^2 - \frac{\tau^2 S^2}{4\varepsilon^3} \|\nabla(w - \tilde{w})\|^2 - \varepsilon \|\nabla(u - \tilde{u})\|^2 . \end{aligned} \tag{3.25}$$

In (3.25), we have used Young's inequality with $\alpha_Y = 1$ (Lemma 2.12). Now, adding (3.22), (3.24), and (3.25), we get

$$0 \geq \tau \left(1 - \frac{\tau S^2}{4\varepsilon^3} \right) \|\nabla(w - \tilde{w})\|^2 .$$

Hence, we obtain uniqueness if $1 - \frac{\tau S^2}{4\varepsilon^3} > 0$, which is equivalent to $\tau < \frac{4\varepsilon^3}{S^2}$. Since then, it follows that $\|\nabla(w - \tilde{w})\| = 0$, which implies that $w - \tilde{w}$ is constant. Using this, (3.20) yields $(u - \tilde{u}, v) = 0$ for all $v \in H^1(\Omega)$ and therefore $u = \tilde{u}$ almost everywhere. Finally, (3.21) gives $w = \tilde{w}$ almost everywhere. □

Theorem 3.2. *Under the condition $\tau < \frac{8\varepsilon^3}{S^2}$ and provided that $\psi = \psi_{\text{pol}}$ is replaced by $\tilde{\psi}$, the time discretization scheme (3.17)–(3.18) is energy stable. This means its solution satisfies $\mathcal{E}(u^{(n)}) \leq \mathcal{E}(u^{(n-1)})$ for all $n \geq 1$.*

Proof. Choosing $v = w$ in (3.17), gives

$$0 = \left(u - u^{\text{old}}, w \right) + \tau \|\nabla w\|^2 . \tag{3.26}$$

Choosing $v = u - u^{\text{old}}$ in (3.18), gives

$$0 = -\left(u - u^{\text{old}}, w \right) + \frac{\varepsilon}{2} \left(\|\nabla u\|^2 - \left\|\nabla u^{\text{old}}\right\|^2 + \left\| \nabla \left(u - u^{\text{old}} \right) \right\|^2 \right) + \frac{1}{\varepsilon} \left(\psi'(u), u - u^{\text{old}} \right) . \tag{3.27}$$

The last term in (3.27) can be reformulated using the Taylor expansion of the potential

$$\psi(u^{\text{old}}) = \psi(u + u^{\text{old}} - u) = \psi(u) + (u^{\text{old}} - u)\,\psi'(u) + \frac{1}{2}(u - u^{\text{old}})^2\,\psi''(s),$$

where s lies between u and u^{old}. This yields

$$\psi'(u)\left(u - u^{\text{old}}\right) = \psi(u) - \psi(u^{\text{old}}) + \frac{1}{2}\left(u - u^{\text{old}}\right)^2 \psi''(s) \overset{(3.19)}{\geq} \psi(u) - \psi(u^{\text{old}}) - \frac{S}{2}\left(u - u^{\text{old}}\right)^2.$$

Therefore, we obtain in (3.27)

$$\begin{aligned} 0 \geq -\left(u - u^{\text{old}}, w\right) + \frac{\varepsilon}{2}\left(\|\nabla u\|^2 - \left\|\nabla u^{\text{old}}\right\|^2 + \left\|\nabla\left(u - u^{\text{old}}\right)\right\|^2\right) \\ + \frac{1}{\varepsilon}\left(\psi(u), 1\right) - \frac{1}{\varepsilon}\left(\psi(u^{\text{old}}), 1\right) - \frac{S}{2\varepsilon}\left\|u - u^{\text{old}}\right\|^2. \end{aligned} \tag{3.28}$$

For the last equation, we multiply (3.17) by $\frac{S}{2\varepsilon}$ and choose $v = u - u^{\text{old}}$:

$$\begin{aligned} 0 &= \frac{S}{2\varepsilon}\left\|u - u^{\text{old}}\right\|^2 + \frac{\tau S}{2\varepsilon}\left(\nabla w, \nabla\left(u - u^{\text{old}}\right)\right) \\ &= \frac{S}{2\varepsilon}\left\|u - u^{\text{old}}\right\|^2 + \left(\frac{\tau S}{2\varepsilon\sqrt{\varepsilon}}\nabla w,\ \sqrt{\varepsilon}\nabla\left(u - u^{\text{old}}\right)\right) \qquad (3.29) \\ &\geq \frac{S}{2\varepsilon}\left\|u - u^{\text{old}}\right\|^2 - \frac{\tau^2 S^2}{8\varepsilon^3}\|\nabla w\|^2 - \frac{\varepsilon}{2}\left\|\nabla\left(u - u^{\text{old}}\right)\right\|^2. \qquad (3.30) \end{aligned}$$

In (3.30), we have used Young's inequality with $\alpha_{\text{Y}} = 1$ (Lemma 2.12). Now, adding (3.26), (3.28), and (3.30), we get

$$0 \geq \tau\left(1 - \frac{\tau S^2}{8\varepsilon^3}\right)\|\nabla w\|^2 + \frac{\varepsilon}{2}\left(\|\nabla u\|^2 - \left\|\nabla u^{\text{old}}\right\|^2\right) + \frac{1}{\varepsilon}\left(\psi(u), 1\right) - \frac{1}{\varepsilon}\left(\psi(u^{\text{old}}), 1\right).$$

Now, we can bound the energy in (3.1):

$$\mathcal{E}(u) - \mathcal{E}(u^{\text{old}}) = \frac{\varepsilon}{2}\left(\|\nabla u\|^2 - \left\|\nabla u^{\text{old}}\right\|^2\right) + \frac{1}{\varepsilon}\left(\psi(u), 1\right) - \frac{1}{\varepsilon}\left(\psi(u^{\text{old}}), 1\right) \leq \tau\left(\frac{\tau S^2}{8\varepsilon^3} - 1\right)\|\nabla w\|^2.$$

Hence, we obtain energy stability if $\frac{\tau S^2}{8\varepsilon^3} - 1 \leq 0$, which is equivalent to $\tau \leq \frac{8\varepsilon^3}{S^2}$. □

Remark 3.1. *The time step conditions in Theorem 3.1 and 3.2 differ by a factor of two. This factor is introduced in the right term of (3.29) in order to cancel out the second last term in the right-hand side of (3.27). Hence, the real reason for the factor of two appears in (3.27), where we replace*

$$\varepsilon\left(\nabla u, \nabla(u - u^{\text{old}})\right)$$

by

$$\frac{\varepsilon}{2}\left(\|\nabla u\|^2 - \left\|\nabla u^{\text{old}}\right\|^2 + \left\|\nabla\left(u - u^{\text{old}}\right)\right\|^2\right)$$

in order to match the gradient energy term in $\mathcal{E}(u)$ and $\mathcal{E}(u^{\text{old}})$, respectively.

The resulting time step restrictions are similar to the stability conditions in [140]. Although these conditions appear to be quite restrictive for $\varepsilon \ll 1$, the authors of [140] pointed out that they are in fact needed for the sake of convergence. Moreover, note that explicit schemes usually lead to even more severe time step restrictions.

The approach of the truncated polynomial is only used for the theoretical part. In praxis, the polynomial potential ψ_{pol} behaves quite well and does not result in blow ups of the solution. Violations of $u \in [-1,1]$ in form of $u \in [-1-\delta(\varepsilon), 1+\delta(\varepsilon)]$ occur. However, $\delta(\varepsilon)$ is relatively small. We investigate this issue further in Section 3.8.5.

After having stated and analyzed our time-discrete Cahn–Hilliard equations in the smooth setting, we proceed to the nonsmooth case.

3.3.2 Nonsmooth systems

In the following, we concentrate on the nonsmooth setting and the corresponding Cahn–Hilliard variational inequality (3.14)–(3.16). By utilizing the implicit Euler scheme, we obtain the following problem: Find u, $w \in H^1(\Omega)$ such that

$$(u,v) + \tau(\nabla w, \nabla v) = (u^{\mathrm{old}}, v) \quad \forall v \in H^1(\Omega), \tag{3.31}$$

$$(w, v-u) \leq \varepsilon(\nabla u, \nabla(v-u)) + \varepsilon^{-1}(\psi_0'(u), v-u) \quad \forall v \in H^1(\Omega),\ |v| \leq 1, \tag{3.32}$$

$$|u| \leq 1 \quad \text{a.e. in } \Omega. \tag{3.33}$$

The system (3.31)–(3.33) is supplemented by the initial condition $u_0 \in H^1(\Omega)$ with $|u_0| \leq 1$ in Ω. As in the smooth case, choosing $v = 1$ in (3.31), we obtain the conservation of mass, i.e., $(u,1) = (u^{\mathrm{old}}, 1) = m$ where $m \in (-1,1)$. Without loss of generality, we assume that $m = 0$ and $|\Omega| = 1$, with $|\Omega|$ being the Lebesgue measure of Ω, hold true.

During the next three sections, we follow the analysis presented in Hintermüller et al. [91]. Note that they studied the Cahn–Hilliard variational inequality with a semi-implicit Euler scheme. The difference between our and their scheme is the treatment of the potential term ψ_0' in (3.32). We handle it implicitly whilst Hintermüller et al. handle it explicitly. In this sense, we extend parts of Hintermüller et al's analysis from the semi-implicit time-discrete problem to the fully implicit one.

As done in [28, 83, 91, 20], we rewrite (3.31)–(3.33) as the first-order optimality system of an optimization problem. For this, we define

$$\mathcal{K} := \left\{ v \in H^1(\Omega) \colon |v| \leq 1 \text{ a.e. in } \Omega \right\}, \qquad \mathcal{V}_0 := \left\{ v \in H^1(\Omega) \colon (v,1) = 0 \right\},$$

and consider the following minimization problem

$$\min_{(u,w)\in\mathcal{K}\times\mathcal{V}_0} \mathcal{J}(u,w) := \frac{\varepsilon}{2}\|\nabla u\|^2 + \frac{1}{\varepsilon}\int_\Omega \psi_0(u)\,\mathrm{d}\mathbf{x} + \frac{\tau}{2}\|\nabla w\|^2 \quad \text{subject to (3.31).} \tag{$\mathcal{P}$}$$

Let

$$\mathcal{F} = \{(u,w) \in \mathcal{K} \times \mathcal{V}_0 \colon (u,w) \text{ achieves (3.31)}\}$$

be the admissible set of ($\mathcal{P}$). Analogous to [91, Lemma 3.1], we have the following result.

Lemma 3.3. *The following properties hold true:*

(i) $\mathcal{F} \neq \emptyset$ *and* $\mathcal{F} \subset \mathcal{V}_0 \times \mathcal{V}_0$.

(ii) $\mathcal{F}$ *is a closed and convex set of* $H^1(\Omega) \times H^1(\Omega)$.

(iii) Let $\tau < 4\varepsilon^3$. *Then,* $\mathcal{J}$ *is strictly convex on* $\mathcal{F}$.

(iv) Let $\tau < 4\varepsilon^3$. *Then, for every sequence* $(u_m, w_m)_{m\in\mathbb{N}}$ *in* $\mathcal{F}$ *such that* $\lim_{m\to\infty} \|u_m\|_1 = \infty$ *or* $\lim_{m\to\infty} \|w_m\|_1 = \infty$, *we have* $\lim_{m\to\infty} \mathcal{J}(u_m, w_m) = \infty$.

Proof. (i) $\mathcal{F} \neq \emptyset$ since $(u^{\text{old}}, 0) \in \mathcal{F}$. Let $(u, w) \in \mathcal{F}$. It follows $w \in \mathcal{V}_0$. By taking $v = 1$ in (3.31), we obtain $(u, 1) = (u^{\text{old}}, 1) = 0$. Hence, $u \in \mathcal{V}_0$.

(ii) First, we proof that $\mathcal{F}$ is convex. Let $(u_1, w_1), (u_2, w_2) \in \mathcal{F}$ and $\lambda \in [0, 1]$. We have to show that $(\lambda u_1 + (1-\lambda)u_2, \lambda w_1 + (1-\lambda)w_2) \in \mathcal{F}$ holds true. From

$$(\lambda w_1 + (1-\lambda)w_2, 1) = \lambda \underbrace{(w_1, 1)}_{=0} + (1-\lambda) \underbrace{(w_2, 1)}_{=0} = 0,$$

it follows $\lambda w_1 + (1-\lambda)w_2 \in \mathcal{V}_0$. Further,

$$|\lambda u_1 + (1-\lambda)u_2| \le \lambda \underbrace{|u_1|}_{\le 1 \text{ a.e. in } \Omega} + (1-\lambda) \underbrace{|u_2|}_{\le 1 \text{ a.e. in } \Omega} \le 1 \text{ a.e. in } \Omega$$

and hence $\lambda u_1 + (1-\lambda)u_2 \in \mathcal{K}$. Finally,

$$\begin{aligned}(\lambda u_1 + (1-\lambda)u_2, v) &+ \tau(\nabla(\lambda w_1 + (1-\lambda)w_2), \nabla v) \\ &= \lambda[\underbrace{(u_1, v) + \tau(\nabla w_1, \nabla v)}_{=(u^{\text{old}}, v)}] + (1-\lambda)[\underbrace{(u_2, v) + \tau(\nabla w_2, \nabla v)}_{=(u^{\text{old}}, v)}] \\ &= (u^{\text{old}}, v) \quad \forall v \in H^1(\Omega),\end{aligned}$$

which means that $(\lambda u_1 + (1-\lambda)u_2, \lambda w_1 + (1-\lambda)w_2)$ fulfills (3.31). Altogether, $\mathcal{F}$ is convex.

Now, let us proof the closedness of $\mathcal{F}$ in $H^1(\Omega) \times H^1(\Omega)$. Let $(u_m, w_m)_{m\in\mathbb{N}} \subset \mathcal{F}$ converge strongly to $(u, w) \in H^1(\Omega) \times H^1(\Omega)$ for $m \to \infty$. We have to show that $(u, w) \in \mathcal{F}$. According to Theorem 2.6, every strongly convergent sequence is weakly convergent, i.e.,

$$\begin{aligned}(u_m, v)_1 &\overset{m\to\infty}{\longrightarrow} (u, v)_1 \quad \forall v \in H^1(\Omega) \\ \Leftrightarrow (u_m, v) + (\nabla u_m, \nabla v) &\overset{m\to\infty}{\longrightarrow} (u, v) + (\nabla u, \nabla v) \quad \forall v \in H^1(\Omega),\end{aligned}$$

as well as

$$\begin{aligned}(w_m, v)_1 &\overset{m\to\infty}{\longrightarrow} (w, v)_1 \quad \forall v \in H^1(\Omega) \\ \Leftrightarrow (w_m, v) + (\nabla w_m, \nabla v) &\overset{m\to\infty}{\longrightarrow} (w, v) + (\nabla w, \nabla v) \quad \forall v \in H^1(\Omega).\end{aligned}$$

Hence, we obtain

$$(u, v) + \tau(\nabla w, \nabla v) = (u^{\text{old}}, v) \quad \forall v \in H^1(\Omega)$$

and

$$(w, 1) = 0.$$

What is left to show is that $u \in \mathcal{K}$. As stated in [91, p. 781], $\mathcal{K}$ is weakly closed in $H^1(\Omega)$. Hence, Definition 2.9 yields the desired result. Altogether, $(u, w) \in \mathcal{F}$.

(iii) Let $(u_1, w_1), (u_2, w_2) \in \mathcal{F}$ and $\alpha \in (0, 1)$. We define

$$r(\alpha) := \alpha\mathcal{J}(u_1, w_1) + (1-\alpha)\mathcal{J}(u_2, w_2) - \mathcal{J}\left(\alpha u_1 + (1-\alpha)u_2, \alpha w_1 + (1-\alpha)w_2\right).$$

We have to show $r(\alpha) > 0$. We start with proving $r(\alpha) \geq 0$. It holds

$$\begin{aligned}
r(\alpha) &= \alpha\left(\frac{\varepsilon}{2}\|\nabla u_1\|^2 + \frac{\tau}{2}\|\nabla w_1\|^2 + \frac{1}{\varepsilon}(\psi_0(u_1), 1)\right) - \frac{\varepsilon}{2}\|\nabla(\alpha u_1 + (1-\alpha)u_2)\|^2 \\
&\quad + (1-\alpha)\left(\frac{\varepsilon}{2}\|\nabla u_2\|^2 + \frac{\tau}{2}\|\nabla w_2\|^2 + \frac{1}{\varepsilon}(\psi_0(u_2), 1)\right) - \frac{\tau}{2}\|\nabla(\alpha w_1 + (1-\alpha)w_2)\|^2 \\
&\quad - \frac{1}{\varepsilon}(\psi_0(\alpha u_1 + (1-\alpha)u_2), 1) \\
&= \alpha(1-\alpha)\Big[\frac{\varepsilon}{2}\left(\|\nabla u_1\|^2 + \|\nabla u_2\|^2 - 2\left(\nabla u_1, \nabla u_2\right)\right) - \frac{1}{2\varepsilon}\left(\|u_1\|^2 + \|u_2\|^2 - 2\left(u_1, u_2\right)\right) \\
&\qquad + \frac{\tau}{2}\left(\|\nabla w_1\|^2 + \|\nabla w_2\|^2 - 2\left(\nabla w_1, \nabla w_2\right)\right)\Big] \\
&= \frac{\alpha(1-\alpha)}{2}\left(\varepsilon\|\nabla(u_1 - u_2)\|^2 + \tau\|\nabla(w_1 - w_2)\|^2 - \frac{1}{\varepsilon}\|u_1 - u_2\|^2\right).
\end{aligned}$$

Since $(u_1, w_1), (u_2, w_2) \in \mathcal{F}$, they satisfy (3.31)

$$(u_1, v) + \tau(\nabla w_1, \nabla v) = (u^{\text{old}}, v) \quad \forall v \in H^1(\Omega), \tag{3.34}$$

$$(u_2, v) + \tau(\nabla w_2, \nabla v) = (u^{\text{old}}, v) \quad \forall v \in H^1(\Omega). \tag{3.35}$$

Choosing $v = u_1 - u_2$ in (3.34)–(3.35), we obtain

$$(u_1, u_1 - u_2) + \tau(\nabla w_1, \nabla(u_1 - u_2)) = (u^{\text{old}}, u_1 - u_2), \tag{3.36}$$

$$(u_2, u_1 - u_2) + \tau(\nabla w_2, \nabla(u_1 - u_2)) = (u^{\text{old}}, u_1 - u_2). \tag{3.37}$$

Subtracting (3.36)–(3.37) from each other, we get

$$-\|u_1 - u_2\|^2 = \tau(\nabla(w_1 - w_2), \nabla(u_1 - u_2)). \tag{3.38}$$

Applying Young's inequality with $\alpha_Y = 2\beta > 0$ (Lemma 2.12) to the right-hand side of (3.38), gives

$$-\|u_1 - u_2\|^2 = \tau(\nabla(w_1 - w_2), \nabla(u_1 - u_2)) \geq -\tau\beta\|\nabla(w_1 - w_2)\|^2 - \frac{\tau}{4\beta}\|\nabla(u_1 - u_2)\|^2. \tag{3.39}$$

Substituting (3.39) into the equation for $r(\alpha)$ yields

$$r(\alpha) \geq \frac{\alpha(1-\alpha)}{2}\left[\left(\varepsilon - \frac{\tau}{4\beta\varepsilon}\right)\|\nabla(u_1 - u_2)\|^2 + \left(\tau - \frac{\tau\beta}{\varepsilon}\right)\|\nabla(w_1 - w_2)\|^2\right].$$

For the strict convexity, we require

$$\varepsilon - \frac{\tau}{4\beta\varepsilon} > 0 \quad \Leftrightarrow \quad \beta > \frac{\tau}{4\varepsilon^2}, \tag{3.40}$$

$$\tau - \frac{\tau\beta}{\varepsilon} > 0 \quad \Leftrightarrow \quad \beta < \varepsilon. \tag{3.41}$$

Hence,

$$\frac{\tau}{4\varepsilon^2} < \beta < \varepsilon,$$

which leads to the time step restriction

$$\frac{\tau}{4\varepsilon^2} < \varepsilon \quad \Leftrightarrow \quad \tau < 4\varepsilon^3.$$

Now, assume $r(\alpha) = 0$. Then,

$$\|\nabla(u_1 - u_2)\|^2 = \|\nabla(w_1 - w_2)\|^2 = 0. \tag{3.42}$$

Since $\mathcal{F} \subset \mathcal{V}_0 \times \mathcal{V}_0$, we have $\int_\Omega u_1 - u_2 \,\mathrm{d}\mathbf{x} = \int_\Omega w_1 - w_2 \,\mathrm{d}\mathbf{x} = 0$. Hence, we can apply the Poincaré inequality (Theorem 2.22)

$$\begin{aligned} \|u_1 - u_2\|_1^2 \le c_P \|\nabla(u_1 - u_2)\|^2 \quad &\Leftrightarrow \quad \|u_1 - u_2\|^2 + \|\nabla(u_1 - u_2)\|^2 \le c_P \|\nabla(u_1 - u_2)\|^2, \\ \|w_1 - w_2\|_1^2 \le c_P \|\nabla(w_1 - w_2)\|^2 \quad &\Leftrightarrow \quad \|w_1 - w_2\|^2 + \|\nabla(w_1 - w_2)\|^2 \le c_P \|\nabla(w_1 - w_2)\|^2. \end{aligned}$$

It follows from (3.42) that $\|u_1 - u_2\| = \|w_1 - w_2\| = 0$. This implies $(u_1, w_1) = (u_2, w_2)$ almost everywhere in Ω. In summary, $\mathcal{J}$ is strictly convex on $\mathcal{F}$ provided that $\tau < 4\varepsilon^3$.

(iv) Let $(u, w) \in \mathcal{F}$. Using

$$\int_\Omega \psi_0(u) \,\mathrm{d}\mathbf{x} = \int_\Omega \frac{1}{2}(1 - u^2) \,\mathrm{d}\mathbf{x} = \frac{1}{2}|\Omega| - \frac{1}{2}\int_\Omega u^2 \,\mathrm{d}\mathbf{x} = \frac{1}{2} - \frac{1}{2}\|u\|^2,$$

we obtain

$$\mathcal{J}(u, w) = \frac{\varepsilon}{2}\|\nabla u\|^2 + \frac{\tau}{2}\|\nabla w\|^2 + \underbrace{\frac{1}{2\varepsilon}}_{>0} - \frac{1}{2\varepsilon}\|u\|^2 > \frac{\varepsilon}{2}\|\nabla u\|^2 + \frac{\tau}{2}\|\nabla w\|^2 - \frac{1}{2\varepsilon}\|u\|^2.$$

Since $(u, w) \in \mathcal{F}$, Equation (3.31) is fulfilled. Choosing $v = u$ in (3.31) leads to

$$-\|u\|^2 = \tau(\nabla w, \nabla u) - (u^{\mathrm{old}}, u). \tag{3.43}$$

Applying Young's inequality with $\alpha_Y = 2\beta_1 > 0$ to the left term in the right-hand side of (3.43) and with $\alpha_Y = 2\beta_2 > 0$ to the right term in the right-hand side of (3.43), we get

$$-\|u\|^2 = \tau(\nabla w, \nabla u) - (u^{\mathrm{old}}, u) \ge -\tau\beta_1\|\nabla w\|^2 - \frac{\tau}{4\beta_1}\|\nabla u\|^2 - \beta_2\|u^{\mathrm{old}}\|^2 - \frac{1}{4\beta_2}\|u\|^2. \tag{3.44}$$

Since $\mathcal{F} \subset \mathcal{V}_0 \times \mathcal{V}_0$, we have $\int_\Omega u \,\mathrm{d}\mathbf{x} = 0$. Hence, we can apply the Poincaré inequality (Theorem 2.22)

$$\begin{aligned} \|u\|_1^2 = \|u\|^2 + \|\nabla u\|^2 &\le c_{\mathrm{P}} \|\nabla u\|^2 \\ \Rightarrow \|u\|^2 \le c_{\mathrm{P}} \|\nabla u\|^2 - \|\nabla u\|^2 &\le c_{\mathrm{P}} \|\nabla u\|^2 \end{aligned} \tag{3.45}$$

in (3.44) and obtain

$$-\|u\|^2 \ge -\tau\beta_1\|\nabla w\|^2 - \left(\frac{\tau}{4\beta_1} + \frac{c_{\mathrm{P}}}{4\beta_2}\right)\|\nabla u\|^2 - \beta_2\|u^{\mathrm{old}}\|^2.$$

Substituting this result into the equation of $\mathcal{J}(u,w)$ above, we get

$$\mathcal{J}(u,w) > \left(\frac{\varepsilon}{2} - \frac{\tau}{8\beta_1\varepsilon} - \frac{c_P}{8\beta_2\varepsilon}\right)\|\nabla u\|^2 + \left(\frac{\tau}{2} - \frac{\tau\beta_1}{2\varepsilon}\right)\|\nabla w\|^2 - \frac{\beta_2}{2\varepsilon}\|u^{\text{old}}\|^2.$$

This inequality holds for all $\beta_1, \beta_2 > 0$. Now, we want to choose β_1, β_2 such that

$$\frac{\varepsilon}{2} - \frac{\tau}{8\beta_1\varepsilon} - \frac{c_P}{8\beta_2\varepsilon} > 0 \quad \Leftrightarrow \quad \frac{\varepsilon}{2} - \frac{\tau}{8\beta_1\varepsilon} > \frac{c_P}{8\beta_2\varepsilon}, \tag{3.46}$$

$$\frac{\tau}{2} - \frac{\tau\beta_1}{2\varepsilon} > 0 \quad \Leftrightarrow \quad \beta_1 < \varepsilon. \tag{3.47}$$

Since $\beta_2 > 0$, we need in (3.46)

$$\frac{\varepsilon}{2} - \frac{\tau}{8\beta_1\varepsilon} > 0 \quad \Leftrightarrow \quad \beta_1 > \frac{\tau}{4\varepsilon^2}.$$

Hence,

$$\frac{\tau}{4\varepsilon^2} < \beta_1 < \varepsilon,$$

which leads to the time step restriction

$$\frac{\tau}{4\varepsilon^2} < \varepsilon \quad \Leftrightarrow \quad \tau < 4\varepsilon^3.$$

Under these conditions for τ and β_1, we can choose β_2 such that (3.46) is fulfilled. Next, due to (3.45), it holds $\lim_{m\to\infty}\|\nabla u_m\| = \infty$ for every sequence $u_m \in \mathcal{V}_0$ with $\lim_{m\to\infty}\|u_m\|_1 = \infty$. The same result is true if we replace u_m by w_m. Therefore, $\lim_{m\to\infty}\mathcal{J}(u_m, w_m) = \infty$ for every sequence $(u_m, w_m)_{m\in\mathbb{N}} \subset \mathcal{F}$ with $\lim_{m\to\infty}\|u_m\|_1 = \infty$ or $\lim_{m\to\infty}\|w_m\|_1 = \infty$, provided that $\tau < 4\varepsilon^3$. □

Remark 3.2. *Note the necessity of the time step restriction $\tau < 4\varepsilon^3$ for the implicit scheme, which has not to be claimed for the semi-implicit one in [91]. Nevertheless, the time step restriction is an essential characteristic of the nature of the problem. The results obtained for large time steps within the semi-implicit system are highly inaccurate for capturing the evolution of the sharp interface limit; see Section 3.8.4 or [42, 138, 20].*

The relation between ($\mathcal{P}$) and (3.31)–(3.33) is established next.

Theorem 3.4. *Let $\tau \in (0, 4\varepsilon^3)$. The problem ($\mathcal{P}$) has a unique solution (u^*, w^*). Moreover, there exists a unique Lagrange multiplier $p^* \in H^1(\Omega)$, such that $w^* = p^* - (p^*, 1)$ and (u^*, p^*) is a solution of (3.31)–(3.33). Conversely, if (u^*, p^*) is a solution of (3.31)–(3.33), then (u^*, w^*) with $w^* = p^* - (p^*, 1)$ is the unique solution of ($\mathcal{P}$).*

Proof. The existence and uniqueness of the solution of ($\mathcal{P}$) follow from the previous lemma. In order to prove the existence of a Lagrange multiplier p^*, we need a constraint qualification. For the semi-implicit time-discrete scheme, Hintermüller et al. [91, pp. 781–782] make use of the constraint qualification by Zowe and Kurcyusz [154]. The proof for our time-discrete system follows analogously and we refer the reader to [91, pp. 781–782] for the complete proof. Moreover, Hintermüller et al. proved the uniqueness of p^*, which can be adapted to our problem formulation as well. The same holds true for the reverse implication stated in the last part of the theorem. □

After having analyzed the nonsmooth implicit time-discrete Cahn–Hilliard system, we want to tackle this problem. The presence of the variational inequality in (3.32) makes this problem hard. In the next section, we make use of a technique which transforms the variational inequality into an equation.

3.4 Moreau–Yosida regularization

Variational inequalities like (3.32) may be reformulated by introducing Lagrange multipliers associated with the constraints in $\mathcal{K}$ as done, e.g., in [20]. However, they are elements of $H^1(\Omega)^*$ and do not allow a pointwise interpretation. This complicates the numerical treatment. Motivated by [91], we replace the optimization problem ($\mathcal{P}$) by its Moreau–Yosida regularized version

$$\min_{(u,w)\in H^1(\Omega)\times\mathcal{V}_0} \mathcal{J}_c(u,w) \quad \text{subject to (3.31)} \tag{$\mathcal{P}_c$}$$

with the objective

$$\mathcal{J}_c(u,w) = \mathcal{J}(u,w) + \frac{1}{2c}\|\max(0,u-1)\|^2 + \frac{1}{2c}\|\min(0,u+1)\|^2.$$

Here, $0 < c \ll 1$ denotes the associated regularization or penalty parameter. Note that the constraint $u \in \mathcal{K}$ in ($\mathcal{P}$) has been relaxed to $u \in H^1(\Omega)$ in the regularized problem ($\mathcal{P}_c$). At the same time, a damped version of the box constraints in $\mathcal{K}$ has been inserted into the objective function. The smaller c is the larger is the penalization for the violation of the condition $|u| \leq 1$. Hence, the limit $c \to 0$ represents the original minimization problem ($\mathcal{P}$). Indeed, this convergence is proven below in Proposition 3.6. Further, we will see in Theorem 3.5 that the regularization of the box constraints answers for the disappearance of the variational inequality.

Besides the minimization perspective, there is another way to explain the regularization. Basically, the double-obstacle potential ψ_{obs} in (3.3) is regularized by

$$\begin{aligned}\psi_c(u) &= \frac{1}{2}\left(1 - u^2 + \frac{\varepsilon}{c}\left[\max(0,u-1) + \min(0,u+1)\right]^2\right)\\ &= \psi_0(u) + \frac{\varepsilon}{2c}\left[\max(0,u-1) + \min(0,u+1)\right]^2.\end{aligned}$$

Figure 3.1 illustrates the regularized potential for different values of c. This figure is not to scale and is intended for illustrative purposes only. Note that the black curve represents the original double-obstacle potential. Analyzing ($\mathcal{P}_c$), we obtain a result

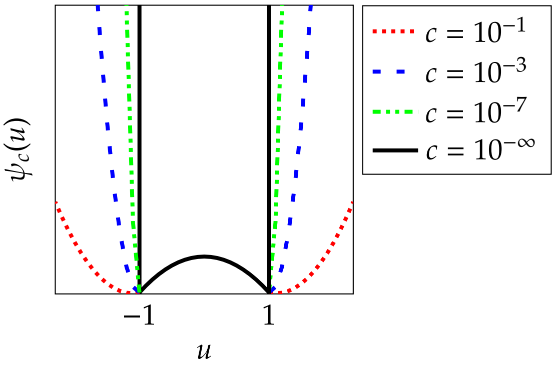


Figure 3.1: Illustration of the Moreau–Yosida regularized potential for different values of the penalty parameter c.

similar to Theorem 3.4.

Theorem 3.5. *Let $\tau \in (0, 4\varepsilon^3)$. The problem ($\mathcal{P}_c$) has a unique solution (u_c, w_c). Moreover, there exists a unique $p_c \in H^1(\Omega)$ such that*

$$p_c - (p_c, 1) = w_c, \tag{3.48}$$

$$\tau(\nabla p_c, \nabla v) + (u_c, v) - (u^{\text{old}}, v) = 0 \quad \forall v \in H^1(\Omega), \tag{3.49}$$

$$\begin{aligned} \varepsilon(\nabla u_c, \nabla v) + c^{-1}\,(\max(0, u_c - 1) + \min(0, u_c + 1), v)& \\ - (p_c, v) - \varepsilon^{-1}(u_c, v) = 0 \quad \forall v \in H^1(\Omega).& \end{aligned} \tag{3.50}$$

Conversely, if (u_c, p_c) is a solution of (3.49)–(3.50), then (u_c, w_c) with $w_c = p_c - (p_c, 1)$ is the unique solution of ($\mathcal{P}_c$).

Proof. The functionals $u \to \|\max(0, u-1)\|^2$ and $u \to \|\min(0, u+1)\|^2$ are convex and Fréchet-differentiable on $H^1(\Omega)$ as noted in [91, p. 783]. We can show that $\mathcal{F}_c$, the feasible set of ($\mathcal{P}_c$), and $\mathcal{J}_c$ satisfy the analogue of Lemma 3.3 for ($\mathcal{P}_c$). In fact, the proof is exactly the same. Hence, ($\mathcal{P}_c$) has a unique solution (u_c, w_c), provided that $\tau < 4\varepsilon^3$. The rest of the proof follows analogously to [91, p. 783]. □

Proposition 3.6. *Let $\tau \in (0, 4\varepsilon^3)$. Let $\{(u_c, w_c)\}_{c>0}$ be a sequence of solutions of ($\mathcal{P}_c$) as $c \to 0$. Then, there exists a subsequence still denoted by $\{(u_c, w_c)\}_{c>0}$ such that*

$$(u_c, w_c) \longrightarrow (u^*, w^*) \;\text{ in } H^1(\Omega) \times H^1(\Omega) \tag{3.51}$$

as $c \to 0$, where (u^*, w^*) is the unique solution of ($\mathcal{P}$). In particular, u^* is the order parameter corresponding to the solution of (3.31)-(3.33).

Proof. First of all, we have

$$\mathcal{J}(u_c, w_c) \le \mathcal{J}_c(u_c, w_c) \le \mathcal{J}_c(u^*, w^*) = \mathcal{J}(u^*, w^*). \tag{3.52}$$

Hence, there exists a constant $\beta > 0$ independent of c such that

$$\mathcal{J}_c(u_c, w_c) \le \beta.$$

Next, we estimate $\mathcal{J}_c(u_c, w_c)$ from below. As in the proof of Lemma 3.3(iv) we obtain with $\tau < 4\varepsilon^3$ and suitable $\beta_1, \beta_2 > 0$

$$\begin{aligned} \mathcal{J}_c(u_c, w_c) > &\underbrace{\left(\frac{\varepsilon}{2} - \frac{\tau}{8\beta_1\varepsilon} - \frac{c_P}{8\beta_2\varepsilon}\right)}_{>0} \underbrace{\|\nabla u_c\|^2}_{\ge \frac{1}{c_P}\|u_c\|_1^2} + \underbrace{\left(\frac{\tau}{2} - \frac{\tau\beta_1}{2\varepsilon}\right)}_{>0} \underbrace{\|\nabla w_c\|^2}_{\ge \frac{1}{c_P}\|w_c\|_1^2} - \frac{\beta_2}{2\varepsilon}\|u^{\text{old}}\|^2 \\ &+ \frac{1}{2}\left\|\frac{1}{\sqrt{c}}\max(0, u_c - 1)\right\|^2 + \frac{1}{2}\left\|\frac{1}{\sqrt{c}}\min(0, u_c + 1)\right\|^2 \\ \ge\; & C_1\|u_c\|_1^2 + C_2\|w_c\|_1^2 - \frac{\beta_2}{2\varepsilon}\|u^{\text{old}}\|^2 + \frac{1}{2}\left\|\frac{1}{\sqrt{c}}\max(0, u_c - 1)\right\|^2 \\ &+ \frac{1}{2}\left\|\frac{1}{\sqrt{c}}\min(0, u_c + 1)\right\|^2. \end{aligned}$$

In order to ease the notation, we have introduced the constants $C_1, C_2 > 0$, which depend on ε, τ, β_1, β_2, c_{P}. This results in

$$
\begin{aligned}
\{u_c\} &\text{ bounded in } H^1(\Omega),\\
\{w_c\} &\text{ bounded in } H^1(\Omega),\\
\left\{\frac{1}{\sqrt{c}}\max(0, u_c - 1)\right\} &\text{ bounded in } L^2(\Omega), \qquad (3.53)\\
\left\{\frac{1}{\sqrt{c}}\min(0, u_c + 1)\right\} &\text{ bounded in } L^2(\Omega). \qquad (3.54)
\end{aligned}
$$

Since $\{(u_c, w_c)\}_{c>0}$ is a bounded sequence in the Hilbert space $H^1(\Omega) \times H^1(\Omega)$, it has a weakly convergent subsequence. Hence, there exists a $(u, w) \in H^1(\Omega) \times H^1(\Omega)$ and a subsequence $\{(u_{c_m}, w_{c_m})\}_{m\in\mathbb{N}} \subset H^1(\Omega) \times H^1(\Omega)$ such that

$$
(u_{c_m}, w_{c_m}) \overset{m\to\infty}{\rightharpoonup} (u, w) \quad \text{in } H^1(\Omega) \times H^1(\Omega).
$$

Because of the compact embedding $H^1(\Omega) \hookrightarrow L^2(\Omega)$, weakly convergent sequences in $H^1(\Omega)$ are strongly convergent in $L^2(\Omega)$, i.e.,

$$
(u_{c_m}, w_{c_m}) \overset{m\to\infty}{\to} (u, w) \quad \text{in } L^2(\Omega) \times L^2(\Omega). \tag{3.55}
$$

According to Proposition 2.10, the strong convergence in $L^2(\Omega)$ implies

$$
\|u\|^2 \geq \limsup_{m\to\infty} \|u_{c_m}\|^2 \geq \liminf_{m\to\infty} \|u_{c_m}\|^2. \tag{3.56}
$$

According to the proof of Lemma 3.3(ii), the weak convergence of $\{(u_{c_m}, w_{c_m})\}_{m\in\mathbb{N}}$ in $H^1(\Omega) \times H^1(\Omega)$ implies

$$
(u, v) + \tau(\nabla w, \nabla v) = (u^{\mathrm{old}}, v) \quad \forall v \in H^1(\Omega)
$$

and

$$
(w, 1) = 0.
$$

From (3.55) and Lebesgue's dominated convergence theorem, it follows

$$
\begin{aligned}
\max(0, u_{c_m} - 1) &\overset{m\to\infty}{\longrightarrow} \max(0, u - 1) \quad \text{in } L^2(\Omega),\\
\min(0, u_{c_m} + 1) &\overset{m\to\infty}{\longrightarrow} \min(0, u + 1) \quad \text{in } L^2(\Omega).
\end{aligned}
$$

This together with (3.53)–(3.54) yields

$$
-1 \leq u \leq 1 \quad \text{a.e. in } \Omega. \tag{3.57}
$$

Hence, $(u, w) \in \mathcal{F}$. It holds that the $H^1(\Omega)$-seminorm

$$
|v|_1^2 = \|\nabla v\|^2
$$

is weakly lower semicontinuous. Hence, according to Definition 2.10, the weak convergence of a sequence $v_m \rightharpoonup v$ in $H^1(\Omega)$ for $m \to \infty$ implies $|v|_1 \leq \liminf_{m\to\infty} |v_m|_1$.

Therefore, we have

$$\begin{aligned}
\mathcal{J}(u,w) &= \frac{\varepsilon}{2}\|\nabla u\|^2 + \frac{1}{2\varepsilon} - \frac{1}{2\varepsilon}\|u\|^2 + \frac{\tau}{2}\|\nabla w\|^2 \\
&\leq \frac{\varepsilon}{2}\liminf_{m\to\infty}\|\nabla u_{c_m}\|^2 + \frac{1}{2\varepsilon} - \frac{1}{2\varepsilon}\|u\|^2 + \frac{\tau}{2}\liminf_{m\to\infty}\|\nabla w_{c_m}\|^2 \\
&\overset{(3.56)}{\leq} \liminf_{m\to\infty}\left(\frac{\varepsilon}{2}\|\nabla u_{c_m}\|^2 + \frac{1}{2\varepsilon} - \frac{1}{2\varepsilon}\|u_{c_m}\|^2 + \frac{\tau}{2}\|\nabla w_{c_m}\|^2\right) \\
&= \liminf_{m\to\infty}\mathcal{J}(u_{c_m}, w_{c_m}).
\end{aligned}$$

Together with (3.52), we obtain

$$\mathcal{J}(u,w) \leq \liminf_{m\to\infty}\mathcal{J}(u_{c_m}, w_{c_m}) \leq \mathcal{J}(u^*, w^*). \tag{3.58}$$

The pair (u^*, w^*) is the unique solution of $(\mathcal{P})$. In contrast, (u,w) is a feasible solution of $(\mathcal{P})$. Hence, $\mathcal{J}(u,w) \geq \mathcal{J}(u^*, w^*)$ and (3.58) becomes an equation. This gives $(u,w) = (u^*, w^*)$.

It remains to show the strong convergence in (3.51). We have already proven the weak convergence. Hence, what is left to show is the following norm convergence:

$$\begin{aligned}
\|u_{c_m}\|_1 &\overset{m\to\infty}{\longrightarrow} \|u^*\|_1, \\
\|w_{c_m}\|_1 &\overset{m\to\infty}{\longrightarrow} \|w^*\|_1.
\end{aligned}$$

(3.52) and (3.58) imply

$$\frac{1}{2c_m}\|\max(0, u_{c_m} - 1)\|^2 + \frac{1}{2c_m}\|\min(0, u_{c_m} + 1)\|^2 \overset{m\to\infty}{\longrightarrow} 0.$$

Thus,

$$\mathcal{J}(u^*, w^*) \leq \liminf_{m\to\infty}\mathcal{J}_{c_m}(u_{c_m}, w_{c_m}) \leq \limsup_{m\to\infty}\mathcal{J}_{c_m}(u_{c_m}, w_{c_m}) \leq \mathcal{J}(u^*, w^*).$$

This means

$$\mathcal{J}(u^*, w^*) = \lim_{m\to\infty}\mathcal{J}_{c_m}(u_{c_m}, w_{c_m})$$

and it follows

$$\lim_{m\to\infty}\|\nabla u_{c_m}\| = \|\nabla u^*\| \quad \text{and} \quad \lim_{m\to\infty}\|\nabla w_{c_m}\| = \|\nabla w^*\|.$$

From $\|u_{c_m}\|_1^2 = \|u_{c_m}\|^2 + \|\nabla u_{c_m}\|^2$ and the strong convergence in (3.55), we imply the norm convergence of $\|u_{c_m}\|_1^2$. The same holds for w_{c_m}. The weak and norm convergence yield the strong convergence result (3.51). □

We have seen how the application of a Moreau–Yosida regularization technique can circumvent the treatment of the variational inequality in (3.32) as well as the box constraints in (3.33). Indeed, it results in an iterative way for solving the time-discrete Cahn–Hilliard system (3.31)–(3.33): For a sequence $\{c_p\}_{p\in\mathbb{N}}$ with $c_p \to 0$, solve the optimality system (3.49)–(3.50).

Now, we have arrived at a time-discrete nonlinear equation for both cases, the smooth potential setting as well as the regularized nonsmooth one. In order to solve the former system, we apply standard Newton methods. Since this is a straightforward step, we will not discuss it here. Regarding the second case, we have to pay attention to the maximum and minimum operator present in (3.50). However, we can solve the corresponding nonlinear system via a weaker notion of Newton methods — the SSN method. This will be the topic of the following chapter.

3.5 Semismooth Newton method

We apply the function space-based algorithm motivated in [91] for solving the time-discrete Cahn–Hilliard problem (3.31)–(3.33). For a specified sequence $c \to 0$, we solve the optimality system (3.49)–(3.50), compactly written as

$$F_c(u_c, w_c) = \left(F_c^{(1)}(u_c, w_c), F_c^{(2)}(u_c, w_c)\right) = 0, \tag{3.59}$$

for every c by an SSN algorithm. In (3.59), the components are defined by

$$\begin{aligned}\left\langle F_c^{(1)}(u,w), v\right\rangle &= \tau(\nabla w, \nabla v) + (u, v) - (u^{\text{old}}, v),\\ \left\langle F_c^{(2)}(u,w), v\right\rangle &= \varepsilon(\nabla u, \nabla v) + c^{-1}\left(\max(0, u-1) + \min(0, u+1), v\right) - (w, v) - \varepsilon^{-1}(u, v),\end{aligned}$$

for all $u, w, v \in H^1(\Omega)$. Due to the presence of the maximum and minimum operator, F_c is not Fréchet-differentiable. However, it satisfies the weaker notion of Newton differentiability; see Definition 2.11.

Lemma 3.7. *The mapping $F_c\colon H^1(\Omega)\times H^1(\Omega) \to H^1(\Omega)^*\times H^1(\Omega)^*$ is Newton-differentiable. Furthermore, the operator $G_c(u,w)$ given by*

$$\left\langle G_c(u,w)(\delta u, \delta w), (\phi, \psi)\right\rangle = \begin{pmatrix} \tau(\nabla \delta w, \nabla \phi) + (\delta u, \phi) \\ \varepsilon(\nabla \delta u, \nabla \psi) + c^{-1}(\chi_{\mathcal{M}(u)}\delta u, \psi) - (\delta w, \psi) - \varepsilon^{-1}(\delta u, \psi) \end{pmatrix}$$

serves as a Newton derivative for F_c, where $\chi_{\mathcal{M}(u)}$ is the characteristic function of the set

$$\mathcal{M}(u) := \{\mathbf{x} \in \Omega : |u(\mathbf{x})| > 1\}.$$

For the proof, we refer to [91, p. 788] and [92, pp. 885-886].

Lemma 3.8. *Let $\tau \in (0, 4\varepsilon^3)$. For a given $u \in H^1(\Omega)$ and $(y_1, y_2) \in H^1(\Omega)^* \times H^1(\Omega)^*$, the optimization problem*

$$\begin{aligned}\min_{(\delta u, \delta p)\in H^1(\Omega)\times \mathcal{V}_0} \quad & \mathcal{J}(\delta u, \delta p) + c^{-1}(\chi_{\mathcal{M}(u)}\delta u, \delta u) - \langle y_2, \delta u\rangle \\ \text{subject to} \quad & \tau(\nabla \delta p, \nabla \phi) + (\delta u, \phi) = \langle y_1, \phi\rangle \quad \forall \phi \in H^1(\Omega)\end{aligned} \tag{$\mathcal{P}_{G_c}$}$$

admits a unique solution $(\delta u, \delta p)$. Moreover, there exists a unique $\delta w \in H^1(\Omega)$ such that

$$\tau(\nabla \delta w, \nabla \phi) + (\delta u, \phi) = \langle y_1, \phi\rangle, \tag{3.60}$$

$$\varepsilon(\nabla \delta u, \nabla \psi) + c^{-1}(\chi_{\mathcal{M}(u)}\delta u, \psi) - (\delta w, \psi) - \varepsilon^{-1}(\delta u, \psi) = \langle y_2, \psi\rangle \tag{3.61}$$

for all $\phi, \psi \in H^1(\Omega)$. Conversely, if $(\delta u, \delta w)$ is a solution of (3.60)–(3.61), then $(\delta u, \delta p)$ with $\delta p = \delta w - (\delta w, 1)$ is the unique solution of ($\mathcal{P}_{G_c}$).

For the proof, one proceeds as in the proofs of Theorem 3.4 and 3.5.

Lemma 3.9. *The SSN method (2.1) (with F and G replaced by F_c and G_c) converges superlinearly to (u_c, w_c), the solution of (3.59), provided that $\|(u^{(0)}, w^{(0)}) - (u_c, w_c)\|_{H^1(\Omega)\times H^1(\Omega)}$ is sufficiently small and $\tau \in (0, 4\varepsilon^3)$.*

Proof. From Lemma 3.8, we deduce that for all $(u, w) \in H^1(\Omega) \times H^1(\Omega)$, $G_c(u, w)$ is invertible, i.e., for given $(y_1, y_2) \in H^1(\Omega)^* \times H^1(\Omega)^*$, there exists a unique pair $(\delta u, \delta w) \in H^1(\Omega) \times H^1(\Omega)$, such that (3.60)–(3.61) is satisfied. Multiplying (3.60) by ε^{-1}, choosing $\phi = \delta u$, and applying Young's inequality with $\alpha_Y = 2\beta_1 > 0$ (Lemma 2.12) and $\beta_1 \in \left(\frac{\tau}{4\varepsilon^2}, \varepsilon\right)$ (note that $\tau < 4\varepsilon^3$), we get

$$\begin{aligned}\frac{1}{\varepsilon}\|\delta u\|^2 &= \frac{1}{\varepsilon}\langle y_1, \delta u\rangle - \frac{\tau}{\varepsilon}(\nabla\delta w, \nabla\delta u)\\ &\le \frac{1}{\varepsilon}\langle y_1, \delta u\rangle + \frac{\tau\beta_1}{\varepsilon}\|\nabla\delta w\|^2 + \frac{\tau}{4\beta_1\varepsilon}\|\nabla\delta u\|^2. \end{aligned}\tag{3.62}$$

Taking $(\phi, \psi) = (\delta w, \delta u)$ in (3.60)–(3.61) and adding the two equations, we obtain

$$\varepsilon\,\|\nabla\delta u\|^2 + \tau\,\|\nabla\delta w\|^2 = \langle y_1, \delta w\rangle + \langle y_2, \delta u\rangle - c^{-1}\underbrace{\left(\chi_{\mathcal{M}(u)}\delta u, \delta u\right)}_{\ge 0} + \varepsilon^{-1}\,\|\delta u\|^2\,.$$

Together with (3.62), this yields

$$\begin{aligned}&\underbrace{\left(\varepsilon - \frac{\tau}{4\beta_1\varepsilon}\right)}_{\ge 0}\|\nabla\delta u\|^2 + \underbrace{\left(\tau - \frac{\tau\beta_1}{\varepsilon}\right)}_{\ge 0}\|\nabla\delta w\|^2\\ &\qquad\le \langle y_1, \delta w\rangle + \langle y_2, \delta u\rangle + \frac{1}{\varepsilon}\langle y_1, \delta u\rangle\\ &\qquad\le \frac{1}{4\beta_2}\,\|\delta w\|_1^2 + \frac{1}{4\beta_3}\left(\|\delta u\|^2 + \|\nabla\delta u\|^2\right) + C\left(\left\|y_1\right\|_*^2 + \left\|y_2\right\|_*^2\right), \end{aligned}\tag{3.63}$$

where we have used Cauchy's inequality (Theorem 2.4) and Young's inequality with $\alpha_Y = 2\beta_2 > 0$ and $\alpha_Y = 2\beta_3 > 0$ (Lemma 2.12) in (3.63). The constant $C > 0$ possibly depends on $\varepsilon, \tau, c, \beta_1, \beta_2$, or β_3, but not on δu or δw. Taking $(\phi, \psi) = (1, 1)$ in (3.60)–(3.61), we get

$$(\delta u, 1) = \langle y_1, 1\rangle, \tag{3.64}$$

$$(\delta w, 1) = c^{-1}(\chi_{\mathcal{M}(u)}\delta u, 1) - \varepsilon^{-1}\langle y_1, 1\rangle - \langle y_2, 1\rangle. \tag{3.65}$$

The Poincaré inequality (Theorem 2.22) yields

$$\|\delta u\|^2 + \|\nabla\delta u\|^2 = \|\delta u\|_1^2 \le c_P\left[\|\nabla\delta u\|^2 + (\delta u, 1)^2\right] \overset{(3.64)}{=} c_P\left[\|\nabla\delta u\|^2 + \langle y_1, 1\rangle^2\right].$$

Cauchy's inequality (Theorem 2.4) applied to the last term gives

$$\langle y_1, 1\rangle^2 = \langle y_1, \langle y_1, 1\rangle\rangle \le \|y_1\|_*\,\langle y_1, 1\rangle\,\|1\|_1 \le \|y_1\|_*^2\,\|1\|_1^2 = \|y_1\|_*^2.$$

Note that we have made use of the assumption $1 = |\Omega| = \|1\|^2 = \|1\|_1^2$ imposed in Section 3.3.2. Altogether, we obtain

$$\|\delta u\|^2 \le c_P\left[\|\nabla\delta u\|^2 + \|y_1\|_*^2\right] - \|\nabla\delta u\|^2 \le c_P\left[\|\nabla\delta u\|^2 + \|y_1\|_*^2\right]. \tag{3.66}$$

The Poincaré inequality (Theorem 2.22) applied to δw yields

$$\|\delta w\|_1^2 \leq c_{\mathrm{P}} \left[\|\nabla \delta w\|^2 + (\delta w, 1)^2 \right]. \tag{3.67}$$

From (3.65), we obtain

$$\begin{aligned}
(\delta w, 1)^2 &= \left[\frac{1}{c} (\chi_{\mathcal{M}(u)} \delta u, 1) - \frac{1}{\varepsilon} \langle y_1, 1 \rangle - \langle y_2, 1 \rangle \right]^2 \\
&= \frac{1}{c^2} (\chi_{\mathcal{M}(u)} \delta u, 1)^2 - \frac{2}{c\varepsilon} (\chi_{\mathcal{M}(u)} \delta u, 1) \langle y_1, 1 \rangle - \frac{2}{c} (\chi_{\mathcal{M}(u)} \delta u, 1) \langle y_2, 1 \rangle + \frac{1}{\varepsilon^2} \langle y_1, 1 \rangle^2 \\
&\quad + \frac{2}{\varepsilon} \langle y_1, 1 \rangle \langle y_2, 1 \rangle + \langle y_2, 1 \rangle^2 \\
&= \frac{1}{c^2} (\chi_{\mathcal{M}(u)} \delta u, (\chi_{\mathcal{M}(u)} \delta u, 1)) - \frac{2}{c\varepsilon} (\chi_{\mathcal{M}(u)} \delta u, \langle y_1, 1 \rangle) - \frac{2}{c} (\chi_{\mathcal{M}(u)} \delta u, \langle y_2, 1 \rangle) \\
&\quad + \frac{1}{\varepsilon^2} \langle y_1, \langle y_1, 1 \rangle \rangle + \frac{2}{\varepsilon} \langle y_1, \langle y_2, 1 \rangle \rangle + \langle y_2, \langle y_2, 1 \rangle \rangle.
\end{aligned} \tag{3.68}$$

Now, we apply Cauchy's inequality (Theorem 2.4) to each of the terms in (3.68). Further, we again make use of the assumption $1 = |\Omega| = \|1\|^2 = \|1\|_1^2$ imposed in Section 3.3.2. We obtain

$$(\chi_{\mathcal{M}(u)} \delta u, (\chi_{\mathcal{M}(u)} \delta u, 1)) \leq \|\chi_{\mathcal{M}(u)} \delta u\| \, (\chi_{\mathcal{M}(u)} \delta u, 1) \, \|1\| \leq \|\chi_{\mathcal{M}(u)} \delta u\|^2 \, \|1\|^2 \leq \|\delta u\|^2,$$

$$\begin{aligned}
(-\chi_{\mathcal{M}(u)} \delta u, \langle y_1, 1 \rangle) &\leq \|\chi_{\mathcal{M}(u)} \delta u\| \, \langle y_1, 1 \rangle \, \|1\| \leq \|\chi_{\mathcal{M}(u)} \delta u\| \, \|y_1\|_* \, \|1\|^2 \\
&\leq \delta_2 \|\chi_{\mathcal{M}(u)} \delta u\|^2 + \frac{1}{4\delta_2} \|y_1\|_*^2 \leq \delta_2 \|\delta u\|^2 + \frac{1}{4\delta_2} \|y_1\|_*^2,
\end{aligned} \tag{3.69}$$

$$\begin{aligned}
(-\chi_{\mathcal{M}(u)} \delta u, \langle y_2, 1 \rangle) &\leq \|\chi_{\mathcal{M}(u)} \delta u\| \, \langle y_2, 1 \rangle \, \|1\| \leq \|\chi_{\mathcal{M}(u)} \delta u\| \, \|y_2\|_* \, \|1\|^2 \\
&\leq \delta_3 \|\chi_{\mathcal{M}(u)} \delta u\|^2 + \frac{1}{4\delta_3} \|y_2\|_*^2 \leq \delta_3 \|\delta u\|^2 + \frac{1}{4\delta_3} \|y_2\|_*^2,
\end{aligned} \tag{3.70}$$

$$\langle y_1, \langle y_1, 1 \rangle \rangle \leq \|y_1\|_* \, \langle y_1, 1 \rangle \, \|1\|_1 \leq \|y_1\|_*^2 \, \|1\|_1^2 = \|y_1\|_*^2,$$

$$\langle y_2, \langle y_2, 1 \rangle \rangle \leq \|y_2\|_* \, \langle y_2, 1 \rangle \, \|1\|_1 \leq \|y_2\|_*^2 \, \|1\|_1^2 = \|y_2\|_*^2,$$

$$\begin{aligned}
\langle y_1, \langle y_2, 1 \rangle \rangle &\leq \|y_1\|_* \, \langle y_2, 1 \rangle \, \|1\|_1 \leq \|y_1\|_* \|y_2\|_* \, \|1\|_1^2 \\
&\leq \delta_1 \|y_1\|_*^2 + \frac{1}{4\delta_1} \|y_2\|_*^2,
\end{aligned} \tag{3.71}$$

where we have applied Young's inequality (Lemma 2.12) with $\alpha_{\mathrm{Y}} = 2\delta_2 > 0$ in (3.69), $\alpha_{\mathrm{Y}} = 2\delta_3 > 0$ in (3.70), and $\alpha_{\mathrm{Y}} = 2\delta_1 > 0$ in (3.71). Substituting the inequalities above into (3.68) gives

$$\begin{aligned}
(\delta w, 1)^2 \leq \frac{1}{c} \left(\frac{1}{c} + \frac{2\delta_2}{\varepsilon} + 2\delta_3 \right) \|\delta u\|^2 + \frac{1}{\varepsilon} \left(\frac{1}{2c\delta_2} + \frac{1}{\varepsilon} + 2\delta_1 \right) \|y_1\|_*^2 \\
+ \left(\frac{1}{2c\delta_3} + 1 + \frac{1}{2\varepsilon\delta_1} \right) \|y_2\|_*^2.
\end{aligned}$$

Hence, we get in (3.67)

$$\begin{aligned}
\|\delta w\|_1^2 \leq c_{\mathrm{P}} \Bigg[&\|\nabla \delta w\|^2 + \frac{1}{c} \left(\frac{1}{c} + \frac{2\delta_2}{\varepsilon} + 2\delta_3 \right) \|\delta u\|^2 \\
&+ \frac{1}{\varepsilon} \left(\frac{1}{2c\delta_2} + \frac{1}{\varepsilon} + 2\delta_1 \right) \|y_1\|_*^2 + \left(\frac{1}{2c\delta_3} + 1 + \frac{1}{2\varepsilon\delta_1} \right) \|y_2\|_*^2 \Bigg].
\end{aligned} \tag{3.72}$$

Substituting (3.72) into (3.63) yields

$$\begin{aligned}
&\underbrace{\left(\varepsilon - \frac{\tau}{4\beta_1\varepsilon}\right)}_{\geq 0}\|\nabla\delta u\|^2 + \underbrace{\left(\tau - \frac{\tau\beta_1}{\varepsilon}\right)}_{\geq 0}\|\nabla\delta w\|^2 \\
&\quad\leq \frac{c_{\mathrm{P}}}{4\beta_2}\|\nabla\delta w\|^2 + \frac{1}{4\beta_3}\|\nabla\delta u\|^2 + \left[\frac{1}{4\beta_3} + \frac{c_{\mathrm{P}}}{4c\beta_2}\left(\frac{1}{c} + \frac{2\delta_2}{\varepsilon} + 2\delta_3\right)\right]\|\delta u\|^2 \\
&\qquad + \left[\frac{c_{\mathrm{P}}}{4\varepsilon\beta_2}\left(\frac{1}{2c\delta_2} + \frac{1}{\varepsilon} + 2\delta_1\right) + C\right]\|y_1\|_*^2 + \left[\frac{c_{\mathrm{P}}}{4\beta_2}\left(\frac{1}{2c\delta_3} + 1 + \frac{1}{2\varepsilon\delta_1}\right) + C\right]\|y_2\|_*^2 \\
&\quad\leq \frac{c_{\mathrm{P}}}{4\beta_2}\|\nabla\delta w\|^2 + \left[\frac{1 + c_{\mathrm{P}}}{4\beta_3} + \frac{c_{\mathrm{P}}^2}{4c\beta_2}\left(\frac{1}{c} + \frac{2\delta_2}{\varepsilon} + 2\delta_3\right)\right]\|\nabla\delta u\|^2 \\
&\qquad + \left[\frac{c_{\mathrm{P}}}{4\beta_3} + \frac{c_{\mathrm{P}}^2}{4c\beta_2}\left(\frac{1}{c} + \frac{2\delta_2}{\varepsilon} + 2\delta_3\right) + \frac{c_{\mathrm{P}}}{4\varepsilon\beta_2}\left(\frac{1}{2c\delta_2} + \frac{1}{\varepsilon} + 2\delta_1\right) + C\right]\|y_1\|_*^2 \\
&\qquad + \left[\frac{c_{\mathrm{P}}}{4\beta_2}\left(\frac{1}{2c\delta_3} + 1 + \frac{1}{2\varepsilon\delta_1}\right) + C\right]\|y_2\|_*^2,
\end{aligned} \tag{3.73}$$

where we have used (3.66) in (3.73). This can be written as

$$\begin{aligned}
&\left[\varepsilon - \frac{\tau}{4\beta_1\varepsilon} - \frac{c_{\mathrm{P}}^2}{4c\beta_2}\left(\frac{1}{c} + \frac{2\delta_2}{\varepsilon} + 2\delta_3\right) - \frac{1 + c_{\mathrm{P}}}{4\beta_3}\right]\|\nabla\delta u\|^2 \\
&\qquad\qquad + \left(\tau - \frac{\tau\beta_1}{\varepsilon} - \frac{c_{\mathrm{P}}}{4\beta_2}\right)\|\nabla\delta w\|^2 \leq C_1\left(\left\|y_1\right\|_*^2 + \left\|y_2\right\|_*^2\right).
\end{aligned} \tag{3.74}$$

The constant $C_1 > 0$ possibly depends on $\varepsilon, \tau, c, c_{\mathrm{P}}, \beta_1, \beta_2, \beta_3, \delta_1, \delta_2, \delta_3$, but not on δu or δw. Now, we have to choose $\beta_2, \beta_3, \delta_1, \delta_2, \delta_3$ such that

$$\tau - \frac{\tau\beta_1}{\varepsilon} - \frac{c_{\mathrm{P}}}{4\beta_2} > 0, \tag{3.75}$$

$$\varepsilon - \frac{\tau}{4\beta_1\varepsilon} - \frac{c_{\mathrm{P}}^2}{4c\beta_2}\left(\frac{1}{c} + \frac{2\delta_2}{\varepsilon} + 2\delta_3\right) - \frac{1 + c_{\mathrm{P}}}{4\beta_3} > 0. \tag{3.76}$$

Due to our choice of β_1, it holds $\tau - \frac{\tau\beta_1}{\varepsilon} > 0$ and $\varepsilon - \frac{\tau}{4\beta_1\varepsilon} > 0$. Hence, (3.75) yields

$$\beta_2 > \frac{\varepsilon\, c_{\mathrm{P}}}{4\tau(\varepsilon - \beta_1)}. \tag{3.77}$$

Next, in (3.76), we first require

$$\varepsilon - \frac{\tau}{4\beta_1\varepsilon} > \frac{c_{\mathrm{P}}^2}{4c\beta_2}\left(\frac{1}{c} + \frac{2\delta_2}{\varepsilon} + 2\delta_3\right),$$

which leads to the choice

$$\beta_2 > \frac{c_{\mathrm{P}}^2\varepsilon\beta_1\left(\frac{1}{c} + \frac{2\delta_2}{\varepsilon} + 2\delta_3\right)}{c\,(4\varepsilon^2\beta_1 - \tau)}$$

for arbitrary $\delta_2, \delta_3 > 0$. Together with (3.77), we have to choose $\beta_2 > 0$ with

$$\beta_2 > \max\left(\frac{\varepsilon\, c_{\mathrm{P}}}{4\tau(\varepsilon - \beta_1)}, \frac{c_{\mathrm{P}}^2\varepsilon\beta_1\left(\frac{1}{c} + \frac{2\delta_2}{\varepsilon} + 2\delta_3\right)}{c\,(4\varepsilon^2\beta_1 - \tau)}\right).$$

Second, we require in (3.76)

$$\varepsilon - \frac{\tau}{4\beta_1\varepsilon} - \frac{c_{\mathrm{P}}^2}{4c\beta_2}\left(\frac{1}{c} + \frac{2\delta_2}{\varepsilon} + 2\delta_3\right) > \frac{1 + c_{\mathrm{P}}}{4\beta_3},$$

which leads to

$$\beta_3 > \frac{\varepsilon c\beta_1\beta_2(1 + c_{\mathrm{P}})}{4c\varepsilon^2\beta_1\beta_2 - \tau c\beta_2 - \varepsilon c_{\mathrm{P}}^2\beta_1\left(\frac{1}{c} + \frac{2\delta_2}{\varepsilon} + 2\delta_3\right)}.$$

Finally, we can write (3.74) as

$$C_2\left(\|\nabla\delta u\|^2 + \|\nabla\delta u\|^2\right) \leq C_1\left(\left\|y_1\right\|_*^2 + \left\|y_2\right\|_*^2\right),$$

where $C_2 > 0$ with

$$C_2 = \min\left(\varepsilon - \frac{\tau}{4\beta_1\varepsilon} - \frac{c_{\mathrm{P}}^2}{4c\beta_2}\left(\frac{1}{c} + \frac{2\delta_2}{\varepsilon} + 2\delta_3\right) - \frac{1 + c_{\mathrm{P}}}{4\beta_3}, \tau - \frac{\tau\beta_1}{\varepsilon} - \frac{c_{\mathrm{P}}}{4\beta_2}\right).$$

Hence

$$\|\nabla\delta u\|^2 + \|\nabla\delta w\|^2 \leq \tilde{C}\left(\left\|y_1\right\|_*^2 + \left\|y_2\right\|_*^2\right). \tag{3.78}$$

This leads to

$$\begin{aligned}\|(\delta u, \delta w)\|_{1\times 1} &= \sqrt{\|\delta u\|_1^2 + \|\delta w\|_1^2} = \sqrt{\|\delta u\|^2 + \|\nabla\delta u\|^2 + \|\delta w\|_1^2} \\ &\leq \sqrt{\hat{C}\left(\left\|y_1\right\|_*^2 + \left\|y_2\right\|_*^2\right)},\end{aligned} \tag{3.79}$$

where we obtained (3.79) after applying first (3.72), second (3.66), and finally (3.78). For $\max\left(\left\|y_1\right\|_*, \left\|y_2\right\|_*\right) \leq \beta$ for some constant $\beta > 0$, we consequently have

$$\begin{aligned}\left\|G_c^{-1}(u,w)\right\|_{\mathcal{L}((H^1(\Omega)^*)^2,(H^1(\Omega))^2)} &= \sup_{(y_1,y_2)\neq(0,0)} \frac{\left\|G_c^{-1}(u,w)(y_1,y_2)\right\|_{1\times 1}}{(y_1,y_2)_{*\times *}} \\ &= \sup_{(y_1,y_2)\neq(0,0)} \frac{\|(\delta u, \delta w)\|_{1\times 1}}{\sqrt{\|y_1\|_*^2 + \|y_2\|_*^2}} \\ &\leq \sup_{(y_1,y_2)\neq(0,0)} \frac{\sqrt{\hat{C}\left(\left\|y_1\right\|_*^2 + \left\|y_2\right\|_*^2\right)}}{\sqrt{\|y_1\|_*^2 + \|y_2\|_*^2}} \\ &= \sqrt{\hat{C}}\end{aligned} \tag{3.80}$$

with some constant $\hat{C} > 0$ possibly depending on $\varepsilon, \tau, c, c_{\mathrm{P}}, \beta_1, \beta_2, \beta_3, \delta_1, \delta_2, \delta_3$, but not on u or w. In (3.80), we have used (3.79). Thus, F_c with associated Newton derivative G_c fulfills the conditions of Theorem 2.8, which completes the proof. □

This finishes our analysis of the function space-based algorithm for solving the time-discrete Cahn–Hilliard problem (3.31)–(3.33). In the next section, we derive the fully discrete problems for both, the smooth system in (3.17)–(3.18) and the regularized nonsmooth system in (3.49)–(3.50).

3.6 Finite element approximation

In this section, we apply FEM [144] to the regularized nonsmooth Cahn–Hilliard system in (3.49)–(3.50). We also want to apply it to the smooth version (3.17)–(3.18). Since both procedures are similar, we only present the methodology based on the nonsmooth setting. Regarding the smooth case, we will state the fully discrete linear system at the end of this section.

In the following, we assume for simplicity that Ω is a polyhedral domain. Generalizations to curved domains are possible using boundary finite elements with curved faces. Let $\{\mathcal{R}_h\}_{h>0}$ be a triangulation of Ω into disjoint open rectangular elements[2] with maximal element size h. Let J_h be the set of nodes of $\mathcal{R}_h$ and $p_j \in J_h$ be the coordinates of these nodes. We approximate the infinite-dimensional space $H^1(\Omega)$ by the finite-dimensional space

$$S_h := \{\phi \in C^0(\overline{\Omega}) : \phi\,|_R \in Q_1(R) \ \ \forall R \in \mathcal{R}_h\} \subset H^1(\Omega)$$

of continuous, piecewise multilinear functions. For instance, in the two-dimensional case $d = 2$, we use bilinear functions, i.e., $Q_1 = \operatorname{span}\{x^{\alpha_i} y^{\alpha_i} : \alpha_i \in \{0,1\},\, i = 1,2\}$. We denote the standard nodal basis functions of S_h by φ_j for all $j \in J_h$. They have the property $\varphi_j(p_i) = \delta_{ij}$, $i, j = 1, \ldots, m$. The discretized version of the problem (3.49)–(3.50) is the following: Given $u_h^{\text{old}} \in S_h$, find $(u_{c,h}, w_{c,h}) \in S_h \times S_h$ such that

$$\left\langle F_{c,h}^{(1)}(u_{c,h}, w_{c,h}), v_h \right\rangle = 0 \quad \forall v_h \in S_h, \tag{3.81}$$

$$\left\langle F_{c,h}^{(2)}(u_{c,h}, w_{c,h}), v_h \right\rangle = 0 \quad \forall v_h \in S_h, \tag{3.82}$$

where the components are

$$\begin{aligned} \left\langle F_{c,h}^{(1)}(u_{c,h}, w_{c,h}), v_h \right\rangle &= \tau(\nabla w_{c,h}, \nabla v_h) + (u_{c,h}, v_h)_h - (u_h^{\text{old}}, v_h)_h, \\ \left\langle F_{c,h}^{(2)}(u_{c,h}, w_{c,h}), v_h \right\rangle &= \varepsilon(\nabla u_{c,h}, \nabla v_h) + c^{-1}(\max(0, u_{c,h} - 1) + \min(0, u_{c,h} + 1), v_h)_h \\ &\quad - (w_{c,h}, v_h)_h - \varepsilon^{-1}(u_{c,h}, v_h)_h. \end{aligned}$$

The semi-inner product $(\cdot,\cdot)_h$ on $C_0(\overline{\Omega})$ is defined by

$$(f, g)_h := \int_\Omega \pi_h(f(\mathbf{x})g(\mathbf{x}))\,\mathrm{d}\mathbf{x} = \sum_{i=1}^{m} (1, \varphi_i) f(p_i) g(p_i) \quad \forall f, g \in C_0(\overline{\Omega}),$$

where $\pi_h \colon C_0(\overline{\Omega}) \to S_h$ is the Lagrange interpolation operator. Within our finite element framework, for a given $(u_h, w_h) \in S_h \times S_h$, every step of the SSN method for solving (3.81)–(3.82) requires to compute $(\delta u_h, \delta w_h) \in S_h \times S_h$ satisfying

$$\tau(\nabla \delta w_h, \nabla v_h) + (\delta u_h, v_h)_h = -F_{c,h}^{(1)}(u_h, w_h), \tag{3.83}$$

$$\varepsilon(\nabla \delta u_h, \nabla v_h) + c^{-1}(\chi^h_{\mathcal{M}(u_h)} \delta u_h, v_h)_h - (\delta w_h, v_h)_h - \varepsilon^{-1}(\delta u_h, v_h)_h = -F_{c,h}^{(2)}(u_h, w_h), \tag{3.84}$$

for all $v_h \in S_h$, where $\chi^h_{\mathcal{M}(u_h)} := \sum_{i=1}^m \chi^h_{\mathcal{M}(u_h)}(p_i)\varphi_i$ with $\chi^h_{\mathcal{M}(u_h)}(p_i) = 0$ if $-1 \le u_h(p_i) \le 1$ and $\chi^h_{\mathcal{M}(u_h)}(p_i) = 1$ otherwise. If we now write a function $v_h \in S_h$ by $v_h = \sum_{j \in J_h} v_{h,j}\,\varphi_j$

[2] The use of rectangles is motivated by performing the implementation with deal.II [8].

and denote the vector of coefficients by **v**, the fully discrete linear system per SSN step reads in matrix form as

$$\begin{bmatrix} -\varepsilon \boldsymbol{K} - c^{-1}\boldsymbol{G} + \varepsilon^{-1}\boldsymbol{M} & \boldsymbol{M} \\ \boldsymbol{M} & \tau \boldsymbol{K} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}^{(k+1)} \\ \boldsymbol{w}^{(k+1)} \end{bmatrix} = \begin{bmatrix} -c^{-1}\left(\boldsymbol{G}_+\boldsymbol{1} - \boldsymbol{G}_-\boldsymbol{1}\right) \\ \boldsymbol{M}\boldsymbol{u}^{\text{old}} \end{bmatrix}. \tag{3.85}$$

Here, $\boldsymbol{u}^{(k+1)}, \boldsymbol{w}^{(k+1)} \in \mathbb{R}^m$ and $\boldsymbol{u}^{\text{old}} \in \mathbb{R}^m$ is the solution vector from the previous time step. Remember that k denotes the iteration step of the SSN method. The lumped mass matrix and the stiffness matrix are defined as

$$\begin{aligned} \boldsymbol{M} &:= ((\varphi_i, \varphi_j)_h)_{i,j=1,\dots,m} = \operatorname{diag}((1, \varphi_i))_{i=1,\dots,m} \in \mathbb{R}^{m\times m}, \\ \boldsymbol{K} &:= ((\nabla\varphi_i, \nabla\varphi_j))_{i,j=1,\dots,m} \in \mathbb{R}^{m\times m}. \end{aligned}$$

Note that $\boldsymbol{M}$ is a diagonal, symmetric positive definite matrix and $\boldsymbol{K}$ is symmetric positive semidefinite. In particular, they have the following eigenvalue characterization.

Proposition 3.10 ([63, pp. 57–60]). *Let $d \in \{2, 3\}$ be the spatial dimension.*

$$\begin{aligned} \tilde{c}h^d \le \frac{(\boldsymbol{M}\boldsymbol{v}, \boldsymbol{v})}{(\boldsymbol{v}, \boldsymbol{v})} &\le Ch^d, \\ 0 \le \frac{(\boldsymbol{K}\boldsymbol{v}, \boldsymbol{v})}{(\boldsymbol{v}, \boldsymbol{v})} &\le \tilde{C}h^{d-2}, \end{aligned}$$

for all $\boldsymbol{v} \in \mathbb{R}^m$. The constants $\tilde{c}, C, \tilde{C}$ are positive and independent of h. In particular, $\boldsymbol{K}$ has a one-dimensional kernel spanned by the constant vector $\boldsymbol{1} = [1, \dots, 1]^T \in \mathbb{R}^m$.

In terms of the condition number, $\kappa(\boldsymbol{M}) \le \tilde{c}^{-1}C$ and $\kappa(\boldsymbol{K}) = \infty$. Note that the original version of Proposition 3.10 in [63, pp. 57–60] is actually stated under proper mesh subdivisions. We will not go into these details but note that they hold true in our setting. Moreover, the stiffness matrix in [63, pp. 57–60] is symmetric positive definite due to the imposition of Dirichlet boundary conditions instead of Neumann ones. The matrix representations coming from the generalized derivative of the term $(\chi^h_{\mathcal{M}(u_h)}\delta u_h, v_h)_h$ are the following diagonal matrices

$$\begin{aligned} \boldsymbol{G} = \boldsymbol{G}(\boldsymbol{u}^{(k)}) &= \operatorname{diag}\begin{pmatrix} [\boldsymbol{M}]_{ii} & \text{if } |u^{(k)}_{h,i}| > 1, \\ 0 & \text{otherwise} \end{pmatrix}_{i=1,\dots,m} \in \mathbb{R}^{m\times m}, \\ \boldsymbol{G}_+ = \boldsymbol{G}_+(\boldsymbol{u}^{(k)}) &= \operatorname{diag}\begin{pmatrix} [\boldsymbol{M}]_{ii} & \text{if } u^{(k)}_{h,i} > 1, \\ 0, & \text{otherwise} \end{pmatrix}_{i=1,\dots,m} \in \mathbb{R}^{m\times m}, \\ \boldsymbol{G}_- = \boldsymbol{G}_-(\boldsymbol{u}^{(k)}) &= \operatorname{diag}\begin{pmatrix} [\boldsymbol{M}]_{ii} & \text{if } u^{(k)}_{h,i} < -1 \\ 0 & \text{otherwise} \end{pmatrix}_{i=1,\dots,m} \in \mathbb{R}^{m\times m}, \end{aligned}$$

where $\boldsymbol{u}^{(k)} = \left[u^{(k)}_{h,1}, \dots, u^{(k)}_{h,m}\right]^T$ is the solution from the SSN step k.

At the very beginning of Section 3.1, we mentioned that we are also interested in the FEM solution of the semi-implicit time-discrete Cahn–Hilliard problem. Remember that we have motivated the use of an implicit scheme due to accuracy requirements. However, the bottleneck of the time step restriction $\tau < 4\varepsilon^3$ arose. In contrast, no time step restrictions are imposed on the semi-implicit scheme. The corresponding fully discrete linear system per SSN step reads in matrix form as

$$\begin{bmatrix} -\varepsilon \boldsymbol{K} - c^{-1}\boldsymbol{G} & \boldsymbol{M} \\ \boldsymbol{M} & \tau \boldsymbol{K} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}^{(k+1)} \\ \boldsymbol{w}^{(k+1)} \end{bmatrix} = \begin{bmatrix} -c^{-1}\left(\boldsymbol{G}_{+}\boldsymbol{1} - \boldsymbol{G}_{-}\boldsymbol{1}\right) - \varepsilon^{-1}\boldsymbol{M}\boldsymbol{u}^{\text{old}} \\ \boldsymbol{M}\boldsymbol{u}^{\text{old}} \end{bmatrix}. \tag{3.86}$$

We also apply FEM to the implicit time-discrete smooth Cahn–Hilliard system (3.17)–(3.18). The fully discrete linear system per Newton step reads in matrix form as

$$\begin{bmatrix} -\varepsilon \boldsymbol{K} - \varepsilon^{-1}\boldsymbol{F} & \boldsymbol{M} \\ \boldsymbol{M} & \tau \boldsymbol{K} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}^{(k+1)} \\ \boldsymbol{w}^{(k+1)} \end{bmatrix} = \begin{bmatrix} -2\varepsilon^{-1}\boldsymbol{M}\left(\boldsymbol{u}^{(k)}\right)^{3} \\ \boldsymbol{M}\boldsymbol{u}^{\text{old}} \end{bmatrix}. \tag{3.87}$$

The matrix representation coming from the derivative of the potential term is the diagonal matrix

$$\boldsymbol{F} = \boldsymbol{F}(\boldsymbol{u}^{(k)}) = \operatorname{diag}\left([\boldsymbol{M}]_{ii}\left[3\left(u_{h,i}^{(k)}\right)^{2} - 1\right]\right)_{i=1,\ldots,m} \in \mathbb{R}^{m\times m}.$$

Note that the powers of the form in $\left(\boldsymbol{u}^{(k)}\right)^{p}, p \in \mathbb{N}$, have to be understood elementwise. The use of a semi-implicit time discretization scheme, where the potential term $\psi'_{\text{pol}}(u)$ is treated explicitly, leads to a similar time step restriction as we have in the fully implicit case. We refer the reader to [140, p. 12] for the analysis. A scheme, which is known to be unconditional gradient stable, is the convexity splitting method by Eyre [67] presented in Chapter 2.1.5. Eyre [67, p. 12] proposed the following splitting of the potential term $\psi'_{\text{pol}}(u) = u^3 - u$: The first part, u^3, is treated implicitly and the second part, u, is treated explicitly. This leads to the linear system

$$\begin{bmatrix} -\varepsilon \boldsymbol{K} - \varepsilon^{-1}\boldsymbol{H} & \boldsymbol{M} \\ \boldsymbol{M} & \tau \boldsymbol{K} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}^{(k+1)} \\ \boldsymbol{w}^{(k+1)} \end{bmatrix} = \begin{bmatrix} -2\varepsilon^{-1}\boldsymbol{M}\left(\boldsymbol{u}^{(k)}\right)^{3} - \varepsilon^{-1}\boldsymbol{M}\boldsymbol{u}^{\text{old}} \\ \boldsymbol{M}\boldsymbol{u}^{\text{old}} \end{bmatrix} \tag{3.88}$$

per Newton step. The matrix representation coming from the derivative of the potential term u^3 is the diagonal matrix

$$\boldsymbol{H} = \boldsymbol{H}(\boldsymbol{u}^{(k)}) = \operatorname{diag}\left([\boldsymbol{M}]_{ii} 3\left(u_{h,i}^{(k)}\right)^{2}\right)_{i=1,\ldots,m} \in \mathbb{R}^{m\times m}.$$

It holds

$$3\left(u_{h,i}^{(k)}\right)^{2} \in [0,\, 3\,\|\boldsymbol{u}^{(k)}\|_{\infty}^{2}] =: [0, \alpha] \tag{3.89}$$

for $i = 1, \ldots, m$.

Now, we have arrived at the core of our numerical algorithms — the numerical solution of systems of linear equations. Due to the use of FEM, all the matrix blocks $\boldsymbol{M}, \boldsymbol{K}, \boldsymbol{G}, \boldsymbol{F}$, and $\boldsymbol{H}$ are large and sparse. In the next section, we design optimal practical preconditioners for each of the four linear systems (3.85)–(3.88).

3.7 Preconditioning

This section is devoted to the development of practical preconditioners for the efficient solution of the four linear systems (3.85)–(3.88). We begin with the simplest problem and go step by step to the next harder one. Note that the construction of efficient preconditioners in the smooth case is already well established by Boyanova et al. [35, 37, 36, 3]. The authors discussed among others the fully implicit time-discrete scheme. We apply their main procedure to the semi-implicit scheme (3.88). However, our theoretical proofs differ halfway through. Moreover, we will generalize these theoretical results to vector-valued problems in Chapter 4.7. Note that this technique ignores the symmetry inherent in our coefficient matrices. Hence, we develop another type of preconditioner, which preserves the symmetry. This allows us to make use of symmetric Krylov subspace solvers, which are cheaper than the nonsymmetric ones. In the following, we start with such a symmetry preserving preconditioner.

3.7.1 Smooth semi-implicit systems

The smooth semi-implicit Cahn–Hilliard system (3.88) can be written as

$$\begin{aligned}\begin{bmatrix} \boldsymbol{A} & \boldsymbol{M} \\ \boldsymbol{M} & -\tau\boldsymbol{K} \end{bmatrix}\begin{bmatrix} \boldsymbol{u}^{(k+1)} \\ -\boldsymbol{w}^{(k+1)} \end{bmatrix} &:= \begin{bmatrix} \varepsilon\boldsymbol{K}+\varepsilon^{-1}\boldsymbol{H} & \boldsymbol{M} \\ \boldsymbol{M} & -\tau\boldsymbol{K} \end{bmatrix}\begin{bmatrix} \boldsymbol{u}^{(k+1)} \\ -\boldsymbol{w}^{(k+1)} \end{bmatrix} \\ &= \begin{bmatrix} 2\varepsilon^{-1}\boldsymbol{M}\left(\boldsymbol{u}^{(k)}\right)^3+\varepsilon^{-1}\boldsymbol{M}\boldsymbol{u}^{\text{old}} \\ \boldsymbol{M}\boldsymbol{u}^{\text{old}} \end{bmatrix}\end{aligned} \tag{3.90}$$

and is hence of saddle point form. In the following, we denote the coefficient matrix by $\mathcal{A}$. It can be easily seen that $\mathcal{A}$ is symmetric. The matrix $\boldsymbol{H}$ cannot be identical to the zero matrix since this would imply that the solution $\boldsymbol{u}^{(k)}$ from the previous Newton step k is identical to the zero vector. Hence, the $(1,1)$ block $\boldsymbol{A}$ is in general symmetric positive definite. According to Theorem 2.35, $\mathcal{A}$ is nonsingular. Due to Remark 2.3, the Schur complement

$$\boldsymbol{S} = -(\tau\boldsymbol{K}+\boldsymbol{M}\boldsymbol{A}^{-1}\boldsymbol{M})$$

is symmetric negative definite, and $\mathcal{A}$ is indefinite with m positive and m negative eigenvalues.

Next, we design a preconditioner. Since $\mathcal{A}$ is symmetric indefinite, our Krylov method of choice is MINRES. Hence, we need to construct a symmetric positive definite preconditioner. We propose the block diagonal preconditioner

$$\mathcal{P} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{0} \\ \boldsymbol{0} & -\hat{\boldsymbol{S}} \end{bmatrix}. \tag{3.91}$$

As Schur complement approximation, we design $\hat{\boldsymbol{S}}$ as

$$\begin{aligned}\hat{\boldsymbol{S}} &= -\boldsymbol{S}_1\boldsymbol{A}^{-1}\boldsymbol{S}_1 \\ &= -\left(\boldsymbol{M}+\sqrt{\tau\varepsilon}\boldsymbol{K}\right)\boldsymbol{A}^{-1}\left(\boldsymbol{M}+\sqrt{\tau\varepsilon}\boldsymbol{K}\right) && (3.92) \\ &= -\tau\varepsilon\boldsymbol{K}\boldsymbol{A}^{-1}\boldsymbol{K}-\boldsymbol{M}\boldsymbol{A}^{-1}\boldsymbol{M}-\sqrt{\tau\varepsilon}\boldsymbol{M}\boldsymbol{A}^{-1}\boldsymbol{K}-\sqrt{\tau\varepsilon}\boldsymbol{K}\boldsymbol{A}^{-1}\boldsymbol{M}. && (3.93)\end{aligned}$$

The second term in (3.93) matches the second term in the exact Schur complement. The first term in (3.93) approximates the first term in the exact Schur complement. Due to the balanced distribution of $\tau\varepsilon$ in form of $\sqrt{\tau\varepsilon}$ in the factor $\boldsymbol{S}_1$, the influence of both remainder terms in (3.93) is reduced.

Lemma 3.11. *$\hat{\boldsymbol{S}}$ is symmetric negative definite.*

Proof. It can be easily seen that $\hat{\boldsymbol{S}}$ is symmetric. Let $\mathbf{0} \neq \boldsymbol{v} \in \mathbb{R}^m$. We introduce the vectors $\boldsymbol{a} = \boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{M}\boldsymbol{v}$ and $\boldsymbol{b} = \sqrt{\tau\varepsilon}\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{K}\boldsymbol{v}$. Hence, we can write

$$\boldsymbol{v}^T\hat{\boldsymbol{S}}\boldsymbol{v} = -\boldsymbol{a}^T\boldsymbol{a} - \boldsymbol{a}^T\boldsymbol{b} - \boldsymbol{b}^T\boldsymbol{a} - \boldsymbol{b}^T\boldsymbol{b} = -(\boldsymbol{a}+\boldsymbol{b})^T(\boldsymbol{a}+\boldsymbol{b}) = -\|\boldsymbol{a}+\boldsymbol{b}\|^2 \leq 0.$$

It remains to show that $\boldsymbol{v}^T\hat{\boldsymbol{S}}\boldsymbol{v} < 0$. It holds $\|\boldsymbol{a}+\boldsymbol{b}\| = 0$ if and only if $\boldsymbol{a}+\boldsymbol{b} = \mathbf{0}$. Assume there exists a vector $\mathbf{0} \neq \boldsymbol{v} \in \mathbb{R}^m$ such that $\boldsymbol{a} = -\boldsymbol{b}$. This is equivalent to

$$\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{M}\boldsymbol{v} = -\sqrt{\tau\varepsilon}\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{K}\boldsymbol{v}.$$

If we multiply this equation from the left by $-(\tau\varepsilon)^{-\frac{1}{2}}\boldsymbol{M}^{-1}\boldsymbol{A}^{\frac{1}{2}}$, we obtain

$$\boldsymbol{M}^{-1}\boldsymbol{K}\boldsymbol{v} = -(\tau\varepsilon)^{-\frac{1}{2}}\boldsymbol{v}.$$

This means $-(\tau\varepsilon)^{-\frac{1}{2}} \in \sigma(\boldsymbol{M}^{-1}\boldsymbol{K})$. However, due to Lemma 2.30, it holds $\sigma(\boldsymbol{M}^{-1}\boldsymbol{K}) \subset \mathbb{R}_{\geq 0}$. Hence, $\boldsymbol{v}^T\hat{\boldsymbol{S}}\boldsymbol{v} < 0$ for all $\mathbf{0} \neq \boldsymbol{v} \in \mathbb{R}^m$. □

To illustrate the performance of $\hat{\boldsymbol{S}}^{-1}\boldsymbol{S}$, we show eigenvalue plots in Section 3.8.1. Let us conclude the discussion of the preconditioner $\boldsymbol{\mathcal{P}}$ with a statement about its practical realization. The action of the inverse of $\boldsymbol{S}_1$ is performed with an AMG since $\boldsymbol{S}_1$ forms the discretization of an elliptic operator. The same holds for the $(1,1)$ block $\boldsymbol{A}$. Hence, the practical block diagonal preconditioner is given by

$$\boldsymbol{\mathcal{P}}_0 = \begin{bmatrix} \boldsymbol{A}_0 & \mathbf{0} \\ \mathbf{0} & -\boldsymbol{S}_0 \end{bmatrix},$$

where $\boldsymbol{A}_0 = \mathrm{AMG}(\boldsymbol{A})$ and $\boldsymbol{S}_0 = \mathrm{AMG}(\boldsymbol{S}_1)\boldsymbol{A}^{-1}\mathrm{AMG}(\boldsymbol{S}_1)$. In Section 3.8.2, we illustrate the robust performance of the preconditioner $\boldsymbol{\mathcal{P}}_0$ applied with MINRES.

In the following, we discuss a second way to develop a preconditioner for the smooth semi-implicit Cahn–Hilliard system (3.88). This can be achieved by applying the steps in [35], which we explain below in the proof of Theorem 3.12. In fact, the procedure is straightforward since our coefficient matrix in (3.88) has only a negligible difference to the coefficient matrix in [35]. More precisely, the authors of [35] consider a system of the form (3.87) (implicit discretization) while we focus on the form (3.88) (semi-implicit discretization) in this section. Hence, the only difference occurs in the matrix $\boldsymbol{H}$ or $\boldsymbol{F}$. But this distinction is negligible for the following approach. More importantly, the proof of the theorem below differs from the one in [35] at a marked point. For the development of a preconditioner, we rewrite (3.90) again and consider

$$\begin{aligned}\begin{bmatrix} \boldsymbol{M} & -\boldsymbol{A} \\ \tau\boldsymbol{K} & \boldsymbol{M} \end{bmatrix}\begin{bmatrix} \boldsymbol{w}^{(k+1)} \\ \boldsymbol{u}^{(k+1)} \end{bmatrix} &:= \begin{bmatrix} \boldsymbol{M} & -\varepsilon\boldsymbol{K} - \varepsilon^{-1}\boldsymbol{H} \\ \tau\boldsymbol{K} & \boldsymbol{M} \end{bmatrix}\begin{bmatrix} \boldsymbol{w}^{(k+1)} \\ \boldsymbol{u}^{(k+1)} \end{bmatrix} \\ &= \begin{bmatrix} -2\varepsilon^{-1}\boldsymbol{M}\left(\boldsymbol{u}^{(k)}\right)^3 - \varepsilon^{-1}\boldsymbol{M}\boldsymbol{u}^{\mathrm{old}} \\ \boldsymbol{M}\boldsymbol{u}^{\mathrm{old}} \end{bmatrix}.\end{aligned} \tag{3.94}$$

In the following, we denote this coefficient matrix by $\boldsymbol{\mathcal{A}}$. Since we obtained $\boldsymbol{\mathcal{A}}$ from (3.90) by interchanging m columns and multiplying m rows by -1, its determinant does not change. Hence, $\boldsymbol{\mathcal{A}}$ remains nonsingular. Note that $\boldsymbol{\mathcal{A}}$ is not symmetric anymore as it was in the previous case. Hence, nonsymmetric Krylov subspace solvers have to be used. Due to (3.89) together with Proposition 3.10, the diagonal entries of $\boldsymbol{H}$ lie in the interval $[0,\, Ch^d\alpha]$. Hence, the estimated order for the diagonal entries in $\varepsilon^{-1}\boldsymbol{H}$ lie in the interval $[0,\, 3C\varepsilon^{d-1}\|\boldsymbol{u}^{(k)}\|_\infty^2]$, since h is of order ε. Hence, as in [35], we suggest to neglect the block $\boldsymbol{H}$ in $\boldsymbol{A}$. Therefore, we approximate $\mathcal{A}$ as

$$\mathcal{A}_0 = \begin{bmatrix} \boldsymbol{M} & -\varepsilon\boldsymbol{K} \\ \tau\boldsymbol{K} & \boldsymbol{M} \end{bmatrix}.$$

In what follows, we discuss the quality of the approximation $\mathcal{A}_0$. We denote the Schur complement of $\mathcal{A}_0$ by $\tilde{\boldsymbol{S}} = \boldsymbol{M} + \tau\varepsilon\boldsymbol{K}\boldsymbol{M}^{-1}\boldsymbol{K}$. Note that both, the $(1,1)$ and $(2,2)$ block of $\mathcal{A}_0$, are nonsingular. In particular, they are symmetric positive definite. Consider the generalized eigenvalue problem

$$\mathcal{A}\begin{bmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_2 \end{bmatrix} = \lambda\mathcal{A}_0\begin{bmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_2 \end{bmatrix}. \tag{3.95}$$

Theorem 3.12. *It holds*

$$\sigma(\mathcal{A}_0^{-1}\mathcal{A}) \subset B_\varsigma(1),$$

where $B_\varsigma(1)$ is a circle in the complex plane around one with radius ς. The radius is bounded by $\varsigma \leq \frac{3}{2}\sqrt{\frac{\tau}{\varepsilon^3}}\|\boldsymbol{u}^{(k)}\|_\infty^2$. In particular, m eigenvalues are equal to one. We get $\varsigma \leq 0.5$ when $\tau \leq \varepsilon^3/(9\|\boldsymbol{u}^{(k)}\|_\infty^4)$.

Proof. We transform (3.95) to

$$(\mathcal{A} - \mathcal{A}_0)\begin{bmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_2 \end{bmatrix} = \mu\mathcal{A}_0\begin{bmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_2 \end{bmatrix}, \tag{3.96}$$

where $\mu = \lambda - 1$. The inverse of $\mathcal{A}_0$ can be expressed via a combination of (2.23) and (2.24) as

$$\mathcal{A}_0^{-1} = \begin{bmatrix} \tilde{\boldsymbol{S}}^{-1} & \varepsilon\tilde{\boldsymbol{S}}^{-1}\boldsymbol{K}\boldsymbol{M}^{-1} \\ -\tau\tilde{\boldsymbol{S}}^{-1}\boldsymbol{K}\boldsymbol{M}^{-1} & \tilde{\boldsymbol{S}}^{-1} \end{bmatrix}. \tag{3.97}$$

This yields

$$\mathcal{A}_0^{-1}(\mathcal{A} - \mathcal{A}_0) = \begin{bmatrix} \boldsymbol{0} & -\varepsilon^{-1}\tilde{\boldsymbol{S}}^{-1}\boldsymbol{H} \\ \boldsymbol{0} & \tau\varepsilon^{-1}\tilde{\boldsymbol{S}}^{-1}\boldsymbol{K}\boldsymbol{M}^{-1}\boldsymbol{H} \end{bmatrix}. \tag{3.98}$$

Hence, (3.96) has m zero eigenvalues corresponding to eigenvectors $[\boldsymbol{q}_1^T, \boldsymbol{0}^T]^T$. Thus, (3.95) has m one eigenvalues. Next, we write (3.96) out as

$$-\varepsilon^{-1}\boldsymbol{H}\boldsymbol{q}_2 = \mu(\boldsymbol{M}\boldsymbol{q}_1 - \varepsilon\boldsymbol{K}\boldsymbol{q}_2), \tag{3.99}$$

$$\boldsymbol{0} = \mu(\tau\boldsymbol{K}\boldsymbol{q}_1 + \boldsymbol{M}\boldsymbol{q}_2). \tag{3.100}$$

We express $\mu\boldsymbol{q}_1$ from (3.99) and substitute it into (3.100)

$$\tau\varepsilon^{-1}\boldsymbol{K}\boldsymbol{M}^{-1}\boldsymbol{H}\boldsymbol{q}_2 = \mu\left(\tau\varepsilon\boldsymbol{K}\boldsymbol{M}^{-1}\boldsymbol{K} + \boldsymbol{M}\right)\boldsymbol{q}_2.$$

Multiplying this equation from the left by $\boldsymbol{M}^{-1}$ yields the following generalized eigenvalue problem

$$\tau\varepsilon^{-1}\boldsymbol{M}^{-1}\boldsymbol{K}\boldsymbol{M}^{-1}\boldsymbol{H}\boldsymbol{q}_2 = \mu\left(\boldsymbol{I} + \tau\varepsilon(\boldsymbol{M}^{-1}\boldsymbol{K})^2\right)\boldsymbol{q}_2.$$

We introduce

$$\boldsymbol{R} := \tau\varepsilon^{-1}\left(\boldsymbol{I} + \tau\varepsilon(\boldsymbol{M}^{-1}\boldsymbol{K})^2\right)^{-1}\boldsymbol{M}^{-1}\boldsymbol{K}\boldsymbol{M}^{-1}\boldsymbol{H}$$

and estimate its eigenvalues in the following. Therefore, we first perform a similarity transformation; see Definition 2.23, on $\boldsymbol{R}$: $\boldsymbol{M}^{\frac{1}{2}}\boldsymbol{R}\boldsymbol{M}^{-\frac{1}{2}} =: \tilde{\boldsymbol{R}}$. Note that $\boldsymbol{R}$ and $\tilde{\boldsymbol{R}}$ have the same eigenvalues. Next, we analyze the eigenvalues of $\tilde{\boldsymbol{R}}$. Therefore, we first reformulate $\tilde{\boldsymbol{R}}$ as

$$\begin{aligned}
\tilde{\boldsymbol{R}} &= \tau\varepsilon^{-1}\boldsymbol{M}^{\frac{1}{2}}\left(\boldsymbol{I} + \tau\varepsilon(\boldsymbol{M}^{-1}\boldsymbol{K})^2\right)^{-1}\boldsymbol{M}^{-1}\boldsymbol{K}\boldsymbol{M}^{-1}\boldsymbol{H}\boldsymbol{M}^{-\frac{1}{2}}\\
&= \tau\varepsilon^{-1}\left[\left(\boldsymbol{I} + \tau\varepsilon(\boldsymbol{M}^{-1}\boldsymbol{K})^2\right)\boldsymbol{M}^{-\frac{1}{2}}\right]^{-1}\boldsymbol{M}^{-1}\boldsymbol{K}\boldsymbol{M}^{-1}\boldsymbol{H}\boldsymbol{M}^{-\frac{1}{2}}\\
&= \tau\varepsilon^{-1}\left(\boldsymbol{M}^{-\frac{1}{2}} + \tau\varepsilon(\boldsymbol{M}^{-1}\boldsymbol{K})^2\boldsymbol{M}^{-\frac{1}{2}}\right)^{-1}\boldsymbol{M}^{-1}\boldsymbol{K}\boldsymbol{M}^{-1}\boldsymbol{H}\boldsymbol{M}^{-\frac{1}{2}}\\
&= \tau\varepsilon^{-1}\left[\boldsymbol{M}^{-\frac{1}{2}}\left(\boldsymbol{I} + \tau\varepsilon\boldsymbol{M}^{-\frac{1}{2}}\boldsymbol{K}\boldsymbol{M}^{-1}\boldsymbol{K}\boldsymbol{M}^{-\frac{1}{2}}\right)\right]^{-1}\boldsymbol{M}^{-1}\boldsymbol{K}\boldsymbol{M}^{-1}\boldsymbol{H}\boldsymbol{M}^{-\frac{1}{2}}\\
&= \tau\varepsilon^{-1}\left(\boldsymbol{I} + \tau\varepsilon\boldsymbol{M}^{-\frac{1}{2}}\boldsymbol{K}\boldsymbol{M}^{-1}\boldsymbol{K}\boldsymbol{M}^{-\frac{1}{2}}\right)^{-1}\boldsymbol{M}^{-\frac{1}{2}}\boldsymbol{K}\boldsymbol{M}^{-1}\boldsymbol{H}\boldsymbol{M}^{-\frac{1}{2}}\\
&= \tau\varepsilon^{-1}\left(\boldsymbol{I} + \tau\varepsilon\tilde{\boldsymbol{K}}^2\right)^{-1}\tilde{\boldsymbol{K}}\tilde{\boldsymbol{H}},
\end{aligned} \tag{3.101}$$

where $\tilde{\boldsymbol{K}} = \boldsymbol{M}^{-\frac{1}{2}}\boldsymbol{K}\boldsymbol{M}^{-\frac{1}{2}}$ and $\tilde{\boldsymbol{H}} = \boldsymbol{M}^{-\frac{1}{2}}\boldsymbol{H}\boldsymbol{M}^{-\frac{1}{2}}$. Note that $\tilde{\boldsymbol{K}}$ is symmetric positive semidefinite. From now on, this proof differs from the one in [35]. Due to Theorem 2.27 (symmetric Schur decomposition), we can write $\tilde{\boldsymbol{K}} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T$, where $\boldsymbol{Q} = [\boldsymbol{q}_1|\ldots|\boldsymbol{q}_m] \in \mathbb{R}^{m\times m}$ is orthogonal and $\boldsymbol{\Lambda} = \operatorname{diag}(\lambda_1,\ldots,\lambda_m)$ such that $\tilde{\boldsymbol{K}}\boldsymbol{q}_j = \lambda_j\boldsymbol{q}_j$ for $j = 1,\ldots,m$. Hence, $\tilde{\boldsymbol{K}}^2 = \tilde{\boldsymbol{Q}}\boldsymbol{\Lambda}^2\boldsymbol{Q}^T$. Using this Schur decomposition, we can rewrite $\left(\boldsymbol{I} + \tau\varepsilon\tilde{\boldsymbol{K}}^2\right)^{-1}\tilde{\boldsymbol{K}}$ in (3.101) further as

$$\begin{aligned}
\left(\boldsymbol{I} + \tau\varepsilon\tilde{\boldsymbol{K}}^2\right)^{-1}\tilde{\boldsymbol{K}} &= \left(\boldsymbol{Q}\boldsymbol{Q}^T + \tau\varepsilon\boldsymbol{Q}\boldsymbol{\Lambda}^2\boldsymbol{Q}^T\right)^{-1}\boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T = \left[\boldsymbol{Q}\left(\boldsymbol{Q}^T + \tau\varepsilon\boldsymbol{\Lambda}^2\boldsymbol{Q}^T\right)\right]^{-1}\boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T\\
&= \left(\boldsymbol{Q}^T + \tau\varepsilon\boldsymbol{\Lambda}^2\boldsymbol{Q}^T\right)^{-1}\boldsymbol{\Lambda}\boldsymbol{Q}^T = \left[\left(\boldsymbol{I} + \tau\varepsilon\boldsymbol{\Lambda}^2\right)\boldsymbol{Q}^T\right]^{-1}\boldsymbol{\Lambda}\boldsymbol{Q}^T\\
&= \boldsymbol{Q}\left(\boldsymbol{I} + \tau\varepsilon\boldsymbol{\Lambda}^2\right)^{-1}\boldsymbol{\Lambda}\boldsymbol{Q}^T,
\end{aligned}$$

where $\left(\boldsymbol{I} + \tau\varepsilon\boldsymbol{\Lambda}^2\right)^{-1}\boldsymbol{\Lambda}$ is a diagonal matrix. Hence, $\left(\boldsymbol{I} + \tau\varepsilon\tilde{\boldsymbol{K}}^2\right)^{-1}\tilde{\boldsymbol{K}}$ is symmetric. It follows that

$$\left(\boldsymbol{I} + \tau\varepsilon\tilde{\boldsymbol{K}}^2\right)^{-1}\tilde{\boldsymbol{K}}\boldsymbol{q}_j = \frac{\lambda_j}{1 + \lambda_j^2\tau\varepsilon}\boldsymbol{q}_j \tag{3.102}$$

for $j = 1,\ldots,m$. Using the inequality

$$0 \leq (1 - ab)^2 = 1 + a^2b^2 - 2ab$$

with $a, b \in \mathbb{R}$, we can bound the eigenvalues of (3.102) as

$$\frac{\lambda_j}{1 + \lambda_j^2\tau\varepsilon} \leq \frac{\lambda_j}{2\lambda_j\sqrt{\tau\varepsilon}} = \frac{1}{2\sqrt{\tau\varepsilon}}$$

for $j = 1,\ldots,m$. Here, we have used $a^2 = \tau\varepsilon$ and $b^2 = \lambda_j^2$. This yields

$$\rho\left(\left(\boldsymbol{I} + \tau\varepsilon\tilde{\boldsymbol{K}}^2\right)^{-1}\tilde{\boldsymbol{K}}\right) \leq \frac{1}{2\sqrt{\tau\varepsilon}}. \tag{3.103}$$

Finally, we can estimate the eigenvalues of $\tilde{\boldsymbol{R}}$. Note that, due to Theorem 2.31, it holds $\rho(\tilde{\boldsymbol{R}}) \leq \|\tilde{\boldsymbol{R}}\|$. Further, we obtain

$$\begin{aligned}\|\tilde{\boldsymbol{R}}\| &\leq \tau\varepsilon^{-1}\|\left(\boldsymbol{I}+\tau\varepsilon\tilde{\boldsymbol{K}}^2\right)^{-1}\tilde{\boldsymbol{K}}\|\,\|\tilde{\boldsymbol{H}}\| = \tau\varepsilon^{-1}\rho\left(\left(\boldsymbol{I}+\tau\varepsilon\tilde{\boldsymbol{K}}^2\right)^{-1}\tilde{\boldsymbol{K}}\right)\rho(\tilde{\boldsymbol{H}})\\ &\leq \frac{\tau}{2\varepsilon\sqrt{\tau\varepsilon}}\rho(\tilde{\boldsymbol{H}}),\end{aligned} \tag{3.104}$$

where the equality holds due to the symmetry of $\left(\boldsymbol{I}+\tau\varepsilon\tilde{\boldsymbol{K}}^2\right)^{-1}\tilde{\boldsymbol{K}}$ and $\tilde{\boldsymbol{H}}$. Moreover, due to the diagonal structure of $\boldsymbol{H}$, we have

$$\tilde{\boldsymbol{H}} = \boldsymbol{M}^{-\frac{1}{2}}\boldsymbol{H}\boldsymbol{M}^{-\frac{1}{2}} = \operatorname{diag}\left(3\left(u_{h,i}^{(k)}\right)^2\right)_{i=1,\dots,m.}$$

Due to (3.89), the diagonal entries of $\tilde{\boldsymbol{H}}$ lie in the interval $[0,\alpha]$, where $\alpha = 3\|\boldsymbol{u}^{(k)}\|_\infty^2$. Thus, we obtain in (3.104)

$$\|\tilde{\boldsymbol{R}}\| \leq \frac{\sqrt{\tau}}{2\varepsilon\sqrt{\varepsilon}}\alpha = \frac{3}{2}\sqrt{\frac{\tau}{\varepsilon^3}}\|\boldsymbol{u}^{(k)}\|_\infty^2. \tag{3.105}$$

Therefore, for $\tau < \varepsilon^3/(9\|\boldsymbol{u}^{(k)}\|_\infty^4)$, it holds $\sigma(\tilde{\boldsymbol{R}}) = \sigma(\boldsymbol{R}) \subset B_{0.5}(0)$ and hence $\sigma(\mathcal{A}_0^{-1}\mathcal{A}) \subset B_{0.5}(1)$. □

Remark 3.3. *Note that the time step condition in Theorem 3.12 complies with the one in Theorem 3.1 and 3.2 for the implicit time-discrete scheme. Due to accuracy requirements, small time steps are needed in the semi-implicit scheme too; see Section 3.8.4.*

After we have proven that $\mathcal{A}_0$ is a reasonable approximation of $\mathcal{A}$, we can go over to the construction of a suitable preconditioner $\boldsymbol{\mathcal{P}}$ for $\mathcal{A}_0$ and hence for $\mathcal{A}$. We propose the block triangular preconditioner

$$\boldsymbol{\mathcal{P}} = \begin{bmatrix} \boldsymbol{M} & \boldsymbol{0} \\ \tau\boldsymbol{K} & -\hat{\boldsymbol{S}} \end{bmatrix}.$$

As Schur complement approximation, we design $\hat{\boldsymbol{S}}$ as

$$\begin{aligned}\hat{\boldsymbol{S}} &= \boldsymbol{S}_1\boldsymbol{M}^{-1}\boldsymbol{S}_1\\ &= \left(\boldsymbol{M}+\sqrt{\tau\varepsilon}\boldsymbol{K}\right)\boldsymbol{M}^{-1}\left(\boldsymbol{M}+\sqrt{\tau\varepsilon}\boldsymbol{K}\right)\\ &= \boldsymbol{M}+\tau\varepsilon\boldsymbol{K}\boldsymbol{M}^{-1}\boldsymbol{K}+2\sqrt{\tau\varepsilon}\boldsymbol{K}.\end{aligned} \tag{3.106}$$

The first two terms in (3.106) match the exact Schur complement $\tilde{\boldsymbol{S}} = \boldsymbol{M}+\tau\varepsilon\boldsymbol{K}\boldsymbol{M}^{-1}\boldsymbol{K}$ of $\mathcal{A}_0$. The influence of the last term in (3.106) is reduced due to the factor $\sqrt{\tau\varepsilon}$. In fact, this approximation turns out to be an optimal Schur complement preconditioner for $\mathcal{A}_0$ (see also [130]):

Lemma 3.13. *It holds*

$$\sigma\left(\hat{\boldsymbol{S}}^{-1}\tilde{\boldsymbol{S}}\right) \subset [0.5, 1].$$

Proof. Due to Proposition 3.10, $\tilde{\boldsymbol{S}}$ and $\hat{\boldsymbol{S}}$ are symmetric positive definite. Hence, we may prove the result using the Rayleigh quotient argument in Theorem 2.29. We write

$$\frac{\boldsymbol{v}^T\tilde{\boldsymbol{S}}\boldsymbol{v}}{\boldsymbol{v}^T\hat{\boldsymbol{S}}\boldsymbol{v}} = \frac{\boldsymbol{v}^T\left(\boldsymbol{M}+\tau\varepsilon\boldsymbol{K}\boldsymbol{M}^{-1}\boldsymbol{K}\right)\boldsymbol{v}}{\boldsymbol{v}^T\left(\boldsymbol{M}+\tau\varepsilon\boldsymbol{K}\boldsymbol{M}^{-1}\boldsymbol{K}+2\sqrt{\tau\varepsilon}\boldsymbol{K}\right)\boldsymbol{v}} = \frac{\boldsymbol{a}^T\boldsymbol{a}+\boldsymbol{b}^T\boldsymbol{b}}{\boldsymbol{a}^T\boldsymbol{a}+\boldsymbol{b}^T\boldsymbol{b}+2\boldsymbol{a}^T\boldsymbol{b}},$$

where $\boldsymbol{a} = \boldsymbol{M}^{\frac{1}{2}}\boldsymbol{v}$ and $\boldsymbol{b} = \sqrt{\tau\varepsilon}\boldsymbol{M}^{-\frac{1}{2}}\boldsymbol{K}\boldsymbol{v}$. From the properties of $\boldsymbol{M}$ and $\boldsymbol{K}$, it follows $\boldsymbol{a}^T\boldsymbol{a} > 0$ and $\boldsymbol{b}^T\boldsymbol{b}, \boldsymbol{a}^T\boldsymbol{b} \geq 0$ and therefore $\frac{\boldsymbol{v}^T\tilde{\boldsymbol{S}}\boldsymbol{v}}{\boldsymbol{v}^T\hat{\boldsymbol{S}}\boldsymbol{v}} \leq 1$. On the other hand, $(\boldsymbol{a}-\boldsymbol{b})^T(\boldsymbol{a}-\boldsymbol{b}) \geq 0$, which gives $\frac{\boldsymbol{v}^T\tilde{\boldsymbol{S}}\boldsymbol{v}}{\boldsymbol{v}^T\hat{\boldsymbol{S}}\boldsymbol{v}} \geq 0.5$. □

Let us conclude the preconditioner $\boldsymbol{\mathcal{P}}$ with a statement about its practical realization. The action of the inverse of $\boldsymbol{S}_1$ is performed with an AMG since $\boldsymbol{S}_1$ forms the discretization of an elliptic operator. The $(1,1)$ block $\boldsymbol{M}$ is a diagonal matrix with positive entries. Hence, its inverse can be performed by elementwise multiplications.[3] Hence, the practical block diagonal preconditioner is given by

$$\boldsymbol{\mathcal{P}}_0 = \begin{bmatrix} \boldsymbol{M} & \boldsymbol{0} \\ \tau\boldsymbol{K} & -\boldsymbol{S}_0 \end{bmatrix},$$

where $\boldsymbol{S}_0 = \mathrm{AMG}(\boldsymbol{S}_1)\boldsymbol{A}^{-1}\mathrm{AMG}(\boldsymbol{S}_1)$.

