Thermodynamically Consistent Algorithms for the Solution of Phase-Field Models

Dissertation by

Philippe Antoine Vignal Atherton

In Partial Fulfillment of the Requirements

For the Degree of

Doctor of Philosophy

King Abdullah University of Science and Technology, Thuwal,

Kingdom of Saudi Arabia

February, 2016

The dissertation of Philippe Antoine Vignal Atherton is approved by the examination committee

Committee Chairperson: Victor Manuel Calo

Committee Member: Aurélien C. Manchon

Committee Member: Suzana Nunes

Committee Member: Mazen Al-Ghoul

ABSTRACT

Thermodynamically Consistent Algorithms for the Solution of Phase-Field Models

Philippe Antoine Vignal Atherton

Phase-field models are emerging as a promising strategy to simulate interfacial phenomena. Rather than tracking interfaces explicitly as done in sharp interface descriptions, these models use a diffuse order parameter to monitor interfaces implicitly. This implicit description, as well as solid physical and mathematical footings, allow phase-field models to overcome problems found by predecessors. Nonetheless, the method has significant drawbacks.

The phase-field framework relies on the solution of high-order, nonlinear partial differential equations. Solving these equations entails a considerable computational cost, so finding efficient strategies to handle them is important. Also, standard discretization strategies can many times lead to incorrect solutions. This happens because, for numerical solutions to phase-field equations to be valid, physical conditions such as mass conservation and free energy monotonicity need to be guaranteed. In this work, we focus on the development of thermodynamically consistent algorithms for time integration of phase-field models.

The first part of this thesis focuses on an energy-stable numerical strategy de-

veloped for the phase-field crystal equation. This model was put forward to model microstructure evolution. The algorithm developed conserves, guarantees energy stability and is second order accurate in time. The second part of the thesis presents two numerical schemes that generalize literature regarding energy-stable methods for conserved and non-conserved phase-field models. The time discretization strategies can conserve mass if needed, are energy-stable, and second order accurate in time. We also develop an adaptive time-stepping strategy, which can be applied to any second-order accurate scheme. This time-adaptive strategy relies on a backward approximation to give an accurate error estimator. The spatial discretization, in both parts, relies on a mixed finite element formulation and isogeometric analysis. The codes are available online and implemented in PetIGA, a high-performance isogeometric analysis framework.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Professor Victor M. Calo, for his continuous guidance and unwavering support. He taught me how to think about problems and spent countless hours doing so. He has given me every chance to succeed and for this, I am truly thankful.

I would also like to thank Professors Aurélien Manchon, Suzana Nunes and Mazen Al-Ghoul for taking the time and effort to be on my dissertation committee.

I am indebted to Dr. Nathan Collier and Dr. Lisandro Dalcin, for their limitless patience and generosity. This dissertation would not have been possible without your help. I would like to thank all my friends at Numpor, for making the work environment a fantastic one.

To all my close friends at KAUST, thank you for becoming the family I needed in foreign lands. You made this journey a memorable one.

Finally, I would like to thank my beloved family and my amazing girlfriend. You have been a source of inspiration and taught me never to give up.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

AC	Allen–Cahn
CH	Cahn–Hilliard
IGA	Isogeometric Analysis
NURBS	Non-Uniform Rational B-spines
PDE	Partial Differential Equation
PETSc	Portable, Extensible Toolkit for Scientific Computation
PFC	Phase-field crystal
SH	Swift–Hohenberg

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

The purpose of this thesis is to develop thermodynamically consistent algorithms to evolve in time general classes of phase-field models. Phase-field models have become a powerful tool in the field of computational materials science. As we shall demonstrate, many popular phase-field models can be abstracted and solved using the framework we introduce.

In this Introduction, we review some aspects on numerical modeling of interfacial phenomena to justify the choice of phase-field methods as a modeling tool, as well as the use of isogeometric analysis as a numerical method in high-order partial differential equation. In chapter 2, we go through the physical derivation of the Cahn–Hilliard equation, one of the most successful phase-field models to date. This derivation explains the solid mathematical and physical footings that support the method, and motivates the development of energy-stable methods capable of guaranteeing numerical energy dissipation. In chapter 3, we cover a numerical scheme developed for the phase-field crystal equation [5], which is provably energy-stable, guarantees mass conservation, and is second-order accurate in time. We ensure energy stability through a convex splitting of the nonlinear term present in the equation and the use of a stabilization parameter. In chapter 4, we go further and develop two implicit methods

that generalize literature on the topic of stable methods for both conserved and non-conserved phase-field models. In this chapter, we guarantee energy stability through the use of Tayor expansions. We also develop an adaptive time-stepping scheme, whose time step size control relies on the approximation of a local truncation error through a backward differentiation procedure. This adaptive strategy has the added advantage that it can be applied to any second-order accurate scheme, and in particular, to both methods developed in 4. The time discretization, in chapters 3 and 4, relies on a mixed finite element formulation and isogeometric analysis. The codes are available online and implemented in PetIGA, a high-performance isogeometric analysis framework.

The main body of the dissertation is structured so that, to a large extent, each chapter can be read independently. This requires repetition of content, but we believe the advantages in readability outweigh the disadvantages.

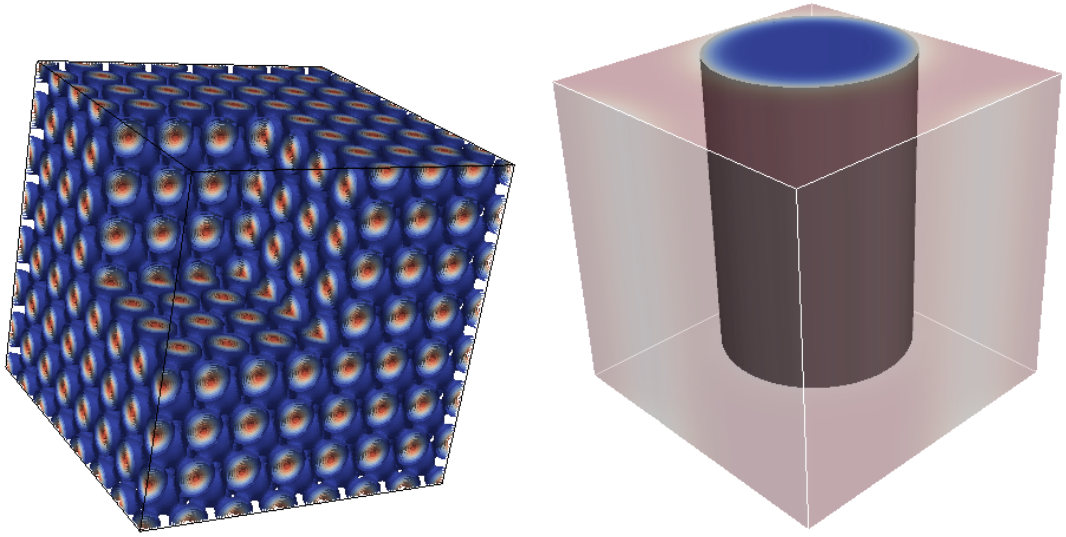
1.1 Modeling interfacial phenomena

The boundary between two phases, the interface, can have very different properties from that of the bulk phases [6] and is important in a variety of engineering processes. The interface may control transport between the phases (liquid-liquid extraction), reaction rates (heterogeneous catalysis), or accumulation of a component from one of the bulk phases (adsorption). All of these are examples of interfacial phenomena that are non-trivial to study experimentally, where modeling has allowed to make meaningful discoveries [7, 8].

Modeling has become a ubiquitous part of state-of-the-art engineering design, and its role in the materials setting is quite diverse. Modeling today plays a role in the quest to control materials [9] and the processes used to produce them. At the most fundamental level, modeling delivers an understanding of materials and

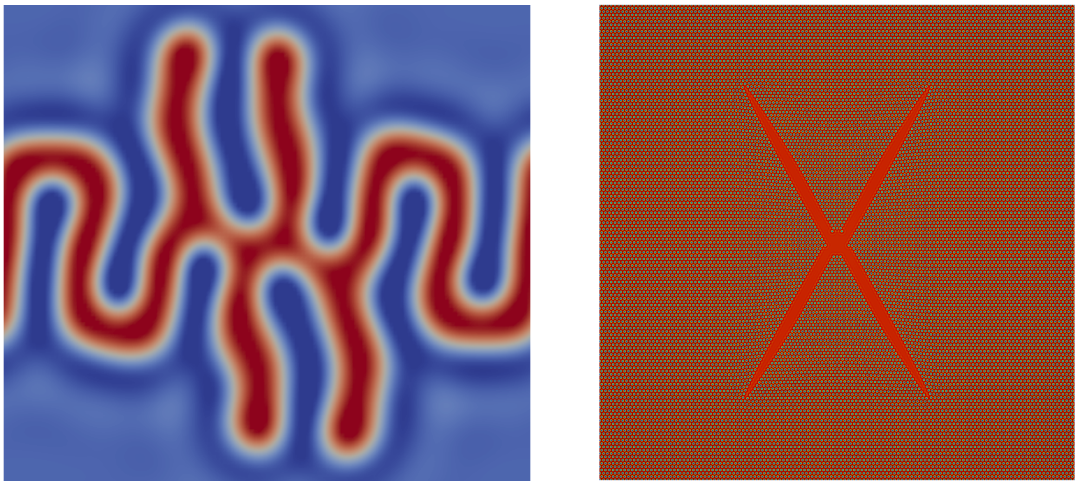
tries to predict their response to external stimuli [10]. Most materials are heterogeneous at the mesoscale. Their microstructure consists of grains or domains, which differ in structure, orientation, and chemical composition. The physical and mechanical properties at the macroscopic scale highly depend on the shape, size, and distribution of the grains. Gaining insight into the mechanisms of microstructure formation and evolution is, therefore, important. Given the diversity of processes involved in microstructure evolution, extensive theoretical and experimental research are required. Within this domain, the phase-field method has become a powerful tool for simulating microstructural evolution in a variety of material processes, such as solidification, solid-state phase transformations, precipitate growth and coarsening, martensitic transformations and grain growth [1]. Computational results obtained by solving some of the most popular phase-field models in recent years are shown in figure 1.1.

The theory behind phase-field models, started by Van der Waals [11] over a century ago, has taken off in the last couple of decades [1, 12]. One of the fundamental postulates presented in [11] is that the interface possesses non-zero thickness, accross which the physical properties are smoothly distributed. This is precisely what takes place in the phase-field framework, and explains why this method is also known as the *diffuse-interface* approach [1]. Interfaces are implicitly described by continuous scalar-valued fields that take constant values in the bulk phases and vary continuously but steeply across a diffuse front [13]. In this way, problems that come about due to the explicit tracking of interfaces, as is the case in *sharp-interface* approaches, are avoided. In these sharp-interface problems, explicit boundary conditions need to be defined at the interfaces [14], which rely on complicated and expensive numerical simulations. A popular sharp-interface model is the Stefan problem [15], which models the temperature distribution in a homogeneous medium undergoing a phase transition



(a) Crystal growth

(b) Spinodal decomposition



(c) Pattern formation

(d) Crack propagation

Figure 1.1: Understanding phase transitions through simulation. The phase-field method has successfully been applied to a wide array of interfacial problems including crystal growth and crack propagation with the phase-field crystal equation (figures 1.1(a) and 1.1(d)), spinodal decomposition with the Cahn-Hilliard equation (figure (1.1(b))) and pattern formation with the Swift-Hohenberg equation (figure (1.1(c))).

(e.g., water turning into ice), and can be expressed as

$$\frac{\partial T}{\partial t} = \nabla \cdot \left(\frac{k}{\rho c_p} \nabla T \right) \equiv \nabla \cdot (\alpha \nabla T), \tag{1.1}$$

$$\rho L_f V_n = k_S \nabla T \cdot \mathbf{n}_S - k_L \nabla T \cdot \mathbf{n}_L, \tag{1.2}$$

$$T_{\text{int}} = T_M - \left(\frac{\gamma T_M}{L_f} \right) \kappa - \frac{v_n}{\mu}, \tag{1.3}$$

where $T \equiv T(\mathbf{x}, t)$ denotes temperature, k is the thermal conductivity, which assumes different values in the solid (k_S) and the liquid phases (k_L), ρ is the density, c_p is the specific heat at constant pressure, α is the thermal diffusion coefficient, L_f the latent heat of fusion for solidification, v_n the local normal velocity of the interface (positive for a growing solid), $\mathbf{n}$ is the normal vector at the interface of the solid ($\mathbf{n}_S$) or the liquid ($\mathbf{n}_L$), γ is the solid-liquid surface energy, T_M is the melting temperature, κ is the local solid-liquid interface curvature and μ the local atomic interface mobility. This model can be used to describe the solidification of a pure substance, where the motion of the solidification front is limited by the diffusion of latent heat away from the solid-liquid interface [16]. This moving front also depends on the ability of the interface to maintain two specific boundary conditions. The first boundary condition involves the flux of heat toward one side of the interface, which needs to be balanced by an equivalent flux away from the interface as expressed in equation (1.2). The second boundary condition, described through equation (1.3) involves the temperature at the interface, which undergoes a curvature correction popularly known as the Gibbs-Thompson condition.

Other diffusion-limited phase-transformations exist, whose interface properties can be described by equations similar to (1.1)-(1.3). Models such as these usually operate on larger scales than the solid-liquid interface width, itself of atomic dimensions. Consequently, relevant information from the atomic level can be taken into account through effective constants such as the capillary length, which depends on surface en-

ergy, the kinetic attachment coefficient, and the thermal impurity diffusion coefficient. Nonetheless, these sharp-interface models face severe difficulties from both the simulation as well as the physical point of view. From the numerical simulation standpoint, solving equations (1.2)-(1.3) in a multidimensional setting is a nontrivial challenge. Topologically complex interfaces can merge or pinch-off during a phase transformation, and these situations are usually addressed through somewhat arbitrary criteria, as illustrated through figure 1.2. In extreme cases, the interface topology is adjusted manually. From the physical standpoint, even though the general Stefan problem described above is well suited to represent a simplified case of isotropic solidification without convection, and heat transport only takes place by diffusion [13], similar equations for more complicated problems are often unknown. This is the case of phase separation when mobile dislocations and their effect on domain coarsening is included [17], where further research is needed.

Other strategies had considerable success modeling interfacial phenomena, namely the volume-of-fluid [18] and the level-set [19] methods. Even though these two interface-capturing methods use additional unknowns to identify the different phases, they tend to be less accurate than the interface-tracking strategies discussed previously. They are nonetheless far easier to implement, and more efficient computationally speaking given that the mesh does not need to be constantly updated [20]. They are also better suited to handle hard topological transitions without ad hoc techniques. Unfortunately, both methods have important shortcomings. The volume-of-fluid method requires the calculation of a curvature, which is hard to perform accurately and leads to problems in the calculation of the physical quantities near the interface [21] as it is artificially smoothed by the numerical method. The level-set method captures a more accurate interface, but does not conserve mass as time evolves [20]. The phase-field method can overcome these issues.

Other features that validate the use of phase-field methods include the method of

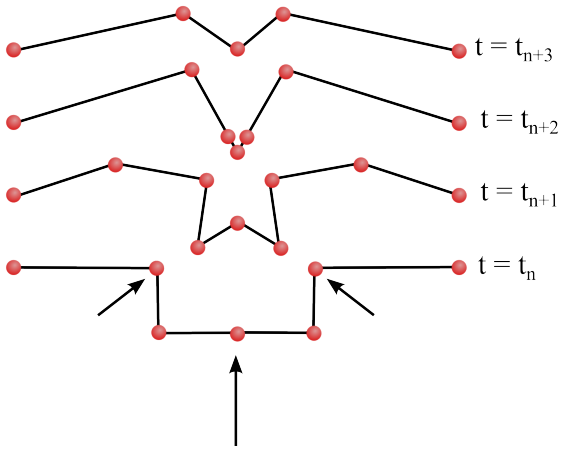


Figure 1.2: Explicit interface tracking in a solidification process. Nodes are placed on the interface, which evolves in time. Given the topologically complex structure it has, the interface can undergo merging and pinch-off during the course of the phase transformation. As velocities might differ along the solidification front, choices need to be made when merging or pinch-off takes place.

matched asymptotic expansions. That is, some phase-field models under appropriate assumptions asymptotically converge to their sharp-interface counterparts, when decreasing the interface width [22, 23, 24]. The method of asymptotic expansions, where the bulk and interfacial scales are separated by using expansions of the primary variables, can be used to understand the model behavior, but more importantly, to establish the link between the model parameters and the physical parameters of the system that is represented. It is a variant of boundary-layer methods used in fields such as fluid mechanics or porous media [25] where transition layers are present. These facts support the use of the phase-field method to model phase transitions.