Since the above theoretical analysis proves the optimality of the preconditioner $\boldsymbol{\mathcal{P}}$, we will not study the numerical robustness. We will see in the next section that we can apply the above procedure, i.e., approximating the coefficient matrix by $\boldsymbol{\mathcal{A}}_0$ and applying the optimal block triangular preconditioner $\boldsymbol{\mathcal{P}}_0$, to the smooth implicit system as well.

3.7.2 Smooth implicit systems

The difference to the system from the last section occurs in the $(1,1)$ block of (3.87). In particular, the matrix $\boldsymbol{F}$ is in general indefinite: Due to (3.89) together with Proposition 3.10, the diagonal entries of $\boldsymbol{F}$ lie in the interval $[-Ch^d, Ch^d(\alpha-1)]$. This implies that $\boldsymbol{A} = \varepsilon\boldsymbol{K} + \varepsilon^{-1}\boldsymbol{F}$ can be easily indefinite. In this case, $\boldsymbol{A}$ resembles a discrete Helmholtz operator. In [64], it is described how difficult it is to solve Helmholtz problems with classical iterative methods. For this reason, and also because of the demand of a positive definite preconditioner for symmetric Krylov subspace methods, we switch to the nonsymmetric system matrix as done in the second part of the last section. We rewrite (3.87) and consider

$$\begin{aligned}\begin{bmatrix} \boldsymbol{M} & -\boldsymbol{A} \\ \tau\boldsymbol{K} & \boldsymbol{M} \end{bmatrix}\begin{bmatrix} \boldsymbol{w}^{(k+1)} \\ \boldsymbol{u}^{(k+1)} \end{bmatrix} &:= \begin{bmatrix} \boldsymbol{M} & -\varepsilon\boldsymbol{K}-\varepsilon^{-1}\boldsymbol{F} \\ \tau\boldsymbol{K} & \boldsymbol{M} \end{bmatrix}\begin{bmatrix} \boldsymbol{w}^{(k+1)} \\ \boldsymbol{u}^{(k+1)} \end{bmatrix} \\ &= \begin{bmatrix} -2\varepsilon^{-1}\boldsymbol{M}\left(\boldsymbol{u}^{(k)}\right)^3 \\ \boldsymbol{M}\boldsymbol{u}^{\mathrm{old}} \end{bmatrix}.\end{aligned} \tag{3.107}$$

In the following, we denote the coefficient matrix by $\boldsymbol{\mathcal{A}}$. The estimated order for the diagonal entries in $\varepsilon^{-1}\boldsymbol{F}$ lie in the interval $[-C\varepsilon^{d-1}, C\varepsilon^{d-1}(3\|\boldsymbol{u}^{(k)}\|_\infty^2 - 1)]$, since h

[3] For consistent mass matrices, the Chebyshev semi-iteration [81, 82] provides a powerful preconditioner [150, 132].

is of order ε. Hence, as in the last section, we suggest to neglect the block $\boldsymbol{F}$ in $\boldsymbol{A}$. Therefore, we approximate $\mathcal{A}$ as

$$\mathcal{A}_0 = \begin{bmatrix} \boldsymbol{M} & -\varepsilon\boldsymbol{K} \\ \tau\boldsymbol{K} & \boldsymbol{M} \end{bmatrix}.$$

We denote the Schur complement of $\mathcal{A}_0$ by $\tilde{\boldsymbol{S}} = \boldsymbol{M} + \tau\varepsilon\boldsymbol{K}\boldsymbol{M}^{-1}\boldsymbol{K}$. As in the semi-implicit case, we can prove:

Theorem 3.14. *It holds*

$$\sigma(\mathcal{A}_0^{-1}\mathcal{A}) \subset B_\varsigma(1).$$

The circle radius is bounded by $\varsigma \le \frac{1}{2}\sqrt{\frac{\tau}{\varepsilon^3}}\max\left\{1, |3\|\boldsymbol{u}^{(k)}\|_\infty^2 - 1|\right\}$. *In particular, m eigenvalues are equal to one. We get* $\varsigma \le 0.5$ *when* $\tau \le \varepsilon^3/\max^2\left\{1, |3\|\boldsymbol{u}^{(k)}\|_\infty^2 - 1|\right\}$.

Proof. The proof is the same as the one for Theorem 3.12. The only negligible difference appears at the end in (3.104), where we have to replace $\tilde{\boldsymbol{H}}$ by $\tilde{\boldsymbol{F}} = \boldsymbol{M}^{-\frac{1}{2}}\boldsymbol{F}\boldsymbol{M}^{-\frac{1}{2}}$. Due to the diagonal structure of $\boldsymbol{F}$, we have

$$\tilde{\boldsymbol{F}} = \operatorname{diag}\left(3\left(u_{h,i}^{(k)}\right)^2 - 1\right)_{i=1,\dots,m.}$$

Due to (3.89), the diagonal entries of $\tilde{\boldsymbol{F}}$ lie in the interval $[-1, \alpha - 1]$, where $\alpha = 3\|\boldsymbol{u}^{(k)}\|_\infty^2$. Thus, we obtain in (3.104)

$$\|\tilde{\boldsymbol{R}}\| \le \frac{\sqrt{\tau}}{2\varepsilon\sqrt{\varepsilon}}\max\{1, |\alpha - 1|\} = \frac{1}{2}\sqrt{\frac{\tau}{\varepsilon^3}}\max\left\{1, |3\|\boldsymbol{u}^{(k)}\|_\infty^2 - 1|\right\}.$$

Therefore, for $\tau \le \varepsilon^3/\max^2\left\{1, |3\|\boldsymbol{u}^{(k)}\|_\infty^2 - 1|\right\}$, it holds $\sigma(\tilde{\boldsymbol{R}}) = \sigma(\boldsymbol{R}) \subset B_{0.5}(0)$ and hence $\sigma(\mathcal{A}_0^{-1}\mathcal{A}) \subset B_{0.5}(1)$. □

Remark 3.4. *Note that the time step condition in Theorem 3.14 complies with the one in Theorem 3.1 and 3.2.*

The proposed block triangular preconditioner

$$\mathcal{P} = \begin{bmatrix} \boldsymbol{M} & \boldsymbol{0} \\ \tau\boldsymbol{K} & -\hat{\boldsymbol{S}} \end{bmatrix},$$

where

$$\hat{\boldsymbol{S}} = \left(\boldsymbol{M} + \sqrt{\tau\varepsilon}\boldsymbol{K}\right)\boldsymbol{M}^{-1}\left(\boldsymbol{M} + \sqrt{\tau\varepsilon}\boldsymbol{K}\right),$$

is exactly the same as before in the semi-implicit case. In fact, $\sigma\left(\hat{\boldsymbol{S}}^{-1}\tilde{\boldsymbol{S}}\right) \subset [0.5, 1]$ as proved in Lemma 3.13. Again, since the above theoretical analysis proves the optimality of the preconditioner $\mathcal{P}$, we will not study the numerical robustness.

Here, we finish the discussion about preconditioning of smooth Cahn–Hilliard systems. Next, we come to the harder case of nonsmooth systems. We will see that a simplification of the coefficient matrix in form of $\mathcal{A}_0$ is not satisfying anymore. We start with the nonsmooth semi-implicit Cahn–Hilliard system (3.86).

3.7.3 Nonsmooth semi-implicit systems

Consider the matrix system in (3.86) with the coefficient matrix

$$\mathcal{A} = \begin{bmatrix} -\varepsilon \boldsymbol{K} - c^{-1}\boldsymbol{G} & \boldsymbol{M} \\ \boldsymbol{M} & \tau \boldsymbol{K} \end{bmatrix} =: \begin{bmatrix} -\boldsymbol{A} & \boldsymbol{M} \\ \boldsymbol{M} & \tau \boldsymbol{K} \end{bmatrix}. \tag{3.108}$$

It can be easily seen that $\mathcal{A}$ is symmetric. The $(1,1)$ block $\boldsymbol{A}$ is symmetric positive semidefinite. In particular, $\boldsymbol{A}$ is symmetric positive definite when $\boldsymbol{G} \not\equiv \boldsymbol{0}$. This can be seen by letting $\boldsymbol{0} \neq \boldsymbol{v} \in \mathbb{R}^m$ and considering the following two cases:

- $\boldsymbol{v} = \boldsymbol{1} \;\Rightarrow\; \boldsymbol{v}^T\boldsymbol{G}\boldsymbol{v} = 0$ if and only if $\boldsymbol{G} \equiv \boldsymbol{0}$,
- $\boldsymbol{v} \neq \boldsymbol{1} \;\Rightarrow\; \boldsymbol{v}^T\boldsymbol{K}\boldsymbol{v} > 0$, $\boldsymbol{v}^T\boldsymbol{G}\boldsymbol{v} \geq 0$.

This means, only for the very first few time steps (as long as no pure phases are present), we have $|u_{h,i}^{(k)}| \leq 1$ for $i = 1, \ldots, m$, and hence $\boldsymbol{G} \equiv \boldsymbol{0}$. As soon as active sets (pure phases) have formed, i.e., $|u_{h,i}^{(k)}| > 1$ for some i, $\boldsymbol{G} \not\equiv \boldsymbol{0}$ and $\boldsymbol{A}$ remains positive definite. Let us have a closer look at the matrix $c^{-1}\boldsymbol{G}$ and assume that $\boldsymbol{G} \not\equiv \boldsymbol{0}$. Then, penalized entries, i.e., the nonzero entries, are in general scattered throughout its diagonal. The intensity of the penalization can be controlled by the penalty parameter c. The smaller c is the stronger is the penalization and the more accurate is the numerical approximation of the nonsmoothness. In particular, the penalized entries of $c^{-1}\boldsymbol{G}$ lie in the interval $c^{-1}h^d[\tilde{c}, C]$. The nonpenalized entries of $c^{-1}\boldsymbol{G}$ are equal to zero. Under the assumption $\boldsymbol{G} \not\equiv \boldsymbol{0}$, the Schur complement of $\mathcal{A}$ is $\boldsymbol{S} = \tau\boldsymbol{K} + \boldsymbol{M}\boldsymbol{A}^{-1}\boldsymbol{M}$. A symmetric Schur complement preconditioner of the form

$$\begin{aligned} &\left(\boldsymbol{M} + \sqrt{\tau\varepsilon}\boldsymbol{K}\right)\boldsymbol{A}^{-1}\left(\boldsymbol{M} + \sqrt{\tau\varepsilon}\boldsymbol{K}\right) \\ &\qquad = \tau\varepsilon\boldsymbol{K}\boldsymbol{A}^{-1}\boldsymbol{K} + \boldsymbol{M}\boldsymbol{A}^{-1}\boldsymbol{M} + \sqrt{\tau\varepsilon}\boldsymbol{M}\boldsymbol{A}^{-1}\boldsymbol{K} + \sqrt{\tau\varepsilon}\boldsymbol{K}\boldsymbol{A}^{-1}\boldsymbol{M}, \end{aligned}$$

as done in (3.92), would only be satisfying if $c \geq \varepsilon^{d-1}$ since

$$\tau\varepsilon\boldsymbol{K}\boldsymbol{A}^{-1} = \tau\boldsymbol{K}\left(\boldsymbol{K} + c^{-1}\varepsilon^{-1}\boldsymbol{G}\right)^{-1}$$

and the estimated order for the diagonal entries in $c^{-1}\varepsilon^{-1}\boldsymbol{G}$ lie in the interval $[0, Cc^{-1}\varepsilon^{d-1}]$, where we have used that h is of order ε. Note that sufficient sizes of c are $c \leq 10^{-4}$, and in our numerical examples we usually work with $c = 10^{-7}$. Moreover, we have in mind that we want to go over to adaptive mesh strategies, where we coarsen the mesh inside the penalized regions. This excludes a symmetric preconditioner $\mathcal{P}$ for $\mathcal{A}$. Moreover, neglecting the block $\boldsymbol{G}$, as done before with $\boldsymbol{H}$ and $\boldsymbol{F}$, would give a worse approximation for small penalization parameters, which is summarized as follows:

Theorem 3.15. *Let*

$$\mathcal{A} = \begin{bmatrix} \boldsymbol{M} & -\varepsilon\boldsymbol{K} - c^{-1}\boldsymbol{G} \\ \tau\boldsymbol{K} & \boldsymbol{M} \end{bmatrix} =: \begin{bmatrix} \boldsymbol{M} & -\boldsymbol{A} \\ \tau\boldsymbol{K} & \boldsymbol{M} \end{bmatrix} \quad \text{and} \quad \mathcal{A}_0 = \begin{bmatrix} \boldsymbol{M} & -\varepsilon\boldsymbol{K} \\ \tau\boldsymbol{K} & \boldsymbol{M} \end{bmatrix}.$$

It holds

$$\sigma(\mathcal{A}_0^{-1}\mathcal{A}) \subset B_\varsigma(1).$$

The circle radius is bounded by $\varsigma \leq \frac{\sqrt{\tau}}{2c\sqrt{\varepsilon}}$. In particular, m eigenvalues are equal to one. We get $\varsigma \leq 0.5$ when $\tau \leq \varepsilon c^2$.

Proof. The proof is the same as the one for Theorem 3.12. The crucial difference appears at the end in (3.104), where we obtain

$$\|\tilde{\boldsymbol{R}}\| \leq \tau c^{-1} \| \left(\boldsymbol{I} + \tau\varepsilon \tilde{\boldsymbol{K}}^2\right)^{-1} \tilde{\boldsymbol{K}} \| \, \|\tilde{\boldsymbol{G}}\| \leq \frac{\tau}{2c\sqrt{\tau\varepsilon}} \rho(\tilde{\boldsymbol{G}}) \tag{3.109}$$

with $\tilde{\boldsymbol{G}} = \boldsymbol{M}^{-\frac{1}{2}} \boldsymbol{G} \boldsymbol{M}^{-\frac{1}{2}}$. Due to the diagonal structure of $\boldsymbol{G}$, we have

$$\tilde{\boldsymbol{G}} = \operatorname{diag}\left(\begin{array}{ll} 1 & \text{if } |u_{h,i}^{(k)}| > 1, \\ 0 & \text{otherwise.} \end{array} \right)_{i=1,\ldots,m} \in \mathbb{R}^{m \times m}$$

Thus, we obtain in (3.109)

$$\|\tilde{\boldsymbol{R}}\| \leq \frac{\sqrt{\tau}}{2c\sqrt{\varepsilon}}.$$

Therefore, for $\tau \leq \varepsilon c^2$, it holds $\sigma(\tilde{\boldsymbol{R}}) = \sigma(\boldsymbol{R}) \subset B_{0.5}(0)$ and hence $\sigma(\boldsymbol{\mathcal{A}}_0^{-1}\boldsymbol{\mathcal{A}}) \subset B_{0.5}(1)$. □

Hence, neglecting the block $\boldsymbol{G}$ would only be satisfying for tiny time step sizes τ, which is far away from being practical.

In the following, we concentrate on the coefficient matrix in (3.108). The first block triangular preconditioner we propose is

$$\boldsymbol{\mathcal{P}} = \begin{bmatrix} -\hat{\boldsymbol{A}} & \boldsymbol{0} \\ \boldsymbol{M} & -\hat{\boldsymbol{S}} \end{bmatrix}. \tag{3.110}$$

As mentioned, $\boldsymbol{A}$ is only symmetric positive semidefinite if $\boldsymbol{G} \equiv \boldsymbol{0}$. Hence, we suggest

$$\hat{\boldsymbol{A}} = \begin{cases} \boldsymbol{A} + \varepsilon\tau\boldsymbol{I} & \text{if } \boldsymbol{G} \equiv \boldsymbol{0}, \\ \boldsymbol{A} & \text{otherwise,} \end{cases} \tag{3.111}$$

where the shift $\varepsilon\tau$ is an heuristic choice. As Schur complement approximation, we design $\hat{\boldsymbol{S}}$ as

$$\begin{aligned} \hat{\boldsymbol{S}} &= \boldsymbol{S}_1 \hat{\boldsymbol{A}}^{-1} \boldsymbol{S}_2 \\ &= \left(\boldsymbol{M} + \sqrt{\tau}\boldsymbol{K}\right) \hat{\boldsymbol{A}}^{-1} \left(\boldsymbol{M} + \sqrt{\tau}\hat{\boldsymbol{A}}\right) \qquad (3.112) \\ &= \tau\boldsymbol{K} + \boldsymbol{M}\hat{\boldsymbol{A}}^{-1}\boldsymbol{M} + \sqrt{\tau}\boldsymbol{M} + \sqrt{\tau}\boldsymbol{K}\hat{\boldsymbol{A}}^{-1}\boldsymbol{M}. \qquad (3.113) \end{aligned}$$

The first term in (3.113) matches the first term in the exact Schur complement $\boldsymbol{S} = \tau\boldsymbol{K} + \boldsymbol{M}\boldsymbol{A}^{-1}\boldsymbol{M}$. The second term in (3.113) approximates the second term in the exact Schur complement. Due to the balanced distribution of τ in form of $\sqrt{\tau}$ in both factors $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$, the influence of both remainder terms in (3.113) is reduced. To illustrate the performance of $\hat{\boldsymbol{S}}^{-1}\boldsymbol{S}$, we show eigenvalue plots in Section 3.8.1. Let us conclude the preconditioner $\boldsymbol{\mathcal{P}}$ with a statement about its practical realization. The action of the inverse of $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ is performed with an AMG each since both form the discretization of an elliptic operator. The same holds for the $(1,1)$ block $\hat{\boldsymbol{A}}$. Hence, the practical block triangular preconditioner is given by

$$\boldsymbol{\mathcal{P}}_0 = \begin{bmatrix} -\boldsymbol{A}_0 & \boldsymbol{0} \\ \boldsymbol{M} & -\boldsymbol{S}_0 \end{bmatrix},$$

where $\boldsymbol{A}_0 = \mathrm{AMG}(\hat{\boldsymbol{A}})$ and $\boldsymbol{S}_0 = \mathrm{AMG}(\boldsymbol{S}_1)\hat{\boldsymbol{A}}^{-1}\mathrm{AMG}(\boldsymbol{S}_2)$. In Section 3.8.2, we illustrate the robust performance of the preconditioner $\boldsymbol{\mathcal{P}}_0$ applied with BiCG.

In the following, we discuss a second way to develop a preconditioner for the nonsmooth semi-implicit Cahn–Hilliard system (3.86). We can avoid the case analysis in (3.111), which is done to make the block $\boldsymbol{A}$ symmetric positive definite. By interchanging the column blocks in (3.86), we obtain

$$\begin{aligned}\begin{bmatrix} \boldsymbol{M} & -\boldsymbol{A} \\ \tau\boldsymbol{K} & \boldsymbol{M} \end{bmatrix}\begin{bmatrix} \boldsymbol{w}^{(k+1)} \\ \boldsymbol{u}^{(k+1)} \end{bmatrix} &:= \begin{bmatrix} \boldsymbol{M} & -\varepsilon\boldsymbol{K} - c^{-1}\boldsymbol{G} \\ \tau\boldsymbol{K} & \boldsymbol{M} \end{bmatrix}\begin{bmatrix} \boldsymbol{w}^{(k+1)} \\ \boldsymbol{u}^{(k+1)} \end{bmatrix} \\ &= \begin{bmatrix} -c^{-1}\left(\boldsymbol{G}_+\boldsymbol{1} - \boldsymbol{G}_-\boldsymbol{1}\right) - \varepsilon^{-1}\boldsymbol{M}\boldsymbol{u}^{\mathrm{old}} \\ \boldsymbol{M}\boldsymbol{u}^{\mathrm{old}} \end{bmatrix}.\end{aligned} \tag{3.114}$$

In the following, we denote the coefficient matrix by $\mathcal{A}$. The Schur complement is now $\boldsymbol{S} = \boldsymbol{M} + \tau\boldsymbol{K}\boldsymbol{M}^{-1}\boldsymbol{A}$. It can be easily seen that $\mathcal{A}$ is not symmetric anymore. However, the preconditioner above has already been built based on a nonsymmetric Schur complement approximation, which results in the use of nonsymmetric Krylov subspace solvers anyway. Hence, without thought, we can work with the nonsymmetric system in (3.114). This has the advantage that the $(1,1)$ block is now diagonal and symmetric positive definite and hence cheap to invert. The block triangular preconditioner we propose is

$$\boldsymbol{\mathcal{P}} = \begin{bmatrix} \boldsymbol{M} & \boldsymbol{0} \\ \tau\boldsymbol{K} & -\hat{\boldsymbol{S}} \end{bmatrix}.$$

As Schur complement approximation we design $\hat{\boldsymbol{S}}$ as

$$\begin{aligned}\hat{\boldsymbol{S}} &= \boldsymbol{S}_1\boldsymbol{M}^{-1}\boldsymbol{S}_2 \\ &= \left(\boldsymbol{M} + \sqrt{\tau}\boldsymbol{K}\right)\boldsymbol{M}^{-1}\left(\boldsymbol{M} + \sqrt{\tau}\boldsymbol{A}\right) && (3.115) \\ &= \boldsymbol{M} + \tau\boldsymbol{K}\boldsymbol{M}^{-1}\boldsymbol{A} + \sqrt{\tau}\boldsymbol{A} + \sqrt{\tau}\boldsymbol{K}. && (3.116)\end{aligned}$$

The first two terms in (3.116) match the exact Schur complement. Due to the balanced distribution of τ in form of $\sqrt{\tau}$ in both factors $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$, the influence of both remainder terms in (3.116) is reduced.

In fact, this is the preconditioner we suggest to solve the nonsmooth implicit system (3.85).

3.7.4 Nonsmooth implicit systems

Consider the matrix system in (3.85). As before, the coefficient matrix is symmetric. However, the $(1,1)$ block can easily become indefinite now. For this reason, and also because of the demand of a positive definite preconditioner for symmetric Krylov subspace methods, we switch to the nonsymmetric system

$$\begin{aligned}\begin{bmatrix} \boldsymbol{M} & -\boldsymbol{A} \\ \tau\boldsymbol{K} & \boldsymbol{M} \end{bmatrix}\begin{bmatrix} \boldsymbol{w}^{(k+1)} \\ \boldsymbol{u}^{(k+1)} \end{bmatrix} &:= \begin{bmatrix} \boldsymbol{M} & -\varepsilon\boldsymbol{K} - c^{-1}\boldsymbol{G} + \varepsilon^{-1}\boldsymbol{M} \\ \tau\boldsymbol{K} & \boldsymbol{M} \end{bmatrix}\begin{bmatrix} \boldsymbol{w}^{(k+1)} \\ \boldsymbol{u}^{(k+1)} \end{bmatrix} \\ &= \begin{bmatrix} -c^{-1}\left(\boldsymbol{G}_+\boldsymbol{1} - \boldsymbol{G}_-\boldsymbol{1}\right) \\ \boldsymbol{M}\boldsymbol{u}^{\mathrm{old}} \end{bmatrix}.\end{aligned} \tag{3.117}$$

In the following, we denote the coefficient matrix by $\mathcal{A}$. The Schur complement is now $\boldsymbol{S} = \boldsymbol{M} + \tau\boldsymbol{K}\boldsymbol{M}^{-1}\boldsymbol{A}$. The outer structure of (3.117) is the same as we have in the second part of the last section. Again, neglecting the blocks $\boldsymbol{G}$ and $\boldsymbol{M}$ in $\boldsymbol{A}$ would give a worse approximation for small penalization parameters, which is summarized as follows:

Theorem 3.16. *Let*

$$\mathcal{A}_0 = \begin{bmatrix} \boldsymbol{M} & -\varepsilon\boldsymbol{K} \\ \tau\boldsymbol{K} & \boldsymbol{M} \end{bmatrix}.$$

It holds

$$\sigma(\mathcal{A}_0^{-1}\mathcal{A}) \subset B_\varsigma(1).$$

The circle radius is bounded by

$$\varsigma \leq \frac{1}{2}\sqrt{\frac{\tau}{\varepsilon^3}}\begin{cases} 1 & \text{if } c > \frac{\varepsilon}{2}, \\ \left(\frac{\varepsilon}{c} - 1\right) & \text{if } c \leq \frac{\varepsilon}{2}. \end{cases}$$

In particular, m eigenvalues are equal to one. We get $\varsigma \leq 0.5$ when

$$\tau \leq \varepsilon^3 \begin{cases} 1 & \text{if } c > \frac{\varepsilon}{2}, \\ \left(\frac{\varepsilon}{c} - 1\right)^{-2} & \text{if } c \leq \frac{\varepsilon}{2}. \end{cases}$$

Proof. The proof is almost the same as the one for Theorem 3.12. The matrix $\boldsymbol{R}$ becomes

$$\boldsymbol{R} = \tau\left(\boldsymbol{I} + \tau\varepsilon(\boldsymbol{M}^{-1}\boldsymbol{K})^2\right)^{-1}\boldsymbol{M}^{-1}\boldsymbol{K}\boldsymbol{M}^{-1}\left(c^{-1}\boldsymbol{G} - \varepsilon^{-1}\boldsymbol{M}\right)$$

and hence $\tilde{\boldsymbol{R}}$ becomes

$$\tilde{\boldsymbol{R}} = \tau\left(\boldsymbol{I} + \tau\varepsilon\tilde{\boldsymbol{K}}^2\right)^{-1}\tilde{\boldsymbol{K}}\left(c^{-1}\tilde{\boldsymbol{G}} - \varepsilon^{-1}\boldsymbol{I}\right),$$

where $\tilde{\boldsymbol{G}} = \boldsymbol{M}^{-\frac{1}{2}}\boldsymbol{G}\boldsymbol{M}^{-\frac{1}{2}}$. We finally end up with

$$\begin{aligned} \|\tilde{\boldsymbol{R}}\| &\leq \tau\left\|\left(\boldsymbol{I} + \tau\varepsilon\tilde{\boldsymbol{K}}^2\right)^{-1}\tilde{\boldsymbol{K}}\right\| \|c^{-1}\tilde{\boldsymbol{G}} - \varepsilon^{-1}\boldsymbol{I}\| = \tau\rho\left(\left(\boldsymbol{I} + \tau\varepsilon\tilde{\boldsymbol{K}}^2\right)^{-1}\tilde{\boldsymbol{K}}\right)\rho(c^{-1}\tilde{\boldsymbol{G}} - \varepsilon^{-1}\boldsymbol{I}) \\ &\leq \frac{\tau}{2\sqrt{\tau\varepsilon}}\rho(c^{-1}\tilde{\boldsymbol{G}} - \varepsilon^{-1}\boldsymbol{I}), \end{aligned}$$

where the equality holds due to the symmetry of $\left(\boldsymbol{I} + \tau\varepsilon\tilde{\boldsymbol{K}}^2\right)^{-1}\tilde{\boldsymbol{K}}$ and $c^{-1}\tilde{\boldsymbol{G}} - \varepsilon^{-1}\boldsymbol{I}$. Moreover, due to the diagonal structure of $\boldsymbol{G}$, we have

$$\tilde{\boldsymbol{G}} = \boldsymbol{M}^{-\frac{1}{2}}\boldsymbol{G}\boldsymbol{M}^{-\frac{1}{2}} = \operatorname{diag}\begin{pmatrix} 1 & \text{if } |u_{h,i}^{(k)}| > 1, \\ 0 & \text{otherwise.} \end{pmatrix}_{i=1,\dots,m}$$

Therefore,

$$c^{-1}\tilde{\boldsymbol{G}} - \varepsilon^{-1}\boldsymbol{I} = \operatorname{diag}\begin{pmatrix} c^{-1} - \varepsilon^{-1} & \text{if } |u_{h,i}^{(k)}| > 1, \\ -\varepsilon^{-1} & \text{otherwise,} \end{pmatrix}_{i=1,\dots,m}$$

which results in

$$\rho(c^{-1}\tilde{\boldsymbol{G}} - \varepsilon^{-1}\boldsymbol{I}) = \max\left\{\varepsilon^{-1}, |c^{-1} - \varepsilon^{-1}|\right\}.$$

Now, we do a case analysis for c. Assume that $|c^{-1} - \varepsilon^{-1}| < \varepsilon^{-1}$, which is equivalent to

$$-\varepsilon^{-1} < c^{-1} - \varepsilon^{-1} < \varepsilon^{-1} \quad \Leftrightarrow \quad 0 < c^{-1} < 2\varepsilon^{-1}.$$

This is the case for small penalizations, i.e., large $c > \varepsilon/2$. We obtain

$$\|\tilde{\boldsymbol{R}}\| \leq \frac{\sqrt{\tau}}{2\varepsilon\sqrt{\varepsilon}} = \frac{1}{2}\sqrt{\frac{\tau}{\varepsilon^3}}.$$

Now, assume that $|c^{-1} - \varepsilon^{-1}| \geq \varepsilon^{-1}$, which is equivalent to

$$c^{-1} - \varepsilon^{-1} \geq \varepsilon^{-1} \quad \Leftrightarrow \quad c^{-1} \geq 2\varepsilon^{-1}.$$

This is the case for large penalizations, i.e., small $c \leq \varepsilon/2$. We obtain

$$\|\tilde{\boldsymbol{R}}\| \leq \frac{\sqrt{\tau}(c^{-1} - \varepsilon^{-1})}{2\sqrt{\varepsilon}} = \frac{1}{2}\sqrt{\frac{\tau}{\varepsilon^3}}\left(\frac{\varepsilon}{c} - 1\right).$$

Therefore, for

$$\tau \leq \varepsilon^3 \begin{cases} 1 & \text{if } c > \frac{\varepsilon}{2}, \\ \left(\frac{\varepsilon}{c} - 1\right)^{-2} & \text{if } c \leq \frac{\varepsilon}{2}, \end{cases}$$

it holds $\sigma(\tilde{\boldsymbol{R}}) = \sigma(\boldsymbol{R}) \subset B_{0.5}(0)$ and hence $\sigma(\mathcal{A}_0^{-1}\mathcal{A}) \subset B_{0.5}(1)$. □

Remark 3.5. *Note that the time step condition in Theorem 3.16 for small penalizations, i.e., large $c > \varepsilon/2$, complies with the uniqueness condition in Lemma 3.3 and the subsequent analysis. For large penalizations, i.e., small $c \leq \varepsilon/2$, the above time step condition in Theorem 3.16 is even more restrictive than the uniqueness condition in Lemma 3.3.*

Hence, we build on the second part of the last section and propose the block triangular preconditioner

$$\mathcal{P} = \begin{bmatrix} \boldsymbol{M} & \boldsymbol{0} \\ \tau\boldsymbol{K} & -\hat{\boldsymbol{S}} \end{bmatrix}. \tag{3.118}$$

As Schur complement approximation, we design $\hat{\boldsymbol{S}}$ as

$$\begin{aligned} \hat{\boldsymbol{S}} &= \boldsymbol{S}_1\boldsymbol{M}^{-1}\boldsymbol{S}_2 \\ &= \left(\boldsymbol{M} + \sqrt{\tau}\boldsymbol{K}\right)\boldsymbol{M}^{-1}\left(\boldsymbol{M} + \sqrt{\tau}\boldsymbol{A}\right) && (3.119) \\ &= \boldsymbol{M} + \tau\boldsymbol{K}\boldsymbol{M}^{-1}\boldsymbol{A} + \sqrt{\tau}\boldsymbol{A} + \sqrt{\tau}\boldsymbol{K}. && (3.120) \end{aligned}$$

The first two terms in (3.120) match the exact Schur complement. Due to the balanced distribution of τ in form of $\sqrt{\tau}$ in both factors $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$, the influence of both remainder terms in (3.120) is reduced. Note that we do not need to solve a linear system with the difficult indefinite block $\boldsymbol{A}$ anymore. Instead, we have to solve a shifted linear system with $\boldsymbol{A}$. More precisely, if we write out the block $\boldsymbol{S}_2$

$$\boldsymbol{S}_2 = \boldsymbol{M} + \sqrt{\tau}\boldsymbol{A} = \sqrt{\tau}\varepsilon\boldsymbol{K} + \frac{\sqrt{\tau}}{c}\boldsymbol{G} + \left(1 - \frac{\sqrt{\tau}}{\varepsilon}\right)\boldsymbol{M},$$

we see that $\boldsymbol{S}_2$ is positive definite if $\tau < \varepsilon^2$. Remember the uniqueness condition $\tau < 4\varepsilon^3$ in Lemma 3.3 that is imposed on our time-discrete formulation anyway. For $\varepsilon \leq 0.25$, it holds that $4\varepsilon^3 < \varepsilon^2$. In the numerical experiments, we never use interfacial parameters larger than 0.25. Instead, ε should be as small as possible. Hence, the positive definiteness of $\boldsymbol{S}_2$ is guaranteed if $\tau < 4\varepsilon^3$ and $\varepsilon \leq 0.25$. In fact, the positive definiteness of $\boldsymbol{S}_2$ is always guaranteed in our numerical experiments. To illustrate the performance of $\hat{\boldsymbol{S}}^{-1}\boldsymbol{S}$, we show eigenvalue plots in Section 3.8.1. In Section 3.8.2, we illustrate the robust performance of the preconditioner $\boldsymbol{\mathcal{P}}_0$ applied with BiCG.

Here, we finish the theoretical discussion about the preconditioners. In the next section, we illustrate their efficiency via various numerical experiments. Moreover, we outline an algorithm for the numerical solution of the Cahn–Hilliard problems.

3.8 Numerical results

In this section, we show numerical results for the presented Cahn–Hilliard problems. First, we explain our implementation framework.

The connection between the spatial mesh size h, the interfacial parameter ε, and the time step size τ is as follows. As discussed in Chapter 1.1, it is essential to ensure that at least eight grid points lie on the interface. Using smooth potentials, this leads to the condition $h \leq 2\sqrt{2}\varepsilon \cdot \operatorname{atanh}(0.95)/9$; see, e.g., [56, Section 7.9] or the note in [6, p. 374]. Using nonsmooth potentials, this leads to the condition $h \leq \varepsilon\pi/9$; see, e.g., [42, p. 17]. As far as we know, there is no theory available for the case of the regularized nonsmooth potential. In our numerical examples, we use the condition $h \leq \varepsilon\pi/9$ for the regularized nonsmooth potential. Regarding the time step size, we have analyzed the time step conditions for the smooth and (regularized) nonsmooth implicit time-discrete system in Section 3.3. For the former, we use $\tau < \varepsilon^3$ and for the latter it is $\tau < 4\varepsilon^3$.

Our finite element implementation is done in C++ on a 64-bit server with CPU type Intel® Xeon® X5650 @2.67 GHz, with 2 CPUs, 12 Cores (6 Cores per CPU), and 48 GB main memory available. We make use of the open source finite element library deal.II version 7.1.0 [8], which generates finite elements on rectangles. Deal.II works with the Trilinos library [89], version 10.4.2 in our case, which we use for our multilevel approximations. More precisely, we use the smoothed aggregation AMG implemented as part of the Trilinos Multi Level Preconditioning Package (ML) [76]. The ML package allows for a variety of smoothers. In our experiences, ten steps of a Chebyshev smoother were superior to similar or smaller numbers of Gauss-Seidel or Jacobi smoothing steps. The Chebyshev smoother is the recommended option for the discretization of elliptic operators; see, e.g., [89]. The use of two AMG V-cycles was always sufficient for our experiments, and, in fact, we did not observe improvements when a larger number of V-cycles was used.

In the following, we present two algorithms for the numerical solution of the implicit time-discrete Cahn–Hilliard problem with a nonsmooth potential. Algorithm 3.1 is based on a uniform spatial mesh. Algorithm 3.2 is based on an adaptive spatial mesh. For a description of our adaptive mesh strategy, we refer to Section 3.8.3. If not mentioned otherwise, we consider the unit square $\Omega = [0,1]^2$ as spatial domain. Let us explain some parts of Algorithm 3.1. We provide an initial solution $\boldsymbol{u}^{(0)}$ and set $\boldsymbol{w}^{(0)} = \boldsymbol{0}$. If not mentioned otherwise, we set the entries of the initial vector

Algorithm 3.1: The numerical solution of the nonsmooth implicit time-discrete Cahn–Hilliard problem via an SSN method combined with a Moreau–Yosida regularization technique on a uniform mesh.

```
Choose $h, \varepsilon, \tau, c_1, c_2, \ldots, c_{p_{\max}}, n_c, \epsilon_{\mathrm{rel}}, \epsilon_{\mathrm{abs}}$
Build the spatial mesh
Initialize $\boldsymbol{M}, \boldsymbol{K}$, and the AMG solver for $\boldsymbol{S_1}$
Set $\boldsymbol{u}^{(0)}, \boldsymbol{w}^{(0)}$
for $n = 1, \ldots, n_T$ do
    Update the second right-hand side of the linear system
    for $p = 1, 2, \ldots, p_{\max}$ do
        if $n > n_c$ then
            $p = p_{\max}$
        end
        $c = c_p$
        if $p = 1$ or $n > n_c$ then
            Set $\boldsymbol{u}^{(n,p,0)} = \boldsymbol{u}^{(n-1)}, \boldsymbol{w}^{(n,p,0)} = \boldsymbol{w}^{(n-1)}$
        else
            Set $\boldsymbol{u}^{(n,p,0)} = \boldsymbol{u}^{(n,p-1)}, \boldsymbol{w}^{(n,p,0)} = \boldsymbol{w}^{(n,p-1)}$
        end
        for $k = 0, 1, 2, \ldots$ until convergence do
            Update the first right-hand side of the linear system
            Update the block $\boldsymbol{A}$
            Update the AMG solver for $\boldsymbol{S_2}$
            Solve the linear system and obtain $\boldsymbol{u}^{(n,p,k+1)}, \boldsymbol{w}^{(n,p,k+1)}$
            if $||F_{c,h}(\boldsymbol{u}^{(n,p,k+1)}, \boldsymbol{w}^{(n,p,k+1)})|| \leq \epsilon_{\mathrm{rel}} ||F_{c,h}(\boldsymbol{u}^{(n,p,0)}, \boldsymbol{w}^{(n,p,0)})||_2 + \epsilon_{\mathrm{abs}}$ then
                Set $\boldsymbol{u}^{(n,p)} = \boldsymbol{u}^{(n,p,k+1)}, \boldsymbol{w}^{(n,p)} = \boldsymbol{w}^{(n,p,k+1)}$
                break
            end
        end
    end
    Set $\boldsymbol{u}^{(n)} = \boldsymbol{u}^{(n,p_{\max})}, \boldsymbol{w}^{(n)} = \boldsymbol{w}^{(n,p_{\max})}$
end
```

$\boldsymbol{u}^{(0)}$ randomly between -0.3 and 0.5, i.e., no pure phases are present at time $t = 0$. The first loop we enter is the one over the time step n. Therein is the second loop, which runs over the Moreau–Yosida regularization parameter c. In Algorithm 3.1, we run this loop over an integer p, which serves as a subindex for the sequence of regularization parameters $\{c_p\}_{p=1,\ldots,p_{\max}}$. Inside this second loop, we solve the regularized subproblem $F_{c_p,h}(\boldsymbol{u}^{(n,p)}, \boldsymbol{w}^{(n,p)}) = 0$ by the SSN method. Hence, the third loop is the SSN iteration, which runs over the SSN step k. But first, let us say a few more words about the second loop and the sequence of regularization parameters. In our numerical experiments, we set $c_1 = 10^{-1} \geq c_2 = 10^{-2} \geq \ldots \geq c_{p_{\max}} = 10^{-7}$ if not mentioned otherwise. In praxis, it is sufficient to loop over the sequence of penalization parameters $\{c_p\}_{p=1,\ldots,p_{\max}}$ only for the very first few time steps as soon as pure phases exist. From then on, it suffices to solve only for the smallest parameter $c = c_{p_{\max}}$. This is because the initial solution might not be a good starting point for the SSN method. If not mentioned otherwise, we set $n_c = 5$. This means, we fix $c = c_{p_{\max}}$ from time step 6 onwards, and the loop over p becomes unnecessarily. Now, we come to the SSN iteration, which forms the third loop. Except for the very first SSN call, each is initialized by the approximate solution of the previous one. The very first SSN call is initialized with $(\boldsymbol{u}^{(0)}, \boldsymbol{w}^{(0)})$. As stopping criterion, we use

$$\|F_{c,h}(\boldsymbol{u}^{(n,p,k+1)}, \boldsymbol{w}^{(n,p,k+1)})\| \leq \epsilon_{\text{rel}} \|F_{c,h}(\boldsymbol{u}^{(n,p,0)}, \boldsymbol{w}^{(n,p,0)})\| + \epsilon_{\text{abs}}, \quad k = 1, \ldots, k_{\max},$$

given in [91]. We set $k_{\max} = 20$, $\epsilon_{\text{rel}} = 10^{-12}$, and $\epsilon_{\text{abs}} = 10^{-6}$ in all examples. In each SSN step, we solve the system of linear equations by a Krylov subspace solver. This forms the core of the overall algorithm. We choose either preconditioned BiCG, BiCGstab, or MINRES as pointed out in Section 3.7. We set the BiCG/BiCGstab/MINRES tolerance to be 10^{-7} for the preconditioned relative residual in all examples. Except for the very first Krylov subspace solver call, each is initialized by the approximate solution $(\boldsymbol{u}^{(n,p,k)}, \boldsymbol{0})$ of the previous one. The very first Krylov subspace solver call is initialized with $(\boldsymbol{u}^{(0)}, \boldsymbol{0})$.

Algorithm 3.2 is a version of Algorithm 3.1, which uses adaptive finite elements. This results in hanging nodes. Hence, additional constraints have to be imposed to insure that the solutions of the systems of linear equations are continuous at these nodes. We just note that deal.II handles these constraints via a special object called ConstraintMatrix. Hence, we do not have to take care of these issues. However, the mass matrix $\boldsymbol{M}$ might not be diagonal anymore. Hence, in the adaptive case, we approximate the inverse of $\boldsymbol{M}$ with an AMG. More details about our mesh refinement and coarsening technique are given in Section 3.8.3.

The formulation with a smooth potential is a simplification of the presented two algorithms. Basically, the second loop over the penalty parameter drops out and the SSN method becomes a standard Newton method.

Now, we are ready for numerical results.

3.8.1 Eigenvalue plots

In Section 3.7, we have developed different Schur complement approximations and referred to this section for some corresponding eigenvalue plots. The following eigenvalue plots are simply generated with MATLAB®. The mass and stiffness matrix $\boldsymbol{M}$ and $\boldsymbol{K}$ are generated in C++ using the FEM library deal.II [8] as described above. For the following simple demonstrations, we consider uniform refinements of the unit square $[0,1]^2$ with three different mesh sizes $h_i = 2^{-i-3}$ for $i = 1,2,3$. Let us denote the diagonal matrix $\boldsymbol{H}$ in (3.88) for each mesh by $\boldsymbol{H}^{(i)}$ for $i = 1,2,3$. It is implemented in MATLAB as a random vector with MATLAB's command `rand`. We choose $\delta = 0.03$ in (3.89) and generate three vectors $\boldsymbol{h_i} = [r_1, \ldots, r_{m_i}]^T \in \mathbb{R}^{m_i}$, where $m_i = (h_i^{-1} + 1)^2$ for $i = 1,2,3$, from the uniform distribution on the interval $[0, 3 + 6\delta + 3\delta^2]$. The diagonal values of $\boldsymbol{H}^{(i)}$ are then set to be $[\boldsymbol{H}^{(i)}]_{jj} = [\boldsymbol{M}]_{jj} r_j$ for $j = 1, \ldots m_i$. Similarly, let us denote the diagonal matrix $\boldsymbol{G}$ in (3.85) for each mesh by $\boldsymbol{G}^{(i)}$ for $i = 1,2,3$. It is implemented in MATLAB as a random vector with MATLAB's command `randperm`. First, we initialize three vectors $\boldsymbol{g_i} = [p_1, \ldots, p_{m_i}]^T \in \mathbb{R}^{m_i}$ as one vectors. Then, we set randomly 25 percent of each vector $\boldsymbol{g_i}$ to zero via `randperm`. The diagonal values of $\boldsymbol{G}^{(i)}$ are then set to be $[\boldsymbol{G}^{(i)}]_{jj} = [\boldsymbol{M}]_{jj} p_j$ for $j = 1, \ldots m_i$. The

Algorithm 3.2: The numerical solution of the nonsmooth implicit time-discrete Cahn–Hilliard problem via an SSN method combined with a Moreau–Yosida regularization technique on an adaptive mesh.

```
Choose $h, \varepsilon, \tau, c_1, c_2, \dots, c_{p_{\max}}, n_c, \epsilon_{\mathrm{rel}}, \epsilon_{\mathrm{abs}}$
Build the initial spatial mesh
Initialize $\boldsymbol{M}, \boldsymbol{K}$, and the AMG solver for $\boldsymbol{S_1}$
Set $\boldsymbol{u}^{(0)}, \boldsymbol{w}^{(0)}$
**for** $n = 1, \dots, n_T$ **do**
    **if** $n \geq 2$ **then**
        Refine/coarsen the spatial mesh
        Update $\boldsymbol{M}, \boldsymbol{K}$, and the AMG solver for $\boldsymbol{S_1}$
        Transfer the solution $\boldsymbol{u}^{(n-1)}$ to the new mesh
    **end**
    Update the second right-hand side of the linear system
    **for** $p = 1, 2, \dots, p_{\max}$ **do**
        **if** $n > n_c$ **then**
            $p = p_{\max}$
        **end**
        $c = c_p$
        **if** $p = 1$ *or* $n > n_c$ **then**
            Set $\boldsymbol{u}^{(n,p,0)} = \boldsymbol{u}^{(n-1)}, \boldsymbol{w}^{(n,p,0)} = \boldsymbol{w}^{(n-1)}$
        **else**
            Set $\boldsymbol{u}^{(n,p,0)} = \boldsymbol{u}^{(n,p-1)}, \boldsymbol{w}^{(n,p,0)} = \boldsymbol{w}^{(n,p-1)}$
        **end**
        **for** $k = 0, 1, 2, \dots$ *until convergence* **do**
            Update the first right-hand side of the linear system
            Update the block $\boldsymbol{A}$
            Update the AMG solver for $\boldsymbol{S_2}$
            Solve the linear system and obtain $\boldsymbol{u}^{(n,p,k+1)}, \boldsymbol{w}^{(n,p,k+1)}$
            **if** $||F_{c,h}(\boldsymbol{u}^{(n,p,k+1)}, \boldsymbol{w}^{(n,p,k+1)})|| \leq \epsilon_{\mathrm{rel}}||F_{c,h}(\boldsymbol{u}^{(n,p,0)}, \boldsymbol{w}^{(n,p,0)})||_2 + \epsilon_{\mathrm{abs}}$ **then**
                Set $\boldsymbol{u}^{(n,p)} = \boldsymbol{u}^{(n,p,k+1)}, \boldsymbol{w}^{(n,p)} = \boldsymbol{w}^{(n,p,k+1)}$
                break
            **end**
        **end**
    **end**
    Set $\boldsymbol{u}^{(n)} = \boldsymbol{u}^{(n,p_{\max})}, \boldsymbol{w}^{(n)} = \boldsymbol{w}^{(n,p_{\max})}$
**end**
```

action of all inverses are performed with MATLAB's `backslash` command. This is a direct solver based on the LU-factorization, which works well for our small sized two-dimensional problems. In total, three inverses occur in the implementation: One in the Schur complement $\boldsymbol{S}$, one in the Schur complement approximation $\hat{\boldsymbol{S}}$, as well as one in $\hat{\boldsymbol{S}}^{-1}\boldsymbol{S}$. Finally, we have used MATLAB's `eigs` command to obtain the eigenvalues of the generated matrix $\hat{\boldsymbol{S}}^{-1}\boldsymbol{S}$.

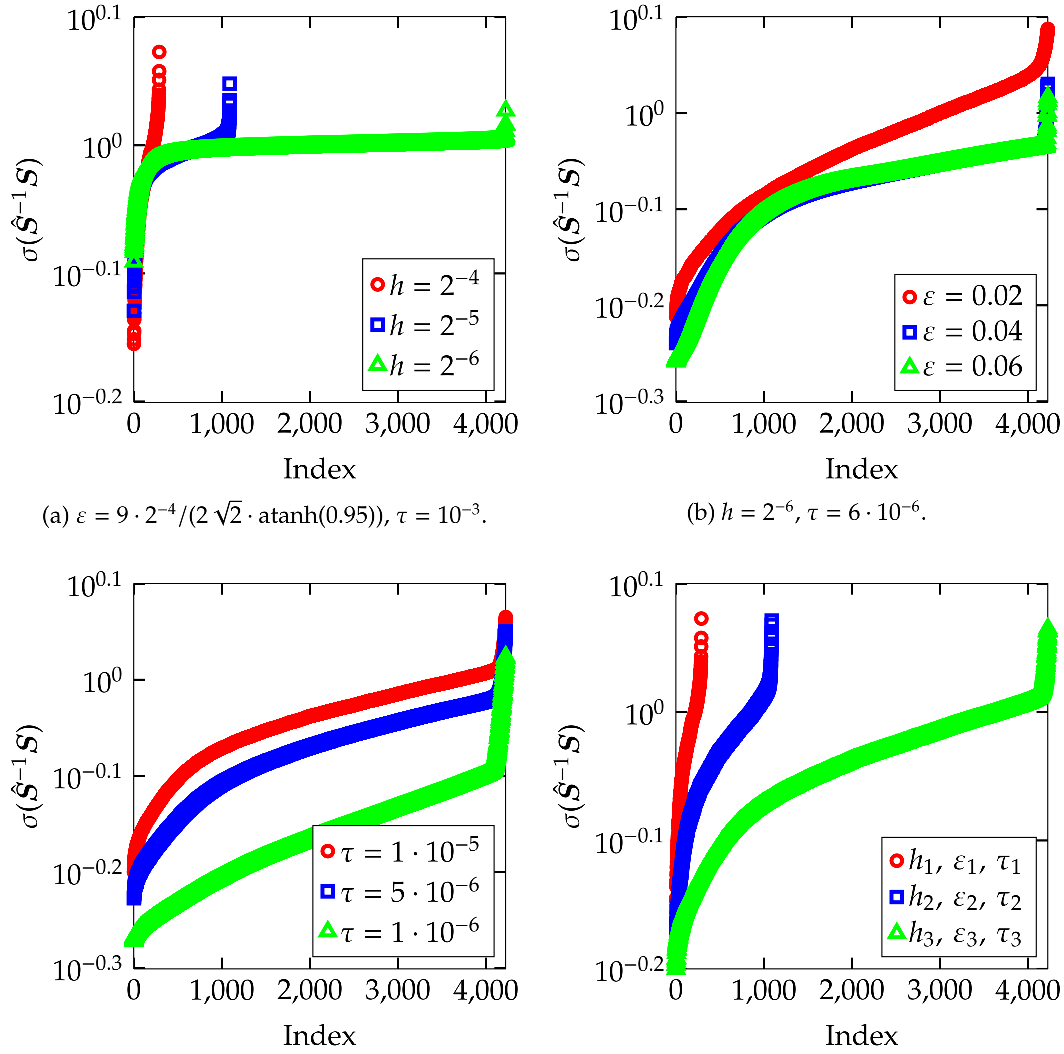
We start with the smooth semi-implicit system (3.90) with the Schur complement approximation (3.92). Each subplot in Figure 3.2(a)–3.2(c) demonstrates the robustness with respect to a different model parameter. In Figure 3.2(a), we vary the mesh size h while fixing $\varepsilon = 9 \cdot 2^{-4}/(2\sqrt{2} \cdot \operatorname{atanh}(0.95))$, $\tau = 10^{-3}$. In Figure 3.2(b), we vary the interfacial parameter ε while fixing $h = 2^{-6}$, $\tau = 6 \cdot 10^{-6}$. In Figure 3.2(c), we vary the time step size τ while fixing $h = 2^{-6}$, $\varepsilon = 9 \cdot 2^{-6}/(2\sqrt{2} \cdot \operatorname{atanh}(0.95))$. Finally, in Figure 3.2(d), we vary simultaneously all three parameters h, ε, τ. In fact, this is the practical procedure: Choose ε as small as possible (depending on h) and τ as large as possible (depending on the accuracy). Each of the four subplots illustrates nicely the eigenvalue clustering around one. Moreover, all eigenvalues are real and positive as expected from Lemma 2.30.

Next, we go over to the nonsmooth semi-implicit system (3.108) with the Schur complement approximation (3.112). Each subplot in Figure 3.3(a)–3.3(d) demonstrates the robustness with respect to a different model parameter. In Figure 3.3(a), we vary the mesh size h while fixing $\varepsilon = 9 \cdot 2^{-4}/\pi$, $\tau = 10^{-2}$, $c = 10^{-7}$. In Figure 3.3(b), we vary the interfacial parameter ε while fixing $h = 2^{-6}$, $\tau = 10^{-4}$, $c = 10^{-7}$. In Figure 3.3(c), we vary the time step size τ while fixing $h = 2^{-6}$, $\varepsilon = 9 \cdot 2^{-6}/\pi$, $c = 10^{-7}$. In Figure 3.3(d), we vary the penalty parameter c while fixing $h = 2^{-6}$, $\varepsilon = 9 \cdot 2^{-6}/\pi$, $\tau = 10^{-4}$. Finally, in Figure 3.3(e), we vary simultaneously all three parameters h, ε, τ while fixing $c = 10^{-7}$. Each of the five subplots illustrates nicely the eigenvalue clustering around one. Moreover, all presented eigenvalues are real and positive.

We proceed with the nonsmooth implicit system (3.117) with the Schur complement approximation (3.119). Figures 3.4(a)–3.4(e) demonstrate the robustness with respect to different model parameters. In Figure 3.4(a), we vary the mesh size h while fixing $\varepsilon = 9 \cdot 2^{-4}/\pi$, $\tau = 10^{-2}$, $c = 10^{-7}$. In Figure 3.4(b), we vary the interfacial parameter ε while fixing $h = 2^{-6}$, $\tau = 10^{-4}$, $c = 10^{-7}$. In Figure 3.4(c) and 3.4(d), we vary the time step size τ while fixing $h = 2^{-6}$, $\varepsilon = 9 \cdot 2^{-6}/\pi$, $c = 10^{-7}$. Here, we observe the appearance of complex eigenvalues. In Figure 3.4(e), we vary the penalty parameter c while fixing $h = 2^{-6}$, $\varepsilon = 9 \cdot 2^{-6}/\pi$, $\tau = 10^{-4}$. Finally, in Figure 3.4(f), we vary simultaneously all three parameters h, ε, τ while fixing $c = 10^{-7}$. Again, each of the six subplots illustrates nicely the eigenvalue clustering around one.

3.8.2 Robustness

In this section, we demonstrate the robustness of our proposed preconditioners regarding all model parameters. We will not present numerical results for the smooth implicit system in (3.107) since the theory in Section 3.7.2 already promises the independence of the preconditioner regarding any model parameter.



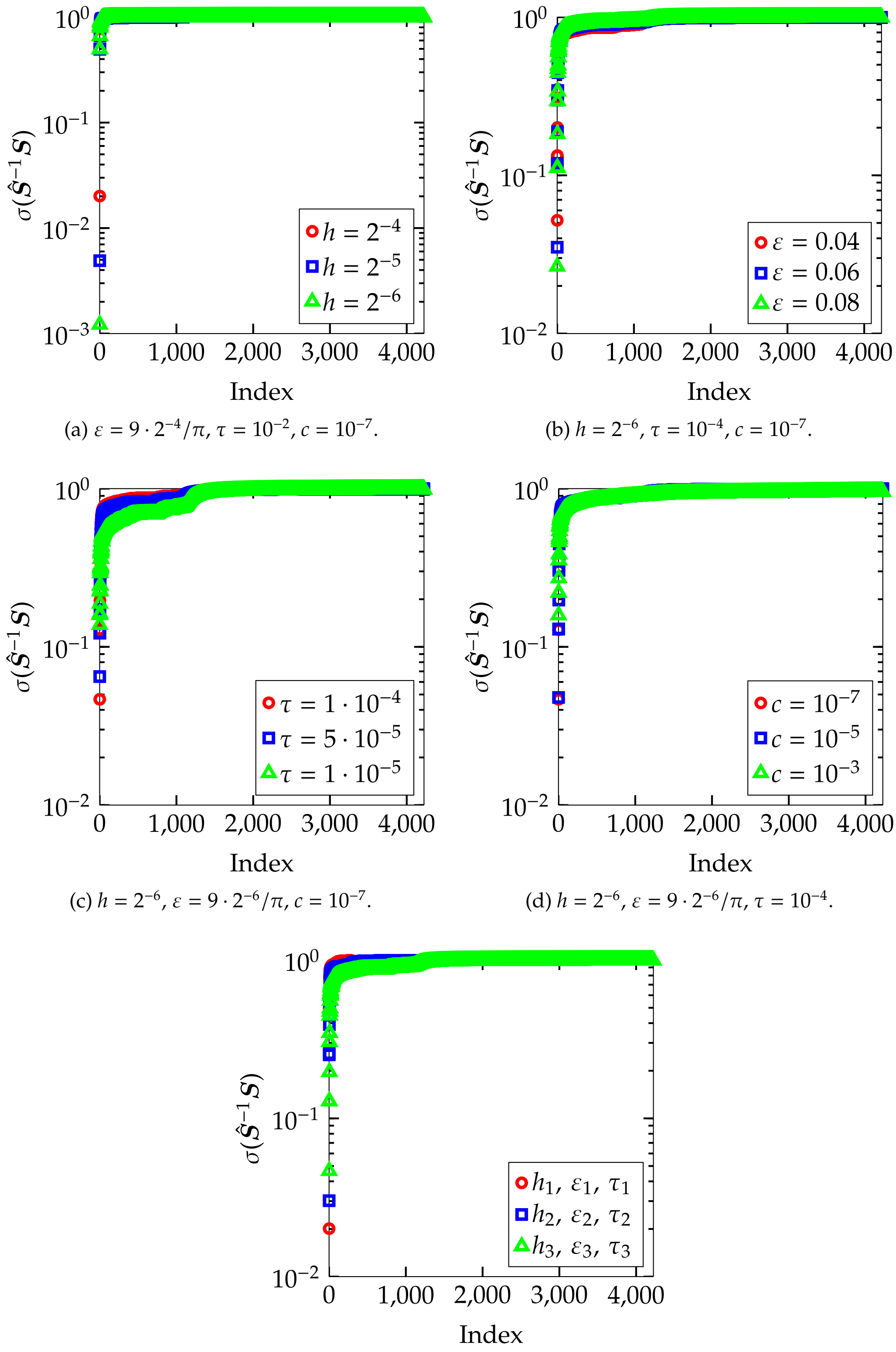
(a) $\varepsilon = 9 \cdot 2^{-4}/(2\sqrt{2} \cdot \operatorname{atanh}(0.95))$, $\tau = 10^{-3}$.

(b) $h = 2^{-6}$, $\tau = 6 \cdot 10^{-6}$.

(c) $h = 2^{-6}$, $\varepsilon = 9 \cdot 2^{-6}/(2\sqrt{2} \cdot \operatorname{atanh}(0.95))$.

(d) $h_j = 2^{-j-3}$, $\varepsilon_j = 9\,h_j/(2\sqrt{2} \cdot \operatorname{atanh}(0.95))$, $\tau_j = 10^{-j-2}$ for $j = 1, 2, 3$.

Figure 3.2: Spectrum of $\hat{\boldsymbol{S}}^{-1}\boldsymbol{S}$ for the smooth semi-implicit system (3.90) with the Schur complement approximation (3.92).



(a) $\varepsilon = 9 \cdot 2^{-4}/\pi$, $\tau = 10^{-2}$, $c = 10^{-7}$.

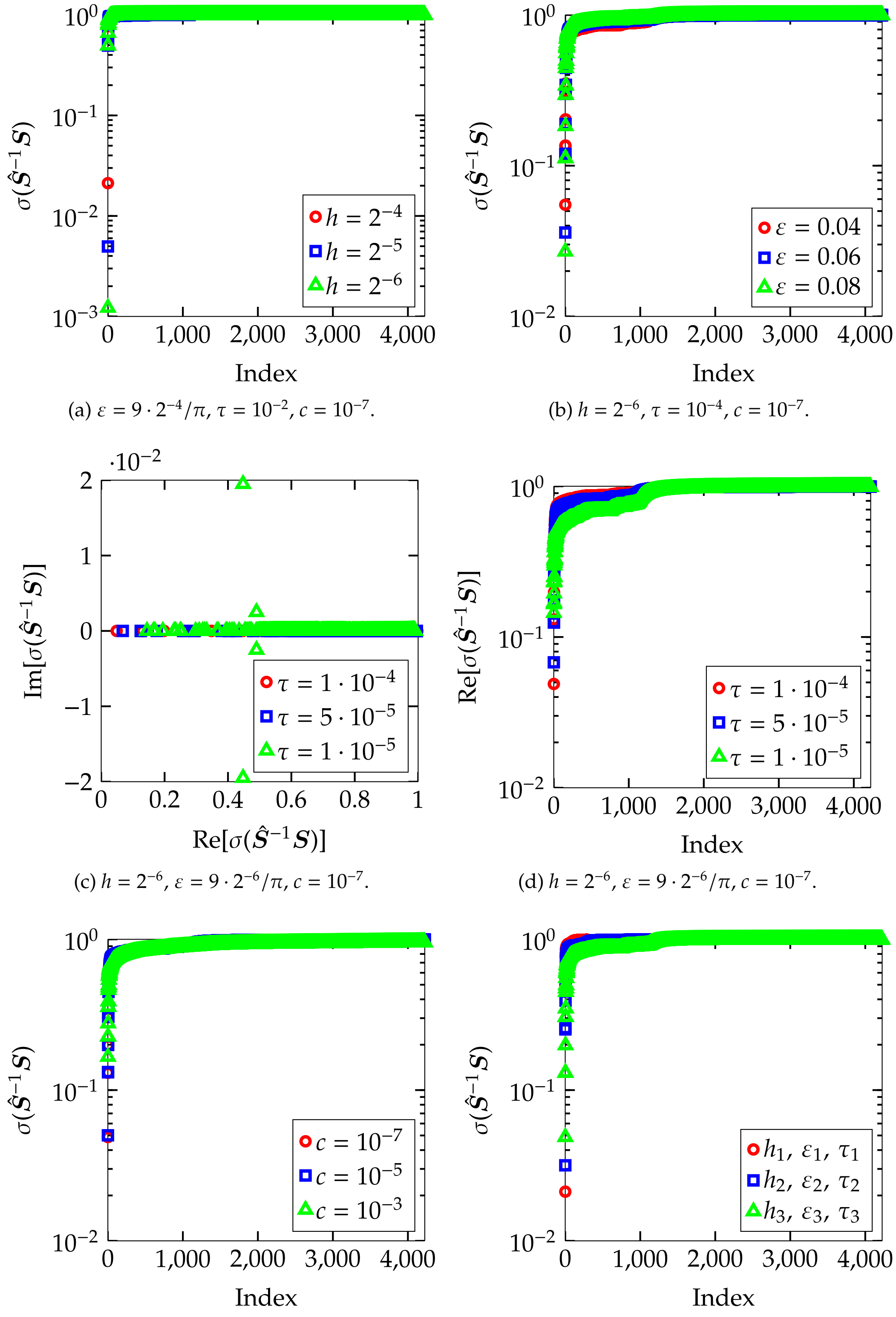
(b) $h = 2^{-6}$, $\tau = 10^{-4}$, $c = 10^{-7}$.

(c) $h = 2^{-6}$, $\varepsilon = 9 \cdot 2^{-6}/\pi$, $c = 10^{-7}$.

(d) $h = 2^{-6}$, $\varepsilon = 9 \cdot 2^{-6}/\pi$, $\tau = 10^{-4}$.

(e) $h_j = 2^{-j-3}$, $\varepsilon_j = 9h_j/\pi$, $\tau_j = 10^{-j-1}$ for $j = 1, 2, 3$, $c = 10^{-7}$.

Figure 3.3: Spectrum of $\hat{S}^{-1}S$ for the nonsmooth semi-implicit system (3.108) with the Schur complement approximation (3.112).



(a) $\varepsilon = 9 \cdot 2^{-4}/\pi$, $\tau = 10^{-2}$, $c = 10^{-7}$.

(b) $h = 2^{-6}$, $\tau = 10^{-4}$, $c = 10^{-7}$.

(c) $h = 2^{-6}$, $\varepsilon = 9 \cdot 2^{-6}/\pi$, $c = 10^{-7}$.

(d) $h = 2^{-6}$, $\varepsilon = 9 \cdot 2^{-6}/\pi$, $c = 10^{-7}$.

(e) $h = 2^{-6}$, $\varepsilon = 9 \cdot 2^{-6}/\pi$, $\tau = 10^{-4}$.

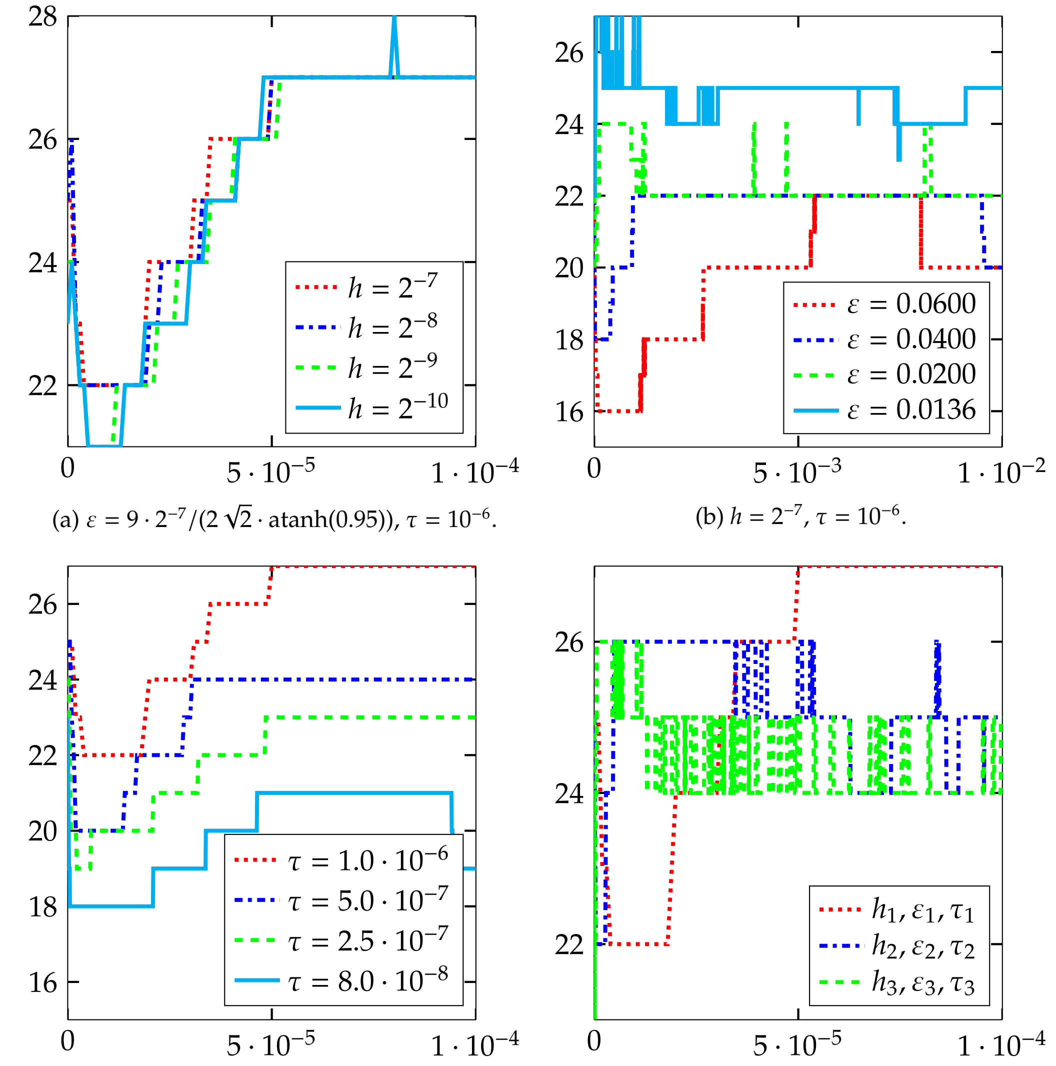
(f) $h_j = 2^{-j-3}$, $\varepsilon_j = 9\,h_j/\pi$, $\tau_j = 10^{-j-1}$ for $j = 1, 2, 3$, $c = 10^{-7}$.

Figure 3.4: Spectrum of $\hat{S}^{-1}S$ for the nonsmooth implicit system (3.117) with the Schur complement approximation (3.119).

We start with the smooth semi-implicit system (3.90) with the preconditioner (3.91) and the Schur complement approximation (3.92). Each subplot in Figure 3.5(a)–3.5(c) demonstrates the robustness with respect to a different model parameter. In Figure 3.5(a), we vary the mesh size h while fixing $\varepsilon = 9 \cdot 2^{-7}/(2\sqrt{2} \cdot \operatorname{atanh}(0.95))$, $\tau = 10^{-6}$, and $T = 10^{-4}$. In Figure 3.5(b), we vary the interfacial parameter ε while fixing $h = 2^{-7}$, $\tau = 10^{-6}$, and $T = 10^{-2}$. In Figure 3.5(c), we vary the time step size τ while fixing $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/(2\sqrt{2} \cdot \operatorname{atanh}(0.95))$, and $T = 10^{-4}$. All in all, the three subplots illustrate the independence of our developed preconditioner regarding the mesh size, the interfacial parameter, as well as the time step size. Finally, in Figure 3.5(d), we vary simultaneously all three parameters h, ε, τ while fixing $T = 10^{-4}$. Although the smooth semi-implicit system is unconditionally stable, we use in the numerical examples of this section the time step condition $\tau < \varepsilon^3$, which was investigated for smooth implicit systems in Section 3.3.1. This is because experiments yield highly inaccurate results for large time steps; see Section 3.8.4. Table 3.1 illustrates the maximum and average number of Newton iterations, the maximum and average number of MINRES iterations, the average central processing unit (CPU) time (in seconds) for MINRES, and the CPU time (in seconds) for the whole simulation for each of the four subplots, respectively.

Next, we go over to the nonsmooth semi-implicit system (3.108) with the preconditioner (3.110) and the Schur complement approximation (3.112). Each subplot in Figure 3.6 demonstrates the robustness with respect to a different model parameter. In Figure 3.6(a), we vary the mesh size h while fixing $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 2 \cdot 10^{-5}$, $c_{p_{\max}} = 10^{-7}$, and $T = 4 \cdot 10^{-4}$. In Figure 3.6(b), we vary the interfacial parameter ε while fixing $h = 2^{-7}$, $\tau = 2 \cdot 10^{-5}$, $c_{p_{\max}} = 10^{-7}$, and $T = 2 \cdot 10^{-2}$. In Figure 3.6(c), we vary the time step size τ while fixing $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $c_{p_{\max}} = 10^{-7}$, and $T = 4 \cdot 10^{-4}$. In Figure 3.6(d), we vary the penalty parameter $c_{p_{\max}}$ while fixing $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 2 \cdot 10^{-5}$, and $T = 4 \cdot 10^{-4}$. All in all, except for Figure 3.6(a), the subplots illustrate the independence of our developed preconditioner regarding the interfacial parameter, the time step, size as well as the penalty parameter. In fact, we observe a decrease of iteration numbers when $c_{p_{\max}}$ is decreased. We observe a benign increase of iteration numbers when the mesh is refined. Finally, in Figure 3.7(a), we vary simultaneously all three parameters h, ε, τ while fixing $c_{p_{\max}} = 10^{-7}$ and $T = 4 \cdot 10^{-4}$. Although the nonsmooth semi-implicit system has no time step restrictions, we use in the numerical examples of this section the time step condition $\tau < 4\varepsilon^3$, which was investigated for nonsmooth implicit systems in Section 3.3.2. This is because experiments yield highly inaccurate results for large time steps; see Section 3.8.4. Table 3.2 illustrates the maximum and average number of SSN iterations, the maximum and average number of BiCG iterations, the average CPU time (in seconds) for BiCG, and the CPU time (in seconds) for the whole simulation for each of the five subplots, respectively. Note that the higher number of SSN iterations in row three and four can be reduced by choosing n_c in Algorithm 3.1 larger.

We proceed with the nonsmooth implicit system (3.117) with the preconditioner (3.118) and the Schur complement approximation (3.119). Each subplot in Figure 3.8 demonstrates the robustness with respect to a different model parameter. In Figure 3.8(a), we vary the mesh size h while fixing $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 2 \cdot 10^{-5}$, $c_{p_{\max}} = 10^{-7}$, and $T = 4 \cdot 10^{-4}$. In order to reduce the number of SSN iterations, we have set



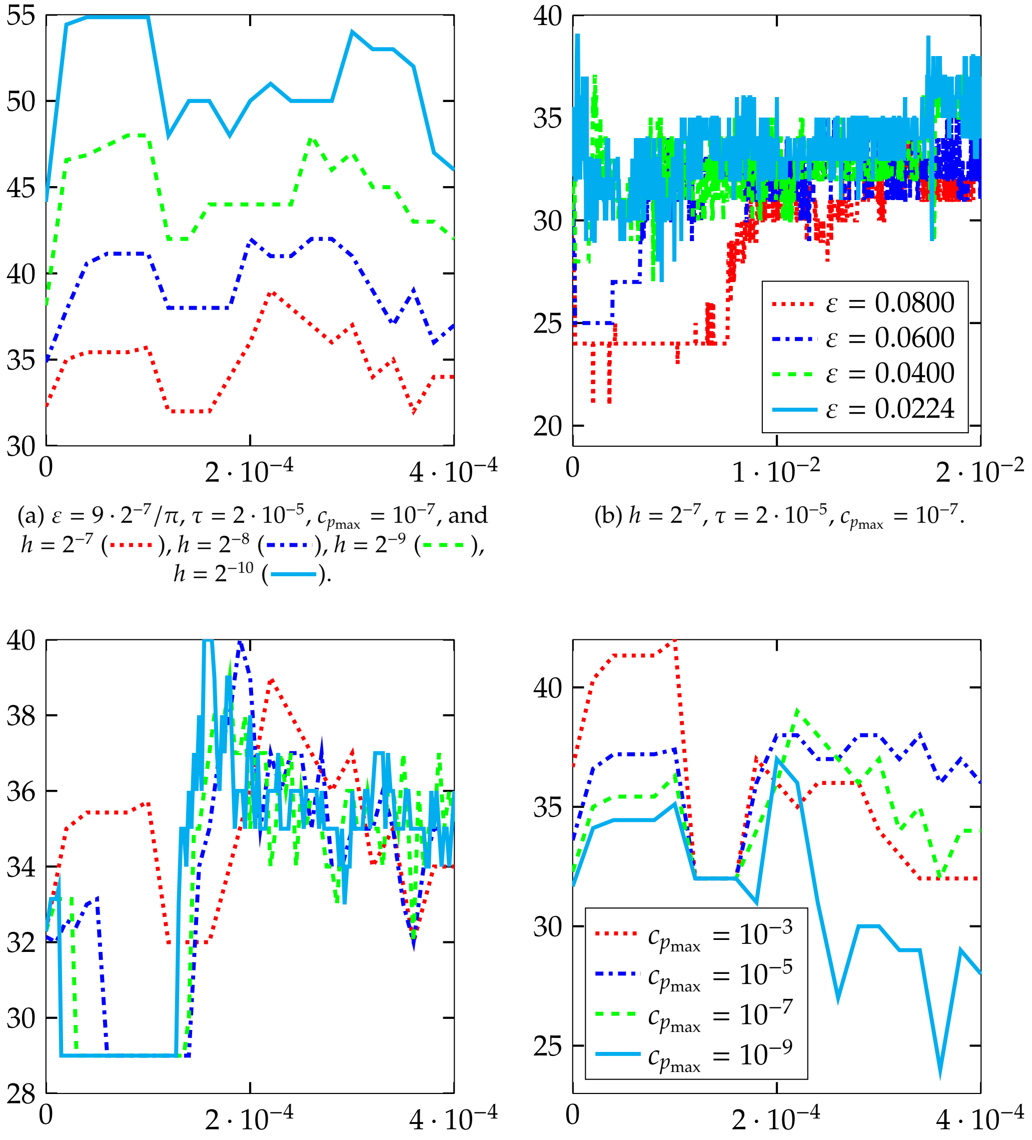
(a) $\varepsilon = 9 \cdot 2^{-7}/(2\sqrt{2} \cdot \operatorname{atanh}(0.95))$, $\tau = 10^{-6}$.

(b) $h = 2^{-7}$, $\tau = 10^{-6}$.

(c) $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/(2\sqrt{2} \cdot \operatorname{atanh}(0.95))$.

(d) $h_j = 2^{-j-6}$, $\varepsilon_j = 9\,h_j/(2\sqrt{2} \cdot \operatorname{atanh}(0.95))$, $\tau_1 = 10^{-6}$, $\tau_2 = 10^{-7}$, $\tau_3 = 1.25 \cdot 10^{-8}$ for $j = 1, 2, 3$.

Figure 3.5: Results for the solution of the smooth semi-implicit system (3.90) with the preconditioner (3.91) and the Schur complement approximation (3.92). The x-axis shows the time t and the y-axis the average number of MINRES iterations per Newton step.



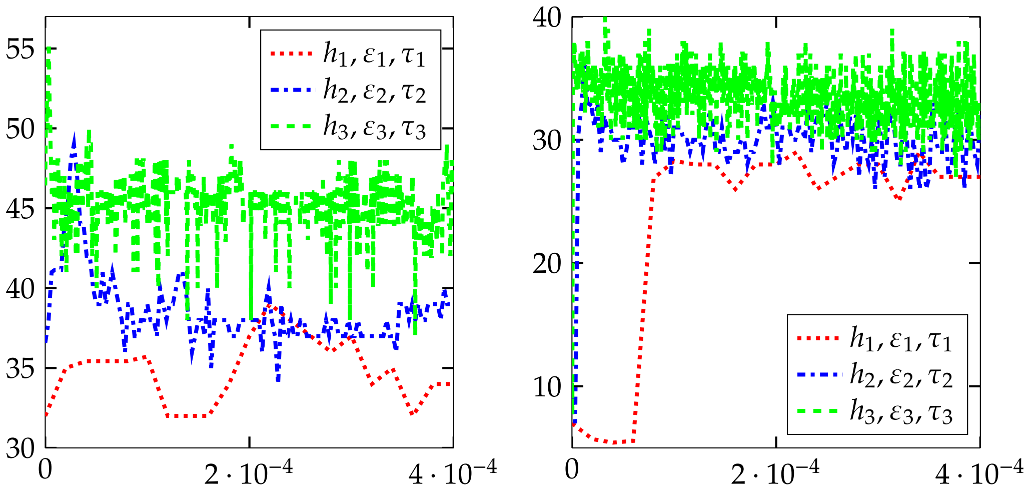
(a) $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 2 \cdot 10^{-5}$, $c_{p_{\max}} = 10^{-7}$, and $h = 2^{-7}$ (⋯⋯), $h = 2^{-8}$ (-·-·), $h = 2^{-9}$ (- - -), $h = 2^{-10}$ (——).

(b) $h = 2^{-7}$, $\tau = 2 \cdot 10^{-5}$, $c_{p_{\max}} = 10^{-7}$.

(c) $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $c_{p_{\max}} = 10^{-7}$, and $\tau = 2 \cdot 10^{-5}$ (⋯⋯), $\tau = 10^{-5}$ (-·-·), $\tau = 5 \cdot 10^{-6}$ (- - -), $\tau = 2.5 \cdot 10^{-6}$ (——).

(d) $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 2 \cdot 10^{-5}$.

Figure 3.6: Results for the solution of the nonsmooth semi-implicit system (3.108) with the preconditioner (3.110) and the Schur complement approximation (3.112). The x-axis shows the time t and the y-axis the average number of BiCG iterations per SSN step.



(a) Nonsmooth semi-implicit system (3.108) with the preconditioner (3.110) and the Schur complement approximation (3.112).

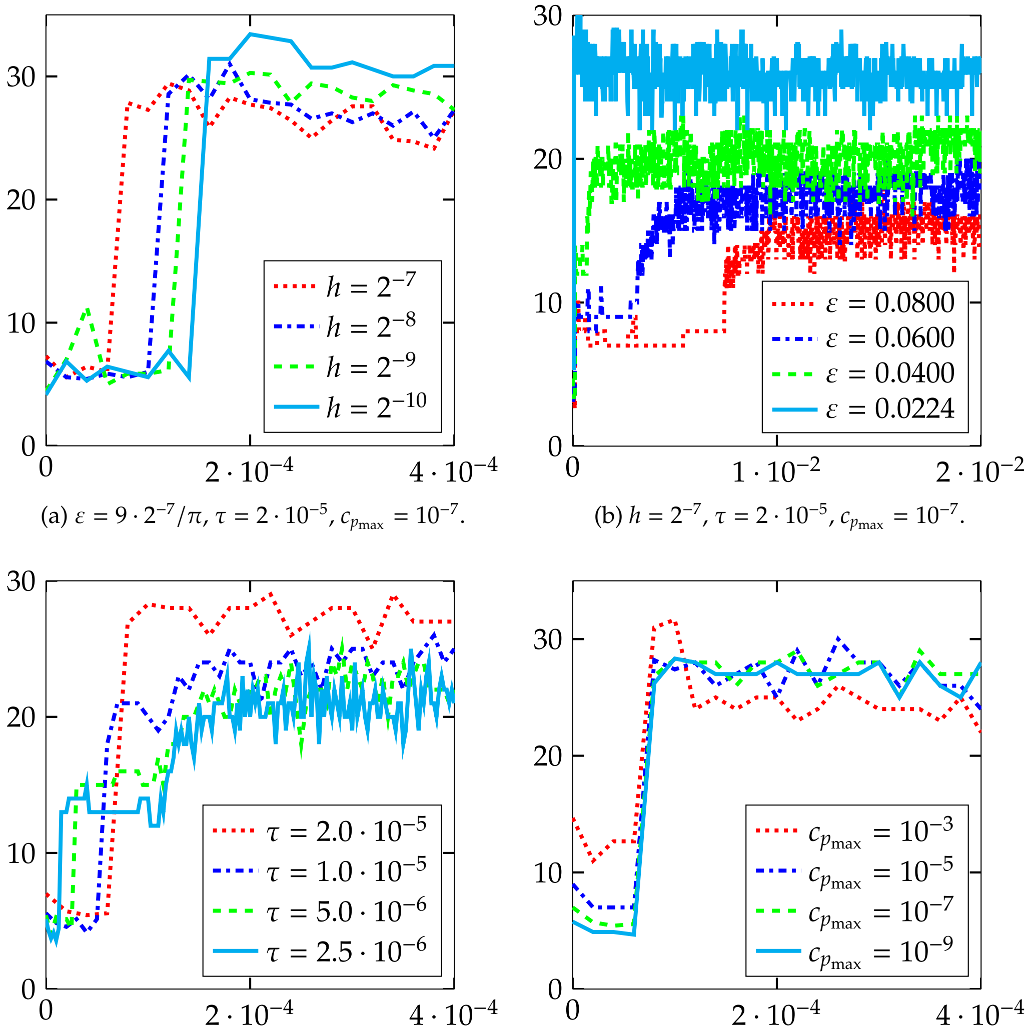
(b) Nonsmooth implicit system (3.117) with the preconditioner (3.118) and the Schur complement approximation (3.119).

Figure 3.7: Results for the solution of the nonsmooth semi-implicit and implicit system. The x-axis shows the time t and the y-axis the average number of BiCG iterations per SSN step. $h_j = 2^{-j-6}$, $\varepsilon_j = 9\, h_j/\pi$, $\tau_1 = 2 \cdot 10^{-5}$, $\tau_2 = 3.125 \cdot 10^{-6}$, $\tau_3 = 4 \cdot 10^{-7}$, $c_{p_{\max}} = 10^{-7}$ for $j = 1, 2, 3$.

$n_c = 20$ in Algorithm 3.1. For the following tests, we use $n_c = 5$. In Figure 3.8(b), we vary the interfacial parameter ε while fixing $h = 2^{-7}$, $\tau = 2 \cdot 10^{-5}$, $c_{p_{\max}} = 10^{-7}$, and $T = 2 \cdot 10^{-2}$. In Figure 3.8(c), we vary the time step size τ while fixing $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $c_{p_{\max}} = 10^{-7}$, and $T = 4 \cdot 10^{-4}$. In Figure 3.8(d), we vary the penalty parameter $c_{p_{\max}}$ while fixing $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 2 \cdot 10^{-5}$, and $T = 4 \cdot 10^{-4}$. All in all, except for Figure 3.8(b), the subplots illustrates the independence of our developed preconditioner regarding the mesh size, the time step size, as well as the penalty parameter. We observe a benign increase of iteration numbers when the interfacial parameter decreases. Finally, in Figure 3.7(b), we vary simultaneously all three parameters h, ε, τ while fixing $c_{p_{\max}} = 10^{-7}$ and $T = 4 \cdot 10^{-4}$. Table 3.3 illustrates the maximum and average number of SSN iterations, the maximum and average number of BiCG iterations, the average CPU time (in seconds) for BiCG, and the CPU time (in seconds) for the whole simulation for each of the five subplots, respectively.

3.8.3 Mesh adaptation

Experiments show that it is essential to ensure that at least eight spatial mesh points lie across the interface in order to avoid mesh effects; see also [29]. This results in the huge linear systems. The thinner we model the interface, the higher is the number of spatial mesh points and hence the system size m. This number can be reduced by the use of adaptive meshes. Hence, we refine the interface up to a level where at least eight mesh points are found across the interface. We coarsen the mesh in areas where the concentration u is (almost) constant. Using the nonsmooth potential, we can easily identify the interfacial and constant areas. The constant areas are the spatial points



(a) $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 2 \cdot 10^{-5}$, $c_{p_{\max}} = 10^{-7}$.

(b) $h = 2^{-7}$, $\tau = 2 \cdot 10^{-5}$, $c_{p_{\max}} = 10^{-7}$.

(c) $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $c_{p_{\max}} = 10^{-7}$.

(d) $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 2 \cdot 10^{-5}$.

Figure 3.8: Results for the solution of the nonsmooth implicit system (3.117) with the preconditioner (3.118) and the Schur complement approximation (3.119). The x-axis shows the time t and the y-axis the average number of BiCG iterations per SSN step.

Simulation		Newton		MINRES			
Figure	Plot	Max	Avg	Max	Avg	CPU (s)	CPU (s)
3.5(a)	(·····)	3	2	27	26	9	2041
	(----)	3	2	27	25	36	7916
	(- - -)	2	2	27	25	177	35257
	(——)	2	2	28	25	670	122054
3.5(b)	(·····)	2	1	28	20	7	82822
	(----)	2	1	27	22	7	89353
	(- - -)	3	1	27	22	8	100229
	(——)	3	1	27	25	9	123230
3.5(c)	(·····)	3	2	27	26	9	2041
	(----)	3	2	26	23	8	3718
	(- - -)	3	2	24	22	8	6910
	(——)	3	1	23	20	7	13190
3.5(d)	(·····)	3	2	27	26	9	2050
	(----)	3	2	26	25	36	78165
	(- - -)	3	1	26	24	175	2143690

Table 3.1: Results for the solution of the smooth semi-implicit system (3.90) with the preconditioner (3.91) and the Schur complement approximation (3.92): The maximum and average number of Newton iterations, the maximum and average number of MINRES iterations, the average CPU time (in seconds) for MINRES, as well as the CPU time (in seconds) for the whole simulation.

Simulation		SSN		BiCG			
Figure	Plot	Max	Avg	Max	Avg	CPU (s)	CPU (s)
3.6(a)	(·····)	5	2	46	32	23	2888
	(----)	7	2	48	36	99	12709
	(- - -)	10	2	54	41	551	74768
	(——)	18	3	60	47	2447	395285
3.6(b)	(·····)	4	2	36	28	20	40680
	(----)	5	2	37	31	27	64461
	(- - -)	6	2	42	32	33	85393
	(——)	5	2	46	33	47	122221
3.6(c)	(·····)	5	2	46	32	23	2943
	(----)	5	2	47	31	24	4859
	(- - -)	5	2	44	32	25	7831
	(——)	4	2	45	33	27	13524
3.6(d)	(·····)	5	2	40	32	21	1691
	(----)	5	2	43	32	22	2236
	(- - -)	5	2	46	32	23	2885
	(——)	6	2	44	31	21	2886
3.7(a)	(·····)	5	2	45	32	23	2954
	(----)	6	3	55	38	143	80432
	(- - -)	6	3	60	45	620	2033630

Table 3.2: Results for the solution of the nonsmooth semi-implicit system (3.108) with the preconditioner (3.110) and the Schur complement approximation (3.112): The maximum and average number of SSN iterations, the maximum and average number of BiCG iterations, the average CPU time (in seconds) for BiCG, and the CPU time (in seconds) for the whole simulation.

Simulation		SSN		BiCG			
Figure	Plot	Max	Avg	Max	Avg	CPU (s)	CPU (s)
3.8(a)	(·····)	4	2	31	20	10	4016
	(-·-·)	5	3	32	19	37	17211
	(- - -)	5	3	42	19	189	91930
	(——)	5	3	41	19	683	322118
3.8(b)	(·····)	5	2	19	12	6	14165
	(-·-·)	5	2	22	16	9	24015
	(- - -)	5	2	25	19	13	34711
	(——)	6	2	34	25	23	62833
3.8(c)	(·····)	6	2	34	16	9	1922
	(-·-·)	5	2	30	13	8	2299
	(- - -)	5	2	28	15	9	3348
	(——)	4	2	27	16	9	5593
3.8(d)	(·····)	6	3	31	19	9	1078
	(-·-·)	6	3	34	17	8	1440
	(- - -)	6	2	34	16	9	1900
	(——)	7	2	34	15	8	2153
3.7(b)	(·····)	6	2	34	16	9	1882
	(-·-·)	7	4	46	28	71	46596
	(- - -)	6	3	51	33	328	1121900

Table 3.3: Results for the solution of the nonsmooth implicit system (3.117) with the preconditioner (3.118) and the Schur complement approximation (3.119): The maximum and average number of SSN iterations, the maximum and average number of BiCG iterations, the average CPU time (in seconds) for BiCG, and the CPU time (in seconds) for the whole simulation.

$\mathbf{x}$ that satisfy $|u(\mathbf{x})| = 1$. The interfacial area is formed by those spatial points $\mathbf{x}$ that satisfy $|u(\mathbf{x})| < 1$. Using the regularized potential, the interfacial area is specified in the same way. The constant areas are the spatial points $\mathbf{x}$ that satisfy $|u(\mathbf{x})| \geq 1$. Using the smooth potential, it is not that clear where to separate the constant areas from the interfacial area as pointed out in the introduction in Chapter 1.1. As our simple approach is based on the knowledge about the location of constant and interfacial areas, we apply our adaptive mesh strategy only to the nonsmooth case. Our mesh adaptation strategy is similar to the one in [20]. As pointed out at the beginning of Chapter 3.8, for a given $\varepsilon > 0$ we use the upper bound $h_{\min} \leq \frac{\varepsilon\pi}{9}$, where $h_{\min}$ is the refinement level across the interface. Since we want to avoid meshes which are too coarse, we additionally define $h_{\max} := 10\, h_{\min}$, where $h_{\max}$ represents the maximal mesh size. Our mesh adaptation is based on the following strategy: An element $R \in \mathcal{R}_h$ is marked for refinement if it satisfies $\mathrm{diam}(R) > h_{\min}$ and if it, or one of its neighboring elements, satisfies $|u(\mathbf{x})| < 1$. Here, $\mathrm{diam}(R)$ denotes the largest diagonal of R. An element $R \in \mathcal{R}_h$ is marked for coarsening if it satisfies $\mathrm{diam}(R) < h_{\max}$ and $|u(\mathbf{x})| \geq 1$. Thereby, we refine in an area, which contains the interface and coarsen within the pure phases. Since we also include the neighboring cells for the refinement process, we do not coarsen too close to the interface. Note that it is also possible to incorporate more sophisticated adaptation strategies. For instance, Hintermüller et al. [91] designed an adaptive finite-element algorithm based on an a-posteriori error analysis.

In Figure 3.9, we illustrate the performance of our preconditioner (3.118) with the

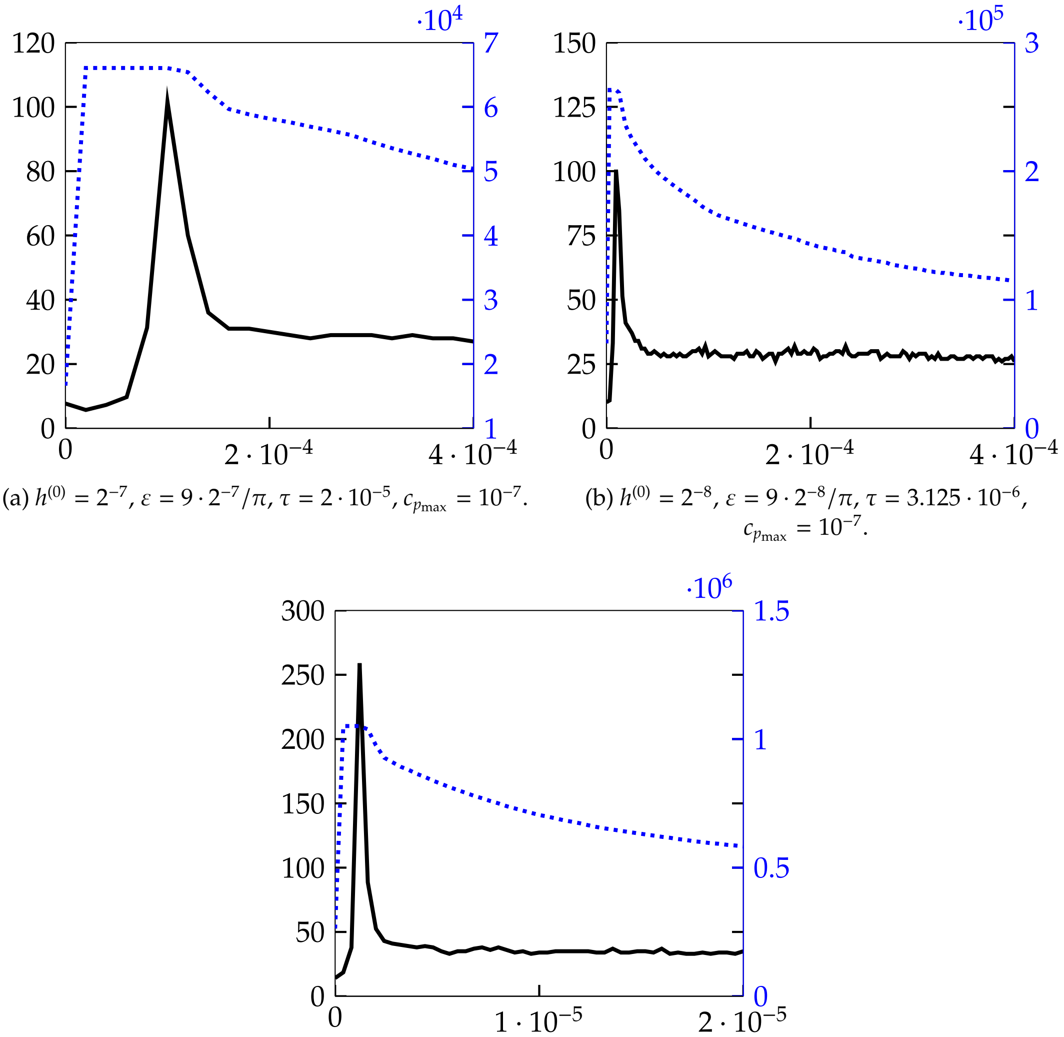
Schur complement approximation (3.119) for the solution of the nonsmooth implicit system (3.117) on adaptive meshes. We test three different settings with varying values of $h^{(0)}, \varepsilon, \tau$, where $h^{(0)}$ denotes the mesh size of the initial uniform mesh. The x-axis shows the time t, the left y-axis displays the average number of BiCG iterations per SSN step, and the right y-axis illustrates the number of degrees of freedom, respectively. Except for the peak, which occurs as soon as pure phases develop, the iteration numbers stay constantly low. Moreover, with the formation of pure phases, the coarsening process begins and the number of degrees of freedoms decreases. Table 3.4 illustrates the maximum and average number of SSN iterations, the maximum and average number of BiCG iterations, the average CPU time (in seconds) for BiCG, and the CPU time (in seconds) for the whole simulation for each of the three subplots, respectively. The final phase variable for each simulation is illustrated in Figure 3.10 together with the spatial mesh.

	SSN		BiCG			
Figure	Max	Avg	Max	Avg	CPU (s)	CPU (s)
3.9(a)	9	3	100	26	62	13952
3.9(b)	9	4	94	33	314	230026
3.9(c)	10	5	248	52	2156	842244

Table 3.4: Results for the solution of the nonsmooth implicit system (3.117) with the preconditioner (3.118) and the Schur complement approximation (3.119) using adaptive meshes: The maximum and average number of SSN iterations, the maximum and average number of BiCG iterations, the average CPU time (in seconds) for BiCG, and the CPU time (in seconds) for the whole simulation.

3.8.4 Implicit versus semi-implicit time discretization

This section confirms the argument in Section 3.3 that the time step restriction is an essential characteristic of the nature of the problem. In the following, we study the convergence of the nonsmooth Cahn–Hilliard model to the corresponding sharp interface model $\varepsilon \to 0$, which is the Mullins-Sekerka model. For more details, we refer to [72, 131, 42]. Moreover, further convergence tests regarding the other model parameters can be found in [42]. In the following, we compare the solutions obtained by the implicit and semi-implicit nonsmooth Cahn–Hilliard model. Remember, the implicit model involves the time step restriction $\tau < 4\varepsilon^3$, whereas no time step restriction is imposed in the semi-implicit case. The numerical test is an analytic example where the exact solution to the Mullins-Sekerka model is known and can be found, e.g., in [42]. We consider the domain $\Omega = B_1(0) \subset \mathbb{R}^2$, which is a disc around the origin $(0, 0)$ with radius one. The initial state for the Mullins-Sekerka model consists of two circles with radii $r_1 = 0.3$ and $r_2 = 0.15$. The innermost and outermost blank area is initialized with the value 1, and the remaining intermediate area is initialized with the value -1. The time evolution of the exact solution to the Mullins-Sekerka model results in a shrinking of both radii until the smaller one vanishes at time $t_c = 1.85 \cdot 10^{-3}$. Now, we want to examine the quality of the solution of the Cahn–Hilliard model with a small interface parameter. Due to the presence of an interface in the Cahn–Hilliard model, we include an interfacial area between the phases in the initial state. More precisely, the initial state given in Figure 3.11(a) is described

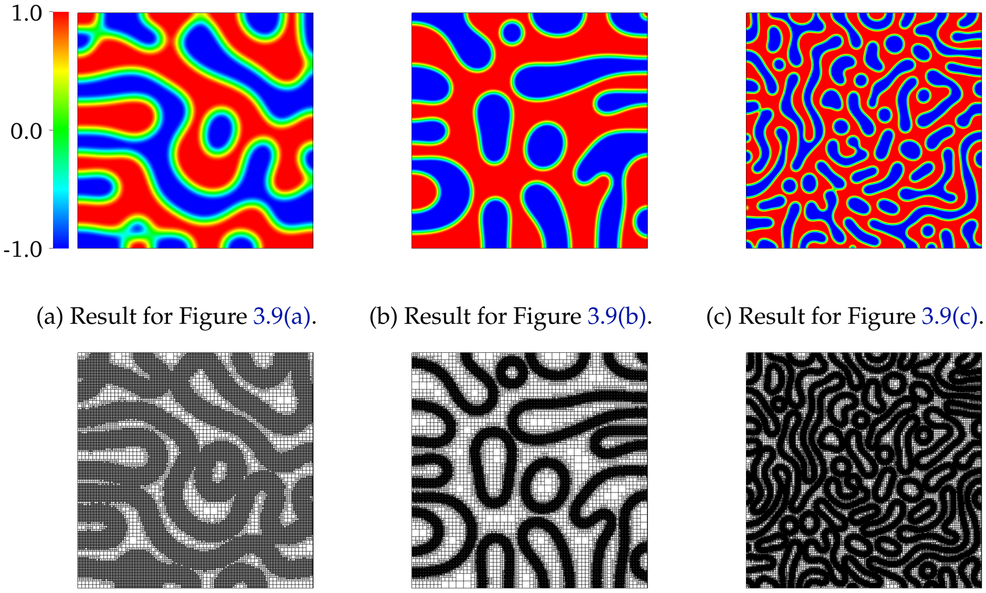


(a) $h^{(0)} = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 2 \cdot 10^{-5}$, $c_{p_{\max}} = 10^{-7}$.

(b) $h^{(0)} = 2^{-8}$, $\varepsilon = 9 \cdot 2^{-8}/\pi$, $\tau = 3.125 \cdot 10^{-6}$, $c_{p_{\max}} = 10^{-7}$.

(c) $h^{(0)} = 2^{-9}$, $\varepsilon = 9 \cdot 2^{-9}/\pi$, $\tau = 4 \cdot 10^{-7}$, $c_{p_{\max}} = 10^{-7}$.

Figure 3.9: Results for the solution of the nonsmooth implicit system (3.117) with the preconditioner (3.118) and the Schur complement approximation (3.119) using adaptive meshes. The x-axis shows the time t, the left y-axis displays the average number of BiCG iterations per SSN step, and the right y-axis illustrates the number of degrees of freedom.



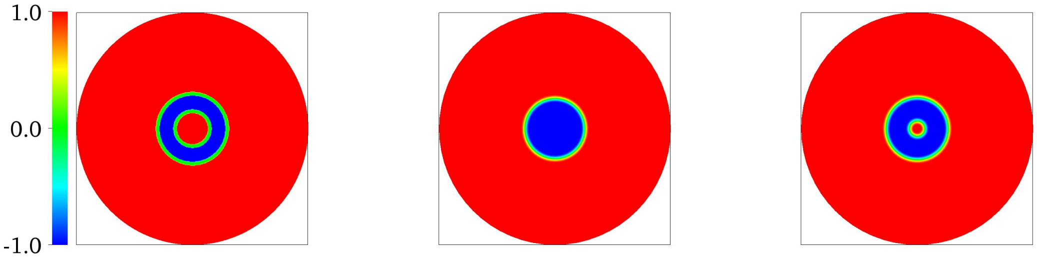
(a) Result for Figure 3.9(a).

(b) Result for Figure 3.9(b).

(c) Result for Figure 3.9(c).

Figure 3.10: Computations with adaptive meshes: The final phase variable with the corresponding spatial mesh for the three simulations in Figure 3.9.

as follows: From the origin, we draw two circles with radii $r_1 = 0.3$ and $r_2 = 0.15$. Each of the two circles forms the median of an interfacial area. More precisely, each of the two interfaces is a circular ring with a width of 1.35ε. We initialize the interface with the value 0. Finally, the innermost and outermost blank area is initialized with the value 1, and the remaining intermediate area is initialized with the value -1. Now, we compare the evolution of the initial state using the implicit and semi-implicit nonsmooth Cahn–Hilliard model. We set $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 1.85 \cdot 10^{-3}/90$ and $T = 1.85 \cdot 10^{-3}$, where $\tau < 4\varepsilon^3$ holds true. Figure 3.11 shows that the smaller radius vanishes later in time with the semi-implicit scheme.



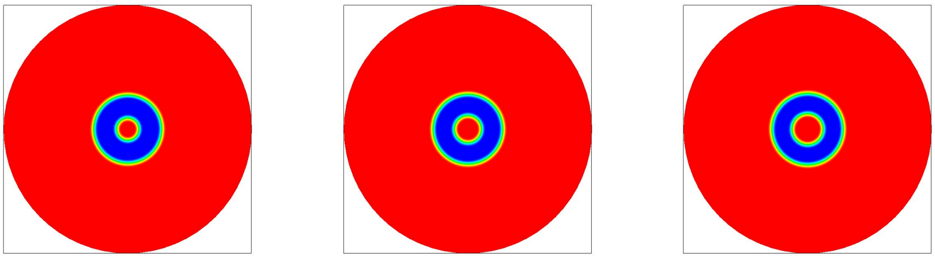
(a) Initial state.