Within this context of solidification, the phase-field approach can be used to represent the interface between the liquid and solid phases. Within each one of the bulk phases, the phase-field variable has nearly constant values, that relates to the degree of ordering in each phase. The interface between the liquid and the solid is defined as a narrow region where the phase-field variable varies continuously and steeply between the neighboring phases, as seen in figure 1.3. The temporal evolution of the position of the interfaces, is implicitly given by the evolution of the phase-field vari-

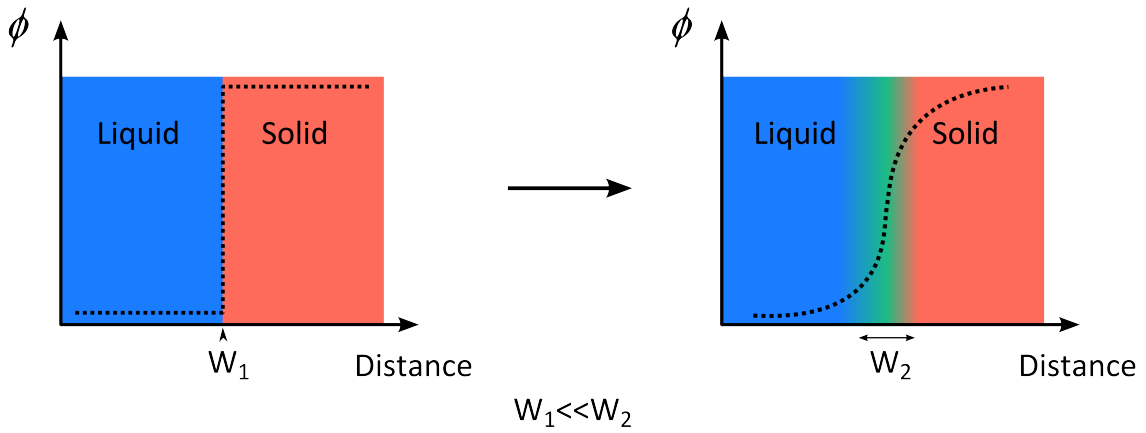


Figure 1.3: Illustrating the phase-field parameter. Rather than dealing with an infinitesimal width as in the case of sharp-interface models, in phase-field models the interface is spread over a larger domain, rendering it diffuse. This is why the method is also known as the diffuse-interface approach [1].

ables. Herein lies one of the advantages of the phase-field method, as the interface no longer needs to be tracked explicitly. Another advantage given by this method involves its phenomenological character, as the equations governing the evolution of the phase field are derived based on general thermodynamic and kinetic principles [26].

In the variational approach to the phase-field method, which leads to partial differential equations (PDEs) that describe the temporal evolution of the phase field, a potential to characterize the system is defined and minimized with respect to the phase-field parameter. By coupling the phase-field parameter to an accurate thermodynamic description of the system, no artificial choices need to be made. The diffuse interface is introduced through an energetic variational procedure that results in a thermodynamically consistent formulation [1]. This gives the method as a whole a solid mathematical and physical background, and explains why applications range from the spinodal decomposition of immiscible binary mixtures [27, 28, 29], solidification and crystal growth [30, 31, 32], tumour angiogenesis [33, 34], wetting [35], image processing [36] to water infiltration in porous media [37].

Even though phase-field modeling has been qualitatively successful, it is not without its drawbacks. Phase-field modeling usually requires the solution of high-order

PDEs and including realistic features comes at the expense of nonlinear terms (such as a nonlinear mobility for the different components involved [28, 29]). Given the nature of the PDEs, using explicit methods is restrictive for stability reasons. This has motivated research on implicit algorithms [28, 38] as well as adaptive algorithms [39, 40]. Another issue has to do with the resolution used to describe the interfacial thickness. If the interfacial layer is not well resolved, the solution can have spurious oscillations, particularly in the vicinity of the interface [37, 41]. Irrespective of the discretization method used, these last two points lead to the need for robust and scalable software to solve these types of problems. Last but not least, phase-field models should possess strong energy stability [42]: free energy must decrease in time. Numerical techniques to satisfy thermodynamic relations at the discrete level in phase-field models have been developed for both conserved [28, 3] (i.e., phase-field associated with the concentration of a component) and non-conserved phase-field models [4, 43] (i.e., phase-field related to the magnetic moment). The main goal of this work is to devise stable time discretization schemes that apply to general classes of phase-field models.

Regarding the spatial discretization, collocation methods such as the finite difference method [42, 44] and spectral methods [45, 46] are the standard methodologies used to solve phase-field models. Nonetheless, these methods face challenges when dealing with complex geometries. These issues can be overcome by using the finite element method [47]. In this work, we solve the resulting PDEs using isogeometric analysis (IGA) [48]. IGA is a finite element method which uses Non-Uniform Rational B-spines (NURBS) [49] as basis functions, rather than standard piecewise polynomials. The method has successfully been applied in its Galerkin version to solve the Cahn-Hilliard equation [28, 2], the advective [50] and Navier-Stokes-Cahn-Hilliard equations [51], the Swift-Hohenberg equation [4] and the phase-field crystal equation [3, 52, 5]. IGA possesses some advantages over conventional finite element methods, which include being able to easily generate high-order, globally continuous

basis functions as well as an exact geometrical representation as the space is refined. However, when used to discretize these high-order partial differential equations, the NURBS-based spaces result in expensive linear systems [53, 54, 55] that need efficient and scalable solvers that mitigate the increase of cost. For this reason, we use PetIGA, a high-performance isogeometric analysis framework [56, 57]. Recently, isogeometric collocation methods have also been used to discretize phase-field models [58, 59], as well as fast solution strategies to solve the resulting algebraic systems of equations [60, 61, 62, 63].

1.2 Thesis layout

1.3 Objectives and Contributions

The contributions of this thesis fold in the following streams: (summarize the chapters)

- Construction of a second-order accurate thermodynamically consistent algorithm for time integration of general phase-field models.
- Development of an adaptive time-stepping strategy for second-order accurate systems.
- Development of open-source applications for solving phase-field equations in a high-performance computing framework.
- Application of these numerical schemes to material science problems.

Chapter 2

The Phase-Field Method

Variations in the development of phase-field equations exist [1, 14]. Many of the key points needed in the description of the method as it relates to this work can be showcased through the physical derivation of the Cahn–Hilliard equation. Essentially, the phase-field framework requires the formulation of an appropriate thermodynamic potential for a given system, which allows to describe mathematically a thermodynamic equilibrium situation [64]. The equilibrium state is defined as the one for which the governing thermodynamic potential becomes extremal. The homogeneous equilibrium states the system takes can be described by defining an appropriate phase field. To describe the inhomogeneity the interface entails, the gradients of the phase field, and possibly higher-order derivatives of the phase field, can be included in the free energy description of the system [27]. Then, by using the formulated free energy functional within an irreversible-thermodynamics setting [26], evolution equations of the phase field parameter can be derived. The transport equations used distinguish between locally conserved and non-conserved quantities. Following the outlined procedure, it is possible to get diffuse-interface models that are both thermodynamically and mathematically consistent.

The chapter is structured as follows: in section 2.1, we go through the derivation of

the Cahn–Hilliard equation and present its mathematical formulation. In section 2.2, we introduce three more phase-field models that have had considerable success in modeling different phenomena. These are the Allen–Cahn [65], the Swift–Hohenberg [66] and the phase-field crystal [30] equations. We conclude with section 2.3, which explains the motivation behind physically consistent algorithms that are the main focus of this thesis.

2.1 Understanding the method through the Cahn–Hilliard equation (CH)

The Cahn-Hilliard equation is a fourth-order, nonlinear partial differential equation. Initially derived to model phase separation of immiscible fluids [27, 67], modified versions of the equation have since then been used in the context of image processing [36, 68], water infiltration in porous media [37], partial wetting [35], celestial mechanics [69], and tumor angiogenesis [33, 34]. The ideas behind the Cahn–Hilliard equation are responsible for what the field of phase-field modeling has become today.

2.1.1 Mathematical formulation

Let $\Omega \in \mathbb{R}^d$ be an open set, where $d = 2, 3$. The boundary of Ω with unit outward normal $\boldsymbol{n}$ is denoted $\partial\Omega$ and is composed of two complementary parts $\partial\Omega_g$ and $\partial\Omega_h$, such that $\partial\Omega = \overline{\partial\Omega_g \cup \partial\Omega_h}$. The phase-field framework relies on finding the fields that yield the minima within Ω of the free energy functionals under consideration. The free energy functional used for the Cahn–Hilliard equation, $\mathcal{F}_{CH}$, often referred to as the Ginzburg-Landau free energy [27], is given by

$$\mathcal{F}_{CH}\left[c(\boldsymbol{x}, t)\right] = \int_{\Omega} \Psi(c) + \frac{\gamma}{2}|\nabla c|^2 d\Omega, \tag{2.1}$$

where c denotes the concentration of one of the components of the mixture, and depends on space $\boldsymbol{x}$ and time t, γ is a positive constant such that $\sqrt{\gamma}$ represents a length scale of the problem related to the interface thickness between the two phases. The term $\frac{\gamma}{2}|\nabla c|^2$ represents the contribution to the free energy coming from the interface [14], while $\Psi(c)$ is the bulk free energy density that includes entropic effects, given by

$$\Psi(c) = \frac{1}{\theta}\left(c\ln(c) + (1-c)\ln(1-c)\right) + 2c(1-c). \tag{2.2}$$

The dimensionless number θ represents the ratio between critical and absolute temperatures. The equation can then be stated in strong form as: find $c : \overline{\Omega} \times (0,T) \mapsto \mathbb{R}$ such that

$$\begin{aligned}
\frac{\partial c}{\partial t} - \nabla \cdot \left(M_c \nabla \left(\Psi'(c) - \gamma \Delta c\right)\right) &= 0 && \text{in} \quad \Omega \times (0,T), \\
c &= g && \text{on} \quad \partial\Omega_g \times (0,T), \\
M_c \nabla \left(\Psi'(c) - \gamma \Delta c\right) \cdot \boldsymbol{n} &= h && \text{on} \quad \partial\Omega_h \times (0,T), \\
M_c \gamma \nabla c \cdot \boldsymbol{n} &= 0 && \text{on} \quad \partial\Omega \times (0,T), \\
c\left(\boldsymbol{x}, 0\right) &= c_0\left(\boldsymbol{x}\right) && \text{in} \quad \overline{\Omega},
\end{aligned}$$

where c_0 is the initial concentration, M_c is the mobility, and $\Psi'(c)$ is given by $\dfrac{\partial \Psi(c)}{\partial c}$. The mobility and chemical potential are nonlinear functions of the concentration, respectively defined as

$$\begin{aligned}
M_c &= Dc(1-c), \\
\Psi'(c) &= \frac{1}{\theta}\ln\frac{c}{1-c} + 2\theta(1-2c),
\end{aligned}$$

in which D is a positive constant with dimensions of diffusivity. This is the dimensionless version of the equation, and we will derive it in the following section. We follow the simplified version of [70] for the sake of brevity, but more thorough derivations can be found in [27, 71, 1, 14].

2.1.2 Physical derivation

In this section, we derive the Ginzburg–Landau free energy presented in equation (2.1). This equation expresses the total free energy of a volume Ω of an isotropic system with a non-uniform composition, which we recast as

$$F(c) = \int_{\Omega} V(c) + \frac{\gamma}{2}|\nabla c|^2 d\Omega, \tag{2.3}$$

where V is the free energy of the homogeneous system, ∇c represents the composition gradient, and γ is a constant parameter for a regular solution [27]. Two parts make up this equation for the free energy. The first contribution calculates the free energy this same volume would have in a homogeneous solution, whereas the second contribution, a gradient energy, accounts for the change in local composition [71].

To derive equation (2.3), assume a binary fluid composed of A and B particles in the limit where diffusion is the major transport mechanism. Let us define the phase field ϕ, such that $\phi = 0$ is a phase completely made of A particles, and $\phi = 1$ is a phase completely made of B particles, with a linear interpolation between those two phases. We assume the interactions of similar particles (A-A/B-B) to be favorable, while the interaction between different particles is unfavorable. This unfavorable interaction is taken into account by $\nabla\phi$. Due to inter-molecular interactions and interactions with any external field, we assume a potential energy landscape, $V(\phi, T)$, to have a temperature dependent transition from a random, high temperature phase, to an ordered, low temperature phase. Invoking mean field approximations [14] and regular

solution theory [27], V can be expressed as

$$V(\phi, T) = k_B\Big(2T_c\phi(1-\phi) + T\big(\phi\ln(\phi) + (1-\phi)\ln(1-\phi)\big)\Big). \tag{2.4}$$

Plotting equation (2.4) for different values of temperature T results in figure 2.1. The plot shows the existence of a single stable state for temperatures above the critical temperature T_c, as well as the emergence of two continuous states when the temperature goes below T_c. Through the double tangent construction [72], we can determine the equilibrium position ϕ_{eq} of these two wells

$$\left.\frac{\partial V}{\partial \phi}\right|_{\phi_{eq}} = 0, \tag{2.5}$$

which results in

$$\phi_{eq} - \frac{1}{2} = \frac{1}{4}\frac{T}{T_c}\ln\left(\frac{\phi_{eq}}{1-\phi_{eq}}\right). \tag{2.6}$$

Two solutions exist for this equation, ϕ_1 and ϕ_2 in figure 2.1, when $T < T_c$. This form of the potential is such that below a critical temperature two states emerge continuously from one.

Then, consider a total free energy functional F, which is a function of the phase-field ϕ, such that

$$F(\phi) = N_\Omega \int_\Omega h \; d\Omega \tag{2.7}$$

where the local energy per molecule, h, in a non-uniform composition region is given by $h(\phi, \nabla\phi, \Delta\phi, \cdots)$. Moreover, we use a multivariate Taylor series to expand h about $\phi_0 = (\phi, 0, 0, \cdots)$, the free energy per molecule of a solution with constant composition (i.e., $V(\phi)$). If we ignore terms of order higher than two, consider the

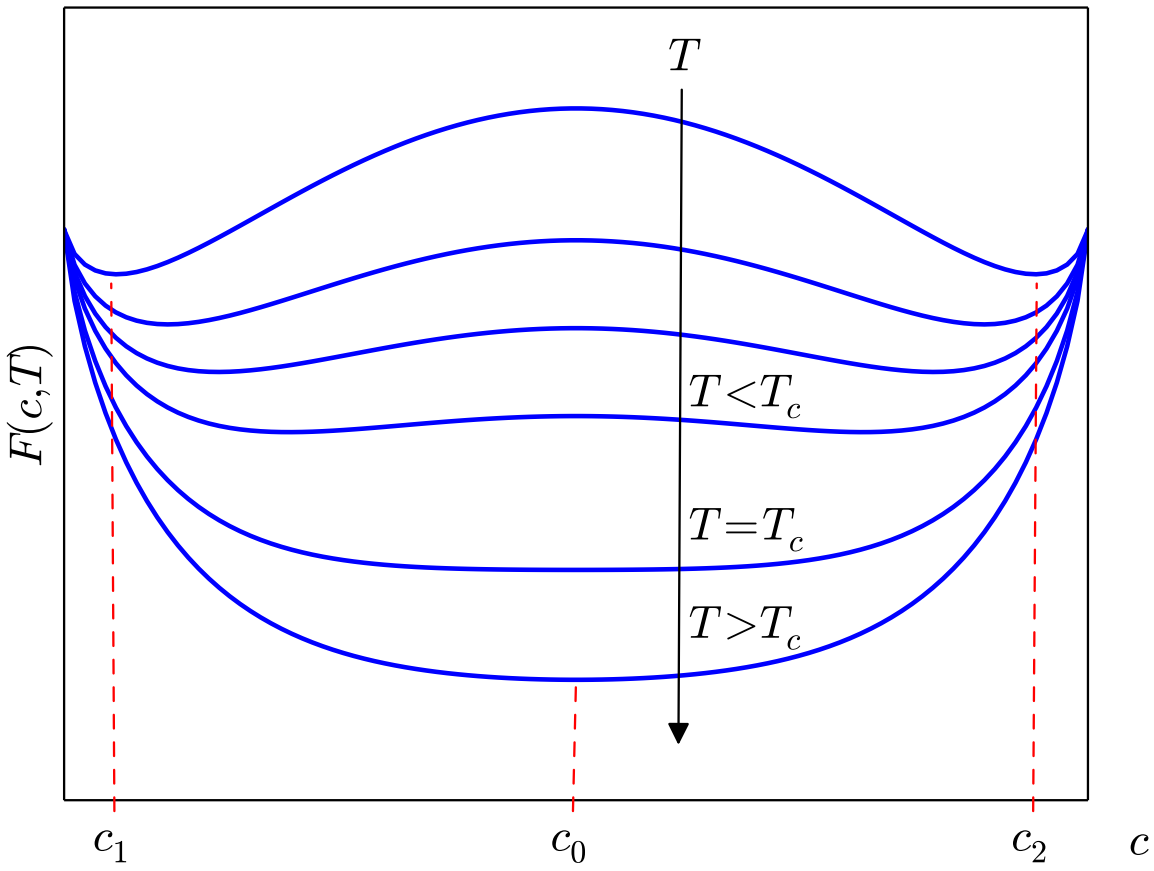


Figure 2.1: Free energy diagram of a binary fluid as a function of the phase field. When the temperature T is higher than the critical temperature T_c, the potential V has one minimum energy state at $\phi = \phi_0$, where $\frac{\partial V}{\partial \phi} = 0$. When the temperature is lowered to induce solidification, the potential V has two minima, located at ϕ_1 and ϕ_2. This phase separation phenomena takes place when the binary fluids are immiscible, with two stable states arising continuously from one for $T < T_c$. The pressure P is assumed constant.

medium as isotropic as well as the invariance to rotations and reflections the system possesses, it can be shown that [27, 71]

$$h(\phi, \nabla\phi, \Delta\phi, \cdots) = V(\phi) + \kappa_1 \Delta\phi + \frac{\kappa_2}{2}|\nabla\phi|^2 + \cdots, \tag{2.8}$$

where

$$\kappa_1 = \frac{\partial h(\phi_0)}{\partial \phi_{ii}}, \quad \kappa_2 = \frac{\partial h(\phi_0)}{\partial \phi_{ii}} \text{ for } i = 1, 2, 3. \tag{2.9}$$

with $V(\phi)$ now understood to be the Helmholtz free energy density. Integrating equation (2.8) over Ω, we obtain that the total free energy of the volume is

$$F(\phi) = N_\Omega \int_\Omega h \, d\Omega \tag{2.10}$$

$$= N_\Omega \int_\Omega \left(V(\phi) + \kappa_1 \Delta\phi + \frac{\kappa_2}{2}|\nabla\phi|^2 + \cdots \right) d\Omega. \tag{2.11}$$

Then, integrating the term $\kappa_1 \Delta\phi$ by parts under the assumption that the term $\dfrac{\partial \phi}{\partial n}$ vanishes at the boundary [27], we get that

$$\int_\Omega (\kappa_1 \Delta\phi) \, d\Omega = -\int_\Omega \left(\frac{\partial \kappa_1}{\partial \phi} |\nabla\phi|^2 \right) d\Omega. \tag{2.12}$$

Finally, substituting (2.12) in (2.10), we recover

$$\begin{aligned} \frac{F(\phi)}{N_\Omega} &= \int_\Omega \left(V(\phi) + \kappa_1 \Delta\phi + \frac{\kappa_2}{2}|\nabla\phi|^2 + \cdots \right) d\Omega \\ &= \int_\Omega \left(V(\phi) + \left(-\frac{\partial \kappa_1}{\partial \phi} + \frac{\kappa_2}{2} \right) |\nabla\phi|^2 + \cdots \right) d\Omega \\ &= \int_\Omega \left(V(\phi) + \frac{\Gamma}{2}|\nabla\phi|^2 + \cdots \right) d\Omega, \end{aligned} \tag{2.13}$$

with $\Gamma \equiv 2\left(-\dfrac{\partial \kappa_1}{\partial \phi} + \dfrac{\kappa_2}{2} \right)$. Disregarding higher-order terms, and substituting equa-

tion (2.4) in (2.13), we now have an energy functional for the entire field ϕ in space and time [14, 70]

$$\mathcal{F}^* = \int \left[\frac{1}{\theta} \big(\phi \ln(\phi) + (1-\phi) \ln(1-\phi) \big) + 2\phi(1-\phi) + \frac{\gamma}{2} |\nabla \phi|^2 \right] d\Omega, \tag{2.14}$$

where $\mathcal{F}^* \equiv \dfrac{\mathcal{F}\left[\phi(\boldsymbol{x}, t)\right]}{k_B T_c}$, $\theta \equiv \dfrac{T_c}{T}$, and $\gamma \equiv \Gamma k_B T_c$ a constant that penalizes phase boundaries. If ϕ is understood as a measure of particle number, and $\mathcal{F}^*$ is the energy of a particular configuration of ϕ, then the variation in $\mathcal{F}^*$ with respect to ϕ is quantifying how the energy changes when particles change position (i.e., the chemical potential μ)

$$\frac{\delta \mathcal{F}^*}{\delta \phi} = \mu, \tag{2.15}$$

where $\dfrac{\delta}{\delta \phi}$ defines the variational derivative operator, given by

$$\frac{\delta}{\delta \phi} = \frac{\partial}{\partial \phi} - \nabla \cdot \frac{\partial}{\partial \nabla \phi} + \Delta \frac{\partial}{\partial \Delta \phi}, \tag{2.16}$$

where $\nabla \cdot$, ∇ and Δ denote the divergence, gradient, and Laplacian operators, respectively. Recalling Fick's law, which states that the flux $\boldsymbol{J}$ of particles in a system is proportional to the gradient of the chemical potential, we have that

$$\boldsymbol{J} = -M \nabla \mu, \tag{2.17}$$

where M represents a diffusivity (mobility of the solute), usually defined as [28]

$$M = D\phi(1-\phi). \tag{2.18}$$

Considering particles need to be conserved, and as such, that the flux must obey a

continuity equation, we have that

$$\frac{\partial \phi}{\partial t} + \nabla \cdot \boldsymbol{J} = 0, \tag{2.19}$$

or, using equation (2.17),

$$\frac{\partial \phi}{\partial t} = \nabla \cdot (M \nabla \mu) \,. \tag{2.20}$$

To conclude the derivation, we need to calculate the chemical potential defined in equation (2.15),

$$\mu = \frac{\delta \mathcal{F}^*}{\delta \phi} = \frac{1}{\theta} \ln \frac{\phi}{1-\phi} + 2(1-2\phi) - \gamma \Delta \phi. \tag{2.21}$$

Finally, substituting the previous expression for the chemical potential in equation (2.20), we have that

$$\frac{\partial \phi}{\partial t} = \nabla \cdot \left(M \nabla \left(\frac{1}{\theta} \ln \frac{\phi}{1-\phi} + 2(1-2\phi) - \gamma \Delta \phi \right) \right), \tag{2.22}$$

where M is playing the role of an effective diffusion coefficient. This is the Cahn–Hilliard equation, which can be used to describe how a high temperature, disordered phase with an average concentration set to ϕ_0 will undergo phase separation when the temperature goes below the melting temperature. Under these conditions, the process is governed by the gradients in chemical potential between phases. The free energy is built to reflect the symmetries present in the phase diagram of the transition that is taking place [1, 14, 13]. This is one of the strengths of phase-field modeling, as a correct free energy choice will yield physically meaningful results, given the tight connection to the thermodynamics of the system. Non-conserved phase-field parameters can also be considered. This can be done by considering a different

evolution equation, defined as

$$\frac{\partial \phi}{\partial t} = -M\mu. \tag{2.23}$$

This equation is known as the Allen–Cahn equation [13, 14], which has successfully been used to model phase transitions involving ferromagnetic materials [73]. To this point, we have only considered thermal equilibrium, but the partial differential equation for the phase field can also be coupled to the change in temperature [1, 74], so that non-isothermal systems can be studied. A lot of ongoing work focuses on modifying these models, as well as the physical parameters involved (interfacial energies, correlation functions, mobilities) for particular purposes, in order to try to recover accurate quantitative results. One of the modifications to deal with a model that could capture elasticity, resulted in the phase-field crystal model [30]. This model is discussed in the following section.

To simplify the numerical solution of the Cahn–Hilliard equation, given the presence of the logarithmic free energy and the existence of singularities at $\phi = 0$ and $\phi = 1$, the logarithmic free energy density is generally approximated by a quartic polynomial function [75]. Equation (2.2) in this case becomes

$$\Psi(\phi) = \frac{1}{4}a_0 - \frac{1}{2}a_2\phi^2 + \frac{1}{4}a_4\phi^4, \tag{2.24}$$

where a_0, a_2 and a_4 are real-valued constants, a_2 and a_4 are positive, and the physical range of the phase-field parameter ϕ lies in $\left]-\sqrt{\frac{a_2}{a_4}}, \sqrt{\frac{a_2}{a_4}}\right[$. This polynomial approximation preserves various physical attributes required for phase separation (such as the binodal and spinodal points) but comes at a cost. The dynamics are faster when using the polynomial approximation of equation (2.24) while the interfacial thickness is artificially affected [75]. Nonetheless, the quartic polynomial function is chosen for this work as current physical and mathematical problems dealing with this version of

the Cahn–Hilliard equation are being solved today [76, 77, 78] Other relevant phase-field models being researched also employ it, such as the Swift–Hohenberg equation and the phase-field crystal equation [14, 79]. These models will be addressed in the following section, and are the ones this work focuses on.

2.2 Gradient flow equations

This section introduces three phase-field models that have had considerable success in modeling different phenomena [14, 1, 12], as well as the simplified version of the Cahn–Hilliard equation [43]. These are the Allen–Cahn [65], the Swift–Hohenberg [66] and the phase-field crystal [30] equations. The Allen–Cahn and Cahn–Hilliard equations, probably the most successful phase-field models to date, are derived as gradient flows of the same free energy functional [43]. The same statement can be made regarding the Swift–Hohenberg and the phase-field crystal equation, which are also derived from the same free energy functional. We detail in the following the fundamental free energy from which each phase-field model considered is derived from, as well as the associated mathematical strong form.

2.2.1 The Allen–Cahn equation (AC)

The Allen–Cahn equation governs the behavior of the Ising ferromagnet [13], where the competition between energy and entropy gives way to a phase transition below some critical temperature [65]. The free energy functional for the system is given by

$$\mathcal{F}_{AC} = \int_{\Omega} \left(\Psi_{AC} + \frac{1}{2} \left| \nabla \phi \right|^2 \right) d\Omega,$$

where $\Psi_{AC} = \frac{1}{4\epsilon^2} \left(\phi^2 - 1 \right)^2$ and ϵ is a parameter related to interface thickness. The phase-field parameter is related to the ordering of the phase, which translates to it

being a non-conserved quantity. The partial differential equation that describes this order-disorder transition is given by

$$\frac{\partial \phi}{\partial t} = -\left(\Psi'_{AC}\left(\phi\right) - \Delta\phi\right),$$

where a constant mobility of value one is commonly considered.

2.2.2 The Cahn–Hilliard equation (CH)

The Cahn–Hilliard equation governs the evolution of an immiscible binary mixture undergoing phase separation [27]. This is the equation that popularized the use of the phase-field method [14]. In this equation, the phase field represents the concentration of one of the components of the mixture. The dimensionless free energy functional is given by

$$\begin{aligned}\mathcal{F}_{CH} &= \mathcal{F}_{AC} \\ &= \int_{\Omega}\left(\Psi_{CH} + \frac{1}{2}\left|\nabla\phi\right|^2\right) d\Omega \end{aligned} \tag{2.25}$$

where $\Psi_{CH} = \dfrac{1}{4\epsilon^2}\left(\phi^2 - 1\right)^2$. The parameter ϵ is related to the thickness of the interface. The partial differential equation, given that the phase-field is in this case a conserved quantity, is given by

$$\frac{\partial \phi}{\partial t} = \nabla \cdot \left(M_{CH}\nabla\Psi'_{CH}\left(\phi\right) - \Delta\phi\right),$$

with $M_{CH} = \bar{M}\left(1 - \phi^2\right)$.

2.2.3 The Swift–Hohenberg equation (SH)

The Swift–Hohenberg equation is derived from a phenomenon known as Rayleigh–Bénard convection, in which a fluid is trapped between a hot and a cold plate. The equation describes a convective instability, which occurs when the difference in temperature between the plates becomes high enough. The Swift–Hohenberg equation, a fourth-order, nonlinear partial differential equation, is derived as a gradient flow of the dimensionless free energy functional $\mathcal{F}_{SH}$ defined by [4, 66]

$$\mathcal{F}_{SH} = \int_{\Omega} \left[\frac{1}{2} \left(\phi^2 - 2|\nabla \phi|^2 + (\Delta \phi)^2 \right) + \Psi_{SH}(\phi) \right] d\Omega, \tag{2.26}$$

where $\Psi_{SH}(\phi) = -\dfrac{\epsilon}{2}\phi^2 + \dfrac{1}{4}\phi^4$. The evolution in time of ϕ

$$\frac{\partial \phi}{\partial t} = -\left((1+\Delta)^2 \phi + \Psi'_{SH}(\phi) \right), \tag{2.27}$$

where $M_{SH} = \bar{M}_{SH} = 1$, $(1+\Delta)^2 = 1 + 2\Delta + \Delta\Delta$, Ψ'_{SH} is defined as $\Psi'_{SH}(\phi) = \dfrac{\partial \Psi(\phi)}{\partial \phi} = \phi^3 - \epsilon\phi$, as well as the fact that the order parameter ϕ is not a conserved quantity in the case of this equation such that $a = 0$.

2.2.4 The phase-field crystal equation (PFC)

The phase-field crystal equation is used to model the evolution of microstructures at atomistic length scales and diffusive time scales. It is a sixth-order, nonlinear partial differential equation. It can be derived by considering a conservative description of the Rayleigh Bénard convection problem [30, 14]. It uses the same dimensionless free

energy functional as the Swift–Hohenberg equation, such that

$$\begin{aligned}\mathcal{F}_{PFC} &= \mathcal{F}_{SH} \\ &= \int_{\Omega} \left[\frac{1}{2} \left(\phi^2 - 2|\nabla\phi|^2 + (\Delta\phi)^2 \right) + \Psi_{PFC}(\phi) \right] d\Omega, \end{aligned} \tag{2.28}$$

with $\Psi_{PFC}(\phi) = -\frac{\epsilon}{2}\phi^2 + \frac{1}{4}\phi^4$. The evolution equation of the phase field ϕ, which now represents an atomistic density field that is a conserved quantity, is given by

$$\frac{\partial \phi}{\partial t} = \Delta \left((1+\Delta)^2 \phi + \Psi'_{PFC}(\phi) \right),$$

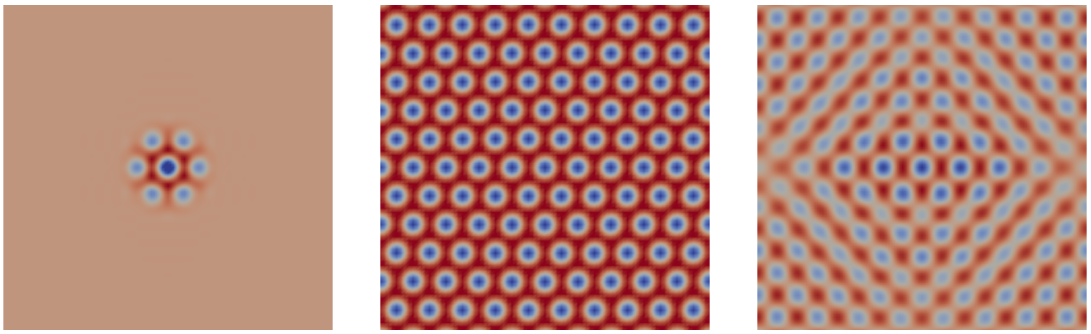
where the mobility $M = 1$.

2.3 Physical consistency: energy stability and mass conservation

The existence of a Lyapunov functional for the diffuse-interface problems presented in section 2.2 implies strong energy stability [44], which can mathematically be expressed through the inequality

$$\frac{d\mathcal{F}}{dt} \leq 0. \tag{2.29}$$

This implies that energy must monotonically decrease in time. Equation (2.29) is the fundamental stability property of the phase-field models considered in this work. This important property can be lost if inadequate algorithms and/or spatial/temporal resolutions are used to solve the partial differential equations [4, 44, 37, 80, 81], which can lead to incorrect numerical results. An example taken from [81] is shown in figure 2.2, where the phase-field crystal equation is used to solve a solidification problem. An initial nucleus is placed at the center of an undercooled liquid (fig-



(a) Initial condition (b) Correct final state (c) Incorrect final state

Figure 2.2: Crystal growth in an undercooled liquid. Snapshots of the atomistic density field of the phase-field crystal equation. An initial nucleus is placed at the center of an undercooled liquid (figure 2.2(a)), which is supposed to replicate its triangular lattice structure over the whole domain. A correct steady state where free energy has decreased in time is shown in figure 2.2(b), whereas an increase in free energy can lead to the incorrect state shown in figure 2.2(c)

ure 2.2(a)), which is supposed to replicate its triangular lattice structure over the whole domain. A correct steady state where free energy has decreased in time is shown in figure 2.2(b), whereas a spike in free energy can lead to the incorrect state shown in figure 2.2(c).

Another condition that needs to be satisfied in the case of conserved phase-field models, such as the CH or the PFC equation, is mass conservation, expressed as

$$\int_{\Omega} \left(\frac{\partial \phi}{\partial t} \right) d\Omega = 0. \tag{2.30}$$

In order for numerical solutions to the partial differential equations to be valid, the discrete versions of equations (2.29)-(2.30) need to be satisfied. The main objective of this work is to develop fully discrete numerical methods which inherit these properties, irrespectively of the mesh and time-step sizes.

This work addresses the nonlinear stability issue for both conserved and non-conserved phase-field variables, and presents processes to handle the nonlinear terms found in the partial differential equation. The equations presented in section 2.2 were

selected because their stable time integration has garnered considerable interest in recent years [2, 4, 5, 78], and this work generalizes some of the numerical schemes that have been put forth in the context of the Cahn-Hilliard [43, 82] and phase-field crystal equations [5].

Chapter 3

An Energy-Stable Method for the Phase-field Crystal Equation

This chapter covers a numerical scheme developed for the phase-field crystal equation, published in [5]. As computational power has increased exponentially through the years, bigger and bigger problems involving microstructure evolution have been solved [83]. However, one of the bigger issues being addressed now is related to the difference in scales: the driving forces in a problem like solidification can be well below the micrometer range. On the one hand, models like the ones solved in molecular dynamics for a many-body problem give an accurate account of what happens, but the domain size that is handled is extremely small [84]. On the other hand, models from continuum theory cannot hope to capture all of the physical effects, as their continuum setting and the assumption of constitutive relationships is trying to model something that is inherently discrete. Within the phase-field modelling setting, the phase-field crystal equation tries to fill this gap by providing a continuum model with an atomic resolution, which is able to evolve in time at diffusive timescales [30].

The phase-field crystal equation is a parabolic, sixth-order and nonlinear partial differential equation, which has generated considerable interest as a possible solution

to problems arising in molecular dynamics. The model was initially developed to study the evolution of microstructures by considering a conservative description of the Rayleigh-Bénard convection problem [14]. The numerical method presented in [5] conserves mass, is energy stable, and is second-order accurate in time. The implementation is done in PetIGA, and the code developed can be found in [85]. The chapter is structured as follows: In section 3.2, we describe the phase field crystal equation. In section 3.3, we present our numerical scheme. Section 3.5 presents numerical examples dealing with crystal growth in a supercooled liquid. We give concluding remarks in section 3.7.