(b) Solution of the nonsmooth implicit Cahn–Hilliard model at time $T = 1.85 \cdot 10^{-3}$.

(c) Solution of the nonsmooth semi-implicit Cahn–Hilliard model at time $T = 1.85 \cdot 10^{-3}$.

Figure 3.11: Results for the analytical example using the nonsmooth implicit and semi-implicit Cahn–Hilliard model.

Now, we explore the evolution of the nonsmooth semi-implicit system for larger values of τ as this scheme has no theoretical time step restrictions. In Figure 3.12, we consider the solutions at time $T = 1.85 \cdot 10^{-3}$ for the time step sizes $\tau = 1.85 \cdot 10^{-3}/40$, $1.85 \cdot 10^{-3}/20$, and $1.85 \cdot 10^{-3}/10$. Note that $\tau > 4\varepsilon^3$ holds for the latter two simulations. Naturally, the approximation error gets larger for large time steps. Hence, the related sharp interface problem is no longer well approximated. We can see that the approximation is crude for larger time steps. Hence, very small time steps are necessary to capture the evolution of the sharp interface model. In conclusion, even though the time step size is allowed to be arbitrary large for the nonsmooth semi-implicit system, the results obtained for large time steps are highly inaccurate for capturing the evolution of the sharp interface limit. The time step restriction is an essential characteristic of the nature of the problem. Note that the same phenomenon was observed for the nonsmooth Allen–Cahn equation; see, e.g., [138].



(a) $\tau = 1.85 \cdot 10^{-3}/40$ (b) $\tau = 1.85 \cdot 10^{-3}/20$ (c) $\tau = 1.85 \cdot 10^{-3}/10$

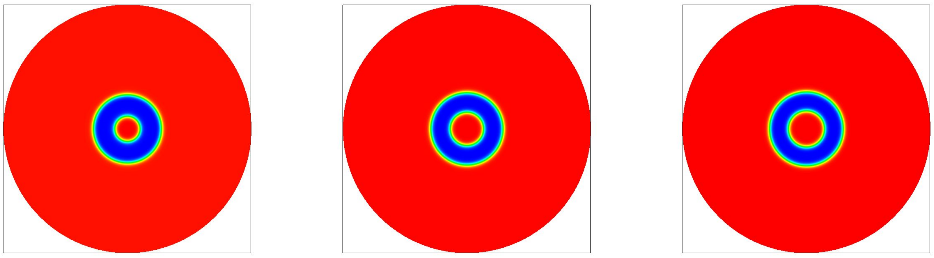
Figure 3.12: Results for the analytical example: Solutions of the nonsmooth semi-implicit Cahn–Hilliard model with different values of the time step size τ at time $T = 1.85 \cdot 10^{-3}$.

The same observation can be made with the smooth semi-implicit Cahn–Hilliard model. The setting is the same as before except for the value of ε, which is now $\varepsilon = 9 \cdot 2^{-7}/(2\sqrt{2} \cdot \operatorname{atanh}(0.95))$. We use this value also for the width of the interface between the circles in the initial state. In Figure 3.13, we consider the solutions at time $T = 1.85 \cdot 10^{-3}$ for the time step sizes $\tau = 1.85 \cdot 10^{-3}/185$, $1.85 \cdot 10^{-3}/18$, and $1.85 \cdot 10^{-3}/4$. Again, we can see that the approximation is crude for larger time steps.

This section has verified our preference for the implicit time discretization scheme despite the inherent time step restriction. This observation also serves as the basis for the next chapter, where we only focus on the implicit time discretization scheme.

3.8.5 Long-time evolution

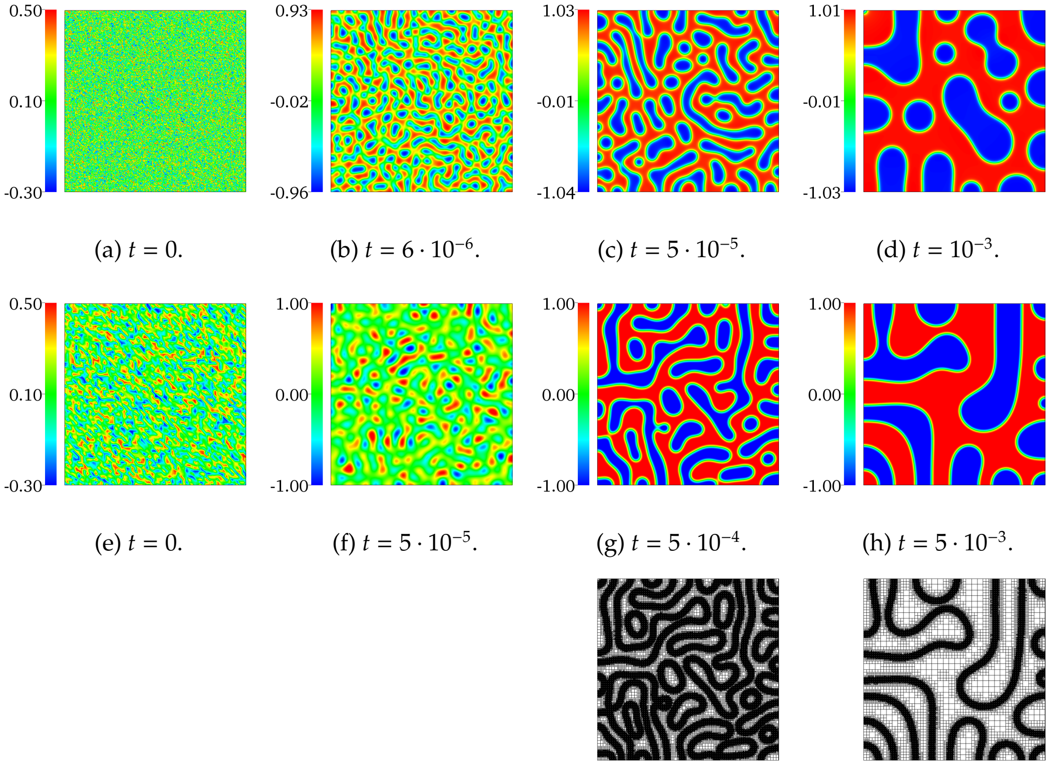
In the following, we consider the long-time evolution of the smooth and nonsmooth implicit Cahn–Hilliard model; see Figure 3.14. In the smooth case, we use the setting $\Omega = [0, 1]^2$, $h = 2^{-8}$, $\varepsilon = 9h/(2\sqrt{2} \cdot \operatorname{atanh}(0.95))$, $\tau = 10^{-7}$, $T = 10^{-3}$. In the nonsmooth



(a) $\tau = 1.85 \cdot 10^{-3}/185$ (b) $\tau = 1.85 \cdot 10^{-3}/18$ (c) $\tau = 1.85 \cdot 10^{-3}/4$

Figure 3.13: Results for the analytical example: Solutions of the smooth semi-implicit Cahn–Hilliard model with different values of the time step size τ at time $T = 1.85 \cdot 10^{-3}$.

case, we use the setting $\Omega = [-1,1]^2$, $n_c = 0$, $h^{(0)} = 2^{-5}$, $\varepsilon = 0.02$, $\tau = 10^{-5}$, $c_{p_{\max}} = 10^{-7}$, $T = 5 \cdot 10^{-3}$, where $h^{(0)}$ denotes the mesh size of the initial uniform mesh.

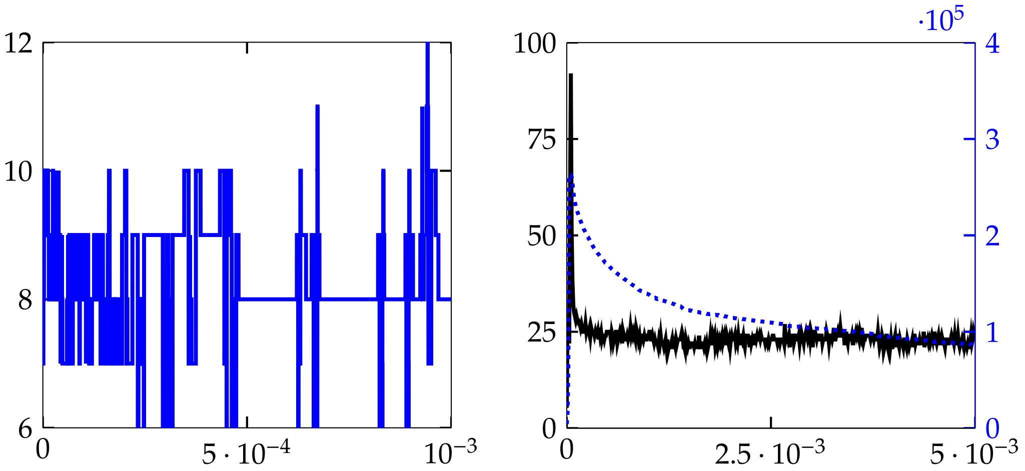


(a) $t = 0$. (b) $t = 6 \cdot 10^{-6}$. (c) $t = 5 \cdot 10^{-5}$. (d) $t = 10^{-3}$.

(e) $t = 0$. (f) $t = 5 \cdot 10^{-5}$. (g) $t = 5 \cdot 10^{-4}$. (h) $t = 5 \cdot 10^{-3}$.

Figure 3.14: Long-time evolution using the smooth (upper row) and nonsmooth (lower two rows) implicit Cahn–Hilliard model.

In Figure 3.15, we illustrate the performance of our preconditioners for the solution of the smooth and nonsmooth implicit system. Both x-axes show the time t. In Figure 3.15(a), the y-axis displays the average number of BiCG iterations per Newton step. In Figure 3.15(b), the left y-axis displays the average number of BiCG iterations



(a) Results for Figure 3.14(a)–3.14(d). $h = 2^{-8}$, $\varepsilon = 9h/(2\sqrt{2} \cdot \operatorname{atanh}(0.95))$, $\tau = 10^{-7}$.

(b) Results for Figure 3.14(e)–3.14(h). $h^{(0)} = 2^{-5}$, $\varepsilon = 0.02$, $\tau = 10^{-5}$, $c_{p_{\max}} = 10^{-7}$.

Figure 3.15: Results for the long-time evolution using the smooth (left) and nonsmooth (right) implicit Cahn–Hilliard model. The x-axis shows the time t. On the left, the y-axis displays the average number of BiCG iterations per Newton step. On the right, the left y-axis displays the average number of BiCG iterations per SSN step and the right y-axis illustrates the number of degrees of freedom. Since we use a uniform mesh for the smooth model, we do not display the number of degrees of freedom on the left.

	Newton/SSN		BiCG			
Figure	Max	Avg	Max	Avg	CPU (s)	CPU (s)
3.14(a)–3.14(d)	3	1	13	8	17	239838
3.14(e)–3.14(h)	7	4	106	23	156	343433

Table 3.5: Results for the long-time evolution: The maximum and average number of Newton/SSN iterations, the maximum and average number of BiCG iterations, the average CPU time (in seconds) for BiCG, and the CPU time (in seconds) for the whole simulation for the smooth and nonsmooth Cahn–Hilliard model, respectively.

per SSN step and the right y-axis illustrates the number of degrees of freedom. Since we use a uniform mesh for the smooth model, we do not display the number of degrees of freedom in Figure 3.15(a). Except for the peak, which occurs as soon as pure phases develop, the iteration numbers stay constantly low. Moreover, with the formation of pure phases, the coarsening process begins and the number of degrees of freedoms decreases. Table 3.5 illustrates the maximum and average number of Newton/SSN iterations, the maximum and average number of BiCG iterations, the average CPU time (in seconds) for BiCG, and the CPU time (in seconds) for the whole simulation for the smooth and nonsmooth case, respectively. Table 3.6 illustrates the minimum and maximum phase values at some time steps. We observe that the concentrations may exceed one and become less than minus one for smooth potentials. However, the overshoots and undershoots are not reported to blow up.

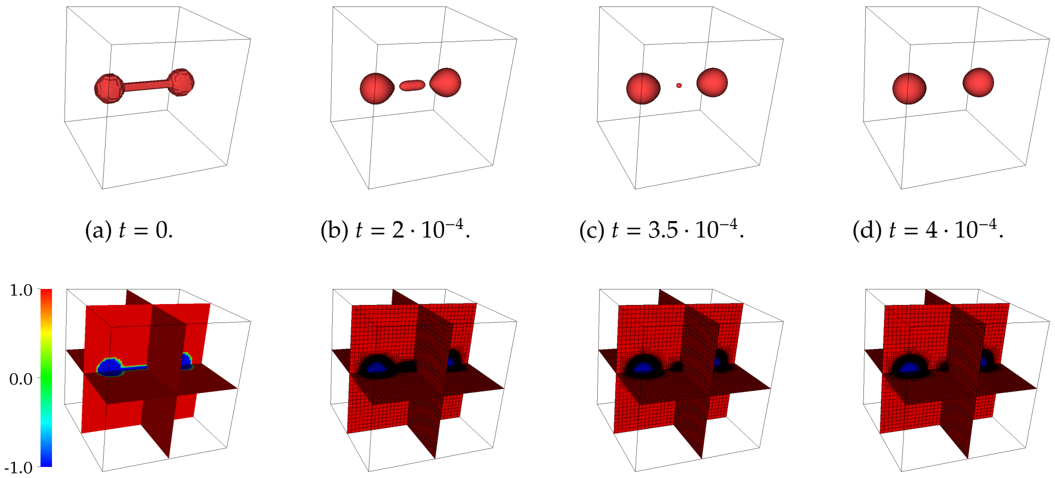
Note that one should not compare the above results in terms of smooth versus nonsmooth. The evolution with smooth and nonsmooth potentials is very different and distinct parameters are used. Moreover, both types of potentials are used in many applications. In some of them, like the deep-quench limit, the nonsmooth potential must be used. In other applications, smooth potentials are preferred and produce satisfactory results. Therefore, the development of efficient solvers is of great interest in both cases.

		t				
value	model	$2 \cdot 10^{-4}$	$4 \cdot 10^{-4}$	$6 \cdot 10^{-4}$	$8 \cdot 10^{-4}$	10^{-3}
min	smooth	−1.03879	−1.03965	−1.03896	−1.03957	−1.03117
	nonsmooth	−1.00001	−1.00001	−1.00001	−1.00001	−1.00001
max	smooth	1.02389	1.03046	1.0179	1.01093	1.00778
	nonsmooth	1.00001	1.00001	1.00001	1.00001	1.00001

Table 3.6: Minimum and maximum phase values during the simulation with the smooth and nonsmooth Cahn–Hilliard model.

3.8.6 Three-dimensional dumbbell

Next, we consider the three dimensional domain $\Omega = [-1, 1]^3$ and choose a dumbbell as initial state. Figure 3.16 shows the evolution for this example using the nonsmooth implicit Cahn–Hilliard model with $n_c = 0$, $h^{(0)} = 2^{-5}$, $\varepsilon = 0.03$, $\tau = 5 \cdot 10^{-5}$, $c_{p_{\max}} = 10^{-5}$, $T = 10^{-3}$, where $h^{(0)}$ denotes the mesh size of the initial uniform mesh. In Figure 3.17, we illustrate the performance of our preconditioner. The x-axis shows the time t, the left y-axis displays the average number of BiCG iterations per SSN step, and the right y-axis illustrates the number of degrees of freedom. The maximum and average number of SSN iterations are 10 and 5. The maximum and average number of BiCG iterations are 77 and 43. The average CPU time for BiCG is 5983s and the CPU time for the whole simulation is 732803s.



(a) $t = 0$. (b) $t = 2 \cdot 10^{-4}$. (c) $t = 3.5 \cdot 10^{-4}$. (d) $t = 4 \cdot 10^{-4}$.

Figure 3.16: Evolution of a dumbbell.

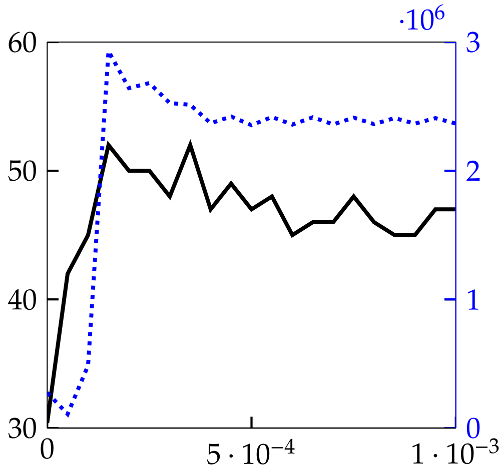


Figure 3.17: Results for the evolution of a dumbbell: The x-axis shows the time t, the left y-axis displays the average number of BiCG iterations per SSN step, and the right y-axis illustrates the number of degrees of freedom.

3.8.7 Two-phase flows

In this section, we apply our preconditioning strategy to the solution of a coupled two-component Cahn–Hilliard/Navier–Stokes system. We consider the diffuse interface model for two-phase flows of two incompressible fluids with different densities, which was introduced by Abels, Garcke, and Grün [1]. It is given as

$$\rho\partial_t \mathbf{v} + ((\rho\mathbf{v} + \mathbf{J}) \cdot \nabla)\,\mathbf{v} - \nabla \cdot (2\eta \mathbf{D}\mathbf{v}) + \nabla p = w\nabla u + \rho\mathbf{g}, \tag{3.121}$$

$$\nabla \cdot \mathbf{v} = 0, \tag{3.122}$$

$$\partial_t u + \mathbf{v} \cdot \nabla u - \nabla \cdot (L(u)\nabla w) = 0, \tag{3.123}$$

$$-\sigma\varepsilon\Delta u + \psi'(u) - w = 0, \tag{3.124}$$

$$\mathbf{v} = \mathbf{0} \quad \text{on } \partial\Omega, \tag{3.125}$$

$$\nabla u \cdot \mathbf{n} = L(u)\nabla w \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega, \tag{3.126}$$

together with the initial conditions $\mathbf{v}(0,\mathbf{x}) = \mathbf{v}_0(\mathbf{x})$ and $u(0,\mathbf{x}) = u_0(\mathbf{x})$ for all $\mathbf{x} \in \Omega \subset \mathbb{R}^d$ with $d \in \{2,3\}$. Consistent with this chapter, u denotes the phase variable, w the chemical potential, ε the interfacial parameter, ψ the potential function, and $L(u)$ the mobility. Moreover, we denote the two fluids by A and B. We indicate the specific (constant) density of the pure fluid A by $\tilde{\rho}_A$ and of the pure fluid B by $\tilde{\rho}_B$. Without loss of generality, we assume $0 < \tilde{\rho}_A \leq \tilde{\rho}_B$. The mean density ρ is given as

$$\rho = \rho(u) = \frac{\tilde{\rho}_B - \tilde{\rho}_A}{2}u + \frac{\tilde{\rho}_A + \tilde{\rho}_B}{2}.$$

We indicate the specific (constant) viscosity of the pure fluid A by $\tilde{\eta}_A$ and of the pure fluid B by $\tilde{\eta}_B$. The viscosity of the mixture is denoted by $\eta = \eta(u) > 0$ and fulfills $\eta(-1) = \tilde{\eta}_A$ and $\eta(1) = \tilde{\eta}_B$. Further, $\mathbf{v}$ is the volume averaged velocity, p the pressure, $\mathbf{g}$ the gravitational force, σ the scaled surface tension, $\mathbf{J} = -\frac{\mathrm{d}\rho}{\mathrm{d}u}L(u)\nabla w$, and $\mathbf{D}\mathbf{v} = \frac{1}{2}\left(\nabla\mathbf{v} + (\nabla\mathbf{v})^T\right)$ denotes the symmetrized gradient. The above model by Abels, Garcke, and Grün couples the Navier–Stokes Equations (3.121)–(3.122) to the Cahn–Hilliard system (3.123)–(3.124) in a thermodynamically consistent way, i.e., an energy inequality holds. Moreover, as mentioned above, their model incorporates the consideration of different densities $\tilde{\rho}_A$ and $\tilde{\rho}_B$.

Recently, Garcke, Hinze, and Kahle [73] developed a special time and space discretization scheme for the model (3.121)–(3.126), which conserves the energy inequality. They considered nondegenerate mobilities $L(u) > 0$ and a Moreau–Yosida regularization of the double-obstacle potential function in the form of

$$\psi = \psi_c(u) = \frac{\sigma}{2\varepsilon}\left(1 - u^2 + \frac{1}{c}\left[\max(0, u-1) + \min(0, u+1)\right]^2\right).$$

As before, $0 < c \ll 1$ denotes the regularization parameter. For the rest of this section, we consider constant mobilities $L = L(u)$. The space discretization in [73] is designed as an adaptive finite-element algorithm based on an a-posteriori error analysis. In the following, we briefly summarize the arising fully discrete setting. Let $\{\mathcal{T}_h\}_{h>0}$ be a triangulation of Ω into disjoint open triangular elements. Garcke et al. define the finite-dimensional spaces

$$\begin{aligned}
S_{h,1} &:= \{\phi \in C(\overline{\Omega})\colon \phi\,|_T \in P_1(T) \ \ \forall T \in \mathcal{T}_h\} =: \operatorname{span}\{\varphi_1^i\}_{i=1}^{m_1},\\
S_{h,2} &:= \{\boldsymbol{\phi} \in C(\overline{\Omega})^d\colon \boldsymbol{\phi}\,|_T \in P_2(T)^d \ \ \forall T \in \mathcal{T}_h,\ \boldsymbol{\phi}\,|_{\partial\Omega} = 0\} =: \operatorname{span}\{\boldsymbol{\varphi}_2^i\}_{i=1}^{m_2},
\end{aligned}$$

where $P_l(T)$ denotes the space of polynomials up to order l defined on T. In each time step, the corresponding Cahn–Hilliard/Navier–Stokes system is solved by an SSN method. As before, $n-1$ denotes the previous time step, τ the time step size, and k the previous SSN step. The coefficient matrix of the arising fully discrete linear scheme is given as

$$\mathcal{A} = \left[\begin{array}{c|c} \boldsymbol{A}_{\text{NS}} & \boldsymbol{C}_I \\ \hline \boldsymbol{C}_T & \boldsymbol{A}_{\text{CH}} \end{array}\right] = \left[\begin{array}{cc|cc} \boldsymbol{F} & \boldsymbol{B}^T & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{B} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{M}_1 & -\sigma\varepsilon \boldsymbol{K}_1 - \sigma\varepsilon^{-1}c^{-1}\boldsymbol{G} \\ \tau\boldsymbol{T} & \boldsymbol{0} & \tau L\boldsymbol{K}_1 & \boldsymbol{M}_1 \end{array}\right], \tag{3.127}$$

where the matrices are defined as

$$\begin{aligned}
\boldsymbol{F} &= \boldsymbol{M}_2 + \boldsymbol{T}_a + \boldsymbol{K}_2 \in \mathbb{R}^{2m_2\times 2m_2},\\
[\boldsymbol{M}_2]_{ij} &= \left(\frac{\rho^{(n-1)} + \rho^{(n-2)}}{2\tau}\boldsymbol{\varphi}_2^j, \boldsymbol{\varphi}_2^i\right),\\
[\boldsymbol{T}_a]_{ij} &= a\left(\rho^{(n-1)}\boldsymbol{v}^{(n-1)} + \boldsymbol{J}^{(n-1)}, \boldsymbol{\varphi}_2^j, \boldsymbol{\varphi}_2^i\right),\\
a(\mathbf{u},\mathbf{v},\mathbf{w}) &= \frac{1}{2}\int_\Omega ((\mathbf{u}\cdot\nabla)\,\mathbf{v})\,\mathbf{w}\,\mathrm{d}\mathbf{x} - \frac{1}{2}\int_\Omega ((\mathbf{u}\cdot\nabla)\,\mathbf{w})\,\mathbf{v}\,\mathrm{d}\mathbf{x},\\
[\boldsymbol{K}_2]_{ij} &= (2\eta^{(n-1)}\,\mathbf{D}\boldsymbol{\varphi}_2^j, \nabla\boldsymbol{\varphi}_2^i),\\
[\boldsymbol{B}]_{ij} &= -(\nabla\cdot\boldsymbol{\varphi}_2^j, \varphi_1^i),\ \boldsymbol{B} \in \mathbb{R}^{m_1\times 2m_2},\\
[\boldsymbol{I}]_{ij} &= -(\varphi_1^j\,\nabla u^{(n-1)}, \boldsymbol{\varphi}_2^i),\ \boldsymbol{I} \in \mathbb{R}^{2m_2\times m_1},\\
[\boldsymbol{T}]_{ij} &= -(\boldsymbol{\varphi}_2^j u^{(n-1)}, \nabla\varphi_1^i),\ \boldsymbol{T} \in \mathbb{R}^{m_1\times 2m_2},\\
[\boldsymbol{M}_1]_{ij} &= (\varphi_1^j, \varphi_1^i),\ \boldsymbol{M}_1 \in \mathbb{R}^{m_1\times m_1},\\
[\boldsymbol{K}_1]_{ij} &= (\nabla\varphi_1^j, \nabla\varphi_1^i),\ \boldsymbol{K}_1 \in \mathbb{R}^{m_1\times m_1},
\end{aligned}$$

$$[\boldsymbol{G}]_{ij} = (\chi_{\mathcal{M}}(u^{(k)})\varphi_1^j, \varphi_1^i),\ \boldsymbol{G} \in \mathbb{R}^{m_1 \times m_1},$$
$$\chi_{\mathcal{M}}(u^{(k)}) = \begin{cases} 1 & |u^{(k)}(\mathbf{x})| > 1, \\ 0 & \text{otherwise.} \end{cases}$$

Note that the mass matrices are not lumped here. The blocks $\boldsymbol{A}_{\mathrm{NS}}$ and $\boldsymbol{A}_{\mathrm{CH}}$ are the discrete realizations of the linearized Navier–Stokes and Cahn–Hilliard system, respectively. Their coupling is represented by $\boldsymbol{C}_I$, the coupling through the interfacial force, and $\boldsymbol{C}_T$, the coupling through the transport at the interface. Garcke et al. solve the system $\boldsymbol{\mathcal{A}}\boldsymbol{z} = \boldsymbol{b}$ by preconditioned GMRES with a restart after 10 iterations. They use the block diagonal preconditioner

$$\boldsymbol{\mathcal{P}} = \begin{bmatrix} \hat{\boldsymbol{A}}_{\mathrm{NS}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}_{\mathrm{CH}} \end{bmatrix}. \tag{3.128}$$

The (2,2) block $\boldsymbol{A}_{\mathrm{CH}}$ is inverted by LU decomposition. The (1,1) block $\hat{\boldsymbol{A}}_{\mathrm{NS}}$ is an upper block triangular preconditioner of the form

$$\hat{\boldsymbol{A}}_{\mathrm{NS}} = \begin{bmatrix} \hat{\boldsymbol{F}} & \boldsymbol{B}^T \\ \boldsymbol{0} & \hat{\boldsymbol{S}}_{\mathrm{NS}} \end{bmatrix}. \tag{3.129}$$

$\hat{\boldsymbol{F}}$ is composed of the diagonal blocks of $\boldsymbol{F}$ and is inverted by LU decomposition. $\hat{\boldsymbol{S}}_{\mathrm{NS}}$ is an approximation of the exact Schur complement $\boldsymbol{S}_{\mathrm{NS}} = -\boldsymbol{B}\boldsymbol{F}^{-1}\boldsymbol{B}^T$ of the Navier–Stokes system. Garcke et al. use

$$\hat{\boldsymbol{S}}_{\mathrm{NS}} = -\boldsymbol{K}_p\boldsymbol{F}_p^{-1}\boldsymbol{M}_p,$$

where $\boldsymbol{F}_p$ is the representation of $\boldsymbol{F}$ on the pressure space, $\boldsymbol{K}_p$ the pressure Laplacian matrix, and $\boldsymbol{M}_p$ and the pressure mass matrix. This Schur complement approximation was proposed, e.g., in [99, 63].

Now, our contribution concerns the fully iterative solution of the system $\boldsymbol{\mathcal{A}}\boldsymbol{z} = \boldsymbol{b}$. This is based on the preconditioning techniques we have developed for the nonsmooth Cahn–Hilliard system in Section 3.7 together with the above methods that have been developed for the Navier–Stokes equations. We propose the upper block triangular preconditioner

$$\boldsymbol{\mathcal{P}}_{\mathrm{out}} = \begin{bmatrix} \hat{\boldsymbol{A}}_{\mathrm{NS}} & \boldsymbol{C}_I \\ \boldsymbol{0} & \hat{\boldsymbol{S}} \end{bmatrix}, \tag{3.130}$$

where $\hat{\boldsymbol{S}}$ is an approximation of the exact Schur complement $\boldsymbol{S} = \boldsymbol{A}_{\mathrm{CH}} - \boldsymbol{C}_T\boldsymbol{A}_{\mathrm{NS}}^{-1}\boldsymbol{C}_I$ of the whole system. We choose $\hat{\boldsymbol{A}}_{\mathrm{NS}}$ as

$$\hat{\boldsymbol{A}}_{\mathrm{NS}} = \begin{bmatrix} \hat{\boldsymbol{F}} & \boldsymbol{B}^T \\ \boldsymbol{0} & -\hat{\boldsymbol{S}}_{\mathrm{NS}} \end{bmatrix}. \tag{3.131}$$

As above, $\hat{\boldsymbol{F}}$ is composed of the diagonal blocks of $\boldsymbol{F}$. We perform the action of its inverse via an AMG. $\hat{\boldsymbol{S}}_{\mathrm{NS}}$ is given as above. The action of the inverse of $\boldsymbol{M}_p$ and of $\boldsymbol{K}_p$ are performed with an AMG each. Now, let us consider the Schur complement $\boldsymbol{S}$, which can be approximated as

$$\boldsymbol{S} \approx \boldsymbol{A}_{\mathrm{CH}} - \boldsymbol{C}_T\hat{\boldsymbol{A}}_{\mathrm{NS}}^{-1}\boldsymbol{C}_I = \begin{bmatrix} \boldsymbol{M}_1 & -\sigma\varepsilon\boldsymbol{K}_1 - \sigma\varepsilon^{-1}c^{-1}\boldsymbol{G} \\ \tau L\boldsymbol{K}_1 - \tau\boldsymbol{T}\hat{\boldsymbol{F}}^{-1}\boldsymbol{I} & \boldsymbol{M}_1 \end{bmatrix} =: \tilde{\boldsymbol{S}}.$$

We propose to approximate the action of the inverse of $\boldsymbol{S}$ by a preconditioned GMRES iteration applied to the system of the form $\tilde{\boldsymbol{S}}\boldsymbol{y} = \boldsymbol{f}$. We call this iteration the inner iteration. As preconditioner for the inner iteration, we suggest the upper block triangular preconditioner

$$\mathcal{P}_{\text{in}} = \begin{bmatrix} \boldsymbol{M}_1 & -\sigma\varepsilon\boldsymbol{K}_1 - \sigma\varepsilon^{-1}c^{-1}\boldsymbol{G} \\ \boldsymbol{0} & -\hat{\boldsymbol{S}}_{CH} \end{bmatrix}. \tag{3.132}$$

$\hat{\boldsymbol{S}}_{CH}$ is an approximation of the exact Schur complement

$$\boldsymbol{S}_{CH} = \boldsymbol{M}_1 + \tau\left(L\boldsymbol{K}_1 - \boldsymbol{T}\hat{\boldsymbol{F}}^{-1}\boldsymbol{I}\right)\boldsymbol{M}_1^{-1}\left(\sigma\varepsilon\boldsymbol{K}_1 + \sigma\varepsilon^{-1}c^{-1}\boldsymbol{G}\right)$$

of $\tilde{\boldsymbol{S}}$. We design $\hat{\boldsymbol{S}}_{CH}$ as

$$\begin{aligned}\hat{\boldsymbol{S}}_{CH} &= \boldsymbol{S}_1\boldsymbol{M}_1^{-1}\boldsymbol{S}_2 \\ &= \left(\boldsymbol{M}_1 + \sqrt{\tau\sigma L}\boldsymbol{K}_1\right)\boldsymbol{M}_1^{-1}\left(\boldsymbol{M}_1 + \sqrt{\tau L\sigma^{-1}}\left[\sigma\varepsilon\boldsymbol{K}_1 + \sigma\varepsilon^{-1}c^{-1}\boldsymbol{G}\right]\right) \\ &= \boldsymbol{M}_1 + \tau L\boldsymbol{K}_1\boldsymbol{M}_1^{-1}\left(\sigma\varepsilon\boldsymbol{K}_1 + \sigma\varepsilon^{-1}c^{-1}\boldsymbol{G}\right) + \sqrt{\tau L\sigma}\boldsymbol{K}_1 \\ &\quad + \sqrt{\tau L\sigma}\left(\varepsilon\boldsymbol{K}_1 + \varepsilon^{-1}c^{-1}\boldsymbol{G}\right).\end{aligned} \tag{3.133}$$

The first term in (3.133) matches the first term in the exact Schur complement. The second term in (3.133) approximates the second term in the exact Schur complement. Due to the factor $\sqrt{\tau L\sigma}$, the influence of both remainder terms in (3.133) is reduced. The action of the inverse of $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ is performed with an AMG each.

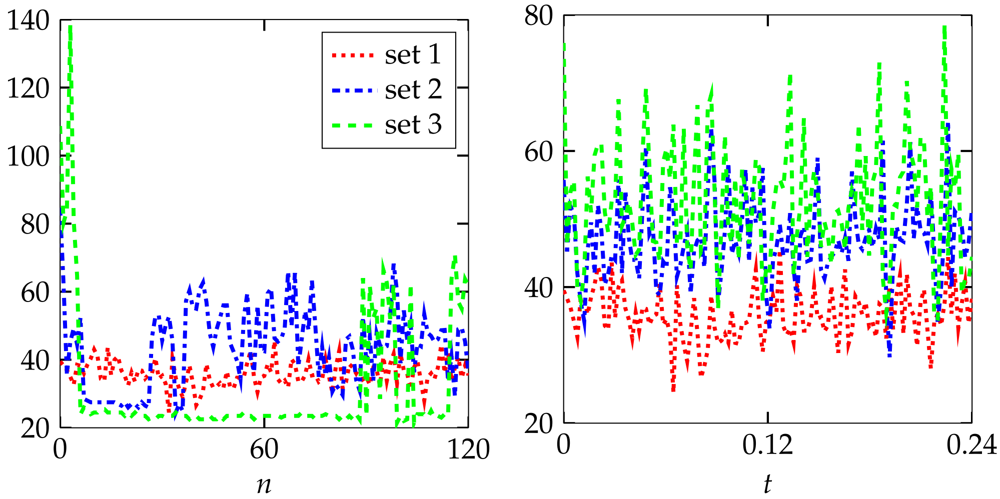
As we use GMRES as inner iteration, we apply the flexible generalized minimum residual method (FGMRES) as outer iteration. FGMRES is a variant of GMRES and was introduced by Saad [135]. This method allows changes in the preconditioner at every step. In contrast to the rest of this thesis, we apply right preconditioning here. Note that a left preconditioner modifies the right-hand side, whereas a right preconditioner does not modify it. As stated in [46, p. 4], a major hurdle for developing variable preconditioners for left preconditioning is the disconnection between the preconditioned residuals and the actual residuals.

We set the GMRES tolerance to be 10^{-1} for the preconditioned relative residual. We use FGMRES with the tolerance $\min(10^{-4}/\|\boldsymbol{b}\|, 10^{-4})$ and a restart after 30 iterations. The solution of the linear system $\mathcal{A}\boldsymbol{z} = \boldsymbol{b}$ is executed in MATLAB® R2012a on a 32-bit server with CPU type Intel® Core™ E6850 @3.00 GHz with 2 CPUs. Note that Garcke et al. [73] implemented the whole numerical simulation in C++, and I would like to thank Christian Kahle for providing his code. We use the MATLAB Engine API in order to call MATLAB® from C++.

In the following, we illustrate the performance of our proposed preconditioner with respect to the mesh size and the Reynolds number. Further numerical tests are a topic of ongoing research. As test example, we use a quantitative benchmark for rising bubble dynamics; see [96]. A simulation is illustrated in Figure 1.8. The setup is described as follows; see also [73, p. 168]. The spatial domain is $\Omega = (0, 1)\times(0, 2)$ with no-slip boundary conditions for the velocity field on the top and bottom wall and free-slip boundary conditions on the left and right wall. The initial state consists of a bubble of radius $r = 0.25$ centered at the spatial point $(0.5, 0.5)$. The initial velocity is zero. The fixed parameters in our experiments are given as $\sigma = 15.5972$, $\tilde{\rho}_2 = 100$, $\tilde{\eta}_1 = 10$, $\tilde{\eta}_2 = 1$, $\mathbf{g} = [0, -0.98]^T$, $c = 10^{-6}$. In Figure 3.18(a), we simultaneously vary the mesh size via refinements of the initial spatial mesh $\{\mathcal{T}_h\}^{(0)}_{h>0}$, the interfacial parameter ε, the time step size τ, as well as the mobility L. In Figure 3.18(b), we vary the Reynolds number via increasing the density $\tilde{\rho}_1$. In our experiments, the Reynolds number is given as

$$Re = \frac{0.35\,\tilde{\rho}_1}{\tilde{\eta}_1}.$$

Figure 3.18 illustrates the promising behavior when the Reynolds number is increased as well as when the parameter set $\left(\{\mathcal{T}_h\}^{(0)}_{h>0}, \varepsilon, \tau, L\right)$ is refined as a whole, which is the practical procedure.



(a) set 1: $m_1^{(0)} = 6599$, $m_2^{(0)} = 26213$, $\varepsilon = 0.04$, $\tau = 2 \cdot 10^{-3}$, $L = 4 \cdot 10^{-5}$. set 2: $m_1^{(0)} = 10399$, $m_2^{(0)} = 41413$, $\varepsilon = 0.02$, $\tau = 5 \cdot 10^{-4}$, $L = 2 \cdot 10^{-5}$. set 3: $m_1^{(0)} = 17831$, $m_2^{(0)} = 71141$, $\varepsilon = 0.01$, $\tau = 1.25 \cdot 10^{-4}$, $L = 10^{-5}$. We fix Re = 35.

(b) $m_1^{(0)} = 6599$, $m_2^{(0)} = 26213$, $\varepsilon = 0.04$, $\tau = 2 \cdot 10^{-3}$, $L = 4 \cdot 10^{-5}$, and Re = 35 (······), Re = 350 (-·-·-), Re = 700 (- - -).

Figure 3.18: The average number of FGMRES iterations per SSN step for the solution of a coupled two-component Cahn–Hilliard/Navier–Stokes system. Here, $m_1^{(0)}$ and $m_2^{(0)}$ denote the initial numbers of degrees of freedom. Note that the numbers of degrees of freedom during every simulation stay in the range of $m_1^{(0)}$ and $m_2^{(0)}$, respectively.

3.9 Existing solvers

In this section, we briefly discuss existing solution methods for the nonsmooth Cahn–Hilliard equation.

Traditional iterative solvers such as (nonlinear) Gauss-Seidel have been used for the smooth and nonsmooth case, but these suffer from deteriorating convergence rates for increasing refinements [57, 14, 120].

Gräser and Kornhuber [83] proposed a preconditioned Uzawa iteration method for the saddle point formulation of the discrete (semi-implicit in time) Cahn–Hilliard system with an obstacle potential. The method proceeds as follows. Consider the kth Uzawa iteration, which consists of two sub-steps: First, an elliptic obstacle problem with box constraints needs to be solved in order to obtain $u^{(k)}$ and the appropriate coincidence set

$$\mathcal{N}_h^*(u^{(k)}) = \{p \in J_h : |u^{(k)}(p)| = 1\}.$$

This step includes the direct work with the inverse $(\boldsymbol{A} + \partial I_{\mathcal{K}_h})^{-1}$, where $\mathcal{K}_h = \mathcal{K} \cap S_h$, $\partial I_{\mathcal{K}_h}$ is the subdifferential of the indicator function of $\mathcal{K}_h$, and $\langle \boldsymbol{A}u, v\rangle = \varepsilon(\nabla u, \nabla v) + \varepsilon\langle u, 1\rangle\langle v, 1\rangle \ \forall v \in S_h$. In order to solve this, Gräser and Kornhuber apply a monotone multigrid method [107, 108], which can be stopped after a finite (usually quite moderate) number of steps. With the help of the coincidence set, the second substep reduces to a linear saddle point problem, which has to be solved for obtaining $w^{(k+1)}$. In order to solve this problem, Gräser and Kornhuber apply a multigrid method with a block Gauss-Seidel smoother. Once the exact coincidence set $\mathcal{N}_h^*(u)$ is detected, the Uzawa iteration provides the exact solution (for the considered time step). The solution of the second substep in the Uzawa iteration forms the main computational cost. About 15 multigrid steps are necessary to solve this problem to machine accuracy, and the computational cost for each spatial problem is obtained approximately by multiplying that number with the number of Uzawa steps. The first substep uses 4 to 7 monotone multigrid steps, which plays a minor role considering the CPU time.

Baňas and Nürnberg [7] proposed a fully nonlinear multigrid method for the discrete Cahn–Hilliard problem. For a sequence of triangulations $\mathcal{T}_k$, the algorithm consists of alternating pre-smoothing steps for the approximate solution by projected Gauss-Seidel iterations [14] and restrictions to the next coarser grid. On the coarsest triangulation, the appropriate system is solved exactly. By prolongation to the next finer grid, the solution is updated together with post-smoothing steps by projected Gauss-Seidel iterations. In practice, this method exhibits mesh-independent convergence properties for arbitrary time steps in 2D and 3D and also for a small interfacial parameter ε. However, Baňas and Nürnberg are unable to prove convergence of the multigrid solver (except for the case, when the discrete Cahn–Hilliard system reduces to a linear problem, which occurs when $|u^{(n)}| < 1$). The same holds for methods including a smooth potential; see [101, 105].

An important point for the future is a comparison with our approach.

3.10 Conclusions

In this chapter, we have investigated the numerical solution of the two-component Cahn–Hilliard model. We have considered smooth and nonsmooth potentials with a focus on the latter. For the discretization in time, we have used a fully implicit scheme. This is due to accuracy reasons and the numerical examples justify this choice. Regarding the smooth setting, we have proved the energy stability and uniqueness of the solution of the time-discrete scheme under reasonable assumptions. Concerning the nonsmooth framework, following [91], we have extended the analysis from the semi-implicit to the implicit time-discrete case in terms of three aspects: First, we have shown that the time-discrete problem is equivalent to an optimal control problem with pointwise constraints on the control. Second, we have handled the control constraints by a Moreau–Yosida regularization. Third, we have applied a local superlinear SSN method for solving the optimality systems of the resulting subproblems. For the discretization in space, we have used classical FEM for both systems, the smooth and regularized nonsmooth one. At the heart of our method lies the solution of large and sparse systems of linear equations of saddle point form. We have introduced and studied block diagonal and block-triangular preconditioners using efficient and cheap Schur complement approximations. For these approximations, we have used multilevel techniques, algebraic multigrid in our case. Furthermore, we have designed preconditioners for the linear systems arising from a semi-implicit time-discrete scheme. For the smooth systems, we have derived optimal preconditioners, which are proven to be robust with respect to crucial model parameters. For the nonsmooth systems, extensive numerical experiments show a nearly parameter independent behavior of our developed preconditioners. Additionally, we have implemented a simple adaptive spatial mesh refinement approach, which reduces the number of degrees of freedoms. Together with our preconditioners, this allows us to perform three-dimensional experiments in an efficient way. As another application, we have applied our preconditioner to a coupled Cahn–Hilliard/Navier–Stokes system equipped with a nonsmooth potential. The numerical results illustrate the efficiency of our approach.

Chapter 4

Vector-Valued Cahn–Hilliard Equations

4.1 Introduction

In the last chapter, we have studied two-component systems. In practice, often more than two phases occur; see, e.g., [123, 66, 60, 59, 22, 100, 75], and the phase field model has been extended to deal with multi-component systems. Instead of two phases, we consider $N > 2$ components now. Imagine a molten N-component alloy inside a bounded spatial domain $\Omega \subset \mathbb{R}^d$ with $d \in \{1, 2, 3\}$. The N pure phases are denoted by A_i for $i = 1, \dots, N$. As before, we are interested in the evolution of the N components or their mixture in the period $(0, T)$ with a fixed time $T > 0$. In the case of two phases, we could describe their local concentrations via a single scalar phase variable. However, this is not possible anymore in the case of $N > 2$ phases. Instead, a vector-valued phase variable $\mathbf{u} = [u_1, \dots, u_N]^T \colon \Omega \times (0, T) \to \mathbb{R}^N$ is introduced. Here, u_i describes the concentration of phase A_i for $i = 1, \dots, N$. If $u_i(\mathbf{x}, t) \approx 1$, then only phase A_i (the pure phase A_i) is present at point $\mathbf{x}$ at time t. The case $u_i(\mathbf{x}, t) \approx 0$ means phase A_i is absent at point $\mathbf{x}$ at time t. Values of u_i between 0 and 1 represent mixed regions. In particular, these regions include the interfacial area. Here, the interface is a small boundary layer that separates the pure phases $A_i, i = 1, \dots, N$, from each other. As in the previous chapter, it acts as a diffuse phase transition and we can control its width via the model parameter $\varepsilon > 0$. For the limit case $\varepsilon \downarrow 0$, which gives the sharp interface model, we refer to, e.g., [74, 72, 131]. Due to the model properties, it holds

$$\sum_{i=1}^{N} u_i = 1 \tag{4.1}$$

and $u_i \geq 0$ for $i = 1, \dots, N$, so that admissible states belong to the Gibbs simplex

$$\mathcal{G}^N := \left\{ \mathbf{v} = [v_1, \dots, v_N]^T \in \mathbb{R}^N \colon \sum_{i=1}^{N} v_i = 1,\ v_i \geq 0 \text{ for } i = 1, \dots, N \right\}. \tag{4.2}$$

The motion of the interfaces separating N components can be modeled with the Ginzburg–Landau energy. The energy (3.1) for two components generalizes to

$$\mathcal{E}(\mathbf{u}) = \int_\Omega \frac{\varepsilon^2}{2} \sum_{i=1}^{N} |\nabla u_i|^2 + \psi(\mathbf{u}) \; \mathrm{d}\mathbf{x} \tag{4.3}$$

for $N > 2$ components. An equilibrium profile of our considered mixture minimizes the Ginzburg–Landau energy (4.3) subject to the mass conservation

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_\Omega u_i \; \mathrm{d}\mathbf{x} = 0, \quad i = 1, \dots, N.$$

The parameter $\varepsilon > 0$ is proportional to the thickness of the interfacial region as mentioned above. The first part of (4.3) is large whenever u_i changes rapidly for some $i \in \{1, \dots, N\}$. Hence, its minimization gives rise to the interfacial area. The potential function $\psi\colon \mathbb{R}^N \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ in (4.3) gives rise to phase separation. It has N distinct minima, one for each pure phase A_i. As in the last chapter, we consider potential functions of polynomial and obstacle type with a main focus on the latter. The former is the smooth multi-well potential; see, e.g., [57]. It is an extension of the double-well potential (3.2) to N components and is given as

$$\psi_{\mathrm{pol}}(\mathbf{u}) = \frac{1}{4} \sum_{i=1}^{N} u_i^2 (1 - u_i)^2. \tag{4.4}$$

Following [119], the interfacial equilibrium profile in one space dimension can be described by $\hat{u}_1(x) = \frac{1}{2}\left(1 + \tanh\left(\frac{x}{2\sqrt{2}\varepsilon}\right)\right)$. Let us describe the interface thickness as the distance between x_1 and x_2 with $u_1(x_1) = 0.05$ and $u_1(x_2) = 0.95$. Then, we can express the equilibrium thickness via ε by

$$0.95 = \frac{1}{2}\left(1 + \tanh\left(\frac{x_2}{2\sqrt{2}\varepsilon}\right)\right),$$

which is equivalent to

$$x_2 = 2\sqrt{2}\varepsilon \cdot \operatorname{atanh}(0.9).$$

Similar, we obtain

$$x_1 = -2\sqrt{2}\varepsilon \cdot \operatorname{atanh}(0.9).$$

Hence, the equilibrium interfacial thickness is given by $4\sqrt{2}\varepsilon \cdot \operatorname{atanh}(0.9)$ using a polynomial potential. As in the previous chapter, we want to have eight or nine grid points across the interface transition. If we denote by h the spatial mesh size across the interface, this leads to the condition

$$h \leq \frac{4\sqrt{2}\varepsilon \cdot \operatorname{atanh}(0.9)}{9} \approx 0.9253\,\varepsilon.$$

The second potential is the nonsmooth multi-obstacle potential; see, e.g., [12]. It is an extension of the double-obstacle potential (3.3) to N components and is given as

$$\psi_{\mathrm{obs}}(\mathbf{u}) = \begin{cases} \psi_0(\mathbf{u}) = -\frac{1}{2}\mathbf{u} \cdot \boldsymbol{T}\mathbf{u} & \mathbf{u} \in \mathcal{G}^N, \\ \infty & \text{otherwise.} \end{cases} \tag{4.5}$$

Here, $\boldsymbol{T} \in \mathbb{R}^{N\times N}$ is a symmetric matrix, which contains constant interaction parameters $[\boldsymbol{T}]_{ij}$. From physical considerations, $\boldsymbol{T}$ must have at least one positive eigenvalue. During the rest of this chapter, we denote by $\lambda_{\max}(\boldsymbol{T})$ the largest positive eigenvalue of $\boldsymbol{T}$. A typical choice is $\boldsymbol{T} = \boldsymbol{I} - \boldsymbol{1}\boldsymbol{1}^T$ with $\boldsymbol{1} = [1,\dots,1]^T \in \mathbb{R}^N$ and the identity matrix $\boldsymbol{I} \in \mathbb{R}^{N\times N}$, which means that the interaction between all different components is equal and no self-interaction occurs. As in the last chapter, we choose

$$h \leq \frac{\varepsilon\pi}{9} \approx 0.3491\,\varepsilon$$

in order to have at least eight or nine grid points across the interface transition.

For logarithmic potentials we refer to, e.g., [11]. Using a polynomial potential, the Cahn–Hilliard model results in a system of time-dependent, nonlinear PDEs. As in the last chapter, we will name this formulation the *smooth system*. Using an obstacle potential, the Cahn–Hilliard model results in a system of variational inequalities. We will call this formulation the *nonsmooth system*.

As we will show in the course of this chapter, the solution of linear systems $\boldsymbol{\mathcal{A}}\boldsymbol{z} = \boldsymbol{b}$ with a large and sparse matrix $\boldsymbol{\mathcal{A}}$ is at the heart of our method. They have the following saddle point structure

$$\boldsymbol{\mathcal{A}} = \begin{bmatrix} -\boldsymbol{A} & \boldsymbol{I}\otimes\boldsymbol{M} \\ \boldsymbol{I}\otimes\boldsymbol{M} & \boldsymbol{L}\otimes\boldsymbol{K} \end{bmatrix}$$

with $\boldsymbol{I} \in \mathbb{R}^{N\times N}$ being the identity matrix, $\boldsymbol{M} \in \mathbb{R}^{m\times m}$ being symmetric positive definite, $\boldsymbol{K}, \boldsymbol{L} \in \mathbb{R}^{m\times m}$ being symmetric positive semidefinite, and $\boldsymbol{A} \in \mathbb{R}^{Nm\times Nm}$ being nonsymmetric and possibly indefinite. In the last chapter, we could sometimes exploit the symmetry of $\boldsymbol{\mathcal{A}}$. However, this is not possible anymore in this chapter. No matter how we rearrange the blocks in $\boldsymbol{\mathcal{A}}$, $\boldsymbol{\mathcal{A}}$ is always nonsymmetric due to the nonsymmetry of $\boldsymbol{A}$. Moreover, the size of $\boldsymbol{\mathcal{A}}$ has increased manifold. In the last chapter, we have dealt with $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{2m\times 2m}$. In contrast, in this chapter we have $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{2Nm\times 2Nm}$. Due to the nonsymmetry of $\boldsymbol{\mathcal{A}}$, a nonsymmetric Krylov subspace solver is our method of choice. The crucial parameters represented in $\boldsymbol{\mathcal{A}}$ are the spatial mesh size h, the time step size τ, the interface parameter ε, the number of phases N, as well as the Moreau–Yosida regularization parameter c. We develop efficient preconditioners $\boldsymbol{\mathcal{P}}$ for the solution of the linear systems above. This is based on effective Schur complement approximations as well as (algebraic) multigrid solvers developed for elliptic systems [68, 136, 134]. In particular, our preconditioners behave promising regarding parameter changes. Moreover, we state a theoretical robustness proof for the smooth setting.

The structure of the chapter is as follows. The Cahn–Hilliard model is derived in Section 4.2. We first consider the smooth multi-well potential (4.4), which leads to a system of fourth-order PDEs. Then, we study the nonsmooth multi-obstacle potential (4.5), which yields a system of variational inequalities. Both formulations are discretized in time in Section 4.3. We focus on a fully implicit time-discrete scheme as motivated in the previous chapter. Regarding the smooth setting, we proof the energy stability and uniqueness of the solution of our time discretization scheme under reasonable assumptions. Concerning the nonsmooth framework, we consider the underlying minimization problem for which we derive existence and uniqueness conditions. In Section 4.4, we apply the Moreau–Yosida regularization technique and derive a convergence result. Section 4.5 shortly introduces the SSN method for solving the regularized subproblems. We derive the linear systems arising from the discretization using finite elements in Section 4.6. In Section 4.7, we analyze the linear systems and propose preconditioning strategies for the saddle point problems. Section 4.8 illustrates the efficiency of our preconditioners for both problem setups. In Section 4.9, we discuss alternative approaches. In Section 4.10, we summarize our findings and discuss possible future directions.

4.2 Derivation

There are two ways of deriving the multi-component Cahn–Hilliard equation. First, it can be derived as the H^{-1}-gradient flow of the Ginzburg-Landau energy (4.3) under the constraint (4.1). The second kind comes from the mass balance law; see, e.g., [60, 59]. We consider the latter case and briefly review the derivation of the multi-component Cahn–Hilliard equation. First of all, the smooth multi-well potential (4.4) setting is used. Then, we go over to the nonsmooth multi-obstacle potential (4.5) setting.

4.2.1 Smooth systems

In the following, we focus on the smooth multi-well potential (4.4). We briefly derive the multi-component Cahn–Hilliard equation in the framework of [60, 59]. We assume that the considered system is isothermal. The law of mass conservation is given as

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_R u_i \,\mathrm{d}\mathbf{x} = -\int_{\partial R} \mathbf{J}_i \cdot \mathbf{n} \,\mathrm{d}\mathbf{s} \quad \forall i = 1, \dots, N,$$

for any subregion $R \subset \Omega$. Here, $\mathbf{J}_i$ denotes the mass flux for each component $i = 1, \dots, N$. Due to Lemma 2.21, we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_R u_i \,\mathrm{d}\mathbf{x} = -\int_R \nabla \cdot \mathbf{J}_i \,\mathrm{d}\mathbf{x} \quad \forall i = 1, \dots, N.$$

Since R is fixed and arbitrary, we can derive

$$\partial_t u_i = -\nabla \cdot \mathbf{J}_i \quad \forall i = 1, \dots, N.$$

As in [59, p. 243], we assume the thermodynamical principle that the fluxes $\mathbf{J}_i, i = 1, \dots, N$, are linear and homogeneous functions of the forces $\nabla w_j, j = 1, \dots, N$, see also [106, p. 136]. We make the ansatz

$$\mathbf{J}_i = -\sum_{j=1}^{N} [\boldsymbol{L}(\mathbf{u})]_{ij} \nabla w_j = -\left(\boldsymbol{L}(\mathbf{u})\nabla \mathbf{w}\right)_i ,$$

where $\mathbf{w} = [w_1, \dots, w_N]^T$ is the vector of chemical potentials. The Onsager coefficients $[\boldsymbol{L}(\mathbf{u})]_{ij}$ may depend on $\mathbf{u}$; see, e.g., [59]. This thesis deals with constant coefficients $[\boldsymbol{L}]_{ij}$. During the rest of this chapter, we denote by $\boldsymbol{L} = \left([\boldsymbol{L}]_{ij}\right)_{i,j=1,\dots,N}$ the mobility matrix. Concentration dependent mobilities are a topic of future research. The vector of chemical potentials $\mathbf{w}$ is defined via the variational derivative of $\mathcal{E}$ with respect to $\mathbf{u}$ (see Definition 2.20), whereby

$$\mathcal{U} = \left\{ \mathbf{v} = [v_1, \dots, v_N]^T \in H^1(\Omega)^N : \sum_{i=1}^N v_i = 1 \right\},$$
$$\mathcal{Y} = \left\{ \mathbf{v} = [v_1, \dots, v_N]^T \in H^1(\Omega)^N : \sum_{i=1}^N v_i = 0 \right\}.$$

Note that for all $\mathbf{g} = [g_1, \dots, g_N]^T \in \mathcal{Y}$ there exists a vector $\mathbf{d} = [d_1, \dots, d_N]^T \in H^1(\Omega)^N$ such that

$$\mathbf{g} = \mathbf{d} - \frac{1}{N}\left(\sum_{i=1}^N d_i\right)\mathbf{1}. \tag{4.6}$$

For $\mathbf{v} = [v_1, \dots, v_N]^T \in \mathbb{R}^N$ let

$$\frac{\partial \psi_{\text{pol}}}{\partial \mathbf{u}}(\mathbf{v}) = \left[\frac{\partial \psi_{\text{pol}}}{\partial u_1}(\mathbf{v}), \dots, \frac{\partial \psi_{\text{pol}}}{\partial u_N}(\mathbf{v})\right]^T =: [\psi'_{\text{pol}}(v_1), \dots, \psi'_{\text{pol}}(v_N)]^T =: \psi'_{\text{pol}}(\mathbf{v})$$

in which $\psi'_{\text{pol}}(v_i) = v_i^3 - \frac{3}{2}v_i^2 + \frac{1}{2}v_i$. Calculating the variational derivative of $\mathcal{E}$, we obtain for $\mathbf{u} \in \mathcal{U}$ and $\mathbf{g} \in \mathcal{Y}$ satisfying (4.6) with $\mathbf{d} \in H^1(\Omega)^N$

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}\eta}\mathcal{E}(\mathbf{u} + \eta\mathbf{g}) &= \lim_{\eta \to 0} \frac{\mathcal{E}(\mathbf{u} + \eta\mathbf{g}) - \mathcal{E}(\mathbf{u})}{\eta} \\
&= \lim_{\eta \to 0} \frac{1}{\eta} \int_\Omega \left(\frac{\varepsilon^2}{2} \sum_{i=1}^N |\nabla(u_i + \eta g_i)|^2 + \frac{1}{4} \sum_{i=1}^N (u_i + \eta g_i)^2 (1 - u_i - \eta g_i)^2 \right. \\
&\qquad \left. - \frac{\varepsilon^2}{2} \sum_{i=1}^N |\nabla u_i|^2 - \frac{1}{4} \sum_{i=1}^N u_i^2 (1 - u_i)^2 \right) \mathrm{d}\mathbf{x} \\
&= \int_\Omega \sum_{i=1}^N \left(u_i^3 - \frac{3}{2}u_i^2 + \frac{1}{2}u_i \right) g_i \, \mathrm{d}\mathbf{x} + \varepsilon^2 \int_\Omega \sum_{i=1}^N \nabla u_i \cdot \nabla g_i \, \mathrm{d}\mathbf{x} \\
&= \int_\Omega \sum_{i=1}^N \left(\psi'_{\text{pol}}(u_i) - \varepsilon^2 \Delta u_i \right) g_i \, \mathrm{d}\mathbf{x} \qquad (4.7) \\
&\overset{(4.6)}{=} \int_\Omega \sum_{i=1}^N \left(\psi'_{\text{pol}}(u_i) - \varepsilon^2 \Delta u_i \right) \left(d_i - \frac{1}{N} \sum_{j=1}^N d_j \right) \mathrm{d}\mathbf{x} \\
&\overset{(4.1)}{=} \int_\Omega \sum_{i=1}^N \left(\psi'_{\text{pol}}(u_i) - \frac{1}{N} \sum_{j=1}^N \psi'_{\text{pol}}(u_j) - \varepsilon^2 \Delta u_i \right) d_i \, \mathrm{d}\mathbf{x} = \int_\Omega \sum_{i=1}^N w_i \, d_i \, \mathrm{d}\mathbf{x}.
\end{aligned}$$

The identity in (4.7) is supplemented with Lemma 2.21 together with the natural zero Neumann boundary condition $\nabla u_i \cdot \mathbf{n} = 0$ on $\partial\Omega$ for $i = 1, \dots, N$. Here, $\mathbf{n}$ is the unit normal vector to $\partial\Omega$ pointing outwards from Ω. Moreover, we impose the mass conserving boundary condition

$$(\boldsymbol{L}\nabla\mathbf{w})_i \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega, \ i = 1, \dots, N.$$

In conclusion, we obtain the vector-valued Cahn–Hilliard equation:

$$\partial_t u_i = (\boldsymbol{L}\Delta\mathbf{w})_i, \tag{4.8}$$

$$w_i = -\varepsilon^2\Delta u_i + \psi'_{\mathrm{pol}}(u_i) - \frac{1}{N}\sum_{j=1}^{N}\psi'_{\mathrm{pol}}(u_j), \tag{4.9}$$

$$\nabla u_i \cdot \mathbf{n} = (\boldsymbol{L}\nabla\mathbf{w})_i \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega, \tag{4.10}$$

for $i = 1, \dots, N$.

In order to fulfill Onsager's reciprocity law, we require that $\boldsymbol{L}$ is symmetric; see [106, p. 137] or [59, p. 243]. In order to ensure that the constraint (4.1) is fulfilled during the evolution, a common way in the literature is to assume that

$$\boldsymbol{L}\mathbf{1} = \mathbf{0}, \tag{4.11}$$

see, e.g., [60, 26, 59]. Since summing (4.8) over $i = 1, \dots, N$ then leads to

$$\frac{\partial}{\partial t}\sum_{i=1}^{N} u_i = \sum_{i=1}^{N}\partial_t u_i = \sum_{i=1}^{N}\nabla\cdot(\boldsymbol{L}\nabla\mathbf{w})_i = \nabla\cdot\sum_{i,j=1}^{N}[\boldsymbol{L}]_{ij}\nabla w_j = \nabla\cdot\sum_{j=1}^{N}\nabla w_j\sum_{i=1}^{N}[\boldsymbol{L}]_{ij} = 0.$$

Therefore, (4.1) is fulfilled during the evolution if $\sum_{i=1}^{N} u_i = 1$ holds at time $t = 0$. Summing (4.9) over $i = 1, \dots, N$, then leads to

$$\sum_{i=1}^{N} w_i = -\varepsilon^2\Delta\sum_{i=1}^{N} u_i = 0. \tag{4.12}$$

It is further assumed that $\boldsymbol{L}$ is positive semidefinite, which gives that the total energy is non-increasing in time. Differentiating the energy $\mathcal{E}$ in (4.3) with respect to the time yields

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{E}(\mathbf{u}) &= \frac{\mathrm{d}}{\mathrm{d}t}\int_\Omega \frac{\varepsilon^2}{2}\sum_{i=1}^{N}|\nabla u_i|^2 + \psi_{\mathrm{pol}}(\mathbf{u})\,\mathrm{d}\mathbf{x} = \int_\Omega\sum_{i=1}^{N}\left[\varepsilon^2\nabla u_i\cdot\nabla\partial_t u_i + \frac{\partial\psi_{\mathrm{pol}}(\mathbf{u})}{\partial u_i}\partial_t u_i\right]\mathrm{d}\mathbf{x} \\
&= \int_\Omega\sum_{i=1}^{N}\left[-\varepsilon^2\Delta u_i + \psi'_{\mathrm{pol}}(u_i)\right]\partial_t u_i\,\mathrm{d}\mathbf{x} \overset{(4.9)}{=} \int_\Omega\sum_{i=1}^{N}\left[w_i + \frac{1}{N}\sum_{j=1}^{N}\psi'_{\mathrm{pol}}(u_j)\right]\partial_t u_i\,\mathrm{d}\mathbf{x} \\
&\overset{(4.1)}{=} \int_\Omega\sum_{i=1}^{N} w_i(\boldsymbol{L}\Delta\mathbf{w})_i\,\mathrm{d}\mathbf{x} = -\int_\Omega\sum_{i=1}^{N}\nabla w_i\cdot(\boldsymbol{L}\nabla\mathbf{w})_i\,\mathrm{d}\mathbf{x} \le 0,
\end{aligned}$$

where we have used Lemma 2.21 with (4.10). Therefore, the total energy is non-increasing in time. Differentiating the total mass $\int_\Omega u_i\,\mathrm{d}\mathbf{x}$ with respect to the time gives

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_\Omega u_i\,\mathrm{d}\mathbf{x} = \int_\Omega\frac{\partial u_i}{\partial t}\,\mathrm{d}\mathbf{x} \overset{(4.8)}{=} \int_\Omega(\boldsymbol{L}\Delta\mathbf{w})_i\,\mathrm{d}\mathbf{x} = -\int_{\partial\Omega}(\boldsymbol{L}\nabla\mathbf{w})_i\cdot\mathbf{n}\,\mathrm{d}s \overset{(4.10)}{=} 0.$$

Hence, the total mass of each phase is conserved.

Since $\boldsymbol{L}$ is symmetric positive semidefinite, we can make use of its symmetric Schur decomposition

$$\boldsymbol{L} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T = \boldsymbol{Q}\boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{Q}^T\boldsymbol{Q}\boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{Q}^T =: \boldsymbol{L}_s\boldsymbol{L}_s, \tag{4.13}$$

where $\boldsymbol{\Lambda}^{\frac{1}{2}}$ is a diagonal matrix containing the square roots of the eigenvalues of $\boldsymbol{L}$ and $\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}$. Note that $\boldsymbol{L}_s \in \mathbb{R}^{N\times N}$ is symmetric positive semidefinite. Further, as stated in [60, p. 7] or [26, p. 112], it holds

$$\boldsymbol{v}^T\boldsymbol{L}\boldsymbol{v} \geq l_0(\boldsymbol{P}\boldsymbol{v})^T\boldsymbol{P}\boldsymbol{v} = l_0\boldsymbol{v}^T\boldsymbol{P}\boldsymbol{v} = l_0\left(\boldsymbol{v}^T\boldsymbol{v} - \frac{1}{N}\left(\sum_{i=1}^N v_i\right)\boldsymbol{v}^T\boldsymbol{1}\right), \tag{4.14}$$

where l_0 is the smallest positive eigenvalue of $\boldsymbol{L}$ and $\boldsymbol{P}\boldsymbol{v} = \boldsymbol{v} - \frac{1}{N}\left(\sum_{i=1}^N v_i\right)\boldsymbol{1}$ for all $\boldsymbol{v} = [v_1, \dots, v_N]^T \in \mathbb{R}^N$. Note that $\boldsymbol{P} = \boldsymbol{P}^T = \boldsymbol{P}^2$.

Remark 4.1. *We have explained above the common assumption $\boldsymbol{L}\boldsymbol{1} = \boldsymbol{0}$. Nevertheless, it is possible to work with $\boldsymbol{L} = \boldsymbol{I}$ for convenience; see, e.g., [118]. Therefore, we also consider this case in our work. We will see in Section 4.7 that our numerical solver simplifies in this case.*

4.2.2 Nonsmooth systems

In the last section, we focused on the smooth potential ψ_{pol}. We could easily calculate the derivative of the smooth potential with respect to $\mathbf{u}$. Now, we turn to the nonsmooth potential ψ_{obs} given in (4.5). It can be written via the indicator function

$$\mathcal{I}_{\mathcal{G}^N}(\mathbf{u}) = \begin{cases} 0 & \mathbf{u} \in \mathcal{G}^N, \\ \infty & \text{otherwise} \end{cases}$$

as

$$\psi_{\text{obs}}(\mathbf{u}) = \psi_0(\mathbf{u}) + \mathcal{I}_{\mathcal{G}^N}(\mathbf{u}) = -\frac{1}{2}\mathbf{u}\cdot\boldsymbol{T}\mathbf{u} + \mathcal{I}_{\mathcal{G}^N}(\mathbf{u}).$$

As in Chapter 3.2.2, $\mathcal{I}_{\mathcal{G}^N}$ can only be differentiated in the sense of subdifferentials. The resulting system can be formulated as a system of variational inequalities; see [60]: Find $(\mathbf{u}, \mathbf{w}) \in H^1(\Omega)^N \times H^1(\Omega)^N$ such that

$$\langle \partial_t \mathbf{u}, \mathbf{v}\rangle = -(\boldsymbol{L}\nabla\mathbf{w}, \nabla\mathbf{v}) \quad \forall \mathbf{v} \in H^1(\Omega)^N, \tag{4.15}$$

$$(\mathbf{w}, \mathbf{v}-\mathbf{u}) \leq \varepsilon^2(\nabla\mathbf{u}, \nabla(\mathbf{v}-\mathbf{u})) - (\boldsymbol{T}\mathbf{u}, \mathbf{v}-\mathbf{u}) + \frac{1}{N}\sum_{j=1}^N ((\boldsymbol{T}\mathbf{u})_j\boldsymbol{1}, \mathbf{v}-\mathbf{u}) \qquad \forall \mathbf{v} \in H^1(\Omega)^N, \mathbf{v} \in \mathcal{G}^N \text{ a.e. in } \Omega, \tag{4.16}$$

$$\mathbf{u} \in \mathcal{G}^N \quad \text{a.e. in } \Omega. \tag{4.17}$$

The system (4.15)–(4.17) is supplemented by the initial condition $\mathbf{u}_0 \in H^1(\Omega)^N, \mathbf{u}_0 \in \mathcal{G}^N$ a.e. in Ω. Existence and uniqueness of a solution to (4.15)–(4.17) was shown in [60].

Remark 4.2. *Another formulation was studied in [12]: Given* $\mathbf{u}_0 \in H^1(\Omega)$, $\mathbf{u}_0 \in \mathcal{G}^N$ a.e. in Ω, *find* $(\mathbf{u}, \mathbf{w}, \eta) \in H^1(\Omega)^N \times H^1(\Omega)^N \times L^2(\Omega)$ *such that*

$$\langle \partial_t \mathbf{u}, \mathbf{v} \rangle = -(\boldsymbol{L}\nabla \mathbf{w}, \nabla \mathbf{v}) \quad \forall \mathbf{v} \in H^1(\Omega)^N, \tag{4.18}$$

$$(\mathbf{w} + \eta \mathbf{1}, \mathbf{v} - \mathbf{u}) \leq \varepsilon^2 (\nabla \mathbf{u}, \nabla(\mathbf{v} - \mathbf{u})) - (\boldsymbol{T}\mathbf{u}, \mathbf{v} - \mathbf{u}) + \frac{1}{N} \sum_{j=1}^{N} ((\boldsymbol{T}\mathbf{u})_j \mathbf{1}, \mathbf{v} - \mathbf{u}) \qquad \forall \mathbf{v} \in H^1(\Omega)^N, \mathbf{v} \geq \mathbf{0} \text{ a.e. in } \Omega, \tag{4.19}$$

$$\mathbf{u} \geq \mathbf{0} \quad \text{a.e. in } \Omega, \tag{4.20}$$

$$\sum_{i=1}^{N} w_i = 0 \quad \text{a.e. in } \Omega. \tag{4.21}$$

This formulation relaxes the condition $\mathbf{u} \in \mathcal{G}^N$ *almost everywhere in* Ω *in (4.15)–(4.17) to* $\mathbf{u} \geq \mathbf{0}$ *almost everywhere in* Ω *by introducing the Lagrange multipliers explicitly. Existence and uniqueness of a solution to (4.18)–(4.21) was shown in [12].*

Due to the system of variational inequalities, the nonsmooth vector-valued Cahn–Hilliard problem poses a harder challenge compared to the smooth one from the last section. This holds for both, the mathematical as well as numerical analysis, as we shall see during the following sections.

After the derivation of the constitutive vector-valued Cahn–Hilliard equation or inequality, we are going to study their discretizations in order to be able to solve them numerically. We start with the discretization in time in the next section.

4.3 Time discretization

In Chapter 3.3, we have motivated the use of a fully implicit time discretization scheme. This means we use the backward Euler discretization for the time derivatives $\partial_t u_i, i = 1, \ldots, N$, and treat all the other terms implicitly. In particular, we treat the potential function implicitly. Let $\tau > 0$ denote the time step size and $t_{n-1} = (n-1)\tau$, $n \in \mathbb{N}$, discrete times. We denote by $\mathbf{u}^{(n-1)} \in H^1(\Omega)^N$ the time-discrete solution at time step t_{n-1}. Further, $\mathbf{u}^{(n)}$, $\mathbf{w}^{(n)} \in H^1(\Omega)^N$ form the time-discrete solution at time step $t_n = t_{n-1} + \tau$. In order to ease the notation, from now on we write $\mathbf{u}^{\text{old}}$, $\mathbf{u}$, and $\mathbf{w}$ instead of $\mathbf{u}^{(n-1)}$, $\mathbf{u}^{(n)}$, and $\mathbf{w}^{(n)}$, respectively.

Again, we start with the smooth setting and consider the weak formulation of (4.8)–(4.10). We discretize this problem in time and give stability and uniqueness conditions. Afterwards, we go over to the nonsmooth setting (4.15)–(4.17). We consider the corresponding optimization problem which allows us to specify the conditions for a unique solution.

4.3.1 Smooth systems

Let us focus on the smooth setting and the corresponding system of Cahn–Hilliard Equations (4.8)–(4.10). We consider their weak formulation and utilize the implicit Euler scheme. Then, $(\mathbf{u}, \mathbf{w})$ solves the following problem: Find $\mathbf{u}$, $\mathbf{w} \in H^1(\Omega)^N$ such that

$$\left(u_i - u_i^{\text{old}}, v\right) + \tau\left((\boldsymbol{L}\nabla\mathbf{w})_i, \nabla v\right) = 0 \quad \forall v \in H^1(\Omega), \tag{4.22}$$

$$-\left(w_i, v\right) + \varepsilon^2\left(\nabla u_i, \nabla v\right) + \left(\psi'_{\text{pol}}(u_i), v\right) - \frac{1}{N}\left(\sum_{j=1}^{N} \psi'_{\text{pol}}(u_j), v\right) = 0 \quad \forall v \in H^1(\Omega), \tag{4.23}$$

for $i = 1, \dots, N$. Choosing $v = 1$ in (4.22), we obtain the conservation of mass, i.e., $(u_i, 1) = (u_i^{\text{old}}, 1)$ for $i = 1, \dots, N$, the specific feature of the Cahn–Hilliard model.

Now, we want to give stability and uniqueness conditions for the time step. However, the quartic growth of $\psi_{\text{pol}}(\mathbf{u})$ at infinity introduces various technical difficulties in the analysis. Therefore, as in Chapter 3.3.1, we consider a truncated multi-well potential. To be more precise, we restrict the growth of $\psi_{\text{pol}}(\mathbf{u})$ to be quadratic for $u_i \leq 1 - M$ and $u_i \geq M$ for a given constant M. In the following, we write $\tilde{\psi}$ for the truncated version of ψ_{pol}. Using the truncation technique, we obtain the following condition: There exists a constant S such that

$$\max_{\mathbf{s}\in\mathbb{R}^N}\left|\frac{\partial^2\tilde{\psi}}{\partial {u_i}^2}(\mathbf{s})\right| \leq S \quad \forall i = 1, \dots, N. \tag{4.24}$$

With the use of (4.24), we can prove:

Theorem 4.1. *The solution of (4.22)–(4.23) is unique provided that* $\tau < \frac{4\varepsilon^2}{S^2\rho(\boldsymbol{L})}$ *and* $\psi = \psi_{\text{pol}}$ *is replaced by its truncated version* $\tilde{\psi}$.

Proof. Assume there exist two solutions $(\mathbf{u}, \mathbf{w})$ and $(\tilde{\mathbf{u}}, \tilde{\mathbf{w}})$ of (4.22)–(4.23). Then, we get

$$(\mathbf{u} - \tilde{\mathbf{u}}, \mathbf{v}) + \tau\left(\boldsymbol{L}\nabla(\mathbf{w} - \tilde{\mathbf{w}}), \nabla\mathbf{v}\right) = 0, \tag{4.25}$$

$$\begin{aligned} -(\mathbf{w} - \tilde{\mathbf{w}}, \mathbf{v}) + \varepsilon^2\left(\nabla(\mathbf{u} - \tilde{\mathbf{u}}), \nabla\mathbf{v}\right) + \left(\psi'(\mathbf{u}) - \psi'(\tilde{\mathbf{u}}), \mathbf{v}\right)& \\ - \frac{1}{N}\left(\sum_{j=1}^{N}\left(\psi'(u_j) - \psi'(\tilde{u}_j)\right)\mathbf{1}, \mathbf{v}\right) &= 0, \end{aligned} \tag{4.26}$$

for all $\mathbf{v} \in H^1(\Omega)^N$. Choosing $\mathbf{v} = \mathbf{w} - \tilde{\mathbf{w}}$ in (4.25) gives

$$\begin{aligned} 0 &= (\mathbf{u} - \tilde{\mathbf{u}}, \mathbf{w} - \tilde{\mathbf{w}}) + \tau(\boldsymbol{L}\nabla(\mathbf{w} - \tilde{\mathbf{w}}), \nabla(\mathbf{w} - \tilde{\mathbf{w}})) \\ &\overset{(4.13)}{=} (\mathbf{u} - \tilde{\mathbf{u}}, \mathbf{w} - \tilde{\mathbf{w}}) + \tau(\boldsymbol{L}_s\boldsymbol{L}_s\nabla(\mathbf{w} - \tilde{\mathbf{w}}), \nabla(\mathbf{w} - \tilde{\mathbf{w}})) \\ &= (\mathbf{u} - \tilde{\mathbf{u}}, \mathbf{w} - \tilde{\mathbf{w}}) + \tau(\boldsymbol{L}_s\nabla(\mathbf{w} - \tilde{\mathbf{w}}), \boldsymbol{L}_s\nabla(\mathbf{w} - \tilde{\mathbf{w}})) \\ &= (\mathbf{u} - \tilde{\mathbf{u}}, \mathbf{w} - \tilde{\mathbf{w}}) + \tau\left\|\boldsymbol{L}_s\nabla(\mathbf{w} - \tilde{\mathbf{w}})\right\|^2. \end{aligned} \tag{4.27}$$

Choosing $\mathbf{v} = \mathbf{u} - \tilde{\mathbf{u}}$ in (4.26) gives

$$\begin{aligned} 0 = &-(\mathbf{u} - \tilde{\mathbf{u}}, \mathbf{w} - \tilde{\mathbf{w}}) + \varepsilon^2\left\|\nabla(\mathbf{u} - \tilde{\mathbf{u}})\right\|^2 \\ &+ \left(\psi'(\mathbf{u}) - \psi'(\tilde{\mathbf{u}}), \mathbf{u} - \tilde{\mathbf{u}}\right) - \frac{1}{N}\left(\sum_{j=1}^{N}\left(\psi'(u_j) - \psi'(\tilde{u}_j)\right)\mathbf{1}, \mathbf{u} - \tilde{\mathbf{u}}\right). \end{aligned} \tag{4.28}$$

The last term in (4.28) is zero since we can reorder this term to

$$\left(\sum_{j=1}^{N}\left(\psi'(u_j)-\psi'(\tilde{u}_j)\right)\mathbf{1},\mathbf{u}-\tilde{\mathbf{u}}\right)=\sum_{i=1}^{N}\left(\sum_{j=1}^{N}\left(\psi'(u_j)-\psi'(\tilde{u}_j)\right),u_i-\tilde{u}_i\right)$$
$$=\sum_{j=1}^{N}\left(\psi'(u_j)-\psi'(\tilde{u}_j),\sum_{i=1}^{N}(u_i-\tilde{u}_i)\right)$$

and $\sum_{i=1}^{N}(u_i-\tilde{u}_i)=0$ for a.e. $\mathbf{x}\in\Omega$ due to (4.1). The second last term in (4.28) can be reformulated using the Taylor expansion of the potential

$$\psi(\mathbf{u})=\psi(\tilde{\mathbf{u}}+\mathbf{u}-\tilde{\mathbf{u}})=\psi(\tilde{\mathbf{u}})+\psi'(\tilde{\mathbf{u}})\cdot(\mathbf{u}-\tilde{\mathbf{u}})+\frac{1}{2}\sum_{i=1}^{N}f'(s_i)(u_i-\tilde{u}_i)^2,$$
$$\psi(\tilde{\mathbf{u}})=\psi(\mathbf{u}+\tilde{\mathbf{u}}-\mathbf{u})=\psi(\mathbf{u})+\psi'(\mathbf{u})\cdot(\tilde{\mathbf{u}}-\mathbf{u})+\frac{1}{2}\sum_{i=1}^{N}f'(\tilde{s}_i)(u_i-\tilde{u}_i)^2,$$

where $f'(s_i)=\frac{\partial^2\tilde{\psi}}{\partial u_i{}^2}(\mathbf{s})$ and $\mathbf{s},\tilde{\mathbf{s}}$ lie between $\mathbf{u}$ and $\tilde{\mathbf{u}}$. Adding these two equations gives

$$(\psi'(\mathbf{u})-\psi'(\tilde{\mathbf{u}}))\cdot(\mathbf{u}-\tilde{\mathbf{u}})=\frac{1}{2}\sum_{i=1}^{N}\left(f'(s_i)+f'(\tilde{s}_i)\right)(u_i-\tilde{u}_i)^2\overset{(4.24)}{\geq}-S\sum_{i=1}^{N}(u_i-\tilde{u}_i)^2.$$

Therefore, we obtain in (4.28)

$$0\geq-(\mathbf{u}-\tilde{\mathbf{u}},\mathbf{w}-\tilde{\mathbf{w}})+\varepsilon^2\left\|\nabla(\mathbf{u}-\tilde{\mathbf{u}})\right\|^2-S\left\|\mathbf{u}-\tilde{\mathbf{u}}\right\|^2. \tag{4.29}$$

For the last equation, we choose $\mathbf{v}=S\,(\mathbf{u}-\tilde{\mathbf{u}})$ in (4.25) and get

$$\begin{aligned}0&=S\left\|\mathbf{u}-\tilde{\mathbf{u}}\right\|^2+\tau S\left(\boldsymbol{L}\nabla(\mathbf{w}-\tilde{\mathbf{w}}),\nabla(\mathbf{u}-\tilde{\mathbf{u}})\right)\\&=S\left\|\mathbf{u}-\tilde{\mathbf{u}}\right\|^2+\left(\frac{\tau S\sqrt{\rho(\boldsymbol{L})}}{\sqrt{2}\varepsilon}\boldsymbol{L}_s\nabla(\mathbf{w}-\tilde{\mathbf{w}}),\frac{\sqrt{2}\varepsilon}{\sqrt{\rho(\boldsymbol{L})}}\boldsymbol{L}_s\nabla(\mathbf{u}-\tilde{\mathbf{u}})\right)\\&\geq S\left\|\mathbf{u}-\tilde{\mathbf{u}}\right\|^2-\frac{\tau^2S^2\rho(\boldsymbol{L})}{4\varepsilon^2}\left\|\boldsymbol{L}_s\nabla(\mathbf{w}-\tilde{\mathbf{w}})\right\|^2-\frac{\varepsilon^2}{\rho(\boldsymbol{L})}\left\|\boldsymbol{L}_s\nabla(\mathbf{u}-\tilde{\mathbf{u}})\right\|^2,\end{aligned} \tag{4.30}$$

where we have used Young's inequality with $\alpha_Y=1$ (Lemma 2.12). Due to

$$\begin{aligned}\left\|\boldsymbol{L}_s\nabla(\mathbf{u}-\tilde{\mathbf{u}})\right\|^2&=\left(\boldsymbol{L}_s\nabla(\mathbf{u}-\tilde{\mathbf{u}}),\boldsymbol{L}_s\nabla(\mathbf{u}-\tilde{\mathbf{u}})\right)=\left(\boldsymbol{L}\nabla(\mathbf{u}-\tilde{\mathbf{u}}),\nabla(\mathbf{u}-\tilde{\mathbf{u}})\right)\\&\leq\rho(\boldsymbol{L})\left\|\nabla(\mathbf{u}-\tilde{\mathbf{u}})\right\|^2,\end{aligned}$$

we can estimate (4.30) further to

$$0\geq S\left\|\mathbf{u}-\tilde{\mathbf{u}}\right\|^2-\frac{\tau^2S^2\rho(\boldsymbol{L})}{4\varepsilon^2}\left\|\boldsymbol{L}_s\nabla(\mathbf{w}-\tilde{\mathbf{w}})\right\|^2-\varepsilon^2\left\|\nabla(\mathbf{u}-\tilde{\mathbf{u}})\right\|^2. \tag{4.31}$$

Now, adding (4.27), (4.29), and (4.31), we get

$$0\geq\tau\left(1-\frac{\tau S^2\rho(\boldsymbol{L})}{4\varepsilon^2}\right)\left\|\boldsymbol{L}_s\nabla(\mathbf{w}-\tilde{\mathbf{w}})\right\|^2. \tag{4.32}$$

Hence, we obtain uniqueness if $1 - \frac{\tau S^2 \rho(\boldsymbol{L})}{4\varepsilon^2} > 0$, which is equivalent to $\tau < \frac{4\varepsilon^2}{S^2\rho(\boldsymbol{L})}$. Since then, it follows with (4.14) that

$$\begin{aligned} \|\boldsymbol{L}_s \nabla(\mathbf{w} - \tilde{\mathbf{w}})\|^2 &= (\boldsymbol{L}\nabla(\mathbf{w} - \tilde{\mathbf{w}}), \nabla(\mathbf{w} - \tilde{\mathbf{w}})) \geq l_0\,(\boldsymbol{P}\nabla(\mathbf{w} - \tilde{\mathbf{w}}), \nabla(\mathbf{w} - \tilde{\mathbf{w}})) \\ &= l_0 \|\nabla(\mathbf{w} - \tilde{\mathbf{w}})\|^2, \end{aligned}$$

where the last equality is a result from (4.12) in a weak sense. Therefore, we obtain in (4.32)

$$0 \geq \tau\left(1 - \frac{\tau S^2 \rho(\boldsymbol{L})}{4\varepsilon^2}\right)\|\nabla(\mathbf{w} - \tilde{\mathbf{w}})\|^2 .$$

Finally, $\|\nabla(\mathbf{w} - \tilde{\mathbf{w}})\| = 0$ implies that $\mathbf{w} - \tilde{\mathbf{w}}$ is constant. Using this, (4.25) yields $(\mathbf{u} - \tilde{\mathbf{u}}, \mathbf{v}) = 0\ \forall \mathbf{v} \in H^1(\Omega)^N$ and therefore $\mathbf{u} = \tilde{\mathbf{u}}$ almost everywhere. Finally, (4.26) then gives $\mathbf{w} = \tilde{\mathbf{w}}$ almost everywhere. □

Theorem 4.2. *Under the condition $\tau < \frac{8\varepsilon^2}{S^2\rho(\boldsymbol{L})}$ and provided that $\psi = \psi_{\text{pol}}$ is replaced by its truncated version $\tilde{\psi}$, the time discretization scheme (4.22)–(4.23) is energy stable, i.e., its solution satisfies $\mathcal{E}(\mathbf{u}^{(n)}) \leq \mathcal{E}(\mathbf{u}^{(n-1)})$ for all $n \geq 1$.*

Proof. Choosing $\mathbf{v} = \mathbf{w}$ in (4.22) gives

$$\begin{aligned} 0 &= \left(\mathbf{u} - \mathbf{u}^{\text{old}}, \mathbf{w}\right) + \tau\,(\boldsymbol{L}\nabla\mathbf{w}, \nabla\mathbf{w}) \overset{(4.13)}{=} \left(\mathbf{u} - \mathbf{u}^{\text{old}}, \mathbf{w}\right) + \tau\,(\boldsymbol{L}_s\boldsymbol{L}_s\nabla\mathbf{w}, \nabla\mathbf{w}) \\ &= \left(\mathbf{u} - \mathbf{u}^{\text{old}}, \mathbf{w}\right) + \tau\,(\boldsymbol{L}_s\nabla\mathbf{w}, \boldsymbol{L}_s\nabla\mathbf{w}) = \left(\mathbf{u} - \mathbf{u}^{\text{old}}, \mathbf{w}\right) + \tau\,\|\boldsymbol{L}_s\nabla\mathbf{w}\|^2 . \end{aligned} \tag{4.33}$$

Choosing $\mathbf{v} = \mathbf{u} - \mathbf{u}^{\text{old}}$ in (4.23) gives

$$\begin{aligned} 0 = &-\left(\mathbf{u} - \mathbf{u}^{\text{old}}, \mathbf{w}\right) + \frac{\varepsilon^2}{2}\left(\|\nabla\mathbf{u}\|^2 - \left\|\nabla\mathbf{u}^{\text{old}}\right\|^2 + \left\|\nabla(\mathbf{u} - \mathbf{u}^{\text{old}})\right\|^2\right) \\ &+ \left(\psi'(\mathbf{u}), \mathbf{u} - \mathbf{u}^{\text{old}}\right) - \frac{1}{N}\left(\sum_{j=1}^{N}\psi'(u_j)\,\mathbf{1}, \mathbf{u} - \mathbf{u}^{\text{old}}\right). \end{aligned} \tag{4.34}$$

As in the proof before, we can show that the last term in (4.34) is zero, and the second last term in (4.34) can be reformulated using the Taylor expansion of the potential

$$\begin{aligned} \psi'(\mathbf{u}) \cdot \left(\mathbf{u} - \mathbf{u}^{\text{old}}\right) &= \psi(\mathbf{u}) - \psi(\mathbf{u}^{\text{old}}) + \frac{1}{2}\sum_{i=1}^{N} f'(s_i)(u_i - u_i^{\text{old}})^2 \\ &\overset{(4.24)}{\geq} \psi(\mathbf{u}) - \psi(\mathbf{u}^{\text{old}}) - \frac{S}{2}\sum_{i=1}^{N}(u_i - u_i^{\text{old}})^2, \end{aligned}$$

where $\mathbf{s}$ lies between $\mathbf{u}$ and $\mathbf{u}^{\text{old}}$. Therefore, we obtain in (4.34)

$$\begin{aligned} 0 \geq &-\left(\mathbf{u} - \mathbf{u}^{\text{old}}, \mathbf{w}\right) + \frac{\varepsilon^2}{2}\left(\|\nabla\mathbf{u}\|^2 - \left\|\nabla\mathbf{u}^{\text{old}}\right\|^2 + \left\|\nabla(\mathbf{u} - \mathbf{u}^{\text{old}})\right\|^2\right) \\ &+ (\psi(\mathbf{u}), 1) - \left(\psi(\mathbf{u}^{\text{old}}), 1\right) - \frac{S}{2}\left\|\mathbf{u} - \mathbf{u}^{\text{old}}\right\|^2 . \end{aligned} \tag{4.35}$$

For the last equation, we choose $\mathbf{v} = \frac{S}{2}\left(\mathbf{u} - \mathbf{u}^{\text{old}}\right)$ in (4.22) and get

$$\begin{aligned} 0 &= \frac{S}{2}\left\|\mathbf{u} - \tilde{\mathbf{u}}\right\|^2 + \frac{\tau S}{2}\left(\boldsymbol{L}\nabla\mathbf{w}, \nabla(\mathbf{u} - \mathbf{u}^{\text{old}})\right) \\ &= \frac{S}{2}\left\|\mathbf{u} - \mathbf{u}^{\text{old}}\right\|^2 + \left(\frac{\tau S\sqrt{\rho(\boldsymbol{L})}}{2\varepsilon}\boldsymbol{L}_s\nabla\mathbf{w}, \frac{\varepsilon}{\sqrt{\rho(\boldsymbol{L})}}\boldsymbol{L}_s\nabla(\mathbf{u} - \mathbf{u}^{\text{old}})\right) \\ &\geq \frac{S}{2}\left\|\mathbf{u} - \mathbf{u}^{\text{old}}\right\|^2 - \frac{\tau^2 S^2\rho(\boldsymbol{L})}{8\varepsilon^2}\left\|\boldsymbol{L}_s\nabla\mathbf{w}\right\|^2 - \frac{\varepsilon^2}{2\rho(\boldsymbol{L})}\left\|\boldsymbol{L}_s\nabla(\mathbf{u} - \mathbf{u}^{\text{old}})\right\|^2, \end{aligned} \tag{4.36}$$

where we have used Young's inequality with $\alpha_Y = 1$ (Lemma 2.12). Due to

$$\begin{aligned} \left\|\boldsymbol{L}_s\nabla(\mathbf{u} - \mathbf{u}^{\text{old}})\right\|^2 &= \left(\boldsymbol{L}_s\nabla(\mathbf{u} - \mathbf{u}^{\text{old}}), \boldsymbol{L}_s\nabla(\mathbf{u} - \mathbf{u}^{\text{old}})\right) \\ &= \left(\boldsymbol{L}\nabla(\mathbf{u} - \mathbf{u}^{\text{old}}), \nabla(\mathbf{u} - \mathbf{u}^{\text{old}})\right) \\ &\leq \rho(\boldsymbol{L})\left\|\nabla(\mathbf{u} - \mathbf{u}^{\text{old}})\right\|^2, \end{aligned}$$

we can estimate (4.36) further to

$$0 \geq \frac{S}{2}\left\|\mathbf{u} - \mathbf{u}^{\text{old}}\right\|^2 - \frac{\tau^2 S^2\rho(\boldsymbol{L})}{8\varepsilon^2}\left\|\boldsymbol{L}_s\nabla\mathbf{w}\right\|^2 - \frac{\varepsilon^2}{2}\left\|\boldsymbol{L}_s\nabla(\mathbf{u} - \mathbf{u}^{\text{old}})\right\|^2. \tag{4.37}$$

Now, adding (4.33), (4.35), and (4.37), we get

$$0 \geq \tau\left(1 - \frac{\tau S^2\rho(\boldsymbol{L})}{8\varepsilon^2}\right)\left\|\boldsymbol{L}_s\nabla\mathbf{w}\right\|^2 + \frac{\varepsilon^2}{2}\left(\left\|\nabla\mathbf{u}\right\|^2 - \left\|\nabla\mathbf{u}^{\text{old}}\right\|^2\right) + (\psi(\mathbf{u}), 1) - \left(\psi(\mathbf{u}^{\text{old}}), 1\right).$$

Now, we can bound the energy

$$\begin{aligned} \mathcal{E}(\mathbf{u}) - \mathcal{E}(\mathbf{u}^{\text{old}}) &= \frac{\varepsilon^2}{2}\left(\left\|\nabla\mathbf{u}\right\|^2 - \left\|\nabla\mathbf{u}^{\text{old}}\right\|^2\right) + (\psi(\mathbf{u}), 1) - \left(\psi(\mathbf{u}^{\text{old}}), 1\right) \\ &\leq \tau\left(\frac{\tau S^2\rho(\boldsymbol{L})}{8\varepsilon^2} - 1\right)\left\|\boldsymbol{L}_s\nabla\mathbf{w}\right\|^2. \end{aligned}$$

Hence, we obtain energy stability if $\frac{\tau S^2\rho(\boldsymbol{L})}{8\varepsilon^2} - 1 \leq 0$, which is equivalent to $\tau \leq \frac{8\varepsilon^2}{S^2\rho(\boldsymbol{L})}$. □

The resulting time step restrictions comply with the ones for the two-component system in Chapter 3.3.1. Although these conditions appear to be quite restrictive for $\varepsilon \ll 1$, the authors of [140] pointed out that they are in fact needed for the sake of convergence. Moreover, note that explicit schemes usually lead to even more severe time step restrictions of order $O(\varepsilon^4)$.

As in the previous chapter, the approach of the truncated polynomial is only used for the theoretical part. In praxis, the polynomial potential ψ_{pol} behaves quite well and does not result in blow ups of the solution. Violations of $u \in [0, 1]$ in form of $u \in [-\delta(\varepsilon), 1 + \delta(\varepsilon)]$ occur. However, $\delta(\varepsilon)$ is relatively small. We investigate this issue further in Section 4.8.3.

After having stated and analyzed our system of time-discrete Cahn–Hilliard equations in the smooth setting, we proceed with the nonsmooth case.

4.3.2 Nonsmooth systems

In the following, we concentrate on the nonsmooth setting and the corresponding system of Cahn–Hilliard variational inequalities (4.15)–(4.17). By utilizing the implicit Euler scheme, we obtain the following problem: Find $\mathbf{u}$, $\mathbf{w} \in H^1(\Omega)^N$ such that

$$(\mathbf{u}, \mathbf{v}) = -\tau(\boldsymbol{L}\nabla\mathbf{w}, \nabla\mathbf{v}) + (\mathbf{u}^{\text{old}}, \mathbf{v}) \quad \forall \mathbf{v} \in H^1(\Omega)^N, \tag{4.38}$$

$$(\mathbf{w}, \mathbf{v}-\mathbf{u}) \leq \varepsilon^2(\nabla\mathbf{u}, \nabla(\mathbf{v}-\mathbf{u})) - (\boldsymbol{T}\mathbf{u}, \mathbf{v}-\mathbf{u}) + \frac{1}{N}\sum_{j=1}^{N}((\boldsymbol{T}\mathbf{u})_j\mathbf{1}, \mathbf{v}-\mathbf{u}) \tag{4.39}$$

$$\forall \mathbf{v} \in H^1(\Omega)^N, \mathbf{v} \in \mathcal{G}^N \text{ a.e. in } \Omega,$$

$$\mathbf{u} \in \mathcal{G}^N \quad \text{a.e. in } \Omega. \tag{4.40}$$

As in the smooth case, choosing $\mathbf{v} = \mathbf{e}_i$ in (4.38), we obtain the conservation of mass, i.e., $\int_\Omega \mathbf{u}\,\mathrm{d}\mathbf{x} = \mathbf{m} = [m_1, \ldots, m_N]^T$. In other words, $(\mathbf{u}, \mathbf{e}_i) = (\mathbf{u}^{\text{old}}, \mathbf{e}_i) = m_i$ for $i = 1, \ldots, N$, where $m_i \in (0,1)$ and $\sum_{i=1}^N m_i = 1$. Here, $\mathbf{e}_i$ is the function which is identical to one in the ith component and zero otherwise. Without loss of generality, we assume that $|\Omega| = 1$, with $|\Omega|$ being the Lebesgue measure of Ω, holds true.

During the next three sections, we extend parts of the analysis presented in Chapters 3.3.2–3.5 to the multi-component case. We define

$$\mathcal{K} := \left\{\mathbf{v} = [v_1, \ldots, v_N]^T \in H^1(\Omega)^N \colon \mathbf{v}(\mathbf{x}) \geq \mathbf{0}, \sum_{i=1}^N v_i(\mathbf{x}) = 1 \text{ a.e. in } \Omega\right\},$$

$$\mathcal{V}_0 := \left\{\mathbf{v} = [v_1, \ldots, v_N]^T \in H^1(\Omega)^N \colon (v_i, 1) = 0,\ i = 1, \ldots, N,\ \sum_{i=1}^N v_i(\mathbf{x}) = 0 \text{ a.e. in } \Omega\right\},$$

and consider the following minimization problem

$$\min_{(\mathbf{u},\mathbf{w})\in\mathcal{K}\times\mathcal{V}_0} \quad \mathcal{J}(\mathbf{u}, \mathbf{w}) := \frac{\varepsilon^2}{2}\|\nabla\mathbf{u}\|^2 + \int_\Omega -\frac{1}{2}\mathbf{u}\cdot\boldsymbol{T}\mathbf{u} + \frac{\tau}{2}\nabla\mathbf{w}\cdot\boldsymbol{L}\nabla\mathbf{w}\,\mathrm{d}\mathbf{x} \qquad \text{subject to} \quad (4.38). \tag{$\mathcal{P}$}$$

Let

$$\mathcal{F} = \{(\mathbf{u}, \mathbf{w}) \in \mathcal{K}\times\mathcal{V}_0 \colon (\mathbf{u}, \mathbf{w}) \text{ achieves } (4.38)\}$$

be the admissible set of ($\mathcal{P}$). Analogous to Lemma 3.3, we have the following result.

Lemma 4.3. *The following properties hold true:*

(i) $\mathcal{F} \neq \emptyset$ *and* $\mathcal{F} \subset (\mathcal{V}_0 + \mathbf{m}) \times \mathcal{V}_0$.

(ii) $\mathcal{F}$ *is a closed and convex set of* $H^1(\Omega)^N \times H^1(\Omega)^N$.

(iii) Let $\tau < 4\varepsilon^2\lambda_{\max}(\boldsymbol{T})^{-2}\rho(\boldsymbol{L})^{-1}$. *Then,* $\mathcal{J}$ *is strictly convex on* $\mathcal{F}$.

(iv) Let $\tau < 4\varepsilon^2\lambda_{\max}(\boldsymbol{T})^{-2}\rho(\boldsymbol{L})^{-1}$. *Then, we have* $\lim_{m\to\infty}\mathcal{J}(\mathbf{u}_m, \mathbf{w}_m) = \infty$ *for every sequence* $(\mathbf{u}_m, \mathbf{w}_m)_{m\in\mathbb{N}}$ *in* $\mathcal{F}$ *such that* $\lim_{m\to\infty}\|\mathbf{u}_m\|_1 = \infty$ *or* $\lim_{m\to\infty}\|\mathbf{w}_m\|_1 = \infty$.

Proof. (i) $\mathcal{F} \neq \emptyset$ since $(\mathbf{u}^{\text{old}}, \mathbf{0}) \in \mathcal{F}$. Let $(\mathbf{u}, \mathbf{w}) \in \mathcal{F}$. It follows $\mathbf{w} \in \mathcal{V}_0$. By taking $\mathbf{v} = \mathbf{e}_i$ in (4.38), we obtain $(u_i, 1) = (\mathbf{u}, \mathbf{e}_i) = (\mathbf{u}^{\text{old}}, \mathbf{e}_i) = (u_i^{\text{old}}, 1) = m_i$. Hence, $\mathbf{u} - \mathbf{m} \in \mathcal{V}_0$.

(ii) First, we proof that $\mathcal{F}$ is convex. Let $(\mathbf{u}, \mathbf{w}), (\mathbf{y}, \mathbf{z}) \in \mathcal{F}$ with $\mathbf{u} = [u_1, \dots, u_N]^T$, $\mathbf{w} = [w_1, \dots, w_N]^T$, $\mathbf{y} = [y_1, \dots, y_N]^T$, $\mathbf{z} = [z_1, \dots, z_N]^T$ and $\lambda \in [0, 1]$. We have to show that $(\lambda\mathbf{u} + (1-\lambda)\mathbf{y}, \lambda\mathbf{w} + (1-\lambda)\mathbf{z}) \in \mathcal{F}$ holds true. From

$$(\lambda\mathbf{w} + (1-\lambda)\mathbf{z}, \mathbf{1}) = \lambda \underbrace{(\mathbf{w}, \mathbf{1})}_{=\mathbf{0}} + (1-\lambda) \underbrace{(\mathbf{z}, 1)}_{=\mathbf{0}} = \mathbf{0}$$

and

$$\sum_{i=1}^{N} \lambda w_i + (1-\lambda) z_i = \lambda \underbrace{\sum_{i=1}^{N} w_i}_{=0 \text{ a.e. in } \Omega} + (1-\lambda) \underbrace{\sum_{i=1}^{N} z_i}_{=0 \text{ a.e. in } \Omega} = 0 \text{ a.e. in } \Omega,$$

it follows $\lambda\mathbf{w} + (1-\lambda)\mathbf{z} \in \mathcal{V}_0$. Further,

$$\lambda\mathbf{u} + (1-\lambda)\mathbf{y} \geq \mathbf{0} \text{ a.e. in } \Omega,$$

since $\mathbf{u}, \mathbf{y} \geq \mathbf{0}$ a.e. in Ω, and

$$\sum_{i=1}^{N} \lambda u_i + (1-\lambda) y_i = \lambda \underbrace{\sum_{i=1}^{N} u_i}_{=1 \text{ a.e. in } \Omega} + (1-\lambda) \underbrace{\sum_{i=1}^{N} y_i}_{=1 \text{ a.e. in } \Omega} = 1 \text{ a.e. in } \Omega$$

and hence $\lambda\mathbf{u} + (1-\lambda)\mathbf{y} \in \mathcal{K}$. Finally,

$$\begin{aligned}
&(\lambda\mathbf{u} + (1-\lambda)\mathbf{y}, \mathbf{v}) + \tau(\boldsymbol{L}\nabla(\lambda\mathbf{w} + (1-\lambda)\mathbf{z}), \nabla\mathbf{v}) \\
&\qquad = \lambda\underbrace{[(\mathbf{u}, \mathbf{v}) + \tau(\boldsymbol{L}\nabla\mathbf{w}, \nabla\mathbf{v})]}_{=(\mathbf{u}^{\text{old}}, \mathbf{v})} + (1-\lambda)\underbrace{[(\mathbf{y}, \mathbf{v}) + \tau(\boldsymbol{L}\nabla\mathbf{z}, \nabla\mathbf{v})]}_{=(\mathbf{u}^{\text{old}}, \mathbf{v})} \\
&\qquad = (\mathbf{u}^{\text{old}}, \mathbf{v}) \quad \forall \mathbf{v} \in H^1(\Omega)^N,
\end{aligned}$$

which means that $(\lambda\mathbf{u} + (1-\lambda)\mathbf{y}, \lambda\mathbf{w} + (1-\lambda)\mathbf{z})$ fulfills (4.38). Altogether, $\mathcal{F}$ is convex.

Now, let us proof the closedness of $\mathcal{F}$ in $H^1(\Omega)^N \times H^1(\Omega)^N$. Let $(\mathbf{u}_m, \mathbf{w}_m)_{m \in \mathbb{N}} \subset \mathcal{F}$ converge strongly to $(\mathbf{u}, \mathbf{w}) \in H^1(\Omega)^N \times H^1(\Omega)^N$ for $m \to \infty$, whereby $\mathbf{u}_m = [u_{m,1}, \dots, u_{m,N}]^T$, $\mathbf{w}_m = [w_{m,1}, \dots, w_{m,N}]^T$, $\mathbf{u} = [u_1, \dots, u_N]^T$, $\mathbf{w} = [w_1, \dots, w_N]^T$. We have to show that $(\mathbf{u}, \mathbf{w}) \in \mathcal{F}$. According to Theorem 2.6, every strongly convergent sequence is weakly convergent, i.e.,

$$\begin{aligned}
&(\mathbf{u}_m, \mathbf{v})_1 \overset{m\to\infty}{\longrightarrow} (\mathbf{u}, \mathbf{v})_1 \quad \forall \mathbf{v} \in H^1(\Omega)^N \\
\Leftrightarrow\ &(\mathbf{u}_m, \mathbf{v}) + (\nabla\mathbf{u}_m, \nabla\mathbf{v}) \overset{m\to\infty}{\longrightarrow} (\mathbf{u}, \mathbf{v}) + (\nabla\mathbf{u}, \nabla\mathbf{v}) \quad \forall \mathbf{v} \in H^1(\Omega)^N
\end{aligned}$$

as well as

$$\begin{aligned}
&(\mathbf{w}_m, \mathbf{v})_1 \overset{m\to\infty}{\longrightarrow} (\mathbf{w}, \mathbf{v})_1 \quad \forall \mathbf{v} \in H^1(\Omega)^N \\
\Leftrightarrow\ &(\mathbf{w}_m, \mathbf{v}) + (\nabla\mathbf{w}_m, \nabla\mathbf{v}) \overset{m\to\infty}{\longrightarrow} (\mathbf{w}, \mathbf{v}) + (\nabla\mathbf{w}, \nabla\mathbf{v}) \quad \forall \mathbf{v} \in H^1(\Omega)^N.
\end{aligned}$$

Hence, we obtain

$$(\mathbf{u},\mathbf{v}) + \tau(\boldsymbol{L}\nabla\mathbf{w},\nabla\mathbf{v}) = (\mathbf{u}^{\text{old}},\mathbf{v}) \quad \forall\mathbf{v}\in H^1(\Omega)^N$$

and

$$(\mathbf{w},\mathbf{e}_i) = 0$$

for $i = 1,\dots,N$. What is left to show is that $\mathbf{u}\in\mathcal{K}$ and $\mathbf{w}\in\mathcal{V}_0$. As stated in [21, p. 238], $\mathcal{K}$ is convex and closed in $H^1(\Omega)^N$. According to Lemma 2.7, $\mathcal{K}$ is weakly closed in $H^1(\Omega)^N$. Hence, Definition 2.9 yields $\mathbf{u}\in\mathcal{K}$. Similarly, $\mathcal{V}_0$ is convex and closed in $H^1(\Omega)^N$. Altogether, $(\mathbf{u},\mathbf{w})\in\mathcal{F}$.