3.1 Introduction

Even though the connection between material processing, structure and properties is known and has been studied for years, a microstructural model taking into account atomic scale features affecting the macroscale properties of a material has not yet to be developed. This chapter tackles one of the recently proposed solution strategies to address this issue, through a model popularly known as the phase-field crystal (PFC) equation. This equation extends the phase-field formalism. Rather than minimizing a free energy through spatially constant values at equilibrium [30, 31], the phase-field variable minimizes the energy functional through periodic states. These periodic minima allow this phase-field model to represent crystalline lattices in two and three dimensions [14, 86]. Remarkably, this equation captures the interaction of atomic-scale defects without the use of additional fields [87]. It features other essential advantages, such as the bridging of time scales [88], possible because the phase-field variable describes a coarse-grained temporal average (the number density of atoms). These benefits justify the consideration of this tool for quantitative modeling [89, 90]: these periodic density states naturally give rise to elasticity as well as to the motion

and nucleation of dislocations, at a fraction of the computational cost of molecular dynamics [14]. Nonetheless, the numerical solution of the PFC equation is far from trivial.

The phase-field crystal equation is a sixth-order, nonlinear, partial differential equation (see equation (2.28)), that must strictly verify the thermodynamic conditions presented in section 2.3. Recent work on this topic includes [2, 3, 42, 44, 46, 91, 79]. Inspired by literature for the Cahn–Hilliard equation in the context of tumor-growth [92], we present in this chapter a numerical formulation that conserves mass, guarantees discrete energy stability and is second-order accurate in time. The time-discrete versions of equations (2.29) and (2.30) are satisfied through

- a convex splitting of Ψ,
- the use of a stabilization term,
- a mixed formulation that segregates the PDE into a system of three, second-order equations.

Although similar to the strategy presented in [3] that also employs a mixed form, the well-posedness of the variational form we present does not require globally $\mathcal{C}^1$-continuous basis functions. The fact that linear, $\mathcal{C}^0$ finite elements can be used is advantageous regarding computational cost [53, 54, 55]. In the following, we detail the derivation of the method, and mathematically prove the physical properties our scheme guarantees. We also present numerical results in two and three spatial dimensions that showcase the robustness of our algorithm.

3.2 Phase-field crystal model

The phase-field crystal equation uses a free energy functional that attains minimal values through periodic density fields [30]. This periodicity allows the model to rep-

resent and capture the evolution of crystalline lattices implicitly [30]. To achieve this, the model uses a conserved phase-field variable related to the atomic number density, which is constant in the liquid phase and spatially periodic in the solid phase. The phase-field crystal equation has been connected to other continuum field theories such as density-functional theory [88, 93]. This work shows examples related to crystalline growth as the PFC equation has found much of its success in modelling microstructural evolution [31, 94, 95, 88, 96], while it has also been used to model other physical phenomena such as foam dynamics [97], glass formation [98], liquid crystals [99], elasticity [30] and in the estimation of material properties [100].

Given some of the coarse approximations used to derive the model [88], it is not surprising that experimental and computational results differ. Nonetheless, work is being done to reduce the mismatch [101, 102, 100, 103]. By increasing the number of critical wavelengths considered in the free energy functional, the accuracy of the simplified model considered in this work [14, 104] can be improved. However, the improved model increases the computational cost, as the partial differential equation becomes harder to solve, as shown in [90, 102]. Also, molecular dynamics in a multi-scale setting can be used to estimate some of the parameters going into the phase-field crystal equation [105], and inverse formulations of the problem could be considered to validate the calculations [106]. Eventually, these multi-scale approaches should allow for more thorough studies on polycrystalline growth using the PFC equation, such as the ones presented in [107, 108] in the setting of phase-field modeling.

3.2.1 Model formulation

The order parameter ϕ present in the free energy functional the phase-field crystal equation represents an atomistic density field, which as previously mentioned is periodic in the solid state and constant in the liquid one. We recall the free energy functional for the phase-field crystal equation in its dimensionless form [31, 3, 86],

given by equation (2.28) is

$$\mathcal{F}[\phi(\mathbf{x})] = \int_{\Omega} \left[\Psi(\phi) + \frac{1}{2} \left(\phi^2 - 2|\nabla\phi|^2 + (\Delta\phi)^2 \right) \right] d\Omega, \tag{3.1}$$

where $\Omega \in \mathbb{R}^d$ represents an arbitrary open domain, with $d = 2$ or 3, and $\Psi(\phi) = -\frac{\epsilon}{2}\phi^2 + \frac{1}{4}\phi^4$. The parameter ϵ is related to the degree of undercooling. To minimize the free energy functional, we solve the Euler–Lagrange equation of equation (3.1), and take its variational derivative with respect to ϕ. Recalling equation (2.16), the variational derivative is given by

$$\frac{\delta\mathcal{F}}{\delta\phi} = (1+\Delta)^2\phi + \Psi'(\phi), \tag{3.2}$$

where $\Psi'(\phi) = -\epsilon\phi + \phi^3$ with $(1+\Delta)^2 = 1 + 2\Delta + \Delta\Delta$. Considering mass conservation [31] and equation (2.19), the partial differential equation is formulated as

$$\frac{\partial\phi}{\partial t} = \nabla \cdot \left(M\nabla\frac{\delta\mathcal{F}}{\delta\phi} \right), \tag{3.3}$$

where $\phi \equiv \phi(\mathbf{x}, t)$ represents the phase field, $\mathbf{x}$ and t represent space and time, respectively, M is the mobility, and $\mathcal{F}$ is the free energy functional of the system. The partial differential equation, after substituting equation (3.2) into (3.3), becomes

$$\begin{aligned} \frac{\partial\phi}{\partial t} &= \nabla \cdot \nabla \left[(1+\Delta)^2 \phi + \Psi'(\phi) \right] \\ &= \Delta \left[(1+\Delta)^2 \phi + \Psi'(\phi) \right], \end{aligned}$$

where the mobility M is assumed equal to a constant of value one.

3.2.2 Strong form

The initial value problem is stated as follows: over the spatial domain Ω and the time interval $]0,T[$, given $\phi_0 : \Omega \longmapsto \mathbb{R}$, find $\phi : \Omega \times [0,T] \longmapsto \mathbb{R}$ such that

$$\begin{cases} \dfrac{\partial \phi}{\partial t} = \Delta \left[(1+\Delta)^2 \phi + \Psi'(\phi) \right] & \text{on } \Omega \times]0,T], \\ \phi(\mathbf{x},0) = \phi_0(\mathbf{x}) & \text{on } \Omega, \end{cases} \tag{3.4}$$

where the function $\phi_0(\mathbf{x})$ approximates a crystalline nucleus, and periodic boundary conditions are considered in all directions. We discuss the handling of initial conditions further in section 3.5.

3.3 Stable time discretization for the phase-field crystal equation

Current research on energy-stable, time integration schemes focuses on implicit [3, 42, 44, 91] and adaptive algorithms [40]. Explicit methods are not an obvious choice, given their numerical stability issues. They face severe time step size restrictions when solving a high-order partial differential equation such as the PFC equation (i.e., the time step size is on the order of the sixth power of the grid size). On top of this, the discrete version of equation (2.29) defined as

$$\mathcal{F}\left[\phi\left(t_{n+1}\right)\right] \leq \mathcal{F}\left[\phi\left(t_n\right)\right] \quad \forall n = 1, 2, ..., N, \tag{3.5}$$

related to energy stability, as well as the discrete version of (2.30), defined as

$$\int_\Omega \left(\frac{\partial \phi}{\partial t} \right) d\Omega = 0 \tag{3.6}$$

involving mass conservation, need to be satisfied for numerical solutions to the equation to be valid. In the following, we detail an algorithm that extends ideas presented in [29, 91], guarantees the properties presented in equations (3.5) and (3.6), and achieves second-order accuracy in time. As stated in chapter 1, the spatial discretization is done using isogeometric analysis (IGA), a finite element method based on NURBS as basis functions [48]. IGA provides a framework that improves upon standard finite element analysis [47]. The method can better represent some complex geometrical shapes, and provides systematic refinement procedures. IGA allows us to control the spatial resolution of the mesh (h-refinement) and the polynomial degree of the basis (p-refinement), and provides a refinement procedure (k-refinement) that can increase the smoothness of element functions beyond standard $\mathcal{C}^0$ continuity [109]. IGA has successfully been applied to multiple phase-field models [3, 50, 52, 110, 111, 112]. Given the high-order nature of the equations involved, many choices are possible regarding discretizations and time stepping schemes [2, 28]. As high-order, globally continuous basis functions can be easily generated within the IGA framework, it is possible to discretize straightforwardly high-order partial differential equations. Primal formulations of the equations, which cannot always be solved using standard $\mathcal{C}^0$-linear finite elements, can be handled using this technology [28, 52]. Alternatively, mixed formulations can be employed to reduce continuity requirements down to standard $\mathcal{C}^0$ spaces used in traditional finite element methods. This work uses a mixed form, where the system solved involves a coupled system of three second-order equations.

3.3.1 Mixed form $2+2+2$: triple second-order split

Equation (3.4) can be written as a system that consists of three coupled second-order equations, given by

$$\frac{\partial \phi}{\partial t} = \Delta \sigma \qquad \text{in } \Omega \times]0,T], \tag{3.7a}$$

$$\sigma = (1+\Delta)\,\theta + \Psi'(\phi) \qquad \text{in } \Omega \times]0,T], \tag{3.7b}$$

$$\theta = (1+\Delta)\,\phi \qquad \text{in } \Omega \times]0,T]. \tag{3.7c}$$

3.3.2 Weak form

Let us denote by $\mathcal{V}_1$ a functional space, which is a subset of $\mathcal{H}^1$, where $\mathcal{H}^1$ is the Sobolev space of square integrable functions with square integrable first derivatives. Assuming periodic boundary conditions in all directions, a weak form can be derived by multiplying (3.7a) to (3.7c) by test functions $q, s, w \in \mathcal{V}_1$, respectively, and integrating the equations by parts. The variational problem can then be defined as that of finding $\phi, \theta, \sigma \in \mathcal{V}_1$ such that for all $q, s, w \in \mathcal{V}_1$

$$\begin{aligned} 0 = &\left(q, \dot{\phi}\right)_\Omega + (\nabla q, \nabla \sigma)_\Omega \\ &+ (s, \sigma - \Psi'(\phi) - \theta)_\Omega + (\nabla s, \nabla \theta)_\Omega \\ &+ (w, \theta - \phi)_\Omega + (\nabla w, \nabla \phi)_\Omega \,, \end{aligned}$$

where the dependence of ϕ on space and time is not explicitly stated, the $\mathcal{L}^2$ inner product over the domain Ω is indicated by $(.,.)_\Omega$ and $\dot{\phi} := \dfrac{\partial \phi}{\partial t}$.

3.3.3 Semi-discrete formulation

Splitting the equation with the help of the auxiliary variables σ and θ allows us to use C^0 finite elements, as only $\mathcal{H}^1$-conforming spaces are needed. We let $\mathcal{V}_1^h \subset \mathcal{V}_1$

denote the finite dimensional functional space spanned by these C^0 B-spline basis functions in two or three spatial dimensions. The problem is then stated as follows: find $\phi^h, \theta^h, \sigma^h \in \mathcal{V}_1^h$ such that for all $q^h, s^h, w^h \in \mathcal{V}_1^h$

$$\begin{aligned} 0 = & \left(q^h, \dot{\phi^h}\right)_\Omega + (\nabla q^h, \nabla \sigma^h)_\Omega \\ & + \left(s^h, \sigma^h - \Psi'\left(\phi^h\right) - \theta^h\right)_\Omega + \left(\nabla s^h, \nabla \theta^h\right)_\Omega \\ & + \left(w^h, \theta^h - \phi^h\right)_\Omega + \left(\nabla w^h, \nabla \phi^h\right)_\Omega, \end{aligned} \tag{3.8}$$

where the weighting functions q^h, s^h and w^h, and trial solutions σ^h, θ^h and ϕ^h can be defined as

$$\begin{aligned} q^h &= \sum_{A=1}^{n_b} q_A N_A, \quad s^h = \sum_{A=1}^{n_b} s_A N_A, \quad w^h = \sum_{A=1}^{n_b} w_A N_A, \\ \sigma^h &= \sum_{A=1}^{n_b} \sigma_A N_A, \quad \theta^h = \sum_{A=1}^{n_b} \theta_A N_A, \quad \phi^h = \sum_{A=1}^{n_b} \phi_A N_A, \end{aligned}$$

where the B-spline basis functions N_A define the discrete space $\mathcal{V}_1^h$ of dimension n_b and the coefficients q_A, s_A, w_A, σ_A, θ_A and ϕ_A represent the control variables.

3.3.4 Time discretization

The time discretization proposed in this Chapter adapts what was done in [29] for the Cahn–Hilliard equation, to the formulation presented in equation (3.8) for the phase-field-crystal equation. To do this, the nonlinear term $\Psi(\phi) = \dfrac{\phi^4}{4} - \dfrac{\epsilon\phi^2}{2}$ is split as

$$\Psi(\phi) = \Psi_c(\phi) - \Psi_e(\phi),$$

where $\Psi_c(\phi) = \dfrac{\phi^4}{4}$ and $\Psi_e(\phi) = \dfrac{\epsilon\phi^2}{2}$. Both of these functions are convex, which allows us to discretize the nonlinearity in time using a convex-implicit, concave-explicit

treatment, giving the following fully discrete system

$$
\begin{aligned}
0 = & \left(q^h, \frac{[\![\phi_n^h]\!]}{\Delta t}\right)_\Omega + \left(\nabla q^h, \nabla \sigma^h\right)_\Omega \\
& + \left(s^h, \sigma^h - \theta^h - \left(\Psi_c'\left(\phi_{n+1}^h\right) - \Psi_c''\left(\phi_{n+1}^h\right)\frac{[\![\phi_n^h]\!]}{2}\right)\right)_\Omega \\
& + \left(s^h, \left(\Psi_e'\left(\phi_n^h\right) + \Psi_e''\left(\phi_n^h\right)\frac{[\![\phi_n^h]\!]}{2}\right)\right)_\Omega \\
& + \left(\nabla s^h, \nabla \theta^h - \alpha_n \Delta t \nabla [\![\phi_n^h]\!]\right)_\Omega \\
& + \left(w^h, \theta^h - \{\phi_n^h\}\right)_\Omega + \left(\nabla w^h, \nabla \{\phi_n^h\}\right)_\Omega, \qquad (3.9)
\end{aligned}
$$

where

- $[\![\phi_n^h]\!] = \phi_{n+1}^h - \phi_n^h$,
- $\{\phi_n^h\} = \frac{1}{2}\left(\phi_{n+1}^h + \phi_n^h\right)$,
- $\Psi_c'(\phi_{n+1}^h) = \left(\phi_{n+1}^h\right)^3$,
- $\Psi_c''(\phi_{n+1}^h) = 3\left(\phi_{n+1}^h\right)^2$,
- $\Psi_e'(\phi_n^h) = \epsilon\phi_n^h$,
- $\Psi_e''(\phi_n^h) = \epsilon$,

and the stabilization parameter α_n needs to comply with

$$
\alpha_n \geq \frac{\left(\sup\left(\Psi_c''\left(\phi_{n+1}^h\right) + \Psi_e''\left(\phi_n^h\right)\right)\right)^2}{16} = \frac{\left(\sup\left(3\left(\phi_{n+1}^h\right)^2 + \epsilon\right)\right)^2}{16}
$$

3.4 Properties of the numerical scheme

The discretization presented in section 3.3.4 guarantees mass conservation, is second-order accurate in time, and possesses energy stability by construction.

3.4.1 Mass conservation

Mass conservation can be verified by taking equation (3.9), and letting the test function q^h be equal to one while having $s^h = w^h = 0$, such that

$$0 = \left(1, \frac{[\![\phi_n^h]\!]}{\Delta t}\right)_\Omega + \left(0, \nabla\sigma^h\right) = \int_\Omega \frac{[\![\phi_n^h]\!]}{\Delta t} d\Omega,$$

which implies that mass is conserved at the discrete time levels, that is

$$\int_\Omega \phi_{n+1} d\Omega = \int_\Omega \phi_n d\Omega.$$

3.4.2 Second-order accuracy in time

A bound on the local truncation error can be obtained by comparing our method to the Crank-Nicolson scheme, a well-known second-order accurate time-stepping algorithm. If we do not spatially discretize (3.4), but instead apply the Crank-Nicolson scheme to it, we obtain

$$\frac{[\![\phi_n]\!]}{\Delta t} = \Delta\left[(1+\Delta)^2\{\phi_n\} + \Psi'(\{\phi_n\})\right].$$

Substituting the discrete time solution $\{\phi_n\}$ by the time-continuous solution $\{\phi(t_n)\}$ into the above equation gives rise to the local truncation error. Indeed, we have

$$\frac{[\![\phi(t_n)]\!]}{\Delta t} = \Delta\left[(1+\Delta)^2\{\phi(t_n)\} + \Psi'(\{\phi(t_n)\})\right] + \tau(t_n), \tag{3.10}$$

where $\tau(t_n)$ represents the global truncation error. It can be shown, using Taylor series, that such a scheme will give a bound $\tau(t_n) \leq C\Delta t^2$, as was done in [2] in a similar context.

To prove second-order accuracy in time for our scheme, we compute the next

time-step approximation via the scheme applied to the exact solution and compare the result to Taylor expansions. A similar procedure was performed in [29] in the context of Cahn–Hilliard equations. By looking at only the time discretization part of (3.9), and reorganizing the splitting into one equation, we have that

$$\begin{aligned}\phi_{n+1} = \phi(t_n) + \Delta t \Delta \Bigg(& (1+\Delta)^2 \phi(\{t\}) \\ & + \Psi_c'(\phi(t_{n+1})) - \Psi_e'(\phi(t_n)) \\ & - \frac{1}{2}[\![\phi(t)]\!]\Psi_c''(\phi(t_{n+1})) - \frac{1}{2}[\![\phi(t)]\!]\Psi_e''(\phi(t_n)) \\ & - \alpha_n \Delta t \Delta [\![\phi(t)]\!] \Bigg), \end{aligned} \tag{3.11}$$

where $\phi(\{t\})$ is defined as the Crank-Nicolson (mid-point rule) approximation

$$\phi(\{t\}) = \phi\left(\frac{t_{n+1}+t_n}{2}\right) = \frac{\phi(t_{n+1})+\phi(t_n)}{2} + \mathcal{O}(\Delta t^2).$$

We expand $\Psi_c'\left(\phi\left(t_{n+1}\right)\right)$ such that

$$\begin{aligned}\Psi_c'(\phi(t_{n+1})) &= \Psi_c'(\phi(\{t\})) - \Psi_c''(\phi(t_{n+1}))\big(\phi(\{t\}) - \phi(t_{n+1})\big) + \mathcal{O}(\Delta t^2) \\ &= \Psi_c'(\phi(\{t\})) + \frac{[\![\phi(t)]\!]}{2}\Psi_c''(\phi(t_{n+1})) + \mathcal{O}(\Delta t^2).\end{aligned}$$

Thus,

$$\Psi_c'(\{\phi\}) = \Psi_c'(\phi(t_{n+1})) - \frac{[\![\phi(t)]\!]}{2}\Psi_c''(\phi(t_{n+1})) + \mathcal{O}(\Delta t^2). \tag{3.12}$$

Similarly, we have for the explicit part

$$\Psi_e'(\{\phi\}) = \Psi_e'(\phi(t_n)) + \frac{[\![\phi(t)]\!]}{2}\Psi_e''(\phi(t_n)) + \mathcal{O}(\Delta t^2). \tag{3.13}$$

The stabilization term is of order $\mathcal{O}(\Delta t^2)$ and can be written as

$$\begin{aligned}\alpha_n \Delta t \Delta [\![\phi(t)]\!] &= \alpha_n (\Delta t)^2 \Delta \left(\frac{[\![\phi(t)]\!]}{\Delta t} \right) \\ &= \alpha_n (\Delta t)^2 \Delta \left(\frac{\partial \phi}{\partial t} + \mathcal{O}(\Delta t) \right) = \mathcal{O}(\Delta t^2). \end{aligned} \tag{3.14}$$

Using (3.12)-(3.14), and substituting them into (3.11), we obtain

$$\begin{aligned}\phi_{n+1} = \phi(t_n) + \Delta t \Delta \Bigg(& (1+\Delta)^2 \phi(\{t\}) \\ & + \Psi_c'(\{\phi\}) - \Psi_e'(\{\phi\}) + \mathcal{O}(\Delta t^2) \Bigg). \end{aligned} \tag{3.15}$$

Alternatively, by Taylor expansion of the solution, we have

$$\begin{aligned}\phi(\{t\}) &= \phi(t_{n+1}) - \frac{\Delta t}{2} \phi'(\{t\}) - \frac{1}{2} \left(\frac{\Delta t}{2} \right) \phi''(\{t\}) + \mathcal{O}(\Delta t^3), \\ \phi(\{t\}) &= \phi(t_n) + \frac{\Delta t}{2} \phi'(\{t\}) - \frac{1}{2} \left(\frac{\Delta t}{2} \right) \phi''(\{t\}) + \mathcal{O}(\Delta t^3). \end{aligned}$$

Taking the difference of the above two equations and using (3.4) yields

$$\begin{aligned}\phi(t_{n+1}) - \phi(t_n) &= \Delta t \frac{\partial \phi(\{t\})}{\partial t} + \mathcal{O}(\Delta t^3) \\ &= \Delta t \Delta \left((1+\Delta)^2 \phi(\{t\}) + \Psi_c'(\{\phi\}) - \Psi_e'(\{\phi\}) + \mathcal{O}(\Delta t^3) \right). \end{aligned}$$

Finally, taking the difference of the above expression with (3.15), we obtain the local truncation error

$$\phi(t_{n+1}) - \phi_{n+1} = \mathcal{O}(\Delta t^3).$$

Thus, using the fact that the global truncation error $\tau(t_n)$ loses an order of Δt, the

scheme is second-order accurate in time.

3.4.3 Energy stability

To prove energy stability, we first consider the time-discrete form of the scheme, given by

$$\llbracket \phi_n \rrbracket = \Delta t \Delta \tilde{\sigma}, \tag{3.16}$$

$$\begin{aligned}\tilde{\sigma} &= (1+\Delta)\,\tilde{\theta} - \alpha_n \Delta t \Delta \llbracket \phi_n \rrbracket \\ &\quad + \Psi_c'(\phi_{n+1}) - \frac{1}{2}\llbracket \phi_n \rrbracket \Psi_c''(\phi_{n+1}) - \Psi_e'(\phi_n) - \frac{1}{2}\llbracket \phi_n \rrbracket \Psi_e''(\phi_n),\end{aligned} \tag{3.17}$$

$$\tilde{\theta} = (1+\Delta)\{\phi_n\}. \tag{3.18}$$

Considering that for any smooth function Ψ we have

$$\begin{aligned}\llbracket \Psi \rrbracket &= \Psi'(\phi_n)\llbracket \phi_n \rrbracket + \Psi''(\xi_1(\phi_{n+1},\phi_n))\frac{\llbracket \phi_n \rrbracket^2}{2} \\ &= \Psi'(\phi_{n+1})\llbracket \phi_n \rrbracket - \Psi''(\xi_2(\phi_{n+1},\phi_n))\frac{\llbracket \phi_n \rrbracket^2}{2},\end{aligned}$$

for some $\xi_1(\phi_{n+1},\phi_n)$ in between ϕ_n and ϕ_{n+1}, similarly for $\xi_2(\phi_{n+1},\phi_n)$. The above formula is the exact Taylor series with remainder term and no additional terms are required in the expansion.

Applying these expansions to our particular form of the nonlinearity, by Taylor's theorem, for some ξ_c, ξ_e between ϕ_n and ϕ_{n+1}, we have that

$$\begin{aligned}\llbracket \Psi \rrbracket &= \llbracket \Psi_c \rrbracket - \llbracket \Psi_e \rrbracket \\ &= \Psi_c'(\phi_{n+1})\llbracket \phi_n \rrbracket - \Psi_c''(\xi_c)\frac{\llbracket \phi_n \rrbracket^2}{2} - \Psi_e'(\phi_n)\llbracket \phi_n \rrbracket - \Psi_e''(\xi_e)\frac{\llbracket \phi_n \rrbracket^2}{2}.\end{aligned}$$

Since Ψ_c and Ψ_e are globally convex, we have that $\Psi_c'', \Psi_e'' \geq 0$. Observe that

$$\begin{aligned}\llbracket \Psi \rrbracket &= \left(\Psi_c'(\phi_{n+1}) - \Psi_e'(\phi_n)\right) \llbracket \phi_n \rrbracket - \left(\Psi_c''(\xi_c) + \Psi_e''(\xi_e)\right) \frac{\llbracket \phi \rrbracket^2}{2} \\ &\leq \left(\Psi_c'(\phi_{n+1}) - \Psi_e'(\phi_n)\right) \llbracket \phi_n \rrbracket. \end{aligned} \tag{3.19}$$

Here, we use the fact that the second derivatives are non-negative and the overall sign of the second derivative terms is negative. Recalling equation (3.1), we have that the free energy is given by

$$\mathcal{F}[\phi(\mathbf{x})] = \int_\Omega \left[\Psi(\phi) + \frac{1}{2}\left(\phi^2 - 2|\nabla\phi|^2 + (\Delta\phi)^2\right)\right] d\Omega,$$

with which we can write, given equation (3.19), that

$$\begin{aligned}\llbracket \mathcal{F}[\phi(\mathbf{x})] \rrbracket &= \llbracket \mathcal{F} \rrbracket \\ &= \int_\Omega \left(\llbracket \Psi\left(\phi\right) \rrbracket + \frac{1}{2} \llbracket \left(\phi\right)^2 - 2|\nabla\phi|^2 + (\Delta\phi)^2 \rrbracket \right) d\Omega \\ &\leq \int_\Omega \left(\Psi_c'\left(\phi_{n+1}\right) - \Psi_e'\left(\phi_n\right)\right) \llbracket \phi \rrbracket d\Omega \\ &+ \frac{1}{2} \int_\Omega \llbracket \left(\left(\phi\right)^2 - 2|\nabla\phi|^2 + (\Delta\phi)^2 \right) \rrbracket d\Omega,\end{aligned}$$

given that Ψ_c and Ψ_e are convex. From equation (3.17), we have that

$$\begin{aligned}\Psi_c'\left(\phi_{n+1}\right) - \Psi_e'\left(\phi_n\right) &= \tilde{\sigma} - \left(1 + \Delta\right)\tilde{\theta} \\ &+ \frac{1}{2}\llbracket \phi_n \rrbracket \Psi_c''\left(\phi_{n+1}\right) + \frac{1}{2}\llbracket \phi_n \rrbracket \Psi_e''\left(\phi_n\right) + \alpha_n \Delta t \Delta \llbracket \phi_n \rrbracket. \end{aligned} \tag{3.20}$$

We now simplify the notation for the explicit-implicit treatment of the second deriva-

tive and write

$$\begin{aligned}\Psi''_{n,n+1} &= \Psi''_c(\phi_{n+1}) + \Psi''_e(\phi_n) \\ &= 3\phi_{n+1}^2 + \epsilon. \end{aligned} \tag{3.21}$$

We then multiply equation (3.20) by $[\![\phi_n]\!] = \phi_{n+1} - \phi_n$, use equation (3.18), and integrate over the domain, to obtain

$$\begin{aligned}\int_\Omega \left(\Psi'_c(\phi_{n+1}) - \Psi'_e(\phi_n)\right)[\![\phi_n]\!]d\Omega = &\int_\Omega \left(\tilde{\sigma}[\![\phi_n]\!] - (1+\Delta)^2\{\phi_n\}[\![\phi_n]\!]\right)d\Omega \\ &+ \int_\Omega \left(\frac{1}{2}[\![\phi_n]\!]\left(\Psi''_{n,n+1}\right)[\![\phi_n]\!]\right)d\Omega \\ &+ \int_\Omega \left(\alpha_n \Delta t \Delta[\![\phi_n]\!][\![\phi_n]\!]\right)d\Omega. \end{aligned} \tag{3.22}$$

We now proceed to expand the different terms on the right-hand side of equation (3.22). Integrating the first term by parts, we obtain

$$\begin{aligned}\int_\Omega [\![\phi_n]\!]\tilde{\sigma}d\Omega &= \int_\Omega \Delta t\left(\Delta\tilde{\sigma}\right)\tilde{\sigma}d\Omega \\ &= -\int_\Omega \Delta t|\nabla\tilde{\sigma}|^2 d\Omega. \end{aligned} \tag{3.23}$$

Then, for the second term in equation (3.22), we have

$$\begin{aligned}\int_\Omega (1+\Delta)^2\{\phi_n\}[\![\phi_n]\!]d\Omega &= \int_\Omega (1+\Delta)\{\phi_n\}(1+\Delta)[\![\phi_n]\!]d\Omega \\ &= \frac{1}{2}\int_\Omega [\![(\phi_n)^2 - 2|\nabla\phi_n|^2 + (\Delta\phi_n)^2]\!]d\Omega. \end{aligned} \tag{3.24}$$

Using $[\![\phi_n]\!] = \Delta t\Delta\tilde{\sigma}$, taking the supremum, and integrating by parts the third term

of equation (3.22), we have

$$\begin{aligned}\int_\Omega \frac{1}{2}[\![\phi_n]\!]^2\Big(\Psi''_{n,n+1}\Big)d\Omega &\leq \sup\Big(\Psi''_{n,n+1}\Big)\int_\Omega \frac{\Delta t}{2}[\![\phi_n]\!]^2 d\Omega \\ &= \sup\Big(\Psi''_{n,n+1}\Big)\int_\Omega \frac{\Delta t}{2}[\![\phi_n]\!]\Delta\sigma d\Omega \\ &= -\frac{\Delta t}{2}\sup\Big(\Psi''_{n,n+1}\Big)\int_\Omega \nabla[\![\phi_n]\!]\nabla\sigma d\Omega. \end{aligned} \tag{3.25}$$

Remark 1. *Recalling equation* (3.21), $\Psi''_{n,n+1} = 3\phi^2_{n+1} + \epsilon$ *and the term is always positive. Thus, we may pull out a supremum without an absolute value needed. Moreover, since we assume the existence of a solution at each time step in* $H^3(\Omega)$*, such a supremum exists.*

Integrating the last term of equation (3.22) by parts results in

$$\int_\Omega \alpha_n \Delta t \Delta[\![\phi_n]\!][\![\phi_n]\!]d\Omega = -\int_\Omega \alpha_n \Delta t \left|\nabla[\![\phi_n]\!]\right|^2 d\Omega. \tag{3.26}$$

Finally, by collecting the terms in equations (3.23)-(3.26), and replacing them in equation (3.22), we obtain

$$\begin{aligned}[\![\mathcal{F}]\!] \leq \int_\Omega \Big(&- \Delta t|\nabla\tilde{\sigma}|^2 - \alpha_n\Delta t\left(\nabla[\![\phi_n]\!]\right)^2 \\ &- \frac{\Delta t}{2}\sup\Big(\Psi''_{n,n+1}\Big)\nabla[\![\phi_n]\!]\nabla\tilde{\sigma}\Big)d\Omega. \end{aligned} \tag{3.27}$$

Using Young's inequality, $2fg \leq \beta f^2 + \beta^{-1}g^2$, with $f = -\nabla[\![\phi_n]\!]$ and $g = \nabla\tilde{\sigma}$, we then have that

$$\frac{\Delta t}{2}\sup\Big(\Psi''_{n,n+1}\Big)(-\nabla[\![\phi_n]\!])\nabla\tilde{\sigma} \leq \frac{\Delta t}{4}\sup\Big(\Psi''_{n,n+1}(\phi)\Big)\left(\beta\left(\nabla[\![\phi_n]\!]\right)^2 + \frac{|\nabla\tilde{\sigma}|^2}{\beta}\right),$$

such that inequality (3.27) becomes

$$[\![\mathcal{F}]\!] \leq \int_{\Omega} \Bigg(-\Delta t|\nabla\tilde{\sigma}|^2 - \alpha_n \Delta t \left(\nabla[\![\phi_n]\!]\right)^2 + \frac{\Delta t}{4\beta}\sup\left(\Psi''_{n,n+1}\left(\phi\right)\right)|\nabla\tilde{\sigma}|^2 + \frac{\Delta t}{4}\sup\left(\Psi''_{n,n+1}\left(\phi\right)\right)\beta(\nabla[\![\phi_n]\!])^2\Bigg) d\Omega, \tag{3.28}$$

which is verified as long as

$$\beta \geq \frac{\sup\left(\Psi''_{n,n+1}\right)}{4} \quad \text{and} \quad \alpha_n \geq \frac{\left(\sup\left(\Psi''_{n,n+1}\right)\right)^2}{16}. \tag{3.29}$$

Equation (3.28) and the fulfilment of the conditions in (3.29), guarantee free energy stability.

Remark 2. *The above condition is effectively nonlinear, since the choice of α_n depends on ϕ_{n+1}. However, since the smoothness at each time step is assumed to be $\mathcal{H}^3$, it is continuous and a global supremum of $\Psi''_{n,n+1}=3\phi^2_{n+1}+\epsilon$ exists at each time step. The supremum of such a quantity is however a priori unknown. Thus, in our implementation the above stability condition is a lagging condition where α_n is computed using the current time step. Another approach involves truncating the second derivative of Ψ outside the regions $[-1,1]$, and interpolating with polynomials as in [92] in the context of Cahn–Hilliard to obtain a global bound, such that α_n can be evaluated independently from Δt.*

3.4.4 Alternative formulation

This stabilization procedure is also suitable for the following alternative formulation

$$\frac{\partial\phi}{\partial t} = \Delta\sigma \qquad \text{in } \Omega\times]0,T], \tag{3.30a}$$

$$\sigma = \left(1+\Delta\right)^2\phi + \Psi'(\phi) \qquad \text{in } \Omega\times]0,T]. \tag{3.30b}$$

Let us denote by $\mathcal{V}_2$ a functional space belonging to $\mathcal{H}^2$, where $\mathcal{H}^2$ is the Sobolev space of square-integrable functions with square-integrable first and second derivatives. Assuming periodic boundary conditions in all directions, a weak form can be derived multiplying (3.30a)-(3.30b) by test functions q, $w \in \mathcal{V}_2$, respectively, and integrating the equations by parts. The problem can then be defined as that of finding $\phi, \sigma \in \mathcal{V}_2$ such that for all $q, w \in \mathcal{V}_2$

$$\begin{aligned} 0 = &\left(q, \dot{\phi}\right)_\Omega + (\nabla q, \nabla \sigma)_\Omega \\ &+ (w, \sigma - \Psi'(\phi) - \phi)_\Omega + 2(\nabla w, \nabla \phi)_\Omega - (\Delta w, \Delta \phi)_\Omega . \end{aligned} \tag{3.31}$$

This formulation requires the use of at least $\mathcal{C}^1$ continuity, but the use of a convex-implicit and concave-explicit discretization of the nonlinearity can also be done, such that the fully discrete formulation becomes

$$\begin{aligned} 0 = &\left(q^h, \frac{[\![\phi_n^h]\!]}{\Delta t}\right)_\Omega + \left(\nabla q^h, \nabla \sigma^h\right)_\Omega + \left(w^h, \sigma^h - \{\phi_n^h\}\right)_\Omega \\ &- \left(w^h, \Psi_c'\left(\phi_{n+1}^h\right) - \Psi_c''\left(\phi_{n+1}^h\right)\frac{[\![\phi_n^h]\!]}{2}\right)_\Omega \\ &+ \left(w^h, \Psi_e'\left(\phi_n^h\right) + \Psi_e''\left(\phi_n^h\right)\frac{[\![\phi_n^h]\!]}{2}\right)_\Omega \\ &+ \left(\nabla w^h, 2\nabla\{\phi_n^h\} - \alpha_n \Delta t \nabla [\![\phi_n^h]\!]\right)_\Omega - \left(\Delta w^h, \Delta\{\phi_n^h\}\right)_\Omega , \end{aligned}$$

where

- $\alpha_n \geq \dfrac{\left[\sup\left(\Psi''_{n,n+1}\right)\right]^2}{16}$,

- $\Psi''_{n,n+1} = 3\phi_{n+1}^2 + \epsilon$.