(iii) Let $(\mathbf{u},\mathbf{w}),(\mathbf{y},\mathbf{z})\in\mathcal{F}$ and $\alpha\in(0,1)$. We define

$$r(\alpha) := \alpha\mathcal{J}(\mathbf{u},\mathbf{w}) + (1-\alpha)\mathcal{J}(\mathbf{y},\mathbf{z}) - \mathcal{J}\left(\alpha\mathbf{u}+(1-\alpha)\mathbf{y},\alpha\mathbf{w}+(1-\alpha)\mathbf{z}\right).$$

We have to show $r(\alpha) > 0$. We start with proving $r(\alpha)\geq 0$. It holds

$$\begin{aligned} r(\alpha) &= \alpha\left(\frac{\varepsilon^2}{2}\|\nabla\mathbf{u}\|^2 + \frac{\tau}{2}\left(\nabla\mathbf{w},\boldsymbol{L}\nabla\mathbf{w}\right) - \frac{1}{2}\left(\mathbf{u},\boldsymbol{T}\mathbf{u}\right)\right) - \frac{\varepsilon^2}{2}\|\nabla(\alpha\mathbf{u}+(1-\alpha)\mathbf{y})\|^2 \\ &\quad + (1-\alpha)\left(\frac{\varepsilon^2}{2}\|\nabla\mathbf{y}\|^2 + \frac{\tau}{2}\left(\nabla\mathbf{z},\boldsymbol{L}\nabla\mathbf{z}\right) - \frac{1}{2}\left(\mathbf{y},\boldsymbol{T}\mathbf{y}\right)\right) \\ &\quad + \frac{1}{2}\left(\alpha\mathbf{u}+(1-\alpha)\mathbf{y},\boldsymbol{T}(\alpha\mathbf{u}+(1-\alpha)\mathbf{y})\right) \\ &\quad - \frac{\tau}{2}\left(\nabla(\alpha\mathbf{w}+(1-\alpha)\mathbf{z}),\boldsymbol{L}\nabla(\alpha\mathbf{w}+(1-\alpha)\mathbf{z})\right) \\ &= \alpha(1-\alpha)\left[\frac{\varepsilon^2}{2}\left(\|\nabla\mathbf{u}\|^2 + \left\|\nabla\mathbf{y}\right\|^2 - 2\left(\nabla\mathbf{u},\nabla\mathbf{y}\right)\right) - \frac{1}{2}\left(\left(\mathbf{u},\boldsymbol{T}\mathbf{u}\right) + \left(\mathbf{y},\boldsymbol{T}\mathbf{y}\right) - 2\left(\mathbf{u},\boldsymbol{T}\mathbf{y}\right)\right)\right. \\ &\qquad\qquad \left. + \frac{\tau}{2}\left(\left(\nabla\mathbf{w},\boldsymbol{L}\nabla\mathbf{w}\right) + \left(\nabla\mathbf{z},\boldsymbol{L}\nabla\mathbf{z}\right) - 2\left(\nabla\mathbf{w},\boldsymbol{L}\nabla\mathbf{z}\right)\right)\right] \\ &= \frac{\alpha(1-\alpha)}{2}\left(\varepsilon^2\|\nabla(\mathbf{u}-\mathbf{y})\|^2 + \tau\left(\nabla(\mathbf{w}-\mathbf{z}),\boldsymbol{L}\nabla(\mathbf{w}-\mathbf{z})\right) - \left(\mathbf{u}-\mathbf{y},\boldsymbol{T}(\mathbf{u}-\mathbf{y})\right)\right) \\ &= \frac{\alpha(1-\alpha)}{2}\left(\varepsilon^2\|\nabla(\mathbf{u}-\mathbf{y})\|^2 + \tau\left\|\boldsymbol{L}_s\nabla(\mathbf{w}-\mathbf{z})\right\|^2 - \left(\mathbf{u}-\mathbf{y},\boldsymbol{T}(\mathbf{u}-\mathbf{y})\right)\right). \end{aligned}$$

Since

$$(\mathbf{u}-\mathbf{y})^T\boldsymbol{T}(\mathbf{u}-\mathbf{y}) \leq \lambda_{\max}(\boldsymbol{T})(\mathbf{u}-\mathbf{y})^T(\mathbf{u}-\mathbf{y}),$$

we can bound $r(\alpha)$ from above to

$$r(\alpha) \geq \frac{\alpha(1-\alpha)}{2}\left(\varepsilon^2\|\nabla(\mathbf{u}-\mathbf{y})\|^2 + \tau\left(\nabla(\mathbf{w}-\mathbf{z}),\boldsymbol{L}\nabla(\mathbf{w}-\mathbf{z})\right) - \lambda_{\max}(\boldsymbol{T})\left\|\mathbf{u}-\mathbf{y}\right\|^2\right). \tag{4.41}$$

Since $(\mathbf{u},\mathbf{w}),(\mathbf{y},\mathbf{z})\in\mathcal{F}$, they satisfy (4.38)

$$(\mathbf{u},\mathbf{v}) + \tau(\boldsymbol{L}\nabla\mathbf{w},\nabla\mathbf{v}) = (\mathbf{u}^{\text{old}},\mathbf{v}) \quad \forall\mathbf{v}\in H^1(\Omega)^N, \tag{4.42}$$

$$(\mathbf{y},\mathbf{v}) + \tau(\boldsymbol{L}\nabla\mathbf{z},\nabla\mathbf{v}) = (\mathbf{u}^{\text{old}},\mathbf{v}) \quad \forall\mathbf{v}\in H^1(\Omega)^N. \tag{4.43}$$

Choosing $\mathbf{v} = \mathbf{u}-\mathbf{y}$ in (4.42)–(4.43), we obtain

$$(\mathbf{u},\mathbf{u}-\mathbf{y}) + \tau(\boldsymbol{L}\nabla\mathbf{w},\nabla(\mathbf{u}-\mathbf{y})) = (\mathbf{u}^{\text{old}},\mathbf{u}-\mathbf{y}), \tag{4.44}$$

$$(\mathbf{y},\mathbf{u}-\mathbf{y}) + \tau(\boldsymbol{L}\nabla\mathbf{z},\nabla(\mathbf{u}-\mathbf{y})) = (\mathbf{u}^{\text{old}},\mathbf{u}-\mathbf{y}). \tag{4.45}$$

Subtracting (4.44)–(4.45) from each other, we get

$$-\|\mathbf{u}-\mathbf{y}\|^2 = \tau(\boldsymbol{L}\nabla(\mathbf{w}-\mathbf{z}), \nabla(\mathbf{u}-\mathbf{y})). \tag{4.46}$$

Applying Young's inequality with $\alpha_{\mathrm{Y}} > 0$ (Lemma 2.12) to the right-hand side of (4.46), gives

$$\begin{aligned}-\|\mathbf{u}-\mathbf{y}\|^2 &= \tau\left(\sqrt{2}\boldsymbol{L}_s\nabla(\mathbf{w}-\mathbf{z}), \frac{1}{\sqrt{2}}\boldsymbol{L}_s\nabla(\mathbf{u}-\mathbf{y})\right)\\ &\geq -\tau\alpha_{\mathrm{Y}}\|\boldsymbol{L}_s\nabla(\mathbf{w}-\mathbf{z})\|^2 - \frac{\tau}{4\alpha_{\mathrm{Y}}}\|\boldsymbol{L}_s\nabla(\mathbf{u}-\mathbf{y})\|^2. \end{aligned}\tag{4.47}$$

Due to

$$\left\|\boldsymbol{L}_s\nabla(\mathbf{u}-\mathbf{y})\right\|^2 = (\boldsymbol{L}_s\nabla(\mathbf{u}-\mathbf{y}), \boldsymbol{L}_s\nabla(\mathbf{u}-\mathbf{y})) = (\boldsymbol{L}\nabla(\mathbf{u}-\mathbf{y}), \nabla(\mathbf{u}-\mathbf{y})) \leq \rho(\boldsymbol{L})\left\|\nabla(\mathbf{u}-\mathbf{y})\right\|^2,$$

we can estimate (4.47) further to

$$-\|\mathbf{u}-\mathbf{y}\|^2 \geq -\tau\alpha_{\mathrm{Y}}\|\boldsymbol{L}_s\nabla(\mathbf{w}-\mathbf{z})\|^2 - \frac{\tau\rho(\boldsymbol{L})}{4\alpha_{\mathrm{Y}}}\|\nabla(\mathbf{u}-\mathbf{y})\|^2. \tag{4.48}$$

Substituting (4.48) into (4.41) yields

$$r(\alpha) \geq \frac{\alpha(1-\alpha)}{2}\left[\left(\varepsilon^2 - \frac{\lambda_{\max}(\boldsymbol{T})\rho(\boldsymbol{L})\tau}{4\alpha_{\mathrm{Y}}}\right)\|\nabla(\mathbf{u}-\mathbf{y})\|^2 + (\tau - \tau\alpha_{\mathrm{Y}}\lambda_{\max}(\boldsymbol{T}))\,\|\boldsymbol{L}_s\nabla(\mathbf{w}-\mathbf{z})\|^2\right].$$

For the strict convexity, we require

$$\varepsilon^2 - \frac{\lambda_{\max}(\boldsymbol{T})\rho(\boldsymbol{L})\tau}{4\alpha_{\mathrm{Y}}} > 0 \quad\Leftrightarrow\quad \alpha_{\mathrm{Y}} > \frac{\lambda_{\max}(\boldsymbol{T})\rho(\boldsymbol{L})\tau}{4\varepsilon^2}, \tag{4.49}$$

$$\tau - \tau\alpha_{\mathrm{Y}}\lambda_{\max}(\boldsymbol{T}) > 0 \quad\Leftrightarrow\quad \alpha_{\mathrm{Y}} < \frac{1}{\lambda_{\max}(\boldsymbol{T})}. \tag{4.50}$$

Hence,

$$\frac{\lambda_{\max}(\boldsymbol{T})\rho(\boldsymbol{L})\tau}{4\varepsilon^2} < \alpha_{\mathrm{Y}} < \frac{1}{\lambda_{\max}(\boldsymbol{T})},$$

which leads to the time step restriction

$$\frac{\lambda_{\max}(\boldsymbol{T})\rho(\boldsymbol{L})\tau}{4\varepsilon^2} < \frac{1}{\lambda_{\max}(\boldsymbol{T})} \quad\Leftrightarrow\quad \tau < \frac{4\varepsilon^2}{\lambda_{\max}(\boldsymbol{T})^2\rho(\boldsymbol{L})}.$$

Under these conditions, we can estimate $r(\alpha)$ further by using

$$\|\boldsymbol{L}_s\nabla(\mathbf{w}-\mathbf{z})\|^2 = (\boldsymbol{L}\nabla(\mathbf{w}-\mathbf{z}), \nabla(\mathbf{w}-\mathbf{z})) \geq l_0\,(\boldsymbol{P}\nabla(\mathbf{w}-\mathbf{z}), \nabla(\mathbf{w}-\mathbf{z})) = l_0\,\|\nabla(\mathbf{w}-\mathbf{z})\|^2,$$

where the inequality results from (4.14), and the last equality is a result from $\sum_{i=1}^N w_i - z_i = 0$ a.e. in Ω. Hence, we obtain

$$r(\alpha) \geq \frac{\alpha(1-\alpha)}{2}\left[\left(\varepsilon^2 - \frac{\lambda_{\max}(\boldsymbol{T})\rho(\boldsymbol{L})\tau}{4\alpha_{\mathrm{Y}}}\right)\|\nabla(\mathbf{u}-\mathbf{y})\|^2 + l_0\,(\tau - \tau\alpha_{\mathrm{Y}}\lambda_{\max}(\boldsymbol{T}))\,\|\nabla(\mathbf{w}-\mathbf{z})\|^2\right].$$

Now, assume $r(\alpha) = 0$. Then,

$$\|\nabla(\mathbf{u}-\mathbf{y})\|^2 = \|\nabla(\mathbf{w}-\mathbf{z})\|^2 = 0. \tag{4.51}$$

Since $\mathbf{u}-\mathbf{y}, \mathbf{w}-\mathbf{z} \in \boldsymbol{\mathcal{V}}_0$, we have $\int_\Omega u_i - y_i \, \mathrm{d}\mathbf{x} = \int_\Omega w_i - z_i \, \mathrm{d}\mathbf{x} = 0$ for $i = 1, \dots, N$. Hence, we can apply the Poincaré inequality (Theorem 2.22)

$$\begin{aligned} \|u_i - y_i\|_1^2 \le c_P \, \|\nabla(u_i - y_i)\|^2 \quad &\Leftrightarrow \quad \|u_i - y_i\|^2 + \|\nabla(u_i - y_i)\|^2 \ \le c_P \, \|\nabla(u_i - y_i)\|^2, \\ \|w_i - z_i\|_1^2 \le c_P \, \|\nabla(w_i - z_i)\|^2 \quad &\Leftrightarrow \quad \|w_i - z_i\|^2 + \|\nabla(w_i - z_i)\|^2 \ \le c_P \, \|\nabla(w_i - z_i)\|^2, \end{aligned}$$

for $i = 1, \dots, N$. Hence,

$$\begin{aligned} \|\mathbf{u}-\mathbf{y}\|^2 + \|\nabla(\mathbf{u}-\mathbf{y})\|^2 &\le c_P \, \|\nabla(\mathbf{u}-\mathbf{y})\|^2, \\ \|\mathbf{w}-\mathbf{z}\|^2 + \|\nabla(\mathbf{w}-\mathbf{z})\|^2 &\le c_P \, \|\nabla(\mathbf{w}-\mathbf{z})\|^2. \end{aligned}$$

It follows from (4.51) that $\|\mathbf{u}-\mathbf{y}\| = \|\mathbf{w}-\mathbf{z}\| = 0$. This implies $(\mathbf{u}, \mathbf{w}) = (\mathbf{y}, \mathbf{z})$ a.e. in Ω. In summary, $\mathcal{J}$ is strictly convex on $\mathcal{F}$ provided that $\tau < 4\varepsilon^2 \lambda_{\max}(\boldsymbol{T})^{-2} \rho(\boldsymbol{L})^{-1}$.

(iv) Let $(\mathbf{u}, \mathbf{w}) \in \mathcal{F}$. Using

$$\mathbf{u}^T \boldsymbol{T} \mathbf{u} \le \lambda_{\max}(\boldsymbol{T}) \mathbf{u}^T \mathbf{u},$$

we obtain

$$\mathcal{J}(\mathbf{u}, \mathbf{w}) = \frac{\varepsilon^2}{2} \|\nabla \mathbf{u}\|^2 + \frac{\tau}{2} (\nabla \mathbf{w}, \boldsymbol{L} \nabla \mathbf{w}) - \frac{1}{2} (\mathbf{u}, \boldsymbol{T} \mathbf{u}) \ge \frac{\varepsilon^2}{2} \|\nabla \mathbf{u}\|^2 + \frac{\tau}{2} \|\boldsymbol{L}_s \nabla \mathbf{w}\|^2 - \frac{\lambda_{\max}(\boldsymbol{T})}{2} \|\mathbf{u}\|^2.$$

Since $(\mathbf{u}, \mathbf{w}) \in \mathcal{F}$, (4.38) is fulfilled. Choosing $\mathbf{v} = \mathbf{u}$ in (4.38) leads to

$$-\|\mathbf{u}\|^2 = \tau(\boldsymbol{L} \nabla \mathbf{w}, \nabla \mathbf{u}) - (\mathbf{u}^{\text{old}}, \mathbf{u}) = \tau(\boldsymbol{L}_s \nabla \mathbf{w}, \boldsymbol{L}_s \nabla \mathbf{u}) - (\mathbf{u}^{\text{old}}, \mathbf{u}). \tag{4.52}$$

Applying Young's inequality with $\alpha_Y = 2\beta_1 > 0$ (Lemma 2.12) to the left term in the right-hand side of (4.52) and with $\alpha_Y = 2\beta_2 > 0$ to the right term in the right-hand side of (4.52), we get

$$\begin{aligned} -\|\mathbf{u}\|^2 &= \tau(\boldsymbol{L}_s \nabla \mathbf{w}, \boldsymbol{L}_s \nabla \mathbf{u}) - (\mathbf{u}^{\text{old}}, \mathbf{u}) \\ &\ge -\tau\beta_1 \|\boldsymbol{L}_s \nabla \mathbf{w}\|^2 - \frac{\tau}{4\beta_1} \|\boldsymbol{L}_s \nabla \mathbf{u}\|^2 - \beta_2 \|\mathbf{u}^{\text{old}}\|^2 - \frac{1}{4\beta_2} \|\mathbf{u}\|^2. \end{aligned} \tag{4.53}$$

Due to

$$\|\boldsymbol{L}_s \nabla \mathbf{u}\|^2 = (\boldsymbol{L}_s \nabla \mathbf{u}, \boldsymbol{L}_s \nabla \mathbf{u}) = (\boldsymbol{L} \nabla \mathbf{u}, \nabla \mathbf{u}) \le \rho(\boldsymbol{L}) \, \|\nabla \mathbf{u}\|^2,$$

we can estimate (4.53) further to

$$-\|\mathbf{u}\|^2 \ge -\tau\beta_1 \|\boldsymbol{L}_s \nabla \mathbf{w}\|^2 - \frac{\tau\rho(\boldsymbol{L})}{4\beta_1} \|\nabla \mathbf{u}\|^2 - \beta_2 \|\mathbf{u}^{\text{old}}\|^2 - \frac{1}{4\beta_2} \|\mathbf{u}\|^2. \tag{4.54}$$

Applying the Poincaré inequality (Theorem 2.22) yields

$$\|u_i\|_1^2 = \|u_i\|^2 + \|\nabla u_i\|^2 \le c_{\mathrm{P}} \left(\|\nabla u_i\|^2 + \left(\int_\Omega u_i \, \mathrm{d}\mathbf{x} \right)^2 \right) = c_{\mathrm{P}} \left(\|\nabla u_i\|^2 + m_i^2 \right) \tag{4.55}$$

and hence

$$\|u_i\|^2 \le c_{\mathrm{P}} \left(\|\nabla u_i\|^2 + m_i^2 \right) - \|\nabla u_i\|^2 \le c_{\mathrm{P}} \left(\|\nabla u_i\|^2 + m_i^2 \right)$$

for $i = 1, \dots, N$, which leads to

$$\|\mathbf{u}\|^2 \le c_{\mathrm{P}} \|\nabla \mathbf{u}\|^2 + c_{\mathrm{P}} \, \mathbf{m}^T \mathbf{m}.$$

Hence, we can estimate (4.54) further to

$$-\|\mathbf{u}\|^2 \geq -\tau\beta_1\|\boldsymbol{L}_s\nabla\mathbf{w}\|^2 - \left(\frac{\tau\rho(\boldsymbol{L})}{4\beta_1} + \frac{c_P}{4\beta_2}\right)\|\nabla\mathbf{u}\|^2 - \beta_2\|\mathbf{u}^{\text{old}}\|^2 - \frac{c_P}{4\beta_2}\mathbf{m}^T\mathbf{m}. \tag{4.56}$$

Substituting this result into the equation of $\mathcal{J}(\mathbf{u},\mathbf{w})$ above, we get

$$\begin{aligned}\mathcal{J}(\mathbf{u},\mathbf{w}) \geq &\left(\frac{\varepsilon^2}{2} - \frac{\lambda_{\max}(\boldsymbol{T})\tau\rho(\boldsymbol{L})}{8\beta_1} - \frac{\lambda_{\max}(\boldsymbol{T})c_P}{8\beta_2}\right)\|\nabla\mathbf{u}\|^2 \\ &+ \left(\frac{\tau}{2} - \frac{\lambda_{\max}(\boldsymbol{T})\tau\beta_1}{2}\right)\|\boldsymbol{L}_s\nabla\mathbf{w}\|^2 - \frac{\lambda_{\max}(\boldsymbol{T})\beta_2}{2}\|\mathbf{u}^{\text{old}}\|^2 - \frac{\lambda_{\max}(\boldsymbol{T})c_P}{8\beta_2}\mathbf{m}^T\mathbf{m}.\end{aligned}$$

This inequality holds for all $\beta_1, \beta_2 > 0$. Now, we want to choose β_1, β_2 such that

$$\frac{\varepsilon^2}{2} - \frac{\lambda_{\max}(\boldsymbol{T})\tau\rho(\boldsymbol{L})}{8\beta_1} - \frac{\lambda_{\max}(\boldsymbol{T})c_P}{8\beta_2} > 0 \quad \Leftrightarrow \quad \frac{\varepsilon^2}{2} - \frac{\lambda_{\max}(\boldsymbol{T})\tau\rho(\boldsymbol{L})}{8\beta_1} > \frac{\lambda_{\max}(\boldsymbol{T})c_P}{8\beta_2}, \tag{4.57}$$

$$\frac{\tau}{2} - \frac{\lambda_{\max}(\boldsymbol{T})\tau\beta_1}{2} > 0 \quad \Leftrightarrow \quad \beta_1 < \frac{1}{\lambda_{\max}(\boldsymbol{T})}. \tag{4.58}$$

Since $\beta_2 > 0$, we need in (4.57)

$$\frac{\varepsilon^2}{2} - \frac{\lambda_{\max}(\boldsymbol{T})\tau\rho(\boldsymbol{L})}{8\beta_1} > 0 \quad \Leftrightarrow \quad \beta_1 > \frac{\lambda_{\max}(\boldsymbol{T})\tau\rho(\boldsymbol{L})}{4\varepsilon^2}.$$

Hence,

$$\frac{\lambda_{\max}(\boldsymbol{T})\tau\rho(\boldsymbol{L})}{4\varepsilon^2} < \beta_1 < \frac{1}{\lambda_{\max}(\boldsymbol{T})},$$

which leads to the time step restriction

$$\frac{\lambda_{\max}(\boldsymbol{T})\tau\rho(\boldsymbol{L})}{4\varepsilon^2} < \frac{1}{\lambda_{\max}(\boldsymbol{T})} \quad \Leftrightarrow \quad \tau < \frac{4\varepsilon^2}{\lambda_{\max}(\boldsymbol{T})^2\rho(\boldsymbol{L})}.$$

Under these conditions for τ and β_1, we can choose β_2 such that (4.57) is fulfilled. Now, we can estimate $\mathcal{J}(\mathbf{u},\mathbf{w})$ further by using

$$\|\boldsymbol{L}_s\nabla\mathbf{w}\|^2 = (\boldsymbol{L}\nabla\mathbf{w}, \nabla\mathbf{w}) \geq l_0\,(\boldsymbol{P}\nabla\mathbf{w}, \nabla\mathbf{w}) = l_0\,\|\nabla\mathbf{w}\|^2\,,$$

where the last equality is a result from $\sum_{i=1}^N w_i = 0$ a.e. in Ω. Hence, we obtain

$$\begin{aligned}\mathcal{J}(\mathbf{u},\mathbf{w}) \geq &\left(\frac{\varepsilon^2}{2} - \frac{\lambda_{\max}(\boldsymbol{T})\tau\rho(\boldsymbol{L})}{8\beta_1} - \frac{\lambda_{\max}(\boldsymbol{T})c_P}{8\beta_2}\right)\|\nabla\mathbf{u}\|^2 + l_0\left(\frac{\tau}{2} - \frac{\lambda_{\max}(\boldsymbol{T})\tau\beta_1}{2}\right)\|\nabla\mathbf{w}\|^2 \\ &- \frac{\lambda_{\max}(\boldsymbol{T})\beta_2}{2}\|\mathbf{u}^{\text{old}}\|^2 - \frac{\lambda_{\max}(\boldsymbol{T})c_P}{8\beta_2}\mathbf{m}^T\mathbf{m}.\end{aligned}$$

Next, due to the Poincaré inequality stated in (4.55), which yields

$$\|\mathbf{u}\|_1^2 \leq c_P\left(\|\nabla\mathbf{u}\|^2 + \mathbf{m}^T\mathbf{m}\right), \tag{4.59}$$

it holds $\lim_{m\to\infty}\|\nabla\mathbf{u}_m\| = \infty$ for every sequence $\mathbf{u}_m \in \mathcal{K}$ with $\lim_{m\to\infty}\|\mathbf{u}_m\|_1 = \infty$. The same result is true if we replace $\mathbf{u}_m$ by $\mathbf{w}_m$: Since $\mathbf{w} \in \mathcal{V}_0$, the Poincaré inequality (Theorem 2.22) yields

$$\|\mathbf{w}\|_1^2 \leq c_P\|\nabla\mathbf{w}\|^2. \tag{4.60}$$

Therefore, it holds $\lim_{m\to\infty}\mathcal{J}(\mathbf{u}_m,\mathbf{w}_m) = \infty$ for every sequence $(\mathbf{u}_m,\mathbf{w}_m)_{m\in\mathbb{N}} \subset \mathcal{F}$ with $\lim_{m\to\infty}\|\mathbf{u}_m\|_1 = \infty$ or $\lim_{m\to\infty}\|\mathbf{w}_m\|_1 = \infty$, provided that $\tau < 4\varepsilon^2\lambda_{\max}(\boldsymbol{T})^{-2}\rho(\boldsymbol{L})^{-1}$. □

Remark 4.3. *The time step condition is the same as in [12].*

The existence and uniqueness of the solution of ($\mathcal{P}$) immediately follow from the previous lemma:

Theorem 4.4. *Let* $\tau \in (0, 4\varepsilon^2\lambda_{\max}(\boldsymbol{T})^{-2}\rho(\boldsymbol{L})^{-1})$*. Then, the problem* ($\mathcal{P}$) *has a unique solution* $(\mathbf{u}^*, \mathbf{w}^*)$*.*

After having analyzed the nonsmooth implicit time-discrete Cahn–Hilliard system, we want to tackle the solution of this problem. The presence of the variational inequalities in (4.39) makes this problem hard. As in Chapter 3.4, we make use of the Moreau–Yosida regularization technique.

4.4 Moreau–Yosida regularization

In Chapter 3.4, we have motivated the use of the Moreau–Yosida regularization. More precisely, we have incorporated the bound constraints $|u| \leq 1$ almost everywhere with this technique. In this section, we apply it to the bound constraints $\mathbf{u} \geq \mathbf{0}$ almost everywhere. We define

$$\mathcal{K}_1 := \left\{ \mathbf{v} = [v_1, \ldots, v_N]^T \in H^1(\Omega)^N : \sum_{i=1}^{N} v_i(\mathbf{x}) = 1 \text{ a.e. in } \Omega \right\}$$

and replace the optimization problem ($\mathcal{P}$) by its Moreau–Yosida regularized version

$$\min_{(\mathbf{u},\mathbf{w}) \in \mathcal{K}_1 \times \mathcal{V}_0} \mathcal{J}_c(\mathbf{u}, \mathbf{w}) \quad \text{subject to (4.38)} \qquad (\mathcal{P}_c)$$

with the objective

$$\mathcal{J}_c(\mathbf{u}, \mathbf{w}) = \mathcal{J}(\mathbf{u}, \mathbf{w}) + \frac{1}{2c}\|\min(\mathbf{0}, \mathbf{u})\|^2.$$

As before, $0 < c \ll 1$ denotes the associated regularization or penalty parameter. Note that the constraint $\mathbf{u} \in \mathcal{K}$ in ($\mathcal{P}$) has been relaxed to $\mathbf{u} \in \mathcal{K}_1$ in the regularized problem ($\mathcal{P}_c$). At the same time, a damped version of the bound constraints in $\mathcal{K}$ has been inserted into the objective function. The smaller c is the larger is the penalization for the violation of the condition $\mathbf{u} \geq \mathbf{0}$. Hence, the limit $c \to 0$ represents the original minimization problem ($\mathcal{P}$). Indeed, this convergence is proven below in Proposition 4.6.

Remark 4.4. *Note that we still have the condition* $\sum_{i=1}^N v_i(\mathbf{x}) = 1$ a.e. in Ω *in* $\mathcal{K}_1$*. We will include it in the same manner as in the smooth case in Section 4.2.1. More precisely, we will incorporate it via the variational derivative at the end of this section.*

Analyzing ($\mathcal{P}_c$), we obtain a result similar to Theorem 4.4.

Theorem 4.5. *Let* $\tau \in (0, 4\varepsilon^2\lambda_{\max}(\boldsymbol{T})^{-2}\rho(\boldsymbol{L})^{-1})$*. Then, the problem* ($\mathcal{P}_c$) *has a unique solution* $(\mathbf{u}_c, \mathbf{w}_c)$*.*

Proof. As noted in the proof of Theorem 3.5, the functionals $u_i \to \|\min(0, u_i)\|^2, i = 1, \ldots, N$, are convex and Fréchet-differentiable on $H^1(\Omega)$. We can show that $\mathcal{F}_c$, the feasible set of ($\mathcal{P}_c$), and $\mathcal{J}_c$ satisfy the analogue of Lemma 4.3 for ($\mathcal{P}_c$). In fact, the proof is exactly the same. Hence, ($\mathcal{P}_c$) has a unique solution $(\mathbf{u}_c, \mathbf{w}_c)$, provided that $\tau \in (0, 4\varepsilon^2\lambda_{\max}(\boldsymbol{T})^{-2}\rho(\boldsymbol{L})^{-1})$. □

Proposition 4.6. *Let $\tau \in (0, 4\varepsilon^2 \lambda_{\max}(\boldsymbol{T})^{-2} \rho(\boldsymbol{L})^{-1})$. Let $\{(\mathbf{u}_c, \mathbf{w}_c)\}_{c>0}$ be a sequence of solutions of ($\mathcal{P}_c$) as $c \to 0$. Then, there exists a subsequence still denoted by $\{(\mathbf{u}_c, \mathbf{w}_c)\}_{c>0}$ such that*

$$(\mathbf{u}_c, \mathbf{w}_c) \longrightarrow (\mathbf{u}^*, \mathbf{w}^*) \text{ in } H^1(\Omega)^N \times H^1(\Omega)^N \tag{4.61}$$

as $c \to 0$, where $(\mathbf{u}^*, \mathbf{w}^*)$ is the unique solution of ($\mathcal{P}$).

Proof. First of all,

$$\mathcal{J}(\mathbf{u}_c, \mathbf{w}_c) \le \mathcal{J}_c(\mathbf{u}_c, \mathbf{w}_c) \le \mathcal{J}_c(\mathbf{u}^*, \mathbf{w}^*) = \mathcal{J}(\mathbf{u}^*, \mathbf{w}^*). \tag{4.62}$$

Hence, there exists a constant $\beta > 0$ independent of c such that

$$\mathcal{J}_c(\mathbf{u}_c, \mathbf{w}_c) \le \beta.$$

Next, we estimate $\mathcal{J}_c(\mathbf{u}_c, \mathbf{w}_c)$ from below. As in the proof of Lemma 4.3(iv) we obtain with $\tau < 4\varepsilon^2 \lambda_{\max}(\boldsymbol{T})^{-2} \rho(\boldsymbol{L})^{-1}$ and suitable $\beta_1, \beta_2 > 0$

$$\begin{aligned}\mathcal{J}_c(\mathbf{u}_c, \mathbf{w}_c) \ge \underbrace{\left(\frac{\varepsilon^2}{2} - \frac{\lambda_{\max}(\boldsymbol{T})\tau\rho(\boldsymbol{L})}{8\beta_1} - \frac{\lambda_{\max}(\boldsymbol{T})c_{\mathrm{P}}}{8\beta_2}\right)}_{>0} \|\nabla \mathbf{u}_c\|^2 + l_0 \underbrace{\left(\frac{\tau}{2} - \frac{\lambda_{\max}(\boldsymbol{T})\tau\beta_1}{2}\right)}_{>0} \|\nabla \mathbf{w}_c\|^2 \\ - \frac{\lambda_{\max}(\boldsymbol{T})\beta_2}{2}\|\mathbf{u}^{\text{old}}\|^2 - \frac{\lambda_{\max}(\boldsymbol{T})c_{\mathrm{P}}}{8\beta_2}\mathbf{m}^T\mathbf{m} + \frac{1}{2}\left\|\frac{1}{\sqrt{c}}\min(0, \mathbf{u}_c)\right\|^2.\end{aligned}$$

Next, due to the Poincaré inequality stated in (4.59) and (4.60), i.e.,

$$\begin{aligned}\|\mathbf{u}_c\|_1^2 &\le c_{\mathrm{P}}\left(\|\nabla \mathbf{u}_c\|^2 + \mathbf{m}^T\mathbf{m}\right), \\ \|\mathbf{w}_c\|_1^2 &\le c_{\mathrm{P}}\|\nabla \mathbf{w}_c\|^2,\end{aligned}$$

we can estimate $\mathcal{J}_c(\mathbf{u}_c, \mathbf{w}_c)$ further to

$$\mathcal{J}_c(\mathbf{u}_c, \mathbf{w}_c) \ge C_1\|\nabla \mathbf{u}_c\|^2 + C_2\|\nabla \mathbf{w}_c\|^2 - C_3\mathbf{m}^T\mathbf{m} - \frac{\lambda_{\max}(\boldsymbol{T})\beta_2}{2}\|\mathbf{u}^{\text{old}}\|^2 + \frac{1}{2}\left\|\frac{1}{\sqrt{c}}\min(0, \mathbf{u}_c)\right\|^2.$$

In order to ease the notation, we have introduced the constants $C_1, C_2, C_3 > 0$, which depend on ε, τ, β_1, β_2, c_{P}, $\lambda_{\max}(\boldsymbol{T})$, $\rho(\boldsymbol{L})$. This results in

$$\begin{aligned}\{\mathbf{u}_c\} &\text{ bounded in } H^1(\Omega)^N, \\ \{\mathbf{w}_c\} &\text{ bounded in } H^1(\Omega)^N, \\ \left\{\frac{1}{\sqrt{c}}\min(0, \mathbf{u}_c)\right\} &\text{ bounded in } L^2(\Omega)^N.\end{aligned} \tag{4.63}$$

Since $\{(\mathbf{u}_c, \mathbf{w}_c)\}_{c>0}$ is a bounded sequence in the Hilbert space $H^1(\Omega)^N \times H^1(\Omega)^N$, it has a weakly convergent subsequence. Hence, there exists a $(\mathbf{u}, \mathbf{w}) \in H^1(\Omega)^N \times H^1(\Omega)^N$ and a subsequence $\{(\mathbf{u}_{c_m}, \mathbf{w}_{c_m})\}_{m\in\mathbb{N}} \subset H^1(\Omega)^N \times H^1(\Omega)^N$ such that

$$(\mathbf{u}_{c_m}, \mathbf{w}_{c_m}) \overset{m\to\infty}{\rightharpoonup} (\mathbf{u}, \mathbf{w}) \quad \text{in } H^1(\Omega)^N \times H^1(\Omega)^N.$$

Because of the compact embedding $H^1(\Omega)^N \hookrightarrow L^2(\Omega)^N$, weakly convergent sequences in $H^1(\Omega)^N$ are strongly convergent in $L^2(\Omega)^N$, i.e.,

$$(\mathbf{u}_{c_m}, \mathbf{w}_{c_m}) \overset{m\to\infty}{\rightarrow} (\mathbf{u}, \mathbf{w}) \quad \text{in } L^2(\Omega)^N \times L^2(\Omega)^N. \tag{4.64}$$

According to Proposition 2.10, the strong convergence in $L^2(\Omega)^N$ implies

$$\|\mathbf{u}\|^2 \geq \limsup_{m\to\infty} \|\mathbf{u}_{c_m}\|^2 \geq \liminf_{m\to\infty} \|\mathbf{u}_{c_m}\|^2. \tag{4.65}$$

According to the proof of Lemma 4.3(ii), the weak convergence of $\{(\mathbf{u}_{c_m}, \mathbf{w}_{c_m})\}_{m\in\mathbb{N}}$ in $H^1(\Omega)^N \times H^1(\Omega)^N$ implies

$$(\mathbf{u}, \mathbf{v}) + \tau(\boldsymbol{L}\nabla\mathbf{w}, \nabla\mathbf{v}) = (\mathbf{u}^{\text{old}}, \mathbf{v}) \quad \forall \mathbf{v} \in H^1(\Omega)^N$$

and

$$(\mathbf{w}, \mathbf{e}_i) = 0$$

for $i = 1, \dots, N$. Moreover, $\mathcal{K}_1$ and $\mathcal{V}_0$ are weakly closed in $H^1(\Omega)^N$. Hence, Definition 2.9 yields $(\mathbf{u}, \mathbf{w}) \in \mathcal{K}_1 \times \mathcal{V}_0$. From (4.64) and Lebesgue's dominated convergence theorem, it follows

$$\min(0, \mathbf{u}_{c_m}) \overset{m\to\infty}{\longrightarrow} \min(0, \mathbf{u}) \quad \text{in } L^2(\Omega)^N.$$

Together with (4.63), this yields

$$\mathbf{u} \geq \mathbf{0} \quad \text{a.e. in } \Omega. \tag{4.66}$$

Hence, $(\mathbf{u}, \mathbf{w}) \in \mathcal{F}$. Now, consider

$$\mathcal{J}(\mathbf{u}, \mathbf{w}) = \frac{\varepsilon^2}{2}\|\nabla\mathbf{u}\|^2 - \frac{1}{2}(\mathbf{u}, \boldsymbol{T}\mathbf{u}) + \frac{\tau}{2}(\nabla\mathbf{w}, \boldsymbol{L}\nabla\mathbf{w}). \tag{4.67}$$

In the following, we estimate each term on the right-hand side of (4.67) from above. First, it holds that the $H^1(\Omega)^N$-seminorm

$$|\mathbf{v}|_1^2 = \|\nabla\mathbf{v}\|^2$$

is weakly lower semicontinuous. Hence, according to Definition 2.10, the weak convergence of a sequence $\mathbf{v}_m \rightharpoonup \mathbf{v}$ in $H^1(\Omega)^N$ for $m \to \infty$ implies $|\mathbf{v}|_1 \leq \liminf_{m\to\infty} |\mathbf{v}_m|_1$. Second, Lemma 2.11 together with the strong convergence $\mathbf{u}_{c_m} \overset{m\to\infty}{\longrightarrow} \mathbf{u}$ in $L^2(\Omega)^N$ yields

$$(\mathbf{u}_{c_m}, \boldsymbol{T}\mathbf{u}_{c_m}) = \sum_{i=1}^{N} [\boldsymbol{T}]_{ij}(u_{c_m,i}, u_{c_m,j}) \overset{m\to\infty}{\longrightarrow} \sum_{i=1}^{N} [\boldsymbol{T}]_{ij}(u_i, u_j) = (\mathbf{u}, \boldsymbol{T}\mathbf{u}),$$

where $\mathbf{u}_{c_m} = [u_{c_m,1}, \dots, u_{c_m,N}]^T$ and $\mathbf{u} = [u_1, \dots, u_N]^T$. Third, since $(\mathbf{u}, \mathbf{w}) \in \mathcal{F}$ and $(\mathbf{u}_{c_m}, \mathbf{w}_{c_m}) \in \mathcal{F}_c$, where $\mathcal{F}_c$ is the feasible set of $(\mathcal{P}_c)$, both pairs achieve (4.38). Hence, choosing $\mathbf{v} = \mathbf{w}$ for the former pair and $\mathbf{v} = \mathbf{w}_{c_m}$ for the latter, we get

$$\tau(\boldsymbol{L}\nabla\mathbf{w}, \nabla\mathbf{w}) = (\mathbf{u}^{\text{old}}, \mathbf{w}) - (\mathbf{u}, \mathbf{w}),$$
$$\tau(\boldsymbol{L}\nabla\mathbf{w}_{c_m}, \nabla\mathbf{w}_{c_m}) = (\mathbf{u}^{\text{old}}, \mathbf{w}_{c_m}) - (\mathbf{u}_{c_m}, \mathbf{w}_{c_m}).$$

Lemma 2.11 together with the strong convergence in (4.64) yields

$$\tau(\boldsymbol{L}\nabla\mathbf{w}_{c_m}, \nabla\mathbf{w}_{c_m}) \overset{m\to\infty}{\longrightarrow} (\mathbf{u}^{\text{old}}, \mathbf{w}) - (\mathbf{u}, \mathbf{w}) = \tau(\boldsymbol{L}\nabla\mathbf{w}, \nabla\mathbf{w}).$$

Therefore,

$$\begin{aligned}\mathcal{J}(\mathbf{u},\mathbf{w}) &\leq \frac{\varepsilon^2}{2}\liminf_{m\to\infty}\|\nabla\mathbf{u}_{c_m}\|^2 - \frac{1}{2}\liminf_{m\to\infty}(\mathbf{u}_{c_m},\boldsymbol{T}\mathbf{u}_{c_m}) + \frac{\tau}{2}\liminf_{m\to\infty}(\nabla\mathbf{w}_{c_m},\boldsymbol{L}\nabla\mathbf{w}_{c_m})\\ &\leq \liminf_{m\to\infty}\left(\frac{\varepsilon^2}{2}\|\nabla\mathbf{u}_{c_m}\|^2 - \frac{1}{2}(\mathbf{u}_{c_m},\boldsymbol{T}\mathbf{u}_{c_m}) + \frac{\tau}{2}(\nabla\mathbf{w}_{c_m},\boldsymbol{L}\nabla\mathbf{w}_{c_m})\right)\\ &= \liminf_{m\to\infty}\mathcal{J}(\mathbf{u}_{c_m},\mathbf{w}_{c_m}).\end{aligned}$$

Together with (4.62), we obtain

$$\mathcal{J}(\mathbf{u},\mathbf{w}) \leq \liminf_{m\to\infty}\mathcal{J}(\mathbf{u}_{c_m},\mathbf{w}_{c_m}) \leq \mathcal{J}(\mathbf{u}^*,\mathbf{w}^*). \tag{4.68}$$

The pair $(\mathbf{u}^*,\mathbf{w}^*)$ is the unique solution of $(\boldsymbol{\mathcal{P}})$. In contrast, $(\mathbf{u},\mathbf{w})$ is a feasible solution of $(\boldsymbol{\mathcal{P}})$. Hence, $\mathcal{J}(\mathbf{u},\mathbf{w}) \geq \mathcal{J}(\mathbf{u}^*,\mathbf{w}^*)$ and (4.68) becomes an equation. This gives $(\mathbf{u},\mathbf{w}) = (\mathbf{u}^*,\mathbf{w}^*)$.

It remains to show the strong convergence in (4.61). We have already proven the weak convergence. Hence, what is left to show is the following norm convergence:

$$\begin{aligned}\|\mathbf{u}_{c_m}\|_1 &\overset{m\to\infty}{\longrightarrow} \|\mathbf{u}^*\|_1,\\ \|\mathbf{w}_{c_m}\|_1 &\overset{m\to\infty}{\longrightarrow} \|\mathbf{w}^*\|_1.\end{aligned}$$

(4.62) and (4.68) imply

$$\frac{1}{2c_m}\|\min(\mathbf{0},\mathbf{u}_{c_m})\|^2 \overset{m\to\infty}{\longrightarrow} 0.$$

Thus,

$$\mathcal{J}(\mathbf{u}^*,\mathbf{w}^*) \leq \liminf_{m\to\infty}\mathcal{J}_{c_m}(\mathbf{u}_{c_m},\mathbf{w}_{c_m}) \leq \limsup_{m\to\infty}\mathcal{J}_{c_m}(\mathbf{u}_{c_m},\mathbf{w}_{c_m}) \leq \mathcal{J}(\mathbf{u}^*,\mathbf{w}^*).$$

This means

$$\mathcal{J}(\mathbf{u}^*,\mathbf{w}^*) = \lim_{m\to\infty}\mathcal{J}_{c_m}(\mathbf{u}_{c_m},\mathbf{w}_{c_m}),$$

and it follows

$$\lim_{m\to\infty}\|\nabla\mathbf{u}_{c_m}\| = \|\nabla\mathbf{u}^*\| \quad\text{and}\quad \lim_{m\to\infty}\|\nabla\mathbf{w}_{c_m}\| = \|\nabla\mathbf{w}^*\|.$$

From $\|\mathbf{u}_{c_m}\|_1^2 = \|\mathbf{u}_{c_m}\|^2 + \|\nabla\mathbf{u}_{c_m}\|^2$ and the strong convergence in (4.64), we imply the norm convergence of $\|\mathbf{u}_{c_m}\|_1^2$. The same holds for $\mathbf{w}_{c_m}$. The weak and norm convergence yield the strong convergence result (4.61). □

As already pointed out in Chapter 3.4, there is another way to explain the regularization. The multi-obstacle potential ψ_{obs} in (4.5) is regularized by

$$\psi_{\text{obs}}(\mathbf{u}) = \psi_0(\mathbf{u}) + \frac{1}{2c}\sum_{i=1}^{N}\min(0,u_i)^2.$$

Instead of the energy functional $\mathcal{E}$ in (4.3), we consider

$$\begin{aligned}\mathcal{E}_c(\mathbf{u}) &= \int_\Omega \frac{\varepsilon^2}{2}\sum_{i=1}^{N}|\nabla u_i|^2 + \psi_0(\mathbf{u}) + \frac{1}{2c}\sum_{i=1}^{N}|\min(0,u_i)|^2\,\mathrm{d}\mathbf{x}\\ &= \int_\Omega \frac{\varepsilon^2}{2}\sum_{i=1}^{N}|\nabla u_i|^2 - \frac{1}{2}\mathbf{u}^T\boldsymbol{T}\mathbf{u} + \frac{1}{2c}\sum_{i=1}^{N}|\min(0,u_i)|^2\,\mathrm{d}\mathbf{x}.\end{aligned}$$

As done for the smooth potential in Section 4.2.1, the vector of chemical potentials $\mathbf{w}$ is defined via the variational derivative of $\mathcal{E}_c$ with respect to $\mathbf{u}$ (see Definition 2.20), whereby

$$\mathcal{U} = \left\{ \mathbf{v} = [v_1, \dots, v_N]^T \in H^1(\Omega)^N \colon \sum_{i=1}^N v_i = 1 \right\},$$
$$\mathcal{Y} = \left\{ \mathbf{v} = [v_1, \dots, v_N]^T \in H^1(\Omega)^N \colon \sum_{i=1}^N v_i = 0 \right\}.$$

Note that for all $\mathbf{g} = [g_1, \dots, g_N]^T \in \mathcal{Y}$ there exists a vector $\mathbf{d} = [d_1, \dots, d_N]^T \in H^1(\Omega)^N$ such that

$$\mathbf{g} = \mathbf{d} - \frac{1}{N}\left(\sum_{i=1}^N d_i \right) \mathbf{1}. \tag{4.69}$$

Calculating the variational derivative of $\mathcal{E}_c$ under the assumption $\mathbf{u} < \mathbf{0}$, we obtain for $\mathbf{u} \in \mathcal{U}$ and $\mathbf{g} \in \mathcal{Y}$ satisfying (4.69) with $\mathbf{d} \in H^1(\Omega)^N$

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}\eta}\mathcal{E}_c(\mathbf{u} + \eta\mathbf{g}) &= \lim_{\eta\to 0} \frac{\mathcal{E}_c(\mathbf{u} + \eta\mathbf{g}) - \mathcal{E}_c(\mathbf{u})}{\eta} \\
&= \lim_{\eta\to 0} \frac{1}{\eta} \int_\Omega \Bigg(\frac{\varepsilon^2}{2} \sum_{i=1}^N |\nabla(u_i + \eta g_i)|^2 - \frac{1}{2} \sum_{i=1}^N (u_i + \eta g_i)\, (\boldsymbol{T}(\mathbf{u} + \eta\mathbf{g}))_i \\
&\quad + \frac{1}{2c} \sum_{i=1}^N (u_i + \eta g_i)^2 - \frac{\varepsilon^2}{2} \sum_{i=1}^N |\nabla u_i|^2 + \frac{1}{2} \sum_{i=1}^N u_i\, (\boldsymbol{T}\mathbf{u})_i - \frac{1}{2c} \sum_{i=1}^N u_i^2 \Bigg) \,\mathrm{d}\mathbf{x} \\
&= \int_\Omega \sum_{i=1}^N \left(-(\boldsymbol{T}\mathbf{u})_i + \frac{1}{c} u_i \right) g_i \,\mathrm{d}\mathbf{x} + \varepsilon^2 \int_\Omega \sum_{i=1}^N \nabla u_i \cdot \nabla g_i \,\mathrm{d}\mathbf{x} \\
&= \int_\Omega \sum_{i=1}^N \left(-(\boldsymbol{T}\mathbf{u})_i + \frac{1}{c} u_i - \varepsilon^2 \Delta u_i \right) g_i \,\mathrm{d}\mathbf{x} \qquad (4.70) \\
&\overset{(4.69)}{=} \int_\Omega \sum_{i=1}^N \left(-(\boldsymbol{T}\mathbf{u})_i + \frac{1}{c} u_i - \varepsilon^2 \Delta u_i \right) \left(d_i - \frac{1}{N} \sum_{j=1}^N d_j \right) \mathrm{d}\mathbf{x} \\
&\overset{(4.1)}{=} \int_\Omega \sum_{i=1}^N \left(-(\boldsymbol{T}\mathbf{u})_i + \frac{1}{c} u_i + \frac{1}{N} \sum_{j=1}^N \left((\boldsymbol{T}\mathbf{u})_j - \frac{1}{c} u_j \right) - \varepsilon^2 \Delta u_i \right) d_i \,\mathrm{d}\mathbf{x} \qquad (4.71) \\
&= \int_\Omega \sum_{i=1}^N w_i\, d_i \,\mathrm{d}\mathbf{x}.
\end{aligned}$$

The identity in (4.70) is supplemented with Lemma 2.21 together with the natural zero Neumann boundary condition $\nabla u_i \cdot \mathbf{n} = 0$ on $\partial\Omega$ for $i = 1, \dots, N$. In conclusion, we obtain the following time-discrete vector-valued Cahn–Hilliard equation: Find

$\mathbf{u}_c, \mathbf{w}_c \in H^1(\Omega)^N$ with $\mathbf{u}_c = [u_{c,1}, \ldots, u_{c,N}]^T$ and $\mathbf{w}_c = [w_{c,1}, \ldots, w_{c,N}]^T$ such that

$$\left(u_{c,i} - u_i^{\text{old}}, v\right) + \tau\left((\boldsymbol{L}\nabla \mathbf{w}_c)_i, \nabla v\right) = 0 \quad \forall v \in H^1(\Omega), \tag{4.72}$$

$$\begin{aligned}(w_{c,i}, v) - \varepsilon^2(\nabla u_{c,i}, \nabla v) + ((\boldsymbol{T}\mathbf{u}_c)_i, v) - \frac{1}{c}(\min(0, u_{c,i}), v)& \\ + \frac{1}{N}\sum_{j=1}^{N}\left[\frac{1}{c}(\min(0, u_{c,j}), v) - ((\boldsymbol{T}\mathbf{u}_c)_j, v)\right] = 0 \quad \forall v \in H^1(\Omega),&\end{aligned} \tag{4.73}$$

for $i = 1, \ldots, N$. We supplement (4.72)–(4.73) by the initial condition $\mathbf{u}_0 \in H^1(\Omega)^N$, $\mathbf{u}_0 \in \mathcal{G}^N$ a.e. in Ω. As in Section 4.2, one can show that $\sum_{i=1}^N u_i = 1$ and $\sum_{i=1}^N w_i = 0$ hold true.

We have seen again, how the application of a Moreau–Yosida regularization technique can circumvent the treatment of the variational inequalities in (4.39) as well as the box constraint in (4.40). Indeed, it results in an iterative way for solving the time-discrete Cahn–Hilliard system (4.38)–(4.40): For a sequence $\{c_p\}_{p\in\mathbb{N}}$ with $c_p \to 0$ solve the system (4.72)–(4.73).

Now, we have arrived at a system of time-discrete nonlinear equations for both cases, the smooth potential setting as well as the regularized nonsmooth one. In order to solve the former system, we apply standard Newton methods. Since this is a straightforward step, we will not discuss it here. Regarding the second case, we have to pay attention to the minimum operator present in (4.73). However, as in Chapter 3.5, we can solve the corresponding nonlinear system via the SSN method. This will be the topic of the following chapter.

4.5 Semismooth Newton method

For a specified sequence $c \to 0$, we solve the system (4.72)–(4.73), compactly written as

$$\mathbf{F}_c(\mathbf{u}_c, \mathbf{w}_c) = \left(\mathbf{F}_c^{(1)}(\mathbf{u}_c, \mathbf{w}_c), \mathbf{F}_c^{(2)}(\mathbf{u}_c, \mathbf{w}_c)\right) = 0, \tag{4.74}$$

for every c by an SSN algorithm. In (4.74), the components are defined by

$$\begin{aligned}\left\langle \mathbf{F}_c^{(1)}(\mathbf{u}, \mathbf{w}), \mathbf{v}\right\rangle &= \tau\left(\boldsymbol{L}\nabla \mathbf{w}, \nabla \mathbf{v}\right) + (\mathbf{u}, \mathbf{v}) - \left(\mathbf{u}^{\text{old}}, \mathbf{v}\right), \\ \left\langle \mathbf{F}_c^{(2)}(\mathbf{u}, \mathbf{w}), \mathbf{v}\right\rangle &= \varepsilon^2(\nabla \mathbf{u}, \nabla \mathbf{v}) + \frac{1}{c}(\min(\mathbf{0}, \mathbf{u}), \mathbf{v}) - (\mathbf{w}, \mathbf{v}) - (\boldsymbol{T}\mathbf{u}, \mathbf{v}) \\ &\quad - \frac{1}{N}\sum_{j=1}^{N}\left[\frac{1}{c}(\min(0, u_j)\mathbf{1}, \mathbf{v}) - ((\boldsymbol{T}\mathbf{u})_j\mathbf{1}, \mathbf{v})\right],\end{aligned}$$

for all $\mathbf{u}, \mathbf{w}, \mathbf{v} \in H^1(\Omega)^N$. $\mathbf{F}_c$ is not Fréchet-differentiable due to the presence of the minimum operator. However, the minimum operator satisfies the weaker notion of Newton differentiability; see Definition 2.11. As for the two-component case in Chapter 3.5, we can state a Newton derivative for $\mathbf{F}_c$:

Lemma 4.7. *The mapping $\mathbf{F}_c\colon H^1(\Omega)^N \times H^1(\Omega)^N \to \left(H^1(\Omega)^N\right)^* \times \left(H^1(\Omega)^N\right)^*$ is Newton-differentiable. Furthermore, the operator $\mathbf{G}_c(\mathbf{u},\mathbf{w})$ given by*

$$\langle \mathbf{G}_c(\mathbf{u},\mathbf{w})(\delta\mathbf{u},\delta\mathbf{w}),(\phi,\psi)\rangle = \begin{pmatrix} \tau\left(\boldsymbol{L}\nabla\delta\mathbf{w},\nabla\phi\right) \\ \varepsilon^2(\nabla\delta\mathbf{u},\nabla\psi) + \frac{1}{c}(\boldsymbol{\chi}_{\mathcal{M}(\mathbf{u})}\delta\mathbf{u},\psi) - (\boldsymbol{T}\delta\mathbf{u},\psi) \\ +(\delta\mathbf{u},\phi) \\ -(\delta\mathbf{w},\psi) - \frac{1}{N}\sum_{j=1}^N \left[\frac{1}{c}(\chi_{\mathcal{M}(u_j)}\delta u_j\mathbf{1},\psi) - ((\boldsymbol{T}\delta\mathbf{u})_j\mathbf{1},\psi)\right] \end{pmatrix},$$

serves as a Newton derivative for $\mathbf{F}_c$. Here, $\chi_{\mathcal{M}(u_i)}$ is the characteristic function of the set

$$\mathcal{M}(u_i) := \{\mathbf{x} \in \Omega : u_i(\mathbf{x}) < 0\}.$$

The term $\boldsymbol{\chi}_{\mathcal{M}(\mathbf{u})}\delta\mathbf{u}$ is given as

$$\boldsymbol{\chi}_{\mathcal{M}(\mathbf{u})}\delta\mathbf{u} = \left[\chi_{\mathcal{M}(u_1)}\delta u_1,\ldots,\chi_{\mathcal{M}(u_N)}\delta u_N\right]^T.$$

For the proof, we refer to [91, p. 788] and [92, pp. 885-886].

Lemma 4.8. *Let $\tau \in (0, 4\varepsilon^2\lambda_{\max}(\boldsymbol{T})^{-2}\rho(\boldsymbol{L})^{-1})$. For a given $\mathbf{u} \in H^1(\Omega)^N$ and $(\mathbf{y}_1,\mathbf{y}_2) \in \left(H^1(\Omega)^N\right)^* \times \left(H^1(\Omega)^N\right)^*$, the optimization problem*

$$\begin{aligned} \min_{(\delta\mathbf{u},\delta\mathbf{w})\in\mathcal{K}_1\times\mathcal{V}_0} \quad & \mathcal{J}(\delta\mathbf{u},\delta\mathbf{w}) + \frac{1}{c}(\boldsymbol{\chi}_{\mathcal{M}(\mathbf{u})}\delta\mathbf{u},\delta\mathbf{u}) - \langle\mathbf{y}_2,\delta\mathbf{u}\rangle \\ \text{subject to} \quad & \tau\left(\boldsymbol{L}\nabla\delta\mathbf{w},\nabla\phi\right) + (\delta\mathbf{u},\phi) = \langle\mathbf{y}_1,\phi\rangle \quad \forall\phi\in H^1(\Omega)^N \end{aligned} \tag{$\mathcal{P}_{\mathbf{G}_c}$}$$

admits a unique solution $(\delta\mathbf{u},\delta\mathbf{w})$.

Proof. One proceeds as in the proofs of Theorems 4.4 and 4.5. □

In the next section, we derive the fully discrete problems for both, the smooth system in (4.22)–(4.23) and the regularized nonsmooth system in (4.72)–(4.73).

4.6 Finite element approximation

In this section, we apply FEM [144] to the regularized nonsmooth Cahn–Hilliard system in (4.72)–(4.73). We also want to apply it to the smooth version (4.22)–(4.23). Since both procedures are similar, we only present the methodology based on the nonsmooth setting. Regarding the smooth case, we will state the fully discrete linear system at the end of this section. Moreover, the following presentation complies with the FEM Section 3.6 for the two-component model.

In the following, we assume for simplicity that Ω is a polyhedral domain. Let $\{\mathcal{R}_h\}_{h>0}$ be a triangulation of Ω into disjoint open rectangular elements with maximal element size h, J_h be the set of nodes of $\mathcal{R}_h$, and $p_j \in J_h$ be the coordinates of these nodes. We approximate the infinite-dimensional space $H^1(\Omega)$ by the finite-dimensional space

$$S_h := \{\phi \in C^0(\overline{\Omega}) : \phi\,|_R \in Q_1(R) \ \ \forall R \in \mathcal{R}_h\} \subset H^1(\Omega),$$

of continuous, piecewise multilinear functions. We denote the standard nodal basis functions of S_h by φ_j for all $j \in J_h$. They have the property $\varphi_j(p_i) = \delta_{ij}$, $i, j =$

$1,\dots,m$. The discretized version of the penalized problem (4.72)–(4.73) is the following. Given $\mathbf{u}_h^{\text{old}} = [u_{h,1}^{\text{old}},\dots,u_{h,N}^{\text{old}}]^T \in S_h^N$, find $(\mathbf{u}_{c,h},\mathbf{w}_{c,h}) \in S_h^N \times S_h^N$, where $\mathbf{u}_{c,h} = [u_{c,h,1},\dots,u_{c,h,N}]^T$, $\mathbf{w}_{c,h} = [w_{c,h,1},\dots,w_{c,h,N}]^T$ such that

$$\left\langle F_{c,h}^{(1,i)}(\mathbf{u}_{c,h},\mathbf{w}_{c,h}),v_h\right\rangle = 0 \quad \forall v_h \in S_h, i = 1,\dots,N, \tag{4.75}$$

$$\left\langle F_{c,h}^{(2,i)}(\mathbf{u}_{c,h},\mathbf{w}_{c,h}),v_h\right\rangle = 0 \quad \forall v_h \in S_h i = 1,\dots,N, \tag{4.76}$$

where the components are

$$\begin{aligned}\left\langle F_{c,h}^{(1,i)}(\mathbf{u}_{c,h},\mathbf{w}_{c,h}),v_h\right\rangle &= \tau\left((\boldsymbol{L}\nabla\mathbf{w}_{c,h})_i,\nabla v_h\right) + (u_{c,h,i},v_h)_h - (u_{h,i}^{\text{old}},v_h)_h,\\ \left\langle F_{c,h}^{(2,i)}(\mathbf{u}_{c,h},\mathbf{w}_{c,h}),v_h\right\rangle &= \varepsilon^2(\nabla u_{c,h,i},\nabla v_h) + \frac{1}{c}(\min(0,u_{c,h,i}),v_h)_h - (w_{c,h,i},v_h)_h\\ &\quad - ((\boldsymbol{T}\mathbf{u}_{c,h})_i,v_h)_h - \frac{1}{N}\sum_{j=1}^{N}\left[\frac{1}{c}(\min(0,u_{c,h,j}),v_h)_h - ((\boldsymbol{T}\mathbf{u}_{c,h})_j,v_h)_h\right].\end{aligned}$$

Within our finite element framework, for a given $(\mathbf{u}_h,\mathbf{w}_h) \in S_h^N \times S_h^N$, every step of the SSN method for solving (4.75)–(4.76) requires to compute $(\delta\mathbf{u}_h,\delta\mathbf{w}_h) \in S_h^N \times S_h^N$ satisfying

$$\begin{aligned}\tau\left((\boldsymbol{L}\nabla\delta\mathbf{w}_h)_i,\nabla v_h\right) + \left(\delta u_{h,i},v_h\right)_h &= -F_{c,h}^{(1,i)}(\mathbf{u}_h,\mathbf{w}_h),\\ \varepsilon^2(\nabla\delta u_{h,i},\nabla v_h) + \frac{1}{c}(\chi_{\mathcal{M}(u_{h,i})}^h\delta u_{h,i},v_h)_h - (\delta w_{h,i},v_h)_h - ((\boldsymbol{T}\delta\mathbf{u}_h)_i,v_h)_h&\\ - \frac{1}{N}\sum_{j=1}^{N}\left[\frac{1}{c}(\chi_{\mathcal{M}(u_{h,j})}^h\delta u_{h,j},v_h)_h - ((\boldsymbol{T}\delta u)_{h,j},v_h)_h\right] &= -F_{c,h}^{(2,i)}(\mathbf{u}_h,\mathbf{w}_h),\end{aligned}$$

for all $v_h \in S_h$ and $i = 1,\dots,N$. Here, $\mathbf{u}_h = [u_{h,1},\dots,u_{h,N}]^T$, $\mathbf{w}_h = [w_{h,1},\dots,w_{h,N}]^T$, and $\delta\mathbf{u}_h = [\delta u_{h,1},\dots,\delta u_{h,N}]^T$, $\delta\mathbf{w}_h = [\delta w_{h,1},\dots,\delta w_{h,N}]^T$. Further, we define $\chi_{\mathcal{M}(u_{h,i})}^h := \sum_{j=1}^m \chi_{\mathcal{M}(u_{h,i})}^h(p_j)\,\varphi_j$ with $\chi_{\mathcal{M}(u_{h,i})}^h(p_j) = 0$ if $u_{h,i}(p_j) \geq 0$ and $\chi_{\mathcal{M}(u_{h,i})}^h(p_j) = 1$ otherwise. In the following, we consider $\boldsymbol{T} = \boldsymbol{I} - \mathbf{1}\mathbf{1}^T$, which is a typical example as mentioned in Section 4.1. If we now write a function $v_h \in S_h$ by $v_h = \sum_{j\in J_h} v_{h,j}\,\varphi_j$ and denote the vector of coefficients by $\mathbf{v}$, the fully discrete linear systems (smooth and nonsmooth) per (semismooth) Newton step read in matrix form as

$$\begin{bmatrix} -\boldsymbol{A} & \boldsymbol{I}\otimes\boldsymbol{M}\\ \boldsymbol{I}\otimes\boldsymbol{M} & \tau\boldsymbol{L}\otimes\boldsymbol{K}\end{bmatrix}\begin{bmatrix}\boldsymbol{u}^{(k+1)}\\ \boldsymbol{w}^{(k+1)}\end{bmatrix} = \begin{bmatrix}\boldsymbol{b}\\ (\boldsymbol{I}\otimes\boldsymbol{M})\,\boldsymbol{u}^{\text{old}}\end{bmatrix}. \tag{4.77}$$

Here, $\boldsymbol{u}^{(k+1)} = \left[\left(\boldsymbol{u}_1^{(k+1)}\right)^T,\dots,\left(\boldsymbol{u}_N^{(k+1)}\right)^T\right]^T$, $\boldsymbol{w}^{(k+1)} = \left[\left(\boldsymbol{w}_1^{(k+1)}\right)^T,\dots,\left(\boldsymbol{w}_N^{(k+1)}\right)^T\right]^T \in \mathbb{R}^{Nm}$, and $\boldsymbol{u}^{\text{old}} = \left[\left(\boldsymbol{u}_1^{\text{old}}\right)^T,\dots,\left(\boldsymbol{u}_N^{\text{old}}\right)^T\right]^T \in \mathbb{R}^{Nm}$ is the solution vector from the previous time step. Remember that k denotes the iteration step of the (semismooth) Newton method. The first right-hand side is

$$\boldsymbol{b} = (\boldsymbol{I}\otimes\boldsymbol{M})\left(-2\left(\boldsymbol{u}^{(k)}\right)^3 + \frac{3}{2}\left(\boldsymbol{u}^{(k)}\right)^2\right) + \frac{1}{N}(\boldsymbol{I}\otimes\boldsymbol{M})\left(\sum_{j=1}^{N} 2\left(\boldsymbol{u}_j^{(k)}\right)^3 - \frac{3}{2}\left(\boldsymbol{u}_j^{(k)}\right)^2\right)\mathbf{1}$$

for the use of the smooth potential, where $\boldsymbol{u}^{(k)} = \left[\left(\boldsymbol{u}_1^{(k)}\right)^T,\dots,\left(\boldsymbol{u}_N^{(k)}\right)^T\right]$ and $\mathbf{1} = [1,\dots,1]^T \in \mathbb{R}^{Nm}$. Note that the powers of the form in $\left(\boldsymbol{u}^{(k)}\right)^p$ or $\left(\boldsymbol{u}_j^{(k)}\right)^p$, $p \in \mathbb{N}$, have to be understood elementwise. For the use of the nonsmooth potential, we have

$$\boldsymbol{b} = \boldsymbol{0}.$$

As in the last chapter, $\boldsymbol{M} \in \mathbb{R}^{m\times m}$ is the lumped mass matrix and $\boldsymbol{K} \in \mathbb{R}^{m\times m}$ is the stiffness matrix. Remember that $\boldsymbol{M}$ is a diagonal, symmetric positive definite matrix, and $\boldsymbol{K}$ is symmetric positive semidefinite. Moreover, $\boldsymbol{I} \in \mathbb{R}^{N\times N}$ is the identity matrix. The block $\boldsymbol{A}$ is given as

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_{(1,1)} & \boldsymbol{A}_{(2)} & \cdots & \boldsymbol{A}_{(N-1)} & \boldsymbol{A}_{(N)} \\ \boldsymbol{A}_{(1)} & \boldsymbol{A}_{(2,2)} & \cdots & \boldsymbol{A}_{(N-1)} & \boldsymbol{A}_{(N)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{A}_{(1)} & \boldsymbol{A}_{(2)} & \cdots & \boldsymbol{A}_{(N-1,N-1)} & \boldsymbol{A}_{(N)} \\ \boldsymbol{A}_{(1)} & \boldsymbol{A}_{(2)} & \cdots & \boldsymbol{A}_{(N-1)} & \boldsymbol{A}_{(N,N)} \end{bmatrix},$$

where for $i = 1,\dots,N$

$$\begin{aligned} \boldsymbol{A}_{(i,i)} &= \boldsymbol{A}_{(i,i)}(\boldsymbol{u}_i^{(k)}) = \varepsilon^2\boldsymbol{K} + \left(1 - \frac{1}{N}\right)\boldsymbol{F}_{(i)} \in \mathbb{R}^{m\times m}, \\ \boldsymbol{A}_{(i)} &= \boldsymbol{A}_{(i)}(\boldsymbol{u}_i^{(k)}) = -\frac{1}{N}\boldsymbol{F}_{(i)} \in \mathbb{R}^{m\times m}, \\ \boldsymbol{F}_{(i)} &= \boldsymbol{F}_{(i)}(\boldsymbol{u}_i^{(k)}) = \operatorname{diag}\left([\boldsymbol{M}]_{jj}\left[3\left(u_{h,i,j}^{(k)}\right)^2 - 3u_{h,i,j}^{(k)} + 0.5\right]\right)_{j=1,\dots,m} \in \mathbb{R}^{m\times m}, \end{aligned} \tag{4.78}$$

in the smooth system and

$$\begin{aligned} \boldsymbol{A}_{(i,i)} &= \boldsymbol{A}_{(i,i)}(\boldsymbol{u}_i^{(k)}) = \varepsilon^2\boldsymbol{K} + \left(1 - \frac{1}{N}\right)\left(c^{-1}\boldsymbol{G}_{(i)} - \boldsymbol{M}\right), \\ \boldsymbol{A}_{(i)} &= \boldsymbol{A}_{(i)}(\boldsymbol{u}_i^{(k)}) = -\frac{1}{N}\left(c^{-1}\boldsymbol{G}_{(i)} - \boldsymbol{M}\right), \\ \boldsymbol{G}_{(i)} &= \boldsymbol{G}_{(i)}(\boldsymbol{u}_i^{(k)}) = \operatorname{diag}\left(\begin{array}{cc} [\boldsymbol{M}]_{jj} & \text{if } u_{h,i,j}^{(k)} < 0, \\ 0 & \text{otherwise} \end{array}\right)_{j=1,\dots,m} \in \mathbb{R}^{m\times m}, \end{aligned} \tag{4.79}$$

in the nonsmooth system. Here, $u_{h,i,j}^{(k)}$ denotes the jth element of the vector $\boldsymbol{u}_{h,i}^{(k)}$. Regarding the blocks $\boldsymbol{F}_{(i)}$, $i = 1,\dots,N$, note that the smooth Cahn–Hilliard model yields solutions which satisfy $0 \lesssim u_{h,i,j}^{(k)} \lesssim 1$ for $i = 1,\dots,N$, $j = 1,\dots,m$. Hence,

$$3(u_{h,i,j}^{(k)})^2 - 3u_{h,i,j}^{(k)} + 0.5 \in [-0.25,\, 3\delta^2(\boldsymbol{u}^{(k)}) - 3\delta(\boldsymbol{u}^{(k)}) + 0.5] =: [-0.25,\, \alpha(\boldsymbol{u}^{(k)})] \tag{4.80}$$

for $i = 1,\dots,N$, $j = 1,\dots,m$, where

$$\delta(\boldsymbol{u}^{(k)}) = u_{h,p,q}^{(k)} \quad \text{such that} \quad |u_{h,p,q}^{(k)} - 0.5| = \|\boldsymbol{u}^{(k)} - 0.5\cdot\boldsymbol{1}\|_\infty. \tag{4.81}$$

The lower bound follows from the fact that the function $f\colon \mathbb{R} \to \mathbb{R}$, $z \mapsto 3z^2 - 3z + 0.5$, has a global minimum at $z_0 = 0.5$ with $f(z_0) = -0.25$. For the upper bound, we make use of the symmetry of f around z_0.

Now, we have arrived at the core of our numerical algorithms — the numerical solution of systems of linear equations. Due to the use of FEM, all the matrix blocks $\boldsymbol{M}, \boldsymbol{K}, \boldsymbol{G}_{(\mathbf{i})}, \boldsymbol{F}_{(i)}$, $i = 1, \ldots, N$, are large and sparse. In particular, the higher the number N of phases is the larger is every block of the system matrix in (4.77). In the next section, we design effective practical preconditioners for the two linear systems represented in (4.77).

4.7 Preconditioning

This section is devoted to the development of practical preconditioners for the efficient solution of the two linear systems represented in (4.77). In both cases, smooth and nonsmooth, a linear, nonsymmetric system in saddle point form is at the heart of the computation. Hence, nonsymmetric Krylov subspace solvers have to be used. Due to the complex structure of the block $\boldsymbol{A}$, we rewrite (4.77) and consider

$$\begin{bmatrix} \boldsymbol{I} \otimes \boldsymbol{M} & -\boldsymbol{A} \\ \tau \boldsymbol{L} \otimes \boldsymbol{K} & \boldsymbol{I} \otimes \boldsymbol{M} \end{bmatrix} \begin{bmatrix} \boldsymbol{w}^{(k+1)} \\ \boldsymbol{u}^{(k+1)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b} \\ (\boldsymbol{I} \otimes \boldsymbol{M}) \boldsymbol{u}^{\mathrm{old}} \end{bmatrix}. \tag{4.82}$$

In the following, we denote the coefficient matrix in (4.82) by $\mathcal{A}$. As in the last chapter, we propose the block triangular preconditioner

$$\mathcal{P} = \begin{bmatrix} \boldsymbol{I} \otimes \boldsymbol{M} & \boldsymbol{0} \\ \tau \boldsymbol{L} \otimes \boldsymbol{K} & -\hat{\boldsymbol{S}} \end{bmatrix}, \tag{4.83}$$

where $\hat{\boldsymbol{S}}$ is an approximation of the Schur complement $\boldsymbol{S} = \boldsymbol{I} \otimes \boldsymbol{M} + \tau(\boldsymbol{L} \otimes \boldsymbol{K})(\boldsymbol{I} \otimes \boldsymbol{M})^{-1}\boldsymbol{A}$. Inverting the block $\boldsymbol{I} \otimes \boldsymbol{M}$ is cheap as $\boldsymbol{M}$ is a nonsingular diagonal matrix. The remaining task is to design a Schur complement approximation $\hat{\boldsymbol{S}}$ that is easy to invert and resembles $\boldsymbol{S}$. Finally, the practical block triangular preconditioner is given by

$$\mathcal{P}_0 = \begin{bmatrix} \boldsymbol{I} \otimes \boldsymbol{M} & \boldsymbol{0} \\ \tau \boldsymbol{L} \otimes \boldsymbol{K} & -\boldsymbol{S}_0 \end{bmatrix},$$

where we have to discuss an implementable Schur complement approximation $\boldsymbol{S}_0$ of $\hat{\boldsymbol{S}}$. Overall, the two difficult points are the nondiagonal block matrices $\boldsymbol{A}$ and $\boldsymbol{L} \otimes \boldsymbol{K}$, which couple N equations of size m, respectively. The nonsymmetric block matrix $\boldsymbol{A}$ contains the gradient energy parts, which only arise in the diagonal blocks, as well as the interacting terms coming from the potential. These include in the case of the nonsmooth potential the coupling of all penalization terms. In fact, the latter poses the most challenging part; see Section 4.7.2 for details. As in the last chapter, we begin with the simpler smooth problem and continue with the harder nonsmooth one. Note that the construction of efficient preconditioners in the smooth case is already well established by Boyanova et al. [38]; see also [35, 37, 36, 3] for the two-component model. The authors discussed among others the fully implicit time-discrete scheme. However, they only consider the case $\boldsymbol{L} = \boldsymbol{I}$ in the fully discrete problem. We extend the procedure to the case where $\boldsymbol{L}$ is symmetric positive semidefinite. Moreover, our theoretical proofs differ halfway through. The following presentation generalizes Chapter 3.7 to the vector-valued formulation.

4.7.1 Smooth systems

In the following, we develop a preconditioner for the smooth Cahn–Hilliard system represented in (4.82). This can be achieved by applying the steps in [38], which we explain below in the proof of Theorem 4.10. Note that the proof of Theorem 4.10 differs from the one in [38] at a marked point. Moreover, we generalize the proof from the case $\boldsymbol{L} = \boldsymbol{I}$ to the case of symmetric positive semidefinite matrices $\boldsymbol{L}$. Due to (4.80) together with Proposition 3.10, the diagonal entries of $\boldsymbol{F}_{(i)}$, $i = 1, \ldots, N$, lie in the interval $[-0.25\,Ch^d, Ch^d \alpha(\boldsymbol{u}^{(k)})]$. Hence, as in [38], we suggest to neglect the blocks $\boldsymbol{F}_{(i)}$, $i = 1, \ldots, N$ in $\boldsymbol{A}$. Therefore, we approximate $\mathcal{A}$ as

$$\mathcal{A}_0 = \begin{bmatrix} \boldsymbol{I} \otimes \boldsymbol{M} & -\varepsilon^2 \boldsymbol{I} \otimes \boldsymbol{K} \\ \tau \boldsymbol{L} \otimes \boldsymbol{K} & \boldsymbol{I} \otimes \boldsymbol{M} \end{bmatrix}.$$

In what follows, we discuss the quality of the approximation $\mathcal{A}_0$. We denote the Schur complement of $\mathcal{A}_0$ by $\tilde{\boldsymbol{S}} = \boldsymbol{I} \otimes \boldsymbol{M} + \tau\varepsilon^2 (\boldsymbol{L} \otimes \boldsymbol{K})(\boldsymbol{I} \otimes \boldsymbol{M})^{-1}(\boldsymbol{I} \otimes \boldsymbol{K})$. Note that both the $(1,1)$ and $(2,2)$ block of $\mathcal{A}_0$ are nonsingular. In particular, they are symmetric positive definite.

Theorem 4.9. *$\mathcal{A}_0$ is nonsingular. In particular, $\tilde{\boldsymbol{S}}$ is symmetric positive definite.*

Proof. According to (2.18), $\mathcal{A}_0$ is nonsingular if and only if $\tilde{\boldsymbol{S}}$ is. We rewrite $\tilde{\boldsymbol{S}}$ as

$$\begin{aligned} \tilde{\boldsymbol{S}} &= \boldsymbol{I} \otimes \boldsymbol{M} + \tau\varepsilon^2 (\boldsymbol{L} \otimes \boldsymbol{K})(\boldsymbol{I} \otimes \boldsymbol{M})^{-1}(\boldsymbol{I} \otimes \boldsymbol{K}) \\ &\overset{(2.16)}{=} \boldsymbol{I} \otimes \boldsymbol{M} + \tau\varepsilon^2 (\boldsymbol{L} \otimes \boldsymbol{K})(\boldsymbol{I} \otimes \boldsymbol{M}^{-1})(\boldsymbol{I} \otimes \boldsymbol{K}) \\ &\overset{(2.15)}{=} \boldsymbol{I} \otimes \boldsymbol{M} + \tau\varepsilon^2 (\boldsymbol{L} \otimes \boldsymbol{K}\boldsymbol{M}^{-1}\boldsymbol{K}). \end{aligned}$$

Due to the symmetry of $\boldsymbol{M}, \boldsymbol{K}$, and $\boldsymbol{L}$, as well as the identity (2.14), it follows that $\tilde{\boldsymbol{S}}$ is symmetric. Moreover, Theorem 2.34 yields

$$\sigma(\boldsymbol{L} \otimes \boldsymbol{K}\boldsymbol{M}^{-1}\boldsymbol{K}) = \left\{\lambda_i(\boldsymbol{L})\,\lambda_j(\boldsymbol{K}\boldsymbol{M}^{-1}\boldsymbol{K}),\ i = 1, \ldots, N,\ j = 1, \ldots, m\right\},$$

where $\lambda_i(\boldsymbol{L})$, $i = 1, \ldots, N$, denote the eigenvalues of $\boldsymbol{L}$, and $\lambda_j(\boldsymbol{K}\boldsymbol{M}^{-1}\boldsymbol{K})$, $j = 1, \ldots, m$, denote the eigenvalues of $\boldsymbol{K}\boldsymbol{M}^{-1}\boldsymbol{K}$. Due to Proposition 3.10 as well as the symmetric positive semidefiniteness of $\boldsymbol{L}$, it holds $\sigma(\boldsymbol{L} \otimes \boldsymbol{K}\boldsymbol{M}^{-1}\boldsymbol{K}) \subset \mathbb{R}_{\geq 0}$. Finally, using the Rayleigh quotient argument (Definition 2.27), we obtain

$$\boldsymbol{v}^T \tilde{\boldsymbol{S}} \boldsymbol{v} = \underbrace{\boldsymbol{v}^T (\boldsymbol{I} \otimes \boldsymbol{M}) \boldsymbol{v}}_{>0} + \tau\varepsilon^2 \underbrace{\boldsymbol{v}^T (\boldsymbol{L} \otimes \boldsymbol{K}\boldsymbol{M}^{-1}\boldsymbol{K}) \boldsymbol{v}}_{\geq 0} > 0$$

for all $\boldsymbol{0} \neq \boldsymbol{v} \in \mathbb{R}^{Nm}$. This gives $\sigma(\tilde{\boldsymbol{S}}) \subset \mathbb{R}_{>0}$. □

Consider the generalized eigenvalue problem

$$\mathcal{A} \begin{bmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_2 \end{bmatrix} = \lambda \mathcal{A}_0 \begin{bmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_2 \end{bmatrix}. \tag{4.84}$$

Theorem 4.10. *It holds*

$$\sigma(\mathcal{A}_0^{-1}\mathcal{A}) \subset B_\varsigma(1).$$

The circle radius is bounded by $\varsigma < \frac{\sqrt{\tau\rho(\boldsymbol{L})}}{\varepsilon} \max\left\{0.25, |\alpha(\boldsymbol{u}^{(k)})|\right\}$, where $\alpha(\boldsymbol{u}^{(k)})$ is given in (4.80)–(4.81). In particular, Nm eigenvalues are equal to one. We get $\varsigma < 0.5$ when $\tau \leq 0.25\,\varepsilon^2 \rho(\boldsymbol{L})^{-1} \max^{-2}\left\{0.25, |\alpha(\boldsymbol{u}^{(k)})|\right\}$.

Proof. We transform (4.84) to

$$(\mathcal{A} - \mathcal{A}_0)\begin{bmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_2 \end{bmatrix} = \mu\mathcal{A}_0\begin{bmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_2 \end{bmatrix}, \tag{4.85}$$

where $\mu = \lambda - 1$. The inverse of $\mathcal{A}_0$ can be expressed via a combination of (2.23) and (2.24) as

$$\mathcal{A}_0^{-1} = \begin{bmatrix} \tilde{\boldsymbol{S}}^{-1} & \varepsilon^2\tilde{\boldsymbol{S}}^{-1}(\boldsymbol{I}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M})^{-1} \\ -\tau\tilde{\boldsymbol{S}}^{-1}(\boldsymbol{L}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M})^{-1} & \tilde{\boldsymbol{S}}^{-1} \end{bmatrix}. \tag{4.86}$$

Note that we have used the symmetry of $\tilde{\boldsymbol{S}}$. More precisely, $\tilde{\boldsymbol{S}}$ is equal to the Schur complement of $\mathcal{A}_0$ in its $(2,2)$ block. This yields

$$\mathcal{A}_0^{-1}(\mathcal{A} - \mathcal{A}_0) = \begin{bmatrix} \boldsymbol{0} & \tilde{\boldsymbol{S}}^{-1}\left[-\boldsymbol{A} + \varepsilon^2(\boldsymbol{I}\otimes\boldsymbol{K})\right] \\ \boldsymbol{0} & -\tau\tilde{\boldsymbol{S}}^{-1}(\boldsymbol{L}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}\left[-\boldsymbol{A} + \varepsilon^2(\boldsymbol{I}\otimes\boldsymbol{K})\right] \end{bmatrix}. \tag{4.87}$$

Hence, (4.85) has Nm zero eigenvalues corresponding to eigenvectors $[\boldsymbol{q}_1^T, \boldsymbol{0}^T]^T$. Thus, (4.84) has Nm one eigenvalues. Next, we write (4.85) out

$$(-\boldsymbol{A} + \varepsilon^2(\boldsymbol{I}\otimes\boldsymbol{K}))\boldsymbol{q}_2 = \mu((\boldsymbol{I}\otimes\boldsymbol{M})\boldsymbol{q}_1 - \varepsilon^2(\boldsymbol{I}\otimes\boldsymbol{K})\boldsymbol{q}_2), \tag{4.88}$$

$$\boldsymbol{0} = \mu(\tau(\boldsymbol{L}\otimes\boldsymbol{K})\boldsymbol{q}_1 + (\boldsymbol{I}\otimes\boldsymbol{M})\boldsymbol{q}_2). \tag{4.89}$$

We express $\mu\boldsymbol{q}_1$ from (4.88) and substitute it into (4.89)

$$\begin{aligned}\tau(\boldsymbol{L}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}\left[\boldsymbol{A} - \varepsilon^2(\boldsymbol{I}\otimes\boldsymbol{K})\right]\boldsymbol{q}_2 \\ = \mu\left[\tau\varepsilon^2(\boldsymbol{L}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}(\boldsymbol{I}\otimes\boldsymbol{K}) + (\boldsymbol{I}\otimes\boldsymbol{M})\right]\boldsymbol{q}_2.\end{aligned} \tag{4.90}$$

Multiplying (4.90) from the left by $(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}$ yields the following generalized eigenvalue problem

$$\begin{aligned}\tau(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}(\boldsymbol{L}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}\left[\boldsymbol{A} - \varepsilon^2(\boldsymbol{I}\otimes\boldsymbol{K})\right]\boldsymbol{q}_2 \\ = \mu\left[\boldsymbol{I} + \tau\varepsilon^2(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}(\boldsymbol{L}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}(\boldsymbol{I}\otimes\boldsymbol{K})\right]\boldsymbol{q}_2.\end{aligned} \tag{4.91}$$

Note that the first identity matrix on the right-hand of (4.91) is correctly written as $(\boldsymbol{I}_N \times \boldsymbol{I}_m)$, where $\boldsymbol{I}_N \in \mathbb{R}^{N\times N}$ and $\boldsymbol{I}_m \in \mathbb{R}^{m\times m}$. Under this notation, the remainder identity matrices in (4.91) are correctly written as $\boldsymbol{I}_N$. In order to ease the notation, we write $\boldsymbol{I}$ for all identity matrices. We introduce

$$\begin{aligned}\boldsymbol{R} := \tau\left[\boldsymbol{I} + \tau\varepsilon^2(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}(\boldsymbol{L}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}(\boldsymbol{I}\otimes\boldsymbol{K})\right]^{-1} \\ (\boldsymbol{I}\otimes\boldsymbol{M})^{-1}(\boldsymbol{L}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}\left[\boldsymbol{A} - \varepsilon^2(\boldsymbol{I}\otimes\boldsymbol{K})\right]\end{aligned}$$

and estimate its eigenvalues in the following. Therefore, we first perform a similarity transformation (Definition 2.23) on $\boldsymbol{R}$ in the form of $(\boldsymbol{I}\otimes\boldsymbol{M}^{\frac{1}{2}})\boldsymbol{R}(\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}}) =: \tilde{\boldsymbol{R}}$. Note that $\boldsymbol{R}$ and $\tilde{\boldsymbol{R}}$ have the same eigenvalues. Next, we analyze the eigenvalues of

$\tilde{\boldsymbol{R}}$. Therefore, we first reformulate $\tilde{\boldsymbol{R}}$ as

$$
\begin{aligned}
\tilde{\boldsymbol{R}} &= \tau(\boldsymbol{I}\otimes\boldsymbol{M}^{\frac{1}{2}})\left[\boldsymbol{I}+\tau\varepsilon^2(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}(\boldsymbol{L}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}(\boldsymbol{I}\otimes\boldsymbol{K})\right]^{-1}\\
&\quad(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}(\boldsymbol{L}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}\left[\boldsymbol{A}-\varepsilon^2(\boldsymbol{I}\otimes\boldsymbol{K})\right](\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})\\
&= \tau\left[\left(\boldsymbol{I}+\tau\varepsilon^2(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}(\boldsymbol{L}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}(\boldsymbol{I}\otimes\boldsymbol{K})\right)(\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})\right]^{-1}\\
&\quad(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}(\boldsymbol{L}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}\left[\boldsymbol{A}-\varepsilon^2(\boldsymbol{I}\otimes\boldsymbol{K})\right](\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})\\
&= \tau\left[(\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})+\tau\varepsilon^2(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}(\boldsymbol{L}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}(\boldsymbol{I}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})\right]^{-1}\\
&\quad(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}(\boldsymbol{L}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}\left[\boldsymbol{A}-\varepsilon^2(\boldsymbol{I}\otimes\boldsymbol{K})\right](\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})\\
&= \tau\left[(\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})\left(\boldsymbol{I}+\tau\varepsilon^2(\boldsymbol{I}\otimes\boldsymbol{M})^{-\frac{1}{2}}(\boldsymbol{L}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}(\boldsymbol{I}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})\right)\right]^{-1}\\
&\quad(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}(\boldsymbol{L}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}\left[\boldsymbol{A}-\varepsilon^2(\boldsymbol{I}\otimes\boldsymbol{K})\right](\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})\\
&= \tau\left[\boldsymbol{I}+\tau\varepsilon^2(\boldsymbol{I}\otimes\boldsymbol{M})^{-\frac{1}{2}}(\boldsymbol{L}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}(\boldsymbol{I}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})\right]^{-1}\\
&\quad(\boldsymbol{I}\otimes\boldsymbol{M})^{-\frac{1}{2}}(\boldsymbol{L}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}\left[\boldsymbol{A}-\varepsilon^2(\boldsymbol{I}\otimes\boldsymbol{K})\right](\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})\\
&= \tau\left(\boldsymbol{I}+\tau\varepsilon^2\tilde{\boldsymbol{L}}\tilde{\boldsymbol{K}}\right)^{-1}\tilde{\boldsymbol{L}}\tilde{\boldsymbol{A}},
\end{aligned}
\tag{4.92}
$$

where $\tilde{\boldsymbol{K}} = (\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})(\boldsymbol{I}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})$, $\tilde{\boldsymbol{L}} = (\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})(\boldsymbol{L}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})$, and $\tilde{\boldsymbol{A}} = (\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})\left[\boldsymbol{A}-\varepsilon^2(\boldsymbol{I}\otimes\boldsymbol{K})\right](\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})$. From now on, this proof differs from the one in [38]. It can be easily seen that $\tilde{\boldsymbol{K}}$ is symmetric. If we rewrite $\tilde{\boldsymbol{K}}$ as

$$\tilde{\boldsymbol{K}} = \boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}}\boldsymbol{K}\boldsymbol{M}^{-\frac{1}{2}},$$

Theorem 2.34 yields

$$\sigma(\tilde{\boldsymbol{K}}) = \sigma(\boldsymbol{M}^{-\frac{1}{2}}\boldsymbol{K}\boldsymbol{M}^{-\frac{1}{2}}) \subset \mathbb{R}_{\geq 0}.$$

Hence, $\tilde{\boldsymbol{K}}$ is symmetric positive semidefinite. Due to Theorem 2.27 (symmetric Schur decomposition), we can write $\boldsymbol{M}^{-\frac{1}{2}}\boldsymbol{K}\boldsymbol{M}^{-\frac{1}{2}} = \boldsymbol{Q}_K\boldsymbol{\Lambda}_K\boldsymbol{Q}_K^T$, where $\boldsymbol{Q}_K = [\boldsymbol{q}_{K,1}|\ldots|\boldsymbol{q}_{K,m}] \in \mathbb{R}^{m\times m}$ is orthogonal and $\boldsymbol{\Lambda}_K = \operatorname{diag}(\lambda_{K,1},\ldots,\lambda_{K,m})$ such that $\boldsymbol{M}^{-\frac{1}{2}}\boldsymbol{K}\boldsymbol{M}^{-\frac{1}{2}}\boldsymbol{q}_{K,j} = \lambda_{K,j}\,\boldsymbol{q}_{K,j}$ for $j = 1,\ldots,m$. Similarly, it can be easily seen that $\tilde{\boldsymbol{L}}$ is symmetric since $\boldsymbol{L}$ is symmetric. If we rewrite $\tilde{\boldsymbol{L}}$ as

$$\tilde{\boldsymbol{L}} = \boldsymbol{L}\otimes\boldsymbol{M}^{-\frac{1}{2}}\boldsymbol{K}\boldsymbol{M}^{-\frac{1}{2}},$$

Theorem 2.34 yields

$$\sigma(\boldsymbol{L}\otimes\boldsymbol{M}^{-\frac{1}{2}}\boldsymbol{K}\boldsymbol{M}^{-\frac{1}{2}}) = \left\{\lambda_{L,i}\,\lambda_{K,j},\ i = 1,\ldots,N,\ j = 1,\ldots,m\right\},$$

where $\lambda_{L,i}$, $i = 1,\ldots,N$, denote the eigenvalues of $\boldsymbol{L}$. Due to Proposition 3.10 as well as the symmetric positive semidefiniteness of $\boldsymbol{L}$, it holds $\sigma(\tilde{\boldsymbol{L}}) \subset \mathbb{R}_{\geq 0}$. Due to Theorem 2.27 (symmetric Schur decomposition), we can write $\boldsymbol{L} = \boldsymbol{Q}_L\boldsymbol{\Lambda}_L\boldsymbol{Q}_L^T$, where $\boldsymbol{Q}_L = [\boldsymbol{q}_{L,1}|\ldots|\boldsymbol{q}_{L,N}] \in \mathbb{R}^{N\times N}$ is orthogonal and $\boldsymbol{\Lambda}_L = \operatorname{diag}(\lambda_{L,1},\ldots,\lambda_{L,N})$ such that $\boldsymbol{L}\boldsymbol{q}_{L,j} = \lambda_{L,j}\,\boldsymbol{q}_{L,j}$ for $j = 1,\ldots,N$. Using the two introduced Schur decompositions, we can rewrite $\tilde{\boldsymbol{K}}$ and $\tilde{\boldsymbol{L}}$ as

$$
\begin{aligned}
\tilde{\boldsymbol{K}} &= \boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}}\boldsymbol{K}\boldsymbol{M}^{-\frac{1}{2}} = \boldsymbol{Q}_L\boldsymbol{Q}_L^T\otimes\boldsymbol{Q}_K\boldsymbol{\Lambda}_K\boldsymbol{Q}_K^T \overset{(2.15)}{=} (\boldsymbol{Q}_L\otimes\boldsymbol{Q}_K)(\boldsymbol{I}\otimes\boldsymbol{\Lambda}_K)(\boldsymbol{Q}_L^T\otimes\boldsymbol{Q}_K^T),\\
\tilde{\boldsymbol{L}} &= \boldsymbol{L}\otimes\boldsymbol{M}^{-\frac{1}{2}}\boldsymbol{K}\boldsymbol{M}^{-\frac{1}{2}} = \boldsymbol{Q}_L\boldsymbol{\Lambda}_L\boldsymbol{Q}_L^T\otimes\boldsymbol{Q}_K\boldsymbol{\Lambda}_K\boldsymbol{Q}_K^T \overset{(2.15)}{=} (\boldsymbol{Q}_L\otimes\boldsymbol{Q}_K)(\boldsymbol{\Lambda}_L\otimes\boldsymbol{\Lambda}_K)(\boldsymbol{Q}_L^T\otimes\boldsymbol{Q}_K^T).
\end{aligned}
$$

Hence,

$$\begin{aligned}\tilde{\boldsymbol{L}}\tilde{\boldsymbol{K}} &= (\boldsymbol{Q}_L \otimes \boldsymbol{Q}_K)(\boldsymbol{\Lambda}_L \otimes \boldsymbol{\Lambda}_K)(\boldsymbol{Q}_L^T \otimes \boldsymbol{Q}_K^T)(\boldsymbol{Q}_L \otimes \boldsymbol{Q}_K)(\boldsymbol{I} \otimes \boldsymbol{\Lambda}_K)(\boldsymbol{Q}_L^T \otimes \boldsymbol{Q}_K^T) \\ &\overset{(2.15)}{=} (\boldsymbol{Q}_L \otimes \boldsymbol{Q}_K)(\boldsymbol{\Lambda}_L \otimes \boldsymbol{\Lambda}_K^2)(\boldsymbol{Q}_L^T \otimes \boldsymbol{Q}_K^T), \qquad (4.93)\end{aligned}$$

and we can rewrite $\left(\boldsymbol{I} + \tau\varepsilon^2\tilde{\boldsymbol{L}}\tilde{\boldsymbol{K}}\right)^{-1}\tilde{\boldsymbol{L}}$ in (4.92) further as

$$\begin{aligned}\left(\boldsymbol{I} + \tau\varepsilon^2\tilde{\boldsymbol{L}}\tilde{\boldsymbol{K}}\right)^{-1}\tilde{\boldsymbol{L}} &= \left[(\boldsymbol{Q}_L \otimes \boldsymbol{Q}_K)(\boldsymbol{Q}_L^T \otimes \boldsymbol{Q}_K^T) + \tau\varepsilon^2(\boldsymbol{Q}_L \otimes \boldsymbol{Q}_K)(\boldsymbol{\Lambda}_L \otimes \boldsymbol{\Lambda}_K^2)(\boldsymbol{Q}_L^T \otimes \boldsymbol{Q}_K^T)\right]^{-1} \\ &\quad (\boldsymbol{Q}_L \otimes \boldsymbol{Q}_K)(\boldsymbol{\Lambda}_L \otimes \boldsymbol{\Lambda}_K)(\boldsymbol{Q}_L^T \otimes \boldsymbol{Q}_K^T) \\ &= \left[(\boldsymbol{Q}_L \otimes \boldsymbol{Q}_K)\left(\boldsymbol{I} + \tau\varepsilon^2(\boldsymbol{\Lambda}_L \otimes \boldsymbol{\Lambda}_K^2)\right)(\boldsymbol{Q}_L^T \otimes \boldsymbol{Q}_K^T)\right]^{-1} \\ &\quad (\boldsymbol{Q}_L \otimes \boldsymbol{Q}_K)(\boldsymbol{\Lambda}_L \otimes \boldsymbol{\Lambda}_K)(\boldsymbol{Q}_L^T \otimes \boldsymbol{Q}_K^T) \\ &= (\boldsymbol{Q}_L \otimes \boldsymbol{Q}_K)\left[\boldsymbol{I} + \tau\varepsilon^2(\boldsymbol{\Lambda}_L \otimes \boldsymbol{\Lambda}_K^2)\right]^{-1}(\boldsymbol{\Lambda}_L \otimes \boldsymbol{\Lambda}_K)(\boldsymbol{Q}_L^T \otimes \boldsymbol{Q}_K^T), \qquad (4.94)\end{aligned}$$

where $\left[\boldsymbol{I} + \tau\varepsilon^2(\boldsymbol{\Lambda}_L \otimes \boldsymbol{\Lambda}_K^2)\right]^{-1}(\boldsymbol{\Lambda}_L \otimes \boldsymbol{\Lambda}_K)$ is a diagonal matrix. Hence, $\left(\boldsymbol{I} + \tau\varepsilon^2\tilde{\boldsymbol{L}}\tilde{\boldsymbol{K}}\right)^{-1}\tilde{\boldsymbol{L}}$ is symmetric. It follows that

$$\left(\boldsymbol{I} + \tau\varepsilon^2\tilde{\boldsymbol{L}}\tilde{\boldsymbol{K}}\right)^{-1}\tilde{\boldsymbol{L}}(\boldsymbol{q}_{L,i} \otimes \boldsymbol{q}_{K,j}) = \frac{\lambda_{L,i}\lambda_{K,j}}{1 + \tau\varepsilon^2\lambda_{L,i}\lambda_{K,j}^2}(\boldsymbol{q}_{L,i} \otimes \boldsymbol{q}_{K,j}) \qquad (4.95)$$

for $i = 1, \ldots, N$, $j = 1, \ldots, m$. Using the inequality

$$0 \leq (1 - ab)^2 = 1 + a^2b^2 - 2ab$$

with $a, b \in \mathbb{R}$, we can bound the eigenvalues of (4.95) as

$$\frac{\lambda_{L,i}\lambda_{K,j}}{1 + \tau\varepsilon^2\lambda_{L,i}\lambda_{K,j}^2} \leq \frac{\lambda_{L,i}\lambda_{K,j}}{2\varepsilon\sqrt{\tau}\lambda_{K,j}\sqrt{\lambda_{L,i}}} = \frac{\sqrt{\lambda_{L,i}}}{2\varepsilon\sqrt{\tau}}$$

for $j = 1, \ldots, m$. Here, we have used $a^2 = \tau\varepsilon^2\lambda_{K,j}$ and $b^2 = \lambda_{L,i}\lambda_{K,j}$. This yields

$$\rho\left(\left(\boldsymbol{I} + \tau\varepsilon^2\tilde{\boldsymbol{L}}\tilde{\boldsymbol{K}}\right)^{-1}\tilde{\boldsymbol{L}}\right) \leq \frac{\sqrt{\rho(\boldsymbol{L})}}{2\varepsilon\sqrt{\tau}}. \qquad (4.96)$$

Finally, we can estimate the eigenvalues of $\tilde{\boldsymbol{R}}$. Note that due to Theorem 2.31, it holds $\rho(\tilde{\boldsymbol{R}}) \leq \|\tilde{\boldsymbol{R}}\|$. Further, we obtain

$$\|\tilde{\boldsymbol{R}}\| \leq \tau\|\left(\boldsymbol{I} + \tau\varepsilon^2\tilde{\boldsymbol{L}}\tilde{\boldsymbol{K}}\right)^{-1}\tilde{\boldsymbol{L}}\|\,\|\tilde{\boldsymbol{A}}\| = \tau\,\rho\left(\left(\boldsymbol{I} + \tau\varepsilon^2\tilde{\boldsymbol{L}}\tilde{\boldsymbol{K}}\right)^{-1}\tilde{\boldsymbol{L}}\right)\|\tilde{\boldsymbol{A}}\| \leq \frac{\tau\sqrt{\rho(\boldsymbol{L})}}{2\varepsilon\sqrt{\tau}}\|\tilde{\boldsymbol{A}}\|, \qquad (4.97)$$

where the equality holds due to the symmetry of $\left(\boldsymbol{I} + \tau\varepsilon^2\tilde{\boldsymbol{L}}\tilde{\boldsymbol{K}}\right)^{-1}\tilde{\boldsymbol{L}}$. Moreover, due to the diagonal structure of each block in $\boldsymbol{A} - \varepsilon^2(\boldsymbol{I} \otimes \boldsymbol{K})$ we have

$$\begin{aligned}\tilde{\boldsymbol{A}} &= (\boldsymbol{I} \otimes \boldsymbol{M}^{-\frac{1}{2}})\left[\boldsymbol{A} - \varepsilon^2(\boldsymbol{I} \otimes \boldsymbol{K})\right](\boldsymbol{I} \otimes \boldsymbol{M}^{-\frac{1}{2}}) \\ &= \frac{1}{N}\begin{bmatrix} (N-1)\tilde{\boldsymbol{F}}_{(1)} & -\tilde{\boldsymbol{F}}_{(2)} & \cdots & -\tilde{\boldsymbol{F}}_{(N-1)} & -\tilde{\boldsymbol{F}}_{(N)} \\ -\tilde{\boldsymbol{F}}_{(1)} & (N-1)\tilde{\boldsymbol{F}}_{(2)} & \cdots & -\tilde{\boldsymbol{F}}_{(N-1)} & -\tilde{\boldsymbol{F}}_{(N)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -\tilde{\boldsymbol{F}}_{(1)} & -\tilde{\boldsymbol{F}}_{(2)} & \cdots & (N-1)\tilde{\boldsymbol{F}}_{(N-1)} & -\tilde{\boldsymbol{F}}_{(N)} \\ -\tilde{\boldsymbol{F}}_{(1)} & -\tilde{\boldsymbol{F}}_{(2)} & \cdots & -\tilde{\boldsymbol{F}}_{(N-1)} & (N-1)\tilde{\boldsymbol{F}}_{(N)} \end{bmatrix},\end{aligned}$$

where

$$\tilde{\boldsymbol{F}}_{(i)} = \tilde{\boldsymbol{F}}_{(i)}(\boldsymbol{u}_i^{(k)}) = \operatorname{diag}\left(3(u_{h,i,j}^{(k)})^2 - 3u_{h,i,j}^{(k)} + 0.5\right)_{j=1,\dots,m.}$$

Due to (4.80)–(4.81), the diagonal entries of $\tilde{\boldsymbol{F}}_{(i)}$, $i = 1, \dots, N$, lie in the interval $[-0.25,\, \alpha(\boldsymbol{u}^{(k)})]$. Since each block in $\tilde{\boldsymbol{A}}$ is diagonal, the number of nonzero entries per row or column is N. Moreover,

$$\|\tilde{\boldsymbol{A}}\|_1 \leq 2\frac{N-1}{N} \max\left\{0.25,\, |\alpha(\boldsymbol{u}^{(k)})|\right\} \quad \text{and} \quad \|\tilde{\boldsymbol{A}}\|_\infty \leq 2\frac{N-1}{N} \max\left\{0.25,\, |\alpha(\boldsymbol{u}^{(k)})|\right\}.$$

Thus, (2.12) yields

$$\|\tilde{\boldsymbol{A}}\| \leq \sqrt{\|\tilde{\boldsymbol{A}}\|_1 \|\tilde{\boldsymbol{A}}\|_\infty} \leq 2\frac{N-1}{N} \max\left\{0.25,\, |\alpha(\boldsymbol{u}^{(k)})|\right\}.$$

Hence, we obtain in (4.97)

$$\|\tilde{\boldsymbol{R}}\| \leq \frac{N-1}{N} \frac{\sqrt{\tau \rho(\boldsymbol{L})}}{\varepsilon} \max\left\{0.25,\, |\alpha(\boldsymbol{u}^{(k)})|\right\} < \frac{\sqrt{\tau \rho(\boldsymbol{L})}}{\varepsilon} \max\left\{0.25,\, |\alpha(\boldsymbol{u}^{(k)})|\right\}. \tag{4.98}$$

Therefore, for $\tau \leq 0.25\, \varepsilon^2 \rho(\boldsymbol{L})^{-1} \max^{-2}\left\{0.25,\, |\alpha(\boldsymbol{u}^{(k)})|\right\}$, it holds $\sigma(\tilde{\boldsymbol{R}}) = \sigma(\boldsymbol{R}) \subset B_{0.5}(0)$ and hence $\sigma(\boldsymbol{\mathcal{A}}_0^{-1}\boldsymbol{\mathcal{A}}) \subset B_{0.5}(1)$. □

Remark 4.5. *Note that the time step condition in Theorem 4.10 complies with the one in Theorem 4.1 and 4.2.*

After we have proven that $\boldsymbol{\mathcal{A}}_0$ is a reasonable approximation of $\boldsymbol{\mathcal{A}}$, we can go over to the construction of a suitable preconditioner for $\boldsymbol{\mathcal{A}}_0$ and hence for $\boldsymbol{\mathcal{A}}$. We propose the block triangular preconditioner $\boldsymbol{\mathcal{P}}$ in (4.83), where we design $\hat{\boldsymbol{S}}$ to be an approximation of $\tilde{\boldsymbol{S}} = \boldsymbol{I} \otimes \boldsymbol{M} + \tau\varepsilon^2(\boldsymbol{L} \otimes \boldsymbol{K})(\boldsymbol{I} \otimes \boldsymbol{M})^{-1}(\boldsymbol{I} \otimes \boldsymbol{K})$, which is the Schur complement of $\boldsymbol{\mathcal{A}}_0$. More precisely, we use

$$\begin{aligned}
\hat{\boldsymbol{S}} &= \boldsymbol{S}_1(\boldsymbol{I} \otimes \boldsymbol{M})^{-1}\boldsymbol{S}_2 && (4.99)\\
&= \left(\boldsymbol{I} \otimes \boldsymbol{M} + \varepsilon\sqrt{\tau}(\boldsymbol{L} \otimes \boldsymbol{K})\right)(\boldsymbol{I} \otimes \boldsymbol{M})^{-1}\left(\boldsymbol{I} \otimes \boldsymbol{M} + \varepsilon\sqrt{\tau}(\boldsymbol{I} \otimes \boldsymbol{K})\right) && (4.100)\\
&= \boldsymbol{I} \otimes \boldsymbol{M} + \tau\varepsilon^2(\boldsymbol{L} \otimes \boldsymbol{K})(\boldsymbol{I} \otimes \boldsymbol{M})^{-1}(\boldsymbol{I} \otimes \boldsymbol{K}) + \varepsilon\sqrt{\tau}(\boldsymbol{I} \otimes \boldsymbol{K}) + \varepsilon\sqrt{\tau}(\boldsymbol{L} \otimes \boldsymbol{K}). && (4.101)
\end{aligned}$$

The first two terms in (4.101) match the exact Schur complement $\tilde{\boldsymbol{S}}$ of $\boldsymbol{\mathcal{A}}_0$. Due to the balanced distribution of $\varepsilon^2\tau$ in form of $\varepsilon\sqrt{\tau}$ in both factors $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$, the influence of both remainder terms in (4.101) is reduced.

Lemma 4.11. *It holds*

$$\sigma\left(\hat{\boldsymbol{S}}^{-1}\tilde{\boldsymbol{S}}\right) \subset (0, 1].$$

In particular,

$$\sigma\left(\hat{\boldsymbol{S}}^{-1}\tilde{\boldsymbol{S}}\right) \subset [\beta(\varepsilon, \tau, \boldsymbol{M}, \boldsymbol{K}, \boldsymbol{L}),\, 1]\,,$$

where

$$\beta(\varepsilon, \tau, \boldsymbol{M}, \boldsymbol{K}, \boldsymbol{L}) = \min\left\{\frac{1}{1 + \varepsilon\sqrt{\tau}\rho(\boldsymbol{M}^{-\frac{1}{2}}\boldsymbol{K}\boldsymbol{M}^{-\frac{1}{2}})}, \frac{1}{1 + \frac{1+l_0}{2\sqrt{l_0}}}, \frac{1}{1 + \frac{1+\rho(\boldsymbol{L})}{2\sqrt{\rho(\boldsymbol{L})}}}\right\}$$

and l_0 denotes the smallest positive eigenvalue of $\boldsymbol{L}$.

Proof. As in the proof of Theorem 4.9, we can show that $\hat{\boldsymbol{S}}$ is symmetric positive definite. Hence, Lemma 2.30 implies $\sigma(\hat{\boldsymbol{S}}^{-1}\tilde{\boldsymbol{S}}) \subset \mathbb{R}_{>0}$. In order to obtain sharper bounds, we proceed as in the proof of Theorem 4.10. First, we rewrite $\hat{\boldsymbol{S}}^{-1}\tilde{\boldsymbol{S}}$ as

$$\begin{aligned}
\hat{\boldsymbol{S}}^{-1}\tilde{\boldsymbol{S}} &= \left[\boldsymbol{I}\otimes\boldsymbol{M} + \tau\varepsilon^2(\boldsymbol{L}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}(\boldsymbol{I}\otimes\boldsymbol{K}) + \varepsilon\sqrt{\tau}(\boldsymbol{I}\otimes\boldsymbol{K}) + \varepsilon\sqrt{\tau}(\boldsymbol{L}\otimes\boldsymbol{K})\right]^{-1} \\
&\quad \left[\boldsymbol{I}\otimes\boldsymbol{M} + \tau\varepsilon^2(\boldsymbol{L}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}(\boldsymbol{I}\otimes\boldsymbol{K})\right] \\
&= \Big[(\boldsymbol{I}\otimes\boldsymbol{M}^{\frac{1}{2}})\Big(\boldsymbol{I} + \tau\varepsilon^2(\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})(\boldsymbol{L}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}(\boldsymbol{I}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}}) \\
&\quad + \varepsilon\sqrt{\tau}(\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})(\boldsymbol{I}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}}) \\
&\quad + \varepsilon\sqrt{\tau}(\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})(\boldsymbol{L}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})\Big)(\boldsymbol{I}\otimes\boldsymbol{M}^{\frac{1}{2}})\Big]^{-1}(\boldsymbol{I}\otimes\boldsymbol{M}^{\frac{1}{2}}) \\
&\quad \left[\boldsymbol{I} + \tau\varepsilon^2(\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})(\boldsymbol{L}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}(\boldsymbol{I}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})\right](\boldsymbol{I}\otimes\boldsymbol{M}^{\frac{1}{2}}) \\
&= (\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})\Big[\boldsymbol{I} + \tau\varepsilon^2(\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})(\boldsymbol{L}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}(\boldsymbol{I}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}}) \\
&\quad + \varepsilon\sqrt{\tau}(\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})(\boldsymbol{I}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}}) + \varepsilon\sqrt{\tau}(\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})(\boldsymbol{L}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})\Big]^{-1} \\
&\quad \left[\boldsymbol{I} + \tau\varepsilon^2(\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})(\boldsymbol{L}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}(\boldsymbol{I}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})\right](\boldsymbol{I}\otimes\boldsymbol{M}^{\frac{1}{2}}) \\
&= (\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})\left[\boldsymbol{I} + \varepsilon^2\tau\tilde{\boldsymbol{L}}\tilde{\boldsymbol{K}} + \varepsilon\sqrt{\tau}\tilde{\boldsymbol{K}} + \varepsilon\sqrt{\tau}\tilde{\boldsymbol{L}}\right]^{-1}\left[\boldsymbol{I} + \varepsilon^2\tau\tilde{\boldsymbol{L}}\tilde{\boldsymbol{K}}\right](\boldsymbol{I}\otimes\boldsymbol{M}^{\frac{1}{2}}).
\end{aligned}$$

Hence, $\hat{\boldsymbol{S}}^{-1}\tilde{\boldsymbol{S}}$ and $\boldsymbol{R} := \left[\boldsymbol{I} + \varepsilon^2\tau\tilde{\boldsymbol{L}}\tilde{\boldsymbol{K}} + \varepsilon\sqrt{\tau}\tilde{\boldsymbol{K}} + \varepsilon\sqrt{\tau}\tilde{\boldsymbol{L}}\right]^{-1}\left[\boldsymbol{I} + \varepsilon^2\tau\tilde{\boldsymbol{L}}\tilde{\boldsymbol{K}}\right]$ are similar and have the same eigenvalues. Here, $\tilde{\boldsymbol{K}} = (\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})(\boldsymbol{I}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})$ and $\tilde{\boldsymbol{L}} = (\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})(\boldsymbol{L}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})$ are the same matrices as in the proof of Theorem 4.10. Again, we make use of the symmetric Schur decompositions of $\boldsymbol{M}^{-\frac{1}{2}}\boldsymbol{K}\boldsymbol{M}^{-\frac{1}{2}}$ and $\boldsymbol{L}$ in the form of $\boldsymbol{M}^{-\frac{1}{2}}\boldsymbol{K}\boldsymbol{M}^{-\frac{1}{2}} = \boldsymbol{Q}_K\boldsymbol{\Lambda}_K\boldsymbol{Q}_K^T$, where $\boldsymbol{Q}_K = [\boldsymbol{q}_{K,1}|\ldots|\boldsymbol{q}_{K,m}] \in \mathbb{R}^{m\times m}$ is orthogonal and $\boldsymbol{\Lambda}_K = \operatorname{diag}(\lambda_{K,1},\ldots,\lambda_{K,m})$ such that $\boldsymbol{M}^{-\frac{1}{2}}\boldsymbol{K}\boldsymbol{M}^{-\frac{1}{2}}\boldsymbol{q}_{K,j} = \lambda_{K,j}\,\boldsymbol{q}_{K,j}$ for $j = 1,\ldots,m$, and $\boldsymbol{L} = \boldsymbol{Q}_L\boldsymbol{\Lambda}_L\boldsymbol{Q}_L^T$, where $\boldsymbol{Q}_L = [\boldsymbol{q}_{L,1}|\ldots|\boldsymbol{q}_{L,N}] \in \mathbb{R}^{N\times N}$ is orthogonal and $\boldsymbol{\Lambda}_L = \operatorname{diag}(\lambda_{L,1},\ldots,\lambda_{L,N})$ such that $\boldsymbol{L}\boldsymbol{q}_{L,j} = \lambda_{L,j}\,\boldsymbol{q}_{L,j}$ for $j = 1,\ldots,N$. Hence,

$$\begin{aligned}
\boldsymbol{R} &= \Big[(\boldsymbol{Q}_L\otimes\boldsymbol{Q}_K)(\boldsymbol{Q}_L^T\otimes\boldsymbol{Q}_K^T) + \varepsilon^2\tau(\boldsymbol{Q}_L\otimes\boldsymbol{Q}_K)(\boldsymbol{\Lambda}_L\otimes\boldsymbol{\Lambda}_K^2)(\boldsymbol{Q}_L^T\otimes\boldsymbol{Q}_K^T) \\
&\quad + \varepsilon\sqrt{\tau}(\boldsymbol{Q}_L\otimes\boldsymbol{Q}_K)(\boldsymbol{I}\otimes\boldsymbol{\Lambda}_K)(\boldsymbol{Q}_L^T\otimes\boldsymbol{Q}_K^T) + \varepsilon\sqrt{\tau}(\boldsymbol{Q}_L\otimes\boldsymbol{Q}_K)(\boldsymbol{\Lambda}_L\otimes\boldsymbol{\Lambda}_K)(\boldsymbol{Q}_L^T\otimes\boldsymbol{Q}_K^T)\Big]^{-1} \\
&\quad \left[(\boldsymbol{Q}_L\otimes\boldsymbol{Q}_K)(\boldsymbol{Q}_L^T\otimes\boldsymbol{Q}_K^T) + \varepsilon^2\tau(\boldsymbol{Q}_L\otimes\boldsymbol{Q}_K)(\boldsymbol{\Lambda}_L\otimes\boldsymbol{\Lambda}_K^2)(\boldsymbol{Q}_L^T\otimes\boldsymbol{Q}_K^T)\right] \\
&= \Big[(\boldsymbol{Q}_L\otimes\boldsymbol{Q}_K)\Big(\boldsymbol{I} + \varepsilon^2\tau(\boldsymbol{\Lambda}_L\otimes\boldsymbol{\Lambda}_K^2) + \varepsilon\sqrt{\tau}(\boldsymbol{I}\otimes\boldsymbol{\Lambda}_K) + \varepsilon\sqrt{\tau}(\boldsymbol{\Lambda}_L\otimes\boldsymbol{\Lambda}_K)\Big)(\boldsymbol{Q}_L^T\otimes\boldsymbol{Q}_K^T)\Big]^{-1} \\
&\quad (\boldsymbol{Q}_L\otimes\boldsymbol{Q}_K)\left[\boldsymbol{I} + \varepsilon^2\tau(\boldsymbol{\Lambda}_L\otimes\boldsymbol{\Lambda}_K^2)\right](\boldsymbol{Q}_L^T\otimes\boldsymbol{Q}_K^T) \\
&= (\boldsymbol{Q}_L\otimes\boldsymbol{Q}_K)\left[\boldsymbol{I} + \varepsilon^2\tau(\boldsymbol{\Lambda}_L\otimes\boldsymbol{\Lambda}_K^2) + \varepsilon\sqrt{\tau}(\boldsymbol{I}\otimes\boldsymbol{\Lambda}_K) + \varepsilon\sqrt{\tau}(\boldsymbol{\Lambda}_L\otimes\boldsymbol{\Lambda}_K)\right]^{-1} \\
&\quad \left[\boldsymbol{I} + \varepsilon^2\tau(\boldsymbol{\Lambda}_L\otimes\boldsymbol{\Lambda}_K^2)\right](\boldsymbol{Q}_L^T\otimes\boldsymbol{Q}_K^T).
\end{aligned}$$

It follows that

$$\sigma(\boldsymbol{R}) = \left\{\frac{1 + \varepsilon^2\tau\lambda_{L,i}\lambda_{K,j}^2}{1 + \varepsilon\lambda_{K,j}\left(\varepsilon\tau\lambda_{L,i}\lambda_{K,j} + \sqrt{\tau} + \sqrt{\tau}\lambda_{L,i}\right)} : j = 1,\ldots,m,\ i = 1,\ldots,N\right\}. \tag{4.102}$$

Remember that $\lambda_{K,j}, \lambda_{L,i} \geq 0$ for $j = 1, \dots, m$, $i = 1, \dots, N$. Hence, we obtain the upper bound

$$\frac{1 + \varepsilon^2 \tau \lambda_{L,i} \lambda_{K,j}^2}{1 + \varepsilon^2 \tau \lambda_{L,i} \lambda_{K,j}^2 + \underbrace{\varepsilon \sqrt{\tau} \lambda_{K,j} + \varepsilon \sqrt{\tau} \lambda_{L,i} \lambda_{K,j}}_{\geq 0}} \leq \frac{1 + \varepsilon^2 \tau \lambda_{L,i} \lambda_{K,j}^2}{1 + \varepsilon^2 \tau \lambda_{L,i} \lambda_{K,j}^2} = 1.$$

If $\lambda_{K,j} = 0$, we obtain $1 \in \sigma(\boldsymbol{R})$. If $\lambda_{L,i} = 0$ and $\lambda_{K,j} \neq 0$, we obtain

$$\frac{1}{1 + \varepsilon \sqrt{\tau} \lambda_{K,j}} \in \sigma(\boldsymbol{R}).$$

As a first lower bound for the spectrum of $\boldsymbol{R}$, we get

$$\frac{1}{1 + \varepsilon \sqrt{\tau} \lambda_{K,j}} \geq \frac{1}{1 + \varepsilon \sqrt{\tau} \rho(\boldsymbol{M}^{-\frac{1}{2}} \boldsymbol{K} \boldsymbol{M}^{-\frac{1}{2}})}.$$

In the following, we assume $\lambda_{K,j}, \lambda_{L,i} \neq 0$. We can rewrite the eigenvalues in (4.102) as

$$\frac{1 + \varepsilon^2 \tau \lambda_{L,i} \lambda_{K,j}^2}{1 + \varepsilon^2 \tau \lambda_{L,i} \lambda_{K,j}^2 + \varepsilon \sqrt{\tau} \lambda_{K,j} + \varepsilon \sqrt{\tau} \lambda_{L,i} \lambda_{K,j}} = \frac{1}{1 + \frac{\varepsilon \sqrt{\tau} \lambda_{K,j} (1 + \lambda_{L,i})}{1 + \varepsilon^2 \tau \lambda_{L,i} \lambda_{K,j}^2}}.$$

Using the inequality

$$0 \leq (1 - ab)^2 = 1 + a^2 b^2 - 2ab$$

with $a, b \in \mathbb{R}$, we can bound

$$\frac{\varepsilon \sqrt{\tau} \lambda_{K,j} (1 + \lambda_{L,i})}{1 + \varepsilon^2 \tau \lambda_{K,j} \lambda_{L,i} \lambda_{K,j}} \leq \frac{\varepsilon \sqrt{\tau} \lambda_{K,j} (1 + \lambda_{L,i})}{2 \varepsilon \sqrt{\tau} \sqrt{\lambda_{K,j}} \sqrt{\lambda_{L,i} \lambda_{K,j}}} = \frac{1 + \lambda_{L,i}}{2 \sqrt{\lambda_{L,i}}} \leq \max \left\{ \frac{1 + l_0}{2 \sqrt{l_0}}, \frac{1 + \rho(\boldsymbol{L})}{2 \sqrt{\rho(\boldsymbol{L})}} \right\},$$

where l_0 is the smallest positive eigenvalue of $\boldsymbol{L}$. Here, we have used $a^2 = \varepsilon^2 \tau \lambda_{K,j}$ and $b^2 = \lambda_{L,i} \lambda_{K,j}$. □

Lemma 4.12. *If $\boldsymbol{L} = \boldsymbol{I}$, then*

$$\sigma\left(\hat{\boldsymbol{S}}^{-1} \tilde{\boldsymbol{S}}\right) \subset [0.5, 1].$$

Proof. Due to Proposition 3.10, $\tilde{\boldsymbol{S}}$ and $\hat{\boldsymbol{S}}$ are symmetric positive definite. Hence, we may prove the result using the Rayleigh quotient argument in Theorem 2.29. We write

$$\begin{aligned} \frac{\boldsymbol{v}^T \tilde{\boldsymbol{S}} \boldsymbol{v}}{\boldsymbol{v}^T \hat{\boldsymbol{S}} \boldsymbol{v}} &= \frac{\boldsymbol{v}^T \left(\boldsymbol{I} \otimes \boldsymbol{M} + \tau \varepsilon^2 (\boldsymbol{I} \otimes \boldsymbol{K})(\boldsymbol{I} \otimes \boldsymbol{M})^{-1} (\boldsymbol{I} \otimes \boldsymbol{K}) \right) \boldsymbol{v}}{\boldsymbol{v}^T \left(\boldsymbol{I} \otimes \boldsymbol{M} + \tau \varepsilon^2 (\boldsymbol{I} \otimes \boldsymbol{K})(\boldsymbol{I} \otimes \boldsymbol{M})^{-1} (\boldsymbol{I} \otimes \boldsymbol{K}) + 2 \sqrt{\tau} \varepsilon (\boldsymbol{I} \otimes \boldsymbol{K}) \right) \boldsymbol{v}} \\ &= \frac{\boldsymbol{a}^T \boldsymbol{a} + \boldsymbol{b}^T \boldsymbol{b}}{\boldsymbol{a}^T \boldsymbol{a} + \boldsymbol{b}^T \boldsymbol{b} + 2 \boldsymbol{a}^T \boldsymbol{b}}, \end{aligned}$$

where $\boldsymbol{a} = (\boldsymbol{I} \otimes \boldsymbol{M})^{\frac{1}{2}} \boldsymbol{v}$ and $\boldsymbol{b} = \sqrt{\tau} \varepsilon (\boldsymbol{I} \otimes \boldsymbol{M})^{-\frac{1}{2}} (\boldsymbol{I} \otimes \boldsymbol{K}) \boldsymbol{v}$. From the properties of $\boldsymbol{M}$ and $\boldsymbol{K}$, it follows $\boldsymbol{a}^T \boldsymbol{a} > 0$ and $\boldsymbol{b}^T \boldsymbol{b}, \boldsymbol{a}^T \boldsymbol{b} \geq 0$ and therefore $\frac{\boldsymbol{v}^T \tilde{\boldsymbol{S}} \boldsymbol{v}}{\boldsymbol{v}^T \hat{\boldsymbol{S}} \boldsymbol{v}} \leq 1$. On the other hand, $(\boldsymbol{a} - \boldsymbol{b})^T (\boldsymbol{a} - \boldsymbol{b}) \geq 0$, which gives $\frac{\boldsymbol{v}^T \tilde{\boldsymbol{S}} \boldsymbol{v}}{\boldsymbol{v}^T \hat{\boldsymbol{S}} \boldsymbol{v}} \geq 0.5$. □

Now, let us discuss the practical realization of $\boldsymbol{\mathcal{P}}$. The $(1,1)$ block $\boldsymbol{I} \otimes \boldsymbol{M}$ is a diagonal matrix with positive entries. Hence, its inverse can be performed by elementwise multiplications. The block

$$\boldsymbol{S}_2 = \boldsymbol{I} \otimes \boldsymbol{M} + \varepsilon \sqrt{\tau}(\boldsymbol{I} \otimes \boldsymbol{K}) = \boldsymbol{I} \otimes \left(\boldsymbol{M} + \varepsilon \sqrt{\tau} \boldsymbol{K}\right)$$

in (4.99) is block diagonal and contains the same discrete elliptic operator, $\boldsymbol{M} + \varepsilon \sqrt{\tau} \boldsymbol{K}$, on each diagonal block. Therefore, we approximate the inverse of each diagonal block with one and the same AMG preconditioner. The resulting practical approximation of $\boldsymbol{S}_2$ is

$$\boldsymbol{I} \otimes \mathrm{AMG}\left(\boldsymbol{M} + \varepsilon \sqrt{\tau} \boldsymbol{K}\right). \tag{4.103}$$

It remains to discuss the action of the inverse of the block $\boldsymbol{S}_2$ in (4.99). Due to the presence of $\boldsymbol{L}$, $\boldsymbol{S}_2$ is in general not block diagonal. During the rest of this section, we restrict the class of considerations to diagonal and circulant (see Definition 2.28) matrices $\boldsymbol{L}$. More general forms are a topic of further research and are not discussed in this thesis. As reference examples, we take $\boldsymbol{L} = \boldsymbol{I}$ (used, e.g., in [118]) and $\boldsymbol{L} = \boldsymbol{I} - \frac{1}{N}\boldsymbol{1}\boldsymbol{1}^T$ (used, e.g., in [59, 85]). The latter is a circulant matrix. Moreover, it satisfies the condition $\boldsymbol{L}\boldsymbol{1} = \boldsymbol{0}$ in (4.11). In the former case $\boldsymbol{L} = \boldsymbol{I}$, we obtain $\boldsymbol{S}_1 = \boldsymbol{S}_2$. Hence, the action of the inverse of $\boldsymbol{S}_1$ is implemented as in (4.103). This results in the following practical Schur complement approximation

$$\boldsymbol{S}_0 = \left[\boldsymbol{I} \otimes \mathrm{AMG}\left(\boldsymbol{M} + \varepsilon \sqrt{\tau} \boldsymbol{K}\right)\right] (\boldsymbol{I} \otimes \boldsymbol{M})^{-1} \left[\boldsymbol{I} \otimes \mathrm{AMG}\left(\boldsymbol{M} + \varepsilon \sqrt{\tau} \boldsymbol{K}\right)\right].$$

Independently from the number N of phases, we have to initialize only one AMG preconditioner.

Now, let us study the case $\boldsymbol{L} = \boldsymbol{I} - \frac{1}{N}\boldsymbol{1}\boldsymbol{1}^T$, which serves as a reference example for circulant matrices $\boldsymbol{L}$. Then, $\boldsymbol{S}_1$ is not block diagonal anymore. In fact, each block is occupied by $\boldsymbol{K}$. However, as $\boldsymbol{L}$ is a circulant matrix, it can be diagonalized using the discrete Fourier transform matrix $\boldsymbol{F}_N$ (see Definition 2.29 and Theorem 2.33), i.e.,

$$\boldsymbol{L} = \boldsymbol{F}_N \, \mathrm{diag}(\lambda_{L,1}, \ldots, \lambda_{L,N}) \, \boldsymbol{F}_N^{-1},$$

see [47]. This property forms the basis of an efficient fast Fourier transform (FFT) based preconditioner, which is used, e.g., in [143], and briefly reviewed in the following. In our case, we do not merely diagonalize $\boldsymbol{L}$, but rather the whole block matrix $\boldsymbol{S}_1$ since $\boldsymbol{S}_1$ is the matrix whose inverse we have to apply. More precisely, if we apply the FFT to a system of the form $\boldsymbol{S}_1\boldsymbol{y} = \boldsymbol{g}$, we get an equivalent system with the block diagonal system matrix

$$(\boldsymbol{F}_N^{-1} \otimes \boldsymbol{I})\boldsymbol{S}_1(\boldsymbol{F}_N \otimes \boldsymbol{I}) = \boldsymbol{I} \otimes \boldsymbol{M} + \sqrt{\tau}\varepsilon \, \mathrm{diag}(\lambda_{L,1}, \ldots, \lambda_{L,N}) \otimes \boldsymbol{K}. \tag{4.104}$$

Inserting the eigenvalues of $\boldsymbol{L}$, which are $\lambda_{L,1} = 0$ and $\lambda_{L,2} = \ldots = \lambda_{L,N} = 1$, we see that the resulting approximation in (4.104) is of block diagonal form, and almost all diagonal blocks are equal. In fact, only two different diagonal blocks occur: $\boldsymbol{M}$ for the case $\lambda_{L,1} = 0$ and $\boldsymbol{M} + \sqrt{\tau}\varepsilon\boldsymbol{K}$ for all remaining eigenvalues $\lambda_{L,j} = 1$, $j = 2, \ldots, N$. As the application of the Fourier transform will in general result in complex valued systems, we formulate the N blocks in (4.104) to 2×2 real-valued block systems. In detail, we have to solve two types of systems: One time

$$\mathcal{A}_L \begin{bmatrix} \tilde{\boldsymbol{y}}_r \\ \tilde{\boldsymbol{y}}_c \end{bmatrix} = \begin{bmatrix} \boldsymbol{M} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{y}}_r \\ \tilde{\boldsymbol{y}}_c \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{g}}_r \\ \tilde{\boldsymbol{g}}_c \end{bmatrix} \tag{4.105}$$

and $(N-1)$ times

$$\mathcal{A}_L \begin{bmatrix} \tilde{\boldsymbol{y}}_r \\ \tilde{\boldsymbol{y}}_c \end{bmatrix} = \begin{bmatrix} \boldsymbol{M} + \sqrt{\tau}\varepsilon \boldsymbol{K} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M} + \sqrt{\tau}\varepsilon \boldsymbol{K} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{y}}_r \\ \tilde{\boldsymbol{y}}_c \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{g}}_r \\ \tilde{\boldsymbol{g}}_c \end{bmatrix}. \tag{4.106}$$

Again, the system (4.105) arises for the diagonal block with $\lambda_{L,1} = 0$ and (4.106) for all remaining eigenvalues $\lambda_{L,j}$, $j = 2, \ldots, N$. As in [143], we solve each of the above N real-valued systems with a fixed number of steps of an inexact Uzawa-type method

$$\begin{bmatrix} \tilde{\boldsymbol{y}}_r \\ \tilde{\boldsymbol{y}}_c \end{bmatrix}^{(l+1)} = \begin{bmatrix} \tilde{\boldsymbol{y}}_r \\ \tilde{\boldsymbol{y}}_c \end{bmatrix}^{(l)} + \omega \mathcal{P}_L^{-1} \begin{bmatrix} \tilde{\boldsymbol{r}}_r \\ \tilde{\boldsymbol{r}}_c \end{bmatrix},$$

where

$$\begin{bmatrix} \tilde{\boldsymbol{r}}_r \\ \tilde{\boldsymbol{r}}_c \end{bmatrix}^{(l)} = \begin{bmatrix} \tilde{\boldsymbol{g}}_r \\ \tilde{\boldsymbol{g}}_c \end{bmatrix} - \mathcal{A}_L \begin{bmatrix} \tilde{\boldsymbol{y}}_r \\ \tilde{\boldsymbol{y}}_c \end{bmatrix}^{(l)}$$

is the lth residual, and ω is the relaxation parameter. In the case (4.105), we use the preconditioner

$$\mathcal{P}_L = \begin{bmatrix} \boldsymbol{M} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M} \end{bmatrix},$$

and its inverse can be performed by elementwise multiplications. In the case (4.106), we use the preconditioner

$$\mathcal{P}_L = \begin{bmatrix} \mathrm{AMG}(\boldsymbol{M} + \sqrt{\tau}\varepsilon \boldsymbol{K}) & \boldsymbol{0} \\ \boldsymbol{0} & \mathrm{AMG}(\boldsymbol{M} + \sqrt{\tau}\varepsilon \boldsymbol{K}) \end{bmatrix}.$$

Again, independently from the number N of phases, we have to initialize only one AMG preconditioner.

Here, we finish the discussion about preconditioning of smooth Cahn–Hilliard systems. In Section 4.8.1, we illustrate the robust performance of our developed preconditioner applied with BiCG. Next, we come to the harder case of nonsmooth systems. We will see that a simplification of the coefficient matrix in form of $\mathcal{A}_{\boldsymbol{0}}$ is not satisfying anymore.

4.7.2 Nonsmooth systems

In the following, we develop a preconditioner for the nonsmooth Cahn–Hilliard system represented in (4.82). The outer structure is the same as we have in the smooth system. However, as in the previous chapter, neglecting the penalty blocks $\boldsymbol{G}_{(i)}$, $i = 1, \ldots, N$, in $\boldsymbol{A}$ would give a worse approximation for small penalization parameters, which is summarized below in Theorem 4.13. The penalization is even more crucial than in the last chapter. As can be seen from (4.79), penalized entries are in general scattered throughout the diagonals of every block of $\boldsymbol{A}$. In particular, the penalized entries of the nondiagonal blocks $\boldsymbol{A}_{(i)}$, $i = 1, \ldots, N$, of $\boldsymbol{A}$ lie in the interval $-\frac{h^d}{N}\left(c^{-1} - 1\right)[C, \tilde{c}]$, since $c < 1$. The nonpenalized entries lie in the interval $\frac{h^d}{N}[\tilde{c}, C]$. This also applies to the diagonal blocks $\boldsymbol{A}_{(i,i)}$, $i = 1, \ldots, N$, of $\boldsymbol{A}$, whereby the matrix $\varepsilon^2 \boldsymbol{K}$ comes in addition. Again, this indicates a severe dependency between ε and c and hence h. Note that sufficient sizes of c are $c \leq 10^{-4}$, and in our numerical examples we usually work with $c = 10^{-7}$. Moreover, we have in mind that we want to go over to adaptive mesh strategies, where we coarsen the mesh inside the penalized regions. All in all, the estimated order of penalized entries is usually of large size and highly differs to the order of the remaining nonpenalized entries. Hence, they should not be neglected.

Theorem 4.13. *Let*

$$\mathcal{A}_0 = \begin{bmatrix} \boldsymbol{I}\otimes\boldsymbol{M} & -\varepsilon^2\boldsymbol{I}\otimes\boldsymbol{K} \\ \tau\boldsymbol{L}\otimes\boldsymbol{K} & \boldsymbol{I}\otimes\boldsymbol{M} \end{bmatrix}.$$

It holds

$$\sigma(\mathcal{A}_{\boldsymbol{0}}^{-1}\mathcal{A}) \subset B_\varsigma(1).$$

The circle radius is bounded by $\varsigma < \frac{\sqrt{\tau\rho(\boldsymbol{L})}}{c\varepsilon}$. *In particular, Nm eigenvalues are equal to one. We get $\varsigma < 0.5$ when $\tau \leq 0.25\, c^2\varepsilon^2\rho(\boldsymbol{L})^{-1}$.*

Proof. The proof is almost the same as the one for Theorem 4.10. The modification occurs at the end in (4.97)

$$\|\tilde{\boldsymbol{R}}\| \leq \frac{\tau\sqrt{\rho(\boldsymbol{L})}}{2\varepsilon\sqrt{\tau}}\|\tilde{\boldsymbol{A}}\|, \tag{4.107}$$

where $\|\tilde{\boldsymbol{A}}\|$ differs to the one in the smooth system. Due to the diagonal structure of each block in $\boldsymbol{A}-\varepsilon^2(\boldsymbol{I}\otimes\boldsymbol{K})$, we have

$$\begin{aligned}\tilde{\boldsymbol{A}} &= (\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})\left[\boldsymbol{A}-\varepsilon^2(\boldsymbol{I}\otimes\boldsymbol{K})\right](\boldsymbol{I}\otimes\boldsymbol{M}^{-\frac{1}{2}})\\ &= \frac{1}{N}\begin{bmatrix} (N-1)\left(c^{-1}\tilde{\boldsymbol{G}}_{(1)}-\boldsymbol{I}\right) & -\left(c^{-1}\tilde{\boldsymbol{G}}_{(2)}-\boldsymbol{I}\right) & \cdots & -\left(c^{-1}\tilde{\boldsymbol{G}}_{(N)}-\boldsymbol{I}\right)\\ -\left(c^{-1}\tilde{\boldsymbol{G}}_{(1)}-\boldsymbol{I}\right) & (N-1)\left(c^{-1}\tilde{\boldsymbol{G}}_{(2)}-\boldsymbol{I}\right) & \cdots & -\left(c^{-1}\tilde{\boldsymbol{G}}_{(N)}-\boldsymbol{I}\right)\\ \vdots & \vdots & \ddots & \vdots\\ -\left(c^{-1}\tilde{\boldsymbol{G}}_{(1)}-\boldsymbol{I}\right) & -\left(c^{-1}\tilde{\boldsymbol{G}}_{(2)}-\boldsymbol{I}\right) & \cdots & (N-1)\left(c^{-1}\tilde{\boldsymbol{G}}_{(N)}-\boldsymbol{I}\right)\end{bmatrix},\end{aligned}$$

where

$$\tilde{\boldsymbol{G}}_{(i)} = \tilde{\boldsymbol{G}}_{(i)}(\boldsymbol{u}_i^{(k)}) = \operatorname{diag}\begin{pmatrix} 1 & \text{if } u_{h,i,j}^{(k)} < 0,\\ 0 & \text{otherwise.}\end{pmatrix}_{i=1,\dots,m}$$

Therefore,

$$c^{-1}\tilde{\boldsymbol{G}}_{(i)} - \boldsymbol{I} = \operatorname{diag}\begin{pmatrix} c^{-1}-1 & \text{if } u_{h,i,j}^{(k)} < 0,\\ -1 & \text{otherwise,}\end{pmatrix}_{i=1,\dots,m}$$

and the maximum diagonal entry of each block $c^{-1}\tilde{\boldsymbol{G}}_{(i)}-\boldsymbol{I}$, $i=1,\dots,N$, is $c^{-1}-1$. This is because the penalty parameter c should be close to zero. Since each block in $\tilde{\boldsymbol{A}}$ is diagonal, the number of nonzero entries per row or column is N. Moreover,

$$\|\tilde{\boldsymbol{A}}\|_1 \leq 2\frac{N-1}{N}\left(c^{-1}-1\right) < 2\frac{N-1}{N}c^{-1} \quad\text{and}\quad \|\tilde{\boldsymbol{A}}\|_\infty \leq 2\frac{N-1}{N}\left(c^{-1}-1\right) < 2\frac{N-1}{N}c^{-1}.$$

Thus, (2.12) yields

$$\|\tilde{\boldsymbol{A}}\| \leq \sqrt{\|\tilde{\boldsymbol{A}}\|_1\|\tilde{\boldsymbol{A}}\|_\infty} < 2\frac{N-1}{N}c^{-1}.$$

Hence, we obtain in (4.107)

$$\|\tilde{\boldsymbol{R}}\| \leq \frac{N-1}{N}c^{-1}\frac{\sqrt{\tau}\sqrt{\rho(\boldsymbol{L})}}{\varepsilon} < \frac{\sqrt{\tau\rho(\boldsymbol{L})}}{c\varepsilon}. \tag{4.108}$$

Therefore, for $\tau \leq 0.25\, c^2\varepsilon^2\rho(\boldsymbol{L})^{-1}$, it holds $\sigma(\tilde{\boldsymbol{R}}) = \sigma(\boldsymbol{R}) \subset B_{0.5}(0)$ and hence $\sigma(\mathcal{A}_{\boldsymbol{0}}^{-1}\mathcal{A}) \subset B_{0.5}(1)$. □

Hence, neglecting the blocks $\boldsymbol{G}_{(i)}$, $i = 1, \ldots, N$, in $\boldsymbol{A}$, as done in the last section for smooth systems, would only be satisfying for tiny time step sizes τ, which is far away from being practical. Therefore, we build on the last chapter and keep the whole block $\boldsymbol{A}$ within our Schur complement approximation. Our proposed Schur complement preconditioner is

$$\begin{aligned}\hat{\boldsymbol{S}} &= \boldsymbol{S}_1(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}\boldsymbol{S}_2\\ &= \left(\frac{N}{(N-1)\sqrt{\varepsilon}}(\boldsymbol{I}\otimes\boldsymbol{M}) + \sqrt{\tau}(\boldsymbol{L}\otimes\boldsymbol{K})\right)(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}\left(\frac{(N-1)\sqrt{\varepsilon}}{N}(\boldsymbol{I}\otimes\boldsymbol{M}) + \sqrt{\tau}\boldsymbol{A}\right) \qquad (4.109)\\ &= \boldsymbol{I}\otimes\boldsymbol{M} + \tau(\boldsymbol{L}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}\boldsymbol{A} + \frac{\sqrt{\tau\varepsilon}(N-1)}{N}(\boldsymbol{L}\otimes\boldsymbol{K}) + \frac{\sqrt{\tau}N}{\sqrt{\varepsilon}(N-1)}\boldsymbol{A}. \qquad (4.110)\end{aligned}$$

The first two terms in (4.110) match the exact Schur complement. Due to the balanced distribution of τ in form of $\sqrt{\tau}$ as well as the scaling with $\frac{N}{(N-1)\sqrt{\varepsilon}}$ and its inverse in both factors, $\hat{S}_1$ and $\hat{S}_2$, the influence of both remainder terms in (4.110) is reduced. Let us discuss the action of the inverses of $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$. The former was already studied in the previous section. Therefore, let us concentrate on the latter now. The factor $\boldsymbol{S}_2$ still contains the nondiagonal block matrix $\boldsymbol{A}$ shifted by scaled mass matrices. This shift ensures the positive definiteness of the diagonal blocks of $\boldsymbol{S}_2$. More precisely, if we write out the diagonal blocks of $\boldsymbol{S}_2$,

$$\sqrt{\tau}\varepsilon^2\boldsymbol{K} + \frac{\sqrt{\tau}(N-1)}{Nc}\boldsymbol{G}_{(i)} + \frac{N-1}{N}\left(\sqrt{\varepsilon} - \sqrt{\tau}\right)\boldsymbol{M},$$

for $i = 1, \ldots, N$, we see that they are positive definite if $\tau < \varepsilon$. Remember the uniqueness condition $\tau < 4\varepsilon^2\lambda_{\max}(\boldsymbol{T})^{-2}\rho(\boldsymbol{L})^{-1}$ that is imposed on our time-discrete formulation anyway. For our choices of $\boldsymbol{T}$ and $\boldsymbol{L}$, the uniqueness condition reads $\tau < 4\varepsilon^2$. For $\varepsilon \leq 0.25$, it holds $4\varepsilon^2 < \varepsilon$. In the numerical experiments, we never use interfacial parameters larger than 0.25. Instead, it should be as small as possible. Hence, the positive definiteness of the diagonal blocks of $\boldsymbol{S}_2$ is always guaranteed in our numerical experiments.

Remark 4.6. *For other choices of $\boldsymbol{T}$ or $\boldsymbol{L}$, where $\lambda_{\max}(\boldsymbol{T})$ or $\rho(\boldsymbol{L})$ is not equal to one, one might include this information in the Schur complement approximation. For instance,*

$$\hat{\boldsymbol{S}} = \left(\frac{N\lambda_{\max}(\boldsymbol{T})\sqrt{\rho(\boldsymbol{L})}}{(N-1)\sqrt{\varepsilon}}(\boldsymbol{I}\otimes\boldsymbol{M}) + \sqrt{\tau}(\boldsymbol{L}\otimes\boldsymbol{K})\right)$$
$$(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}\left(\frac{(N-1)\sqrt{\varepsilon}}{N\lambda_{\max}(\boldsymbol{T})\sqrt{\rho(\boldsymbol{L})}}(\boldsymbol{I}\otimes\boldsymbol{M}) + \sqrt{\tau}\boldsymbol{A}\right)$$

is a possible candidate. The discussion about further possibilities of $\boldsymbol{T}$ or $\boldsymbol{L}$ is a topic of future research.

The proposed strategy concerning the solution of a system of the form $\boldsymbol{S}_2\boldsymbol{y} = \boldsymbol{g}$ is the use of a block Jacobi method with a fixed number of steps:

$$\boldsymbol{y}^{(l+1)} = \boldsymbol{y}^{(l)} + \omega\mathcal{P}_{\boldsymbol{A}}^{-1}\boldsymbol{r}^{(l)},$$

where

$$\boldsymbol{r}^{(l)} = \boldsymbol{g} - \boldsymbol{S}_2 \boldsymbol{y}^{(l)}$$

is the lth residual, and ω is the relaxation parameter. We use the preconditioner

$$\boldsymbol{\mathcal{P}}_{\boldsymbol{A}} = \begin{bmatrix} \mathrm{AMG}\left(\frac{(N-1)\sqrt{\varepsilon}}{N}\boldsymbol{M} + \sqrt{\tau}\boldsymbol{A}_{(1,1)}\right) & & \\ & \ddots & \\ & & \mathrm{AMG}\left(\frac{(N-1)\sqrt{\varepsilon}}{N}\boldsymbol{M} + \sqrt{\tau}\boldsymbol{A}_{(N,N)}\right) \end{bmatrix},$$

which is an AMG approximation of the block diagonal matrix of $\boldsymbol{S}_2$. In contrast to the previous section, we have to initialize N AMG preconditioners instead of one. Moreover, they have to be recomputed in every Newton step since the position of penalized entries is changing with every Newton step. In detail, the penalized entries in the blocks $\boldsymbol{A}_{(i)}$ or $\boldsymbol{A}_{(i,i)}$ depend on the phase $\boldsymbol{u}_i^{(k)}$. Since all phases are separated in the domain (at least after a few time steps), one cannot expect the penalty parameter to act in the same regions for all phases. That is why an approximation of the matrix $\boldsymbol{A}$, where all diagonal blocks are equal, seems not to be of good quality, and our experiences confirm this observation. Nevertheless, Section 4.8.1 shows a promising performance of our developed preconditioner applied with BiCGstab.

Here, we finish the theoretical discussion of the preconditioners. In the next section, we illustrate their efficiency via various numerical experiments.

4.8 Numerical Results

In this section, we show numerical results for the presented vector-valued Cahn–Hilliard problems. First, we explain our implementation framework. This is already described at the beginning of Chapter 3.8 for the most part. Hence, we only add the differences here.

The connection between the spatial mesh size h, the interfacial parameter ε, and the time step size τ is as follows. As discussed in Chapter 1.1, it is essential to ensure that at least eight grid points lie on the interface. Using smooth potentials, this leads to the condition $h \le 4\sqrt{2}\varepsilon \cdot \operatorname{atanh}(0.9)/9$. Using nonsmooth potentials, this leads to the condition $h \le \varepsilon\pi/9$. As far as we know, there is no theory available for the case of the regularized nonsmooth potential. In our numerical examples, we use the condition $h \le \varepsilon\pi/9$ for the regularized nonsmooth potential. Regarding the time step size, we have analyzed the time step conditions for the smooth and nonsmooth implicit time-discrete system in Section 4.3. For the former, we use $\tau < \varepsilon^2/(3\rho(\boldsymbol{L}))$ and for the latter it is $\tau < 4\varepsilon^2/(\lambda_{\max}(\boldsymbol{T})^2\rho(\boldsymbol{L}))$. As hinted in the previous two sections, we focus in the following on the matrices $\boldsymbol{T} = \boldsymbol{I} - \boldsymbol{1}\boldsymbol{1}^T$, $\boldsymbol{L} = \boldsymbol{I}$, and $\boldsymbol{L} = \boldsymbol{I} - \frac{1}{N}\boldsymbol{1}\boldsymbol{1}^T$. This leads to the time step condition $\tau < \varepsilon^2/3$ in the smooth case and $\tau < 4\varepsilon^2$ in the nonsmooth case.

The algorithm for the numerical solution of the vector-valued Cahn–Hilliard problem with a nonsmooth potential is basically the same as in the last chapter; see Algorithm 3.1 and 3.2. The formulation with a smooth potential is a simplification of the presented two algorithms. If not mentioned otherwise, the initial state $\boldsymbol{u}^{(0)}$ is created via a number of randomly distributed circles with slightly different radii, which are separated by an interfacial area. The N phases are randomly distributed among the circles and the surrounding area. Within the interfacial area, we set all initial phase variables to the value $1/N$ such that they sum up to one.

Now, we are ready for numerical results.

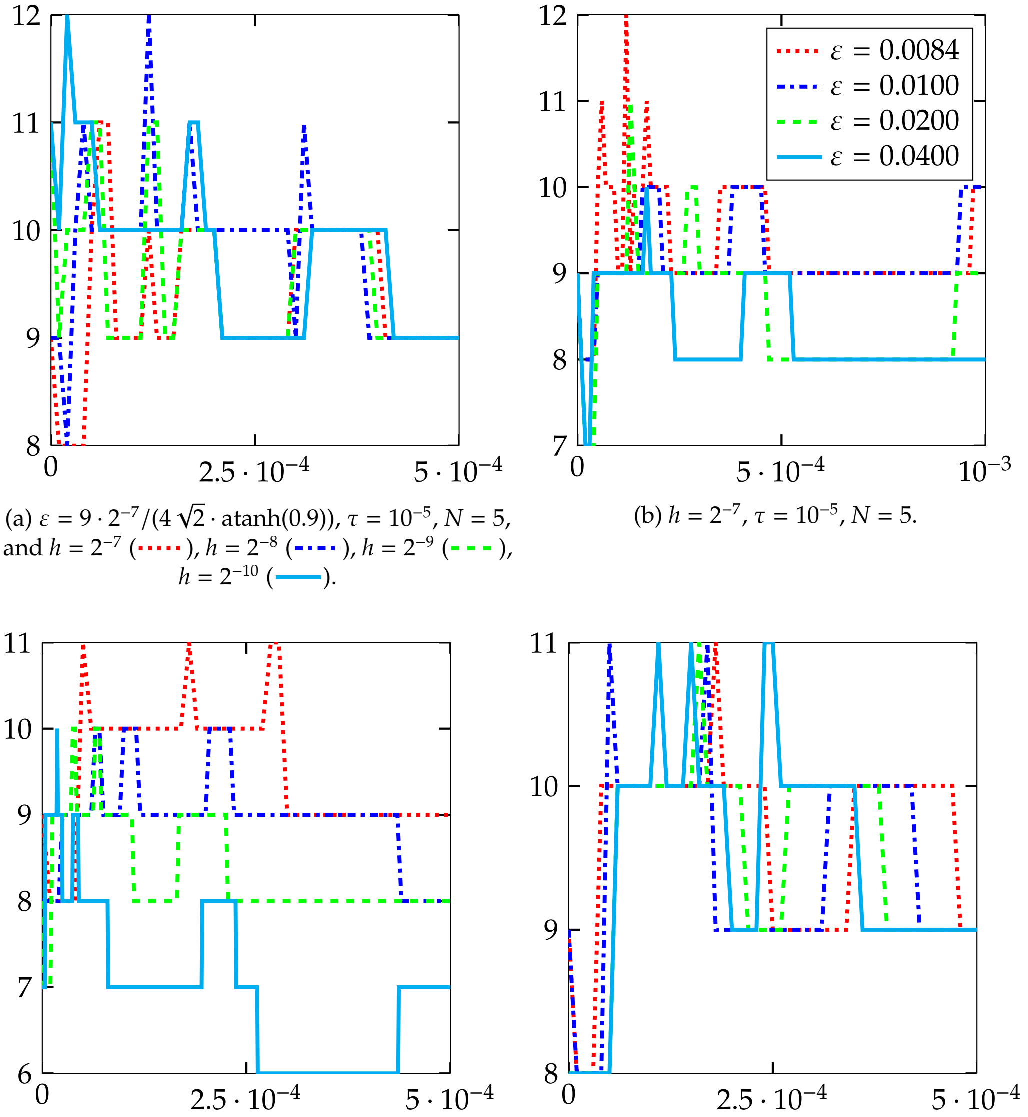
4.8.1 Robustness

In this section, we demonstrate the robustness of our proposed preconditioners regarding all model parameters.

We start with the smooth system in (4.82) with $\boldsymbol{L} = \boldsymbol{I}$, the preconditioner (4.83) and the Schur complement approximation (4.100). Each subplot in Figure 4.1 demonstrates the robustness with respect to a different model parameter. In Figure 4.1(a), we vary the mesh size h while fixing $\varepsilon = 9 \cdot 2^{-7}/(4\sqrt{2} \cdot \operatorname{atanh}(0.9))$, $\tau = 10^{-5}$, $N = 5$, and $T = 5 \cdot 10^{-4}$. In Figure 4.1(b), we vary the interfacial parameter ε while fixing $h = 2^{-7}$, $\tau = 10^{-5}$, $N = 5$, and $T = 10^{-3}$. In Figure 4.1(c), we vary the time step size τ while fixing $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/(4\sqrt{2} \cdot \operatorname{atanh}(0.9))$, $N = 5$, and $T = 5 \cdot 10^{-4}$. In Figure 4.1(d), we vary the number of phases N while fixing $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/(4\sqrt{2} \cdot \operatorname{atanh}(0.9))$, $\tau = 10^{-5}$, and $T = 5 \cdot 10^{-4}$. All in all, the four subplots illustrate the independence of our developed preconditioner regarding all parameters. Finally, in Figure 4.2(a), we vary simultaneously all three parameters h, ε, τ while fixing $N = 5$ and $T = 5 \cdot 10^{-4}$. Table 4.1 illustrates the maximum and average number of Newton iterations, the maximum and average number of BiCG iterations, the average CPU time (in seconds) for BiCG, and the CPU time (in seconds) for the whole simulation for each of the five subplots, respectively.

We repeat the tests above for the case $\boldsymbol{L} = \boldsymbol{I} - \frac{1}{N}\boldsymbol{1}\boldsymbol{1}^T$. Each subplot in Figure 4.3 demonstrates the robustness with respect to a different model parameter. In Figure 4.2(b), we vary simultaneously all three parameters h, ε, τ while fixing $N = 5$ and $T = 5 \cdot 10^{-4}$. Table 4.2 illustrates the maximum and average number of Newton iterations, the maximum and average number of BiCG iterations, the average CPU time (in seconds) for BiCG, and the CPU time (in seconds) for the whole simulation for each of the five subplots, respectively.

We proceed with the nonsmooth system in (4.82) with $\boldsymbol{L} = \boldsymbol{I}$, the preconditioner (4.83) and the Schur complement approximation (4.109). Each subplot in Figure 4.4 and 4.5(a) demonstrates the robustness with respect to a different model parameter. In Figure 4.4(a), we vary the mesh size h while fixing $n_c = 20$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 10^{-3}$, $N = 5$, $c_{p_{\max}} = 10^{-7}$, and $T = 2 \cdot 10^{-2}$. In Figure 4.4(b), we vary the interfacial parameter ε while fixing $n_c = 20$, $h = 2^{-7}$, $\tau = 10^{-3}$, $N = 5$, $c_{p_{\max}} = 10^{-7}$, and $T = 2 \cdot 10^{-2}$. In Figure 4.4(c), we vary the time step size τ while fixing $n_c = 5$, $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $N = 5$, $c_{p_{\max}} = 10^{-7}$, and $T = 2 \cdot 10^{-2}$. In Figure 4.4(d), we vary the number of phases N while fixing $n_c = 20$, $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 10^{-3}$, $c_{p_{\max}} = 10^{-7}$, and $T = 2 \cdot 10^{-2}$. In Figure 4.5(a), we vary the penalty parameter $c_{p_{\max}}$ while fixing $n_c = 20$, $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 10^{-3}$, $N = 5$, and $T = 2 \cdot 10^{-2}$. All in all, the five subplots illustrate the efficiency of our developed preconditioner regarding all parameters. We observe a benign increase of iteration numbers when ε is decreased as well as when N is increased. Finally, in Figure 4.5(b), we vary simultaneously all three parameters h, ε, τ while fixing $n_c = 5$, $N = 5$, $c_{p_{\max}} = 10^{-7}$, and $n = 20$. Ta-



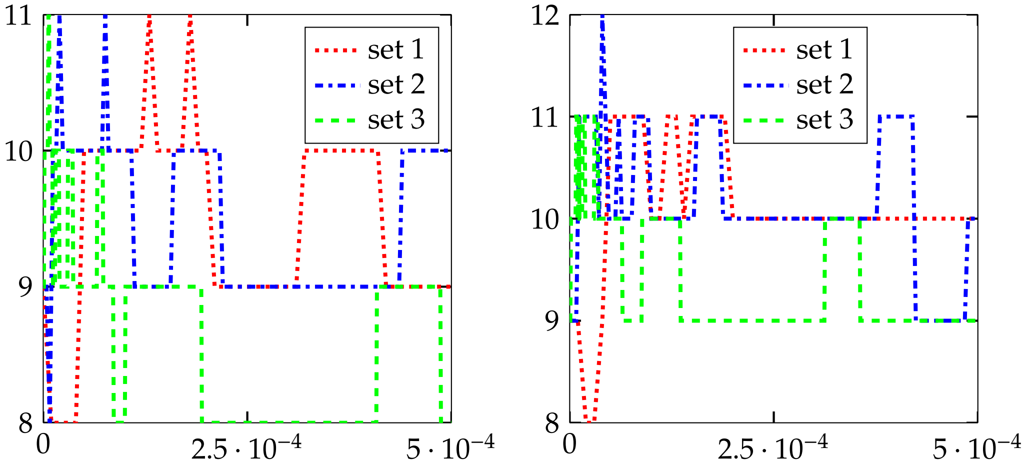
(a) $\varepsilon = 9 \cdot 2^{-7}/(4\sqrt{2} \cdot \text{atanh}(0.9))$, $\tau = 10^{-5}$, $N = 5$, and $h = 2^{-7}$ (⋯⋯), $h = 2^{-8}$ (-·-·), $h = 2^{-9}$ (- - -), $h = 2^{-10}$ (——).

(b) $h = 2^{-7}$, $\tau = 10^{-5}$, $N = 5$.

(c) $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/(4\sqrt{2} \cdot \text{atanh}(0.9))$, $N = 5$, and $\tau = 10^{-5}$ (⋯⋯), $\tau = 5 \cdot 10^{-6}$ (-·-·), $\tau = 2.5 \cdot 10^{-6}$ (- - -), $\tau = 8 \cdot 10^{-7}$ (——).

(d) $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/(4\sqrt{2} \cdot \text{atanh}(0.9))$, $\tau = 10^{-5}$, and $N = 3$ (⋯⋯), $N = 5$ (-·-·), $N = 10$ (- - -), $N = 20$ (——).

Figure 4.1: Results for the solution of the smooth system (4.82) with $\boldsymbol{L} = \boldsymbol{I}$, the preconditioner (4.83), and the Schur complement approximation (4.100). The x-axis shows the time t and the y-axis the average number of BiCG iterations per Newton step.



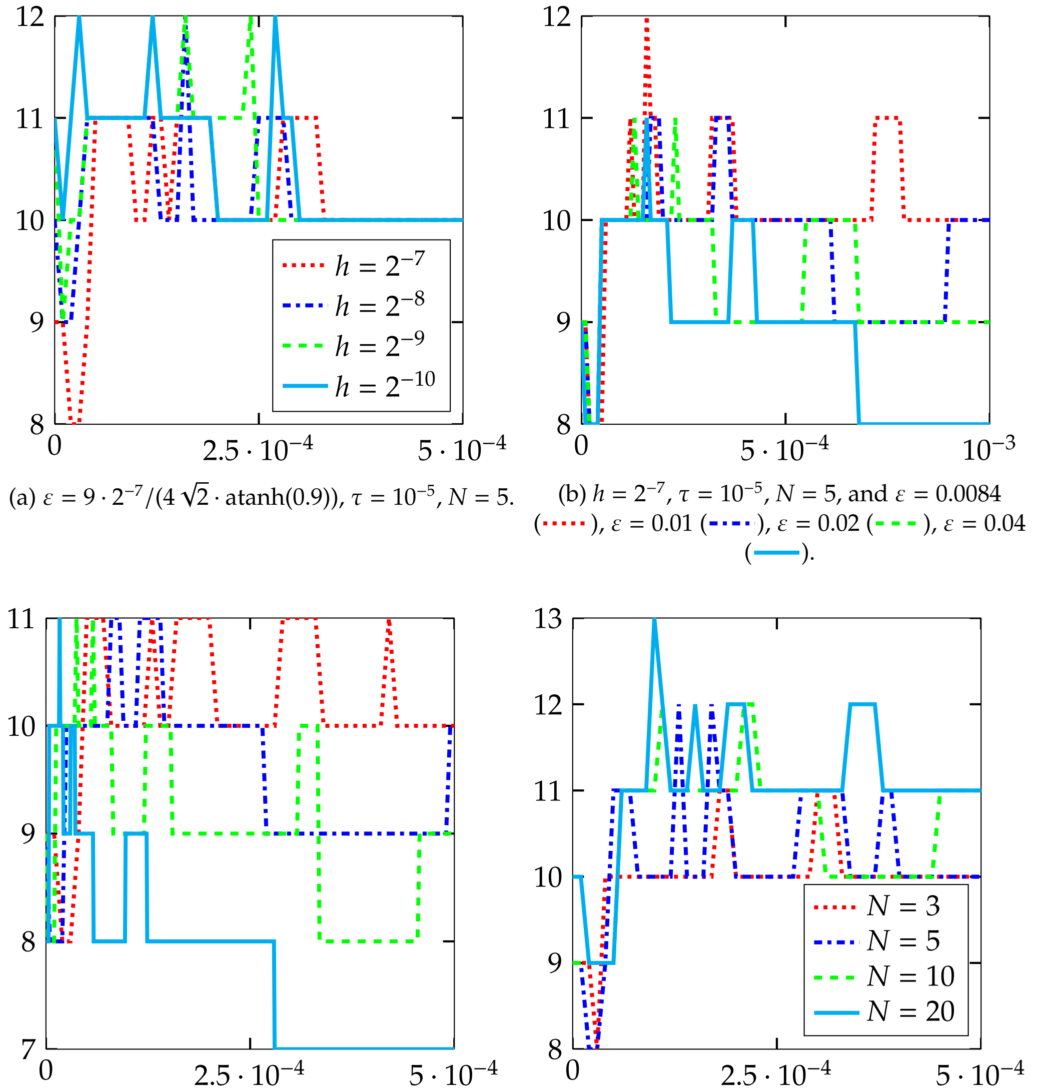
(a) $\boldsymbol{L} = \boldsymbol{I}$.

(b) $\boldsymbol{L} = \boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T$.

Figure 4.2: Results for the solution of the smooth system (4.82) with the preconditioner (4.83) and the Schur complement approximation (4.100). The x-axis shows the time t and the y-axis the average number of BiCG iterations per Newton step. Set j: $h_j = 2^{-j-6}$, $\varepsilon_j = 9\,h_j/(4\sqrt{2} \cdot \operatorname{atanh}(0.9))$, $\tau_1 = 10^{-5}$, $\tau_2 = 4 \cdot 10^{-6}$, $\tau_3 = 8 \cdot 10^{-7}$, $N = 5$ for $j = 1, 2, 3$.

Simulation		Newton		BiCG			
Figure	Plot	Max	Avg	Max	Avg	CPU (s)	CPU (s)
4.1(a)	(······)	2	1	11	9	25	1430
	(-·-·-)	2	1	12	10	102	5589
	(- - -)	2	1	11	10	470	25139
	(——)	2	1	12	10	1785	95731
4.1(b)	(······)	2	1	12	9	24	2632
	(-·-·-)	2	1	10	9	24	2674
	(- - -)	2	1	11	9	22	2502
	(——)	2	1	10	8	22	2438
4.1(c)	(······)	2	1	11	9	25	1443
	(-·-·-)	2	1	10	9	24	2611
	(- - -)	2	1	10	8	22	4763
	(——)	2	1	10	7	19	12539
4.1(d)	(······)	2	1	11	10	15	879
	(-·-·-)	2	1	11	9	25	1440
	(- - -)	2	1	11	9	51	2968
	(——)	2	1	11	9	105	6021
4.2(a)	(······)	2	1	11	9	25	1453
	(-·-·-)	2	1	11	9	100	13064
	(- - -)	2	1	11	9	435	275281

Table 4.1: Results for the solution of the smooth system (4.82) with $\boldsymbol{L} = \boldsymbol{I}$, the preconditioner (4.83), and the Schur complement approximation (4.100): The maximum and average number of Newton iterations, the maximum and average number of BiCG iterations, the average CPU time (in seconds) for BiCG, and the CPU time (in seconds) for the whole simulation.



(a) $\varepsilon = 9 \cdot 2^{-7}/(4\sqrt{2} \cdot \text{atanh}(0.9))$, $\tau = 10^{-5}$, $N = 5$.

(b) $h = 2^{-7}$, $\tau = 10^{-5}$, $N = 5$, and $\varepsilon = 0.0084$ (⋯⋯), $\varepsilon = 0.01$ (-·-·), $\varepsilon = 0.02$ (- - -), $\varepsilon = 0.04$ (—).

(c) $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/(4\sqrt{2} \cdot \text{atanh}(0.9))$, $N = 5$, and $\tau = 10^{-5}$ (⋯⋯), $\tau = 5 \cdot 10^{-6}$ (-·-·), $\tau = 2.5 \cdot 10^{-6}$ (- - -), $\tau = 8 \cdot 10^{-7}$ (—).

(d) $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/(4\sqrt{2} \cdot \text{atanh}(0.9))$, $\tau = 10^{-5}$.

Figure 4.3: Results for the solution of the smooth system (4.82) with $\boldsymbol{L} = \boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T$, the preconditioner (4.83), and the Schur complement approximation (4.100). The x-axis shows the time t and the y-axis the average number of BiCG iterations per Newton step.

Simulation		Newton		BiCG			
Figure	Plot	Max	Avg	Max	Avg	CPU (s)	CPU (s)
4.3(a)	(·····)	2	1	11	10	102	5665
	(-·-·)	2	1	12	10	344	18552
	(- - -)	2	1	12	10	1604	85169
	(——)	2	1	12	10	5932	315821
4.3(b)	(·····)	2	1	12	10	101	10779
	(-·-·)	2	1	11	10	97	10289
	(- - -)	2	1	11	9	95	10087
	(——)	2	1	11	9	88	9380
4.3(c)	(·····)	2	1	11	10	102	5675
	(-·-·)	2	1	11	10	94	9998
	(- - -)	2	1	11	9	91	18863
	(——)	2	1	11	8	80	50744
4.3(d)	(·····)	2	1	11	10	65	3592
	(-·-·)	2	1	12	10	102	5665
	(- - -)	2	1	12	11	202	11426
	(——)	2	1	13	11	417	23518
4.2(b)	(·····)	2	1	11	10	101	5637
	(-·-·)	2	1	12	10	337	43440
	(- - -)	2	1	11	9	1417	891781

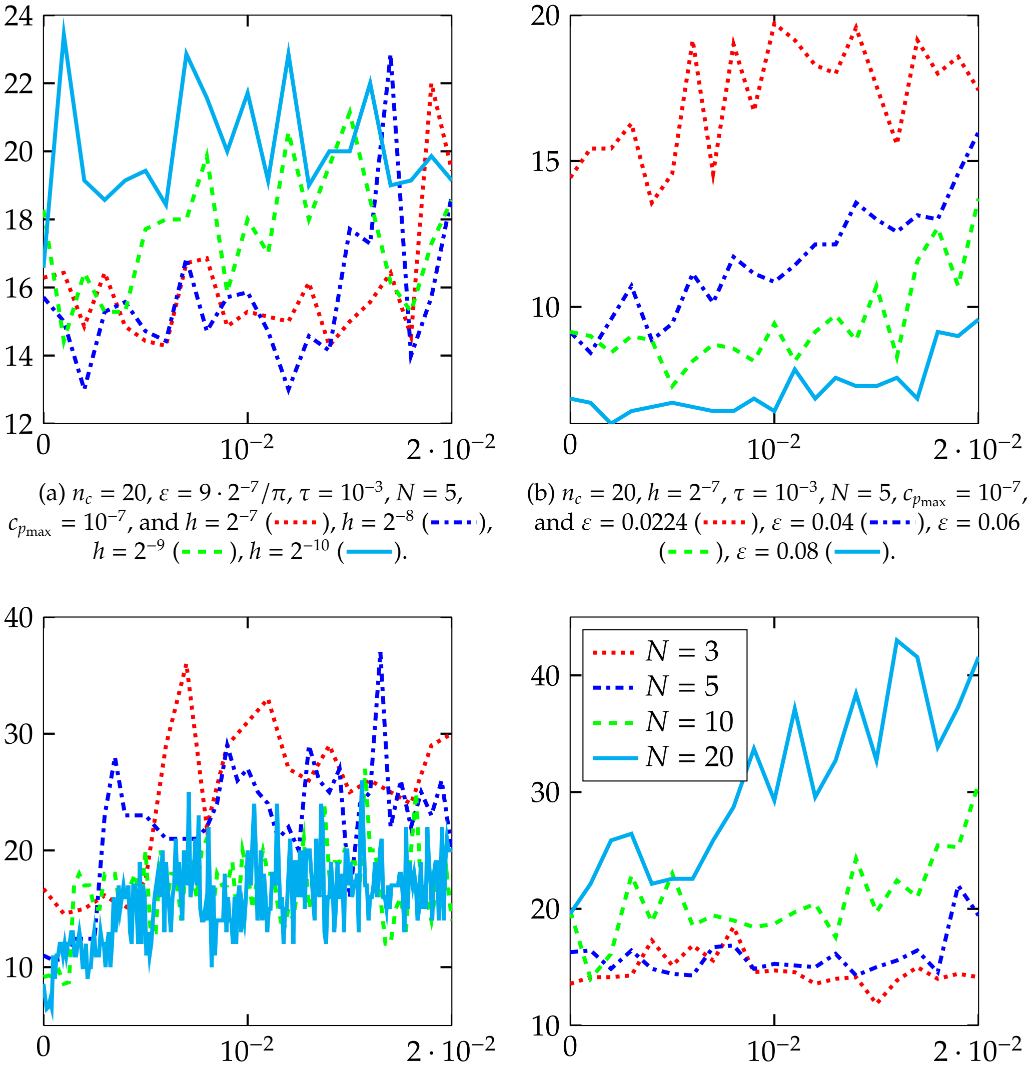
Table 4.2: Results for the solution of the smooth system (4.82) with $\boldsymbol{L} = \boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T$, the preconditioner (4.83), and the Schur complement approximation (4.100): The maximum and average number of Newton iterations, the maximum and average number of BiCG iterations, the average CPU time (in seconds) for BiCG, and the CPU time (in seconds) for the whole simulation.

ble 4.3 illustrates the maximum and average number of SSN iterations, the maximum and average number of BiCGstab iterations, the average CPU time (in seconds) for BiCGstab, and the CPU time (in seconds) for the whole simulation for each of the six subplots, respectively.

We repeat the tests above for the case $\boldsymbol{L} = \boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T$. Each subplot in Figure 4.6 and 4.5(c) demonstrates the robustness with respect to a different model parameter. Except for the case when the mesh size is decreased, our preconditioner shows promising performances. However, what is even more important in praxis, when the mesh size is refined together with the interfacial parameter (and hence with the time step size), the iteration numbers decrease significantly; see Figure 4.5(d). In this figure, we fix $n_c = 20$, $N = 5$, $c_{p_{\max}} = 10^{-7}$, and $n = 20$. Moreover, we observe a decrease of iteration numbers when ε is decreased; see Figure 4.6(b). Table 4.4 illustrates the maximum and average number of SSN iterations, the maximum and average number of BiCGstab iterations, the average CPU time (in seconds) for BiCGstab, as well as the CPU time (in seconds) for the whole simulation for each of the six subplots, respectively.

4.8.2 Mesh adaptation

Similar to Chapter 3.8.3, we can reduce the number of spatial mesh points and hence the system size m by going over to adaptive meshes. Again, we refine the interface up to a level where at least eight mesh points are across the interface. We coarsen the



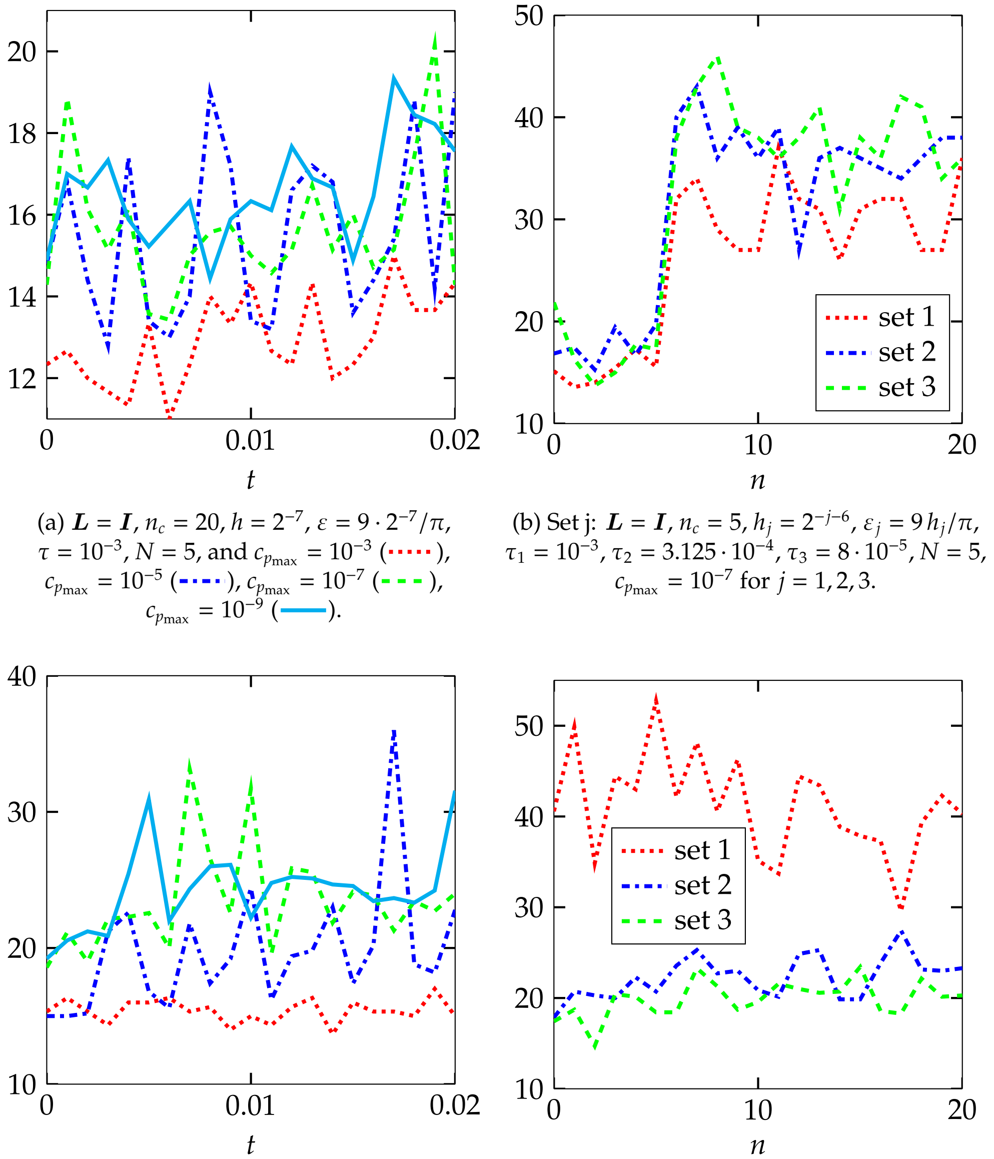
(a) $n_c = 20$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 10^{-3}$, $N = 5$, $c_{p_{\max}} = 10^{-7}$, and $h = 2^{-7}$ (⋯⋯), $h = 2^{-8}$ (-·-·), $h = 2^{-9}$ (- - -), $h = 2^{-10}$ (——).

(b) $n_c = 20$, $h = 2^{-7}$, $\tau = 10^{-3}$, $N = 5$, $c_{p_{\max}} = 10^{-7}$, and $\varepsilon = 0.0224$ (⋯⋯), $\varepsilon = 0.04$ (-·-·), $\varepsilon = 0.06$ (- - -), $\varepsilon = 0.08$ (——).

(c) $n_c = 5$, $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $N = 5$, $c_{p_{\max}} = 10^{-7}$, and $\tau = 10^{-3}$ (⋯⋯), $\tau = 5 \cdot 10^{-4}$ (-·-·), $\tau = 2.5 \cdot 10^{-4}$ (- - -), $\tau = 8 \cdot 10^{-5}$ (——).

(d) $n_c = 20$, $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 10^{-3}$, $c_{p_{\max}} = 10^{-7}$.

Figure 4.4: Results for the solution of the nonsmooth system (4.82) with $\boldsymbol{L} = \boldsymbol{I}$, the preconditioner (4.83), and the Schur complement approximation (4.109). The x-axis shows the time t and the y-axis the average number of BiCGstab iterations per SSN step.



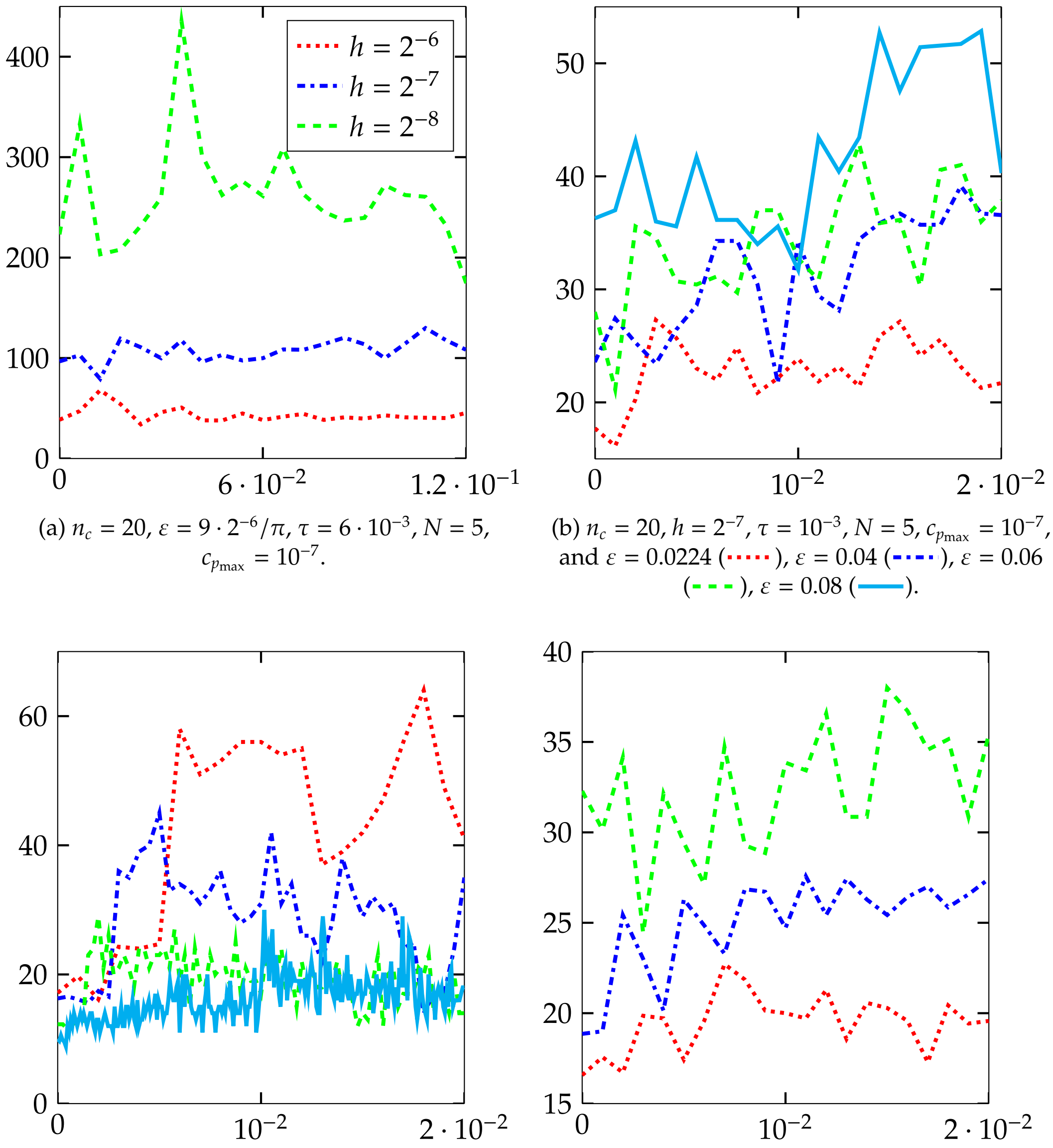
(a) $\boldsymbol{L} = \boldsymbol{I}$, $n_c = 20$, $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 10^{-3}$, $N = 5$, and $c_{p_{\max}} = 10^{-3}$ (⋯⋯), $c_{p_{\max}} = 10^{-5}$ (-·-·-), $c_{p_{\max}} = 10^{-7}$ (- - -), $c_{p_{\max}} = 10^{-9}$ (——).

(b) Set j: $\boldsymbol{L} = \boldsymbol{I}$, $n_c = 5$, $h_j = 2^{-j-6}$, $\varepsilon_j = 9\,h_j/\pi$, $\tau_1 = 10^{-3}$, $\tau_2 = 3.125 \cdot 10^{-4}$, $\tau_3 = 8 \cdot 10^{-5}$, $N = 5$, $c_{p_{\max}} = 10^{-7}$ for $j = 1, 2, 3$.

(c) $\boldsymbol{L} = \boldsymbol{I} - \frac{1}{N}\boldsymbol{1}\boldsymbol{1}^T$, $n_c = 20$, $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 10^{-3}$, $N = 5$, and $c_{p_{\max}} = 10^{-3}$ (⋯⋯), $c_{p_{\max}} = 10^{-5}$ (-·-·-), $c_{p_{\max}} = 10^{-7}$ (- - -), $c_{p_{\max}} = 10^{-9}$ (——).

(d) Set j: $\boldsymbol{L} = \boldsymbol{I} - \frac{1}{N}\boldsymbol{1}\boldsymbol{1}^T$, $n_c = 20$, $h_j = 2^{-j-5}$, $\varepsilon_j = 9\,h_j/\pi$, $\tau_1 = 6 \cdot 10^{-3}$, $\tau_2 = 10^{-3}$, $\tau_3 = 3.125 \cdot 10^{-4}$, $N = 5$, $c_{p_{\max}} = 10^{-7}$ for $j = 1, 2, 3$.

Figure 4.5: Results for the solution of the nonsmooth system (4.82) with the preconditioner (4.83) and the Schur complement approximation (4.109). The y-axis shows the average number of BiCGstab iterations per SSN step.



(a) $n_c = 20$, $\varepsilon = 9 \cdot 2^{-6}/\pi$, $\tau = 6 \cdot 10^{-3}$, $N = 5$, $c_{p_{\max}} = 10^{-7}$.

(b) $n_c = 20$, $h = 2^{-7}$, $\tau = 10^{-3}$, $N = 5$, $c_{p_{\max}} = 10^{-7}$, and $\varepsilon = 0.0224$ (⋯⋯), $\varepsilon = 0.04$ (-·-·), $\varepsilon = 0.06$ (- - -), $\varepsilon = 0.08$ (——).

(c) $n_c = 5$, $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $N = 5$, $c_{p_{\max}} = 10^{-7}$, and $\tau = 10^{-3}$ (⋯⋯), $\tau = 5 \cdot 10^{-4}$ (-·-·), $\tau = 2.5 \cdot 10^{-4}$ (- - -), $\tau = 8 \cdot 10^{-5}$ (——).

(d) $n_c = 20$, $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 10^{-3}$, $c_{p_{\max}} = 10^{-7}$, and $N = 3$ (⋯⋯), $N = 10$ (-·-·), $N = 20$ (- - -).

Figure 4.6: Results for the solution of the nonsmooth system (4.82) with $\boldsymbol{L} = \boldsymbol{I} - \frac{1}{N}\boldsymbol{1}\boldsymbol{1}^T$, the preconditioner (4.83), and the Schur complement approximation (4.109). The x-axis shows the time t and the y-axis the average number of BiCGstab iterations per SSN step.

Simulation		SSN		BiCGstab			
Figure	Plot	Max	Avg	Max	Avg	CPU (s)	CPU (s)
4.4(a)	(·····)	5	2	30	13	137	41794
	(-·-·)	5	3	31	13	573	184483
	(- - -)	6	3	35	14	2425	771196
	(——)	5	3	41	16	8763	2878150
4.4(b)	(·····)	5	2	28	14	147	42294
	(-·-·)	5	2	24	9	100	29883
	(- - -)	5	3	20	8	74	24090
	(——)	4	2	13	6	55	17810
4.4(c)	(·····)	6	3	59	17	191	35730
	(-·-·)	5	3	58	16	186	48658
	(- - -)	6	3	47	14	162	65794
	(——)	5	3	47	14	174	141358
4.4(d)	(·····)	5	2	23	12	68	18792
	(-·-·)	5	2	30	13	137	41794
	(- - -)	5	3	47	17	441	149656
	(——)	6	3	77	25	1313	549691
4.5(a)	(·····)	5	4	22	9	85	20648
	(-·-·)	6	3	28	12	108	31887
	(- - -)	6	3	26	13	136	41770
	(——)	5	2	25	14	169	53356
4.5(b)	(·····)	5	3	62	17	193	38629
	(-·-·)	6	3	66	20	980	202650
	(- - -)	7	4	84	21	4195	1066722

Table 4.3: Results for the solution of the nonsmooth system (4.82) with $\boldsymbol{L} = \boldsymbol{I}$, the preconditioner (4.83), and the Schur complement approximation (4.109): The maximum and average number of SSN iterations, the maximum and average number of BiCGstab iterations, the average CPU time (in seconds) for BiCGstab, and the CPU time (in seconds) for the whole simulation.

mesh in areas where the concentration $\mathbf{u}$ is (almost) constant. Using the nonsmooth potential, we can easily identify the interfacial and constant areas. The constant areas are the spatial points $\mathbf{x}$ that satisfy $u_i(\mathbf{x}) = 1$ for one $i \in \{1, \ldots, N\}$ and $u_j(\mathbf{x}) = 0$ for all $j \in \{1, \ldots, N\} \setminus \{i\}$. The interfacial area is formed by those spatial points $\mathbf{x}$ that satisfy $0 < u_i(\mathbf{x}) < 1$ for some $i \in \{1, \ldots, N\}$. Using the regularized potential, the interfacial area is specified in the same way. The constant areas are the spatial points $\mathbf{x}$ that satisfy $u_i(\mathbf{x}) \geq 1$ for some $i \in \{1, \ldots, N\}$. Using the smooth potential, it is not that clear where to separate the constant areas from the interfacial area as already pointed out in Chapter 3.8.3. As our simple approach is based on the knowledge about the location of constant and interfacial areas, we apply our adaptive mesh strategy only to the nonsmooth case. As in Chapter 3.8.3, for a given $\varepsilon > 0$ we use the upper bound $h_{\text{min}} \leq \frac{\varepsilon\pi}{9}$, where h_{min} is the refinement level across the interface. Since we want to avoid meshes which are too coarse, we additionally define $h_{\text{max}} := 10\, h_{\text{min}}$, where h_{max} represents the maximal mesh size. Our mesh adaptation is based on the following strategy: An element $R \in \mathcal{R}_h$ is marked for refinement if it satisfies $0 < u_i(\mathbf{x}) < 0.99999$ for some $i \in \{1, \ldots, N\}$ and if $\text{diam}(R) > 2 \cdot h_{\text{min}}$. Here, $\text{diam}(R)$ denotes the largest diagonal of R. An element $R \in \mathcal{R}_h$ is marked for coarsening if it satisfies $u_i(\mathbf{x}) \leq 0$ or $u_i(\mathbf{x}) \geq 0.99999$ for all $i \in \{1, \ldots, N\}$ and if $\text{diam}(R) \leq h_{\text{max}}/2$. Note that more sophisticated adaptation strategies based on an a-posteriori error analysis have been developed, e.g., by Hintermüller et al. [91] for the two-component case.

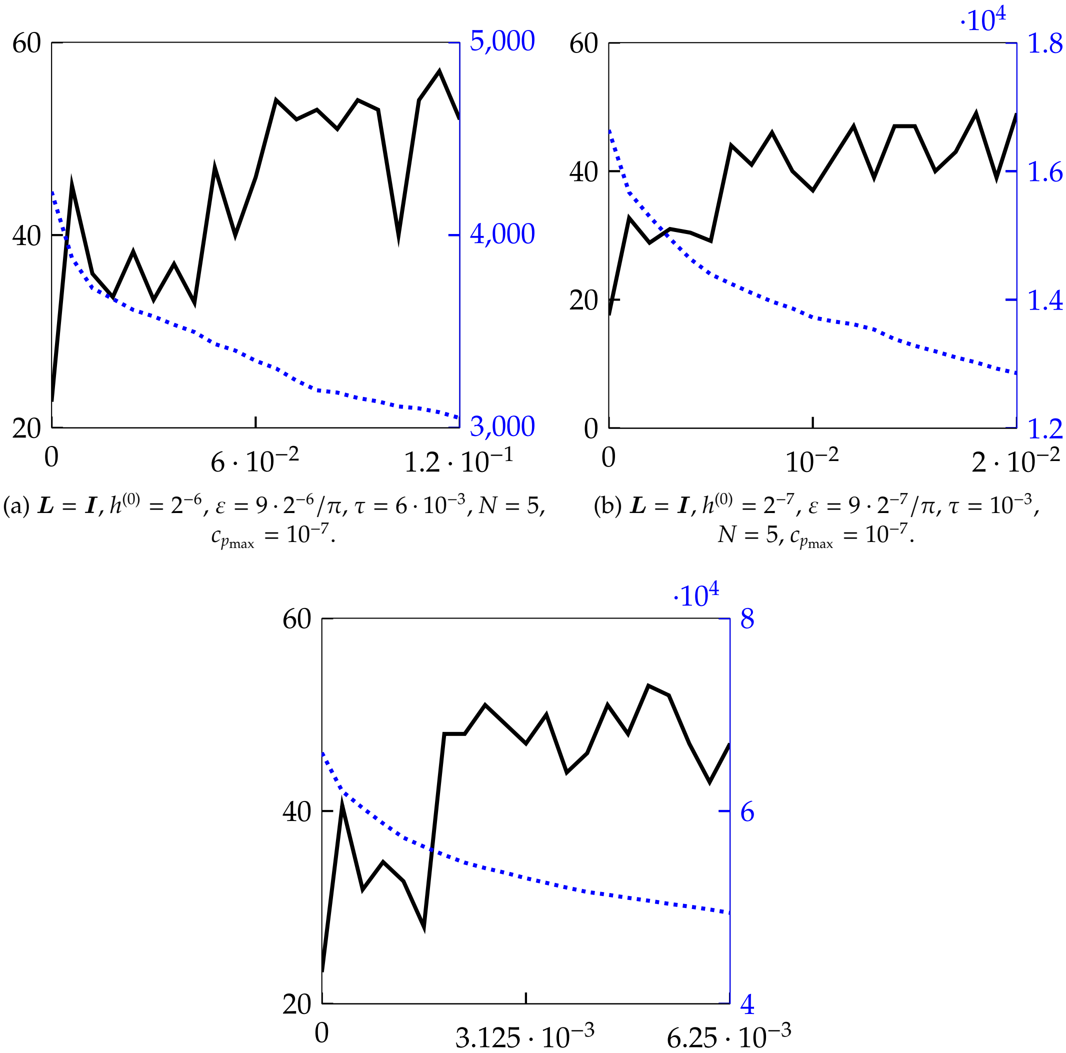
Simulation		SSN		BiCGstab			
Figure	Plot	Max	Avg	Max	Avg	CPU (s)	CPU (s)
4.6(a)	(·····)	7	2	104	35	166	41836
	(-·-·)	7	3	230	83	1614	405727
	(- - -)	7	3	1200	196	12036	2928520
4.6(b)	(·····)	5	2	39	19	336	93934
	(-·-·)	5	2	81	25	455	114646
	(- - -)	4	2	97	26	455	115610
	(——)	4	3	98	32	544	136436
4.6(c)	(·····)	7	3	138	26	489	99505
	(-·-·)	5	3	93	21	402	106175
	(- - -)	5	3	58	16	311	120697
	(——)	5	3	56	15	298	236575
4.6(d)	(·····)	5	2	33	16	168	42543
	(-·-·)	5	3	41	20	759	273560
	(- - -)	6	4	59	26	1977	868903
4.5(c)	(·····)	5	4	27	11	178	41627
	(-·-·)	5	3	64	15	245	68928
	(- - -)	5	2	54	19	344	93893
	(——)	5	2	43	21	403	118859
4.5(d)	(·····)	7	3	88	34	160	41189
	(-·-·)	5	2	47	18	324	92956
	(- - -)	6	3	48	16	1031	329997

Table 4.4: Results for the solution of the nonsmooth system (4.82) with $\boldsymbol{L} = \boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T$, the preconditioner (4.83), and the Schur complement approximation (4.109): The maximum and average number of SSN iterations, the maximum and average number of BiCGstab iterations, the average CPU time (in seconds) for BiCGstab, and the CPU time (in seconds) for the whole simulation.

In Figure 4.7, we illustrate the performance of our preconditioner (4.83) with the Schur complement approximation (4.109) for the solution of the nonsmooth system (4.82) on adaptive meshes. We consider the case $\boldsymbol{L} = \boldsymbol{I}$ and test three different settings with varying values of $h^{(0)}, \varepsilon, \tau$, where $h^{(0)}$ denotes the mesh size of the initial uniform mesh. The x-axis shows the time t, the left y-axis shows the average number of BiCGstab iterations per SSN step, and the right y-axis the number of degrees of freedom, respectively. We can see that the iteration numbers stay constantly low. Moreover, the coarsening process acts in the pure phases and reduces the number of degrees of freedoms. Table 4.5 illustrates the maximum and average number of SSN iterations, the maximum and average number of BiCGstab iterations, the average CPU time (in seconds) for BiCGstab, and the CPU time (in seconds) for the whole simulation for each of the three subplots, respectively. The final phase variables for each simulation is illustrated in Figure 4.8 together with the spatial mesh.

4.8.3 Long-time evolution

In the following, we consider the long-time evolution of the smooth and nonsmooth vector-valued Cahn–Hilliard model; see Figure 4.9. In the smooth case, we use $h = 2^{-7}$, $\varepsilon = 9h/(4\sqrt{2} \cdot \operatorname{atanh}(0.9))$, $\tau = 10^{-5}$, $T = 10^{-1}$. In the nonsmooth case, we use the setting $h = 2^{-7}$, $\varepsilon = 9h/\pi$, $\tau = 10^{-3}$, $c_{p_{\max}} = 10^{-7}$, $T = 2$. In both simulations,



(a) $\boldsymbol{L} = \boldsymbol{I}$, $h^{(0)} = 2^{-6}$, $\varepsilon = 9 \cdot 2^{-6}/\pi$, $\tau = 6 \cdot 10^{-3}$, $N = 5$, $c_{p_{\max}} = 10^{-7}$.

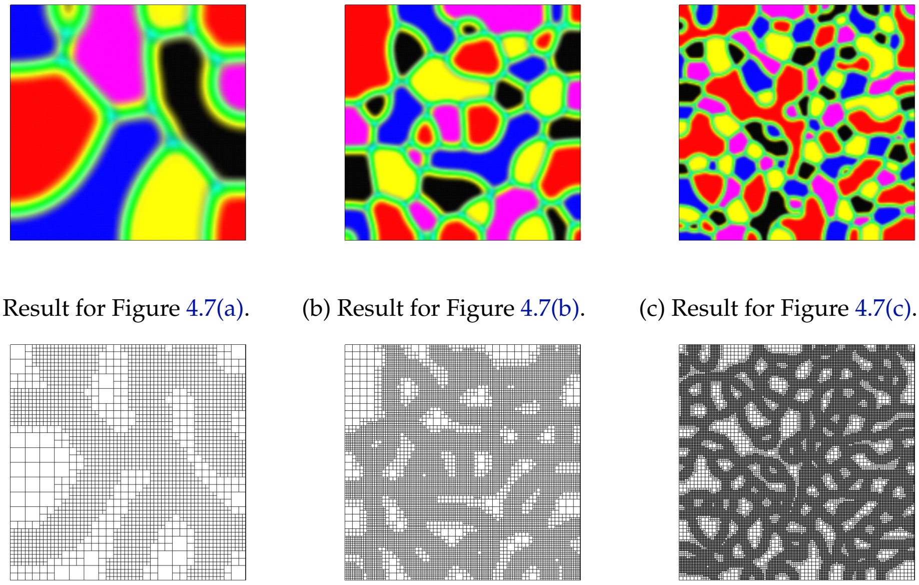
(b) $\boldsymbol{L} = \boldsymbol{I}$, $h^{(0)} = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 10^{-3}$, $N = 5$, $c_{p_{\max}} = 10^{-7}$.

(c) $\boldsymbol{L} = \boldsymbol{I}$, $h^{(0)} = 2^{-8}$, $\varepsilon = 9 \cdot 2^{-8}/\pi$, $\tau = 3.125 \cdot 10^{-4}$, $N = 5$, $c_{p_{\max}} = 10^{-7}$.

Figure 4.7: Results for the solution of the nonsmooth system (4.82) with the preconditioner (4.83) and the Schur complement approximation (4.109) using adaptive meshes. The x-axis shows the time t, the left y-axis the average number of BiCGstab iterations per SSN step and the right y-axis the number of degrees of freedom.

	SSN		BiCG			
Figure	Max	Avg	Max	Avg	CPU (s)	CPU (s)
3.9(a)	7	4	104	34	93	19097
3.9(b)	6	4	91	29	305	67620
3.9(c)	6	4	100	32	1375	328189

Table 4.5: Results for the solution of the nonsmooth system (4.82) with the preconditioner (4.83) and the Schur complement approximation (4.109) using adaptive meshes: The maximum and average number of SSN iterations, the maximum and average number of BiCGstab iterations, the average CPU time (in seconds) for BiCGstab, and the CPU time (in seconds) for the whole simulation.



(a) Result for Figure 4.7(a).

(b) Result for Figure 4.7(b).

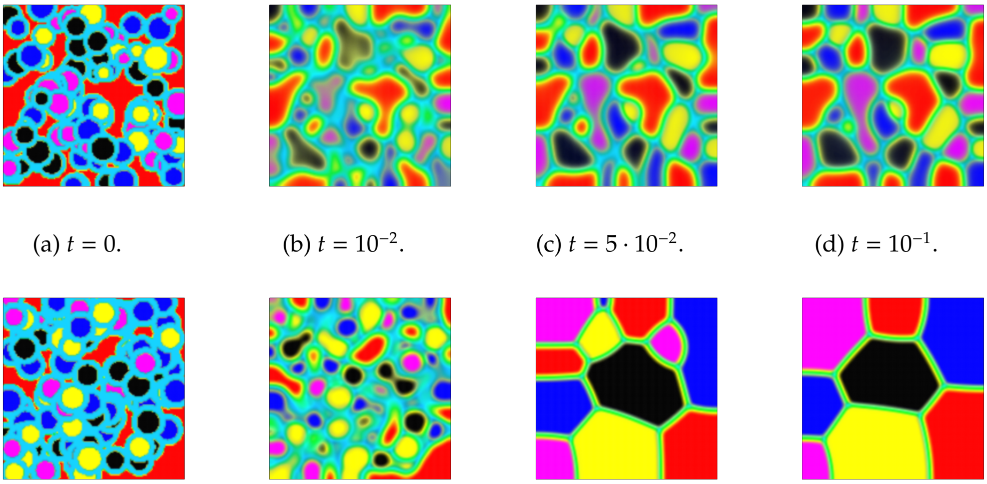
(c) Result for Figure 4.7(c).

Figure 4.8: Computation with adaptive meshes: Final phase variables with the corresponding spatial mesh for the three simulations in Figure 4.7.

we set $N = 5$ and $\boldsymbol{L} = \boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T$.

	Newton/SSN		BiCG/BiCGstab			
Figure	Max	Avg	Max	Avg	CPU (s)	CPU (s)
4.9(a)–4.9(d)	2	1	12	9	86	869137
4.9(e)–4.9(h)	6	2	162	39	822	3756234

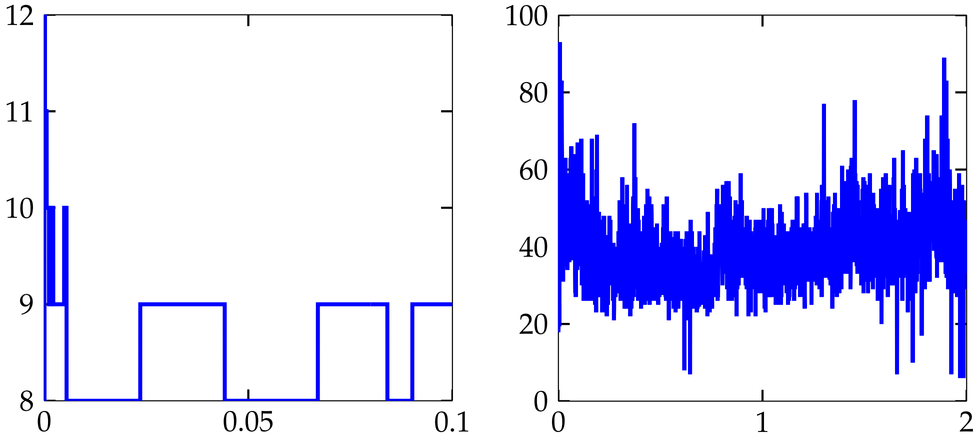
Table 4.6: Results for the long-time evolution: The maximum and average number of Newton/SSN iterations, the maximum and average number of BiCG/BiCGstab iterations, as well as the average CPU time (in seconds) for BiCG/BiCGstab and the CPU time (in seconds) for the whole simulation for the smooth and nonsmooth Cahn–Hilliard model, respectively.



(a) $t = 0$. (b) $t = 10^{-2}$. (c) $t = 5 \cdot 10^{-2}$. (d) $t = 10^{-1}$.

(e) $t = 0$. (f) $t = 10^{-3}$. (g) $t = 1$. (h) $t = 2$.

Figure 4.9: Long-time evolution using the smooth (upper row) and nonsmooth (lower row) vector-valued Cahn–Hilliard model.



(a) Results for Figure 4.9(a)–4.9(d). $h = 2^{-7}$, $\varepsilon = 9h/(4\sqrt{2} \cdot \text{atanh}(0.9))$, $\tau = 10^{-5}$, $N = 5$.

(b) Results for Figure 4.9(e)–4.9(h). $h = 2^{-7}$, $\varepsilon = 9h/\pi$, $\tau = 10^{-3}$, $N = 5$, $c_{p_{\max}} = 10^{-7}$.

Figure 4.10: Results for the long-time evolution using the smooth (left) and nonsmooth (right) vector-valued Cahn–Hilliard model. The x-axis shows the time t and the y-axis displays the average number of BiCG (left) and BiCGstab (right) iterations per Newton/SSN step.

In Figure 4.10, we illustrate the performance of our preconditioners for the solution of the smooth and nonsmooth vector-valued Cahn–Hilliard system. The x-axis shows the time t and the y-axis displays the average number of BiCG (in the smooth case) and BiCGstab (in the nonsmooth case) iterations per Newton/SSN step. Table 4.6 illustrates the maximum and average number of Newton/SSN iterations, the maximum and average number of BiCG/BiCGstab iterations, the average CPU time (in seconds) for BiCG/BiCGstab, and the CPU time (in seconds) for the whole simulation for the smooth and nonsmooth model, respectively. Table 4.7 illustrates the minimum and maximum phase values at some time steps. We observe that the concentrations may become less than zero for smooth potentials. However, no blow ups are reported.

Note that one should not compare the above results in terms of smooth versus nonsmooth. The evolution with smooth and nonsmooth potentials is very different and distinct parameters are used. Moreover, both types of potentials are used in many applications. In some of them, like the deep-quench limit, the nonsmooth potential must be used. In other applications, smooth potentials are preferred and produce satisfactory results. Therefore, the development of efficient solvers is of great interest in both cases.

		t			
value	model	$4 \cdot 10^{-2}$	$6 \cdot 10^{-2}$	$8 \cdot 10^{-2}$	10^{-1}
min	smooth	-0.0265432	-0.0225901	-0.0237592	-0.0277528
	nonsmooth	$-1.19975 \cdot 10^{-7}$	$-1.21102 \cdot 10^{-7}$	$-1.19531 \cdot 10^{-7}$	$-1.178 \cdot 10^{-7}$
max	smooth	0.970885	0.978767	0.989315	0.995248
	nonsmooth	1.00005	1.00004	1.00004	1.00004

Table 4.7: Minimum and maximum phase values during the simulation with the smooth and nonsmooth vector-valued Cahn–Hilliard model.

4.8.4 Three-dimensional example

Next, we consider the three-dimensional domain $\Omega = [0,1]^3$ and simulate the phase separation and coarsening process. Figure 4.11 shows the evolution for this example using the nonsmooth vector-valued Cahn–Hilliard model with $h^{(0)} = 2^{-6}$, $\varepsilon = 9h/\pi$, $\tau = 10^{-3}$, $N = 5$, $c_{p_{\max}} = 10^{-7}$, $\boldsymbol{L} = \boldsymbol{I}$, $T = 10^{-2}$, where $h^{(0)}$ denotes the mesh size of the initial uniform mesh. In Figure 4.12, we illustrate the performance of our preconditioner. The x-axis shows the time t, the left y-axis displays the average number of BiCGstab iterations per SSN step, and the right y-axis illustrates the number of degrees of freedom. The maximum and average number of BiCGstab iterations for the simulation are 107 and 29.

4.9 Existing solvers

In this section, we briefly discuss existing solution methods for the smooth and nonsmooth vector-valued Cahn–Hilliard equation.

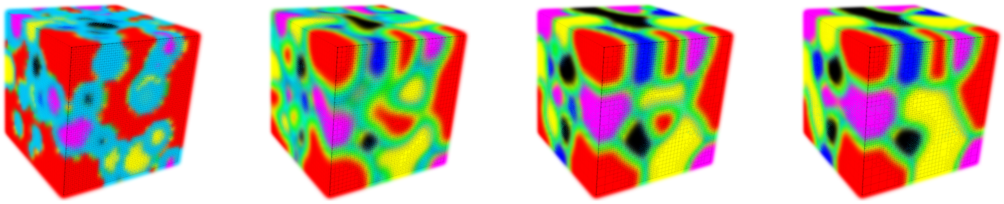

(a) $t = 0$. (b) $t = 10^{-3}$. (c) $t = 5 \cdot 10^{-3}$. (d) $t = 10^{-2}$.

Figure 4.11: A three-dimensional simulation of the phase separation and coarsening process of a five-component mixture.

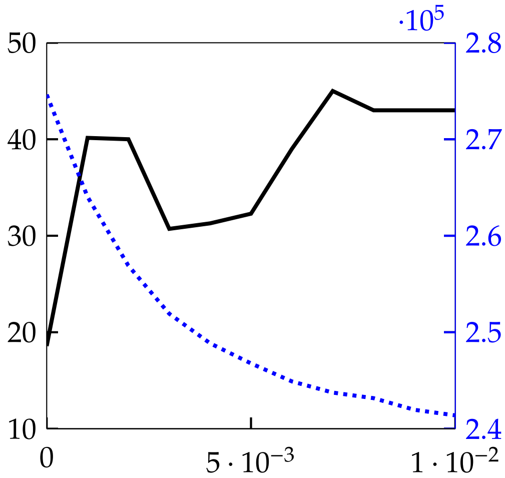


Figure 4.12: Results for the three-dimensional simulation of the phase separation and coarsening process of a five-component mixture: The x-axis shows the time t, the left y-axis displays the average number of BiCGstab iterations per SSN step, and the right y-axis illustrates the number of degrees of freedom.

A nonlinear multigrid method for the smooth vector-valued Cahn–Hilliard equation is proposed by Lee et al. [118, 119]. In [118], the authors consider the case $\boldsymbol{L} = \boldsymbol{I}$. Moreover, they present a practically unconditionally gradient stable scheme, which is based on a nonlinear splitting method. This allows a decoupling of the N-component Cahn–Hilliard system into $N - 1$ scalar Cahn–Hilliard equations. The efficiency of this approach is shown by means of the average CPU time, whose convergence rate is linear with respect to the number of phases. In [119], the authors consider the case of a concentration dependent mobility matrix. Moreover, they apply Crank-Nicolson's method for the discretization in time. The authors develop a Full Approximation Storage multigrid method with a pointwise Gauß-Seidel relaxation scheme as a smoother. The nonlinearity is treated using one Newton step. The authors demonstrated the second-order accuracy of the numerical scheme.

Gräser et al. [85] propose globally convergent nonsmooth Schur–Newton methods (NSNMG) for the solution of discrete multi-component Cahn–Hilliard systems. They consider logarithmic as well as obstacle potentials. NSNMG can be formulated in primal-dual form and results in a preconditioned Uzawa method. Each step consists first of the update of the primal variable, which includes the direct work with the inverse $(\boldsymbol{A} + \partial\varphi)^{-1}$. Here, $\boldsymbol{A}$ is a symmetric positive definite matrix and $\partial\varphi$ is the subdifferential of the nonsmooth part of the potential, which includes the indicator function $\sum_{i=1}^{N} \chi_{[0,\infty)}(u_i)$. The second step of NSNMG is to compute the dual variable, which can be done by solving a truncated linear saddle-point problem and updating the step size for the Uzawa method. The authors solve the linear systems by preconditioned GMRES with a restart after 50 steps. They numerically investigated the local mesh independence of NSNMG as well as a robust convergence speed of NSNMG and of the truncated nonsmooth Newton method for different numbers of phases.

An important point for the future is a comparison with our approach.

4.10 Conclusions and future research perspectives

In this chapter, we have investigated the numerical solution of the multi-component Cahn–Hilliard model. We have considered smooth and nonsmooth potentials with a focus on the latter. Motivated by the previous chapter, we have used a fully implicit scheme for the discretization in time. Regarding the smooth setting, we have extended the proof of the energy stability and uniqueness of the solution of the time-discrete scheme from the two-component to the multi-component case. Concerning the nonsmooth framework, we have interpreted the time-discrete problem as the first-order optimality system of an optimization problem for which we have derived existence and uniqueness conditions. In particular, we extended the analysis from the two-component to the multi-component case. Motivated by Chapter 3, we have applied an SSN method combined with a Moreau–Yosida regularization technique for handling the pointwise constraints. For the discretization in space, we have used classical FEM for both systems, the smooth and regularized nonsmooth one. At the heart of our method lies the solution of large and sparse systems of linear equations of saddle point form. We have introduced and studied block-triangular preconditioners using an efficient and cheap Schur complement approximation. For these approximations, we have used multilevel techniques, algebraic multigrid in our case. For the smooth systems, we have derived optimal preconditioners which are proven to be robust with respect to crucial model parameters. For the nonsmooth systems, extensive numerical experiments show an outstanding behavior of our developed preconditioners. Additionally, we have implemented a simple adaptive spatial mesh refinement approach, which reduces the number of degrees of freedoms. Together with our preconditioners, this allows us to perform three-dimensional experiments in an efficient way.

As pointed out during this chapter, there are several aspects for further research. First of all, we have not extended the whole analysis for the nonsmooth case from the two-component to the multi-component case. Second, we have improved the preconditioners proposed in [33] for the nonsmooth case when the mobility $\boldsymbol{L} = \boldsymbol{I}$ is used. The case $L = \boldsymbol{I} - \frac{1}{N}\boldsymbol{1}\boldsymbol{1}^T$ still seems to be improvable with some fine tuning. Third, we have restricted our attention to constant mobility matrices, likewise in the previous chapter. However, in many applications, concentration dependent mobilities are required. This is for example the case if the mobility in the interface is larger than in the pure phases. Fourth, the study of preconditioners for the linear systems arising from other time discretization schemes, e.g., a semi-implicit scheme, might be of high interest for special applications. Finally, in the previous chapter, we have employed our preconditioner to a coupled Cahn–Hilliard/Navier–Stokes system, which governs the hydrodynamics of two-phase flows. An interesting field of future research is the application of our preconditioners to the numerical solution of multi-component flows.

Chapter 5

Modified Cahn–Hilliard Equations

5.1 Introduction

Image inpainting is the art of modifying parts of an image such that the resulting changes are not easily detectable by an ordinary observer. Applications include the restoration of damaged paintings and photographs [5], the replacement of selected objects, or the reduction of artifacts in medical images [87].

Bertozzi et al. [19, 18] introduced the fourth-order Cahn–Hilliard inpainting approach for binary, i.e., black-and-white, images. This model is based on the scalar smooth Cahn–Hilliard equation discussed in Chapter 3. In [31], we extended the inpainting approach to the scalar nonsmooth Cahn–Hilliard equation discussed in Chapter 3. Further, Bertozzi et al's binary Cahn–Hilliard inpainting model has been recently generalized to gray value images [32, 48]. This model is based on the vector-valued smooth Cahn–Hilliard equation discussed in Chapter 4. In this chapter, we will focus on this gray value Cahn–Hilliard inpainting model. As in the previous two chapters, we will discuss the two types of potential functions, smooth and nonsmooth. Note that the presented study applies to the binary case as well.

Let f be the given gray value image, which is defined on the image domain $\Omega \subset \mathbb{R}^d$ with $d \in \{2, 3\}$. We denote by N the number of gray values which form the image f. These N gray values are collected in the vector $\mathbf{g} = [g_1, \ldots, g_N]^T \in \mathbb{R}^N$. Note that $2 \leq N \leq 256$. The parts of f that are going to be modified are denoted by the inpainting domain $D \subset \Omega$. The target is to reconstruct the image f in this region D in an undetectable way. We denote the reconstructed image by f_r. Let $T > 0$ be a fixed time. We introduce a vector-valued phase variable $\mathbf{u} = [u_1, \ldots, u_N]^T : \Omega \times (0, T) \to \mathbb{R}^N$. Here, u_i describes the concentration of gray value g_i for $i = 1, \ldots, N$. If $u_i(\mathbf{x}, t) \approx 1$, then only gray value g_i (the pure gray value g_i) is present at point $\mathbf{x}$ at time t. The case $u_i(\mathbf{x}, t) \approx 0$ means gray value g_i is absent at point $\mathbf{x}$ at time t. Values of u_i between 0 and 1 represent mixed regions. In particular, these regions include the interfacial area. Here, the interface is a small boundary layer that separates the pure gray values g_i, $i = 1, \ldots, N$, from each other. As in the previous two chapters, it acts as a diffuse phase transition, and we can control its width via the model parameter $\varepsilon > 0$. Basically, we can imagine each component u_i as a “binary“ image that evolves in time. More precisely, u_i represents the evolution of the gray value g_i. We initialize u_i with $u_i(\mathbf{x}, 0) = f_i(\mathbf{x})$. Here, $\mathbf{f} = [f_1, \dots, f_N]^T \in \mathcal{G}^N$, where $\mathcal{G}^N$ is given in (5.2), is the vector of given gray value distributions from the given image f. That means, $f_i(\mathbf{x}) \in \{0, 1\}$ describes the absence or presence of gray value g_i in f at point $\mathbf{x}$ for $i = 1, \dots, N$. Above, we have put the term binary in quotation marks because u_i is actually binary only at the initial state $t = 0$. For $t > 0$, the thin interfacial areas are present, which represent mixed regions. The evolution of the reconstructed image f_r is obtained from the components u_i via

$$f_r = \sum_{i=1}^{N} g_i u_i.$$

The final reconstructed image f_r of f is $f_r(\mathbf{x}, T)$. As in Chapter 4, it holds

$$\sum_{i=1}^{N} u_i = 1 \tag{5.1}$$

and $u_i \geq 0$ for $i = 1, \dots, N$, so that admissible states belong to the Gibbs simplex

$$\mathcal{G}^N := \left\{ \mathbf{v} = [v_1, \dots, v_N]^T \in \mathbb{R}^N \colon \sum_{i=1}^{N} v_i = 1,\ v_i \geq 0 \text{ for } i = 1, \dots, N \right\}. \tag{5.2}$$

The Cahn–Hilliard inpainting model is based on the Ginzburg–Landau energy $\mathcal{E}$, which is used in Chapter 4. Here, it is given as

$$\mathcal{E}(\mathbf{u}) = \int_\Omega \frac{\varepsilon}{2} \sum_{i=1}^{N} |\nabla u_i|^2 + \frac{1}{\varepsilon} \psi(\mathbf{u}) \, \mathrm{d}\mathbf{x}. \tag{5.3}$$

The parameter $\varepsilon > 0$ is proportional to the thickness of the interfacial region as mentioned above. The first part of (5.3) is large whenever u_i changes rapidly for some $i \in \{1, \dots, N\}$. Hence, its minimization gives rise to the interfacial area. The potential function $\psi \colon \mathbb{R}^N \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ in (5.3) gives rise to phase separation. It has N distinct minima, one for each pure gray value g_i. We consider the same two potential functions of polynomial and obstacle type as in the last chapter. For the sake of clarity, we repeat them again: The first one is the smooth multi-well potential given as

$$\psi_{\text{pol}}(\mathbf{u}) = \frac{1}{4} \sum_{i=1}^{N} u_i^2 (1 - u_i)^2. \tag{5.4}$$

The second potential is the nonsmooth multi-obstacle potential given as

$$\psi_{\text{obs}}(\mathbf{u}) = \begin{cases} \psi_0(\mathbf{u}) = -\frac{1}{2} \mathbf{u} \cdot \boldsymbol{T} \mathbf{u} & \mathbf{u} \in \mathcal{G}^N, \\ \infty & \text{otherwise.} \end{cases} \tag{5.5}$$

Again, $\boldsymbol{T} \in \mathbb{R}^{N \times N}$ is a symmetric matrix, which contains constant interaction parameters $[T]_{ij}$. From physical considerations, $\boldsymbol{T}$ must have at least one positive eigenvalue. During the rest of this chapter, we denote by $\lambda_{\max}(\boldsymbol{T})$ the largest positive eigenvalue of $\boldsymbol{T}$. A typical choice is $\boldsymbol{T} = \boldsymbol{I} - \mathbf{1}\mathbf{1}^T$ with $\mathbf{1} = [1, \ldots, 1]^T \in \mathbb{R}^N$ and the identity matrix $\boldsymbol{I} \in \mathbb{R}^{N \times N}$, which means that the interaction between all different gray values is equal and no self-interaction occurs. For logarithmic potentials, we refer to, e.g., [49].