3.4.5 Numerical implementation

With regards to the implementation, we let the global vectors of degrees of freedom associated to ϕ_n^h, σ_n^h and θ_n^h be $\mathbf{\Phi}_n$, $\mathbf{\Sigma}_n$ and $\mathbf{\Theta}_n$, respectively. The residual vectors for this formulation are then given by

$$\begin{aligned}
&\mathbf{R}_\phi(\mathbf{\Phi}_n, \mathbf{\Phi}_{n+1}, \mathbf{\Sigma}_{n+1}, \mathbf{\Theta}_{n+1}); \quad \mathbf{R}_\phi = \{R_A^\phi\}; \quad A = 1, ..., n_b, \\
&\mathbf{R}_\sigma(\mathbf{\Phi}_n, \mathbf{\Phi}_{n+1}, \mathbf{\Sigma}_{n+1}, \mathbf{\Theta}_{n+1}); \quad \mathbf{R}_\sigma = \{R_A^\sigma\}; \quad A = 1, ..., n_b, \\
&\mathbf{R}^\theta(\mathbf{\Phi}_n, \mathbf{\Phi}_{n+1}, \mathbf{\Sigma}_{n+1}, \mathbf{\Theta}_{n+1}); \quad \mathbf{R}_\theta = \{R_A^\theta\}; \quad A = 1, ..., n_b,
\end{aligned}$$

where

$$\begin{aligned}
R_A^\phi &= \left(N_A, \frac{[\![\phi_n^h]\!]}{\Delta t}\right) + \left(\nabla N_A, \nabla \sigma^h\right), \\
R_A^\sigma &= \left(N_A, \sigma^h - \theta^h - \left(\Psi_c'\left(\phi_{n+1}^h\right) - \Psi_c''\left(\phi_{n+1}^h\right)\frac{[\![\phi_n^h]\!]}{2}\right)\right) \\
&\quad + \left(N_A, \left(\Psi_e'\left(\phi_n^h\right) + \Psi_e''\left(\phi_n^h\right)\frac{[\![\phi_n^h]\!]}{2}\right)\right) \\
&\quad + \left(\nabla N_A, \nabla\theta^h - \alpha_n \Delta t \nabla [\![\phi_n^h]\!]\right), \\
R_A^\theta &= \left(N_A, \theta^h - \{\phi_n^h\}\right) + \left(\nabla N_A, \nabla\{\phi_n^h\}\right).
\end{aligned}$$

The resulting system of nonlinear equations for $\mathbf{\Phi}_{n+1}$, $\mathbf{\Sigma}_{n+1}$ and $\mathbf{\Theta}_{n+1}$ is solved using Newton's method, where $\mathbf{\Phi}_{n+1}^{(i)}$, $\mathbf{\Sigma}_{n+1}^{(i)}$ and $\mathbf{\Theta}_{n+1}^{(i)}$ correspond to the i-th iteration of Newton's algorithm. The iterative procedure is specified in Algorithm 2.

3.5 Numerical results

The implementation of the numerical scheme described in section 3.3 was done using PetIGA [113, 56, 57], which is a software framework built on top of PETSc [114, 115], that delivers a high-performance computational framework for IGA. Tutorials for the

Algorithm 1 Iterative procedure to solve the $2+2+2$ mixed form

Taking $\boldsymbol{\Phi}_{n+1}^{(0)} = \boldsymbol{\Phi}_n$, $\boldsymbol{\Sigma}_{n+1}^{(0)} = \boldsymbol{\Sigma}_n$, and $\boldsymbol{\Theta}_{n+1}^{(0)} = \boldsymbol{\Theta}_n$, for $i = 1, .., \mathrm{i}_{max}$,

(1) Compute the residuals $\mathbf{R}_\phi^{(i)}$, $\mathbf{R}_\sigma^{(i)}$, $\mathbf{R}_\theta^{(i)}$, using $\boldsymbol{\Phi}_{n+1}^{(i)}$, $\boldsymbol{\Sigma}_{n+1}^{(i)}$, $\boldsymbol{\Theta}_{n+1}^{(i)}$.
(2) Compute the Jacobian matrix $\mathbf{K}^{(i)}$ using the *i-th* iterates. This matrix is given by

$$\mathbf{K}^{(i)} = \begin{pmatrix} \mathbf{K}^{\phi\phi} & \mathbf{K}^{\phi\sigma} & \mathbf{K}^{\phi\theta} \\ \mathbf{K}^{\sigma\phi} & \mathbf{K}^{\sigma\sigma} & \mathbf{K}^{\sigma\theta} \\ \mathbf{K}^{\theta\phi} & \mathbf{K}^{\theta\sigma} & \mathbf{K}^{\theta\theta} \end{pmatrix}^{(i)}, \tag{3.32}$$

where the individual components of each submatrix of the Jacobian are defined in Appendix B.1 in equations (C.1) through (C.4).
(3) Solve the linear system

$$\begin{pmatrix} \mathbf{K}^{\phi\phi} & \mathbf{K}^{\phi\sigma} & \mathbf{K}^{\phi\theta} \\ \mathbf{K}^{\sigma\phi} & \mathbf{K}^{\sigma\sigma} & \mathbf{K}^{\sigma\theta} \\ \mathbf{K}^{\theta\phi} & \mathbf{K}^{\theta\sigma} & \mathbf{K}^{\theta\theta} \end{pmatrix}^{(i)} \begin{pmatrix} \Delta\boldsymbol{\Phi} \\ \Delta\boldsymbol{\Sigma} \\ \Delta\boldsymbol{\Theta} \end{pmatrix}^{(i+1)} = \begin{pmatrix} \mathbf{R}_\phi \\ \mathbf{R}_\sigma \\ \mathbf{R}_\theta \end{pmatrix}^{(i)}.$$

(4) Update the solution such that

$$\begin{pmatrix} \boldsymbol{\Phi}_{n+1} \\ \boldsymbol{\Sigma}_{n+1} \\ \boldsymbol{\Theta}_{n+1} \end{pmatrix}^{(i+1)} = \begin{pmatrix} \boldsymbol{\Phi}_{n+1} \\ \boldsymbol{\Sigma}_{n+1} \\ \boldsymbol{\Theta}_{n+1} \end{pmatrix}^{(i)} - \begin{pmatrix} \Delta\boldsymbol{\Phi} \\ \Delta\boldsymbol{\Sigma} \\ \Delta\boldsymbol{\Theta} \end{pmatrix}^{(i+1)}.$$

Steps (1) through (4) are repeated until the norms of the global residual vector are reduced to a certain tolerance (10^{-8} in all the examples shown in this work) of their initial value. Convergence is usually achieved in 2 or 3 nonlinear iterations per time step.

framework are being developed and can be found in [116]. This section describes the calculation of the free energy for the discretization, presents numerical evidence to verify the results in section 3.3.4 in two dimensions, and shows the performance of the method on some more challenging three-dimensional problems related to the growth of crystals in a supercooled liquid.

3.5.1 Free-energy computation

If one uses spaces that are at least $\mathcal{C}^1$-continuous, the free energy can be computed as

$$\mathcal{F}[\phi_n^h] = \int_\Omega \left[\Psi(\phi_n^h) + \frac{1}{2} \left(\left(\phi_n^h\right)^2 - 2|\nabla \phi_n^h|^2 + (\Delta \phi_n^h)^2 \right) \right] d\Omega.$$

Modifications are needed in the 2+2+2 case though, as the discrete atomistic density ϕ^h only lives in $\mathcal{H}^1$. As such, $\Delta\phi_n^h$ is undefined. This obstacle can be overcome by making use of the auxiliary variable θ, as

$$\theta = (1 + \Delta)\, \phi \quad \Leftrightarrow \quad \Delta\phi = \theta - \phi,$$

such that the free energy functional can be computed as

$$\mathcal{F}[\phi_n^h] = \int_\Omega \left[\Psi(\phi_n^h) + \frac{1}{2} \left(\left(\phi_n^h\right)^2 - 2|\nabla \phi_n^h|^2 + (\theta_n^h - \phi_n^h)^2 \right) \right] d\Omega.$$

Remark 3. *The use of the auxiliary variables means that they also have to be initialised, as the initial condition is only specified for ϕ. A nonlinear L_2 projection is performed to solve the semidiscrete versions of equations* (3.7b) *and* (3.7c)*, shown in the last two lines of equation* (3.8)*.*

3.5.2 Numerical validation of the stable scheme

As a test example, we simulate the two-dimensional growth of a crystal in a supercooled liquid, using one-mode approximations for the density profiles of the crystalline structures [30, 31]. The one-mode approximation corresponding to a triangular configuration is defined as

$$\phi_S(\mathbf{x}) = \cos\left(\frac{q}{\sqrt{3}}y\right)\cos(qx) - \frac{1}{2}\cos\left(\frac{2q}{\sqrt{3}}y\right), \tag{3.33}$$

where q represents a wavelength related to the lattice constant [14], and x and y represent the Cartesian coordinates. A solid crystallite is initially placed in the centre of a liquid domain, which is assigned an average density $\bar{\phi}$. The initial condition becomes

$$\phi_0(\mathbf{x}) = \bar{\phi} + \omega(\mathbf{x})\left(A\phi_S(\mathbf{x})\right), \tag{3.34}$$

where A represents an amplitude of the fluctuations in density, and the scaling function $\omega(\mathbf{x})$ is defined as

$$\omega(\mathbf{x}) = \begin{cases} \left(1 - \left(\dfrac{||\mathbf{x}-\mathbf{x}_0||}{d_0}\right)^2\right)^2 & \text{if} \quad ||\mathbf{x}-\mathbf{x}_0|| \leq d_0 \\ 0 & \text{otherwise} \end{cases}$$

where $\mathbf{x}_0$ is the coordinate of the center of the domain, and d_0 is $^1/_6$ of the domain length in the x-direction. Different lattices can be reproduced, depending on the values used for ϵ and the average atomistic density $\bar{\phi}$. Phase diagrams have been developed in both two [14] and three [103] dimensions. In order to avoid mismatches on the boundaries when the grain boundaries meet, the computational domain Ω is dimensioned in such a way as to make it periodic along both directions. To do this

while keeping the problem within a reasonable size, we use the frequency present in equation (3.33) to define the domain Ω as

$$\Omega = \left[0, \frac{2\pi}{q}a\right] \times \left[0, \frac{\sqrt{3}\pi}{q}b\right],$$

where a and b are assigned values of 10 and 12, respectively. These numbers are chosen so that the domain is almost square. The number of elements in the y-direction, N_y, is then defined as

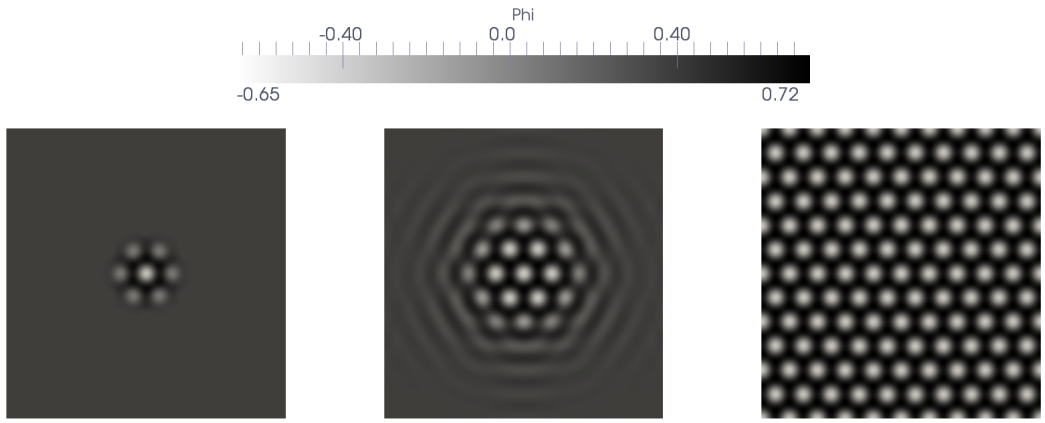
$$N_y = \left\lfloor \frac{b\sqrt{3}}{2a}N_x + \frac{1}{2} \right\rfloor,$$

where N_x represents the number of elements in the x-direction. This adjustment is made to account for the difference in length between both directions, and to have the element size h in both directions be approximately equal. The variables q and A are assigned their corresponding equilibrium values, obtained by minimizing the free energy presented in equation (3.1), with respect to both A and q, while using the approximation of equation (3.33) to define the atomistic density. For the results presented in this section, the values used are

$$\epsilon = 0.325, \quad \bar{\phi} = \frac{\sqrt{\epsilon}}{2}, \quad A = \frac{4}{5}\left(\bar{\phi} + \frac{\sqrt{15\epsilon - 36\bar{\phi}^2}}{3}\right), \quad q = \frac{\sqrt{3}}{2}.$$

The parameter ϵ is chosen such that the triangular structure is stable [14, 103]. Snapshots of the simulation are shown in Fig. 3.1. The initial crystallite placed in the centre of the domain grows at the expense of the supercooled liquid, a state which is enforced by the degree of undercooling ϵ. The non-increasing free energy and mass conservation, properties that need to be verified for a numerical scheme to be valid when solving this equation, can be verified in Figs. 3.2 and 3.3, respectively.

This same example was used to perform the numerical validation of the results



$t = 0$ $t = 25$ $t = 150$

Figure 3.1: Snapshots of the approximate dimensionless atomistic density field showing its evolution throughout the simulation, which was run using a computational mesh composed of 256×266 $\mathcal{C}^0$ linear elements, with a time step size of 1.0.

presented in section 3.3.4. The stabilization term α_n was assigned a value of 0.25 for the range of time and space resolutions covered in this section. This choice of α_n was made after *a-priori* numerical experimentation for this specific example using equations (3.29) and (3.21).

In order to study the convergence in time of the proposed method, a reference solution is required. We obtain this reference solution using a grid with $[128 \times 133]$ elements, $p^2\mathcal{C}^0$ basis functions, and a time step size $\Delta t = 10^{-2}$. This solution was obtained within a matter of hours using a workstation with 32 processor cores. The order of the basis function p was elevated in the case of this solution, as it is a more sensible choice than going for h-refinement with $p^1\mathcal{C}^0$ basis functions, as shown in [81]. Then, to assess the quality of this reference solution in terms of the error in the free energy, an overkill solution was calculated using a grid with $[128 \times 133]$ elements, $p^4\mathcal{C}^0$ spaces, and an order of magnitude smaller time step size $\Delta t = 10^{-3}$. Using the same machine as before, the overkill solution took a week and a half to be completed. The free energy evolutions of the reference and overkill solutions are compared in Fig. 3.4a, while the relative error evolution between the free energies is shown in

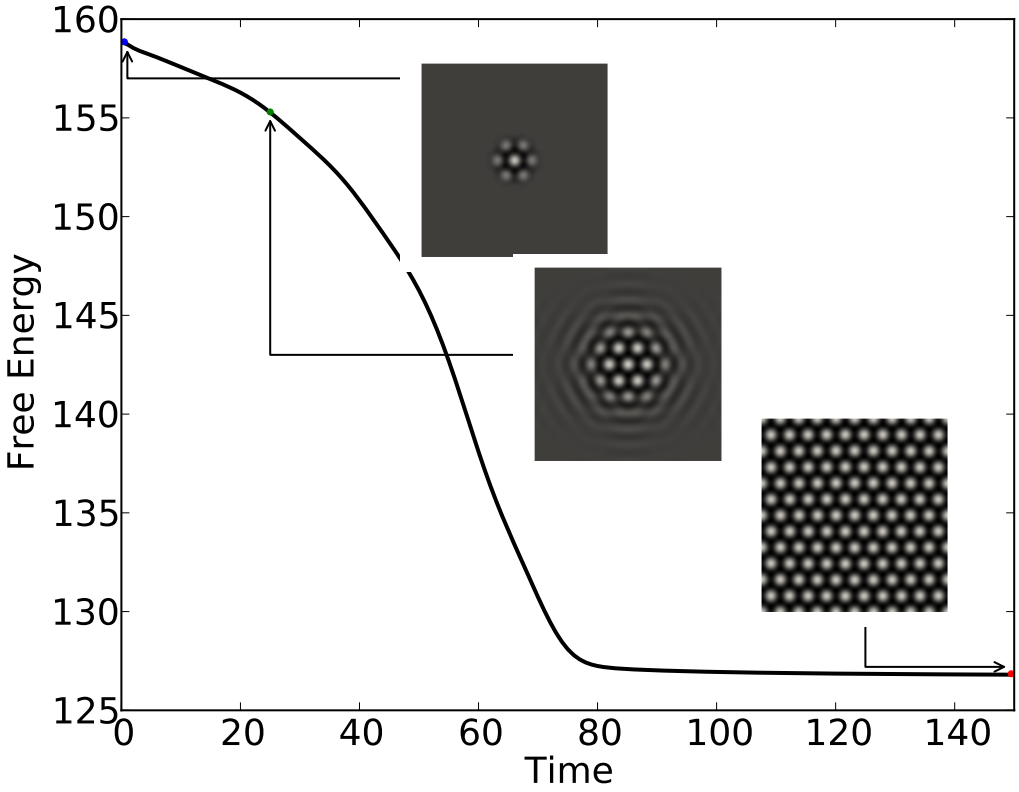


Figure 3.2: Free energy evolution. The free energy is monotonically decreasing throughout the simulation, which was run using a computational mesh composed of 256×266 $\mathcal{C}^0$ linear elements. A time step size of 1 was used, with an α_n value of 0.25.