As we have seen in the previous chapter, the vector-valued Cahn–Hilliard equation is derived by minimizing the Ginzburg–Landau energy (5.3) subject to the mass conservation

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_\Omega u_i \, \mathrm{d}\mathbf{x} = 0, \quad i = 1, \ldots, N. \tag{5.6}$$

In particular, it can be derived as the H^{-1}-gradient flow of the Ginzburg-Landau energy (5.3) under the constraint (5.1).

Image inpainting is based on the knowledge about the given image f as well as the inpainting domain D. Therefore, the actual starting point is the fidelity functional

$$\mathcal{F}(\mathbf{u}) = \int_\Omega \frac{\omega}{2} \sum_{i=1}^{N} (f_i - u_i)^2 \, \mathrm{d}\mathbf{x}, \tag{5.7}$$

where

$$\omega = \omega(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \in D, \\ \omega_0 & \text{if } \mathbf{x} \in \Omega \setminus D, \end{cases} \tag{5.8}$$

is the fidelity parameter. Its minimization keeps the reconstructed image $f_r = \sum_{i=1}^N g_i u_i$ close to the given image f in the undamaged parts $\Omega \setminus D$. In order to obtain Cahn–Hilliard inpainting, this fidelity functional is regularized by the Ginzburg–Landau energy $\mathcal{E}$ in (5.3). As Bertozzi et al's [19, 18] black-and-white Cahn–Hilliard inpainting model, our proposed gray value Cahn–Hilliard inpainting model arises as a superposition of two gradient flows: An H^{-1}-gradient flow under the constraint (5.1) for the Cahn–Hilliard part and an L^2-gradient flow for the fidelity term. We will derive the inpainting model in Section 5.2.

As we will show in the course of this chapter, the solution of linear systems $\boldsymbol{\mathcal{A}}\boldsymbol{z} = \boldsymbol{b}$ with a large and sparse matrix $\boldsymbol{\mathcal{A}}$ is at the heart of our method. They have the following saddle point structure

$$\boldsymbol{\mathcal{A}} = \begin{bmatrix} -\boldsymbol{A} & \boldsymbol{I} \otimes \boldsymbol{M} \\ \boldsymbol{I} \otimes \boldsymbol{M} & \boldsymbol{I} \otimes \boldsymbol{K} \end{bmatrix}$$

with $\boldsymbol{I} \in \mathbb{R}^{N \times N}$ being the identity matrix, $\boldsymbol{M} \in \mathbb{R}^{m \times m}$ being symmetric positive definite, and $\boldsymbol{K} \in \mathbb{R}^{m \times m}$ being symmetric positive semidefinite. In the smooth setting, $\boldsymbol{A} \in \mathbb{R}^{Nm \times Nm}$ is a symmetric, block diagonal matrix. However, in the nonsmooth case, $\boldsymbol{A} \in \mathbb{R}^{Nm \times Nm}$ is nonsymmetric and possibly indefinite. The crucial parameters contained in $\boldsymbol{\mathcal{A}}$ are the spatial mesh size h, the time step size τ, the interface parameter ε, the number of phases N, the Moreau–Yosida regularization parameter c, as well as two parameters that arise in our time discretization approach. We develop efficient preconditioners $\boldsymbol{\mathcal{P}}$ for the solution of the linear systems above. This is based on effective Schur complement approximations as well as (algebraic) multigrid solvers developed for elliptic systems [68, 136, 134]. For the smooth systems, we derive the conditions for optimal preconditioners. For the nonsmooth systems, extensive numerical experiments show a promising behavior of our developed preconditioners.

The structure of the chapter is as follows. The modified Cahn–Hilliard model is derived in Section 5.2. We first consider the smooth multi-well potential (5.4), which leads to a system of fourth-order PDEs. Then, we study the nonsmooth multi-obstacle potential (5.5), which yields a system of variational inequalities. An important difference to the previous two chapters is that the modified Cahn–Hillard equation as a whole is not given by a gradient flow. Especially, the model arises as a superposition of two gradient flows. The convexity splitting method can deal with such systems and, under the right conditions, yields an unconditional gradient stable time-discrete scheme. The smooth and nonsmooth formulations are discretized in time in Section 5.3 using the convexity splitting approach. In the smooth setting, we extend the proof of consistency, unconditional stability, and convergence of the time-discrete scheme from the two-component to the multi-component case. Concerning the nonsmooth framework, following the previous two chapters, we apply an SSN method combined with the Moreau–Yosida regularization technique. Section 5.4 shortly introduces the SSN method for solving the regularized subproblems. We derive the linear systems arising from the discretization using finite elements in Section 5.5. In Section 5.6, we analyze the linear systems and propose preconditioning strategies for the saddle point problems. Section 5.7 illustrates the efficiency of our preconditioners for both problem setups. In Section 5.8, we summarize our findings and discuss possible future directions.

5.2 Derivation

As mentioned in the previous section, the gray value Cahn–Hilliard inpainting model is given as a superposition of the H^{-1}-gradient flow for $\mathcal{E}$ in (5.3) under the constraint (5.1) and the L^2-gradient flow for $\mathcal{F}$ in (5.7), i.e.,

$$\partial_t \mathbf{u} = -\mathrm{grad}_{H^{-1}}^{(5.1)} \mathcal{E}(\mathbf{u}) - \mathrm{grad}_{L^2} \mathcal{F}(\mathbf{u}).$$

We have already derived the first part, the multi-component Cahn–Hilliard equation, in Chapter 4.2. Hence, we will not repeat the details again. First of all, the smooth multi-well potential (5.4) setting is used. Then, we go over to the nonsmooth multi-obstacle potential (5.5) setting. As in the two chapters before, we handle this case with the Moreau–Yosida regularization technique.

Remark 5.1. *A different approach for image inpainting using the Cahn–Hilliard model was studied in [104], see also [21, 24]. There, image inpainting is modeled as minimization problem. In particular, the authors extended the projected gradient method to the arising type of problems.*

5.2.1 Smooth systems

In the following, we focus on the smooth multi-well potential (5.4). In Chapter 4.2.1, we have derived the following vector-valued Cahn–Hilliard equation:

$$\partial_t u_i = (\boldsymbol{L}\Delta \mathbf{w})_i, \tag{5.9}$$

$$w_i = -\varepsilon \Delta u_i + \frac{1}{\varepsilon}\psi'_{\mathrm{pol}}(u_i) - \frac{1}{\varepsilon N}\sum_{j=1}^{N} \psi'_{\mathrm{pol}}(u_j), \tag{5.10}$$

$$\nabla u_i \cdot \mathbf{n} = (\boldsymbol{L}\nabla \mathbf{w})_i \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega, \tag{5.11}$$

for $i = 1,\dots,N$. Here, $\psi'_{\mathrm{pol}}(v_i) = \frac{\partial \psi_{\mathrm{pol}}}{\partial u_i}(\mathbf{v}) = v_i^3 - \frac{3}{2}v_i^2 + \frac{1}{2}v_i$, and $\boldsymbol{L} \in \mathbb{R}^{N\times N}$ is the mobility matrix with constant entries. We have derived that $\boldsymbol{L}$ has to be symmetric positive semidefinite with $\boldsymbol{L}\mathbf{1} = \mathbf{0}$, where $\mathbf{1} = [1,\dots,1]^T \in \mathbb{R}^N$. In Remark 4.1 as well as in the numerical examples of Chapter 4, we have seen that it is nevertheless possible to work with $\boldsymbol{L} = \boldsymbol{I}$. In order to simplify the inpainting model, we will focus on the case $\boldsymbol{L} = \boldsymbol{I}$ in this chapter. Now, if we superpose the system (5.9)–(5.11) with the L^2-gradient flow for $\mathcal{F}$ in (5.7), we obtain the smooth gray value Cahn–Hilliard inpainting model:

$$\partial_t u_i = \Delta w_i + \omega(f_i - u_i), \tag{5.12}$$

$$w_i = -\varepsilon\Delta u_i + \frac{1}{\varepsilon}\psi'_{\mathrm{pol}}(u_i) - \frac{1}{\varepsilon N}\sum_{j=1}^{N}\psi'_{\mathrm{pol}}(u_j), \tag{5.13}$$

$$\nabla u_i \cdot \mathbf{n} = \nabla w_i \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega, \tag{5.14}$$

for $i = 1,\dots,N$. We call this system the *smooth vector-valued modified Cahn–Hilliard equation*.

Remark 5.2. *Recently, Cherfils et al. [48] considered and analyzed a similar model. More precisely, they proved the existence and uniqueness of solutions as well as the existence of the global attractor. Moreover, they constructed finite-dimensional attractors and proved that that their model is algebraically consistent with the two-phase model. In contrast, our work focuses on the numerical analysis of an unconditionally time stepping scheme as well as its efficient numerical solution via FEM.*

5.2.2 Nonsmooth systems

In the last section, we focused on the smooth potential ψ_{pol}. We could easily calculate the derivative of the smooth potential with respect to $\mathbf{u}$. Now, we turn to the nonsmooth potential ψ_{obs} given in (5.5). In Chapter 4.2.2, we have seen that the Cahn–Hilliard model results in a system of variational inequalities. We have circumvented this difficulty by the Moreau–Yosida regularization technique in Chapter 4.4. We apply this technique here as well. We regularize the multi-obstacle potential ψ_{obs} in (5.5) by

$$\psi_c(\mathbf{u}) = \psi_0(\mathbf{u}) + \frac{\varepsilon}{2c}\sum_{i=1}^{N}\min(0,u_i)^2 = -\frac{1}{2}\mathbf{u}\cdot\boldsymbol{T}\mathbf{u} + \frac{\varepsilon}{2c}\sum_{i=1}^{N}\min(0,u_i)^2.$$

As before, $0 < c \ll 1$ denotes the associated regularization or penalty parameter. The smaller c is the larger is the penalization for the violation of the condition $\mathbf{u} \geq \mathbf{0}$. Hence, the limit $c \to 0$ represents the original multi-obstacle potential ψ_{obs} in (5.5). Instead of the energy functional $\mathcal{E}$ in (5.3), we consider

$$\begin{aligned}\mathcal{E}(\mathbf{u}) &= \int_\Omega \frac{\varepsilon}{2}\sum_{i=1}^{N}|\nabla u_i|^2 + \frac{1}{\varepsilon}\psi_0(\mathbf{u}) + \frac{1}{2c}\sum_{i=1}^{N}|\min(0,u_i)|^2\,\mathrm{d}\mathbf{x}\\ &= \int_\Omega \frac{\varepsilon}{2}\sum_{i=1}^{N}|\nabla u_i|^2 - \frac{1}{2\varepsilon}\mathbf{u}^T\boldsymbol{T}\mathbf{u} + \frac{1}{2c}\sum_{i=1}^{N}|\min(0,u_i)|^2\,\mathrm{d}\mathbf{x}. \end{aligned}\tag{5.15}$$

In Chapter 4.4, we have derived the corresponding system of Cahn–Hilliard equations. As already mentioned, we focus on the case $\boldsymbol{L} = \boldsymbol{I}$, and the strong formulation reads as:

$$\partial_t u_i = \Delta w_i, \tag{5.16}$$

$$w_i = -\varepsilon \Delta u_i - \frac{1}{\varepsilon}(\boldsymbol{T}\mathbf{u})_i + \frac{1}{c}\min(0, u_i) + \frac{1}{N}\sum_{j=1}^{N}\left[-\frac{1}{c}\min(0, u_j) + \frac{1}{\varepsilon}(\boldsymbol{T}\mathbf{u})_j\right], \tag{5.17}$$

$$\nabla u_i \cdot \mathbf{n} = \nabla w_i \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega, \tag{5.18}$$

for $i = 1, \ldots, N$. For the sake of clarity, we omit the subindex c in the solution $(\mathbf{u}_c, \mathbf{w}_c)$ of (5.16)–(5.18). Now, if we superpose the system (5.16)–(5.18) with the L^2-gradient flow for $\mathcal{F}$ in (5.7), we obtain the regularized nonsmooth gray value Cahn–Hilliard inpainting model:

$$\partial_t u_i = \Delta w_i + \omega(f_i - u_i), \tag{5.19}$$

$$w_i = -\varepsilon \Delta u_i - \frac{1}{\varepsilon}(\boldsymbol{T}\mathbf{u})_i + \frac{1}{c}\min(0, u_i) + \frac{1}{N}\sum_{j=1}^{N}\left[-\frac{1}{c}\min(0, u_j) + \frac{1}{\varepsilon}(\boldsymbol{T}\mathbf{u})_j\right], \tag{5.20}$$

$$\nabla u_i \cdot \mathbf{n} = \nabla w_i \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega, \tag{5.21}$$

for $i = 1, \ldots, N$. We call this system also as the *(regularized) nonsmooth vector-valued modified Cahn–Hilliard equation*.

After the derivation of the constitutive vector-valued Cahn–Hilliard equations, we are going to study their discretizations in order to be able to solve them numerically. We start with the discretization in time in the next section.

5.3 Time discretization

In Chapter 3.3 and 4.3, we have motivated the use of a fully implicit time discretization scheme in order to accurately capture the dynamics. In this chapter, we pursue a different objective: We wish to obtain a reconstructed image as fast as possible. Hence, in this sense, time discretization schemes with time step restrictions are not the best choice.

In the case of the smooth black-and-white Cahn–Hilliard inpainting model, Bertozzi et al. [19] proposed a semi-implicit scheme, the convexity splitting scheme (see Chapter 2.1.5). The authors conjectured unconditionally stability in the sense that solutions of the numerical scheme are bounded within a finite time interval independent of the time step size. Indeed, Schönlieb et al. [139] proved consistency, unconditional stability, and convergence of this scheme. The convexity splitting method was designed to solve gradient systems. But it can also be applied in a modified form to evolution equations that do not follow a variational principle. In particular, such equations include our Cahn–Hilliard inpainting models (5.12)–(5.14) and (5.19)–(5.21) as described further on.

In the following, we denote by $\tau > 0$ the time step size and by $t_{n-1} = (n-1)\tau$, $n \in \mathbb{N}$, discrete times. We start with the smooth setting (5.12)–(5.14). We will prove consistency, unconditional stability, and convergence of a semi-implicit time-discrete scheme. Afterwards, we go over to the nonsmooth setting (5.19)–(5.21).

5.3.1 Smooth systems

Let us focus on the smooth setting and the fourth-order formulation of (5.12)–(5.14):

$$\partial_t u_i = -\Delta\left(\varepsilon\Delta u_i - \frac{1}{\varepsilon}\psi'_{\mathrm{pol}}(u_i) + \frac{1}{\varepsilon N}\sum_{j=1}^{N}\psi'_{\mathrm{pol}}(u_j)\right) + \omega(f_i - u_i), \tag{5.22}$$

$$\nabla u_i \cdot \mathbf{n} = \nabla(\Delta u_i)\cdot\mathbf{n} = 0 \quad \text{on } \partial\Omega, \tag{5.23}$$

for $i = 1,\dots,N$. Let $\mathbf{u}$ be the exact solution of (5.22)–(5.23) and $\mathbf{u}^{(n)} = \mathbf{u}(n\tau)$ the exact solution at time $n\tau$. We denote by $\mathbf{U}^{(n)}$ be the n-th iterate of the time-discrete scheme (5.26)–(5.27), which is derived next.

In the following, we extend the numerical analysis of the convexity splitting scheme for the smooth binary Cahn–Hilliard inpainting model studied in [139] to the vector-valued inpainting model (5.22)–(5.23).

As described in Chapter 2.1.5, the original idea of convexity splitting applied to gradient systems is to write the considered energy functional as the sum of a convex plus a concave energy functional. The convex part is then treated implicitly whilst the concave part is treated explicitly. Under the right conditions, this approach leads to an unconditionally gradient stable time discretization scheme. As pointed out in the previous sections, the smooth vector-valued modified Cahn–Hilliard equation as a whole is not given by a gradient flow. Especially, our proposed model arises as a superposition of the H^{-1}-gradient flow for $\mathcal{E}$ in (5.3) under the constraint (5.1) and the L^2-gradient flow for $\mathcal{F}$ in (5.7). In this case, convexity splitting is applied to each of these energies. To be more precise, we split $\mathcal{E}$ as $\mathcal{E} = \mathcal{E}_c - \mathcal{E}_e$, where

$$\mathcal{E}_c(\mathbf{u}) = \int_\Omega \frac{\varepsilon}{2}\sum_{i=1}^{N}|\nabla u_i|^2 + \frac{C_1}{2}\sum_{i=1}^{N}u_i^2 \,\mathrm{d}\mathbf{x},$$

$$\mathcal{E}_e(\mathbf{u}) = \int_\Omega -\frac{1}{\varepsilon}\psi_{\mathrm{pol}}(\mathbf{u}) + \frac{C_1}{2}\sum_{i=1}^{N}u_i^2 \,\mathrm{d}\mathbf{x},$$

as well as $\mathcal{F} = \mathcal{F}_c - \mathcal{F}_e$, where

$$\mathcal{F}_c(\mathbf{u}) = \int_\Omega \frac{C_2}{2}\sum_{i=1}^{N}u_i^2 \,\mathrm{d}\mathbf{x},$$

$$\mathcal{F}_e(\mathbf{u}) = \int_\Omega -\frac{\omega}{2}\sum_{i=1}^{N}(f_i - u_i)^2 + \frac{C_2}{2}\sum_{i=1}^{N}u_i^2 \,\mathrm{d}\mathbf{x}.$$

The constants C_1 and C_2 are positive and need to be chosen large enough such that the energies $\mathcal{E}_c$, $\mathcal{E}_e$, $\mathcal{F}_c$, and $\mathcal{F}_e$ are strictly convex. It is easy to see that $\mathcal{E}_c$ and $\mathcal{F}_c$ are already strictly convex for $C_1 > 0$ and $C_2 > 0$. The crucial points are the energy functionals that contain the nonconvex potential function $\psi_{\mathrm{pol}}(\mathbf{u})$ as well as the fidelity terms $\omega(f_i - u_i)^2,\ i = 1, \ldots, N$.

Now, we want to give the requirements for the constants C_1 and C_2 to make sure that $\mathcal{E}_e$ and $\mathcal{F}_e$ are strictly convex. However, as in Chapter 4.3.1, the quartic growth of $\psi_{\text{pol}}(\mathbf{u})$ at infinity introduces various technical difficulties in the analysis. Therefore, as in Chapter 4.3.1, we consider a truncated multi-well potential. To be more precise, we restrict the growth of $\psi_{\text{pol}}(\mathbf{u})$ to be quadratic for $u_i \leq 1 - M$ and $u_i \geq M$ for a given constant M. In the following, we write $\tilde{\psi}$ for the truncated version of ψ_{pol}. Using the truncation technique, we get the following condition: There exists a constant S such that

$$\max_{\mathbf{s} \in \mathbb{R}^N} \left| \frac{\partial^2 \tilde{\psi}}{\partial {u_i}^2}(\mathbf{s}) \right| \leq S \quad \forall i = 1, \ldots, N. \tag{5.24}$$

With the use of (5.24) we can prove:

Lemma 5.1. *$\mathcal{E}_e$ and $\mathcal{F}_e$ are strictly convex if C_1 is comparable to $\frac{1}{\varepsilon}$ and C_2 is comparable to ω_0 provided that $\psi = \psi_{\text{pol}}$ is replaced by its truncated version $\tilde{\psi}$.*

Proof. Let $\mathbf{u} = [u_1, \ldots, u_N]^T \in H^1(\Omega)^N$ and $\mathbf{v} = [v_1, \ldots, v_N]^T \in H^1(\Omega)^N$. Based on [145, p. 54], we have to show

$$\begin{aligned} \mathcal{E}_e(\mathbf{u} + \mathbf{v}) - \mathcal{E}_e(\mathbf{u}) &\geq \lim_{\delta \to 0} \frac{\mathcal{E}_e(\mathbf{u} + \delta\mathbf{v}) - \mathcal{E}_e(\mathbf{u})}{\delta}, \\ \mathcal{F}_e(\mathbf{u} + \mathbf{v}) - \mathcal{F}_e(\mathbf{u}) &\geq \lim_{\delta \to 0} \frac{\mathcal{F}_e(\mathbf{u} + \delta\mathbf{v}) - \mathcal{F}_e(\mathbf{u})}{\delta}. \end{aligned}$$

We have

$$\begin{aligned} \mathcal{E}_e(\mathbf{u} + \mathbf{v}) - \mathcal{E}_e(\mathbf{u}) &= \int_\Omega -\frac{1}{\varepsilon}\left(\psi(\mathbf{u} + \mathbf{v}) - \psi(\mathbf{u})\right) + \frac{C_1}{2} \sum_{i=1}^N \left((u_i + v_i)^2 - u_i^2\right) \mathrm{d}\mathbf{x} \\ &= \int_\Omega -\frac{1}{\varepsilon}\left(\psi(\mathbf{u} + \mathbf{v}) - \psi(\mathbf{u})\right) + \frac{C_1}{2} \sum_{i=1}^N \left(v_i^2 + 2u_i v_i\right) \mathrm{d}\mathbf{x}. \end{aligned} \tag{5.25}$$

As ψ is a smooth function, we can consider its Taylor expansion

$$\psi(\mathbf{u} + \mathbf{v}) = \psi(\mathbf{u}) + \sum_{i=1}^N \left(v_i \frac{\partial \psi}{\partial u_i}(\mathbf{u}) + \frac{1}{2} v_i^2 \frac{\partial^2 \psi}{\partial {u_i}^2}(\mathbf{s}) \right),$$

where $\mathbf{s}$ lies between $\mathbf{u} + \mathbf{v}$ and $\mathbf{u}$. Therefore, we obtain in (5.25)

$$\mathcal{E}_e(\mathbf{u} + \mathbf{v}) - \mathcal{E}_e(\mathbf{u}) = \int_\Omega -\frac{1}{\varepsilon} \sum_{i=1}^N \left(v_i \frac{\partial \psi}{\partial u_i}(\mathbf{u}) + \frac{1}{2} v_i^2 \frac{\partial^2 \psi}{\partial {u_i}^2}(\mathbf{s}) \right) + \frac{C_1}{2} \sum_{i=1}^N \left(v_i^2 + 2u_i v_i\right) \mathrm{d}\mathbf{x}.$$

Similarly, one can show

$$\lim_{\delta \to 0} \frac{\mathcal{E}_e(\mathbf{u} + \delta\mathbf{v}) - \mathcal{E}_e(\mathbf{u})}{\delta} = \int_\Omega -\frac{1}{\varepsilon} \sum_{i=1}^N v_i \frac{\partial \psi}{\partial u_i}(\mathbf{u}) + \frac{C_1}{2} \sum_{i=1}^N 2u_i v_i \ \mathrm{d}\mathbf{x},$$

which leads to

$$\begin{aligned}\mathcal{E}_e(\mathbf{u}+\mathbf{v}) - \mathcal{E}_e(\mathbf{u}) - \lim_{\delta\to 0}\frac{\mathcal{E}_e(\mathbf{u}+\delta\mathbf{v}) - \mathcal{E}_e(\mathbf{u})}{\delta} &= \int_\Omega -\frac{1}{2\varepsilon}\sum_{i=1}^N v_i^2 \frac{\partial^2\psi}{\partial {u_i}^2}(\mathbf{s}) + \frac{C_1}{2}\sum_{i=1}^N v_i^2 \,\mathrm{d}\mathbf{x}\\ &\overset{(5.24)}{\geq} \int_\Omega \left(\frac{C_1}{2} - \frac{S}{2\varepsilon}\right)\sum_{i=1}^N v_i^2 \,\mathrm{d}\mathbf{x}.\end{aligned}$$

Therefore, $\mathcal{E}_e$ is strictly convex if C_1 is comparable to $\frac{1}{\varepsilon}$. Proceeding the same way with the second energy functional $\mathcal{F}_e$ gives

$$\begin{aligned}\mathcal{F}_e(\mathbf{u}+\mathbf{v}) - \mathcal{F}_e(\mathbf{u}) - \lim_{\delta\to 0}\frac{\mathcal{F}_e(\mathbf{u}+\delta\mathbf{v}) - \mathcal{F}_e(\mathbf{u})}{\delta} &= \int_\Omega \left(\frac{C_2}{2} - \frac{\omega}{2}\right)\sum_{i=1}^N v_i^2 \,\mathrm{d}\mathbf{x}\\ &\overset{\omega\leq\omega_0}{\geq} \int_\Omega \left(\frac{C_2}{2} - \frac{\omega_0}{2}\right)\sum_{i=1}^N v_i^2 \,\mathrm{d}\mathbf{x}.\end{aligned}$$

Therefore, $\mathcal{F}_e$ is strictly convex if $C_2 > \omega_0$.

□

Note that these convexity requirements are the same as the ones for the smooth black-and-white Cahn–Hilliard inpainting model.

Remark 5.3. *An assumption consistent with (5.24) is also made in the numerical analysis for the smooth black-and-white Cahn–Hilliard inpainting model, see [139, Theorem 3.1]. There, the authors assume that the second derivative of the smooth potential evaluated at the previous time step is bounded. In particular, this assumption is needed below as well in order to prove the consistency, unconditional stability, and convergence of the resulting time-discrete scheme.*

The resulting time-discrete scheme is given by

$$\frac{\mathbf{U}^{(n)} - \mathbf{U}^{(n-1)}}{\tau} = -\mathrm{grad}_{H^{-1}}^{(5.1)}\Big(\mathcal{E}_c(\mathbf{U}^{(n)}) - \mathcal{E}_e(\mathbf{U}^{(n-1)})\Big) - \mathrm{grad}_{L^2}\Big(\mathcal{F}_c(\mathbf{U}^{(n)}) - \mathcal{F}_e(\mathbf{U}^{(n-1)})\Big).$$

This translates to a numerical scheme of the form

$$\begin{aligned}&\frac{U_i^{(n)} - U_i^{(n-1)}}{\tau} + \varepsilon\Delta^2 U_i^{(n)} - C_1\Delta U_i^{(n)} + C_2 U_i^{(n)}\\ &\quad= \frac{1}{\varepsilon}\Delta\psi'_{\mathrm{pol}}(U_i^{(n-1)}) - \frac{1}{\varepsilon N}\Delta\left(\sum_{j=1}^N \psi'_{\mathrm{pol}}(U_j^{(n-1)})\right) + \omega(f_i - U_i^{(n-1)}) \qquad (5.26)\\ &\qquad - C_1\Delta U_i^{(n-1)} + C_2 U_i^{(n-1)},\\ &\nabla U_i\cdot\mathbf{n} = \nabla(\Delta U_i)\cdot\mathbf{n} = 0 \quad \text{on } \partial\Omega, \qquad (5.27)\end{aligned}$$

for $i = 1,\ldots,N$. Next, we prove the consistency, unconditional stability, and convergence of the time discrete scheme (5.26)–(5.27) in the sense of Definition 2.21. In doing so, we follow [139, pp. 425–434] and extend the proof from the scalar to the vector-valued case. In the following, we assume $\mathbf{f} \in L^2(\Omega)^N$, $\mathbf{f} \in \mathcal{G}^N$ a.e. in Ω, and we write ψ instead of ψ_{pol}.

Theorem 5.2 (See [139, Theorem 3.1] for black-and-white Cahn–Hilliard inpainting.). *Let $\mathbf{u}$ be the exact solution of (5.22)–(5.23) and $\mathbf{u}^{(n)} = \mathbf{u}(n\tau)$ the exact solution at time $n\tau$. Let $\mathbf{U}^{(n)}$ be the n-th iterate of (5.26)–(5.27) with constants $C_1 > \frac{1}{\varepsilon}$, $C_2 > \omega_0$. Then, the following statements are true:*

1. *(Consistency). Under the assumption that $\|\mathbf{u}_{tt}\|_{-1}$ and $\|\nabla\Delta\mathbf{u}_t\|$ are bounded, the numerical scheme (5.26)–(5.27) is consistent with the continuous Equation (5.22)–(5.23) and of order one in time.*

Under the additional assumption that

$$\left|\frac{\partial^2\psi}{\partial {u_i}^2}\left(\mathbf{U}^{(n-1)}\right)\right| \leq S \quad \forall i = 1, \dots, N \tag{5.28}$$

for a nonnegative constant S, we further have:[1]

2. *(Unconditional stability). The solution sequence $\mathbf{U}^{(n)}$ is bounded on a finite time interval $[0, T]$ for all $\tau > 0$. In particular for $n\tau \leq T$, $T > 0$ fixed, we have for every $\tau > 0$*

$$\begin{aligned} &\|\nabla\mathbf{U}^{(n)}\|^2 + \tau K_1\|\Delta\mathbf{U}^{(n)}\|^2 \\ &\quad \leq e^{K_2 T}\left(\|\nabla\mathbf{U}^{(0)}\|^2 + \tau K_1\|\Delta\mathbf{U}^{(0)}\|^2 + \tau C(\Omega, D, \omega_0, \mathbf{f})\right), \end{aligned} \tag{5.29}$$

for suitable constants K_1 and K_2, and a constant C depending on Ω, D, ω_0, $\mathbf{f}$ only.

3. *(Convergence). The discretization error $\mathbf{e}^{(n)}$, given by $\mathbf{e}^{(n)} = \mathbf{u}^{(n)} - \mathbf{U}^{(n)}$, converges to zero in $L^2(\Omega)$ as $\tau \to 0$. In particular, we have for $n\tau \leq T$, $T > 0$ fixed that*

$$\|\nabla\mathbf{e}^{(n)}\|^2 + \tau\frac{C_1}{\tilde{C}}\|\Delta\mathbf{e}^{(n)}\|^2 \leq \frac{T}{\tilde{C}}e^{K_1 T}C\tau^2 \tag{5.30}$$

for suitable constants $C, \tilde{C}, K_1$.

We start with the proof for consistency.

Proof of consistency. By rearranging the terms in (5.26), we can write our time-discrete scheme in the form of (2.10):

$$U_i^{(n)} = U_i^{(n-1)} + \tau G_i^{(n-1)}(\mathbf{U}^{(n-1)}, \mathbf{U}^{(n)}, D^\alpha\mathbf{U}^{(n-1)}, D^\alpha\mathbf{U}^{(n)}),$$

where

$$\begin{aligned} G_i^{(n-1)}(\mathbf{U}^{(n-1)}, \mathbf{U}^{(n)}, D^\alpha\mathbf{U}^{(n-1)}, D^\alpha\mathbf{U}^{(n)}) &= C_2(U_i^{(n-1)} - U_i^{(n)}) - \varepsilon\Delta^2 U_i^{(n)} - C_1\Delta(U_i^{(n-1)} - U_i^{(n)}) \\ &+ \frac{1}{\varepsilon}\Delta\frac{\partial\psi}{\partial u_i}(\mathbf{U}^{(n-1)}) - \frac{1}{\varepsilon N}\Delta\left(\sum_{j=1}^{N}\frac{\partial\psi}{\partial u_j}(\mathbf{U}^{(n-1)})\right) + \omega(f_i - U_i^{(n-1)}). \end{aligned}$$

[1] Note that this assumption complies with the one in (5.24). It is even less restrictive, since we only need the boundedness evaluated at the solution of the previous time step.

Let $\boldsymbol{\eta}^{(n-1)} = [\eta_1^{(n-1)}, \dots, \eta_N^{(n-1)}]^T$ be the local truncation error vector defined as in (2.11):

$$\begin{aligned}\eta_i^{(n-1)} &= \frac{u_i^{(n)} - u_i^{(n-1)}}{\tau} - G_i^{(n-1)}(\mathbf{u}^{(n-1)}, \mathbf{u}^{(n)}, D^\alpha \mathbf{u}^{(n-1)}, D^\alpha \mathbf{u}^{(n)}) \\ &= \frac{u_i^{(n)} - u_i^{(n-1)}}{\tau} - C_2(u_i^{(n-1)} - u_i^{(n)}) + \varepsilon \Delta^2 u_i^{(n)} + C_1 \Delta(u_i^{(n-1)} - u_i^{(n)}) \\ &\quad - \frac{1}{\varepsilon}\Delta \frac{\partial \psi}{\partial u_i}(\mathbf{u}^{(n-1)}) + \frac{1}{\varepsilon N}\Delta\left(\sum_{j=1}^N \frac{\partial \psi}{\partial u_j}(\mathbf{u}^{(n-1)})\right) - \omega(f_i - u_i^{(n-1)}). \end{aligned} \tag{5.31}$$

Then

$$\eta_i^{(n-1)} = \eta_i^{(n-1),1} + \eta_i^{(n-1),2}, \tag{5.32}$$

with

$$\begin{aligned}\eta_i^{(n-1),1} &= \frac{u_i^{(n)} - u_i^{(n-1)}}{\tau} - \partial_t u_i^{(n-1)}, \\ \eta_i^{(n-1),2} &= \varepsilon \tau \Delta^2 \left(\frac{u_i^{(n)} - u_i^{(n-1)}}{\tau}\right) - C_1 \tau \Delta \left(\frac{u_i^{(n)} - u_i^{(n-1)}}{\tau}\right) + C_2 \tau \left(\frac{u_i^{(n)} - u_i^{(n-1)}}{\tau}\right). \end{aligned}$$

Here, we have used that $u_i^{(n-1)} = u_i((n-1)\tau)$ is the exact solution at time $(n-1)\tau$. That means it fulfills the continuous Equation (5.22)–(5.23):

$$\partial_t u_i^{(n-1)} = -\Delta\left(\varepsilon \Delta u_i^{(n-1)} - \frac{1}{\varepsilon}\frac{\partial \psi}{\partial u_i}(\mathbf{u}^{(n-1)}) + \frac{1}{\varepsilon N}\sum_{j=1}^N \frac{\partial \psi}{\partial u_j}(\mathbf{u}^{(n-1)})\right) + \omega(f_i - u_i^{(n-1)}).$$

The partition of the local truncation error into the sum of $\eta_i^{(n-1),1}$ and $\eta_i^{(n-1),2}$ is exactly the same as in the scalar inpainting model, see [139, p. 426]. Hence, the rest of the proof follows the proof of [139, Proposition 3.2]. That means, assuming that $\|\mathbf{u}_{tt}\|_{-1}$ and $\|\nabla \Delta \mathbf{u}_t\|$ are bounded, the global truncation error η is given by

$$\eta = \max_n \|\eta^{(n)}\|_{-1} = \mathcal{O}(\tau) \quad \text{as } \tau \to 0.$$

□

Next, we will proof the unconditional stability.

Proof of unconditional stability. First, we multiply the ith row of the time-discrete model (5.26) with $-\Delta U_i^{(n)}$ and integrate over Ω:

$$\begin{aligned} -\left(\frac{1}{\tau} + C_2\right)&(U_i^{(n)}, \Delta U_i^{(n)}) - \varepsilon(\Delta^2 U_i^{(n)}, \Delta U_i^{(n)}) + C_1(\Delta U_i^{(n)}, \Delta U_i^{(n)}) \\ &= -\left(\frac{1}{\tau} + C_2\right)(U_i^{(n-1)}, \Delta U_i^{(n)}) - (\omega(f_i - U_i^{(n-1)}), \Delta U_i^{(n)}) + C_1(\Delta U_i^{(n-1)}, \Delta U_i^{(n)}) \\ &\quad - \frac{1}{\varepsilon}\left(\Delta \frac{\partial \psi}{\partial u_i}(\mathbf{U}^{(n-1)}), \Delta U_i^{(n)}\right) + \frac{1}{\varepsilon N}\sum_{j=1}^N \left(\Delta \frac{\partial \psi}{\partial u_j}(\mathbf{U}^{(n-1)}), \Delta U_i^{(n)}\right). \end{aligned} \tag{5.33}$$

Lemma 2.21 applied to the second last term on the right-hand side of (5.33) yields

$$\begin{aligned}
&\int_\Omega \Delta \frac{\partial \psi}{\partial u_i}(\mathbf{U}^{(n-1)})\Delta U_i^{(n)}\,\mathrm{d}\mathbf{x}\\
&\quad = \int_{\partial\Omega} \Delta U_i^{(n)} \nabla\left(\frac{\partial \psi}{\partial u_i}(\mathbf{U}^{(n-1)})\right)\cdot \mathbf{n}\,\mathrm{d}\mathbf{s} - \int_\Omega \nabla\left(\frac{\partial \psi}{\partial u_i}(\mathbf{U}^{(n-1)})\right)\cdot \nabla\Delta U_i^{(n)}\,\mathrm{d}\mathbf{x}\\
&\quad = \int_{\partial\Omega} \Delta U_i^{(n)} \frac{\partial^2 \psi}{\partial {u_i}^2}(\mathbf{U}^{(n-1)})\underbrace{\nabla U_i^{(n-1)}\cdot \mathbf{n}}_{=0}\,\mathrm{d}\mathbf{s} - \int_\Omega \frac{\partial^2 \psi}{\partial {u_i}^2}(\mathbf{U}^{(n-1)})\nabla U_i^{(n-1)}\cdot \nabla\Delta U_i^{(n)}\,\mathrm{d}\mathbf{x}\\
&\quad = -\left(\frac{\partial^2 \psi}{\partial u_i^2}(\mathbf{U}^{(n-1)})\nabla U_i^{(n-1)}, \nabla\Delta U_i^{(n)}\right),
\end{aligned}$$

where we have used the Neumann boundary conditions that are imposed in (5.27) as well as the splitting assumption $\psi(\mathbf{u}) = \sum_{i=1}^N \psi_i(u_i)$. The last term on the right-hand side of (5.33) can be handled in the same way. Applying the Neumann boundary conditions again to some remaining terms in (5.33), we obtain

$$\begin{aligned}
&\left(\frac{1}{\tau}+C_2\right)\|\nabla U_i^{(n)}\|^2 + \varepsilon\|\nabla\Delta U_i^{(n)}\|^2 + C_1\|\Delta U_i^{(n)}\|^2\\
&\qquad = \left(\frac{1}{\tau}+C_2\right)(\nabla U_i^{(n-1)}, \nabla U_i^{(n)}) + (\nabla\omega(f_i - U_i^{(n-1)}), \nabla U_i^{(n)})\\
&\qquad\quad + C_1\left(\Delta U_i^{(n-1)}, \Delta U_i^{(n)}\right) + \frac{1}{\varepsilon}\left(\frac{\partial^2 \psi}{\partial u_i^2}(\mathbf{U}^{(n-1)})\nabla U_i^{(n-1)}, \nabla\Delta U_i^{(n)}\right)\\
&\qquad\quad - \frac{1}{\varepsilon N}\sum_{j=1}^N\left(\frac{\partial^2 \psi}{\partial u_j^2}(\mathbf{U}^{(n-1)})\nabla U_j^{(n-1)}, \nabla\Delta U_i^{(n)}\right).
\end{aligned} \tag{5.34}$$

Using Young's inequality (Lemma 2.12)

$$\begin{aligned}
(\nabla U_i^{(n-1)}, \nabla U_i^{(n)}) &\le \frac{1}{2}\|\nabla U_i^{(n-1)}\|^2 + \frac{1}{2}\|\nabla U_i^{(n)}\|^2,\\
\left(\frac{\partial^2 \psi}{\partial u_i^2}(\mathbf{U}^{(n-1)})\nabla U_i^{(n-1)}, \nabla\Delta U_i^{(n)}\right) &\le \frac{1}{2\delta_1}\left\|\frac{\partial^2 \psi}{\partial u_i^2}(\mathbf{U}^{(n-1)})\nabla U_i^{(n-1)}\right\|^2 + \frac{\delta_1}{2}\left\|\nabla\Delta U_i^{(n)}\right\|^2,\\
\left(\Delta U_i^{(n-1)}, \Delta U_i^{(n)}\right) &\le \frac{1}{2}\|\Delta U_i^{(n-1)}\|^2 + \frac{1}{2}\|\Delta U_i^{(n-1)}\|^2,\\
(\nabla\omega(f_i - U_i^{(n-1)}), \nabla U_i^{(n)}) &\le \frac{1}{2}\|\nabla\omega(f_i - U_i^{(n-1)})\|^2 + \frac{1}{2}\|\nabla U_i^{(n)}\|^2,
\end{aligned}$$

where $\delta_1 > 0$, we obtain in (5.34)

$$
\begin{aligned}
0 &\overset{(5.34)}{=} \left(\frac{1}{\tau}+C_2\right)\|\nabla U_i^{(n)}\|^2 + \varepsilon\|\nabla\Delta U_i^{(n)}\|^2 + C_1\|\Delta U_i^{(n)}\|^2 - \left(\frac{1}{\tau}+C_2\right)(\nabla U_i^{(n-1)},\nabla U_i^{(n)}) \\
&\quad - (\nabla\omega(f_i - U_i^{(n-1)}),\nabla U_i^{(n)}) - C_1\left(\Delta U_i^{(n-1)},\Delta U_i^{(n)}\right) \\
&\quad - \frac{1}{\varepsilon}\left(\frac{\partial^2\psi}{\partial u_i^2}(\mathbf{U}^{(n-1)})\nabla U_i^{(n-1)},\nabla\Delta U_i^{(n)}\right) \\
&\quad + \frac{1}{\varepsilon N}\sum_{j=1}^{N}\left(\frac{\partial^2\psi}{\partial u_j^2}(\mathbf{U}^{(n-1)})\nabla U_j^{(n-1)},\nabla\Delta U_i^{(n)}\right) \\
&\geq \left(\frac{1}{\tau}+C_2\right)\|\nabla U_i^{(n)}\|^2 + \varepsilon\|\nabla\Delta U_i^{(n)}\|^2 + C_1\|\Delta U_i^{(n)}\|^2 - \frac{1}{2}\left(\frac{1}{\tau}+C_2\right)\left(\|\nabla U_i^{(n-1)}\|^2 + \|\nabla U_i^{(n)}\|^2\right) \\
&\quad - \frac{1}{2}\|\nabla\omega(f_i - U_i^{(n-1)})\|^2 - \frac{1}{2}\|\nabla U_i^{(n)}\|^2 - \frac{C_1}{2}\left(\|\Delta U_i^{(n-1)}\|^2 + \|\Delta U_i^{(n-1)}\|^2\right) \\
&\quad - \frac{1}{2\varepsilon}\left(\frac{1}{\delta_1}\left\|\frac{\partial^2\psi}{\partial u_i^2}(\mathbf{U}^{(n-1)})\nabla U_i^{(n-1)}\right\|^2 + \delta_1\left\|\nabla\Delta U_i^{(n)}\right\|^2\right) \\
&\quad - \frac{1}{2\varepsilon N}\sum_{j=1}^{N}\left(\frac{1}{\delta_1}\left\|\frac{\partial^2\psi}{\partial u_j^2}(\mathbf{U}^{(n-1)})\nabla U_j^{(n-1)}\right\|^2 + \delta_1\left\|\nabla\Delta U_i^{(n)}\right\|^2\right).
\end{aligned}
$$

After rearranging, we get

$$
\begin{aligned}
&\left(\frac{1}{2\tau}+\frac{C_2}{2}-\frac{1}{2}\right)\|\nabla U_i^{(n)}\|^2 + \frac{C_1}{2}\|\Delta U_i^{(n)}\|^2 + \left(\varepsilon - \frac{\delta_1}{\varepsilon}\right)\|\nabla\Delta U_i^{(n)}\|^2 \\
&\qquad \leq \left(\frac{1}{2\tau}+\frac{C_2}{2}\right)\|\nabla U_i^{(n-1)}\|^2 + \frac{1}{2}\|\nabla\omega(f_i - U_i^{(n-1)})\|^2 + \frac{C_1}{2}\|\Delta U_i^{(n-1)}\|^2 \\
&\qquad\qquad + \frac{1}{2\delta_1\varepsilon}\left\|\frac{\partial^2\psi}{\partial u_i^2}(\mathbf{U}^{(n-1)})\nabla U_i^{(n-1)}\right\|^2 + \frac{1}{2\delta_1 N\varepsilon}\sum_{j=1}^{N}\left\|\frac{\partial^2\psi}{\partial u_j^2}(\mathbf{U}^{(n-1)})\nabla U_j^{(n-1)}\right\|^2.
\end{aligned}
$$

Using the estimate

$$
\|\nabla\omega(f_i - U_i^{(n-1)})\|^2 \leq 2\omega_0^2\|\nabla U_i^{(n-1)}\|^2 + C(\Omega, D, \omega_0, f_i),
$$

as stated in the middle of [139, p. 427], together with $\delta_1 = \varepsilon^2$, we obtain

$$
\begin{aligned}
&\left(\frac{1}{2\tau}+\frac{C_2}{2}-\frac{1}{2}\right)\|\nabla U_i^{(n)}\|^2 + \frac{C_1}{2}\|\Delta U_i^{(n)}\|^2 \\
&\qquad \leq \left(\frac{1}{2\tau}+\frac{C_2}{2}+\omega_0^2\right)\|\nabla U_i^{(n-1)}\|^2 + \frac{1}{2\varepsilon^3}\left\|\frac{\partial^2\psi}{\partial u_i^2}(\mathbf{U}^{(n-1)})\nabla U_i^{(n-1)}\right\|^2 \\
&\qquad\quad + \frac{1}{2N\varepsilon^3}\sum_{j=1}^{N}\left\|\frac{\partial^2\psi}{\partial u_j^2}(\mathbf{U}^{(n-1)})\nabla U_j^{(n-1)}\right\|^2 + \frac{C_1}{2}\|\Delta U_i^{(n-1)}\|^2 + C(\Omega, D, \omega_0, f_i).
\end{aligned}
$$

Summing up these inequalities over $i = 1, \dots, N$, we get

$$\begin{aligned}\left(\frac{1}{2\tau} + \frac{C_2}{2} - \frac{1}{2}\right) \|\nabla \mathbf{U}^{(n)}\|^2 + \frac{C_1}{2} \|\Delta \mathbf{U}^{(n)}\|^2 \\ \leq \left(\frac{1}{2\tau} + \frac{C_2}{2} + \omega_0^2\right) \|\nabla \mathbf{U}^{(n-1)}\|^2 + \frac{1}{\varepsilon^3} \sum_{i=1}^{N} \left\| \frac{\partial^2 \psi}{\partial u_i^2}(\mathbf{U}^{(n-1)}) \nabla U_i^{(n-1)} \right\|^2 \\ + \frac{C_1}{2} \|\Delta \mathbf{U}^{(n-1)}\|^2 + C(\Omega, D, \omega_0, \mathbf{f}).\end{aligned}$$

Because of (5.28), we can estimate

$$\left\| \frac{\partial^2 \psi}{\partial u_i^2}(\mathbf{U}^{(n-1)}) \nabla U_i^{(n-1)} \right\|^2 \leq S^2 \left\| \nabla U_i^{(n-1)} \right\|^2,$$

and we have

$$\begin{aligned}\left(\frac{1}{2\tau} + \frac{C_2}{2} - \frac{1}{2}\right) \|\nabla \mathbf{U}^{(n)}\|^2 + \frac{C_1}{2} \|\Delta \mathbf{U}^{(n)}\|^2 \\ \leq \left(\frac{1}{2\tau} + \frac{C_2}{2} + \omega_0^2 + \frac{S^2}{\varepsilon^3}\right) \|\nabla \mathbf{U}^{(n-1)}\|^2 + \frac{C_1}{2} \|\Delta \mathbf{U}^{(n-1)}\|^2 + C(\Omega, D, \omega_0, \mathbf{f}).\end{aligned}$$

This is (almost) the same estimation as the last inequality in [139, p. 427]. Therefore, the rest of the proof follows the proof of [139, Proposition 3.3]. This results in the desired estimation (5.29). □

In order to proof the convergence, we need two auxiliary lemmas. The first one is the following:

Lemma 5.3 (See [139, Lemma 3.2] for black-and-white Cahn–Hilliard inpainting.)**.** *The error* $\mathbf{e}^{(n-1)}$ *between the exact and approximate solution, defined as in Theorem 5.2, fulfills*

$$\int_{\Omega} \mathbf{e}^{(n-1)} \, \mathrm{d}\mathbf{x} = O(\tau^2).$$

Proof. Because of the fidelity term in (5.22) and (5.26), solutions of these equations are not mass preserving, i.e., $\int_{\Omega} \mathbf{e}^{(n-1)} \, \mathrm{d}\mathbf{x}$ does not in general vanish. In fact, we have for a solution $\mathbf{u}^{(n-1)}$ of (5.22)–(5.23)

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\int_\Omega u_i^{(n-1)}\,\mathrm{d}\mathbf{x} &= -\varepsilon\int_\Omega \Delta^2 u_i^{(n-1)}\,\mathrm{d}\mathbf{x} + \frac{1}{\varepsilon}\int_\Omega \Delta\frac{\partial\psi}{\partial u_i}(\mathbf{u}^{(n-1)})\,\mathrm{d}\mathbf{x} \\
&\quad - \frac{1}{\varepsilon N}\int_\Omega \Delta\sum_{j=1}^N \frac{\partial\psi}{\partial u_j}(\mathbf{u}^{(n-1)})\,\mathrm{d}\mathbf{x} + \int_\Omega \omega(f_i - u_i^{(n-1)})\,\mathrm{d}\mathbf{x} \\
&= -\varepsilon\int_{\partial\Omega} \underbrace{\nabla\Delta u_i^{(n-1)}\cdot\mathbf{n}}_{=0}\,\mathrm{d}\mathbf{s} + \frac{1}{\varepsilon}\int_{\partial\Omega}\nabla\left(\frac{\partial\psi}{\partial u_i}(\mathbf{u}^{(n-1)})\right)\cdot\mathbf{n}\,\mathrm{d}\mathbf{s} \\
&\quad - \frac{1}{\varepsilon N}\sum_{j=1}^N\int_{\partial\Omega}\nabla\left(\frac{\partial\psi}{\partial u_j}(\mathbf{u}^{(n-1)})\right)\cdot\mathbf{n}\,\mathrm{d}\mathbf{s} + \int_\Omega \omega(f_i - u_i^{(n-1)})\,\mathrm{d}\mathbf{x} \\
&= \frac{1}{\varepsilon}\int_{\partial\Omega}\frac{\partial^2\psi}{\partial u_i^2}(\mathbf{u}^{(n-1)})\underbrace{\nabla u_i^{(n-1)}\cdot\mathbf{n}}_{=0}\,\mathrm{d}\mathbf{s} \\
&\quad - \frac{1}{\varepsilon N}\sum_{j=1}^N\int_{\partial\Omega}\frac{\partial^2\psi}{\partial u_j^2}(\mathbf{u}^{(n-1)})\underbrace{\nabla u_j^{(n-1)}\cdot\mathbf{n}}_{=0}\,\mathrm{d}\mathbf{s} + \int_\Omega \omega(f_i - u_i^{(n-1)})\,\mathrm{d}\mathbf{x} \\
&= \int_\Omega \omega(f_i - u_i^{(n-1)})\,\mathrm{d}\mathbf{x},
\end{aligned}$$

where we have used Lemma 2.21. In particular,

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_D u_i^{(n-1)}\,\mathrm{d}\mathbf{x} = 0 \tag{5.35}$$

since $\omega(\mathbf{x}) = 0$ for all $\mathbf{x} \in D$. A similar computation for the discrete solution of (5.26)–(5.27) shows that

$$\left(\frac{1}{\tau} + C_2\right)\int_\Omega \left(U_i^{(n)} - U_i^{(n-1)}\right)\mathrm{d}\mathbf{x} = \int_\Omega \omega(f_i - U_i^{(n-1)})\,\mathrm{d}\mathbf{x},$$

and in particular

$$\left(\frac{1}{\tau} + C_2\right)\int_D \left(U_i^{(n)} - U_i^{(n-1)}\right)\mathrm{d}\mathbf{x} = 0. \tag{5.36}$$

Next, consider

$$\begin{aligned}
&\frac{e_i^{(n)} - e_i^{(n-1)}}{\tau} + \varepsilon\Delta^2 e_i^{(n)} - C_1\Delta e_i^{(n)} + C_2 e_i^{(n)} \\
&\qquad = -\left(\frac{U_i^{(n)} - U_i^{(n-1)}}{\tau} + \varepsilon\Delta^2 U_i^{(n)} - C_1\Delta U_i^{(n)} + C_2 U_i^{(n)}\right) \\
&\qquad\qquad + \frac{u_i^{(n)} - u_i^{(n-1)}}{\tau} + \varepsilon\Delta^2 u_i^{(n)} - C_1\Delta u_i^{(n)} + C_2 u_i^{(n)},
\end{aligned} \tag{5.37}$$

where we have used the definition of the discretization error. In (5.37), we express the terms in brackets using (5.26) and the remaining terms via the local truncation error $\eta_i^{(n-1)}$ in (5.31). Hence, we obtain

$$\begin{aligned}
&\frac{e_i^{(n)} - e_i^{(n-1)}}{\tau} + \varepsilon\Delta^2 e_i^{(n)} - C_1\Delta e_i^{(n)} + C_2 e_i^{(n)} \\
&= -\left(\frac{1}{\varepsilon}\Delta\left(\frac{\partial\psi}{\partial u_i}(\mathbf{U}^{(n-1)})\right) - \frac{1}{\varepsilon N}\Delta\left(\sum_{j=1}^{N}\frac{\partial\psi}{\partial u_j}(\mathbf{U}^{(n-1)})\right)\right. \\
&\quad \left. +\omega(f_i - U_i^{(n-1)}) - C_1\Delta U_i^{(n-1)} + C_2 U_i^{(n-1)}\right) \\
&\quad + \left(C_2 u_i^{(n-1)} - C_1\Delta u_i^{(n-1)} + \frac{1}{\varepsilon}\Delta\left(\frac{\partial\psi}{\partial u_i}(\mathbf{u}^{(n-1)})\right)\right. \\
&\quad \left. -\frac{1}{\varepsilon N}\Delta\left(\sum_{j=1}^{N}\frac{\partial\psi}{\partial u_j}(\mathbf{u}^{(n-1)})\right) + \omega(f_i - u_i^{(n-1)})\right) + \eta_i^{(n-1)} \\
&= -\left[\frac{1}{\varepsilon}\Delta\left(\frac{\partial\psi}{\partial u_i}(\mathbf{U}^{(n-1)}) - \frac{\partial\psi}{\partial u_i}(\mathbf{u}^{(n-1)})\right) - \frac{1}{\varepsilon N}\Delta\sum_{j=1}^{N}\left(\frac{\partial\psi}{\partial u_j}(\mathbf{U}^{(n-1)}) - \frac{\partial\psi}{\partial u_j}(\mathbf{u}^{(n-1)})\right)\right. \\
&\quad \left. -C_1\Delta\left(U_i^{(n-1)} - u_i^{(n-1)}\right) + C_2\left(U_i^{(n-1)} - u_i^{(n-1)}\right) - \omega(U_i^{(n-1)} - u_i^{(n-1)})\right] \\
&\quad + \eta_i^{(n-1)}.
\end{aligned} \tag{5.38}$$

As before, integrating over Ω, applying Lemma 2.21 and the zero Neumann boundary conditions for $\mathbf{u}^{(n)}$, $\mathbf{u}^{(n-1)}$, $\mathbf{U}^{(n)}$, $\mathbf{U}^{(n-1)}$, we get

$$\left(\frac{1}{\tau} + C_2\right)\int_\Omega (e_i^{(n)} - e_i^{(n-1)})\,\mathrm{d}\mathbf{x} + \int_\Omega \omega\, e_i^{(n-1)}\,\mathrm{d}\mathbf{x} = \int_\Omega \eta_i^{(n-1)}\,\mathrm{d}\mathbf{x}, \tag{5.39}$$

where

$$\begin{aligned}
\int_\Omega \eta_i^{(n-1)}\,\mathrm{d}\mathbf{x} &\overset{(5.32)}{=} \int_\Omega \left(\eta_i^{(n-1),1} + \eta_i^{(n-1),2}\right)\mathrm{d}\mathbf{x} \\
&= \left(\frac{1}{\tau} + C_2\right)\int_\Omega (u_i^{(n)} - u_i^{(n-1)})\,\mathrm{d}\mathbf{x} - \int_\Omega \partial_t u_i^{(n-1)}\,\mathrm{d}\mathbf{x} \\
&= \left(\frac{1}{\tau} + C_2\right)\int_\Omega \left((u_i^{(n-1)} + \tau\partial_t u_i^{(n-1)} + \mathcal{O}(\tau^2)) - u_i^{(n-1)}\right)\mathrm{d}\mathbf{x} - \int_\Omega \partial_t u_i^{(n-1)}\,\mathrm{d}\mathbf{x} \\
&= \tau C_2\int_\Omega \partial_t u_i^{(n-1)}\,\mathrm{d}\mathbf{x} + \int_\Omega \mathcal{O}(\tau + C_2\tau^2)\,\mathrm{d}\mathbf{x} \\
&= \mathcal{O}(\tau).
\end{aligned}$$

This is the same estimation as in the upper part of [139, p. 430]. Therefore, the rest of the proof follows the proof of [139, Lemma 3.2]. This results in the desired estimation

$$(1 + C_2\tau)\int_\Omega \mathbf{e}^{(n-1)}\,\mathrm{d}\mathbf{x} = \mathcal{O}(\tau^2).$$

□

The second auxiliary lemma is the following:

Lemma 5.4 (See [139, Lemma 3.3] for black-and-white Cahn–Hilliard inpainting.)**.** *Let $\mathbf{u}^{(n-1)}$ be the exact solution of (5.22)–(5.23) at time $t = (n-1)\tau$, and let $T > 0$. Then, there exists a constant $C > 0$ such that $\|\nabla\mathbf{u}^{(n-1)}\| \le C$ for all $(n-1)\tau < T$.*

Proof. We write the continuous evolution Equation (5.22) in vector form as

$$\partial_t \mathbf{u} = -\Delta\left(\varepsilon\Delta\mathbf{u} - \frac{1}{\varepsilon}\psi'(\mathbf{u}) + \frac{1}{\varepsilon N}\mathbf{1}\sum_{j=1}^{N}\frac{\partial\psi}{\partial u_j}(\mathbf{u})\right) + \omega(\mathbf{f}-\mathbf{u}). \tag{5.40}$$

Let

$$\mathbf{w} = [w_1,\dots,w_N]^T = -\varepsilon\Delta\mathbf{u} + \frac{1}{\varepsilon}\frac{\partial\psi}{\partial\mathbf{u}}(\mathbf{u}) - \frac{1}{\varepsilon N}\mathbf{1}\sum_{j=1}^{N}\frac{\partial\psi}{\partial u_j}(\mathbf{u}).$$

We multiply (5.40) with $\mathbf{w}$, integrate over Ω, and obtain

$$(\partial_t\mathbf{u},\mathbf{w}) = (\Delta\mathbf{w},\mathbf{w}) + (\omega(\mathbf{f}-\mathbf{u}),\mathbf{w}). \tag{5.41}$$

Differentiating the Ginzburg–Landau energy functional (5.3) over t leads to

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{E}(\mathbf{u}) &= \int_\Omega \sum_{i=1}^{N}\frac{1}{\varepsilon}\frac{\partial\psi(\mathbf{u})}{\partial u_i}\frac{\partial u_i}{\partial t} + \varepsilon\nabla u_i\cdot\nabla\frac{\partial u_i}{\partial t}\,\mathrm{d}\mathbf{x}\\
&= \int_\Omega \sum_{i=1}^{N}\left(\frac{1}{\varepsilon}\frac{\partial\psi}{\partial u_i}(\mathbf{u}) - \varepsilon\Delta u_i\right)\frac{\partial u_i}{\partial t}\,\mathrm{d}\mathbf{x}\\
&= \int_\Omega \sum_{i=1}^{N}\left(w_i + \frac{1}{\varepsilon N}\sum_{j=1}^{N}\frac{\partial\psi}{\partial u_j}(\mathbf{u})\right)\frac{\partial u_i}{\partial t}\,\mathrm{d}\mathbf{x}\\
&= (\partial_t\mathbf{u},\mathbf{w}) + \left(\partial_t\mathbf{u}, \frac{1}{\varepsilon N}\mathbf{1}\sum_{j=1}^{N}\frac{\partial\psi}{\partial u_j}(\mathbf{u})\right)\\
&\overset{(5.41)}{=} (\Delta\mathbf{w},\mathbf{w}) + (\omega(\mathbf{f}-\mathbf{u}),\mathbf{w}) + \left(\partial_t\mathbf{u}, \frac{1}{\varepsilon N}\mathbf{1}\sum_{j=1}^{N}\frac{\partial\psi}{\partial u_j}(\mathbf{u})\right)\\
&= -(\nabla\mathbf{w},\nabla\mathbf{w}) + (\omega(\mathbf{f}-\mathbf{u}),-\varepsilon\Delta\mathbf{u}) + \left(\omega(\mathbf{f}-\mathbf{u}),\frac{1}{\varepsilon}\psi'(\mathbf{u})\right)\\
&\quad + \left(-\omega(\mathbf{f}-\mathbf{u}) + \partial_t\mathbf{u}, \frac{1}{\varepsilon N}\mathbf{1}\sum_{j=1}^{N}\frac{\partial\psi}{\partial u_j}(\mathbf{u})\right),
\end{aligned} \tag{5.42}$$

where we have used Lemma 2.21 with zero Neumann boundary conditions in order to obtain the second and last equality. Since $\psi(\mathbf{u})$ is bounded from below, we only have to show that $\mathcal{E}(\mathbf{u})$ is uniformly bounded on $[0,T]$, and we automatically have that $|\nabla\mathbf{u}|$ is uniformly bounded on $[0,T]$. The last term on the right-hand side of (5.42) is zero since

$$
\begin{aligned}
&\left(-\omega(\mathbf{f}-\mathbf{u})+\partial_t\mathbf{u}, \frac{1}{\varepsilon N}\mathbf{1}\sum_{j=1}^{N}\frac{\partial\psi}{\partial u_j}(\mathbf{u})\right)\\
&\qquad = \frac{1}{\varepsilon N}\sum_{i=1}^{N}\int_\Omega(-\omega(f_i-u_i)+\partial_t u_i)\sum_{j=1}^{N}\left(\frac{\partial\psi}{\partial u_j}(\mathbf{u})\right)\mathrm{d}\mathbf{x}\\
&\qquad = \frac{1}{\varepsilon N}\int_\Omega\sum_{j=1}^{N}\left(\frac{\partial\psi}{\partial u_j}(\mathbf{u})\right)\left[-\omega\sum_{i=1}^{N}f_i+\omega\sum_{i=1}^{N}u_i+\partial_t\left(\sum_{i=1}^{N}u_i\right)\right]\mathrm{d}\mathbf{x}\\
&\qquad \overset{(5.2)}{=} 0.
\end{aligned}
$$

Hence, we end up with the same expression of $\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{E}(\mathbf{u})$ in (5.42) as in the last equation of [139, p. 433]. Therefore, the rest of the proof follows the proof of [139, Lemma 3.3], and, by integrating $\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{E}(\mathbf{u})$ in (5.42) over $[0,T]$, we finally end up with

$$
\begin{aligned}
\mathcal{E}(\mathbf{u}(t)) \le \mathcal{E}(\mathbf{u}(0)) + C(T) + T\cdot C(\omega_0,\varepsilon,\delta,\Omega,D,\mathbf{f})&\\
-\int_0^T\left[\int_\Omega|\nabla\mathbf{w}|^2\,\mathrm{d}\mathbf{x}+\frac{\omega_0 C_1}{\varepsilon}(1-\delta C(\mathbf{f},\Omega))\int_{\Omega\backslash D}\mathbf{u}^2\,\mathrm{d}\mathbf{x}\right]\mathrm{d}t,&
\end{aligned}
$$

for each $0\le t\le T$ for a positive constant C_1 and $\delta<\frac{1}{C(\mathbf{f},\Omega)}$. Hence, for a fixed $T>0$, $|\nabla\mathbf{u}|$ is uniformly bounded in $[0,T]$.

□

Finally, we will proof the convergence stated in Theorem 5.2.

Proof of convergence. We multiply (5.38) with $-\Delta e_i^{(n)}$ and integrate over Ω. Thereby, we use

$$
\int_{\partial\Omega}\Delta e_i^{(n)}\nabla\Delta^{-1}\eta_1^{(n-1)}\cdot\mathbf{n}\ \mathrm{d}\mathbf{s}=0
$$

as used in the first equation of [139, p. 431]. Altogether, we obtain

$$
\begin{aligned}
&\frac{1}{\tau}\left(\|\nabla e_i^{(n)}\|^2-\left(\nabla e_i^{(n-1)},\nabla e_i^{(n)}\right)\right)+\varepsilon\|\nabla\Delta e_i^{(n)}\|^2+C_1\|\Delta e_i^{(n)}\|^2+C_2\|\nabla e_i^{(n)}\|^2\\
&=-\frac{1}{\varepsilon}\left(\nabla\left(\frac{\partial\psi}{\partial u_i}(\mathbf{U}^{(n-1)})-\frac{\partial\psi}{\partial u_i}(\mathbf{u}^{(n-1)})\right),\nabla\Delta e_i^{(n)}\right)\\
&\quad+\frac{1}{\varepsilon N}\sum_{j=1}^{N}\left(\nabla\left(\frac{\partial\psi}{\partial u_j}(\mathbf{U}^{(n-1)})-\frac{\partial\psi}{\partial u_j}(\mathbf{u}^{(n-1)})\right),\nabla\Delta e_i^{(n)}\right)\\
&\quad-C_1\left(\Delta(U_i^{(n-1)}-u_i^{(n-1)}),\Delta e_i^{(n)}\right)-C_2\left(\nabla(U_i^{(n-1)}-u_i^{(n-1)}),\nabla e_i^{(n)}\right)\\
&\quad+\left(\nabla\omega(U_i^{(n-1)}-u_i^{(n-1)}),\nabla e_i^{(n)}\right)+\left(\nabla\Delta^{-1}\eta_i^{(n-1)},\nabla\Delta e_i^{(n)}\right).
\end{aligned}
$$

Using the definition of the discretization error as well as the estimation

$$
\frac{1}{\tau}\left(\|\nabla e_i^{(n)}\|^2-\left(\nabla e_i^{(n-1)},\nabla e_i^{(n)}\right)\right)\ge\frac{1}{2\tau}\left(\|\nabla e_i^{(n)}\|^2-\|\nabla e_i^{(n-1)}\|^2\right),
$$

as stated in the first inequality of [139, p. 431], we obtain

$$\begin{aligned}
&\frac{1}{2\tau}\left(\|\nabla e_i^{(n)}\|^2 - \|\nabla e_i^{(n-1)}\|^2\right) + \varepsilon\|\nabla\Delta e_i^{(n)}\|^2 + C_1\|\Delta e_i^{(n)}\|^2 + C_2\|\nabla e_i^{(n)}\|^2 \\
&\le -\frac{1}{\varepsilon}\left(\frac{\partial^2\psi}{\partial u_i^2}(\mathbf{U}^{(n-1)})\nabla U_i^{(n-1)} - \frac{\partial^2\psi}{\partial u_i^2}(\mathbf{u}^{(n-1)})\nabla u_i^{(n-1)}, \nabla\Delta e_i^{(n)}\right) \\
&\quad + \frac{1}{\varepsilon N}\sum_{j=1}^{N}\left(\frac{\partial^2\psi}{\partial u_j^2}(\mathbf{U}^{(n-1)})\nabla U_j^{(n-1)} - \frac{\partial^2\psi}{\partial u_j^2}(\mathbf{u}^{(n-1)})\nabla u_j^{(n-1)}, \nabla\Delta e_i^{(n)}\right) \\
&\quad + C_1\left(\Delta e_i^{(n-1)}, \Delta e_i^{(n)}\right) + C_2\left(\nabla e_i^{(n-1)}, \nabla e_i^{(n)}\right) - \left(\nabla\omega\, e_i^{(n-1)}, \nabla e_i^{(n)}\right) + \left(\nabla\Delta^{-1}\eta_i^{(n-1)}, \nabla\Delta e_i^{(n)}\right).
\end{aligned}$$

Using Young's inequality (Lemma 2.12)

$$\begin{aligned}
\left(\Delta e_i^{(n-1)}, \Delta e_i^{(n)}\right) &\le \frac{\delta_1}{2}\|\Delta e_i^{(n)}\|^2 + \frac{1}{2\delta_1}\|\Delta e_i^{(n-1)}\|^2, \\
\left(\nabla e_i^{(n-1)}, \nabla e_i^{(n)}\right) &\le \frac{\delta_2}{2}\|\nabla e_i^{(n)}\|^2 + \frac{1}{2\delta_2}\|\nabla e_i^{(n-1)}\|^2, \\
\left(\nabla\omega\, e_i^{(n-1)}, \nabla e_i^{(n)}\right) &\le \frac{\delta_3}{2}\|\nabla e_i^{(n)}\|^2 + \frac{\omega_0^2}{2\delta_3}\|\nabla e_i^{(n-1)}\|^2, \\
\left(\nabla\Delta^{-1}\eta_i^{(n-1)}, \nabla\Delta e_i^{(n)}\right) &\le \frac{\delta_4}{2}\|\nabla\Delta e_i^{(n)}\|^2 + \frac{1}{2\delta_4}\|\eta_i^{(n-1)}\|_{-1}^2,
\end{aligned}$$

where $\delta_1, \delta_2, \delta_3, \delta_4 > 0$, we get

$$\begin{aligned}
&\frac{1}{2\tau}\left(\|\nabla e_i^{(n)}\|^2 - \|\nabla e_i^{(n-1)}\|^2\right) + \varepsilon\|\nabla\Delta e_i^{(n)}\|^2 + C_1\|\Delta e_i^{(n)}\|^2 + C_2\|\nabla e_i^{(n)}\|^2 \\
&\le -\frac{1}{\varepsilon}\left(\frac{\partial^2\psi}{\partial u_i^2}(\mathbf{U}^{(n-1)})\nabla U_i^{(n-1)} - \frac{\partial^2\psi}{\partial u_i^2}(\mathbf{u}^{(n-1)})\nabla u_i^{(n-1)}, \nabla\Delta e_i^{(n)}\right) \\
&\quad + \frac{C_1\delta_1}{2}\|\Delta e_i^{(n)}\|^2 + \frac{C_1}{2\delta_1}\|\Delta e_i^{(n-1)}\|^2 \\
&\quad + \frac{1}{\varepsilon N}\sum_{j=1}^{N}\left(\frac{\partial^2\psi}{\partial u_j^2}(\mathbf{U}^{(n-1)})\nabla U_j^{(n-1)} - \frac{\partial^2\psi}{\partial u_j^2}(\mathbf{u}^{(n-1)})\nabla u_j^{(n-1)}, \nabla\Delta e_i^{(n)}\right) \\
&\quad + \frac{C_2\delta_2}{2}\|\nabla e_i^{(n)}\|^2 + \frac{C_2}{2\delta_2}\|\nabla e_i^{(n-1)}\|^2 \\
&\quad + \frac{\delta_3}{2}\|\nabla e_i^{(n)}\|^2 + \frac{\omega_0^2}{2\delta_3}\|\nabla e_i^{(n-1)}\|^2 + \frac{\delta_4}{2}\|\nabla\Delta e_i^{(n)}\|^2 + \frac{1}{2\delta_4}\|\eta_i^{(n-1)}\|_{-1}^2.
\end{aligned} \tag{5.43}$$

We can estimate the first term on the right-hand side of (5.43) in the same way as in the last inequality of [139, p. 431]:

$$\begin{aligned}
&-\frac{1}{\varepsilon}\left(\frac{\partial^2\psi}{\partial u_i^2}(\mathbf{U}^{(n-1)})\nabla U_i^{(n-1)} - \frac{\partial^2\psi}{\partial u_i^2}(\mathbf{u}^{(n-1)})\nabla u_i^{(n-1)}, \nabla\Delta e_i^{(n)}\right) \\
&\qquad\qquad \le \frac{C}{2\delta_5\varepsilon}\|\nabla e_i^{(n-1)}\|^2 + \frac{C}{2\delta_6\varepsilon}\|e_i^{(n-1)}\|^2 + \left(\frac{\delta_5}{2\varepsilon} + \frac{\delta_6}{2\varepsilon}\right)\|\nabla\Delta e_i^{(n)}\|^2.
\end{aligned} \tag{5.44}$$

For that, we have used (5.28), the local Lipschitz continuity of $\frac{\partial^2\psi}{\partial u_i^2}$, and Lemma 5.4. We have set C to be a universal constant for all bounds. Using the estimation

$$\|e_i^{(n-1)}\|^2 \le 2\|e_i^{(n-1)} - O(\tau)^2\|^2 + 2\|O(\tau)^2\|^2, \tag{5.45}$$

as stated in the first inequality of [139, p. 432], and Lemma 5.3, we can apply the Poincaré inequality (Theorem 2.22)

$$\|e_i^{(n-1)} - O(\tau)^2\|^2 \le c_{\mathrm{P}}\, \|\nabla e_i^{(n-1)}\|^2$$

to (5.45):

$$\|e_i^{(n-1)}\|^2 \le 2c_{\mathrm{P}}\, \|\nabla e_i^{(n-1)}\|^2 + 2\|O(\tau)^2\|^2. \tag{5.46}$$

Substituting (5.46) into (5.44), we get

$$\begin{aligned}
&-\frac{1}{\varepsilon}\left(\frac{\partial^2\psi}{\partial u_i^2}(\mathbf{U}^{(n-1)})\nabla U_i^{(n-1)} - \frac{\partial^2\psi}{\partial u_i^2}(\mathbf{u}^{(n-1)})\nabla u_i^{(n-1)}, \nabla\Delta e_i^{(n)}\right)\\
&\qquad\le \frac{C}{2\delta_5\varepsilon}\|\nabla e_i^{(n-1)}\|^2 + \frac{C}{\delta_6\varepsilon}\|\nabla e_i^{(n-1)}\|^2 + \frac{C}{\delta_6\varepsilon}\|O(\tau)^2\|^2 + \left(\frac{\delta_5}{2\varepsilon} + \frac{\delta_6}{2\varepsilon}\right)\|\nabla\Delta e_i^{(n)}\|^2.
\end{aligned}$$

In the same way, we can estimate the fourth term on the right-hand side of (5.43). Altogether, we get in (5.43)

$$\begin{aligned}
&\left(\frac{1}{2\tau} + C_2\left(1 - \frac{\delta_2}{2}\right) - \frac{\delta_3}{2}\right)\|\nabla e_i^{(n)}\|^2 + C_1\left(1 - \frac{\delta_1}{2}\right)\|\Delta e_i^{(n)}\|^2 + \left(\varepsilon - \frac{\delta_4}{2} - \frac{\delta_5+\delta_6}{2\varepsilon}\right)\|\nabla\Delta e_i^{(n)}\|^2\\
&\quad\le \left(\frac{1}{2\tau} + \frac{\omega_0^2}{2\delta_3} + \frac{C_2}{2\delta_2} + \frac{C}{2\delta_5\varepsilon} + \frac{C}{\delta_6\varepsilon}\right)\|\nabla e_i^{(n-1)}\|^2 + \frac{C_1}{2\delta_1}\|\Delta e_i^{(n-1)}\|^2 + \frac{1}{2\delta_4}\|\eta_i^{(n-1)}\|_{-1}^2\\
&\quad+ \frac{C}{\delta_6\varepsilon}\|O(\tau)^2\|^2 + \frac{1}{\varepsilon N}\sum_{j=1}^{N}\left[\left(\frac{\delta_5+\delta_6}{2}\right)\|\nabla\Delta e_i^{(n)}\|^2 + \left(\frac{C}{2\delta_5} + \frac{C}{\delta_6}\right)\|\nabla e_j^{(n-1)}\|^2 + \frac{C}{\delta_6}\|O(\tau)^2\|^2\right].
\end{aligned}$$

Multiplying the inequality with 2τ, choosing $\delta_1 = 1$ and taking the sum over $i = 1, \dots, N$ yields

$$\begin{aligned}
&(1 + \tau\left[C_2\left(2 - \delta_2\right) - \delta_3\right])\,\|\nabla\mathbf{e}^{(n)}\|^2 + \tau C_1\|\Delta\mathbf{e}^{(n)}\|^2 + \tau\left(2\varepsilon - \delta_4 - \frac{2(\delta_5+\delta_6)}{\varepsilon}\right)\|\nabla\Delta\mathbf{e}^{(n)}\|^2\\
&\qquad\le \left(1 + \tau\left[\frac{\omega_0^2}{\delta_3} + \frac{C_2}{\delta_2} + \frac{2C}{\delta_5\varepsilon} + \frac{4C}{\delta_6\varepsilon}\right]\right)\|\nabla\mathbf{e}^{(n-1)}\|^2\\
&\qquad\qquad + \tau C_1\|\Delta\mathbf{e}^{(n-1)}\|^2 + \frac{\tau}{\delta_4}\|\boldsymbol{\eta}^{(n-1)}\|_{-1}^2 + \tau\frac{4C}{\delta_6\varepsilon}\|O(\tau)^2\|^2.
\end{aligned}$$

This is (almost) the same estimation as the second inequality in [139, p. 432]. Therefore, the rest of the proof follows the proof of [139, Proposition 3.4]. This results in the desired estimation (5.30). □

As in the previous two chapters, the approach of the truncated polynomial is only used for the theoretical part. In praxis, the polynomial potential ψ_{pol} behaves quite well and does not result in blow ups of the solution. Violations of $u \in [0,1]$ in form of $u \in [-\delta, 1+\delta]$ occur. However, δ is relatively small. We investigate this issue further in Section 5.7.3.

For the next steps, i.e., the discretization in space in Section 5.5, we consider the weak formulation of the smooth system in (5.26)–(5.27), which is split as in (5.12)–(5.14):

Find $\mathbf{U}^{(n)}, \mathbf{W}^{(n)} \in H^1(\Omega)^N$ with $\mathbf{U}^{(n)} = [U_1^{(n)}, \dots, U_N^{(n)}]^T$ and $\mathbf{W}^{(n)} = [W_1^{(n)}, \dots, W_N^{(n)}]^T$ such that

$$\begin{aligned} \left(\frac{1}{\tau} + C_2\right)(U_i^{(n)}, v) + (\nabla W_i^{(n)}, \nabla v) - (\omega(f_i - U_i^{(n-1)}), v) \\ - \left(\frac{1}{\tau} + C_2\right)(U_i^{(n-1)}, v) = 0 \quad \forall v \in H^1(\Omega), \quad (5.47) \\ (W_i^{(n)}, v) - \varepsilon(\nabla U_i^{(n)}, \nabla v) - C_1(U_i^{(n)}, v) - \frac{1}{\varepsilon}\left(\psi'(U_i^{(n-1)}), v\right) \\ + \frac{1}{\varepsilon N}\sum_{j=1}^{N}\left(\psi'(U_j^{(n-1)}), v\right) + C_1(U_i^{(n-1)}, v) = 0 \quad \forall v \in H^1(\Omega), \quad (5.48) \end{aligned}$$

for $i = 1, \dots, N$. The system (5.47)–(5.48) is supplemented by the initial condition $\mathbf{u}_0 \in H^1(\Omega)^N, \mathbf{u}_0 \in \mathcal{G}^N$ a.e. in Ω.

Remark 5.4. *Another possible weak formulation is:*

$$\begin{aligned} \left(\frac{1}{\tau} + C_2\right)(U_i^{(n)}, v) + (\nabla W_i^{(n)}, \nabla v) + C_1(\nabla U_i^{(n)}, \nabla v) \\ - (\omega(f_i - U_i^{(n-1)}), v) - C_1(\nabla U_i^{(n-1)}, \nabla v) - \left(\frac{1}{\tau} + C_2\right)(U_i^{(n-1)}, v) = 0 \quad \forall v \in H^1(\Omega), \\ (W_i^{(n)}, v) - \varepsilon(\nabla U_i^{(n)}, \nabla v) - \frac{1}{\varepsilon}\left(\psi'(U_i^{(n-1)}), v\right) + \frac{1}{\varepsilon N}\sum_{j=1}^{N}\left(\psi'(U_j^{(n-1)}), v\right) = 0 \quad \forall v \in H^1(\Omega). \end{aligned}$$

Such a formulation is considered in [31] for black-and-white Cahn–Hilliard inpainting. However, working with (5.47)–(5.48) results in systems of linear equations which comply with the ones from the previous chapters.

After having stated and analyzed our system of time-discrete Cahn–Hilliard equations in the smooth setting, we proceed to the nonsmooth case.

5.3.2 Nonsmooth systems

In the following, we concentrate on the nonsmooth setting and the fourth-order formulation of (5.19)–(5.21):

$$\begin{aligned} \partial_t u_i = -\Delta\Bigg(\varepsilon\Delta u_i + \frac{1}{\varepsilon}(\boldsymbol{T}\mathbf{u})_i - \frac{1}{c}\min(0, u_i) \\ -\frac{1}{N}\sum_{j=1}^{N}\left[-\frac{1}{c}\min(0, u_j) + \frac{1}{\varepsilon}(\boldsymbol{T}\mathbf{u})_j\right]\Bigg) + \omega(f_i - u_i), \quad (5.49) \\ \nabla u_i \cdot \mathbf{n} = \nabla(\Delta u_i) \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega, \quad (5.50) \end{aligned}$$

for $i = 1, \dots, N$. In the following, we apply the convexity splitting scheme to the nonsmooth vector-valued Cahn–Hilliard inpainting model (5.49)–(5.50). As in the smooth case, the nonsmooth vector-valued modified Cahn–Hilliard equation as a whole is not given by a gradient flow. Especially, our proposed model arises as a superposition of the H^{-1}-gradient flow for $\mathcal{E}$ in (5.15) under the constraint (5.1) and the L^2-gradient flow for $\mathcal{F}$ in (5.7). In this case, convexity splitting is applied to each of these energies. To be more precise, we split $\mathcal{E}$ as $\mathcal{E} = \mathcal{E}_c - \mathcal{E}_e$, where

$$\mathcal{E}_c(\mathbf{u}) = \int_\Omega \frac{\varepsilon}{2}\sum_{i=1}^N |\nabla u_i|^2 + \frac{C_1}{2}\sum_{i=1}^N u_i^2 + \frac{1}{2c}\sum_{i=1}^N \min(0,u_i)^2 \,\mathrm{d}\mathbf{x},$$
$$\mathcal{E}_e(\mathbf{u}) = \int_\Omega -\frac{1}{\varepsilon}\psi_0(\mathbf{u}) + \frac{C_1}{2}\sum_{i=1}^N u_i^2 \,\mathrm{d}\mathbf{x} = \int_\Omega \frac{1}{2\varepsilon}\mathbf{u}\cdot\boldsymbol{T}\mathbf{u} + \frac{C_1}{2}\sum_{i=1}^N u_i^2 \,\mathrm{d}\mathbf{x}.$$

We use the same splitting for $\mathcal{F}$ as in the previous section, i.e., $\mathcal{F} = \mathcal{F}_c - \mathcal{F}_e$, where

$$\mathcal{F}_c(\mathbf{u}) = \int_\Omega \frac{C_2}{2}\sum_{i=1}^N u_i^2 \,\mathrm{d}\mathbf{x},$$
$$\mathcal{F}_e(\mathbf{u}) = \int_\Omega -\frac{\omega}{2}\sum_{i=1}^N (f_i - u_i)^2 + \frac{C_2}{2}\sum_{i=1}^N u_i^2 \,\mathrm{d}\mathbf{x}.$$

The constants C_1 and C_2 are positive and need to be chosen large enough such that the energies $\mathcal{E}_c$, $\mathcal{E}_e$, $\mathcal{F}_c$, and $\mathcal{F}_e$ are strictly convex. In the last section, we have already proven the convexity condition for $\mathcal{F}$, which is given as $C_2 > \omega_0$. It is easy to see that $\mathcal{E}_c$ is already strictly convex[2] for $C_1 > 0$. The crucial point is the energy functional that contains the potential function $\psi_0(\mathbf{u})$.

Lemma 5.5. *$\mathcal{E}_e$ is strictly convex if $C_1 > \max\left(0, -\lambda_{\min}(\boldsymbol{T})\varepsilon^{-1}\right)$, where $\lambda_{\min}(\boldsymbol{T})$ is the smallest eigenvalue of $\boldsymbol{T}$.*

Proof. Let $\mathbf{u} = [u_1,\dots,u_N]^T \in H^1(\Omega)^N$ and $\mathbf{v} = [v_1,\dots,v_N]^T \in H^1(\Omega)^N$. Based on [145, p. 54], we have to show

$$\mathcal{E}_e(\mathbf{u}+\mathbf{v}) - \mathcal{E}_e(\mathbf{u}) \geq \lim_{\delta\to 0}\frac{\mathcal{E}_e(\mathbf{u}+\delta\mathbf{v}) - \mathcal{E}_e(\mathbf{u})}{\delta}.$$

We have

$$\begin{aligned}\mathcal{E}_e(\mathbf{u}+\mathbf{v}) - \mathcal{E}_e(\mathbf{u}) &= \int_\Omega \frac{1}{2\varepsilon}(\mathbf{u}+\mathbf{v})\cdot\boldsymbol{T}(\mathbf{u}+\mathbf{v}) - \frac{1}{2\varepsilon}\mathbf{u}\cdot\boldsymbol{T}\mathbf{u}\,\mathrm{d}\mathbf{x}\\ &\quad + \int_\Omega \frac{C_1}{2}\sum_{i=1}^N\left((u_i+v_i)^2 - u_i^2\right)\mathrm{d}\mathbf{x}\\ &= \int_\Omega \frac{1}{2\varepsilon}\left(\mathbf{v}\cdot\boldsymbol{T}\mathbf{v} + 2\mathbf{u}\cdot\boldsymbol{T}\mathbf{v}\right) + \frac{C_1}{2}\sum_{i=1}^N\left(v_i^2 + 2u_iv_i\right)\mathrm{d}\mathbf{x}. \qquad (5.51)\end{aligned}$$

Similarly, one can show

$$\lim_{\delta\to 0}\frac{\mathcal{E}_e(\mathbf{u}+\delta\mathbf{v}) - \mathcal{E}_e(\mathbf{u})}{\delta} = \int_\Omega \frac{1}{\varepsilon}\mathbf{u}\cdot\boldsymbol{T}\mathbf{v} + \frac{C_1}{2}\sum_{i=1}^N 2u_iv_i\,\mathrm{d}\mathbf{x},$$

[2] As stated in the proof of Theorem 4.5, the functionals $u_i \to \|\min(0,u_i)\|^2$, $i = 1,\dots,N$, are convex and Fréchet-differentiable on $H^1(\Omega)$.

which leads to

$$\begin{aligned}\mathcal{E}_e(\mathbf{u}+\mathbf{v}) - \mathcal{E}_e(\mathbf{u}) - \lim_{\delta\to 0}\frac{\mathcal{E}_e(\mathbf{u}+\delta\mathbf{v}) - \mathcal{E}_e(\mathbf{u})}{\delta} &= \int_\Omega \frac{1}{2\varepsilon}\mathbf{v}\cdot\boldsymbol{T}\mathbf{v} + \frac{C_1}{2}\sum_{i=1}^N v_i^2 \,\mathrm{d}\mathbf{x}\\ &\geq \int_\Omega \left(\frac{\lambda_{\min}(\boldsymbol{T})}{2\varepsilon} + \frac{C_1}{2}\right)\sum_{i=1}^N v_i^2 \,\mathrm{d}\mathbf{x}.\end{aligned}$$

Therefore, $\mathcal{E}_e$ is strictly convex if $C_1 > \max\left(0, -\lambda_{\min}(\boldsymbol{T})\varepsilon^{-1}\right)$. □

Remark 5.5. *In the corresponding nonsmooth black-and-white Cahn–Hilliard inpainting model, the convexity requirements are $C_1 > 0$ and $C_2 > \omega_0$. Note that we even do not need the C_1-term in $\mathcal{E}_c$ and $\mathcal{E}_e$. Both functionals are already convex without the additional C_1-term, see [31, p. 7].*

The resulting time-discrete scheme is given by

$$\frac{\mathbf{U}^{(n)} - \mathbf{U}^{(n-1)}}{\tau} = -\mathrm{grad}_{H^{-1}}^{(5.1)}\left(\mathcal{E}_c(\mathbf{U}^{(n)}) - \mathcal{E}_e(\mathbf{U}^{(n-1)})\right) - \mathrm{grad}_{L^2}\left(\mathcal{F}_c(\mathbf{U}^{(n)}) - \mathcal{F}_e(\mathbf{U}^{(n-1)})\right).$$

This translates to a numerical scheme of the form

$$\begin{aligned}\frac{U_i^{(n)} - U_i^{(n-1)}}{\tau} + \varepsilon\Delta^2 U_i^{(n)} - C_1\Delta U_i^{(n)} + C_2 U_i^{(n)} - \frac{1}{c}\Delta\min(0, U_i^{(n)}) + \frac{1}{cN}\Delta\left(\sum_{j=1}^N \min(0, U_j^{(n)})\right)&\\ = -\frac{1}{\varepsilon}\Delta(\boldsymbol{T}\mathbf{U}^{(n-1)})_i + \frac{1}{\varepsilon N}\Delta\left(\sum_{j=1}^N (\boldsymbol{T}\mathbf{U}^{(n-1)})_j\right)& \qquad (5.52)\\ + \omega(f_i - U_i^{(n-1)}) - C_1\Delta U_i^{(n-1)} + C_2 U_i^{(n-1)},&\\ \nabla U_i\cdot\mathbf{n} = \nabla(\Delta U_i)\cdot\mathbf{n} = 0 \quad \text{on } \partial\Omega,& \qquad (5.53)\end{aligned}$$

for $i = 1, \dots, N$.

Remark 5.6. *A rigorous analysis, as done in the previous chapter, is a topic of future research.*

Now, we have arrived at a system of linear equations in the smooth case and a system of nonlinear equations in the regularized nonsmooth case. In order to solve the latter system, we have to pay attention to the minimum operator present in (5.52). However, as in Chapter 4.5, we can solve the corresponding nonlinear system via the SSN method. This will be the topic of the following chapter.

5.4 Semismooth Newton method

For the sake of clarity and consistency with the previous two chapters, in the following, we denote by $\mathbf{u}^{(n-1)} \in H^1(\Omega)^N$ the time-discrete solution at time step t_{n-1}. Further, $\mathbf{u}^{(n)}, \mathbf{w}^{(n)} \in H^1(\Omega)^N$ form the time-discrete solution at time step $t_n = t_{n-1} + \tau$. Moreover, from now on we write $\mathbf{u}^{\text{old}}$, $\mathbf{u}$, and $\mathbf{w}$ instead of $\mathbf{u}^{(n-1)}$, $\mathbf{u}^{(n)}$, and $\mathbf{w}^{(n)}$, respectively. In what follows, we consider the weak formulation of (5.52)–(5.53) and split it as in (5.19)–(5.21): Find $\mathbf{u}_c, \mathbf{w}_c \in H^1(\Omega)^N$ with $\mathbf{u}_c = [u_{c,1}, \ldots, u_{c,N}]^T$ and $\mathbf{w}_c = [w_{c,1}, \ldots, w_{c,N}]^T$ such that

$$\begin{aligned}
&\left(\frac{1}{\tau} + C_2\right)(u_{c,i}, v) + (\nabla w_{c,i}, \nabla v) \\
&\qquad - (\omega(f_i - u_i^{\text{old}}), v) - \left(\frac{1}{\tau} + C_2\right)(u_i^{\text{old}}, v) = 0 \quad \forall v \in H^1(\Omega), \qquad (5.54)\\
&(w_{c,i}, v) - \varepsilon(\nabla u_{c,i}, \nabla v) - C_1(u_{c,i}, v) + \frac{1}{\varepsilon}((\boldsymbol{T}\mathbf{u}^{\text{old}})_i, v) \\
&\quad - \frac{1}{c}(\min(0, u_{c,i}), v) + C_1(u_i^{\text{old}}, v) \\
&\quad + \frac{1}{N}\sum_{j=1}^{N}\left[\frac{1}{c}(\min(0, u_{c,j}), v) - \frac{1}{\varepsilon}((\boldsymbol{T}\mathbf{u}^{\text{old}})_j, v)\right] = 0 \quad \forall v \in H^1(\Omega), \qquad (5.55)
\end{aligned}$$

for $i = 1, \ldots, N$.

Remark 5.7. *As pointed out in Remark 5.4, another possible weak formulation is:*

$$\begin{aligned}
&\left(\frac{1}{\tau} + C_2\right)(u_{c,i}, v) + (\nabla w_{c,i}, \nabla v) + C_1(\nabla u_{c,i}, \nabla v) \\
&\qquad - (\omega(f_i - u_i^{\text{old}}), v) - C_1(\nabla u_i^{\text{old}}, \nabla v) - \left(\frac{1}{\tau} + C_2\right)(u_i^{\text{old}}, v) = 0 \quad \forall v \in H^1(\Omega), \\
&(w_{c,i}, v) - \varepsilon(\nabla u_{c,i}, \nabla v) + \frac{1}{\varepsilon}((\boldsymbol{T}\mathbf{u}^{\text{old}})_i, v) - \frac{1}{c}(\min(0, u_{c,i}), v) \\
&\qquad + \frac{1}{N}\sum_{j=1}^{N}\left[\frac{1}{c}(\min(0, u_{c,j}), v) - \frac{1}{\varepsilon}((\boldsymbol{T}\mathbf{u}^{\text{old}})_j, v)\right] = 0 \quad \forall v \in H^1(\Omega),
\end{aligned}$$

for $i = 1, \ldots, N$. Such a formulation is considered in [31] for black-and-white Cahn–Hilliard inpainting.

For a specified sequence $c \to 0$, we solve the system (5.54)–(5.55), compactly written as

$$\mathbf{F}_c(\mathbf{u}_c, \mathbf{w}_c) = \left(\mathbf{F}_c^{(1)}(\mathbf{u}_c, \mathbf{w}_c), \mathbf{F}_c^{(2)}(\mathbf{u}_c, \mathbf{w}_c))\right) = 0, \qquad (5.56)$$

for every c by an SSN algorithm. In (5.56), the components are defined by

$$\begin{aligned}
\left\langle \mathbf{F}_c^{(1)}(\mathbf{u}, \mathbf{w}), \mathbf{v} \right\rangle &= (\nabla \mathbf{w}, \nabla \mathbf{v}) + \left(\frac{1}{\tau} + C_2\right)(\mathbf{u}, \mathbf{v}) - (\omega(\mathbf{f} - \mathbf{u}^{\text{old}}), \mathbf{v}) - \left(\frac{1}{\tau} + C_2\right)(\mathbf{u}^{\text{old}}, \mathbf{v}), \\
\left\langle \mathbf{F}_c^{(2)}(\mathbf{u}, \mathbf{w}), \mathbf{v} \right\rangle &= \varepsilon(\nabla \mathbf{u}, \nabla \mathbf{v}) + C_1(\mathbf{u}, \mathbf{v}) + \frac{1}{c}(\min(\mathbf{0}, \mathbf{u}), \mathbf{v}) - (\mathbf{w}, \mathbf{v}) - \frac{1}{\varepsilon}(\boldsymbol{T}\mathbf{u}^{\text{old}}, \mathbf{v}) \\
&\quad - C_1(\mathbf{u}^{\text{old}}, \mathbf{v}) - \frac{1}{N}\sum_{j=1}^{N}\left[\frac{1}{c}(\min(0, u_j)\mathbf{1}, \mathbf{v}) - \frac{1}{\varepsilon}((\boldsymbol{T}\mathbf{u}^{\text{old}})_j\mathbf{1}, \mathbf{v})\right],
\end{aligned}$$

for all $\mathbf{u}, \mathbf{w}, \mathbf{v} \in H^1(\Omega)^N$. $\mathbf{F}_c$ is not Fréchet-differentiable due to the presence of the minimum operator. However, the minimum operator satisfies the weaker notion of Newton differentiability, see Definition 2.11. As in Chapter 4.5, we can state a Newton derivative for $\mathbf{F}_c$:

Lemma 5.6. *The mapping* $\mathbf{F}_c \colon H^1(\Omega)^N \times H^1(\Omega)^N \to \left(H^1(\Omega)^N\right)^* \times \left(H^1(\Omega)^N\right)^*$ *is Newton-differentiable. Furthermore, the operator* $\mathbf{G}_c(\mathbf{u},\mathbf{w})$ *given by*

$$\begin{aligned}&\langle \mathbf{G}_c(\mathbf{u},\mathbf{w})(\delta\mathbf{u},\delta\mathbf{w}),(\boldsymbol{\phi},\boldsymbol{\psi})\rangle\\ &= \left(\begin{array}{c} (\nabla\delta\mathbf{w},\nabla\boldsymbol{\phi}) + \left(\frac{1}{\tau}+C_2\right)(\delta\mathbf{u},\boldsymbol{\phi}) \\ \varepsilon(\nabla\delta\mathbf{u},\nabla\boldsymbol{\psi}) + C_1(\delta\mathbf{u},\boldsymbol{\psi}) + \frac{1}{c}(\boldsymbol{\chi}_{\mathcal{M}(\mathbf{u})}\delta\mathbf{u},\boldsymbol{\psi}) - (\delta\mathbf{w},\boldsymbol{\psi}) - \frac{1}{cN}\sum_{j=1}^{N}(\chi_{\mathcal{M}(u_j)}\delta u_j\mathbf{1},\boldsymbol{\psi}) \end{array} \right),\end{aligned}$$

serves as a Newton derivative for $\mathbf{F}_c$*. Here,* $\chi_{\mathcal{M}(u_i)}$ *is the characteristic function of the set*

$$\mathcal{M}(u_i) := \{\mathbf{x}\in\Omega : u_i(\mathbf{x}) < 0\}.$$

The term $\boldsymbol{\chi}_{\mathcal{M}(\mathbf{u})}\delta\mathbf{u}$ *is given as*

$$\boldsymbol{\chi}_{\mathcal{M}(\mathbf{u})}\delta\mathbf{u} = \left[\chi_{\mathcal{M}(u_1)}\delta u_1,\ldots,\chi_{\mathcal{M}(u_N)}\delta u_N\right]^T.$$

For the proof, we refer to [91, p. 788] and [92, pp. 885-886].

In the next section, we derive the fully discrete problems for both, the smooth system in (5.47)-(5.48) and the regularized nonsmooth system in (5.54)–(5.55).

5.5 Finite element approximation

In this section, we apply FEM [144] to the regularized nonsmooth Cahn–Hilliard system in (5.54)–(5.55). We also want to apply it to the smooth version (5.47)-(5.48). Since both procedures are similar, we only present the methodology based on the nonsmooth setting. Regarding the smooth case, we will state the fully discrete linear system at the end of this section. Moreover, the following presentation complies with the FEM Section 4.6 for the vector-valued Cahn–Hilliard equation.

In the following, we assume for simplicity that Ω is a polyhedral domain. Let $\{\mathcal{R}_h\}_{h>0}$ be a triangulation of Ω into disjoint open rectangular elements with maximal element size h, J_h be the set of nodes of $\mathcal{R}_h$, and $p_j \in J_h$ be the coordinates of these nodes. We approximate the infinite-dimensional space $H^1(\Omega)$ by the finite-dimensional space

$$S_h := \{\phi\in C^0(\overline{\Omega}) : \phi\mid_R \in Q_1(R)\ \ \forall R\in\mathcal{R}_h\} \subset H^1(\Omega),$$

of continuous, piecewise multilinear functions. We denote the standard nodal basis functions of S_h by φ_j for all $j\in J_h$. They have the property $\varphi_j(p_i) = \delta_{ij}$, $i,j = 1,\ldots,m$. The discretized version of the penalized problem (5.54)–(5.55) is the following. Given $\mathbf{u}_h^{\text{old}} = [u_{h,1}^{\text{old}},\ldots,u_{h,N}^{\text{old}}]^T \in S_h^N$, find $(\mathbf{u}_{c,h},\mathbf{w}_{c,h}) \in S_h^N\times S_h^N$, where $\mathbf{u}_{c,h} = [u_{c,h,1},\ldots,u_{c,h,N}]^T$, $\mathbf{w}_{c,h} = [w_{c,h,1},\ldots,w_{c,h,N}]^T$ such that

$$\left\langle F_{c,h}^{(1,i)}(\mathbf{u}_{c,h},\mathbf{w}_{c,h}),v_h\right\rangle = 0 \quad \forall v_h\in S_h, i=1,\ldots,N, \tag{5.57}$$

$$\left\langle F_{c,h}^{(2,i)}(\mathbf{u}_{c,h},\mathbf{w}_{c,h}),v_h\right\rangle = 0 \quad \forall v_h\in S_h i=1,\ldots,N, \tag{5.58}$$

where the components are

$$\begin{aligned}\left\langle F^{(1,i)}_{c,h}(\mathbf{u}_{c,h},\mathbf{w}_{c,h}),v_h\right\rangle &= (\nabla w_{c,h,i},\nabla v_h)+\left(\frac{1}{\tau}+C_2\right)(u_{c,h,i},v_h)_h\\ &\qquad -(\omega(f_{h,i}-u^{\mathrm{old}}_{h,i}),v_h)_h-\left(\frac{1}{\tau}+C_2\right)(u^{\mathrm{old}}_{h,i},v_h)_h,\\ \left\langle F^{(2,i)}_{c,h}(\mathbf{u}_{c,h},\mathbf{w}_{c,h}),v_h\right\rangle &= \varepsilon(\nabla u_{c,h,i},\nabla v_h)+C_1(u_{c,h,i},v_h)+\frac{1}{c}(\min(0,u_{h,i}),v_h)_h\\ &\qquad -(w_{c,h,i},v_h)_h-\frac{1}{\varepsilon}((\boldsymbol{T}\mathbf{u}^{\mathrm{old}}_{c,h})_i,v_h)_h-C_1(u^{\mathrm{old}}_{h,i},v_h)\\ &\qquad -\frac{1}{N}\sum_{j=1}^{N}\left[\frac{1}{c}(\min(0,u_{c,h,j}),v_h)_h-\frac{1}{\varepsilon}((\boldsymbol{T}\mathbf{u}^{\mathrm{old}}_{h})_j,v_h)_h\right].\end{aligned}$$

Within our finite element framework, for a given $(\mathbf{u}_h,\mathbf{w}_h)\in S_h^N\times S_h^N$, every step of the SSN method for solving (5.57)–(5.58) requires to compute $(\delta\mathbf{u}_h,\delta\mathbf{w}_h)\in S_h^N\times S_h^N$ satisfying

$$\begin{aligned}(\nabla\delta w_{h,i},\nabla v_h)+\left(\frac{1}{\tau}+C_2\right)(\delta u_{h,i},v_h)_h &= -F^{(1,i)}_{c,h}(\mathbf{u}_h,\mathbf{w}_h),\\ \varepsilon(\nabla\delta u_{h,i},\nabla v_h)+C_1(\delta u_{h,i},v_h)+\frac{1}{c}(\chi^h_{\mathcal{M}(u_{h,i})}\delta u_{h,i},v_h)_h-(\delta w_{h,i},v_h)_h &\\ -\frac{1}{cN}\sum_{j=1}^{N}(\chi^h_{\mathcal{M}(u_{h,j})}\delta u_{h,j},v_h)_h &= -F^{(2,i)}_{c,h}(\mathbf{u}_h,\mathbf{w}_h),\end{aligned}$$

for all $v_h\in S_h$ and $i=1,\dots,N$. Here, $\mathbf{u}_h=[u_{h,1},\dots,u_{h,N}]^T$, $\mathbf{w}_h=[w_{h,1},\dots,w_{h,N}]^T$, and $\delta\mathbf{u}_h=[\delta u_{h,1},\dots,\delta u_{h,N}]^T$, $\delta\mathbf{w}_h=[\delta w_{h,1},\dots,\delta w_{h,N}]^T$. Further, we define $\chi^h_{\mathcal{M}(u_{h,i})}:=\sum_{j=1}^m\chi^h_{\mathcal{M}(u_{h,i})}(p_j)\,\varphi_j$ with $\chi^h_{\mathcal{M}(u_{h,i})}(p_j)=0$ if $u_{h,i}(p_j)\geq 0$ and $\chi^h_{\mathcal{M}(u_{h,i})}(p_j)=1$ otherwise. If we now write a function $v_h\in S_h$ by $v_h=\sum_{j\in J_h}v_{h,j}\,\varphi_j$ and denote the vector of coefficients by $\mathbf{v}$, the fully discrete linear systems (smooth and nonsmooth) read in matrix form as

$$\begin{aligned}&\begin{bmatrix}-\boldsymbol{A} & \boldsymbol{I}\otimes\boldsymbol{M}\\ \boldsymbol{I}\otimes\boldsymbol{M} & \frac{\tau}{1+\tau C_2}\boldsymbol{I}\otimes\boldsymbol{K}\end{bmatrix}\begin{bmatrix}\boldsymbol{u}^{(k+1)}\\ \boldsymbol{w}^{(k+1)}\end{bmatrix}\\ &=\begin{bmatrix}-C_1(\boldsymbol{I}\otimes\boldsymbol{M})\,\boldsymbol{u}^{\mathrm{old}}+\frac{1}{\varepsilon}(\boldsymbol{I}\otimes\boldsymbol{M})\psi'\left(\boldsymbol{u}^{\mathrm{old}}\right)-\frac{1}{\varepsilon N}(\boldsymbol{I}\otimes\boldsymbol{M})\left(\sum_{j=1}^N\psi'\left(\boldsymbol{u}^{\mathrm{old}}_j\right)\right)\mathbf{1}\\ (\boldsymbol{I}\otimes\boldsymbol{M})\,\boldsymbol{u}^{\mathrm{old}}+\frac{\tau\omega_0}{1+\tau C_2}(\boldsymbol{I}\otimes\boldsymbol{H})(\boldsymbol{f}-\boldsymbol{u}^{\mathrm{old}})\end{bmatrix}.\end{aligned}\tag{5.59}$$

Here, $\boldsymbol{u}^{(k+1)}=\left[\left(\boldsymbol{u}_1^{(k+1)}\right)^T,\dots,\left(\boldsymbol{u}_N^{(k+1)}\right)^T\right]^T$, $\boldsymbol{w}^{(k+1)}=\left[\left(\boldsymbol{w}_1^{(k+1)}\right)^T,\dots,\left(\boldsymbol{w}_N^{(k+1)}\right)^T\right]^T\in\mathbb{R}^{Nm}$ and $\boldsymbol{u}^{\mathrm{old}}=\left[\left(\boldsymbol{u}_1^{\mathrm{old}}\right)^T,\dots,\left(\boldsymbol{u}_N^{\mathrm{old}}\right)^T\right]^T\in\mathbb{R}^{Nm}$ is the solution vector from the previous time step. Remember that k denotes the iteration step of the SSN method. Moreover, $\boldsymbol{u}^{(k)}=\left[\left(\boldsymbol{u}_1^{(k)}\right)^T,\dots,\left(\boldsymbol{u}_N^{(k)}\right)^T\right]^T$, $\mathbf{1}=[1,\dots,1]^T\in\mathbb{R}^{Nm}$. Note that we do not have to apply a Newton iteration for solving the smooth system since there is no nonlinearity. Hence, in this case, the superindex $k+1$ in (5.59) vanishes. The potential ψ in first right-hand side is $\psi=\psi_{\mathrm{pol}}$ in the smooth case and $\psi=\psi_0$ in the nonsmooth case. As in the previous two chapters, $\boldsymbol{M}\in\mathbb{R}^{m\times m}$ is the lumped mass matrix, and $\boldsymbol{K}\in\mathbb{R}^{m\times m}$ is the stiffness matrix. Remember that $\boldsymbol{M}$ is a diagonal, symmetric positive definite matrix, and $\boldsymbol{K}$ is symmetric positive semidefinite. Moreover, $\boldsymbol{I} \in \mathbb{R}^{N\times N}$ is the identity matrix. The matrix representation coming from the fidelity term is the diagonal matrix

$$\boldsymbol{H} = \boldsymbol{H}_D = \operatorname{diag}\begin{pmatrix} [\boldsymbol{M}]_{ii} & \text{if } p_i \in \Omega \setminus D, \\ 0 & \text{otherwise} \end{pmatrix}_{i=1,\dots,m} \in \mathbb{R}^{m\times m},$$

where D is the inpainting domain. The block $\boldsymbol{A}$ is given in the smooth system as

$$\boldsymbol{A} = \varepsilon(\boldsymbol{I} \otimes \boldsymbol{K}) + C_1(\boldsymbol{I} \otimes \boldsymbol{M}).$$

In the nonsmooth system, it is

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_{(1,1)} & \boldsymbol{A}_{(2)} & \cdots & \boldsymbol{A}_{(N-1)} & \boldsymbol{A}_{(N)} \\ \boldsymbol{A}_{(1)} & \boldsymbol{A}_{(2,2)} & \cdots & \boldsymbol{A}_{(N-1)} & \boldsymbol{A}_{(N)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{A}_{(1)} & \boldsymbol{A}_{(2)} & \cdots & \boldsymbol{A}_{(N-1,N-1)} & \boldsymbol{A}_{(N)} \\ \boldsymbol{A}_{(1)} & \boldsymbol{A}_{(2)} & \cdots & \boldsymbol{A}_{(N-1)} & \boldsymbol{A}_{(N,N)} \end{bmatrix},$$

where for $i = 1, \dots, N$

$$\begin{aligned} \boldsymbol{A}_{(i,i)} &= \boldsymbol{A}_{(i,i)}(\boldsymbol{u}_i^{(k)}) = \varepsilon\boldsymbol{K} + C_1\boldsymbol{M} + \frac{1}{c}\left(1 - \frac{1}{N}\right)\boldsymbol{G}_{(i)}, \\ \boldsymbol{A}_{(i)} &= \boldsymbol{A}_{(i)}(\boldsymbol{u}_i^{(k)}) = -\frac{1}{cN}\boldsymbol{G}_{(i)}, \\ \boldsymbol{G}_{(i)} &= \boldsymbol{G}_{(i)}(\boldsymbol{u}_i^{(k)}) = \operatorname{diag}\begin{pmatrix} [\boldsymbol{M}]_{jj} & \text{if } u_{h,i,j}^{(k)} < 0, \\ 0 & \text{otherwise} \end{pmatrix}_{j=1,\dots,m} \in \mathbb{R}^{m\times m}. \end{aligned} \tag{5.60}$$

Here, $u_{h,i,j}^{(k)}$ denotes the jth element of the vector $\boldsymbol{u}_{h,i}^{(k)}$.

Finally, we also state the fully discrete linear systems for the black-and-white Cahn–Hilliard inpainting models. The linear system in the smooth case reads

$$\begin{bmatrix} -\varepsilon\boldsymbol{K} - C_1\boldsymbol{M} & \boldsymbol{M} \\ \boldsymbol{M} & \frac{\tau}{1+\tau C_2}\boldsymbol{K} \end{bmatrix}\begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{w} \end{bmatrix} = \begin{bmatrix} -C_1\boldsymbol{M}\boldsymbol{u}^{\text{old}} + \frac{1}{\varepsilon}\boldsymbol{M}\psi'_{\text{pol,s}}\left(\boldsymbol{u}^{\text{old}}\right) \\ \boldsymbol{M}\boldsymbol{u}^{\text{old}} + \frac{\tau\omega_0}{1+\tau C_2}\boldsymbol{H}(\boldsymbol{f} - \boldsymbol{u}^{\text{old}}) \end{bmatrix} \tag{5.61}$$

per time step, where the scalar potential is given as

$$\psi_{\text{pol,s}}(u) = u^2(u-1)^2.$$

The linear system in the nonsmooth case reads

$$\begin{aligned} &\begin{bmatrix} -\varepsilon\boldsymbol{K} - C_1\boldsymbol{M} - c^{-1}\boldsymbol{G} & \boldsymbol{M} \\ \boldsymbol{M} & \frac{\tau}{1+\tau C_2}\boldsymbol{K} \end{bmatrix}\begin{bmatrix} \boldsymbol{u}^{(k+1)} \\ \boldsymbol{w}^{(k+1)} \end{bmatrix} \\ &\qquad = \begin{bmatrix} -C_1\boldsymbol{M}\boldsymbol{u}^{\text{old}} + \frac{1}{\varepsilon}\boldsymbol{M}\psi'_{0,s}\left(\boldsymbol{u}^{\text{old}}\right) - c^{-1}\boldsymbol{G}_{+}\boldsymbol{1} \\ \boldsymbol{M}\boldsymbol{u}^{\text{old}} + \frac{\tau\omega_0}{1+\tau C_2}\boldsymbol{H}(\boldsymbol{f} - \boldsymbol{u}^{\text{old}}) \end{bmatrix} \end{aligned} \tag{5.62}$$

per SSN step, where the scalar potential is given as

$$\psi_{0,s}(u) = \frac{1}{2}u(1-u),$$

and

$$\boldsymbol{G} = \boldsymbol{G}(\boldsymbol{u}^{(k)}) = \operatorname{diag}\begin{pmatrix} [\boldsymbol{M}]_{ii} & \text{if } u_{h,i}^{(k)} < 0 \text{ or } u_{h,i}^{(k)} > 1, \\ 0 & \text{otherwise} \end{pmatrix}_{i=1,\dots,m} \in \mathbb{R}^{m\times m},$$

$$\boldsymbol{G}_{+} = \boldsymbol{G}_{+}(\boldsymbol{u}^{(k)}) = \operatorname{diag}\begin{pmatrix} [\boldsymbol{M}]_{ii} & \text{if } u_{h,i}^{(k)} > 1, \\ 0, & \text{otherwise} \end{pmatrix}_{i=1,\dots,m} \in \mathbb{R}^{m\times m}.$$

Now, we have arrived at the core of our numerical algorithms — the numerical solution of systems of linear equations. Due to the use of FEM all the matrix blocks $\boldsymbol{M}, \boldsymbol{K}, \boldsymbol{H}, \boldsymbol{G}, \boldsymbol{G}_{(i)}$, $i = 1, \dots, N$, are large and sparse. In particular, the higher the number N of phases is the larger is every block of the system matrix in (5.59). In the next section, we design effective practical preconditioners for the two scalar linear systems (5.61) and (5.62) as well as for the two vector-valued linear systems represented in (5.59).

5.6 Preconditioning

This section is devoted to the development of practical preconditioners for the efficient solution of the four linear systems in (5.59), (5.61), and (5.62). We begin with the simplest problem and go step by step to the next harder one. We will see in the following two sections that the construction of efficient preconditioners in the smooth case, for both, the scalar and vector-valued system, complies with the study in Chapter 3.7.1. There, we investigated preconditioning techniques for scalar smooth semi-implicit Cahn–Hilliard systems. Those coefficient matrices have a similar structure to the ones we deal with in this chapter in the smooth case. As in Chapter 3.7.1, we develop two preconditioners: The first one uses the symmetry in the coefficient matrices. This allows us to make use of symmetric Krylov subspace solvers, which are cheaper than the nonsymmetric ones. The second preconditioner originates from [35, 37, 36, 3, 38]. However, our theoretical proofs differ halfway through. Note that this technique ignores the symmetry inherent in our coefficient matrices. However, our theoretical results below show there efficiency. The construction of efficient preconditioners in the scalar nonsmooth case complies with the study in Chapter 3.7.3. There, we have investigated preconditioning techniques for scalar nonsmooth semi-implicit Cahn–Hilliard systems. The construction of efficient preconditioners in the vector-valued nonsmooth case complies with the study in Chapter 4.7.2.

In the following, we start with the smooth systems.

5.6.1 Smooth systems

The smooth modified Cahn–Hilliard system (5.61) can be written as

$$\begin{aligned} \begin{bmatrix} \boldsymbol{A} & \boldsymbol{M} \\ \boldsymbol{M} & -\frac{\tau}{1+\tau C_2}\boldsymbol{K} \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ -\boldsymbol{w} \end{bmatrix} &:= \begin{bmatrix} \varepsilon\boldsymbol{K} + C_1\boldsymbol{M} & \boldsymbol{M} \\ \boldsymbol{M} & -\frac{\tau}{1+\tau C_2}\boldsymbol{K} \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ -\boldsymbol{w} \end{bmatrix} \\ &= \begin{bmatrix} C_1\boldsymbol{M}\boldsymbol{u}^{\text{old}} - \frac{1}{\varepsilon}\boldsymbol{M}\psi'_{\text{pol,s}}\left(\boldsymbol{u}^{\text{old}}\right) \\ \boldsymbol{M}\boldsymbol{u}^{\text{old}} + \frac{\tau\omega_0}{1+\tau C_2}\boldsymbol{H}(\boldsymbol{f} - \boldsymbol{u}^{\text{old}}) \end{bmatrix} \end{aligned} \qquad (5.63)$$

and is hence of saddle point form. In the following, we denote the coefficient matrix by $\mathcal{A}$. It can be easily seen that $\mathcal{A}$ is symmetric. Moreover, the $(1,1)$ block $\boldsymbol{A}$ is symmetric positive definite. According to Theorem 2.35, $\mathcal{A}$ is nonsingular. Due to Remark 2.3, the Schur complement

$$\boldsymbol{S} = -\left(\frac{\tau}{1+\tau C_2}\boldsymbol{K} + \boldsymbol{M}\boldsymbol{A}^{-1}\boldsymbol{M}\right)$$

is symmetric negative definite, and $\mathcal{A}$ is indefinite with m positive and m negative eigenvalues.

Next, we design a preconditioner. Since $\mathcal{A}$ is symmetric indefinite, our Krylov method of choice is MINRES. Hence, we need to construct a symmetric positive definite preconditioner, and we propose the block diagonal preconditioner

$$\mathcal{P} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{0} \\ \boldsymbol{0} & -\hat{\boldsymbol{S}} \end{bmatrix}. \tag{5.64}$$

As Schur complement approximation, we design $\hat{\boldsymbol{S}}$ as

$$\begin{aligned}\hat{\boldsymbol{S}} &= -\boldsymbol{S}_1\boldsymbol{A}^{-1}\boldsymbol{S}_1 \\ &= -\left(\boldsymbol{M} + \sqrt{\frac{\tau\varepsilon}{1+\tau C_2}}\boldsymbol{K}\right)\boldsymbol{A}^{-1}\left(\boldsymbol{M} + \sqrt{\frac{\tau\varepsilon}{1+\tau C_2}}\boldsymbol{K}\right) &(5.65)\\ &= -\frac{\tau\varepsilon}{1+\tau C_2}\boldsymbol{K}\boldsymbol{A}^{-1}\boldsymbol{K} - \boldsymbol{M}\boldsymbol{A}^{-1}\boldsymbol{M} - \sqrt{\frac{\tau\varepsilon}{1+\tau C_2}}\boldsymbol{M}\boldsymbol{A}^{-1}\boldsymbol{K} - \sqrt{\frac{\tau\varepsilon}{1+\tau C_2}}\boldsymbol{K}\boldsymbol{A}^{-1}\boldsymbol{M}. \\ &&(5.66)\end{aligned}$$

The second term in (5.66) matches the second term in the exact Schur complement. The first term in (5.66) approximates the first term in the exact Schur complement. Due to the balanced distribution of $\frac{\tau\varepsilon}{1+\tau C_2}$ in form of $\sqrt{\frac{\tau\varepsilon}{1+\tau C_2}}$ in the factor $\boldsymbol{S}_1$, the influence of both remainder terms in (5.66) is reduced.

Lemma 5.7. *$\hat{\boldsymbol{S}}$ is symmetric negative definite.*

The proof is the same as the one for Lemma 3.11. To illustrate the performance of $\hat{\boldsymbol{S}}^{-1}\boldsymbol{S}$, we show eigenvalue plots in Section 5.7.1. Let us conclude the preconditioner $\mathcal{P}$ with a statement about its practical realization. The action of the inverse of $\boldsymbol{S}_1$ is performed with an AMG since $\boldsymbol{S}_1$ forms the discretization of an elliptic operator. The same holds for the $(1,1)$ block $\boldsymbol{A}$. Hence, the practical block diagonal preconditioner is given by

$$\mathcal{P}_0 = \begin{bmatrix} \boldsymbol{A}_0 & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{S}_0 \end{bmatrix},$$

where $\boldsymbol{A}_0 = \text{AMG}(\boldsymbol{A})$ and $\boldsymbol{S}_0 = \text{AMG}(\boldsymbol{S}_1)\boldsymbol{A}^{-1}\text{AMG}(\boldsymbol{S}_1)$. In Section 5.7.2, we illustrate the robust performance of the preconditioner $\mathcal{P}_0$ applied with MINRES.

In the following, we discuss a second way to develop a preconditioner for the smooth modified Cahn–Hilliard system (5.61). We proceed in the same way as in Chapter 3.7.1. For the development of a preconditioner, we rewrite (5.63) again and consider

$$\begin{aligned}\begin{bmatrix} \boldsymbol{M} & -\boldsymbol{A} \\ \frac{\tau}{1+\tau C_2}\boldsymbol{K} & \boldsymbol{M}\end{bmatrix}\begin{bmatrix} \boldsymbol{w}\\ \boldsymbol{u}\end{bmatrix} &:= \begin{bmatrix} \boldsymbol{M} & -\varepsilon \boldsymbol{K} - C_1 \boldsymbol{M} \\ \frac{\tau}{1+\tau C_2}\boldsymbol{K} & \boldsymbol{M}\end{bmatrix}\begin{bmatrix} \boldsymbol{w}\\ \boldsymbol{u}\end{bmatrix} \\ &= \begin{bmatrix} -C_1 \boldsymbol{M}\boldsymbol{u}^{\text{old}} + \frac{1}{\varepsilon}\boldsymbol{M}\psi'_{\text{pol,s}}\left(\boldsymbol{u}^{\text{old}}\right) \\ \boldsymbol{M}\boldsymbol{u}^{\text{old}} + \frac{\tau\omega_0}{1+\tau C_2}\boldsymbol{H}(\boldsymbol{f}-\boldsymbol{u}^{\text{old}})\end{bmatrix}.\end{aligned} \tag{5.67}$$

In the following, we denote this coefficient matrix by $\boldsymbol{\mathcal{A}}$. Since we obtained $\boldsymbol{\mathcal{A}}$ from (5.63) by interchanging m columns and multiplying m rows by -1, its determinant does not change. Hence, $\boldsymbol{\mathcal{A}}$ remains nonsingular. Note that $\boldsymbol{\mathcal{A}}$ is not symmetric anymore is at was in the previous case. Hence, nonsymmetric Krylov subspace solvers have to be used. Due to Proposition 3.10, the diagonal entries of $C_1\boldsymbol{M}$ lie in the interval $C_1 h^d[\tilde{c}, C]$. Due to the convexity condition $C_1 > S\varepsilon^{-1}$ in Lemma 5.1, C_1 is comparable to ε^{-1}. Hence, the estimated order for the diagonal entries in $C_1\boldsymbol{M}$ lie in the interval $\varepsilon^{d-1}[\tilde{c}, C]$, since h is of order ε. Hence, as in Chapter 3.7.1, we suggest to neglect the block $C_1\boldsymbol{M}$ in $\boldsymbol{A}$. Therefore, we approximate $\boldsymbol{\mathcal{A}}$ as

$$\boldsymbol{\mathcal{A}}_0 = \begin{bmatrix} \boldsymbol{M} & -\varepsilon\boldsymbol{K} \\ \frac{\tau}{1+\tau C_2}\boldsymbol{K} & \boldsymbol{M}\end{bmatrix}.$$

In what follows, we discuss the quality of the approximation $\boldsymbol{\mathcal{A}}_0$. We denote the Schur complement of $\boldsymbol{\mathcal{A}}_0$ by $\tilde{\boldsymbol{S}} = \boldsymbol{M} + \frac{\tau\varepsilon}{1+\tau C_2}\boldsymbol{K}\boldsymbol{M}^{-1}\boldsymbol{K}$. Note that both, the $(1,1)$ and $(2,2)$ block of $\boldsymbol{\mathcal{A}}_0$, are nonsingular. In particular, they are symmetric positive definite. Consider the generalized eigenvalue problem

$$\boldsymbol{\mathcal{A}}\begin{bmatrix} q_1 \\ q_2\end{bmatrix} = \lambda\boldsymbol{\mathcal{A}}_0\begin{bmatrix} q_1 \\ q_2\end{bmatrix}. \tag{5.68}$$

Theorem 5.8. *It holds*

$$\sigma(\boldsymbol{\mathcal{A}}_0^{-1}\boldsymbol{\mathcal{A}}) \subset B_\varsigma(1).$$

The circle radius is bounded by $\varsigma \leq C_1\sqrt{\tau}/(2\sqrt{\varepsilon(1+\tau C_2)})$. In particular, m eigenvalues are equal to one. We get $\varsigma \leq 0.5$ when $C_2 \geq C_1^2/\varepsilon - \tau^{-1}$.

Proof. The proof is almost the same as the one for Theorem 3.12. The matrix $\boldsymbol{R}$ becomes

$$\boldsymbol{R} = \frac{C_1\tau}{1+\tau C_2}\left(\boldsymbol{I} + \frac{\tau\varepsilon}{1+\tau C_2}(\boldsymbol{M}^{-1}\boldsymbol{K})^2\right)^{-1}\boldsymbol{M}^{-1}\boldsymbol{K}$$

and hence $\tilde{\boldsymbol{R}}$ becomes

$$\tilde{\boldsymbol{R}} = \frac{C_1\tau}{1+\tau C_2}\left(\boldsymbol{I} + \frac{\tau\varepsilon}{1+\tau C_2}\tilde{\boldsymbol{K}}^2\right)^{-1}\tilde{\boldsymbol{K}}.$$

We finally end up with

$$\begin{aligned}\|\tilde{\boldsymbol{R}}\| &\leq \frac{C_1\tau}{1+\tau C_2}\left\|\left(\boldsymbol{I} + \tau\varepsilon\tilde{\boldsymbol{K}}^2\right)^{-1}\tilde{\boldsymbol{K}}\right\| = \frac{C_1\tau}{1+\tau C_2}\,\rho\left(\left(\boldsymbol{I} + \frac{\tau\varepsilon}{1+\tau C_2}\tilde{\boldsymbol{K}}^2\right)^{-1}\tilde{\boldsymbol{K}}\right) \\ &\leq \frac{C_1\sqrt{\tau}}{2\sqrt{\varepsilon(1+\tau C_2)}},\end{aligned}$$

where the equality holds due to the symmetry of $\left(\boldsymbol{I} + \tau\varepsilon\tilde{\boldsymbol{K}}^2\right)^{-1}\tilde{\boldsymbol{K}}$. Therefore, for $C_2 \geq C_1^2/\varepsilon - \tau^{-1}$, it holds $\sigma(\tilde{\boldsymbol{R}}) = \sigma(\boldsymbol{R}) \subset B_{0.5}(0)$ and hence $\sigma(\boldsymbol{\mathcal{A}}_0^{-1}\boldsymbol{\mathcal{A}}) \subset B_{0.5}(1)$. □

Remark 5.8. *Note that large values of C_2 are needed anyway due to accuracy reasons. The convexity condition in Lemma 5.1 requires $C_2 > \omega_0$, where ω_0 is the fidelity parameter. Large values of ω_0 yield reconstructed images that are close to the given image in the undamaged parts.*

After we have proven that $\mathcal{A}_0$ is a reasonable approximation of $\mathcal{A}$, we can go over to the construction of a suitable preconditioner $\mathcal{P}$ for $\mathcal{A}_0$ and hence for $\mathcal{A}$. We propose the block triangular preconditioner

$$\mathcal{P} = \begin{bmatrix} \boldsymbol{M} & \boldsymbol{0} \\ \frac{\tau}{1+\tau C_2}\boldsymbol{K} & -\hat{\boldsymbol{S}} \end{bmatrix}.$$

As Schur complement approximation, we design $\hat{\boldsymbol{S}}$ as

$$\begin{aligned} \hat{\boldsymbol{S}} &= \boldsymbol{S}_1\boldsymbol{M}^{-1}\boldsymbol{S}_1 \\ &= \left(\boldsymbol{M} + \sqrt{\frac{\varepsilon\tau}{1+\tau C_2}}\boldsymbol{K}\right)\boldsymbol{M}^{-1}\left(\boldsymbol{M} + \sqrt{\frac{\varepsilon\tau}{1+\tau C_2}}\boldsymbol{K}\right) \qquad (5.69) \\ &= \boldsymbol{M} + \frac{\varepsilon\tau}{1+\tau C_2}\boldsymbol{K}\boldsymbol{M}^{-1}\boldsymbol{K} + 2\sqrt{\frac{\varepsilon\tau}{1+\tau C_2}}\boldsymbol{K}. \qquad (5.70) \end{aligned}$$

The first two terms in (5.70) match the exact Schur complement $\tilde{\boldsymbol{S}} = \boldsymbol{M} + \frac{\tau\varepsilon}{1+\tau C_2}\boldsymbol{K}\boldsymbol{M}^{-1}\boldsymbol{K}$ of $\mathcal{A}_0$. The influence of the last term in (5.70) is reduced due to the factor $\sqrt{\frac{\varepsilon\tau}{1+\tau C_2}}$. In fact, this approximation turns out to be an optimal Schur complement preconditioner for $\mathcal{A}_0$ (see also [130]):

Lemma 5.9. *It holds*

$$\sigma\left(\hat{\boldsymbol{S}}^{-1}\tilde{\boldsymbol{S}}\right) \subset [0.5, 1].$$

The proof is the same as the one for Lemma 3.13. Let us conclude the preconditioner $\mathcal{P}$ with a statement about its practical realization. The action of the inverse of $\boldsymbol{S}_1$ is performed with an AMG since $\boldsymbol{S}_1$ forms the discretization of an elliptic operator. The $(1,1)$ block $\boldsymbol{M}$ is a diagonal matrix with positive entries. Hence, its inverse can be performed by elementwise multiplications. Hence, the practical block triangular preconditioner is given by

$$\mathcal{P}_0 = \begin{bmatrix} \boldsymbol{M} & \boldsymbol{0} \\ \frac{\tau}{1+\tau C_2}\boldsymbol{K} & -\boldsymbol{S}_0 \end{bmatrix},$$

where $\boldsymbol{S}_0 = \mathrm{AMG}(\boldsymbol{S}_1)\boldsymbol{M}^{-1}\mathrm{AMG}(\boldsymbol{S}_1)$.

Since the above theoretical analysis proves the optimality of the preconditioner $\mathcal{P}$ for large values of C_2, we will not study the numerical robustness.

Next, we consider the smooth vector-valued modified Cahn–Hilliard system represented in (5.59). Every block in the coefficient matrix is of diagonal form. In fact, the structure is exactly the same as in the smooth scalar version discussed above. Therefore, all results presented in this section hold true for the smooth vector-valued modified Cahn–Hilliard system as well.

Here, we finish the discussion about preconditioning of smooth modified Cahn–Hilliard systems. Next, we come to the harder case of nonsmooth systems. We will see that a simplification of the coefficient matrix in form of $\mathcal{A}_0$ is only satisfying for moderate sizes of the penalty parameter c. We start with the scalar version (5.62).

5.6.2 Scalar nonsmooth systems

Consider the matrix system in (5.62) with the coefficient matrix

$$\mathcal{A} = \begin{bmatrix} -\varepsilon \boldsymbol{K} - C_1 \boldsymbol{M} - c^{-1}\boldsymbol{G} & \boldsymbol{M} \\ \boldsymbol{M} & \frac{\tau}{1+\tau C_2}\boldsymbol{K} \end{bmatrix} =: \begin{bmatrix} -\boldsymbol{A} & \boldsymbol{M} \\ \boldsymbol{M} & \frac{\tau}{1+\tau C_2}\boldsymbol{K} \end{bmatrix}. \tag{5.71}$$

It can be easily seen that $\mathcal{A}$ is symmetric. The $(1,1)$ block $\boldsymbol{A}$ is symmetric positive definite. Let us have a closer look at the matrix $c^{-1}\boldsymbol{G}$ and assume that $\boldsymbol{G} \not\equiv \boldsymbol{0}$. Then, penalized entries, i.e., the nonzero entries, are in general scattered throughout its diagonal. The intensity of the penalization can be controlled by the penalty parameter c. The smaller c is the stronger is the penalization and the more accurate is the numerical approximation of the nonsmoothness. In particular, the penalized entries of $c^{-1}\boldsymbol{G}$ lie in the interval $c^{-1}h^d[\tilde{c}, C]$. The nonpenalized entries of $c^{-1}\boldsymbol{G}$ are equal to zero. The Schur complement of $\mathcal{A}$ is $\boldsymbol{S} = \frac{\tau}{1+\tau C_2}\boldsymbol{K} + \boldsymbol{M}\boldsymbol{A}^{-1}\boldsymbol{M}$. For moderate sizes of c, i.e., $c \geq \varepsilon^{d-1}$, we suggest a symmetric Schur complement preconditioner of the form

$$\begin{aligned}
&\left(\boldsymbol{M} + \sqrt{\frac{\tau\varepsilon}{1+\tau C_2}}\boldsymbol{K}\right)\boldsymbol{A}^{-1}\left(\boldsymbol{M} + \sqrt{\frac{\tau\varepsilon}{1+\tau C_2}}\boldsymbol{K}\right) \\
&\quad = \frac{\tau\varepsilon}{1+\tau C_2}\boldsymbol{K}\boldsymbol{A}^{-1}\boldsymbol{K} + \boldsymbol{M}\boldsymbol{A}^{-1}\boldsymbol{M} + \sqrt{\frac{\tau\varepsilon}{1+\tau C_2}}\boldsymbol{M}\boldsymbol{A}^{-1}\boldsymbol{K} + \sqrt{\frac{\tau\varepsilon}{1+\tau C_2}}\boldsymbol{K}\boldsymbol{A}^{-1}\boldsymbol{M}
\end{aligned}$$

as used in (5.65). The approximation of the first term in the exact Schur complement is satisfying if $c \geq \varepsilon^{d-1}$ since

$$\frac{\tau\varepsilon}{1+\tau C_2}\boldsymbol{K}\boldsymbol{A}^{-1} = \frac{\tau}{1+\tau C_2}\boldsymbol{K}(\boldsymbol{K} + C_1\varepsilon^{-1}\boldsymbol{M} + c^{-1}\varepsilon^{-1}\boldsymbol{G})^{-1}$$

and the estimated order for the diagonal entries in $C_1\varepsilon^{-1}\boldsymbol{M} + c^{-1}\varepsilon^{-1}\boldsymbol{G}$ lie in the interval $\varepsilon^{d-1}[\tilde{c}\varepsilon^{-1}, C(c^{-1} + \varepsilon^{-1})]$, where we have used that h is of order ε and C_1 is of order ε^{-1}. Similarly, neglecting the block $C_1\boldsymbol{M} + c^{-1}\boldsymbol{G}$ in $\boldsymbol{A}$ would give a good approximation only for large penalization parameters c, which is summarized as follows:

Theorem 5.10. *Let*

$$\mathcal{A} = \begin{bmatrix} \boldsymbol{M} & -\varepsilon\boldsymbol{K} - C_1\boldsymbol{M} - c^{-1}\boldsymbol{G} \\ \frac{\tau}{1+\tau C_2}\boldsymbol{K} & \boldsymbol{M} \end{bmatrix} \quad \text{and} \quad \mathcal{A}_0 = \begin{bmatrix} \boldsymbol{M} & -\varepsilon\boldsymbol{K} \\ \frac{\tau}{1+\tau C_2}\boldsymbol{K} & \boldsymbol{M} \end{bmatrix}.$$

It holds

$$\sigma(\mathcal{A}_0^{-1}\mathcal{A}) \subset B_{\varsigma}(1).$$

The circle radius is bounded by $\varsigma \leq \frac{\sqrt{\tau}(c^{-1}+C_1)}{2\sqrt{\varepsilon(1+\tau C_2)}}$. *In particular, m eigenvalues are equal to one. We get $\varsigma \leq 0.5$ when one of the following conditions holds:*

(a) $C_2 \geq \frac{(C_1+c^{-1})^2}{\varepsilon} - \frac{1}{\tau}$, *or*

(b) $C_2 > \frac{C_1^2}{\varepsilon} - \frac{1}{\tau}$ *and* $c \geq \frac{\sqrt{\tau}}{\sqrt{\varepsilon(1+\tau C_2)} - C_1\sqrt{\tau}}$.

Proof. The proof is almost the same as the one for Theorem 3.12. The matrix $\boldsymbol{R}$ becomes

$$\boldsymbol{R} = \frac{\tau}{1+\tau C_2}\left(\boldsymbol{I} + \frac{\tau\varepsilon}{1+\tau C_2}(\boldsymbol{M}^{-1}\boldsymbol{K})^2\right)^{-1}\boldsymbol{M}^{-1}\boldsymbol{K}\boldsymbol{M}^{-1}\left(c^{-1}\boldsymbol{G} + C_1\boldsymbol{M}\right)$$

and hence $\tilde{\boldsymbol{R}}$ becomes

$$\tilde{\boldsymbol{R}} = \frac{\tau}{1+\tau C_2}\left(\boldsymbol{I} + \frac{\tau\varepsilon}{1+\tau C_2}\tilde{\boldsymbol{K}}^2\right)^{-1}\tilde{\boldsymbol{K}}\left(c^{-1}\tilde{\boldsymbol{G}} + C_1\boldsymbol{I}\right),$$

where $\tilde{\boldsymbol{G}} = \boldsymbol{M}^{-\frac{1}{2}}\boldsymbol{G}\boldsymbol{M}^{-\frac{1}{2}}$. We finally end up with

$$\begin{aligned}
\|\tilde{\boldsymbol{R}}\| &\leq \frac{\tau}{1+\tau C_2}\left\|\left(\boldsymbol{I} + \frac{\tau\varepsilon}{1+\tau C_2}\tilde{\boldsymbol{K}}^2\right)^{-1}\tilde{\boldsymbol{K}}\right\|\|c^{-1}\tilde{\boldsymbol{G}} + C_1\boldsymbol{I}\| \\
&= \frac{\tau}{1+\tau C_2}\rho\left(\left(\boldsymbol{I} + \frac{\tau\varepsilon}{1+\tau C_2}\tilde{\boldsymbol{K}}^2\right)^{-1}\tilde{\boldsymbol{K}}\right)\rho(c^{-1}\tilde{\boldsymbol{G}} + C_1\boldsymbol{I}) \\
&\leq \frac{\sqrt{\tau}}{2\sqrt{\varepsilon(1+\tau C_2)}}\rho(c^{-1}\tilde{\boldsymbol{G}} + C_1\boldsymbol{I}),
\end{aligned}$$

where the equality holds due to the symmetry of $\left(\boldsymbol{I} + \frac{\tau\varepsilon}{1+\tau C_2}\tilde{\boldsymbol{K}}^2\right)^{-1}\tilde{\boldsymbol{K}}$ and $c^{-1}\tilde{\boldsymbol{G}} + C_1\boldsymbol{I}$. Moreover, due to the diagonal structure of $\boldsymbol{G}$, we have

$$\tilde{\boldsymbol{G}} = \boldsymbol{M}^{-\frac{1}{2}}\boldsymbol{G}\boldsymbol{M}^{-\frac{1}{2}} = \operatorname{diag}\begin{pmatrix} 1 & \text{if } u_{h,i}^{(k)} < 0 \text{ or } u_{h,i}^{(k)} > 1, \\ 0 & \text{otherwise.} \end{pmatrix}_{i=1,\dots,m}$$

Therefore,

$$c^{-1}\tilde{\boldsymbol{G}} + C_1\boldsymbol{I} = \operatorname{diag}\begin{pmatrix} c^{-1} + C_1 & \text{if } u_{h,i}^{(k)} < 0 \text{ or } u_{h,i}^{(k)} > 1, \\ C_1 & \text{otherwise,} \end{pmatrix}_{i=1,\dots,m}$$

which results in

$$\rho(c^{-1}\tilde{\boldsymbol{G}} + C_1\boldsymbol{I}) = c^{-1} + C_1.$$

We obtain

$$\|\tilde{\boldsymbol{R}}\| \leq \frac{\sqrt{\tau}(c^{-1} + C_1)}{2\sqrt{\varepsilon(1+\tau C_2)}}.$$

Therefore, for $\frac{\sqrt{\tau}(c^{-1}+C_1)}{2\sqrt{\varepsilon(1+\tau C_2)}} \leq \frac{1}{2}$, it holds $\sigma(\tilde{\boldsymbol{R}}) = \sigma(\boldsymbol{R}) \subset B_{0.5}(0)$ and hence $\sigma(\mathcal{A}_{\mathbf{0}}^{-1}\mathcal{A}) \subset B_{0.5}(1)$. If we solve this inequality for C_2, we obtain the condition $C_2 \geq \frac{(C_1+c^{-1})^2}{\varepsilon} - \frac{1}{\tau}$. For small penalty parameters, this bound becomes too large. Another condition arises if we solve the inequality for c. Then, we obtain the inequality $c\left(\frac{\sqrt{\varepsilon(1+\tau C_2)}}{\sqrt{\tau}} - C_1\right) \geq 1$. In order to ensure that the left-hand side is positive, we need $C_2 > \frac{C_1^2}{\varepsilon} - \frac{1}{\tau}$. Finally, we obtain $c \geq \frac{\sqrt{\tau}}{\sqrt{\varepsilon(1+\tau C_2)} - C_1\sqrt{\tau}}$. As in the condition before, c is needed to be large. □

In the following, we propose preconditioners for the case of small penalty parameters c. We build on the last two chapters and keep the whole block $\boldsymbol{A}$ within our Schur complement approximation

We concentrate on the coefficient matrix in (5.71). The first block triangular preconditioner we propose is

$$\boldsymbol{\mathcal{P}} = \begin{bmatrix} -\boldsymbol{A} & \boldsymbol{0} \\ \boldsymbol{M} & -\hat{\boldsymbol{S}} \end{bmatrix}. \tag{5.72}$$

As Schur complement approximation, we design $\hat{\boldsymbol{S}}$ as

$$\begin{aligned} \hat{\boldsymbol{S}} &= \boldsymbol{S}_1 \boldsymbol{A}^{-1} \boldsymbol{S}_2 \\ &= \left(\boldsymbol{M} + \sqrt{\frac{\tau}{1+\tau C_2}}\boldsymbol{K}\right)\boldsymbol{A}^{-1}\left(\boldsymbol{M} + \sqrt{\frac{\tau}{1+\tau C_2}}\boldsymbol{A}\right) \qquad (5.73) \\ &= \frac{\tau}{1+\tau C_2}\boldsymbol{K} + \boldsymbol{M}\boldsymbol{A}^{-1}\boldsymbol{M} + \sqrt{\frac{\tau}{1+\tau C_2}}\boldsymbol{M} + \sqrt{\frac{\tau}{1+\tau C_2}}\boldsymbol{K}\boldsymbol{A}^{-1}\boldsymbol{M}. \qquad (5.74) \end{aligned}$$

The first two terms in (5.74) match the exact Schur complement $\boldsymbol{S} = \frac{\tau}{1+\tau C_2}\boldsymbol{K} + \boldsymbol{M}\boldsymbol{A}^{-1}\boldsymbol{M}$. Due to the balanced distribution of $\frac{\tau}{1+\tau C_2}$ in form of $\sqrt{\frac{\tau}{1+\tau C_2}}$ in both factors $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$, the influence of both remainder terms in (5.74) is reduced. Let us conclude the preconditioner $\boldsymbol{\mathcal{P}}$ with a statement about its practical realization. The action of the inverse of $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ is performed with an AMG each since both form the discretization of an elliptic operator. The same holds for the $(1,1)$ block $\boldsymbol{A}$. Hence, the practical block triangular preconditioner is given by

$$\boldsymbol{\mathcal{P}}_0 = \begin{bmatrix} -\boldsymbol{A}_0 & \boldsymbol{0} \\ \boldsymbol{M} & -\boldsymbol{S}_0 \end{bmatrix},$$

where $\boldsymbol{A}_0 = \mathrm{AMG}(\boldsymbol{A})$ and $\boldsymbol{S}_0 = \mathrm{AMG}(\boldsymbol{S}_1)\boldsymbol{A}^{-1}\mathrm{AMG}(\boldsymbol{S}_2)$.

In the following, we discuss a second way to develop a preconditioner for the scalar nonsmooth modified Cahn–Hilliard system (5.62). By interchanging the column blocks in (5.62) we obtain

$$\begin{aligned} \begin{bmatrix} \boldsymbol{M} & -\boldsymbol{A} \\ \frac{\tau}{1+\tau C_2}\boldsymbol{K} & \boldsymbol{M} \end{bmatrix}\begin{bmatrix} \boldsymbol{w}^{(k+1)} \\ \boldsymbol{u}^{(k+1)} \end{bmatrix} &:= \begin{bmatrix} \boldsymbol{M} & -\varepsilon\boldsymbol{K} - C_1\boldsymbol{M} - c^{-1}\boldsymbol{G} \\ \frac{\tau}{1+\tau C_2}\boldsymbol{K} & \boldsymbol{M} \end{bmatrix}\begin{bmatrix} \boldsymbol{w}^{(k+1)} \\ \boldsymbol{u}^{(k+1)} \end{bmatrix} \\ &= \begin{bmatrix} -C_1\boldsymbol{M}\boldsymbol{u}^{\mathrm{old}} + \frac{1}{\varepsilon}\boldsymbol{M}\psi'_{0,s}\left(\boldsymbol{u}^{\mathrm{old}}\right) - c^{-1}\boldsymbol{G}_{+}\boldsymbol{1} \\ \boldsymbol{M}\boldsymbol{u}^{\mathrm{old}} + \frac{\tau\omega_0}{1+\tau C_2}\boldsymbol{H}(\boldsymbol{f} - \boldsymbol{u}^{\mathrm{old}}) \end{bmatrix}. \end{aligned} \tag{5.75}$$

In the following, we denote the coefficient matrix by $\boldsymbol{\mathcal{A}}$. The Schur complement is now $\boldsymbol{S} = \boldsymbol{M} + \frac{\tau}{1+\tau C_2}\boldsymbol{K}\boldsymbol{M}^{-1}\boldsymbol{A}$. It can be easily seen that $\boldsymbol{\mathcal{A}}$ is not symmetric anymore. However, the preconditioner above has already been built based on a nonsymmetric Schur complement approximation, which results in the use of nonsymmetric Krylov subspace solvers. The advantage of the form (5.75) is that the $(1,1)$ block is now diagonal and symmetric positive definite and hence cheap to invert. The block triangular preconditioner we propose is

$$\boldsymbol{\mathcal{P}} = \begin{bmatrix} \boldsymbol{M} & \boldsymbol{0} \\ \frac{\tau}{1+\tau C_2}\boldsymbol{K} & -\hat{\boldsymbol{S}} \end{bmatrix}. \tag{5.76}$$

As Schur complement approximation, we design $\hat{\boldsymbol{S}}$ as

$$\begin{aligned}
\hat{\boldsymbol{S}} &= \boldsymbol{S_1}\boldsymbol{M}^{-1}\boldsymbol{S_2} \\
&= \left(\sqrt{\varepsilon}\boldsymbol{M} + \sqrt{\frac{\tau}{1+\tau C_2}}\boldsymbol{K}\right)\boldsymbol{M}^{-1}\left(\frac{1}{\sqrt{\varepsilon}}\boldsymbol{M} + \sqrt{\frac{\tau}{1+\tau C_2}}\boldsymbol{A}\right) && (5.77) \\
&= \boldsymbol{M} + \frac{\tau}{1+\tau C_2}\boldsymbol{K}\boldsymbol{M}^{-1}\boldsymbol{A} + \sqrt{\frac{\varepsilon\tau}{1+\tau C_2}}\boldsymbol{A} + \sqrt{\frac{\tau}{\varepsilon(1+\tau C_2)}}\boldsymbol{K}. && (5.78)
\end{aligned}$$

The first two terms in (5.78) match the exact Schur complement. Due to the balanced distribution of $\frac{\tau}{1+\tau C_2}$ in form of $\sqrt{\frac{\tau}{1+\tau C_2}}$ as well as the scaling with $\sqrt{\varepsilon}$ and its inverse in both factors, $\hat{\mathcal{S}}_1$ and $\hat{\mathcal{S}}_2$, the influence of both remainder terms in (5.78) is reduced. To illustrate the performance of $\hat{\boldsymbol{S}}^{-1}\boldsymbol{S}$, we show eigenvalue plots in Section 5.7.1. The practical block triangular preconditioner is given by

$$\boldsymbol{\mathcal{P}_0} = \begin{bmatrix} \boldsymbol{M} & \boldsymbol{0} \\ \frac{\tau}{1+\tau C_2}\boldsymbol{K} & -\boldsymbol{S_0} \end{bmatrix},$$

where $\boldsymbol{S_0} = \mathrm{AMG}(\boldsymbol{S_1})\boldsymbol{M}^{-1}\mathrm{AMG}(\boldsymbol{S_2})$. In Section 5.7.2, we illustrate the robust performance of the preconditioner $\boldsymbol{\mathcal{P}_0}$ applied with BiCG.

In the numerical experiments, we always use the second preconditioner based on the system matrix (5.75), preconditioner (5.76), and Schur complement approximation (5.77). Moreover, we suggest this preconditioner to solve the vector-valued nonsmooth system in (5.59), which is discussed next.

5.6.3 Vector-valued nonsmooth systems

In the following, we develop a preconditioner for the nonsmooth vector-valued modified Cahn–Hilliard system represented in (5.59). In contrast to the vector-valued smooth case in (5.59), where each block in the coefficient matrix is block diagonal, we deal with a nondiagonal block matrix in the (1, 1) block. Moreover, the (1, 1) block is nonsymmetric here. Hence, as in Chapter 4.7, we rewrite (5.59) and consider

$$\begin{aligned}
&\begin{bmatrix} \boldsymbol{I}\otimes\boldsymbol{M} & -\boldsymbol{A} \\ \frac{\tau}{1+\tau C_2}\boldsymbol{I}\otimes\boldsymbol{K} & \boldsymbol{I}\otimes\boldsymbol{M} \end{bmatrix}\begin{bmatrix} \boldsymbol{w}^{(k+1)} \\ \boldsymbol{u}^{(k+1)} \end{bmatrix} \\
&= \begin{bmatrix} -C_1(\boldsymbol{I}\otimes\boldsymbol{M})\,\boldsymbol{u}^{\mathrm{old}} + \frac{1}{\varepsilon}(\boldsymbol{I}\otimes\boldsymbol{M})\psi_0'\left(\boldsymbol{u}^{\mathrm{old}}\right) - \frac{1}{\varepsilon N}(\boldsymbol{I}\otimes\boldsymbol{M})\left(\sum_{j=1}^{N}\psi_0'\left(\boldsymbol{u}_j^{\mathrm{old}}\right)\right)\boldsymbol{1} \\ (\boldsymbol{I}\otimes\boldsymbol{M})\,\boldsymbol{u}^{\mathrm{old}} + \frac{\tau\omega_0}{1+\tau C_2}(\boldsymbol{I}\otimes\boldsymbol{H})(\boldsymbol{f} - \boldsymbol{u}^{\mathrm{old}}) \end{bmatrix}. \qquad (5.79)
\end{aligned}$$

In the following, we denote the coefficient matrix in (5.79) by $\boldsymbol{\mathcal{A}}$. We propose the block triangular preconditioner

$$\boldsymbol{\mathcal{P}} = \begin{bmatrix} \boldsymbol{I}\otimes\boldsymbol{M} & \boldsymbol{0} \\ \frac{\tau}{1+\tau C_2}\boldsymbol{I}\otimes\boldsymbol{K} & -\hat{\boldsymbol{S}} \end{bmatrix}, \qquad (5.80)$$

where $\hat{\boldsymbol{S}}$ is an approximation of the Schur complement $\boldsymbol{S} = \boldsymbol{I}\otimes\boldsymbol{M} + \frac{\tau}{1+\tau C_2}(\boldsymbol{I}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}\boldsymbol{A}$. Inverting the block $\boldsymbol{I}\otimes\boldsymbol{M}$ is cheap as $\boldsymbol{M}$ is a diagonal matrix with positive entries. The remaining task is to design a Schur complement approximation $\hat{\boldsymbol{S}}$ that is easy to invert and resembles $\boldsymbol{S}$. The practical block triangular preconditioner is given by

$$\mathcal{P}_0 = \begin{bmatrix} \boldsymbol{I} \otimes \boldsymbol{M} & \boldsymbol{0} \\ \frac{\tau}{1+\tau C_2} \boldsymbol{I} \otimes \boldsymbol{K} & -\boldsymbol{S}_0 \end{bmatrix},$$

where we have to discuss an implementable Schur complement approximation $\boldsymbol{S}_0$ of $\hat{\boldsymbol{S}}$. As in Chapter 4.7.2, the difficult point is the nondiagonal block matrix $\boldsymbol{A}$. It contains the gradient energy parts, which only arise in the diagonal blocks, as well as the coupling of all penalization terms. As in the last section and previous chapters, neglecting the penalty blocks $\boldsymbol{G}_{(i)}$, $i = 1, \ldots, N$, together with the blocks $C_1 \boldsymbol{M}$ in $\boldsymbol{A}$, would give a worse approximation for small penalization parameters, which is summarized below in Theorem 5.11. The penalization is even more crucial than in the last section. As can be seen from (5.60), penalized entries are in general scattered throughout the diagonals of every block of $\boldsymbol{A}$. In particular, the penalized entries of the nondiagonal blocks $\boldsymbol{A}_{(i)}$, $i = 1, \ldots, N$, of $\boldsymbol{A}$ lie in the interval $-\frac{h^d}{cN}[C, \tilde{c}]$. The nonpenalized entries are equal to zero. This also applies to the diagonal blocks $\boldsymbol{A}_{(i,i)}$, $i = 1, \ldots, N$, of $\boldsymbol{A}$, whereby the matrix $\varepsilon \boldsymbol{K} + C_1 \boldsymbol{M}$ comes in addition. Again, this indicates a severe dependency between ε and c and hence h. All in all, large penalized entries should not be neglected.

Theorem 5.11. *Let*

$$\mathcal{A}_0 = \begin{bmatrix} \boldsymbol{I} \otimes \boldsymbol{M} & -\varepsilon \boldsymbol{I} \otimes \boldsymbol{K} \\ \frac{\tau}{1+\tau C_2} \boldsymbol{I} \otimes \boldsymbol{K} & \boldsymbol{I} \otimes \boldsymbol{M} \end{bmatrix}.$$

It holds

$$\sigma(\mathcal{A}_0^{-1} \mathcal{A}) \subset B_\varsigma(1).$$

The circle radius is bounded by $\varsigma < \frac{\sqrt{\tau}}{2\sqrt{\varepsilon(1+\tau C_2)}} \left(\frac{2}{c} + C_1 \right)$. *In particular, Nm eigenvalues are equal to one. We get* $\varsigma \leq 0.5$ *when one of the following conditions holds:*

(a) $C_2 \geq \frac{(C_1 + \frac{2}{c})^2}{\varepsilon} - \frac{1}{\tau}$, *or*

(b) $C_2 > \frac{C_1^2}{\varepsilon} - \frac{1}{\tau}$ *and* $c \geq \frac{2\sqrt{\tau}}{\sqrt{\varepsilon(1+\tau C_2)} - C_1 \sqrt{\tau}}$.

Proof. The proof is almost the same as the one for Theorem 4.10. The matrix $\boldsymbol{R}$ becomes

$$\boldsymbol{R} = \frac{\tau}{1+\tau C_2} \left(\boldsymbol{I} + \frac{\tau \varepsilon}{1+\tau C_2} \left[(\boldsymbol{I} \otimes \boldsymbol{M})^{-1} (\boldsymbol{I} \otimes \boldsymbol{K}) \right]^2 \right)^{-1} (\boldsymbol{I} \otimes \boldsymbol{M})^{-1} (\boldsymbol{I} \otimes \boldsymbol{K}) (\boldsymbol{I} \otimes \boldsymbol{M})^{-1} \left[\boldsymbol{A} - \varepsilon (\boldsymbol{I} \otimes \boldsymbol{K}) \right]$$

and hence $\tilde{\boldsymbol{R}}$ becomes

$$\tilde{\boldsymbol{R}} = \frac{\tau}{1+\tau C_2} \left(\boldsymbol{I} + \frac{\tau \varepsilon}{1+\tau C_2} \tilde{\boldsymbol{K}}^2 \right)^{-1} \tilde{\boldsymbol{K}} \tilde{\boldsymbol{A}},$$

where $\tilde{\boldsymbol{A}} = (\boldsymbol{I} \otimes \boldsymbol{M}^{-\frac{1}{2}}) \left[\boldsymbol{A} - \varepsilon (\boldsymbol{I} \otimes \boldsymbol{K}) \right] (\boldsymbol{I} \otimes \boldsymbol{M}^{-\frac{1}{2}})$. If we replace in the proof of Theorem 4.10 the matrix $\boldsymbol{Q}_L$ by the identity matrix $\boldsymbol{I} \in \mathbb{R}^{N \times N}$, we obtain in (4.94)

$$\left(\boldsymbol{I} + \frac{\tau \varepsilon}{1+\tau C_2} \tilde{\boldsymbol{K}}^2 \right)^{-1} \tilde{\boldsymbol{K}} = (\boldsymbol{I} \otimes \boldsymbol{Q}_K) \left[\boldsymbol{I} + \frac{\tau \varepsilon}{1+\tau C_2} (\boldsymbol{I} \otimes \boldsymbol{\Lambda}_K^2) \right]^{-1} (\boldsymbol{I} \otimes \boldsymbol{\Lambda}_K)(\boldsymbol{I} \otimes \boldsymbol{Q}_K^T),$$

where $\left[\boldsymbol{I} + \frac{\tau\varepsilon}{1+\tau C_2}(\boldsymbol{I} \otimes \boldsymbol{\Lambda}_K^2)\right]^{-1}(\boldsymbol{I} \otimes \boldsymbol{\Lambda}_K)$ is a diagonal matrix. Hence, $\left(\boldsymbol{I} + \frac{\tau\varepsilon}{1+\tau C_2}\tilde{\boldsymbol{K}}^2\right)^{-1}\tilde{\boldsymbol{K}}$ is symmetric. It follows

$$\left(\boldsymbol{I} + \frac{\tau\varepsilon}{1+\tau C_2}\tilde{\boldsymbol{K}}^2\right)^{-1}\tilde{\boldsymbol{K}}(\boldsymbol{e}_i \otimes \boldsymbol{q}_{K,j}) = \frac{\lambda_{K,j}}{1 + \frac{\tau\varepsilon}{1+\tau C_2}\lambda_{K,j}^2}(\boldsymbol{e}_i \otimes \boldsymbol{q}_{K,j}) \tag{5.81}$$

for $i = 1, \ldots, N$, $j = 1, \ldots, m$. Using the inequality

$$0 \leq (1 - ab)^2 = 1 + a^2b^2 - 2ab$$

with $a, b \in \mathbb{R}$, we can bound the eigenvalues of (5.81) as

$$\frac{\lambda_{K,j}}{1 + \frac{\tau\varepsilon}{1+\tau C_2}\lambda_{K,j}^2} \leq \frac{\sqrt{1+\tau C_2}}{2\sqrt{\tau\varepsilon}}.$$

for $j = 1, \ldots, m$. Here, we have used $a^2 = \frac{\tau\varepsilon}{1+\tau C_2}$ and $b^2 = \lambda_{K,j}^2$. This yields

$$\rho\left(\left(\boldsymbol{I} + \frac{\tau\varepsilon}{1+\tau C_2}\tilde{\boldsymbol{K}}^2\right)^{-1}\tilde{\boldsymbol{K}}\right) \leq \frac{\sqrt{1+\tau C_2}}{2\sqrt{\tau\varepsilon}}. \tag{5.82}$$

We finally end up with

$$\begin{aligned}
\|\tilde{\boldsymbol{R}}\| &\leq \frac{\tau}{1+\tau C_2}\left\|\left(\boldsymbol{I} + \frac{\tau\varepsilon}{1+\tau C_2}\tilde{\boldsymbol{K}}^2\right)^{-1}\tilde{\boldsymbol{K}}\right\|\|\tilde{\boldsymbol{A}}\| \\
&= \frac{\tau}{1+\tau C_2}\rho\left(\left(\boldsymbol{I} + \frac{\tau\varepsilon}{1+\tau C_2}\tilde{\boldsymbol{K}}^2\right)^{-1}\tilde{\boldsymbol{K}}\right)\|\tilde{\boldsymbol{A}}\| \\
&\leq \frac{\sqrt{\tau}}{2\sqrt{\varepsilon(1+\tau C_2)}}\|\tilde{\boldsymbol{A}}\|,
\end{aligned}$$

where the equality holds due to the symmetry of $\left(\boldsymbol{I} + \frac{\tau\varepsilon}{1+\tau C_2}\tilde{\boldsymbol{K}}^2\right)^{-1}\tilde{\boldsymbol{K}}$. Due to the diagonal structure of each block in $\boldsymbol{A} - \varepsilon(\boldsymbol{I} \otimes \boldsymbol{K})$, we have

$$\begin{aligned}
\tilde{\boldsymbol{A}} &= (\boldsymbol{I} \otimes \boldsymbol{M}^{-\frac{1}{2}})\,[\boldsymbol{A} - \varepsilon(\boldsymbol{I} \otimes \boldsymbol{K})]\,(\boldsymbol{I} \otimes \boldsymbol{M}^{-\frac{1}{2}}) \\
&= \frac{1}{N}\begin{bmatrix}
c^{-1}(N-1)\tilde{\boldsymbol{G}}_{(1)} + NC_1 & -c^{-1}\tilde{\boldsymbol{G}}_{(2)} & \cdots & -c^{-1}\tilde{\boldsymbol{G}}_{(N)} \\
-c^{-1}\tilde{\boldsymbol{G}}_{(1)} & c^{-1}(N-1)\tilde{\boldsymbol{G}}_{(2)} + NC_1 & \cdots & -c^{-1}\tilde{\boldsymbol{G}}_{(N)} \\
\vdots & \vdots & \ddots & \vdots \\
-c^{-1}\tilde{\boldsymbol{G}}_{(1)} & -c^{-1}\tilde{\boldsymbol{G}}_{(2)} & \cdots & -c^{-1}\tilde{\boldsymbol{G}}_{(N)} \\
-c^{-1}\tilde{\boldsymbol{G}}_{(1)} & -c^{-1}\tilde{\boldsymbol{G}}_{(2)} & \cdots & c^{-1}(N-1)\tilde{\boldsymbol{G}}_{(N)} + NC_1
\end{bmatrix},
\end{aligned}$$

where

$$\tilde{\boldsymbol{G}}_{(i)} = \tilde{\boldsymbol{G}}_{(i)}(\boldsymbol{u}_i^{(k)}) = \operatorname{diag}\begin{pmatrix} 1 & \text{if } u_{h,i,j}^{(k)} < 0, \\ 0 & \text{otherwise.} \end{pmatrix}_{i=1,\ldots,m}$$

Since each block in $\tilde{\boldsymbol{A}}$ is diagonal, the number of nonzero entries per row or column is N. Moreover,

$$\|\tilde{\boldsymbol{A}}\|_1 \leq 2\frac{N-1}{cN} + C_1 \quad \text{and} \quad \|\tilde{\boldsymbol{A}}\|_\infty \leq 2\frac{N-1}{cN} + C_1.$$

Thus, (2.12) yields

$$\|\tilde{\boldsymbol{A}}\| \leq \sqrt{\|\tilde{\boldsymbol{A}}\|_1 \|\tilde{\boldsymbol{A}}\|_\infty} \leq 2\frac{N-1}{cN} + C_1.$$

Hence, we obtain

$$\|\tilde{\boldsymbol{R}}\| \leq \frac{\sqrt{\tau}}{2\sqrt{\varepsilon(1+\tau C_2)}}\left(2\frac{N-1}{cN} + C_1\right) < \frac{\sqrt{\tau}}{2\sqrt{\varepsilon(1+\tau C_2)}}\left(\frac{2}{c} + C_1\right).$$

Therefore, for $\frac{\sqrt{\tau}}{2\sqrt{\varepsilon(1+\tau C_2)}}\left(\frac{2}{c} + C_1\right) \leq \frac{1}{2}$, it holds $\sigma(\tilde{\boldsymbol{R}}) = \sigma(\boldsymbol{R}) \subset B_{0.5}(0)$ and hence $\sigma(\boldsymbol{\mathcal{A}}_{\mathbf{0}}^{-1}\boldsymbol{\mathcal{A}}) \subset B_{0.5}(1)$. If we solve this inequality for C_2, we obtain the condition $C_2 \geq \frac{(C_1+\frac{2}{c})^2}{\varepsilon} - \frac{1}{\tau}$. For small penalty parameters, this bound becomes too large. Another condition arises if we solve the inequality for c. Then, we obtain the inequality $c\left(\frac{\sqrt{\varepsilon(1+\tau C_2)}}{\sqrt{\tau}} - C_1\right) \geq 2$. In order to ensure that the left-hand side is positive, we need $C_2 > \frac{C_1^2}{\varepsilon} - \frac{1}{\tau}$. Finally, we obtain $c > \frac{2\sqrt{\tau}}{\sqrt{\varepsilon(1+\tau C_2)} - C_1\sqrt{\tau}}$. As in the condition before, c is needed to be large. □

In the following, we propose a preconditioner for the case of small penalty parameters c. Similar to the last section, we keep the whole block $\boldsymbol{A}$ within our Schur complement approximation and suggest

$$\begin{aligned}
\hat{\boldsymbol{S}} &= \boldsymbol{S_1}(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}\boldsymbol{S_2} \\
&= \left(\frac{N\sqrt{\varepsilon}}{N-1}(\boldsymbol{I}\otimes\boldsymbol{M}) + \sqrt{\frac{\tau}{1+\tau C_2}}(\boldsymbol{I}\otimes\boldsymbol{K})\right)(\boldsymbol{I}\otimes\boldsymbol{M})^{-1} \\
&\quad \left(\frac{N-1}{N\sqrt{\varepsilon}}(\boldsymbol{I}\otimes\boldsymbol{M}) + \sqrt{\frac{\tau}{1+\tau C_2}}\boldsymbol{A}\right) \qquad (5.83) \\
&= \boldsymbol{I}\otimes\boldsymbol{M} + \frac{\tau}{1+\tau C_2}(\boldsymbol{I}\otimes\boldsymbol{K})(\boldsymbol{I}\otimes\boldsymbol{M})^{-1}\boldsymbol{A} \\
&\quad + \sqrt{\frac{\tau}{\varepsilon(1+\tau C_2)}}\frac{N-1}{N}(\boldsymbol{I}\otimes\boldsymbol{K}) + \sqrt{\frac{\tau\varepsilon}{1+\tau C_2}}\frac{N}{N-1}\boldsymbol{A}. \qquad (5.84)
\end{aligned}$$

The first two terms in (5.84) match the exact Schur complement. Due to the balanced distribution of $\frac{\tau}{1+\tau C_2}$ in form of $\sqrt{\frac{\tau}{1+\tau C_2}}$ as well as the scaling with $\frac{N\sqrt{\varepsilon}}{N-1}$ and its inverse in both factors, $\hat{S}_1$ and $\hat{S}_2$, the influence of both remainder terms in (5.84) is reduced. Let us discuss the action of the inverses of $\boldsymbol{S_1}$ and $\boldsymbol{S_2}$. This is done in the same way as in Chapter 4. The block

$$\boldsymbol{S_1} = \boldsymbol{I}\otimes\left(\frac{N\sqrt{\varepsilon}}{N-1}\boldsymbol{M} + \sqrt{\frac{\tau}{1+\tau C_2}}\boldsymbol{K}\right)$$

in (5.83) is block diagonal and contains the same discrete elliptic operator, $\frac{N\sqrt{\varepsilon}}{(N-1)}\boldsymbol{M} + \sqrt{\frac{\tau}{1+\tau C_2}}\boldsymbol{K}$, on each diagonal block. Therefore, we approximate the inverse of each diagonal block with one and the same AMG preconditioner. The resulting practical approximation of $\boldsymbol{S_1}$ is

$$\boldsymbol{I}\otimes\mathrm{AMG}\left(\frac{N\sqrt{\varepsilon}}{N-1}\boldsymbol{M} + \sqrt{\frac{\tau}{1+\tau C_2}}\boldsymbol{K}\right).$$

The proposed strategy concerning the solution of a system of the form $\boldsymbol{S}_2\boldsymbol{y} = \boldsymbol{g}$ is the use of a block Jacobi method with a fixed number of steps:

$$\boldsymbol{y}^{(l+1)} = \boldsymbol{y}^{(l)} + \omega \boldsymbol{\mathcal{P}}_{\boldsymbol{A}}^{-1}\boldsymbol{r}^{(l)},$$

where

$$\boldsymbol{r}^{(l)} = \boldsymbol{g} - \boldsymbol{S}_2\boldsymbol{y}^{(l)}$$

is the lth residual, and ω is the relaxation parameter. We use the preconditioner

$$\boldsymbol{\mathcal{P}}_{\boldsymbol{A}} = \begin{bmatrix} \mathrm{AMG}\left(\frac{N-1}{N\sqrt{\varepsilon}}\boldsymbol{M} + \sqrt{\frac{\tau}{1+\tau C_2}}\boldsymbol{A}_{(1,1)}\right) & & \\ & \ddots & \\ & & \mathrm{AMG}\left(\frac{N-1}{N\sqrt{\varepsilon}}\boldsymbol{M} + \sqrt{\frac{\tau}{1+\tau C_2}}\boldsymbol{A}_{(N,N)}\right) \end{bmatrix},$$

which is an AMG approximation of the block diagonal matrix of $\boldsymbol{S}_2$. Note that the diagonal blocks of $\boldsymbol{S}_2$ are symmetric positive definite. In contrast to the smooth vector-valued case, we have to initialize N AMG preconditioners instead of one. Moreover, they have to be recomputed in every Newton step since the position of penalized entries is changing with every Newton step. In detail, the penalized entries in the blocks $\boldsymbol{A}_{(i)}$ or $\boldsymbol{A}_{(i,i)}$ depend on the phase $\boldsymbol{u}_i^{(k)}$. Since all phases are separated in the domain (at least after a few time steps), one cannot expect the penalty parameter to act in the same regions for all phases. That is why an approximation of the matrix $\boldsymbol{A}$, where all diagonal blocks are equal, seems not to be of good quality, and our experiences confirm this observation. Nevertheless, Section 5.7.2 shows a promising performance of our developed preconditioner applied with BiCGstab.

Here, we finish the theoretical discussion about the preconditioners. In the next section, we illustrate their efficiency via various numerical experiments. Moreover, we outline an algorithm for the numerical solution of the modified Cahn–Hilliard problems.

5.7 Numerical results

In this section, we show numerical results for the presented modified Cahn–Hilliard problems. First, we explain our implementation framework. This is already described at the beginning of Chapter 3.8 for the most part. Hence, we only add the differences here.

For all inpainting simulations, we use Bertozzi et al's [19] ε-two-step approach. This procedure successfully connects edges across large inpainting domains. In the first step, we run the Cahn–Hilliard inpainting approach close to steady state with a rather large value of the interfacial parameter ε. In the second step, the approximate solution from the first step serves as initial state for a second run of Cahn–Hilliard inpainting. But this time, we set ε to a small value. In the following, we denote by ε_s and $C_{1,s}$ the interfacial and convexity parameter for the second step. In summary, the first step smoothes the image information. Hence, level lines can merge over large damaged regions. The second step sharpens the image contours. If not mentioned otherwise, the stopping criterion in both rounds is

$$||\mathbf{S}(\boldsymbol{u}^{(n)}, \boldsymbol{w}^{(n)})|| \leq \gamma_{\mathrm{rel}}||\mathbf{S}(\boldsymbol{u}^{(0)}, \boldsymbol{w}^{(0)})||_2 + \gamma_{\mathrm{abs}}, \tag{5.85}$$

where $\mathbf{S}(\boldsymbol{u},\boldsymbol{w}) = \left[\mathbf{S}_1(\boldsymbol{u},\boldsymbol{w})^T, \mathbf{S}_2(\boldsymbol{u},\boldsymbol{w})^T\right]^T$ with

$$\begin{aligned}
\mathbf{S}_1(\boldsymbol{u},\boldsymbol{w}) &= -\boldsymbol{A}_S\boldsymbol{u} + (\boldsymbol{I}\otimes\boldsymbol{M})\boldsymbol{w} - \frac{1}{\varepsilon}(\boldsymbol{I}\otimes\boldsymbol{M})\psi'(\boldsymbol{u}) - \frac{1}{\varepsilon N}(\boldsymbol{I}\otimes\boldsymbol{M})\left(\sum_{j=1}^{N}\psi'\left(\boldsymbol{u}_j\right)\right)\mathbf{1},\\
\mathbf{S}_2(\boldsymbol{u},\boldsymbol{w}) &= (\boldsymbol{I}\otimes\boldsymbol{K})\boldsymbol{w} - \omega_0(\boldsymbol{I}\otimes\boldsymbol{H})(\boldsymbol{f}-\boldsymbol{u}),
\end{aligned}$$

with $\boldsymbol{A}_S = \boldsymbol{A} - C_1(\boldsymbol{I}\otimes\boldsymbol{M})$. The potential ψ in $\mathbf{S}_1$ is $\psi = \psi_{\text{pol}}$ in the smooth case and $\psi = \psi_0$ in the nonsmooth case. Regarding the latter, we consider in the following $\boldsymbol{T} = \boldsymbol{I} - \mathbf{1}\mathbf{1}^T$, which is a typical example as mentioned in Chapter 5.1. Basically, $\mathbf{S}_1(\boldsymbol{u},\boldsymbol{w}) = \mathbf{0}$, $\mathbf{S}_2(\boldsymbol{u},\boldsymbol{w}) = \mathbf{0}$ is the discrete steady state formulation of Cahn–Hilliard inpainting. In the scalar case, the terms in (5.85) become

$$\begin{aligned}
\mathbf{S}_1(\boldsymbol{u},\boldsymbol{w}) &= -\boldsymbol{A}_S\boldsymbol{u} + \boldsymbol{M}\boldsymbol{w} - \frac{1}{\varepsilon}\boldsymbol{M}\psi'(\boldsymbol{u}),\\
\mathbf{S}_2(\boldsymbol{u},\boldsymbol{w}) &= \boldsymbol{K}\boldsymbol{w} - \omega_0\boldsymbol{H}(\boldsymbol{f}-\boldsymbol{u}),
\end{aligned}$$

with $\boldsymbol{A}_S = \varepsilon\boldsymbol{K}$, $\psi = \psi_{\text{pol,s}}$ in the smooth case and $\boldsymbol{A}_S = \varepsilon\boldsymbol{K} + c^{-1}\boldsymbol{G}$, $\psi = \psi_{0,s}$ in the nonsmooth case. Based on our experiences, we set $\gamma_{\text{rel}} = 10^{-2}, \gamma_{\text{abs}} = 10^{-1}$ in the smooth (scalar and vector-valued) case and $\gamma_{\text{rel}} = 5\cdot 10^{-2}, \gamma_{\text{abs}} = \sqrt{5\cdot 10^{-2}}$ in the nonsmooth (scalar and vector-valued) case.

Similar to Algorithm 3.1, Algorithm 5.1 summarizes the steps for the numerical solution of the modified Cahn–Hilliard problem with a nonsmooth potential. The formulation with a smooth potential is a simplification of this algorithm. In all experiments, we set the convexity parameters to $C_1 = 3\varepsilon^{-1}$, $C_{1,s} = 3\varepsilon_s^{-1}$, $C_2 = 3\,\omega_0$ and the time step size to $\tau = 1$ if not mentioned otherwise. Remember, ω_0 defined in (5.8) is the fidelity parameter that keeps the inpainted image close enough to the given picture. The initialization of the N phase variables is done in the following preprocessing stage: Given an image f, we segment N clusters using the standard k-means clustering method. The obtained cluster centroid locations serve as the gray values g_i, $i = 1,\dots,N$. Each phase variable represents one cluster and hence describes the evolution of the corresponding gray value. That means every phase variable is set to be one in its corresponding cluster region and zero everywhere else. Moreover, we set every phase variable in the damaged regions to the value N^{-1}. This assignment fulfills the conditions of the Gibbs simplex (5.2). In the scalar case, we initialize the phase variable in the damaged regions with the value zero. At the moment, we perform the preprocessing stage in MATLAB® with the command `kmeans`. We load images in C++ using the CImg[3] Library version 1.5.3. The final reconstructed image f_r is obtained by

$$f_r = \sum_{i=1}^{N} g_i u_i^{(T)}, \tag{5.86}$$

where $\boldsymbol{u}^{(T)} = [u_1^{(T)},\dots,u_N^{(T)}]^T$ represents the final phase variable. In the scalar case, the final reconstructed image is given by the final phase variable $u^{(T)}$.

Now, we are ready for numerical results.

[3] http://cimg.eu/

Algorithm 5.1: The numerical solution of the nonsmooth (vector-valued) modified Cahn–Hilliard problem via an SSN method combined with a Moreau–Yosida regularization technique on a uniform mesh.

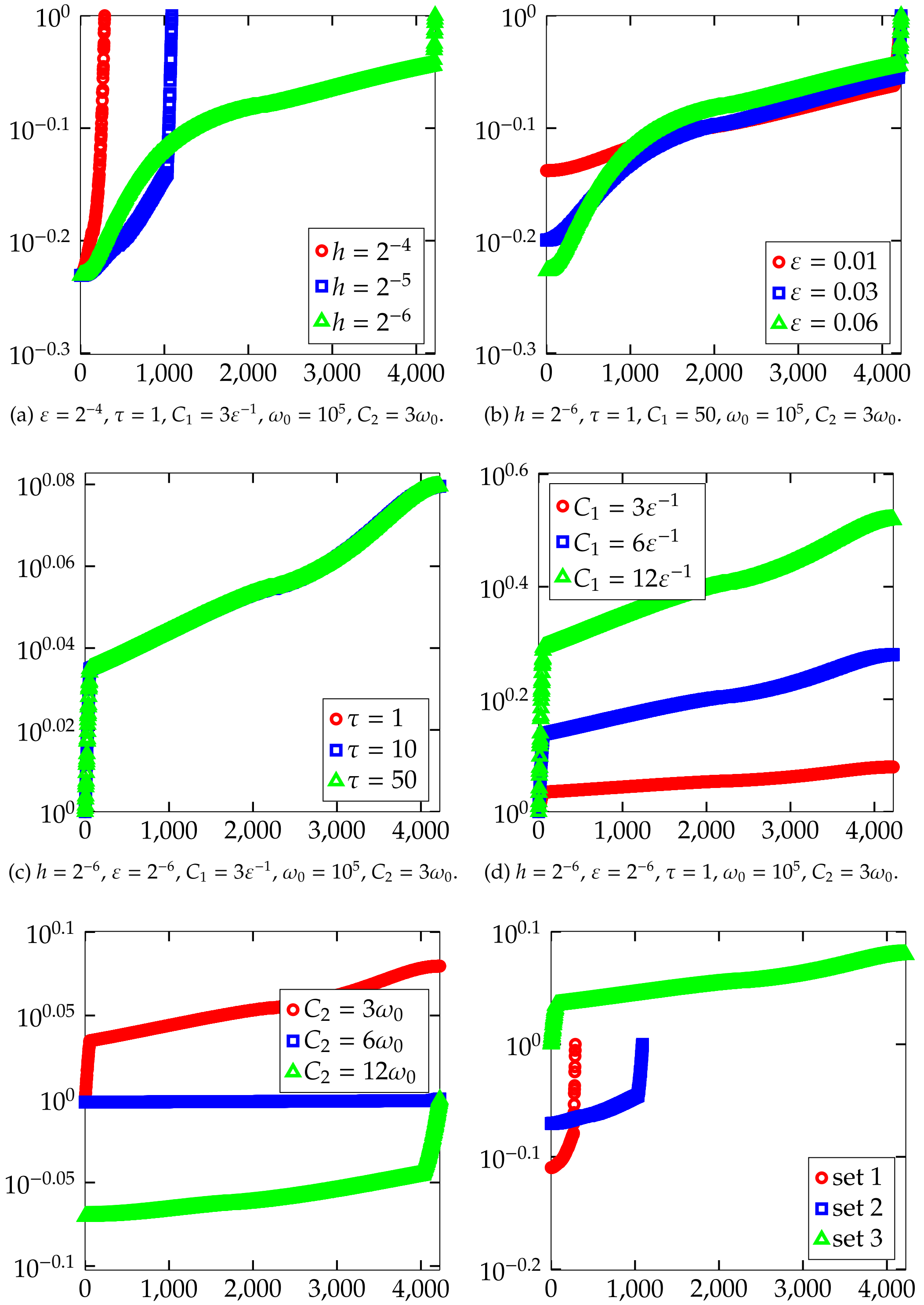
```
Choose $h, \varepsilon, \varepsilon_s, \tau, N, \omega_0, C_1, C_{1,s}, C_2, c_1, c_2, \dots, c_{p_{\max}}, n_c, \epsilon_{\text{rel}}, \epsilon_{\text{abs}}$
Build the spatial mesh
Initialize $\boldsymbol{M}, \boldsymbol{K}$, and the AMG solver for $\boldsymbol{S_1}$
Locate the inpainting domain $D$
Set $\boldsymbol{u}^{(0)} = \mathbf{f}$ on $\Omega \setminus D$ and $\boldsymbol{u}_i^{(0)} = 1/N$ for $i = 1, \dots, N$ on $D$
Set $\boldsymbol{w}^{(0)}$
$n = 1$, run=1
while run=1 || (run=2 && not close to steady state) do
    if run=1 && close to steady state then
        $\varepsilon = \varepsilon_s$
        $C_1 = C_{1,s}$
        Update the AMG solver for $\boldsymbol{S_1}$
        run=2
    end
    Update the right-hand side of the linear system
    for $p = 1, 2, \dots, p_{\max}$ do
        if $n > n_c$ then
            $p = p_{\max}$
        end
        $c = c_p$
        if $p = 1$ or $n > n_c$ then
            Set $\boldsymbol{u}^{(n,p,0)} = \boldsymbol{u}^{(n-1)}, \boldsymbol{w}^{(n,p,0)} = \boldsymbol{w}^{(n-1)}$
        else
            Set $\boldsymbol{u}^{(n,p,0)} = \boldsymbol{u}^{(n,p-1)}, \boldsymbol{w}^{(n,p,0)} = \boldsymbol{w}^{(n,p-1)}$
        end
        for $k = 0, 1, 2, \dots$ until convergence do
            Update the block $\boldsymbol{A}$
            Update the AMG solver for $\boldsymbol{S_2}$
            Solve the linear system and obtain $\boldsymbol{u}^{(n,p,k+1)}, \boldsymbol{w}^{(n,p,k+1)}$
            if $||F_{c,h}(\boldsymbol{u}^{(n,p,k+1)}, \boldsymbol{w}^{(n,p,k+1)})|| \leq \epsilon_{\text{rel}}||F_{c,h}(\boldsymbol{u}^{(n,p,0)}, \boldsymbol{w}^{(n,p,0)})||_2 + \epsilon_{\text{abs}}$ then
                Set $\boldsymbol{u}^{(n,p)} = \boldsymbol{u}^{(n,p,k+1)}, \boldsymbol{w}^{(n,p)} = \boldsymbol{w}^{(n,p,k+1)}$
                break
            end
        end
    end
    Set $\boldsymbol{u}^{(n)} = \boldsymbol{u}^{(n,p_{\max})}, \boldsymbol{w}^{(n)} = \boldsymbol{w}^{(n,p_{\max})}$
    $n = n + 1$
end
```

5.7.1 Eigenvalue plots

In Section 5.6, we have developed different Schur complement approximations and referred to this section for some corresponding eigenvalue plots. In particular, we show eigenvalue plots for the scalar smooth and nonsmooth Cahn–Hilliard inpainting model discussed in Section 5.6.1 and 5.6.2. The following eigenvalue plots are simply generated with MATLAB®. The mass and stiffness matrix $\boldsymbol{M}$ and $\boldsymbol{K}$ are generated in C++ using the FEM library deal.II [8] as described in Chapter 3.8. For the following simple demonstrations, we consider uniform refinements of the unit square $[0,1]^2$ with three different mesh sizes $h_i = 2^{-i-3}$ for $i = 1,2,3$. Let us denote the diagonal matrix $\boldsymbol{G}$ in (5.62) for each mesh by $\boldsymbol{G}^{(i)}$ for $i = 1,2,3$. It is implemented in MATLAB as a random vector with MATLAB's command `randperm`. First, we initialize three vectors $\boldsymbol{g_i} = [p_1,\dots,p_{m_i}]^T \in \mathbb{R}^{m_i}$ as one vectors, where $m_i = (h_i^{-1}+1)^2$ for $i = 1,2,3$. Then, we set randomly 25 percent of each vector $\boldsymbol{g_i}$ to zero via `randperm`. The diagonal values of $\boldsymbol{G}^{(i)}$ are then set to be $[\boldsymbol{G}^{(i)}]_{jj} = [\boldsymbol{M}]_{jj}p_j$ for $j = 1,\dots m_i$. The action of all inverses are performed with MATLAB's `backslash` command. This is a direct solver based on the LU-factorization, which works well for our small sized two-dimensional problems. In total, three inverses occur in the implementation: One in the Schur complement $\boldsymbol{S}$, one in the Schur complement approximation $\hat{\boldsymbol{S}}$, as well as one in $\hat{\boldsymbol{S}}^{-1}\boldsymbol{S}$. Finally, we have used MATLAB's `eigs` command to obtain the eigenvalues of the generated matrix $\hat{\boldsymbol{S}}^{-1}\boldsymbol{S}$.

We start with the smooth system (5.63) with the Schur complement approximation (5.65). Each subplot in Figure 5.1(a)–5.1(e) demonstrates the robustness with respect to a different model parameter. In Figure 5.1(a), we vary the mesh size h while fixing $\varepsilon = 2^{-4}$, $\tau = 1$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^5$, $C_2 = 3\omega_0$. In Figure 5.1(b), we vary the interfacial parameter ε while fixing $h = 2^{-6}$, $\tau = 1$, $C_1 = 50$, $\omega_0 = 10^5$, $C_2 = 3\omega_0$. In Figure 5.1(c), we vary the time step size τ while fixing $h = 2^{-6}$, $\varepsilon = 2^{-6}$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^5$, $C_2 = 3\omega_0$. In Figure 5.1(d), we vary the convexity parameter C_1 while fixing $h = 2^{-6}$, $\varepsilon = 2^{-6}$, $\tau = 1$, $\omega_0 = 10^5$, $C_2 = 3\omega_0$. In Figure 5.1(e), we vary the convexity parameter C_2 while fixing $h = 2^{-6}, \varepsilon = 2^{-6}, \tau = 1, C_1 = 3\varepsilon^{-1}, \omega_0 = 10^5$. Finally, in Figure 5.1(f) we vary simultaneously all five parameters $h, \varepsilon, C_1, \omega_0, C_2$ while fixing $\tau = 1$. Each of the six subplots illustrates nicely the eigenvalue clustering. Moreover, all eigenvalues are real and positive as expected from Lemma 2.30.

Next, we go over to the nonsmooth system (5.75) with the Schur complement approximation (5.77). Each row in Figure 5.2–5.3 demonstrates the robustness with respect to a different model parameter. In all tests, we observe the appearance of complex eigenvalues. In Figure 5.2(a)–5.2(b), we vary the mesh size h while fixing $\varepsilon = 2^{-4}$, $\tau = 1$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^5$, $C_2 = 3\omega_0$, $c = 10^{-7}$. In Figure 5.2(c)–5.2(d), we vary the interfacial parameter ε while fixing $h = 2^{-6}$, $\tau = 1$, $C_1 = 300$, $\omega_0 = 10^5$, $C_2 = 3\omega_0$, $c = 10^{-7}$. In Figure 5.2(e)–5.2(f), we vary the time step size τ while fixing $h = 2^{-6}$, $\varepsilon = 2^{-6}$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^5$, $C_2 = 3\omega_0$, $c = 10^{-7}$. In Figure 5.3(a)–5.3(b), we vary the convexity parameter C_1 while fixing $h = 2^{-6}$, $\varepsilon = 2^{-6}$, $\tau = 1$, $\omega_0 = 10^5$, $C_2 = 3\omega_0$, $c = 10^{-7}$. In Figure 5.3(c)–5.3(d), we vary the convexity parameter C_2 while fixing $h = 2^{-6}$, $\varepsilon = 2^{-6}$, $\tau = 1$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^5$, $c = 10^{-7}$. In Figure 5.3(e)–5.3(f), we vary the penalty parameter c while fixing $h = 2^{-6}$, $\varepsilon = 2^{-6}$, $\tau = 1$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^5$, $C_2 = 3\omega_0$. Finally, in Figure 5.4, we vary simultaneously all five parameters $h, \varepsilon, C_1, \omega_0, C_2$ while fixing $\tau = 1$, $c = 10^{-7}$. Each subplot illustrates nicely the eigenvalue clustering.



(a) $\varepsilon = 2^{-4}$, $\tau = 1$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^5$, $C_2 = 3\omega_0$.

(b) $h = 2^{-6}$, $\tau = 1$, $C_1 = 50$, $\omega_0 = 10^5$, $C_2 = 3\omega_0$.

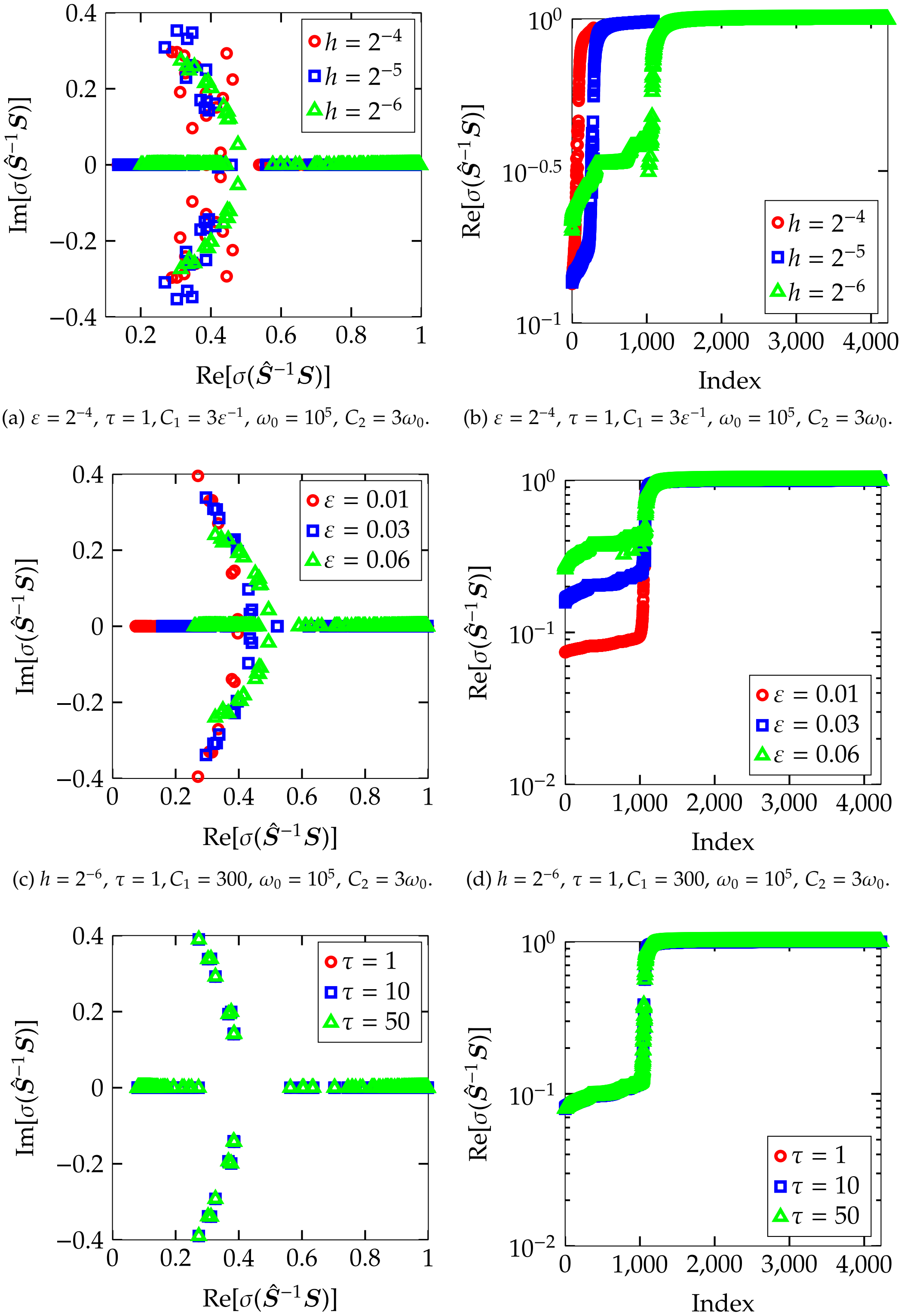
(c) $h = 2^{-6}$, $\varepsilon = 2^{-6}$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^5$, $C_2 = 3\omega_0$.

(d) $h = 2^{-6}$, $\varepsilon = 2^{-6}$, $\tau = 1$, $\omega_0 = 10^5$, $C_2 = 3\omega_0$.

(e) $h = 2^{-6}$, $\varepsilon = 2^{-6}$, $\tau = 1$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^5$.

(f) Set j: $h_j = 2^{-j-3}$, $\varepsilon_j = h_j$, $\tau = 1$, $C_1^{(j)} = 3\varepsilon_j^{-1}$, $\omega_0^{(1)} = 10^4$, $\omega_0^{(2)} = 5 \cdot 10^4$, $\omega_0^{(3)} = 10^5$, $C_2^{(j)} = 3\omega_0^{(j)}$ for $j = 1, 2, 3$.

Figure 5.1: Spectrum of $\hat{S}^{-1}S$ for the scalar smooth system (5.63) with the Schur complement approximation (5.65).



(a) $\varepsilon = 2^{-4}$, $\tau = 1, C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^5$, $C_2 = 3\omega_0$.

(b) $\varepsilon = 2^{-4}$, $\tau = 1, C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^5$, $C_2 = 3\omega_0$.

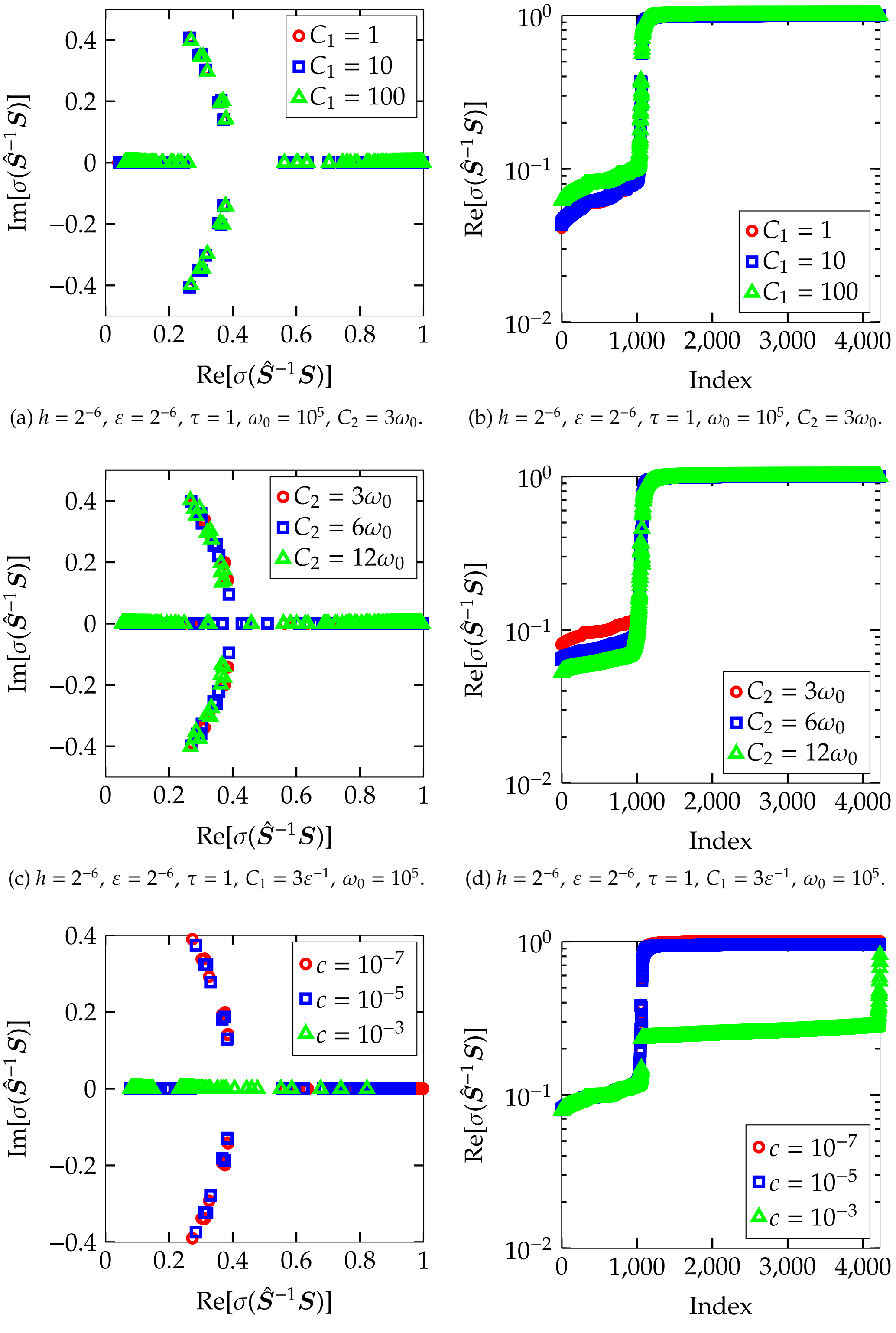
(c) $h = 2^{-6}$, $\tau = 1, C_1 = 300$, $\omega_0 = 10^5$, $C_2 = 3\omega_0$.

(d) $h = 2^{-6}$, $\tau = 1, C_1 = 300$, $\omega_0 = 10^5$, $C_2 = 3\omega_0$.

(e) $h = 2^{-6}$, $\varepsilon = 2^{-6}$, $C_1 = 3\varepsilon^{-1}$ $\omega_0 = 10^5$, $C_2 = 3\omega_0$.

(f) $h_j = 2^{-j-3}$, $\varepsilon_j = h_j$, $\tau = 1$, $C_1 = 3\varepsilon^{-1}$, $\omega_0^{(1)} = 10^4$, $\omega_0^{(2)} = 5 \cdot 10^4$, $\omega_0^{(3)} = 10^5$, $C_2^{(j)} = 3\omega_0^{(j)}$ for $j = 1, 2, 3$.

Figure 5.2: Spectrum of $\hat{S}^{-1}S$ for the scalar nonsmooth system (5.75) with the Schur complement approximation (5.77).



(a) $h = 2^{-6}$, $\varepsilon = 2^{-6}$, $\tau = 1$, $\omega_0 = 10^5$, $C_2 = 3\omega_0$.

(b) $h = 2^{-6}$, $\varepsilon = 2^{-6}$, $\tau = 1$, $\omega_0 = 10^5$, $C_2 = 3\omega_0$.

(c) $h = 2^{-6}$, $\varepsilon = 2^{-6}$, $\tau = 1$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^5$.

(d) $h = 2^{-6}$, $\varepsilon = 2^{-6}$, $\tau = 1$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^5$.

(e) $h = 2^{-6}$, $\varepsilon = 2^{-6}$, $\tau = 1$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^5$, $C_2 = 3\omega_0$.

(f) $h = 2^{-6}$, $\varepsilon = 2^{-6}$, $\tau = 1$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^5$, $C_2 = 3\omega_0$.

Figure 5.3: Spectrum of $\hat{S}^{-1}S$ for the scalar nonsmooth system (5.75) with the Schur complement approximation (5.77).

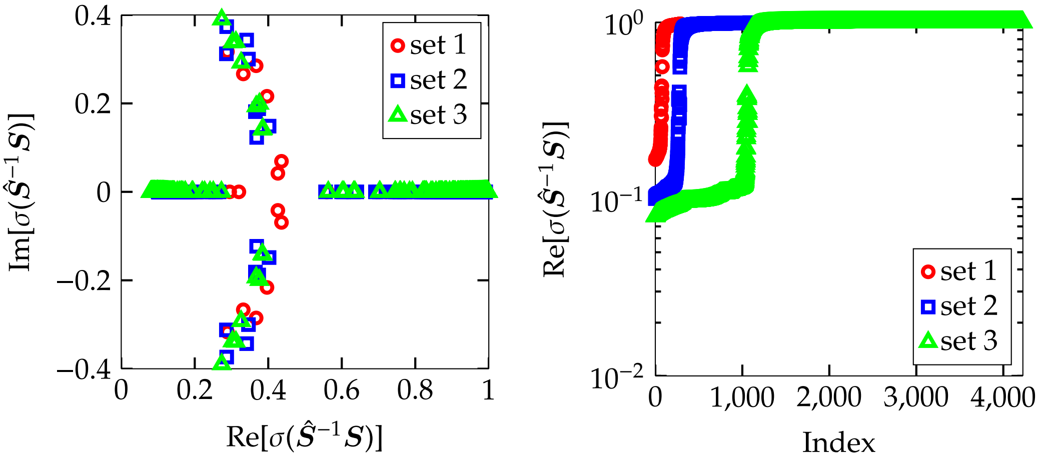
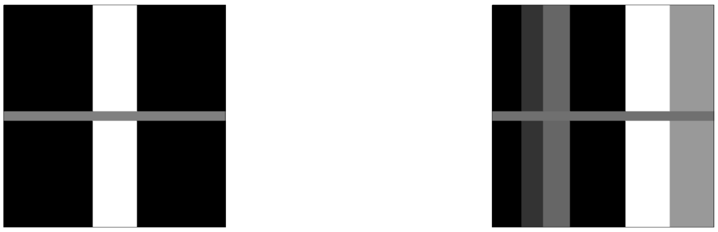


Figure 5.4: Spectrum of $\hat{S}^{-1}S$ for the scalar nonsmooth system (5.75) with the Schur complement approximation (5.77). Set j: $h_j = 2^{-j-3}$, $\varepsilon_j = h_j$, $\tau = 1$, $C_1^{(j)} = 3\varepsilon_j^{-1}$, $\omega_0^{(1)} = 10^4$, $\omega_0^{(2)} = 5 \cdot 10^4$, $\omega_0^{(3)} = 10^5$, $C_2^{(j)} = 3\omega_0^{(j)}$, $c = 10^{-7}$ for $j = 1, 2, 3$

5.7.2 Robustness

In this section, we demonstrate the robustness of our proposed preconditioners regarding all model parameters. The initial state in the scalar models is an image consisting of a vertical white stripe, where the inpainting domain is given by a horizontal stripe, see Figure 5.5(a). The initial state in the vector-valued models is an image consisting of several vertical gray value stripes, where the inpainting domain is given by a horizontal stripe. In the case $N = 5$, we use six vertical stripes as given in Figure 5.5(b). When we test the robustness with respect to N, we increase the number of vertical stripes for larger values of N.



(a) Scalar model.

(b) Vector-valued model with $N = 5$.

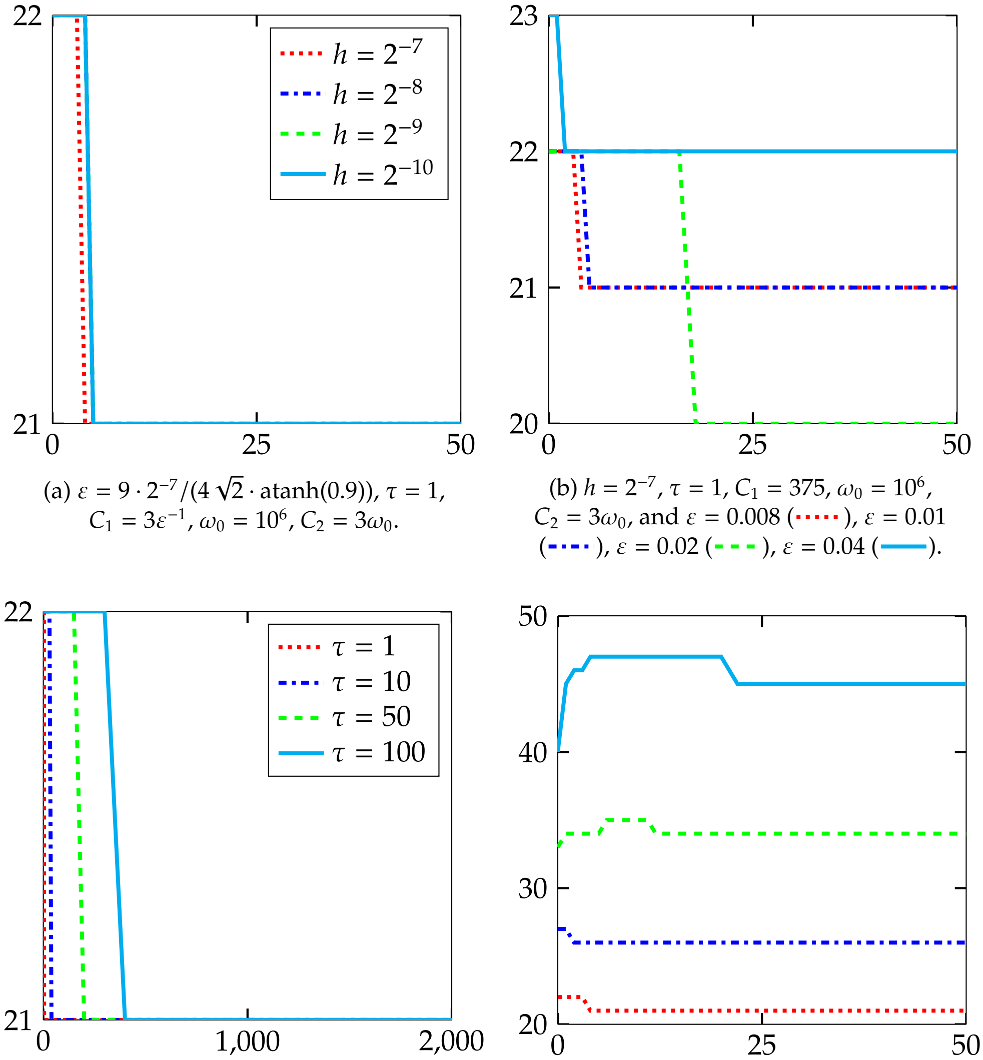
Figure 5.5: Initial images for the robustness tests of our preconditioners developed for Cahn–Hilliard inpainting.

We start with the scalar smooth system in (5.63) with the preconditioner (5.64) and the Schur complement approximation (5.65). Each subplot in Figure 5.6(a)–5.7(a) demonstrates the robustness with respect to a different model parameter. In Figure 5.6(a), we vary the mesh size h while fixing $\varepsilon = 9 \cdot 2^{-7}/(4\sqrt{2} \cdot \mathrm{atanh}(0.9))$, $\tau = 1$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, and $T = 50$. In Figure 5.6(b), we vary the interfacial parameter ε while fixing $h = 2^{-7}$, $\tau = 1$, $C_1 = 375$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, and $T = 50$. In Figure 5.6(c), we vary the time step size τ while fixing $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/(4\sqrt{2} \cdot \operatorname{atanh}(0.9))$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, and $T = 2000$. In Figure 5.6(d), we vary the convexity parameter C_1 while fixing $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/(4\sqrt{2} \cdot \operatorname{atanh}(0.9))$, $\tau = 1$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, and $T = 50$. We observe a benign increase if iteration numbers when C_1 is increased. However, the larger C_1 is the slower is the evolution. Hence, in praxis, one chooses C_1 as small as possible such that the convexity condition in Lemma 5.1 is satisfied. In Figure 5.7(a), we vary the convexity parameter C_2 while fixing $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/(4\sqrt{2} \cdot \operatorname{atanh}(0.9))$, $\tau = 1$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^6$, and $T = 50$. In fact, we observe a decrease of iteration numbers when C_2 is increased. Finally, in Figure 5.7(b), we vary simultaneously all five parameters $h, \varepsilon, C_1, C_2, \omega$ while fixing $\tau = 1$ and $T = 50$. Table 5.1 illustrates the maximum and average number of MINRES iterations, the average CPU time (in seconds) for MINRES, and the CPU time (in seconds) for the whole simulation for each of the six subplots, respectively.

Simulation		MINRES			
Figure	Plot	Max	Avg	CPU (s)	CPU (s)
5.6(a)	(⋯⋯)	22	21	8	399
	(-·-·)	22	21	30	1595
	(- - -)	22	21	147	7587
	(——)	22	21	583	29881
5.6(b)	(⋯⋯)	22	21	8	402
	(-·-·)	22	21	8	402
	(- - -)	22	21	7	392
	(——)	23	22	8	401
5.6(c)	(⋯⋯)	22	21	7	14899
	(-·-·)	22	21	7	1544
	(- - -)	22	21	8	319
	(——)	22	21	7	159
5.6(d)	(⋯⋯)	22	21	8	408
	(-·-·)	27	26	9	489
	(- - -)	35	34	12	626
	(——)	47	46	15	802
5.7(a)	(⋯⋯)	22	21	8	398
	(-·-·)	21	20	7	378
	(- - -)	19	17	6	327
	(——)	18	15	6	295
5.7(b)	(⋯⋯)	22	21	7	389
	(-·-·)	23	22	32	1658
	(- - -)	29	28	196	10060

Table 5.1: Results for the solution of the scalar smooth system (5.63) with the preconditioner (5.64) and the Schur complement approximation (5.65): The maximum and average number of MINRES iterations, the average CPU time (in seconds) for MINRES, and the CPU time (in seconds) for the whole simulation.

We proceed with the scalar nonsmooth system in (5.75) with the preconditioner (5.76) and the Schur complement approximation (5.77). Each subplot in Figure 5.8(a)–5.9(b) demonstrates the robustness with respect to a different model parameter. In Figure 5.8(a), we vary the mesh size h while fixing $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 1$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, $c_{p_{\max}} = 10^{-7}$, and $T = 20$. In Figure 5.8(b), we vary the interfacial parameter ε while fixing $h = 2^{-7}$, $\tau = 1$, $C_1 = 150$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, $c_{p_{\max}} = 10^{-7}$, and $T = 20$. In Figure 5.8(c), we vary the time step size τ while fixing $h = 2^{-7}$, $\varepsilon =$



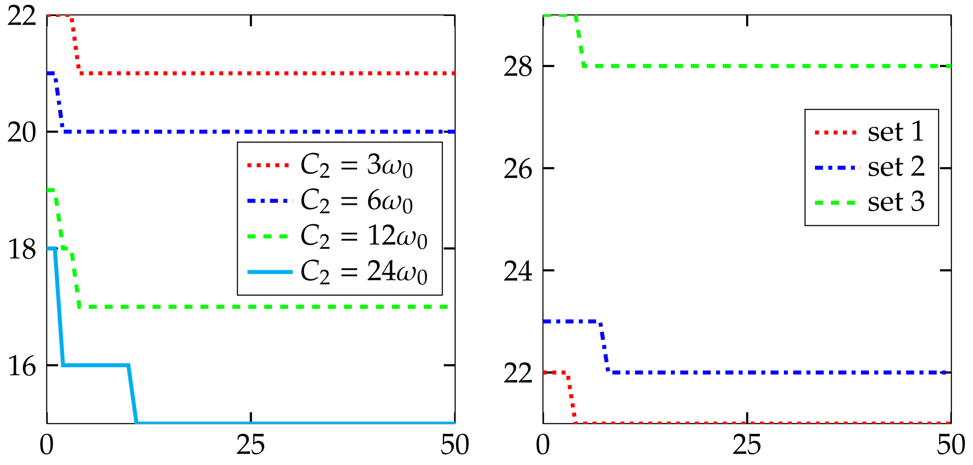
(a) $\varepsilon = 9 \cdot 2^{-7}/(4\sqrt{2} \cdot \text{atanh}(0.9))$, $\tau = 1$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$.

(b) $h = 2^{-7}$, $\tau = 1$, $C_1 = 375$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, and $\varepsilon = 0.008$ (·····), $\varepsilon = 0.01$ (-·-·-), $\varepsilon = 0.02$ (- - -), $\varepsilon = 0.04$ (——).

(c) $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/(4\sqrt{2} \cdot \text{atanh}(0.9))$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$.

(d) $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/(4\sqrt{2} \cdot \text{atanh}(0.9))$, $\tau = 1$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, and $C_1 = 3\varepsilon^{-1}$ (·····), $C_1 = 6\varepsilon^{-1}$ (-·-·-), $C_1 = 12\varepsilon^{-1}$ (- - -), $C_1 = 24\varepsilon^{-1}$ (——).

Figure 5.6: Results for the solution of the scalar smooth system (5.63) with the preconditioner (5.64) and the Schur complement approximation (5.65). The x-axis shows the time t and the y-axis the number of MINRES iterations.



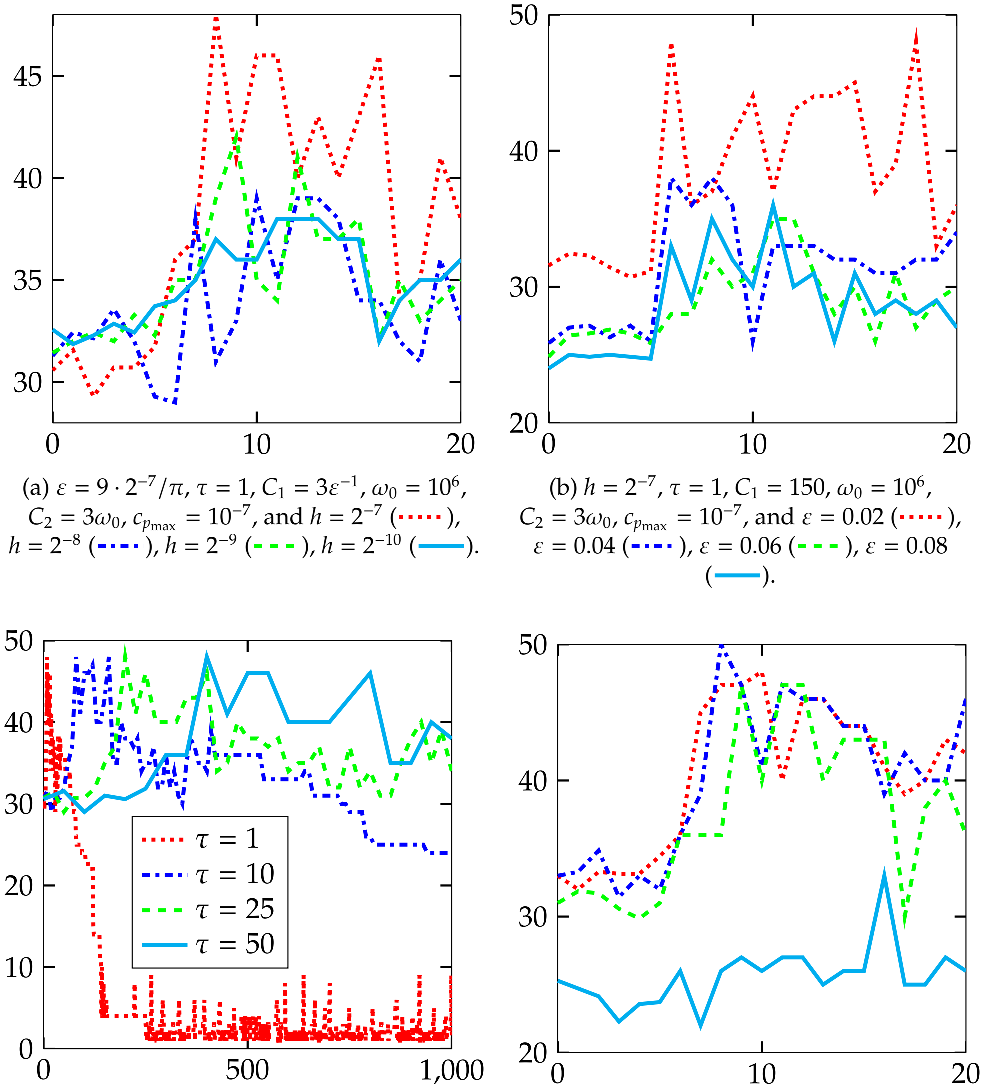
(a) $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/(4\sqrt{2} \cdot \text{atanh}(0.9))$, $\tau = 1$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^6$.

(b) Set j: $h_j = 2^{-6-j}$, $\varepsilon = 9\,h_j/(4\sqrt{2} \cdot \text{atanh}(0.9))$, $\tau = 1$, $C_1^{(j)} = 3\varepsilon_j^{-1}$, $\omega_0^{(1)} = 10^6$, $\omega_0^{(2)} = 5 \cdot 10^6$, $\omega_0^{(3)} = 10^7$, $C_2^{(j)} = 3\omega_0^{(j)}$ for $j = 1, 2, 3$.

Figure 5.7: Results for the solution of the scalar smooth system (5.63) with the preconditioner (5.64) and the Schur complement approximation (5.65). The x-axis shows the time t and the y-axis the number of MINRES iterations.