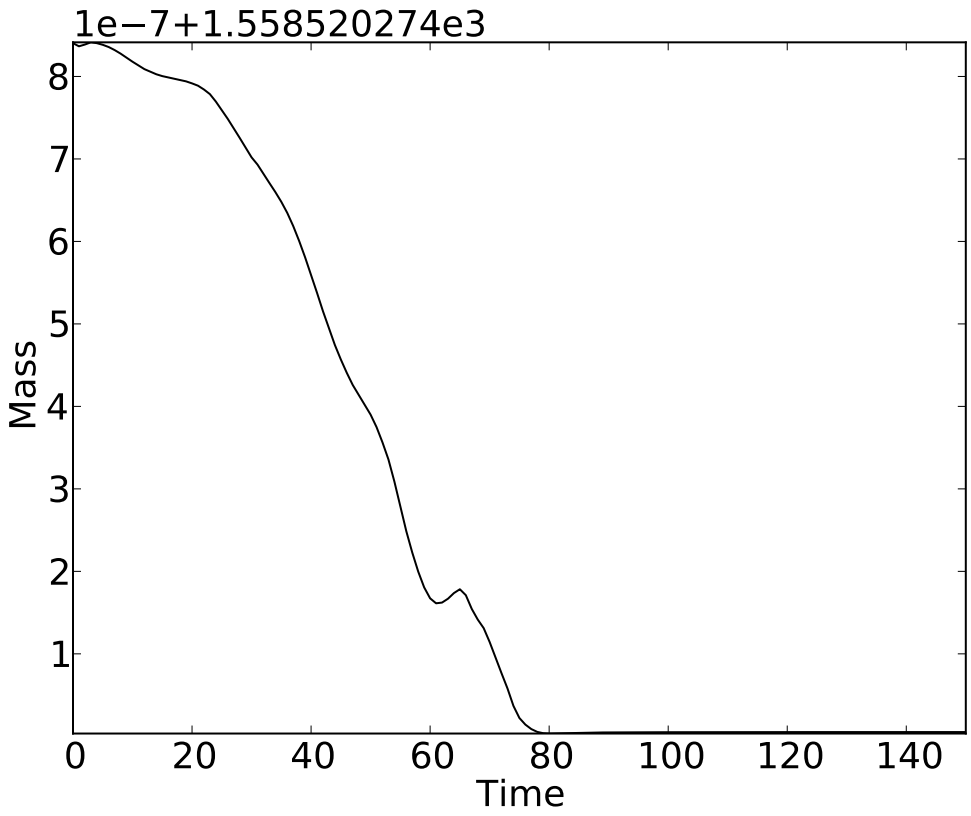


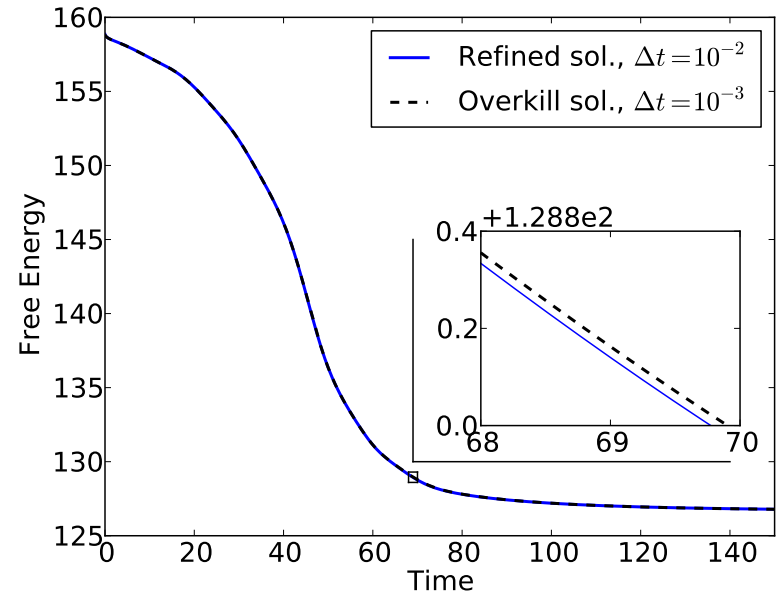
Figure 3.3: Mass evolution. The changes in mass are below the criterion for numerical convergence, which validates numerically that mass is indeed conserved. The error can be attributed to quadrature as well as the iterative solver.

Fig. 3.4b. This comparison allows us to conclude that the reference solution is refined enough in space and time to proceed with the study of convergence in time.

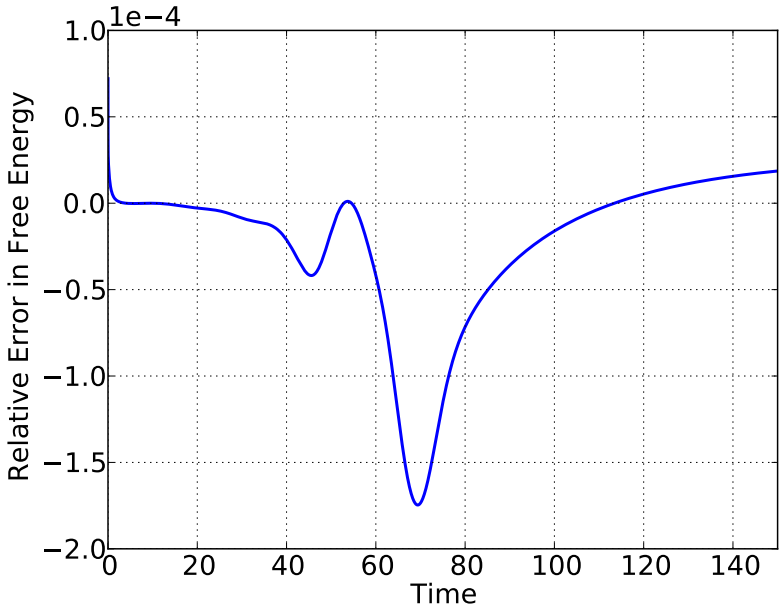
We proceed to study the temporal order of accuracy of the method. Using the same spatial resolution as our reference solution, we perform simulations over a range of time step sizes, and focus on the $\mathcal{L}^2$-error norm in space

$$||e||_2 = \left(\int_\Omega \left(\phi^h - \phi^h_* \right)^2 d\Omega \right)^{1/2}$$

where ϕ^h are the coarse-in-time solutions and ϕ^h_* corresponds to the reference solution. We compute this error at $t = 150$, point in time at which the crystal lattice has already grown over the whole domain. The convergence in time is shown in Fig. 3.5, where we observe that the numerical scheme is indeed second-order accurate. The maximum relative error in mass with different time step sizes is shown in Fig. 3.6. We conclude that the mass is indeed conserved in all cases, as the maximum relative error in mass for different time step sizes stays below 10^{-9}. Fig. 3.7 shows the time evolution of free energy with different time step sizes. Free-energy monotonicity is verified for all the cases, as no increases in free energy are observed. The increase in time step size nonetheless leads to a poorer dynamical representation of the free energy evolution, which is consistent with other published results [40, 28]. Care has to be taken when choosing α_n, as increasing the stabilization parameter has a negative effect on the free energy approximation. This can be seen in Figs. 3.8(a), 3.8(b), 3.8(c) and 3.8(d), where free energy is plotted for different values of α_n and Δt. Nonetetheless, as long as the stabilization parameter α_n complies with the bound presented in equation (29), free energy is dissipated. Even though the dynamics of the equation are influenced by the time step size, the method converges to the right steady state solution. This could be an advantage if what is looked for is the steady state solution to a problem, such as in control of dynamical systems [117]. The use of α_n slows down the dynamics



(a) Free energy evolution of overkill and reference solutions.



(b) Relative error between reference and overkill free energy evolutions.

Figure 3.4: Free energy evolutions of reference and overkill solutions. In (a), the free energy evolution of an overkill solution using $[128 \times 133]$ quartic $\mathcal{C}^0$ elements and a time step size of 10^{-3} is shown along with the free energy evolution corresponding to a reference solution obtained using $[128 \times 133]$ quadratic $\mathcal{C}^0$ elements and a time step size of 10^{-2}. An inset plot is shown on the bottom right corner of (a), in the region where the error is highest throughout the simulation as can be verified in (b), where the relative error between the reference and overkill free energy evolutions is shown.

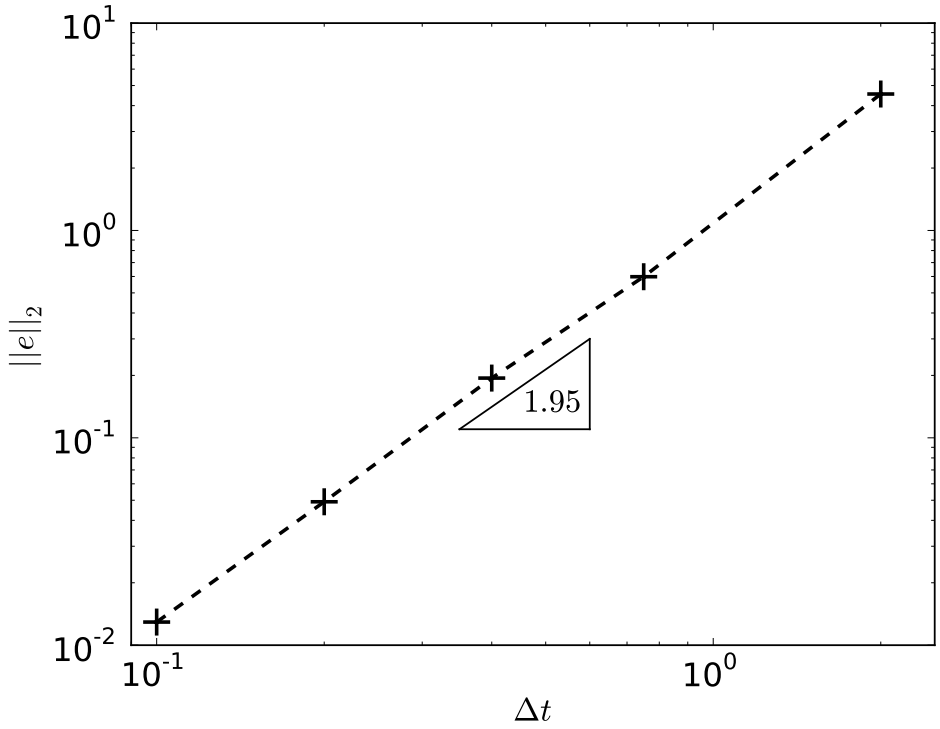


Figure 3.5: Log of $\mathcal{L}^2$-norm of the error at time $T = 150$ versus the log of time step size Δt. The value of the slope confirms the method is second-order accurate in time. The mesh used was made up of $[128 \times 133]$ quadratic $\mathcal{C}^0$ elements, such that the spatial error could be considered negligible in the simulations. The parameter α_n was given a value equal to 0.25, which complies with the bound presented in section 3.3.4 for this problem.

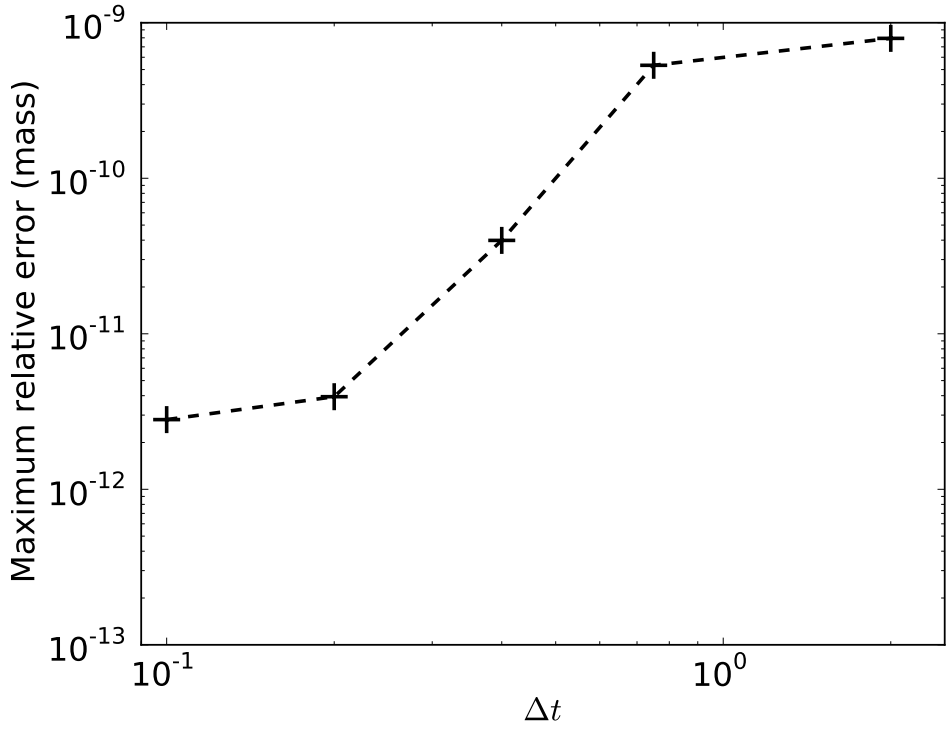


Figure 3.6: Mass conservation. The maximum relative error over the entire evolution of the system remained below 10^{-9} for the simulations considered in this work. The mesh used was made up of $[128 \times 133]$ quadratic $\mathcal{C}^0$ elements, such that the spatial error could be considered negligible in the simulations. The parameter α_n was given a value equal to 0.25, which complies with the bound presented in section 3.3.4 for this problem.

of the equation, and an effective time-step size needs to be determined. We plan to study this point further in future work [81].

The results in this section validate numerically the theoretical results presented in section 3.3.4 with regards to this numerical formulation, and prove that it is indeed mass conserving, unconditionally energy-stable, and is second-order accurate in time.

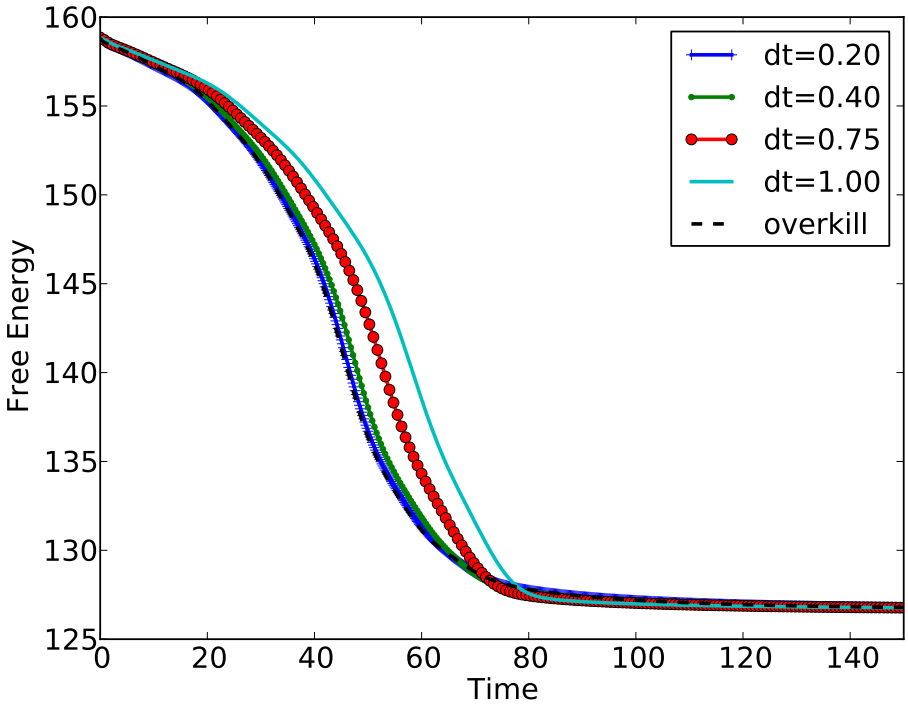
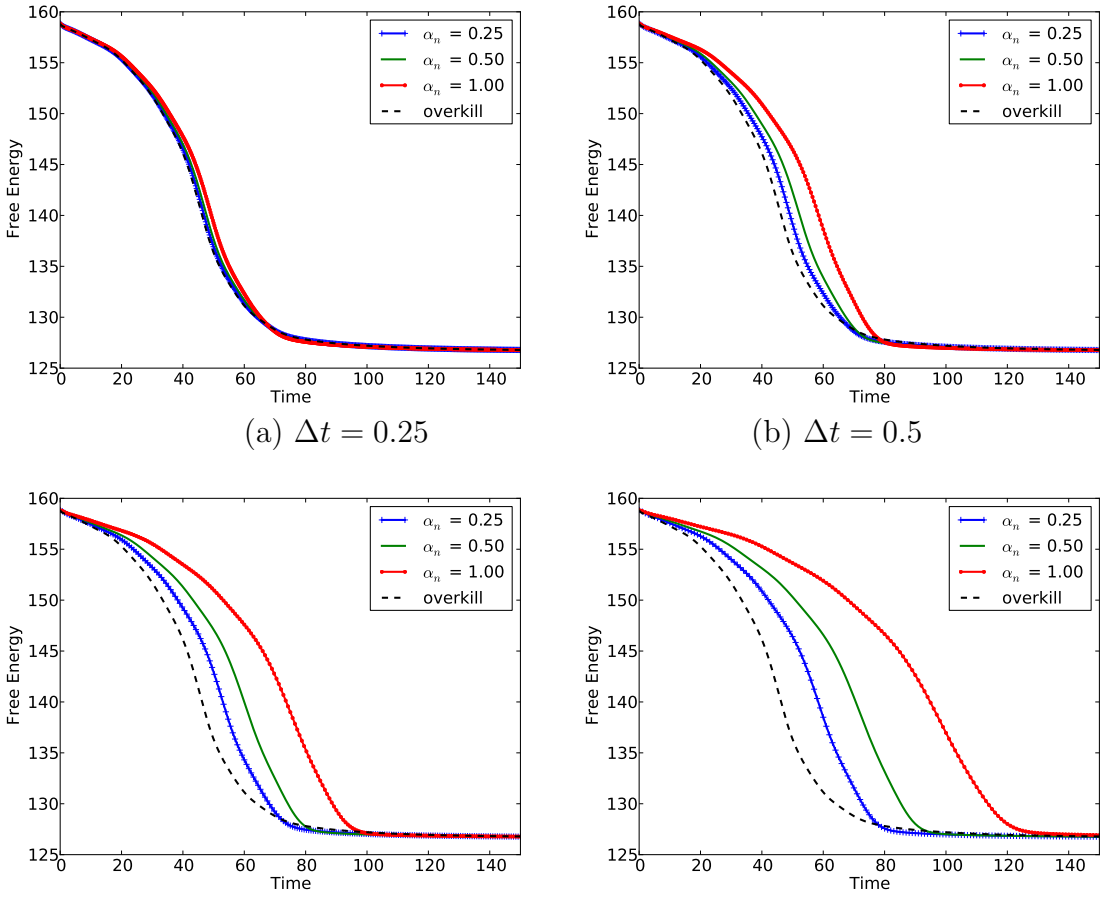