$9 \cdot 2^{-7}/\pi$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, $c_{p_{\max}} = 10^{-7}$, and $T = 1000$. In Figure 5.8(d), we vary the convexity parameter C_1 while fixing $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 1$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, $c_{p_{\max}} = 10^{-7}$, and $T = 20$. In fact, we observe a decrease of iteration numbers when C_1 is increased. In Figure 5.9(a), we vary the convexity parameter C_2 while fixing $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 1$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^6$, $c_{p_{\max}} = 10^{-7}$, and $T = 20$. In Figure 5.9(b), we vary the penalty parameter $c_{p_{\max}}$ while fixing $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 1$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, and $T = 20$. Finally, in Figure 5.9(c), we vary simultaneously all five parameters $h, \varepsilon, C_1, C_2, \omega$ while fixing $\tau = 1$, $c_{p_{\max}} = 10^{-7}$, and $T = 20$. Table 5.2 illustrates the maximum and average number of BiCG iterations, the average CPU time (in seconds) for BiCG, and the CPU time (in seconds) for the whole simulation for each of the seven subplots, respectively.

Next, we consider the vector-valued smooth system represented in (5.59) for which we apply the same preconditioning technique as in the scalar smooth case. Each subplot in Figure 5.10(a)–5.11(b) demonstrates the robustness with respect to a different model parameter. In Figure 5.10(a), we vary the mesh size h while fixing $\varepsilon = 9 \cdot 2^{-7}/(4\sqrt{2} \cdot \text{atanh}(0.9))$, $\tau = 1$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, $N = 5$, and $T = 50$. In Figure 5.10(b), we vary the interfacial parameter ε while fixing $h = 2^{-7}$, $\tau = 1$, $C_1 = 375$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, $N = 5$, and $T = 50$. In Figure 5.10(c), we vary the time step size τ while fixing $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/(4\sqrt{2} \cdot \text{atanh}(0.9))$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, $N = 5$, and $T = 2000$. In Figure 5.10(d), we vary the convexity parameter C_1 while fixing $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/(4\sqrt{2} \cdot \text{atanh}(0.9))$, $\tau = 1$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, $N = 5$, and $T = 50$. As in the scalar smooth case, we observe a benign increase if iteration numbers when C_1 is increased. However, the larger C_1 is the slower is the evolution. Hence, in praxis, one chooses C_1 as small as possible such that the convexity condition in Lemma 5.1 is satisfied. In Figure 5.11(a), we vary the convexity parameter C_2 while fixing $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/(4\sqrt{2} \cdot \text{atanh}(0.9))$, $\tau = 1$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^6$, $N = 5$,



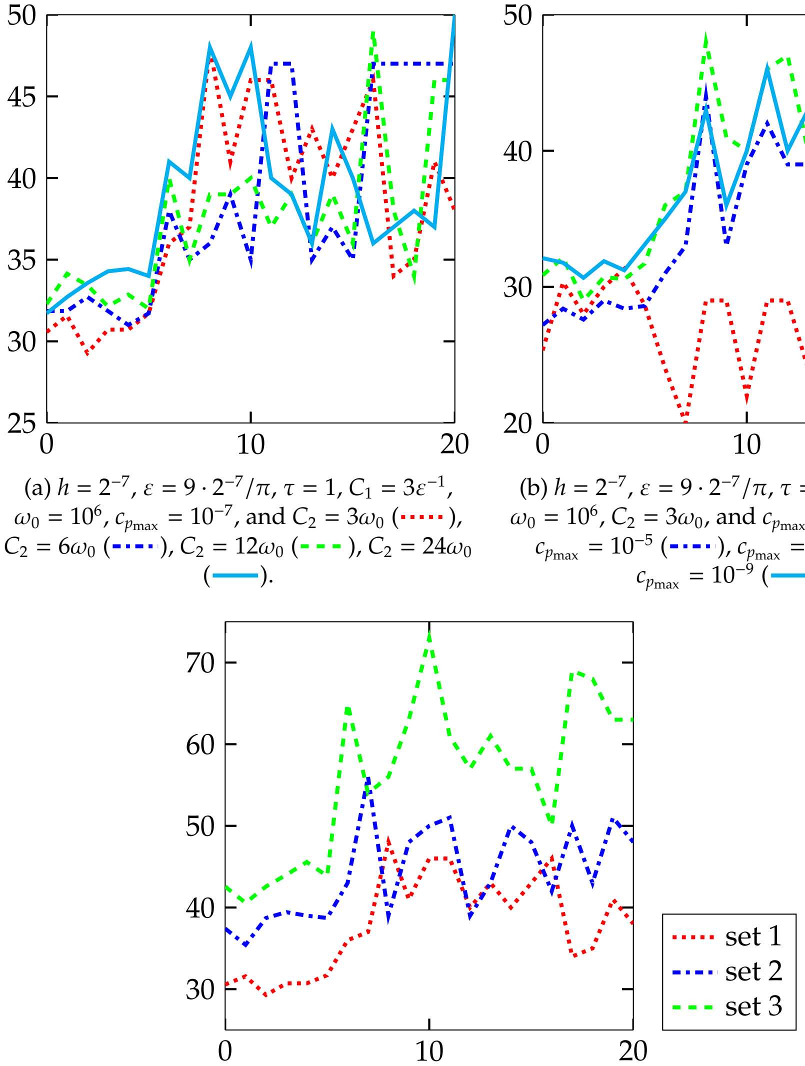
(a) $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 1$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, $c_{p_{\max}} = 10^{-7}$, and $h = 2^{-7}$ (⋯⋯), $h = 2^{-8}$ (-·-·-), $h = 2^{-9}$ (- - -), $h = 2^{-10}$ (——).

(b) $h = 2^{-7}$, $\tau = 1$, $C_1 = 150$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, $c_{p_{\max}} = 10^{-7}$, and $\varepsilon = 0.02$ (⋯⋯), $\varepsilon = 0.04$ (-·-·-), $\varepsilon = 0.06$ (- - -), $\varepsilon = 0.08$ (——).

(c) $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, $c_{p_{\max}} = 10^{-7}$.

(d) $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 1$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, $c_{p_{\max}} = 10^{-7}$, and $C_1 = 1$ (⋯⋯), $C_1 = 10$ (-·-·-), $C_1 = 100$ (- - -), $C_1 = 1000$ (——).

Figure 5.8: Results for the solution of the scalar nonsmooth system (5.75) with the preconditioner (5.76) and the Schur complement approximation (5.77). The x-axis shows the time t and the y-axis the average number of BiCG iterations per SSN step.



(a) $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 1$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^6$, $c_{p_{\max}} = 10^{-7}$, and $C_2 = 3\omega_0$ (····), $C_2 = 6\omega_0$ (-·-·), $C_2 = 12\omega_0$ (- - -), $C_2 = 24\omega_0$ (—).

(b) $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 1$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, and $c_{p_{\max}} = 10^{-3}$ (····), $c_{p_{\max}} = 10^{-5}$ (-·-·), $c_{p_{\max}} = 10^{-7}$ (- - -), $c_{p_{\max}} = 10^{-9}$ (—).

(c) Set j: $h_j = 2^{-6-j}$, $\varepsilon = 9\,h_j/\pi$, $\tau = 1$, $C_1^{(j)} = 3\varepsilon_j^{-1}$, $\omega_0^{(1)} = 10^6$, $\omega_0^{(2)} = 5 \cdot 10^6$, $\omega_0^{(3)} = 10^7$, $C_2^{(j)} = 3\omega_0^{(j)}$, $c_{p_{\max}} = 10^{-7}$ for $j = 1, 2, 3$.

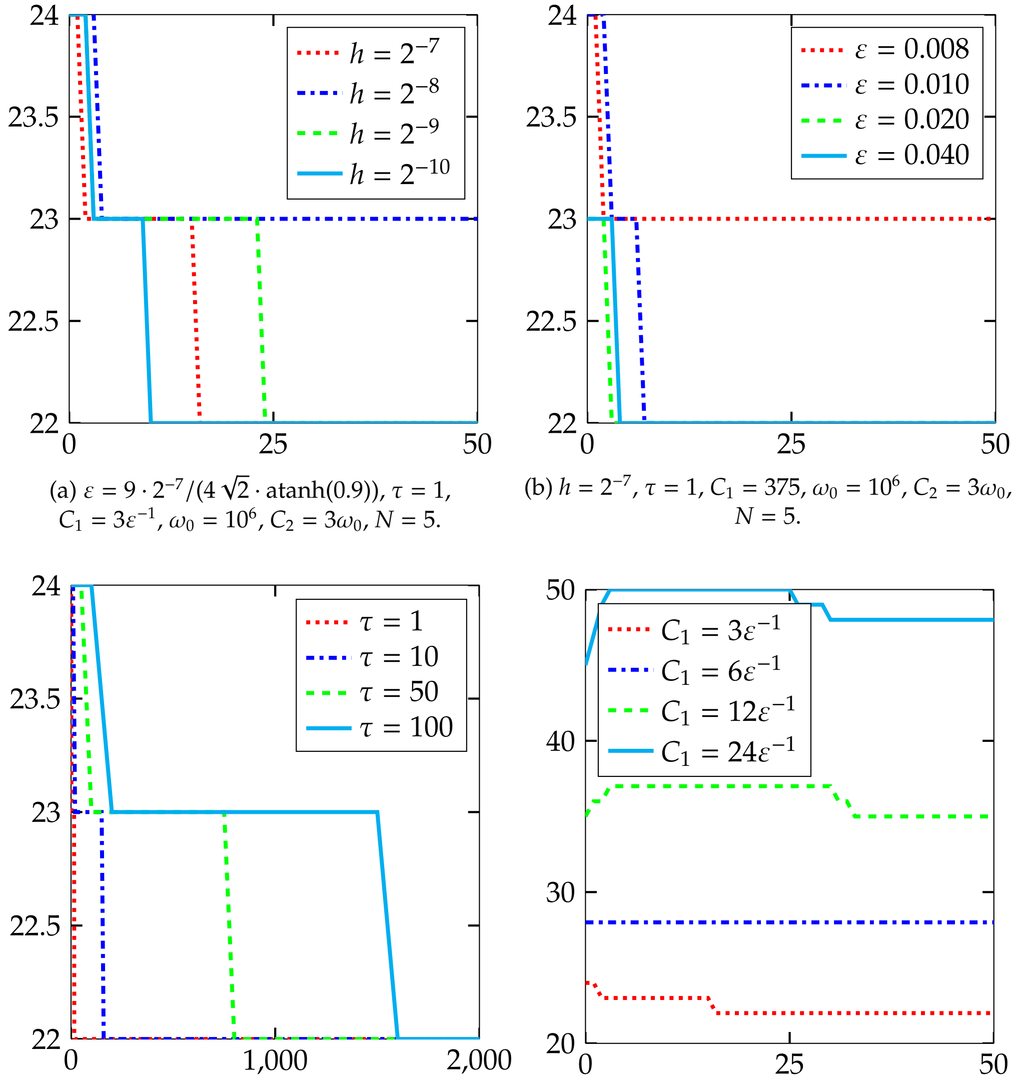
Figure 5.9: Results for the solution of the scalar nonsmooth system (5.75) with the preconditioner (5.76) and the Schur complement approximation (5.77). The x-axis shows the time t and the y-axis the average number of BiCG iterations per SSN step.

Simulation		SSN		BiCG			
Figure	Plot	Max	Avg	Max	Avg	CPU (s)	CPU (s)
5.8(a)	(·····)	5	2	48	30	14	2184
	(-·-·)	5	2	48	29	24	7618
	(- - -)	5	3	51	30	101	42185
	(——)	5	3	49	30	331	177432
5.8(b)	(·····)	5	2	50	30	8	1652
	(-·-·)	5	2	38	25	9	1704
	(- - -)	5	2	37	24	8	1640
	(——)	4	2	36	23	7	1284
5.8(c)	(·····)	5	1	48	7	6	11086
	(-·-·)	5	2	48	31	21	5355
	(- - -)	5	2	48	31	15	3375
	(——)	5	2	48	30	14	2176
5.8(d)	(·····)	6	2	52	32	13	2435
	(-·-·)	6	2	50	32	14	2325
	(- - -)	5	2	50	30	13	2273
	(——)	3	2	34	21	9	1603
5.9(a)	(·····)	5	2	48	30	14	2186
	(-·-·)	6	2	50	30	13	2311
	(- - -)	9	2	50	30	12	2352
	(——)	10	2	50	31	14	2255
5.9(b)	(·····)	5	2	31	24	7	691
	(-·-·)	5	2	44	27	8	1154
	(- - -)	5	2	48	30	14	2185
	(——)	5	2	47	30	11	2271
5.9(c)	(·····)	5	2	48	30	14	2201
	(-·-·)	8	2	66	35	42	10668
	(- - -)	4	2	80	42	191	45737

Table 5.2: Results for the solution of the scalar nonsmooth system (5.75) with the preconditioner (5.76) and the Schur complement approximation (5.77): The maximum and average number of SSN iterations, the maximum and average number of BiCG iterations, the average CPU time (in seconds) for BiCG, and the CPU time (in seconds) for the whole simulation.

and $T = 50$. In fact, we observe a decrease of iteration numbers when C_2 is increased. In Figure 5.11(b), we vary the number of phases N while fixing $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/(4\sqrt{2} \cdot \operatorname{atanh}(0.9))$, $\tau = 1$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, and $T = 50$. Finally, in Figure 5.11(c), we vary simultaneously all five parameters $h, \varepsilon, C_1, C_2, \omega$ while fixing $\tau = 1$, $N = 5$, and $T = 50$. Table 5.3 illustrates the maximum and average number of MINRES iterations, the average CPU time (in seconds) for MINRES, and the CPU time (in seconds) for the whole simulation for each of the eight subplots, respectively.

Finally, we consider the vector-valued nonsmooth system in (5.79) with the preconditioner (5.80) and the Schur complement approximation (5.83). Each subplot in Figure 5.12(a)–5.13(c) demonstrates the robustness with respect to a different model parameter. In Figure 5.12(a), we vary the mesh size h while fixing $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 1$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, $N = 5$, $c_{p_{\max}} = 10^{-7}$, and $T = 20$. In fact, we observe a decrease of iteration numbers when the mesh size is refined. In Figure 5.12(b), we vary the interfacial parameter ε while fixing $h = 2^{-7}$, $\tau = 1$, $C_1 = 150$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, $N = 5$, $c_{p_{\max}} = 10^{-7}$, and $T = 20$. In



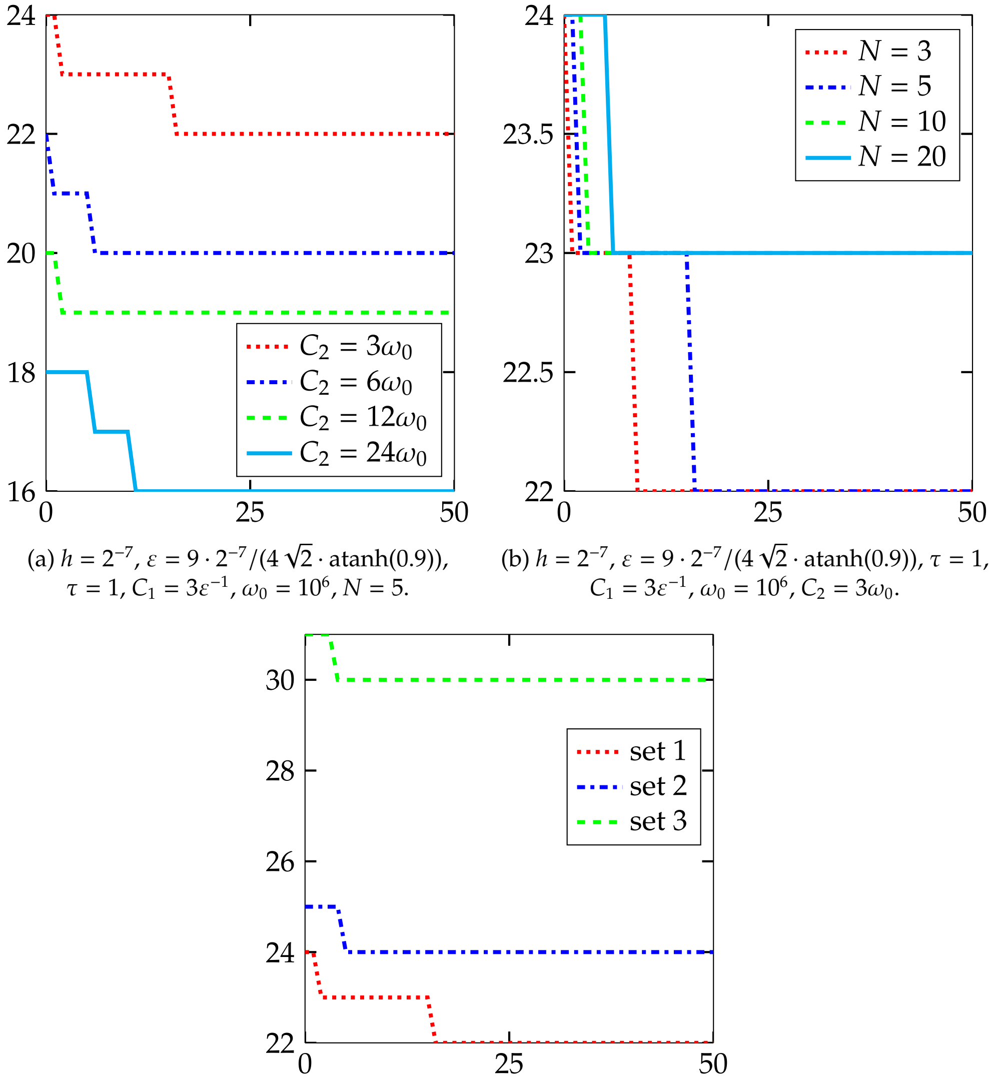
(a) $\varepsilon = 9 \cdot 2^{-7}/(4\sqrt{2} \cdot \text{atanh}(0.9))$, $\tau = 1$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, $N = 5$.

(b) $h = 2^{-7}$, $\tau = 1$, $C_1 = 375$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, $N = 5$.

(c) $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/(4\sqrt{2} \cdot \text{atanh}(0.9))$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, $N = 5$.

(d) $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/(4\sqrt{2} \cdot \text{atanh}(0.9))$, $\tau = 1$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, $N = 5$.

Figure 5.10: Results for the solution of the vector-valued smooth system represented in (5.59). The x-axis shows the time t and the y-axis the number of MINRES iterations.



(a) $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/(4\sqrt{2} \cdot \mathrm{atanh}(0.9))$, $\tau = 1$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^6$, $N = 5$.

(b) $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/(4\sqrt{2} \cdot \mathrm{atanh}(0.9))$, $\tau = 1$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$.

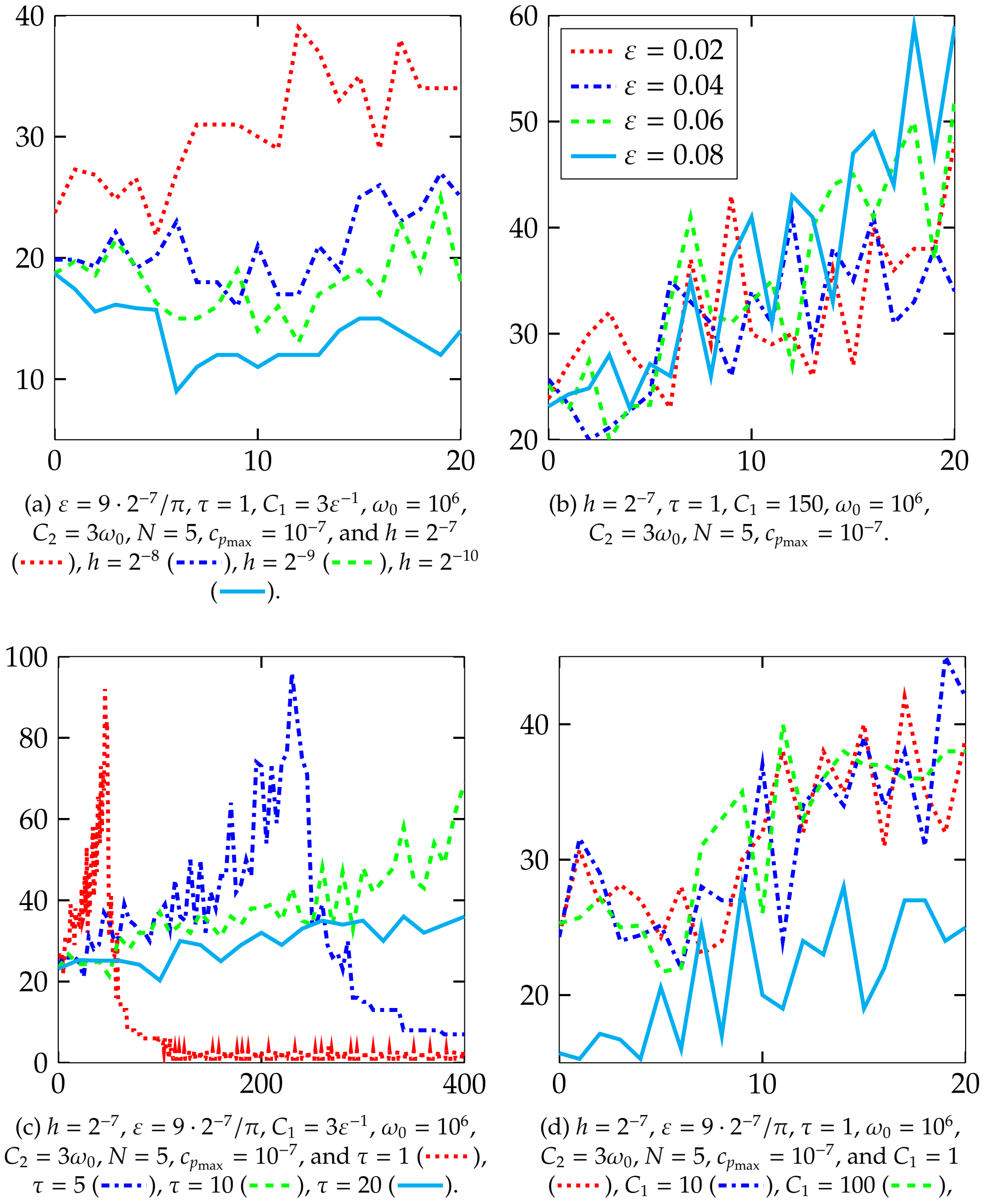
(c) Set j: $h_j = 2^{-6-j}$, $\varepsilon = 9\,h_j/(4\sqrt{2} \cdot \mathrm{atanh}(0.9))$, $\tau = 1$, $C_1^{(j)} = 3\varepsilon_j^{-1}$, $\omega_0^{(1)} = 10^6$, $\omega_0^{(2)} = 5 \cdot 10^6$, $\omega_0^{(3)} = 10^7$, $C_2^{(j)} = 3\omega_0^{(j)}$, $N = 5$ for $j = 1, 2, 3$.

Figure 5.11: Results for the solution of the vector-valued smooth system represented in (5.59). The x-axis shows the time t and the y-axis the number of MINRES iterations.

Simulation		MINRES			
Figure	Plot	Max	Avg	CPU (s)	CPU (s)
5.10(a)	(·····)	24	22	38	1981
	(-·-·)	24	23	164	8410
	(- - -)	24	23	768	39297
	(——)	24	22	2860	146158
5.10(b)	(·····)	24	23	41	2135
	(-·-·)	24	22	38	1988
	(- - -)	23	22	39	2030
	(——)	23	22	39	2043
5.10(c)	(·····)	24	22	39	79732
	(-·-·)	24	22	38	7785
	(- - -)	24	22	39	1627
	(——)	24	23	41	871
5.10(d)	(·····)	24	22	40	2063
	(-·-·)	28	28	50	2566
	(- - -)	37	36	62	3171
	(——)	50	49	83	4247
5.11(a)	(·····)	24	22	40	2073
	(-·-·)	22	20	35	1836
	(- - -)	20	19	34	1792
	(——)	18	16	30	1554
5.11(b)	(·····)	24	22	24	1237
	(-·-·)	24	22	38	1981
	(- - -)	24	23	82	4235
	(——)	24	23	163	8394
5.11(c)	(·····)	24	22	40	2057
	(-·-·)	25	24	169	8705
	(- - -)	31	30	1050	53649

Table 5.3: Results for the solution of the vector-valued smooth system represented in (5.59): The maximum and average number of MINRES iterations, the average CPU time (in seconds) for MINRES, and the CPU time (in seconds) for the whole simulation.

Figure 5.12(c), we vary the time step size τ while fixing $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, $N = 5$, $c_{p_{\max}} = 10^{-7}$, and $T = 400$. In Figure 5.12(d), we vary the convexity parameter C_1 while fixing $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 1$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, $N = 5$, $c_{p_{\max}} = 10^{-7}$, and $T = 20$. We observe a decrease of iteration numbers when C_1 is increased. In Figure 5.13(a), we vary the convexity parameter C_2 while fixing $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 1$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^6$, $N = 5$, $c_{p_{\max}} = 10^{-7}$, and $T = 20$. In Figure 5.13(b), we vary the number of phases N while fixing $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 1$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, $c_{p_{\max}} = 10^{-7}$, and $T = 20$. In Figure 5.13(c), we vary the penalty parameter $c_{p_{\max}}$ while fixing $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 1$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, $N = 5$, and $T = 20$. Finally, in Figure 5.13(d), we vary simultaneously all five parameters $h, \varepsilon, C_1, C_2, \omega$ while fixing $\tau = 1$, $N = 5$, $c_{p_{\max}} = 10^{-7}$, and $T = 20$. Table 5.4 illustrates the maximum and average number of BiCGstab iterations, the average CPU time (in seconds) for BiCGstab, and the CPU time (in seconds) for the whole simulation for each of the six subplots, respectively.



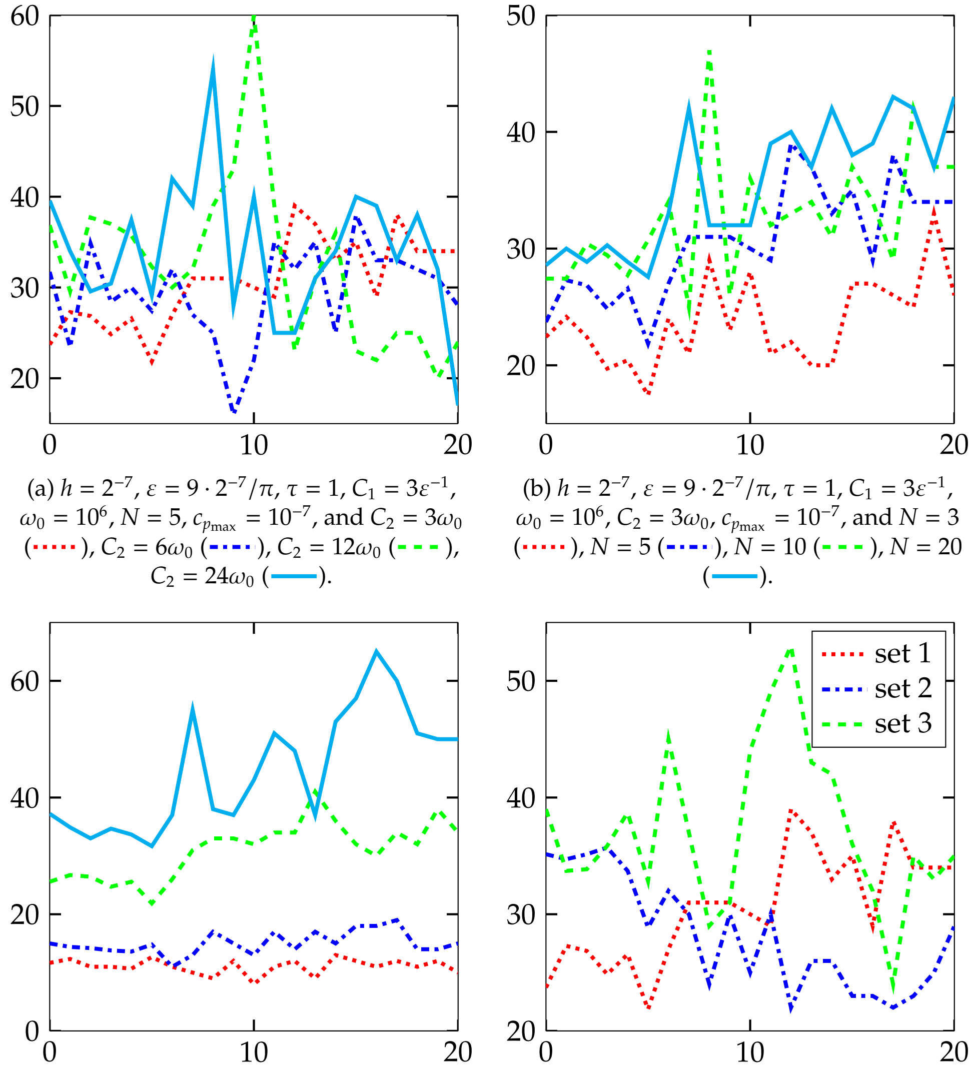
(a) $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 1$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, $N = 5$, $c_{p_{\max}} = 10^{-7}$, and $h = 2^{-7}$ (⋯⋯), $h = 2^{-8}$ (-·-·-), $h = 2^{-9}$ (- - -), $h = 2^{-10}$ (——).

(b) $h = 2^{-7}$, $\tau = 1$, $C_1 = 150$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, $N = 5$, $c_{p_{\max}} = 10^{-7}$.

(c) $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, $N = 5$, $c_{p_{\max}} = 10^{-7}$, and $\tau = 1$ (⋯⋯), $\tau = 5$ (-·-·-), $\tau = 10$ (- - -), $\tau = 20$ (——).

(d) $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 1$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, $N = 5$, $c_{p_{\max}} = 10^{-7}$, and $C_1 = 1$ (⋯⋯), $C_1 = 10$ (-·-·-), $C_1 = 100$ (- - -), $C_1 = 1000$ (——).

Figure 5.12: Results for the solution of the vector-valued nonsmooth system (5.79) with the preconditioner (5.80) and the Schur complement approximation (5.83). The x-axis shows the time t and the y-axis the average number of BiCGstab iterations per SSN step.



(a) $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 1$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^6$, $N = 5$, $c_{p_{\max}} = 10^{-7}$, and $C_2 = 3\omega_0$ (⋯⋯), $C_2 = 6\omega_0$ (-·-·), $C_2 = 12\omega_0$ (- - -), $C_2 = 24\omega_0$ (——).

(b) $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 1$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, $c_{p_{\max}} = 10^{-7}$, and $N = 3$ (⋯⋯), $N = 5$ (-·-·), $N = 10$ (- - -), $N = 20$ (——).

(c) $h = 2^{-7}$, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $\tau = 1$, $C_1 = 3\varepsilon^{-1}$, $\omega_0 = 10^6$, $C_2 = 3\omega_0$, $N = 5$, and $c_{p_{\max}} = 10^{-3}$ (⋯⋯), $c_{p_{\max}} = 10^{-5}$ (-·-·), $c_{p_{\max}} = 10^{-7}$ (- - -), $c_{p_{\max}} = 10^{-9}$ (——).

(d) Set j: $h_j = 2^{-6-j}$, $\varepsilon = 9\,h_j/\pi$, $\tau = 1$, $C_1^{(j)} = 3\varepsilon_j^{-1}$, $\omega_0^{(1)} = 10^6$, $\omega_0^{(2)} = 5 \cdot 10^6$, $\omega_0^{(3)} = 10^7$, $C_2^{(j)} = 3\omega_0^{(j)}$, $N = 5$, $c_{p_{\max}} = 10^{-7}$ for $j = 1, 2, 3$.

Figure 5.13: Results for the solution of the vector-valued nonsmooth system (5.79) with the preconditioner (5.80) and the Schur complement approximation (5.83). The x-axis shows the time t and the y-axis the average number of BiCGstab iterations per SSN step.

Simulation		SSN		BiCGstab			
Figure	Plot	Max	Avg	Max	Avg	CPU (s)	CPU (s)
5.12(a)	(·····)	7	3	70	23	301	55094
	(-·-·-)	6	4	63	18	579	134665
	(- - -)	7	4	61	16	2314	563110
	(——)	8	5	62	13	7049	1893020
5.12(b)	(·····)	6	3	71	25	316	53871
	(-·-·-)	6	4	70	22	234	46653
	(- - -)	6	4	87	23	190	39818
	(——)	6	3	96	24	199	41703
5.12(c)	(·····)	7	1	126	9	136	15043
	(-·-·-)	7	3	125	29	457	144854
	(- - -)	7	3	108	29	428	108555
	(——)	7	3	63	22	290	53231
5.12(d)	(·····)	7	4	74	24	321	64323
	(-·-·-)	7	4	72	24	318	62686
	(- - -)	7	4	69	23	313	59100
	(——)	6	3	51	15	200	30803
5.13(a)	(·····)	7	3	70	23	295	54165
	(-·-·-)	10	4	92	24	317	57526
	(- - -)	8	4	113	26	362	69448
	(——)	9	4	125	26	368	72256
5.13(b)	(·····)	5	3	52	18	105	15355
	(-·-·-)	7	3	70	23	301	55094
	(- - -)	8	4	84	26	770	167565
	(——)	9	4	118	27	1717	432896
5.13(c)	(·····)	4	2	22	10	78	6074
	(-·-·-)	7	3	42	12	102	17276
	(- - -)	7	3	73	23	306	55827
	(——)	7	3	82	32	382	67360
5.13(d)	(·····)	7	3	70	23	306	55967
	(-·-·-)	8	4	94	24	1108	232901
	(- - -)	10	4	149	29	4640	1207240

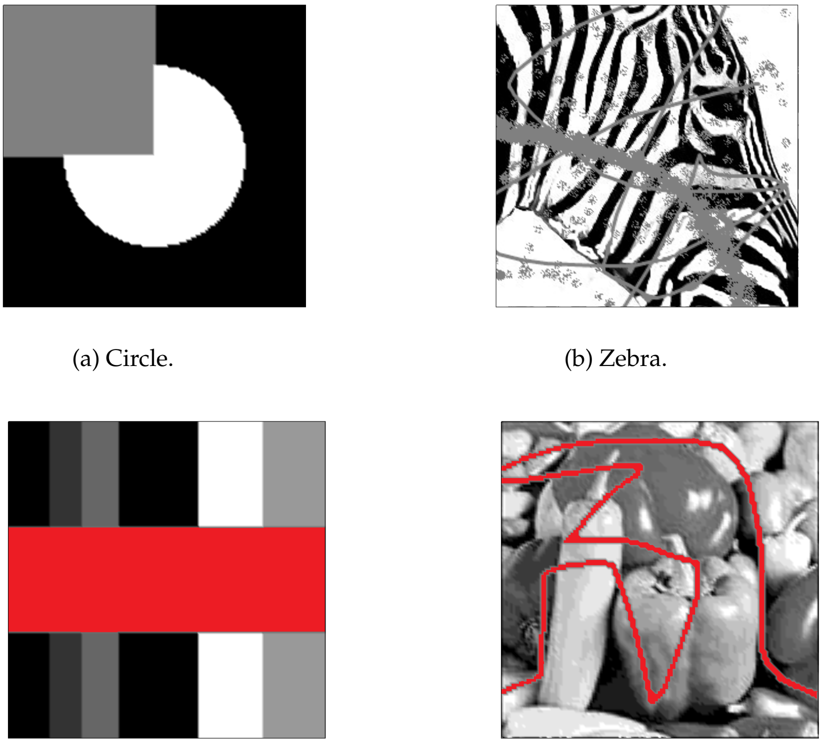
Table 5.4: Results for the solution of the vector-valued nonsmooth system (5.79) with the preconditioner (5.80) and the Schur complement approximation (5.83): The maximum and average number of SSN iterations, the maximum and average number of BiCGstab iterations, the average CPU time (in seconds) for BiCGstab, and the CPU time (in seconds) for the whole simulation.

5.7.3 Image inpainting

In the following, we consider image inpainting results using the smooth and nonsmooth modified Cahn–Hilliard model. We consider two examples for black-and-white inpainting using the scalar modified Cahn–Hilliard approach and two examples for gray value inpainting using the vector-valued modified Cahn–Hilliard approach. Figure 5.14 shows all four initial images. In Figures 5.14(a)–5.14(b), the inpainting domain is marked in gray. The circle image is of size 128 × 128 and the zebra image has the size 256 × 256. The zebra image is based on an extract of a plains zebra photo[4]. We have set all pixel values to either black or white and then added the damaged gray areas. In Figures 5.14(c)–5.14(d), the inpainting domain is marked in red. The stripes image is of size 64 × 64 and consists of $N = 5$ gray values. We

[4] ©2012 Thomas Rolle from the Zoo Magdeburg, Germany.

use a 128×128 version of the peppers image[5]. Using MATLAB® with the command `kmeans`, we have segmented the peppers image into $N = 15$ gray values.



(a) Circle.

(b) Zebra.

(c) Stripes.

(d) Peppers.

Figure 5.14: Initial images for image inpainting.

The parameters for the circle inpainting are $h = 2^{-7}$, $\varepsilon = 0.8$, $\varepsilon_s = h$, $\omega_0 = 10^5$, $c_{p_{\max}} = 10^{-7}$. Figures 5.15(a)–5.15(b) show the results using the scalar smooth modified Cahn–Hilliard equation. Figure 5.15(a) shows the solution after the first step of the ε-two-step approach. The solution of the second step is illustrated in Figure 5.15(b). Similarly, Figures 5.15(c)–5.15(d) show the results using the scalar nonsmooth modified Cahn–Hilliard model. The parameters for the zebra inpainting are $h = 2^{-8}$, $\varepsilon = 10.8$, $\varepsilon_s = 9h/\pi$, $\omega_0 = 10^7$, $c_{p_{\max}} = 10^{-7}$. As in the previous example, Figures 5.15(e)–5.15(h) show the results using the scalar smooth and nonsmooth modified Cahn–Hilliard model. The parameters for the stripes inpainting are $h = 2^{-7}$, $\varepsilon = 0.8$, $\varepsilon_s = 9h/\pi$, $\omega_0 = 10^5$, $c_{p_{\max}} = 10^{-7}$. As before, Figures 5.16(a)–5.16(d) show the results using the vector-valued smooth and nonsmooth modified Cahn–Hilliard model. In the nonsmooth model, we use the c-sequence of penalty parameters also during the first six time steps after the ε-switch. The parameters for the peppers inpainting are $h = 2^{-7}$, $\varepsilon = 1$, $\varepsilon_s = 9h/\pi$, $\omega_0 = 10^7$, $c_{p_{\max}} = 10^{-7}$. As before, Figures 5.16(e)–5.16(h) show the results using the vector-valued smooth and nonsmooth modified Cahn–Hilliard model. In the nonsmooth model, we use the c-sequence of penalty parameters also during the first six time steps after the ε-switch. Table 5.5 shows the maximum and average number of SSN iterations in the nonsmooth cases, the maximum and average number of MINRES/BiCG/BiCGstab

[5] http://mingyuanzhou.github.io/Results/BPFAImage/

iterations, the average CPU time (in seconds) for MINRES/BiCG/BiCGstab, and the CPU time (in seconds) for the whole simulation for each of the eight inpainting simulations, respectively. Table 5.6 displays the peak signal-to-noise ratio (PSNR) value as well as the minimum (min) and maximum (max) phase variable value for each of the eight inpainting simulations, respectively. The PSNR

$$\mathrm{PSNR} = 20\log_{10}\left(\frac{\max_{i,j}(f_r(i,j))}{\sqrt{\frac{1}{m_x m_y}\sum_{i=1}^{m_y}\sum_{j=1}^{m_x}\left(f_o(i,j)-f_r(i,j)\right)^2}}\right)$$

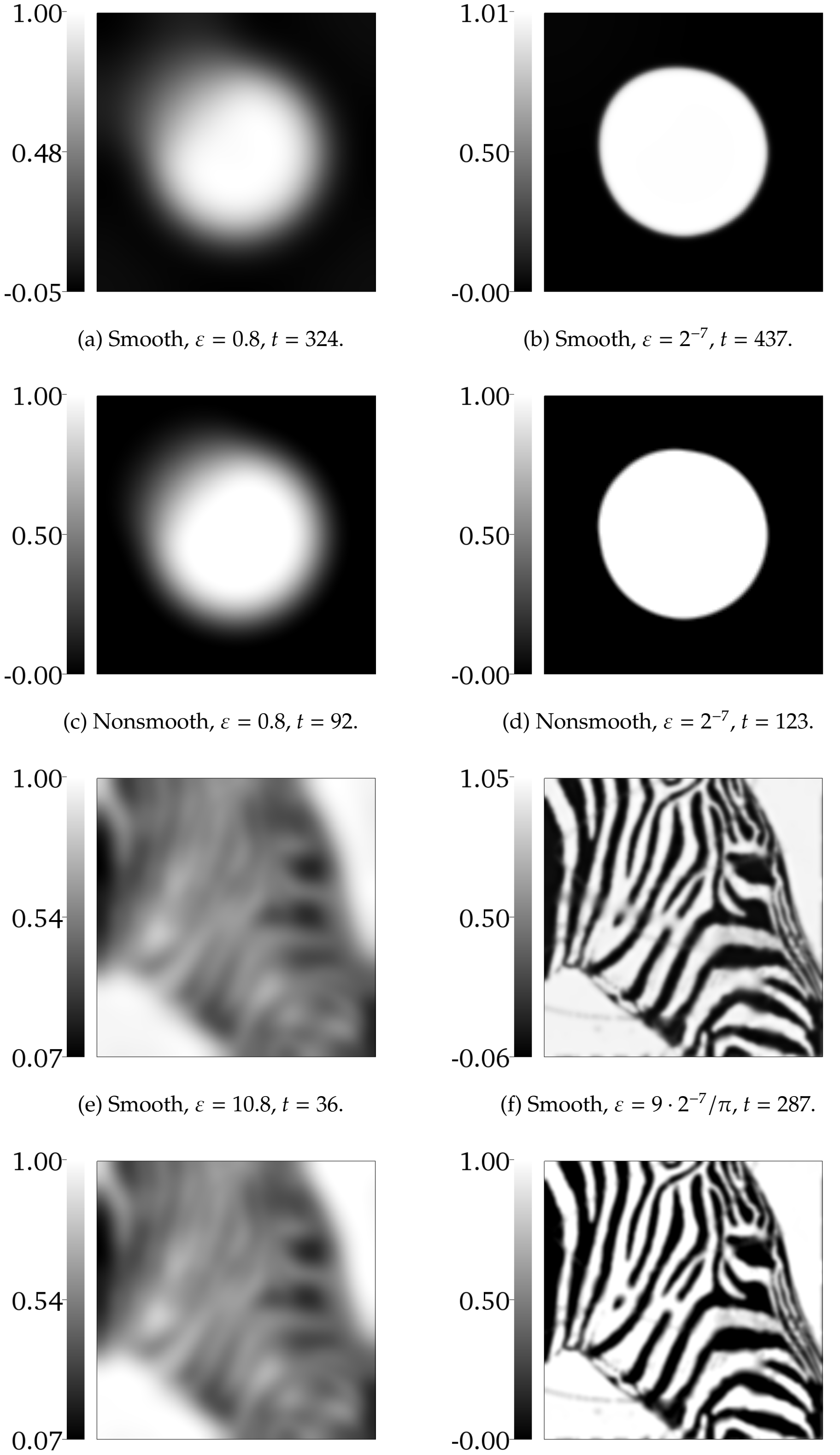
measures the quality of reconstruction. Here, f_o denotes the original image without damaged regions. The higher the PSNR is the better is the approximation of f_o. Since we do not have the pure black-and-white zebra extract, we do not measure the PSNR for this example. Comparing the smooth with the nonsmooth Cahn–Hilliard inpainting results, the nonsmooth ones have the sharper colors in the sense that the minimum and maximum phase variable values are closer to zero and one. Moreover, we observe larger PSNR values using the nonsmooth potential. Regarding the CPU time, the smooth Cahn–Hilliard inpainting model has an advantage over the nonsmooth one.

Simulation		SSN		MINRES/BiCG/BiCGstab			
Figure	Model	Max	Avg	Max	Avg	CPU (s)	CPU (s)
Circle	Smooth	−−	−−	36	31	11	4760
	Nonsmooth	4	3	37	22	4	4983
Zebra	Smooth	−−	−−	36	25	35	10457
	Nonsmooth	6	2	74	47	55	119932
Stripes	Smooth	−−	−−	33	24	42	25303
	Nonsmooth	6	3	99	23	312	243884
Peppers	Smooth	−−	−−	35	25	137	27840
	Nonsmooth	9	3	106	23	1086	668764

Table 5.5: Results for the solution of image inpainting examples: The maximum and average number of SSN iterations, the maximum and average number of MINRES/BiCG/BiCGstab iterations, the average CPU time (in seconds) for MINRES/BiCG/BiCGstab, and the CPU time (in seconds) for the whole simulation.

Simulation		Final image		
Figure	Model	PSNR	Min	Max
Circle	Smooth	21.54	$-2.17646 \cdot 10^{-3}$	1.00562
	Nonsmooth	21.75	$-6.39753 \cdot 10^{-6}$	1.00001
Zebra	Smooth	−−	$-5.56674 \cdot 10^{-2}$	1.05002
	Nonsmooth	−−	$-9.7906 \cdot 10^{-6}$	1.00001
Stripes	Smooth	20.25	$-2.92176 \cdot 10^{-2}$	1.04332
	Nonsmooth	22.72	$-4.72893 \cdot 10^{-6}$	1.00002
Peppers	Smooth	22.73	$-1.3401 \cdot 10^{-1}$	1.08097
	Nonsmooth	22.89	$-2.17629 \cdot 10^{-5}$	1.00027

Table 5.6: Results for the solution of image inpainting examples: The PSNR value as well as the minimum (min) and maximum (max) phase variable value.



(a) Smooth, $\varepsilon = 0.8$, $t = 324$.

(b) Smooth, $\varepsilon = 2^{-7}$, $t = 437$.

(c) Nonsmooth, $\varepsilon = 0.8$, $t = 92$.

(d) Nonsmooth, $\varepsilon = 2^{-7}$, $t = 123$.

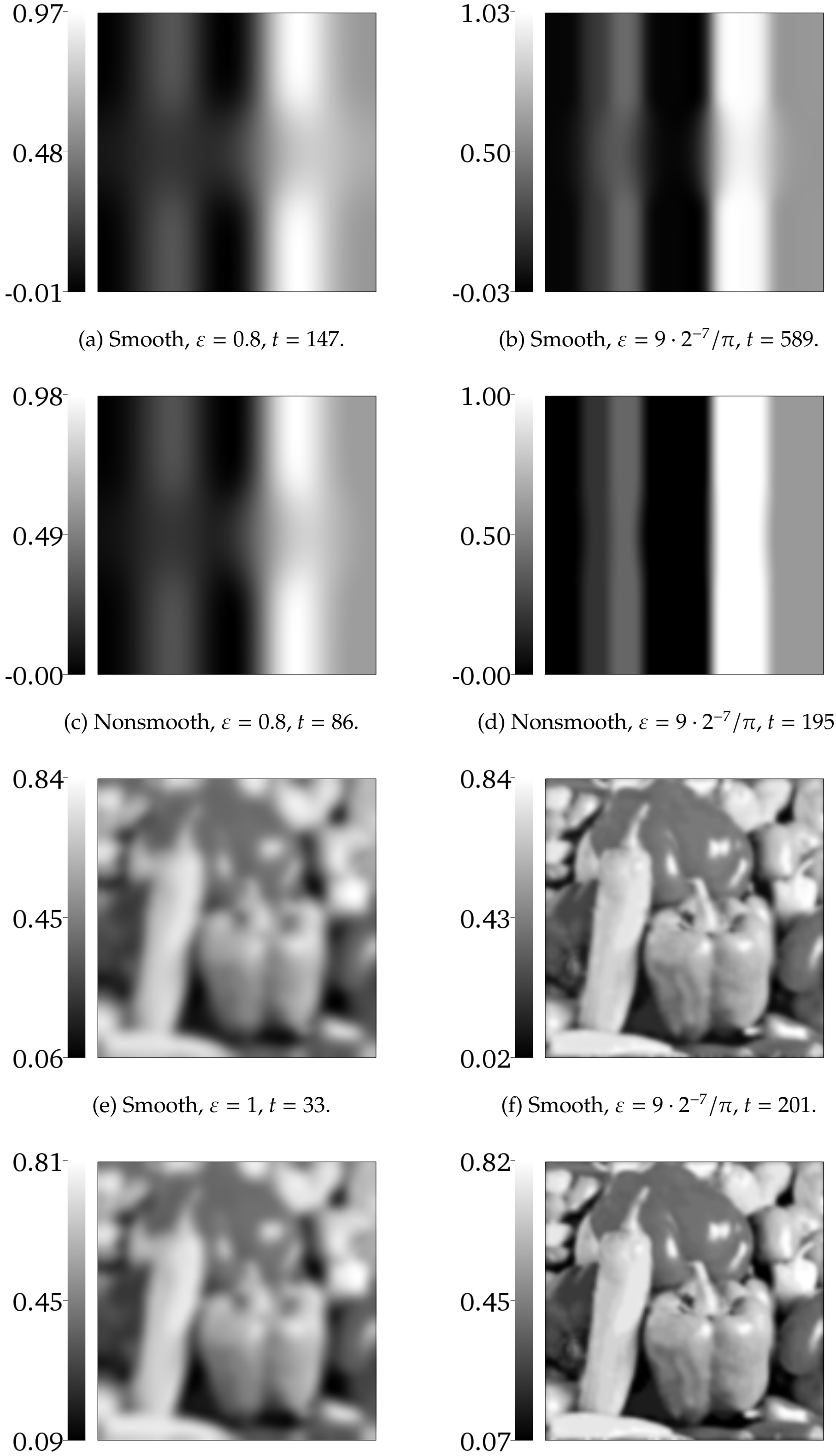
(e) Smooth, $\varepsilon = 10.8$, $t = 36$.

(f) Smooth, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $t = 287$.

(g) Nonsmooth, $\varepsilon = 10.8$, $t = 24$.

(h) Nonsmooth, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $t = 484$.

Figure 5.15: Inpainted black-and-white images using the smooth and nonsmooth scalar modified Cahn–Hilliard model.



(a) Smooth, $\varepsilon = 0.8$, $t = 147$.

(b) Smooth, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $t = 589$.

(c) Nonsmooth, $\varepsilon = 0.8$, $t = 86$.

(d) Nonsmooth, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $t = 195$.

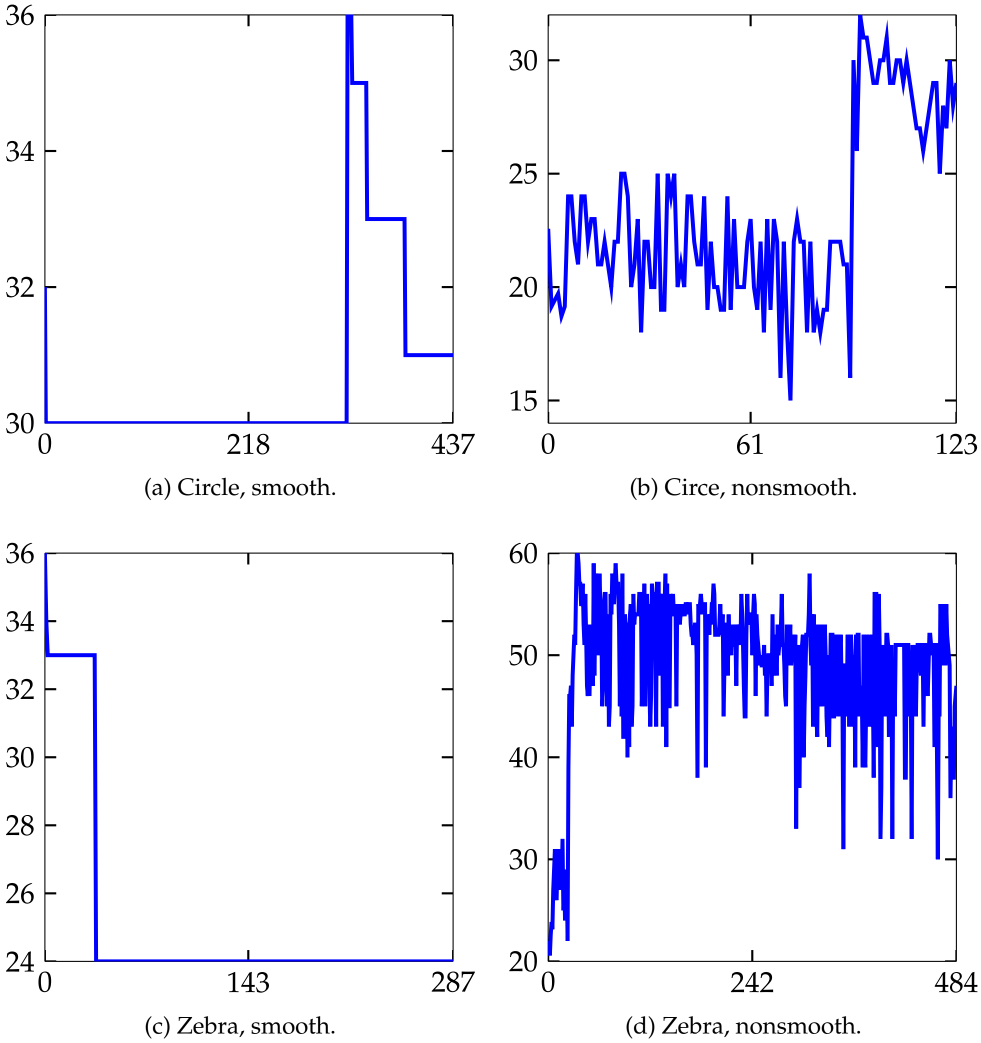
(e) Smooth, $\varepsilon = 1$, $t = 33$.

(f) Smooth, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $t = 201$.

(g) Nonsmooth, $\varepsilon = 1$, $t = 25$.

(h) Nonsmooth, $\varepsilon = 9 \cdot 2^{-7}/\pi$, $t = 152$.

Figure 5.16: Inpainted gray value images using the smooth and nonsmooth vector-valued modified Cahn–Hilliard model.



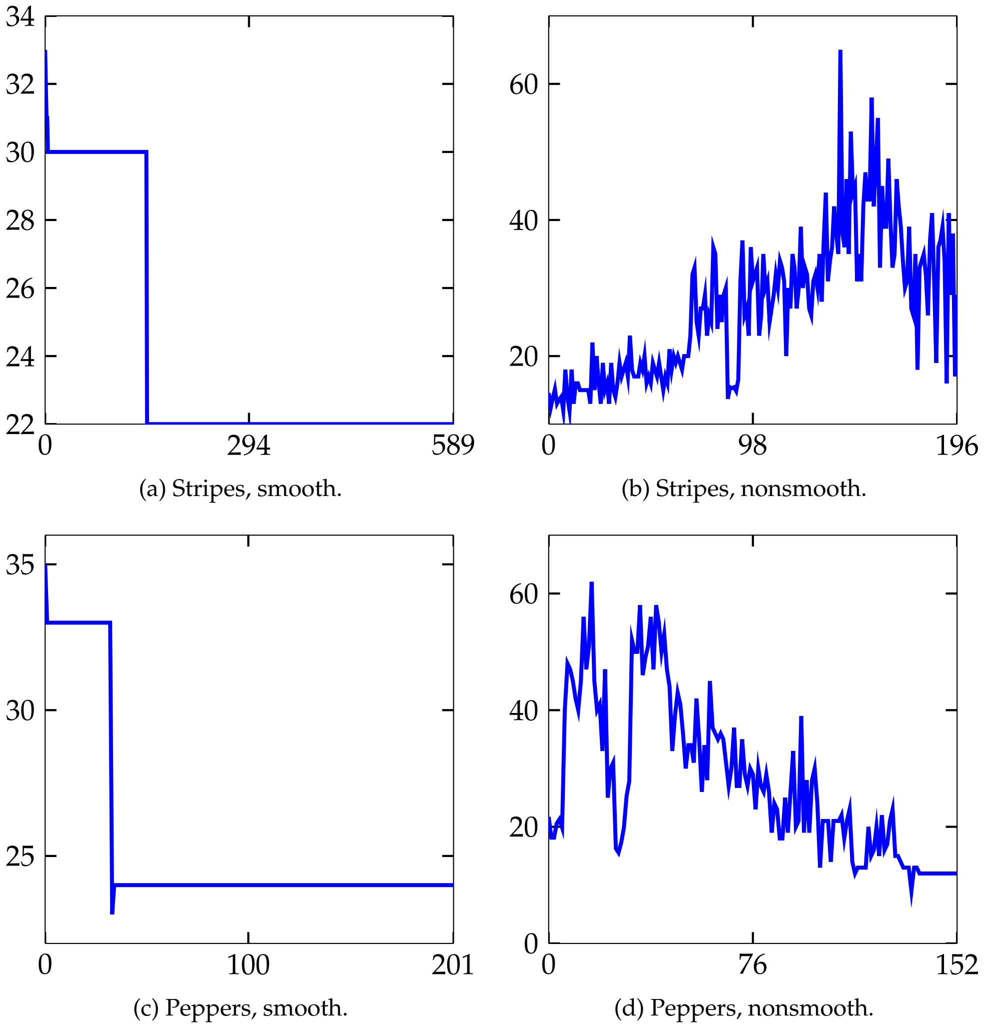
(a) Circle, smooth.

(b) Circe, nonsmooth.

(c) Zebra, smooth.

(d) Zebra, nonsmooth.

Figure 5.17: Results for the solution of four image inpainting examples: The x-axis shows the time t. For the smooth Cahn–Hilliard inpainting models, the y-axis displays the number of MINRES iterations. For the nonsmooth Cahn–Hilliard inpainting models, the y-axis displays the average number of BiCG iterations per SSN step.



(a) Stripes, smooth.

(b) Stripes, nonsmooth.

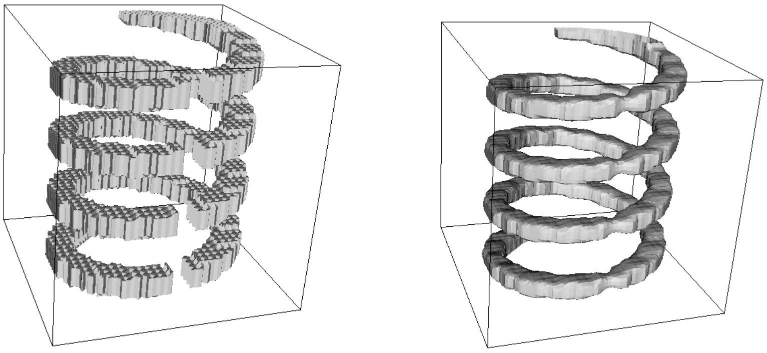
(c) Peppers, smooth.

(d) Peppers, nonsmooth.

Figure 5.18: Results for the solution of four image inpainting examples: The x-axis shows the time t. For the smooth Cahn–Hilliard inpainting models, the y-axis displays the number of MINRES iterations. For the nonsmooth Cahn–Hilliard inpainting models, the y-axis displays the average number of BiCGstab iterations per SSN step.

5.7.4 Three-dimensional example

Next, we consider the three-dimensional domain $\Omega = [-1,1]^3$, which contains a damaged spiral helix, see Figure 5.19(a). Additionally, we set every fourth pixel as damaged. Figure 5.19(b) shows the reconstruction using the scalar nonsmooth modified Cahn–Hilliard model with $h = 2^{-5}$, $\varepsilon = 10$, $\varepsilon_s = 9h/\pi$, $\omega = 10^6$, $c_{p_{\max}} = 10^{-5}$. In Figure 5.20, we illustrate the performance of our preconditioner. The x-axis shows the time t and the y-axis the average number of BiCG iterations per SSN step. The maximum and average number of SSN iterations are 6 and 2. The maximum and average number of BiCG iterations for the simulation are 72 and 24. The average CPU time for BiCG is 151s and the CPU time for the whole simulation is 16738s. The reconstruction has a PSNR value of 23.19.



(a) $t = 0$.

(b) $t = 152$.

Figure 5.19: Three-dimensional inpainting.

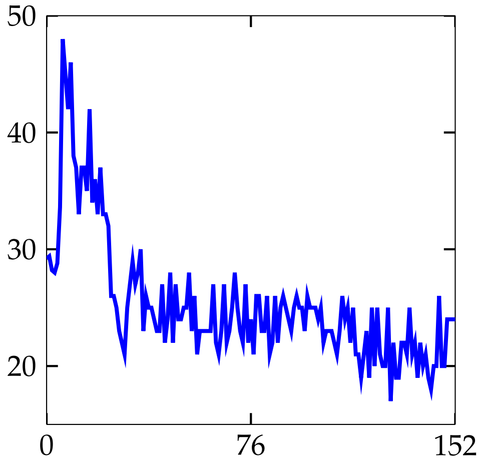


Figure 5.20: Results for the three-dimensional inpainting: The x-axis shows the time t and the y-axis the average number of BiCG iterations per SSN step.

5.7.5 Comparison with existing inpainting methods

In this section, we show the performance of various inpainting methods. The test example consists of six stripes spanning different widths and is of size 64×64, see Figure 5.14(c). It is composed of $N = 5$ gray values. The inpainting domain is given by the horizontal red stripe. Besides the proposed smooth and nonsmooth vector-valued Cahn–Hilliard inpainting model (*vector CH smooth/nonsmooth*), we examine the performance of our smooth vector-valued Cahn–Hilliard inpainting model solved via Fourier spectral methods (*vector CH Fourier*). Moreover, we have implemented the bitwise binary Cahn–Hilliard inpainting approach (*bitwise CH*) [139, pp. 435–436], which splits a gray value image bitwise into channels and applies the binary Cahn–Hilliard inpainting method to each channel. Furthermore, we test the three inpainting codes[6] provided by Schönlieb: These are inpainting using the heat equation (*heat*), total variation (TV) inpainting (*TV*), and TV-H^{-1} inpainting (*TV-H*$^{-1}$). Additionally, we consider the MATLAB® function `inpaintn`[7] (*inpaintn*) [71, 148], which replaces the missing data by extra/interpolating the undamaged elements. Finally, we test Zhou et al's [153] nonparametric Bayesian method[8], which they term the beta process factor analysis (*BPFA*).

All computations are executed on a 64-bit server with CPU type Intel® Xeon® X5650 @2.67 GHz, with 2 CPUs, 12 Cores (6 Cores per CPU), and 48 GB main memory available. As in the previous chapters, our finite element implementation for *vector CH* is done in C++ using the open source finite element library deal.II version 7.1.0 [8]. All remaining inpainting methods are executed in MATLAB® R2012b. Note that we use the MATLAB® Image Processing Toolbox™ for combining the phases to the final image and for visualizations. We provide the MATLAB® code for *vector CH Fourier* to reproduce the numerical example as supplementary material.[9]

In all Cahn–Hilliard inpainting approaches, we set $h = 2^{-6}$, $\omega_0 = 10^5$, $c_{p_{\max}} = 10^{-7}$ and apply the ε-two-step procedure with a switch from $\varepsilon = 1$ to $\varepsilon_s = h$. In both steps, the stopping criterion is

$$\frac{\|\boldsymbol{u}^{(n)} - \boldsymbol{u}^{(n-1)}\|}{\|\boldsymbol{u}^{(n-1)}\|} \leq \gamma \tag{5.87}$$

with $\gamma = 5 \cdot 10^{-4}$. In the nonsmooth model, we use the c-sequence of penalty parameters also during the first six time steps after the ε-switch. We run the heat equation based approach with $h = 1$, $\tau = 0.1$, $\omega_0 = 10$ and both TV models with $h = 1$, $\tau = 1$, $\varepsilon = 2^{-6}$, $\omega_0 = 10$. We apply to these three methods the stopping criterion (5.87) with $\gamma = 5 \cdot 10^{-4}$. We stop the MATLAB® function `inpaintn` after 10^5 iterations. We run BPFA with the default values. In contrast to our proposed vector-valued models, the methods we compare with represent scalar systems. By experience from those models, we set the pixel values in the damaged regions to zero instead of N^{-1}. Note that the gray values range from 0 to 255 for bitwise CH. For the

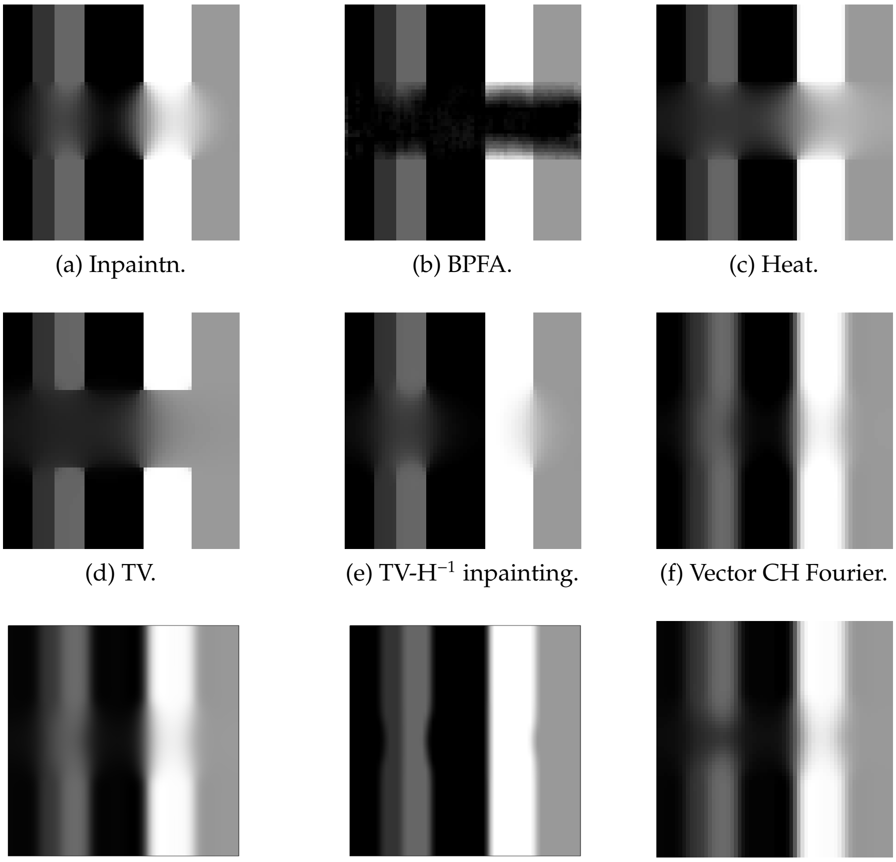
[6] http://www.mathworks.com/matlabcentral/fileexchange/34356-higher-order-total-variation-inpainting

[7] http://www.mathworks.com/matlabcentral/fileexchange/27994-inpaint-over-missing-data-in-1-d--2-d--3-d--n-d-arrays

[8] http://mingyuanzhou.github.io/Results/BPFAImage/

[9] http://www.mpi-magdeburg.mpg.de/2968228/Supplementary_BoschStoll.zip

remaining methods, the range is from 0 to 1.



(a) Inpaintn. (b) BPFA. (c) Heat.

(d) TV. (e) TV-H^{-1} inpainting. (f) Vector CH Fourier.

(g) Vector CH smooth. (h) Vector CH nonsmooth. (i) Bitwise CH.

Figure 5.21: Inpainted gray value image using different inpainting models.

Inpainting method	Iter	CPU (s)	$\overline{\text{CPU}}$ (s)	PSNR	Min	Max
Inpaintn	100000	234.17	0.0023	22.39	$-1.05 \cdot 10^{-3}$	1.0000
BPFA	1072	143.48	0.1338	12.16	$-1.14 \cdot 10^{-1}$	1.0010
Heat	4042	0.86	0.0002	18.89	$4.92 \cdot 10^{-10}$	1.0000
TV	17354	20.77	0.0012	16.50	$5.54 \cdot 10^{-5}$	0.9999
TV-H^{-1}	43099	52.84	0.0012	26.94	$-1.78 \cdot 10^{-3}$	1.0168
Vector CH Fourier	258	5.45	0.0211	21.06	$-2.34 \cdot 10^{-2}$	1.0242
Vector CH smooth	258	3271.75	12.7305	21.35	$-2.16 \cdot 10^{-2}$	1.0223
Vector CH nonsmooth	204	58030.40	284.4627	24.92	$-1.27 \cdot 10^{-5}$	1.0000
Bitwise CH	265	8.38	0.0316	20.64	$-5.98 \cdot 10^{+0}$	261.18

Table 5.7: Performance of different inpainting models: The total number of iterations (iter), CPU time (in seconds) for the whole simulation, average CPU time (in seconds) per time step ($\overline{\text{CPU}}$), PSNR value, as well as the minimum (min) and maximum (max) pixel value of the final image.

Figure 5.21 illustrates the inpainted images using the different approaches mentioned above. The second-order TV inpainting approach is not able to connect the stripes.

Moreover, BPFA fails using the standard parameter set. TV-H^{-1} inpainting results in a partly complete connection. The two rightmost stripes have successfully joined. We observe a similar behavior with the smooth Cahn–Hilliard inpainting approach. In both cases, the stopping criterion (5.87) might not be the optimal choice. In general, the discussion about the stopping criterion should be a task for future work. In [32], we compare different stopping criteria for our proposed model. Both, the smooth and nonsmooth Cahn–Hilliard inpainting model, provide a connection of the stripes over the inpainting domain. Table 5.7 lists the total number of iterations (iter), CPU time (in seconds) for the whole simulation, average CPU time (in seconds) per time step ($\overline{\text{CPU}}$), PSNR value, as well as the minimum (min) and maximum (max) pixel value of the final image. Comparing the smooth with the nonsmooth Cahn–Hilliard inpainting result, the nonsmooth one has the sharper colors and the higher PSNR value at the cost of a higher CPU time. Comparing the image quality between *vector CH* and the remaining approaches, we see that *vector CH* defeats most of the methods. However, comparing the computational times between *vector CH* and the remaining approaches, we are at a disadvantage.

5.8 Conclusions and future research perspectives

In this chapter, we have investigated the numerical solution of a two-component and multi-component modified Cahn–Hilliard model. In particular, we have developed an inpainting model based on the multi-component Cahn–Hilliard equation. This approach generalizes Bertozzi et al's [19] binary Cahn–Hilliard inpainting model to gray value images. We have considered smooth and nonsmooth potentials with a focus on the latter. An important difference to the previous two chapters is that the modified Cahn–Hillard equation as a whole is not given by a gradient flow. Especially, the model arises as a superposition of two gradient flows. We have applied the convexity splitting technique, which yields under the right conditions an unconditional gradient stable time-discrete scheme. Regarding the smooth setting, we have extended the proof of consistency, unconditional stability, and convergence of the time-discrete scheme from the two-component to the multi-component case. Concerning the nonsmooth framework, following the previous two chapters, we have applied an SSN method combined with a Moreau–Yosida regularization technique. For the discretization in space, we have used classical FEM for both systems, the smooth and regularized nonsmooth one. At the heart of our method lies the solution of large and sparse systems of linear equations of saddle point form. We have introduced and studied block diagonal and block-triangular preconditioners using an efficient and cheap Schur complement approximation. For this approximation, we have used multilevel techniques, algebraic multigrid in our case. For the smooth systems, we have derived the conditions for optimal preconditioners. For the nonsmooth systems, extensive numerical experiments have shown that our developed preconditioners are reliable. The use of our preconditioners allows us to perform three-dimensional experiments in an efficient way. The numerical results have shown that the use of a nonsmooth Cahn–Hilliard model leads to visually improved restored images, when compared to existing inpainting tools.

As pointed out in this chapter, there are several aspects for further research. First of all, a rigorous analysis for the nonsmooth case is missing. Second, the study of preconditioners for the linear systems arising from other time discretization schemes might be of interest. A third extension might be the development of an adaptive spatial mesh refinement approach, which reduces the number of degrees of freedoms. Fourth, in [32], we have further generalized our smooth gray value Cahn–Hilliard inpainting model to a fractional-in-space version, which is efficiently solved with Fourier spectral methods on simple domains. The numerical results show the superiority of the fractional Cahn–Hilliard inpainting approach over its nonfractional version in terms of image quality. An interesting topic of future research would be the numerical solution of fractional Cahn–Hilliard models with FEMs. This research area becomes even more challenging when we combine it with nonsmooth potentials. Finally, since image inpainting is a special form of art, one could play around with the potential functions. As an example, one could consider potential functions with different altitude of the wells or potential functions which include different weights. For the latter idea, I would like to thank Helge Dietert from the University of Cambridge for his interest and a fruitful discussion with him. Besides varying the potential function, one could vary the interfacial parameter as well. For example, one could consider the interfacial parameter as a function in space.

Chapter 6

Conclusions and Outlook

In this thesis, we have advanced numerical solution techniques for various types of Cahn–Hilliard problems equipped with smooth and nonsmooth potentials while the attention is on the latter. The considered problems are first the two-component, and in particular the multi-component, Cahn–Hilliard model for phase separation and coarsening processes. Second, we have successfully applied our preconditioner to a coupled two-component Cahn–Hilliard/Navier–Stokes system equipped with a nonsmooth potential. Third, we have enhanced the study of the modified Cahn–Hilliard model as a tool for image inpainting. This thesis makes contributions to both, the theoretical and numerical analysis of those problems. The core theme is the development of efficient preconditioners for the iterative solution of the large and sparse linear systems that arise from classical finite element methods. We have designed, implemented, and analyzed preconditioners that are tailored to the different Cahn–Hilliard problems. In particular, our preconditioners are proven to be robust with respect to parameter changes when smooth potentials are used. Even for the nonsmooth systems, extensive numerical experiments have shown that our preconditioners are promising: We have observed a nearly parameter independent behavior of our developed preconditioners and in some cases only a benign increase of iteration numbers. Note that the construction of efficient preconditioners in the smooth case is already well established by Boyanova et al. [35, 37, 36, 3, 38]. However, we have extended this theory in several ways: First of all, our theoretical proofs differ halfway through. Second, we have generalized the theory to the vector-valued case with symmetric positive semidefinite mobility matrices as well as to the linear systems that arise from the Cahn–Hilliard inpainting model. Third, we have used this technique to analyze the systems of linear equations in the nonsmooth setting. The numerical solution of Cahn–Hilliard problems that include a nonsmooth potential form the challenging part of this thesis. Whereas the use of smooth potentials leads to a system of parabolic partial differential equations, the nonsmooth ones result in a system of variational inequalities. To deal with such systems, we have proposed an SSN method combined with a Moreau–Yosida regularization technique, which is investigated in [91] for the two-component Cahn–Hilliard model discretized in time with a semi-implicit scheme. We have extended the analysis to the two-component

Cahn–Hilliard model discretized in time with an implicit scheme in [30]. For the sake of completeness, parts of the work in [30] appear in Chapter 3. Our new contributions in this direction are the following: We have extended parts of the analysis to the vector-valued Cahn–Hilliard model. In particular, we have interpreted the implicit time-discrete problem as the first-order optimality system of an optimization problem for which we have derived existence and uniqueness conditions. We have analyzed the corresponding optimization problem of the Moreau–Yosida regularized version and have derived existence and uniqueness conditions of its solution. Moreover, we have proven a convergence result that connects the solutions of the regularized optimization problems to the original optimization problems. Finally, we have shown the applicability of the SSN method combined with a Moreau–Yosida regularization technique to solve the scalar and vector-valued modified Cahn–Hilliard equation. The core of this approach is again the solution of large and sparse systems of linear equations of saddle point form. In contrast to the linear systems that arise from the use of smooth potentials, an additional and essential parameter, the regularization parameter, enters the formulation. This complicates the properties of the coefficient matrix and makes the solution of the linear systems a challenge. To the best of our knowledge, practical and robust preconditioners for the iterative solution of these problems are previously unknown. In this thesis, we have developed efficient preconditioners for the solution of all of the above mentioned Cahn–Hilliard problems. Although there are several points that have to be discussed further, we can conclude that our techniques are promising and reliable.

We have already pointed out problem specific discussions for future research in the corresponding chapters. Here, we bring up some general extensions for further research. In the previous chapters, we have mentioned existing approaches for the solution of nonsmooth Cahn–Hilliard problems. In particular, Gräser, Kornhuber, and Sack [85] proposed globally convergent nonsmooth Schur–Newton methods for the solution of discrete multi-component Cahn–Hilliard systems equipped with logarithmic as well as obstacle potentials. An important point for the future is a comparison with our approach. Recently, Kumar [111, 112] explored preconditioners for solving the second substep of the nonsmooth Newton method proposed by Gräser and Kornhuber [84]. In particular, Kumar considered our Schur complement preconditioner from Chapter 3.7 and adapted it to the linear systems he deals with. Experiments showed that the number of preconditioned GMRES iterations remains independent of the mesh size, however it depends on the interfacial parameter ε. But for a fixed ε, the number of preconditioned GMRES iterations decreases significantly when the mesh is refined. This makes the preconditioner effective and useful on finer meshes.

An interesting future direction would be the application of our solvers to real-world problems. This includes the incorporation of realistic model parameters. Examples are the simulation of mineral growth [110] or tumor growth [103]. Usually, the resulting discretized problems are of enormous dimensions and hence too large to be tackled by standard approaches. As a result, high performance computing becomes inevitable. This research area forms another important extension.

Theses

1. This thesis deals with the numerical solution of various types of Cahn–Hilliard problems equipped with smooth and nonsmooth potentials with an emphasis on the latter. The core theme is the development of efficient and practical preconditioners for the iterative solution of the large and sparse linear systems that arise from classical finite element methods.

2. We have designed, implemented, and analyzed preconditioners that are tailored to the different Cahn–Hilliard problems.

3. We have extended Boyanova et al's theory for smooth preconditioners in four ways: First, we have formulated an altered proof that is based on the symmetric Schur decomposition. Second, we have derived the theory for the vector-valued case with symmetric positive semidefinite mobility matrices. Next, we have extended the proof to the linear systems that arise from the Cahn–Hilliard inpainting model. Finally, using this technique, we have analyzed the systems of linear equations in the nonsmooth systems.

4. In the smooth settings, we have derived theoretical conditions for the optimality of our preconditioners.

5. In the nonsmooth settings, we have proven that the use of the preconditioners from the smooth settings give worse approximations for small regularization parameters.

6. The numerical experiments confirm the robustness of our preconditioners applied to the smooth systems with respect to all crucial parameters.

7. Extensive numerical tests show that our preconditioners applied to the non-smooth systems are either nearly parameter independent or show a benign increase of iteration numbers.

8. We have applied a mesh adaptation strategy to the numerical solution of the nonsmooth multi-component Cahn–Hilliard equation. This reduces the number of degrees of freedom and allows us to perform three-dimensional experiments in an efficient way.

9. First numerical experiments show the effectiveness of our developed preconditioner applied to a two-component nonsmooth coupled Cahn–Hilliard/Navier–Stokes system. In particular, we observe a promising behavior when the Reynolds number is increased as well as when the mesh, interfacial, time step, and mobility parameter are refined all four together.

10. Eigenvalue plots for our Schur complement approximations demonstrate the desired eigenvalue clustering.

11. Our extension of Hintermüller et al's theory about the Moreau–Yosida regularization technique combined with the semismooth Newton method is twofold: We have extended parts of the analysis to the vector-valued Cahn–Hilliard model. In particular, we have interpreted the implicit time-discrete problem as the first-order optimality system of an optimization problem for which we have derived existence and uniqueness conditions. We have analyzed the corresponding optimization problem of the Moreau–Yosida regularized version and derived existence and uniqueness conditions of its solution. Moreover, we have proven a convergence result that connects the solution of the regularized optimization problem to the original optimization problem. Second, we have shown the applicability of the SSN method combined with a Moreau–Yosida regularization technique to solve the scalar and vector-valued modified Cahn–Hilliard equation.

12. We have developed a gray value inpainting model based on the vector-valued Cahn–Hilliard equation.

13. We have extended the smooth Cahn–Hilliard inpainting approach to the nonsmooth case.

14. The numerical results show that the use of the nonsmooth Cahn–Hilliard inpainting model visually improves the reconstructed images when compared to existing inpainting tools.

Bibliography

[1] H. Abels, H. Garcke, and G. Grün, *Thermodynamically consistent, frame indifferent diffuse interface models for incompressible two-phase flows with different densities*, Math. Models Methods Appl. Sci., 22 (2012), p. 1150013.

[2] O. Axelsson, *A survey of preconditioned iterative methods for linear systems of algebraic equations*, BIT, 25 (1985), pp. 165–187.

[3] O. Axelsson, P. Boyanova, M. Kronbichler, M. Neytcheva, and X. Wu, *Numerical and computational efficiency of solvers for two-phase problems*, Comput. Math. Appl., 65 (2013), pp. 301–314.

[4] O. Axelsson and M. Neytcheva, *Operator splittings for solving nonlinear, coupled multiphysics problems with an application to the numerical solution of an interface problem*, Tech. Report 2011-009, Department of Information Technology, Uppsala University, 2011.

[5] W. Baatz, M. Fornasier, P. A. Markowich, and C.-B. Schönlieb, *Inpainting of ancient austrian frescoes*, in Bridges Leeuwarden: Mathematics, Music, Art, Architecture, Culture, R. Sarhangi and C. H. Séquin, eds., London, 2008, Tarquin Publications, pp. 163–170.

[6] V. E. Badalassi, H. D. Ceniceros, and S. Banerjee, *Computation of multiphase systems with phase field models*, J. Comput. Phys., 190 (2003), pp. 371–397.

[7] Ľ. Baňas and R. Nürnberg, *A multigrid method for the Cahn–Hilliard equation with obstacle potential*, Appl. Math. Comput., 213 (2009), pp. 290–303.

[8] W. Bangerth, R. Hartmann, and G. Kanschat, *deal.II – A general purpose object oriented finite element library*, ACM Trans. Math. Software, 33 (2007), pp. 24/1–24/27.

[9] R. E. Bank, B. D. Welfert, and H. Yserentant, *A class of iterative methods for solving saddle point problems*, Numer. Math., 56 (1990), pp. 645–666.

[10] E. Bänsch, P. Morin, and R. H. Nochetto, *Preconditioning a class of fourth order problems by operator splitting*, Numer. Math., 118 (2011), pp. 197–228.

[11] J. W. Barrett and J. F. Blowey, *An error bound for the finite element approximation of a model for phase separation of a multi-component alloy*, IMA J. Numer. Anal., 16 (1996), pp. 257–287.

[12] ——, *Finite element approximation of a model for phase separation of a multi-component alloy with non-smooth free energy*, Numer. Math., 77 (1997), pp. 1–34.

[13] 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 (1999), pp. 286–318.

[14] J. W. Barrett, R. Nürnberg, and V. Styles, *Finite element approximation of a phase field model for void electromigration*, SIAM J. Numer. Anal., 42 (2004), pp. 738–772.

[15] M. Benzi, *Preconditioning techniques for large linear systems: A survey*, J. Comput. Phys., 182 (2002), pp. 418–477.

[16] M. Benzi, G. H. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Numer., 14 (2005), pp. 1–137.

[17] M. Bertalmío, G. Sapiro, V. Caselles, and C. Ballester, *Image inpainting*, in Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2000, New York, 2000, ACM Press/Addison–Wesley, pp. 417–424.

[18] A. Bertozzi, S. Esedoḡlu, and A. Gillette, *Analysis of a two-scale Cahn–Hilliard model for binary image inpainting*, Multiscale Model. Simul., 6 (2007), pp. 913–936.

[19] A. L. Bertozzi, S. Esedoḡlu, and A. Gillette, *Inpainting of binary images using the Cahn–Hilliard equation*, IEEE Trans. Image Process., 16 (2007), pp. 285–291.

[20] L. Blank, M. Butz, and H. Garcke, *Solving the Cahn–Hilliard variational inequality with a semi-smooth Newton method*, ESAIM Control Optim. Calc. Var., 17 (2011), pp. 931–954.

[21] L. Blank, M. H. Farshbaf-Shaker, H. Garcke, C. Rupprecht, and V. Styles, *Multi-material phase field approach to structural topology optimization*, in Trends in PDE Constrained Optimization, G. Leugering, P. Benner, S. Engell, A. Griewank, H. Harbrecht, M. Hinze, R. Rannacher, and S. Ulbrich, eds., vol. 165 of Internat. Ser. Numer. Math., Springer International Publishing, Cham, 2014, pp. 231–246.

[22] L. Blank, H. Garcke, L. Sarbu, and V. Styles, *Nonlocal Allen–Cahn systems: Analysis and a primal-dual active set method*, IMA J. Numer. Anal., 33 (2013), pp. 1126–1155.

[23] ——, *Primal-dual active set methods for Allen–Cahn variational inequalities with nonlocal constraints*, Numer. Methods Partial Differential Equations, 29 (2013), pp. 999–1030.

[24] L. Blank and C. Rupprecht, *An extension of the projected gradient method to a Banach space setting with application in structural topology optimization*, Tech. Report 4/2015, Department of Mathematics, University of Regensburg, 2015.

[25] L. Blank, L. Sarbu, and M. Stoll, *Preconditioning for Allen–Cahn variational inequalities with non-local constraints*, J. Comput. Phys., 231 (2012), pp. 5406–5420.

[26] J. F. Blowey, M. I. M. Copetti, and C. M. Elliott, *Numerical analysis of a model for phase separation of a multicomponent alloy*, IMA J. Numer. Anal., 16 (1996), pp. 111–139.

[27] J. F. Blowey and C. M. Elliott, *The Cahn–Hilliard gradient theory for phase separation with non-smooth free energy. Part I: Mathematical analysis*, European J. Appl. Math., 2 (1991), pp. 233–280.

[28] ——, *The Cahn–Hilliard gradient theory for phase separation with non-smooth free energy. Part II: Numerical analysis*, European J. Appl. Math., 3 (1992), pp. 147–179.

[29] ——, *Curvature dependent phase boundary motion and parabolic double obstacle problems*, in Degenerate Diffusions, W.-M. Ni, L. A. Peletier, and J. L. Vazquez, eds., vol. 47 of IMA Vol. Math. Appl., Springer, New York, 1993, pp. 19–60.

[30] J. Bosch, *Schnelle Löser für Cahn–Hilliard Variationsungleichungen*, Master's thesis, Faculty of Mathematics, Otto-von-Guericke-Universität Magdeburg, 2012. In German.

[31] J. Bosch, D. Kay, M. Stoll, and A. J. Wathen, *Fast solvers for Cahn–Hilliard inpainting*, SIAM J. Imaging Sci., 7 (2014), pp. 67–97.

[32] J. Bosch and M. Stoll, *A fractional inpainting model based on the vector-valued Cahn–Hilliard equation*, SIAM J. Imaging Sci., 8 (2015), pp. 2352–2382.

[33] ——, *Preconditioning for vector-valued Cahn–Hilliard equations*, SIAM J. Sci. Comput., 37 (2015), pp. S216–S243.

[34] J. Bosch, M. Stoll, and P. Benner, *Fast solution of Cahn–Hilliard variational inequalities using implicit time discretization and finite elements*, J. Comput. Phys., 262 (2014), pp. 38–57.

[35] P. Boyanova, M. Do-Quang, and M. Neytcheva, *Solution methods for the Cahn–Hilliard equation discretized by conforming and non-conforming finite elements*, Tech. Report 2011-004, Department of Information Technology, Uppsala University, 2011.

[36] ——, *Block-preconditioners for conforming and non-conforming FEM discretizations of the Cahn–Hilliard equation*, in Large-Scale Scientific Computing, I. Lirkov, S. Margenov, and J. Waśniewski, eds., vol. 7116 of Lecture Notes in Comput. Sci., Springer, Berlin Heidelberg, 2012, pp. 549–557.

[37] ——, *Efficient preconditioners for large scale binary Cahn–Hilliard models*, Comput. Methods Appl. Math., 12 (2012), pp. 1–22.

[38] P. Boyanova and N. Neytcheva, *Efficient numerical solution of discrete multi-component Cahn–Hilliard systems*, Comput. Math. Appl., 67 (2014), pp. 106–121.

[39] J. H. Bramble and J. E. Pasciak, *A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems*, Math. Comput., 50 (1988), pp. 1–17.

[40] A. Brandt, *Multi-level adaptive solutions to boundary-value problems*, Math. Comp., 31 (1977), pp. 333–390.

[41] H. Brezis, *Functional analysis, Sobolev spaces and partial differential equations*, Universitext, Springer, New York, 2011.

[42] M. Butz, *Computational methods for Cahn–Hilliard variational inequalities*, PhD thesis, University of Regensburg, 2012.

[43] J. W. Cahn, *Free energy of a nonuniform system. II. Thermodynamic basis*, J. Chem. Phys., 30 (1959), pp. 1121–1124.

[44] J. W. Cahn and J. E. Hilliard, *Free energy of a nonuniform system. I. Interfacial free energy*, J. Chem. Phys., 28 (1958), pp. 258–267.

[45] T. F. Chan and J. Shen, *Image Processing and Analysis*, Other Titles in Applied Mathematics, SIAM, Philadelphia, PA, 2005.

[46] J. Chen, L. C. McInnes, and H. Zhang, *Analysis and practical use of flexible BiCGStab*, J. Sci. Comput., (2016), pp. 1–23.

[47] M. Chen, *On the solution of circulant linear systems*, SIAM J. Numer. Anal., 24 (1987), pp. 668–683.

[48] L. Cherfils, H. Fakih, and A. Miranville, *A Cahn–Hilliard system with a fidelity term for color image inpainting*, J. Math. Imaging Vision, 54 (2015), pp. 117–131.

[49] ——, *On the Bertozzi–Esedoglu–Gillette–Cahn–Hilliard equation with logarithmic nonlinear terms*, SIAM J. Imaging Sci., 8 (2015), pp. 1123–1140.

[50] M. I. M. Copetti and C. M. Elliott, *Numerical analysis of the Cahn–Hilliard equation with a logarithmic free energy*, Numer. Math., 63 (1992), pp. 39–65.

[51] T. A. Davis, *Direct Methods for Sparse Linear Systems*, vol. 2 of Fundam. Algorithms, SIAM, Philadelphia, PA, 2006.

[52] ——, *UMFPACK User Guide*, Tech. Report TR-04-003 (revised), Department of Computer Science and Engineering, Texas A&M University, 2016.

[53] S. R. de Groot and P. Mazur, *Non-equilibrium thermodynamics*, Dover Publications, Inc., New York, 1984. Corrected reprint of the 1962 original.

[54] I. C. Dolcetta, S. F. Vita, and R. March, *Area-preserving curve-shortening flows: From phase separation to image processing*, Interfaces Free Bound., 4 (2002), pp. 325–343.

[55] I. S. Duff, A. M. Erisman, and J. K. Reid, *Direct Methods for Sparse Matrices*, Monogr. Numer. Anal., The Clarendon Press, Oxford Univ. Press, New York, 2nd ed., 1989.

[56] C. Eck, H. Garcke, and P. Knabner, *Mathematische Modellierung*, Springer, Berlin Heidelberg, 2nd ed., 2011. In German.

[57] C. M. Elliott, *The Cahn–Hilliard model for the kinetics of phase separation*, in Mathematical Models for Phase Change Problems, J. F. Rodrigues, ed., vol. 88 of Internat. Ser. Numer. Math., Birkhäuser Verlag, Basel, 1989, pp. 35–73.

[58] C. M. Elliott and H. Garcke, *On the Cahn–Hilliard equation with degenerate mobility*, SIAM J. Math. Anal., 27 (1996), pp. 404–423.

[59] ——, *Diffusional phase transitions in multicomponent systems with a concentration dependent mobility matrix*, Phys. D, 109 (1997), pp. 242–256.

[60] C. M. Elliott and S. Luckhaus, *A generalised diffusion equation for phase separation of a multi-component mixture with interfacial free energy*, Tech. Report 887, University of Minnesota, 1991.

[61] C. M. Elliott and A. M. Stuart, *The global dynamics of discrete semilinear parabolic equations*, SIAM J. Numer. Anal., 30 (1993), pp. 1622–1663.

[62] C. M. Elliott and S. Zheng, *On the Cahn–Hilliard equation*, Arch. Ration. Mech. Anal., 96 (1986), pp. 339–357.

[63] H. C. Elman, D. J. Silvester, and A. J. Wathen, *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*, Numer. Math. Sci. Comput., Oxford Univ. Press, Oxford, 2005.

[64] O. G. Ernst and M. J. Gander, *Why it is difficult to solve Helmholtz problems with classical iterative methods*, in Numerical Analysis of Multiscale Problems, I. G. Graham, T. Y. Hou, O. Lakkis, and R. Scheichl, eds., vol. 83 of Lect. Notes Comput. Sci. Eng., Springer, Berlin Heidelberg, 2012, pp. 325–363.

[65] L. C. Evans, *Partial differential equations*, vol. 19 of Grad. Stud. Math., Amer. Math. Soc., Providence, RI, 2nd ed., 2010.

[66] D. J. Eyre, *Systems of Cahn–Hilliard equations*, SIAM J. Appl. Math., 53 (1993), pp. 1686–1712.

[67] ——, *An unconditionally stable one-step scheme for gradient systems*, tech. report, Department of Mathematics, University of Utah, 1998.

[68] R. D. Falgout, *An introduction to algebraic multigrid computing*, Comput. Sci. Eng., 8 (2006), pp. 24–33.

[69] R. Fletcher, *Conjugate gradient methods for indefinite systems*, in Numerical Analysis: Proceedings of the Dundee Conference on Numerical Analysis, 1975, G. A. Watson, ed., vol. 506 of Lect. Notes Math., Springer, Berlin Heidelberg, 1976, pp. 73–89.

[70] R. W. Freund, G. H. Golub, and N. M. Nachtigal, *Iterative solution of linear systems*, Acta Numer., 1 (1992), pp. 57–100.

[71] D. Garcia, *Robust smoothing of gridded data in one and higher dimensions with missing values*, Computat. Statist. Data Anal., 54 (2010), pp. 1167–1178.

[72] H. Garcke, *Mechanical effects in the Cahn–Hilliard model: A review on mathematical results*, in Mathematical Methods and Models in Phase Transitions, A. Miranville, ed., Nova Sci. Publ., New York, 2005, pp. 43–77.

[73] H. Garcke, M. Hinze, and C. Kahle, *A stable and linear time discretization for a thermodynamically consistent model for two-phase incompressible flow*, Appl. Numer. Math., 99 (2016), pp. 151–171.

[74] H. Garcke, B. Nestler, and B. Stoth, *On anisotropic order parameter models for multi-phase systems and their sharp interface limits*, Phys. D, 115 (1998), pp. 87–108.

[75] ——, *A multiphase field concept: Numerical simulations of moving phasee boundaries and multiple junctions*, SIAM J. Appl. Math., 60 (1999), pp. 295–315.

[76] M. W. Gee, C. M. Siefert, J. J. Hu, R. S. Tuminaro, and M. G. Sala, *ML 5.0 smoothed aggregation user's guide*, Tech. Report SAND2006-2649, Sandia National Laboratories, 2006.

[77] A. George and J. W.-H. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1981.

[78] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Acad. Press, Inc. [Harcourt Brace Jovanovich, Publishers], London-New York, 1981.

[79] R. Glowinski, *Numerical methods for nonlinear variational problems*, Sci. Comput., Springer, Berlin, 2008. Reprint of the 1984 original.

[80] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins Stud. Math. Sci., Johns Hopkins Univ. Press, Baltimore, MD, 4th ed., 2013.

[81] G. H. Golub and R. S. Varga, *Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order Richardson iterative methods. I*, Numer. Math., 3 (1961), pp. 147–156.

[82] ——, *Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order Richardson iterative methods. II*, Numer. Math., 3 (1961), pp. 157–168.

[83] C. Gräser and R. Kornhuber, *On preconditioned Uzawa-type iterations for a saddle point problem with inequality constraints*, in Domain Decomposition Methods in Science and Engineering XVI, O. B. Widlund and D. E. Keyes, eds., vol. 55 of Lect. Notes Comput. Sci. Eng., Springer, Berlin Heidelberg, 2007, pp. 91–102.

[84] ——, *Nonsmooth Newton methods for set-valued saddle point problems*, SIAM J. Numer. Anal., 47 (2009), pp. 1251–1273.

[85] C. Gräser, R. Kornhuber, and U. Sack, *Nonsmooth Schur–Newton methods for multicomponent Cahn–Hilliard systems*, IMA J. Numer. Anal., 35 (2015), pp. 652–679.

[86] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, vol. 17 of Frontiers Appl. Math., SIAM, Philadelphia, PA, 1997.

[87] J. Gu, L. Zhang, G. Yu, Y. Xing, and Z. Chen, *X-ray CT metal artifacts reduction through curvature based sinogram inpainting*, J. X-Ray Sci. Technol., 14 (2006), pp. 73–82.

[88] W. Hackbusch, *Multi-Grid Methods and Applications*, vol. 4 of Springer Ser. Comput. Math., Springer, Berlin Heidelberg, 1985.

[89] M. A. Heroux, R. A. Bartlett, V. E. Howle, R. J. Hoekstra, J. J. Hu, T. G. Kolda, R. B. Lehoucq, K. R. Long, R. P. Pawlowski, E. T. Phipps, A. G. Salinger, H. K. Thornquist, R. S. Tuminaro, J. M. Willenbring, A. Williams, and K. S. Stanley, *An overview of Trilinos*, Tech. Report SAND2003-2927, Sandia National Laboratories, 2003.

[90] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Research Nat. Bur. Standards, 49 (1952), pp. 409–436.

[91] M. Hintermüller, M. Hinze, and M. H. Tber, *An adaptive finite-element Moreau–Yosida-based solver for a non-smooth Cahn–Hilliard problem*, Optim. Methods Softw., 26 (2011), pp. 777–811.

[92] M. Hintermüller, K. Ito, and K. Kunisch, *The primal-dual active set strategy as a semismooth Newton method*, SIAM J. Optim., 13 (2002), pp. 865–888.

[93] M. Hintermüller and M. Ulbrich, *A mesh-independence result for semismooth Newton methods*, Math. Program., 101 (2004), pp. 151–184.

[94] F. Hirsch and G. Lacombe, *Elements of Functional Analysis*, vol. 192 of Grad. Texts in Math., Springer, New York, 1999. Translated from the 1997 French original by S. Levy.

[95] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge Univ. Press, Cambridge, 2nd ed., 2013.

[96] S. Hysing, S. Turek, D. Kuzmin, N. Parolini, E. Burman, S. Ganesan, and L. Tobiska, *Quantitative benchmark computations of two-dimensional bubble dynamics*, Internat. J. Numer. Methods Fluids, 60 (2009), pp. 1259–1288.

[97] I. C. F. Ipsen, *A note on preconditioning nonsymmetric matrices*, SIAM J. Sci. Comput., 23 (2001), pp. 1050–1051.

[98] K. Ito and K. Kunisch, *Lagrange Multiplier Approach to Variational Problems and Applications*, vol. 15 of Adv. Des. Control, SIAM, Philadelphia, PA, 2008.

[99] D. Kay, D. Loghin, and A. J. Wathen, *A preconditioner for the steady-state Navier–Stokes equations*, SIAM J. Sci. Comput., 24 (2002), pp. 237–256.

[100] D. Kay and A. Tomasi, *Color image segmentation by the vector-valued Allen–Cahn phase-field model: A multigrid solution*, IEEE Trans. Image Process., 18 (2009), pp. 2330–2339.

[101] D. Kay and R. Welford, *A multigrid finite element solver for the Cahn–Hilliard equation*, J. Comput. Phys., 212 (2006), pp. 288–304.

[102] K. S. Kazimierski, *Aspects of Regularization in Banach Spaces*, Logos, Berlin, 2010.

[103] E. Khain and L. M. Sander, *Generalized Cahn–Hilliard equation for biological applications*, Phys. Rev. E, 77 (2008), p. 051129.

[104] T. Kies, *Bildrekonstruktion durch Anwendung einer Verallgemeinerung der projizierten Gradientenmethode auf ein Phasenfeldmodell*, Master's thesis, Department of Mathematics, University of Regensburg, 2014. In German.

[105] J. Kim, K. Kang, and J. Lowengrub, *Conservative multigrid methods for Cahn–Hilliard fluids*, J. Comput. Phys., 193 (2004), pp. 511–543.

[106] J. S. Kirkaldy and D. J. Young, *Diffusion in the condensed state*, Institute of Metals, 1987.

[107] R. Kornhuber, *Monotone multigrid methods for elliptic variational inequalities. I*, Numer. Math., 69 (1994), pp. 167–184.

[108] ——, *Monotone multigrid methods for elliptic variational inequalities. II*, Numer. Math., 72 (1996), pp. 481–499.

[109] E. Kreyszig, *Introductory Functional Analysis with Applications*, Wiley, 1978.

[110] E. Kuhl and D. W. Schmid, *Computational modeling of mineral unmixing and growth*, Comput. Mech., 39 (2007), pp. 439–451.

[111] P. Kumar, *Fast solvers for nonsmooth optimization problems in phase separation*, in Proceedings of the 2015 Federated Conference on Computer Science and Information Systems, M. Ganzha, L. Maciaszek, and M. Paprzycki, eds., vol. 5 of ACSIS, IEEE, 2015, pp. 589–594.

[112] ——, *An optimal block diagonal preconditioner for heterogeneous saddle point problems in phase separation*, CoRR, abs/1601.03230 (2016).

[113] Yu. A. Kuznetsov, *Efficient iterative solvers for elliptic finite element problems on nonmatching grids*, Russian J. Numer. Anal. Math. Modelling, 10 (1995), pp. 187–211.

[114] O. A. Ladyzhenskaya, *The Boundary Value Problems of Mathematical Physics*, vol. 49 of Appl. Math. Sci., Springer, New York, 1985. Translated from the Russian by A. J. Lohwater.

[115] C. Lanczos, *Solution of systems of linear equations by minimized iterations*, J. Research Nat. Bur. Standards, 49 (1952), pp. 33–53.

[116] L. P. Lebedev, I. I. Vorovich, and G. M. L. Gladwell, *Functional Analysis: Applications in Mechanics and Inverse Problems*, vol. 100 of Solid Mech. Appl., Springer, Dordrecht, 2nd ed., 2002.

[117] A. A. Lee, A. Münch, and E. Süli, *Degenerate mobilities in phase field models are insufficient to capture surface diffusion*, Appl. Phys. Lett., 107 (2015), p. 081603.

[118] H. G. Lee, J.-W. Choi, and J. Kim, *A practically unconditionally gradient stable scheme for the N-component Cahn–Hilliard system*, Phys. A, 391 (2012), pp. 1009–1019.

[119] H. G. Lee and J. Kim, *A second-order accurate non-linear difference scheme for the N-component Cahn–Hilliard system*, Phys. A, 387 (2008), pp. 4787–4799.

[120] P.-L. Lions and B. Mercier, *Splitting algorithms for the sum of two nonlinear operators*, SIAM J. Numer. Anal., 16 (1979), pp. 964–979.

[121] S. H. Lui, *Numerical Analysis of Partial Differential Equations*, Pure Appl. Math. (Hoboken), Wiley, Hoboken, NJ, 2011.

[122] R. B. Morgan, *A restarted GMRES method augmented with eigenvectors*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1154–1171.

[123] J. E. Morral and J. W. Cahn, *Spinodal decomposition in ternary systems*, Acta Metallurgica, 19 (1971), pp. 1037–1045.

[124] M. F. Murphy, G. H. Golub, and A. J. Wathen, *A note on preconditioning for indefinite linear systems*, SIAM J. Sci. Comput., 21 (2000), pp. 1969–1972.

[125] B. Nicolaenko and B. Scheurer, *Low-dimensional behavior of the pattern formation Cahn–Hilliard equation*, in Trends in The Theory and Practice of Non-Linear Analysis. Proceedings of the VIth International Conference on Trends in the Theory and Practice of Non-Linear Analysis, V. Lakshmikantham, ed., vol. 110 of North-Holland Math. Stud., North-Holland, Amsterdam, 1985, pp. 323–336.

[126] A. Novick-Cohen, *The Cahn–Hilliard equation: Mathematical and modeling perspectives*, Adv. Math. Sci. Appl., 8 (1998), pp. 965–985.

[127] A. Novick-Cohen and L. A. Segel, *Nonlinear aspects of the Cahn–Hilliard equation*, Phys. D, 10 (1984), pp. 277–298.

[128] Y. Oono and S. Puri, *Study of phase-separation dynamics by use of cell dynamical systems. I. Modeling*, Phys. Rev. A, 38 (1988), pp. 434–453.

[129] C. C. Paige and M. A. Saunders, *Solutions of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal, 12 (1975), pp. 617–629.

[130] J. W. Pearson and A. J. Wathen, *A new approximation of the Schur complement in preconditioners for PDE-constrained optimization*, Numer. Linear Algebra Appl., 19 (2012), pp. 816–829.

[131] R. L. Pego, *Front migration in the nonlinear Cahn–Hilliard equation*, Proc. R. Soc. A, 422 (1989), pp. 261–278.

[132] T. Rees and M. Stoll, *Block-triangular preconditioners for PDE-constrained optimization*, Numer. Linear Algebra Appl., 17 (2010), pp. 977–996.

[133] T. J. Rivlin, *An Introduction to the Approximation of Functions*, Blaisdell book in numerical analysis and computer science, Dover Publications, Inc., New York, 1981. Corrected reprint of the 1969 original.

[134] J. W. Ruge and K. Stüben, *Algebraic multigrid*, in Multigrid methods, vol. 3 of Frontiers Appl. Math., SIAM, Philadelphia, PA, 1987, pp. 73–130.

[135] Y. Saad, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM J. Sci. Comput., 14 (1993), pp. 461–469.

[136] ——, *Iterative methods for sparse linear systems*, SIAM, Philadelphia, PA, 2nd ed., 2003.

[137] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. and Stat. Comput., 7 (1986), pp. 856–869.

[138] L. Sarbu, *Primal-dual active set methods for Allen–Cahn variational inequalities*, PhD thesis, University of Sussex, 2010.

[139] C.-B. Schönlieb and A. Bertozzi, *Unconditionally stable schemes for higher order inpainting*, Commun. Math. Sci., 9 (2011), pp. 413–457.

[140] J. Shen and X. Yang, *Numerical approximations of Allen–Cahn and Cahn–Hilliard equations*, Discrete Contin. Dyn. Syst., 28 (2010), pp. 1669–1691.

[141] P. Šolín, *Partial differential equations and the finite element method*, Pure Appl. Math., Wiley, Hoboken, NJ, 2006.

[142] W.-H. Steeb, *Matrix Calculus and Kronecker Product with Applications and C++ Programs*, World Sci. Publ., River Edge, NJ, 1997. With the collaboration of T. K. Shi.

[143] M. Stoll, *One-shot solution of a time-dependent time-periodic PDE-constrained optimization problem*, IMA J. Numer. Anal., 34 (2014), pp. 1554–1577.

[144] G. Strang and G. J. Fix, *An Analysis of the Finite Element Method*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1973.

[145] J. L. Troutman, *Variational Calculus and Optimal Control*, Undergrad. Texts Math., Springer, New York, 2nd ed., 1996. With the assistance of W. Hrusa, Optimization with Elementary Convexity.

[146] M. Ulbrich, *Semismooth Newton methods for operator equations in function spaces*, SIAM J. Optim., 13 (2002), pp. 805–841.

[147] H. A. van der Vorst, *Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. and Stat. Comput., 13 (1992), pp. 631–644.

[148] G. Wang, D. Garcia, Y. Liu, R. de Jeu, and A. J. Dolman, *A three-dimensional gap filling method for large geophysical datasets: Application to global satellite soil moisture observations*, Environ. Model. Softw., 30 (2012), pp. 139–142.

[149] A. J. Wathen, *Preconditioning*, Acta Numer., 24 (2015), pp. 329–376.

[150] A. J. Wathen and T. Rees, *Chebyshev semi-iteration in preconditioning for problems including the mass matrix*, Electron. Trans. Numer. Anal., 34 (2009), pp. 125–135.

[151] P. Wesseling, *An Introduction to Multigrid Methods*, Pure Appl. Math. (New York), Wiley, Ltd., Chichester, 1992.

[152] X.-F. Wu and Y. A. Dzenis, *Phase-field modeling of the formation of lamellar nanostructures in diblock copolymer thin films under inplanar electric fields*, Phys. Rev. E, 77 (2008), p. 031807.

[153] M. Zhou, H. Chen, J. Paisley, L. Ren, L. Li, Z. Xing, D. Dunson, G. Sapiro, and L. Carin, *Nonparametric Bayesian dictionary learning for analysis of noisy and incomplete images*, IEEE Trans. Image Process., 21 (2012), pp. 130–144.

[154] J. Zowe and S. Kurcyusz, *Regularity and stability for the mathematical programming problem in Banach spaces*, Appl. Math. Optim., 5 (1979), pp. 49–62.

Schriftliche Ehrenerklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; verwendete fremde und eigene Quellen sind als solche kenntlich gemacht.

Ich habe insbesondere nicht wissentlich:

- Ergebnisse erfunden oder widersprüchliche Ergebnisse verschwiegen,
- statistische Verfahren absichtlich missbraucht, um Daten in ungerechtfertigter Weise zu interpretieren,
- fremde Ergebnisse oder Veröffentlichungen plagiiert oder verzerrt wiedergegeben.

Mir ist bekannt, dass Verstöße gegen das Urheberrecht Unterlassungs- und Schadenersatzansprüche des Urhebers sowie eine strafrechtliche Ahndung durch die Strafverfolgungsbehörden begründen kann.

Die Arbeit wurde bisher weder im Inland noch im Ausland in gleicher oder ähnlicher Form als Dissertation eingereicht und ist als Ganzes auch noch nicht veröffentlicht.

(Ort, Datum)

(Unterschrift)