Figure 3.7: Free energy monotonicity. The free energy functional of the system exhibits strong energy stability, such that $\mathcal{F}\left[\phi\left(t_{n+1}\right)\right] \leq \mathcal{F}\left[\phi\left(t_{n}\right)\right]$. This is independent of the time step size used as can be observed in the plot. The mesh used was made up of $[128 \times 133]$ quadratic $\mathcal{C}^0$ elements, such that the special error could be considered negligible in the simulations. The parameter α_n was given a value equal to 0.25, which complies with the bound presented in section 3.3.4 for this problem.



(a) $\Delta t = 0.25$ (b) $\Delta t = 0.5$

(c) $\Delta t = 0.75$ (d) $\Delta t = 1.0$

Figure 3.8: Stabilization parameter variation in two dimensions. The free energy is plotted as a function of time using time step sizes (a) $\Delta t = 0.25$, (b) $\Delta t = 0.5$, (c) $\Delta t = 0.75$ and (d) $\Delta t = 1.0$, respectively. Increasing the stabilization parameter α_n or the time step size Δt results in a less accurate dynamical representation, but converges to the correct steady state solution. The mesh consists of $[128 \times 133]$ quadratic $\mathcal{C}^0$ elements, such that the spatial error could be considered negligible in the simulations.

3.6 Three dimensional simulations: Crystalline growth in a supercooled liquid

In this section, we deal with the three dimensional version of the example described in section 3.5.2, as well as a more challenging case, where two crystallites oriented in different directions are grown in the same domain. The PFC equation in this latter case is able to capture the emergence of grain boundaries.

3.6.1 Crystalline growth in a supercooled liquid

In this example, the growth of a single crystal with a BCC structure is simulated. Mathematically, the crystallite is now defined as [14, 86]

$$\begin{aligned}\phi_{\mathrm{BCC}}(\mathbf{x}) = &\cos(xq_{\mathrm{BCC}})\cos(yq_{\mathrm{BCC}}) + \cos(xq_{\mathrm{BCC}})\cos(zq_{\mathrm{BCC}}) \\ &+ \cos(yq_{\mathrm{BCC}})\cos(zq_{\mathrm{BCC}}),\end{aligned} \tag{3.35}$$

where x, y and z represent the three-dimensional Cartesian coordinates and q_{BCC} represents a wavelength related to the BCC crystalline structure. The computational domain is $\Omega = [0, 20\pi]^3$, with periodic boundary conditions being assumed again in all directions. Similarly to what is done in equation (3.34) for the two-dimensional case, the initial condition is defined as

$$\phi_o(\mathbf{x}) = \bar{\phi}_{BCC} + \omega(\mathbf{x})\left(A\phi_{BCC}(\mathbf{x})\right), \tag{3.36}$$

where $\bar{\phi}_{BCC}$ represents again the average density of the liquid domain, and A represents an amplitude of the fluctuations in density. To ensure the stability of the BCC

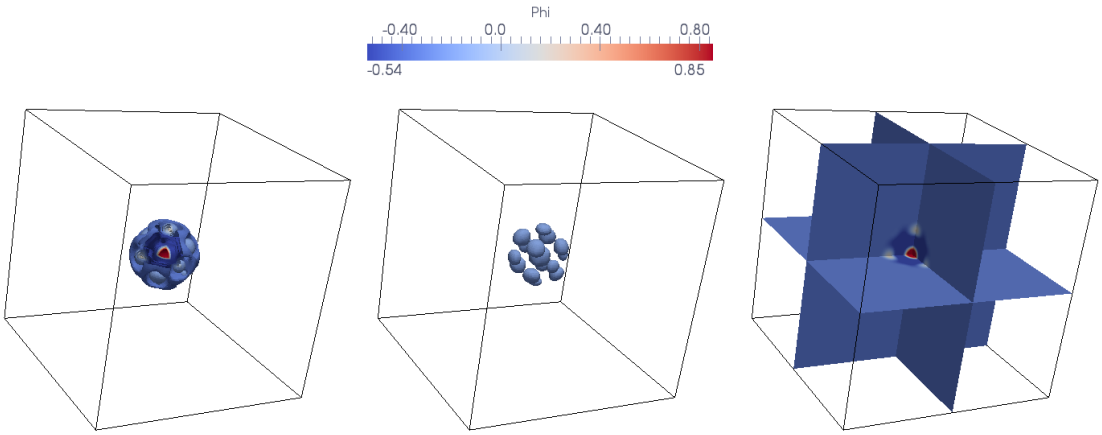
phase, the parameters of the equation are given the following values

$$\epsilon = 0.35, \quad \bar{\phi}_{BCC} = -0.35, \quad q_{BCC} = \frac{1}{\sqrt{2}}, \quad A = 1.$$

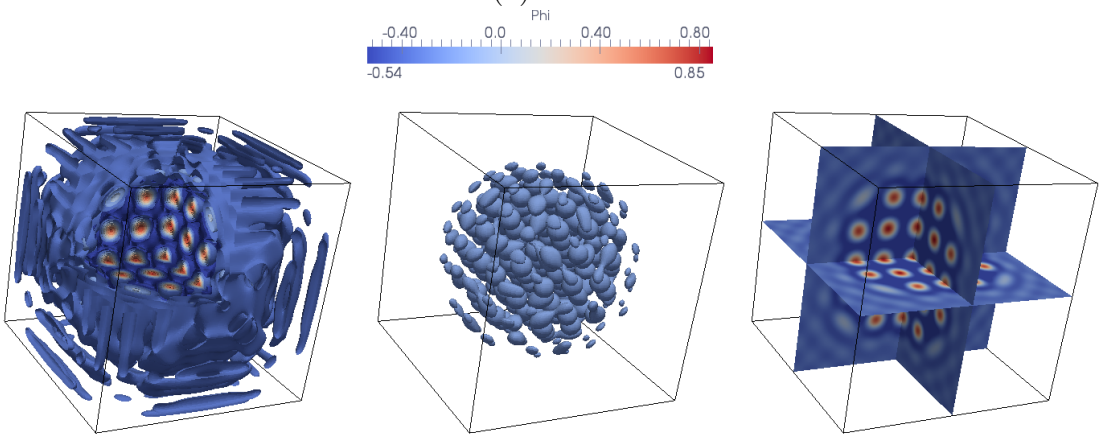
The initial crystallite is placed in the centre of the domain. Similarly to what happens in the two dimensional case, the crystal grows at the expense of the liquid. Snapshots of the solution can be observed in Fig. 3.9. The figure shows that the initial BCC pattern is repeated over the whole domain, until reaching a steady state. The simulation uses a uniform grid composed of $[150]^3$ linear elements, and a time step size $\Delta t = 0.5$. The stabilization parameter α_n is set to 0.5. The free energy evolution for the simulation is shown in Fig. 3.10. There are no increases in free energy. The mass also remains constant throughout the simulation.

3.6.2 Polycrystalline growth of BCC crystals

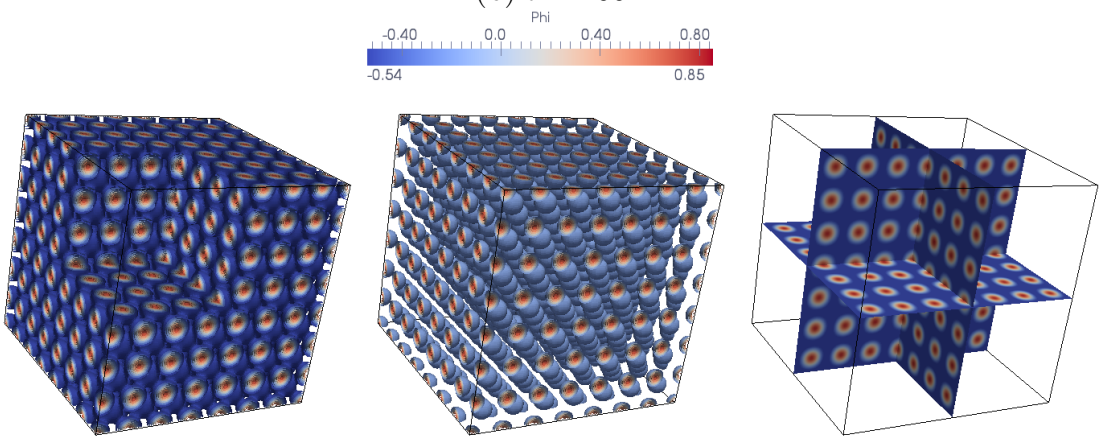
As a more challenging example, we present a case of polycrystalline growth, where two initial crystallites with a BCC configuration oriented in different directions are placed in the domain. They are set at different angles, so as to eventually observe the emergence of grain boundaries when both crystallites meet. The computational domain is $\Omega = [0, 40\pi]^3$, and periodic boundary conditions are imposed in all directions. A uniform mesh comprised of $[150]^3$ linear elements is used, along with a time step size $\Delta t = 0.5$. The stabilization parameter α_n is set to 0.5. A system of local Cartesian coordinates (x_C, y_C, z_C) was used to generate the crystallites in different directions, by doing an affine transformation of the global coordinates (x, y, z) to



(a) $t = 0$



(b) $t = 100$



(c) $t = 250$ (steady state)

Figure 3.9: Crystal growth in a supercooled liquid in three dimensions. The images show the evolution of one crystallite surrounded by liquid. The labels indicate the computational time. On the left-hand side, we show isosurfaces of the solution, in the middle we present the same isosurfaces where a thresholding filter has been applied to only show the atoms, such that the periodic nature of the lattice is clear, while on the right-hand side we present slices of the solution across the indicated planes.

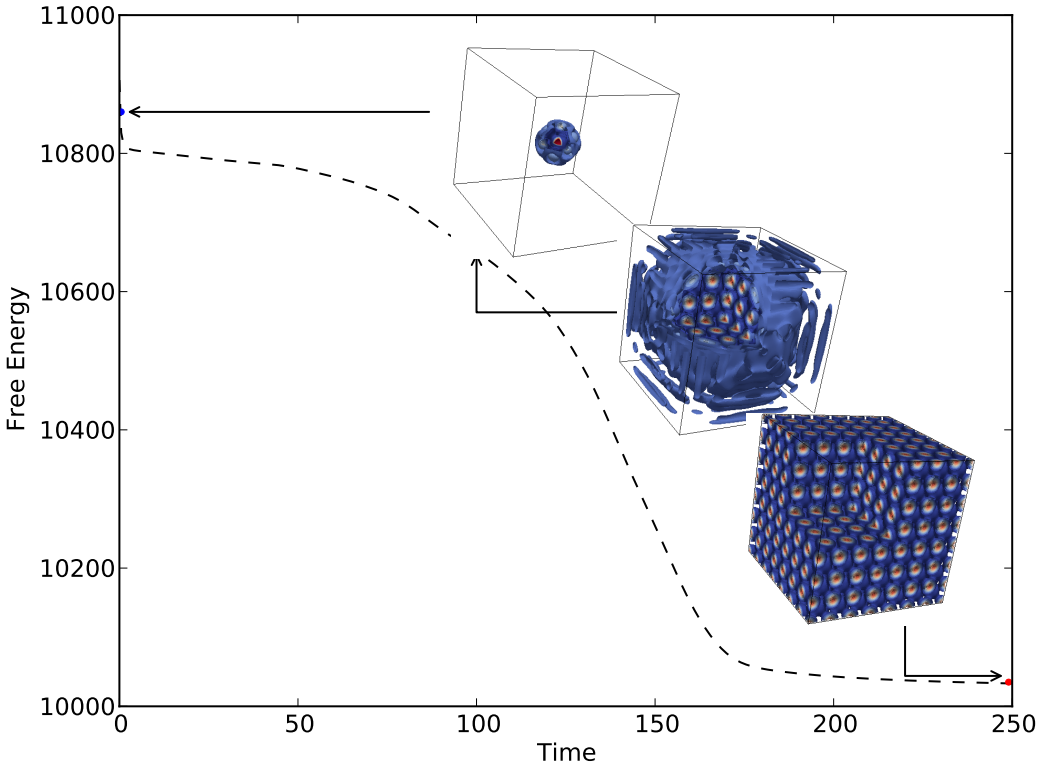


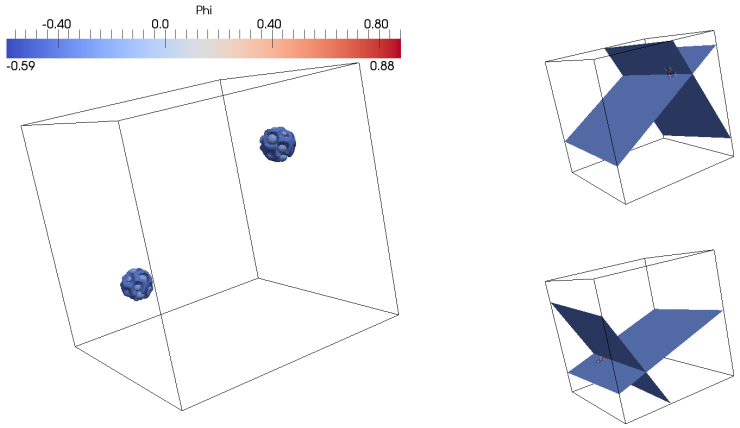
Figure 3.10: Free energy evolution of a single crystal. The free energy is monotonically decreasing while the mass remains constant throughout the simulation (the maximum relative error stays below 10^{-9}), which was run using a mesh composed of $[150]^3$ linear elements. A time step size of 0.5 was used, with an α_n value of 0.5.

produce a rotation β along the z axis, with

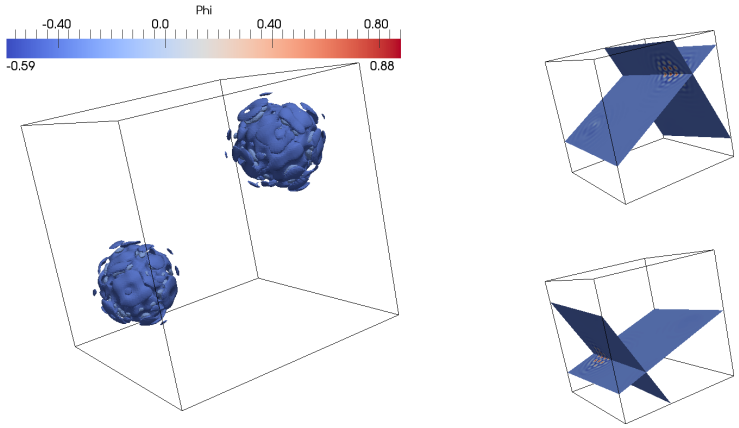
$$\begin{pmatrix} x_C \\ y_C \\ z_C \end{pmatrix} = \begin{pmatrix} \cos(\beta)\, x - \sin(\beta)\, y \\ \cos(\beta)\, x + \sin(\beta)\, y \\ z \end{pmatrix}. \tag{3.37}$$

The first crystallite was defined as in equation (3.35), with $\beta = 0$, while the second one was rotated by an angle $\beta = \frac{\pi}{8}$. The same equation parameters that were used in section 3.6.1 are used in this example, and result in the simulation shown in Fig 3.11. Grain boundaries appear when the two crystals meet while growing, given the orientation mismatch. The free energy evolution is shown in Fig. 3.12, where no increases in free energy are seen.

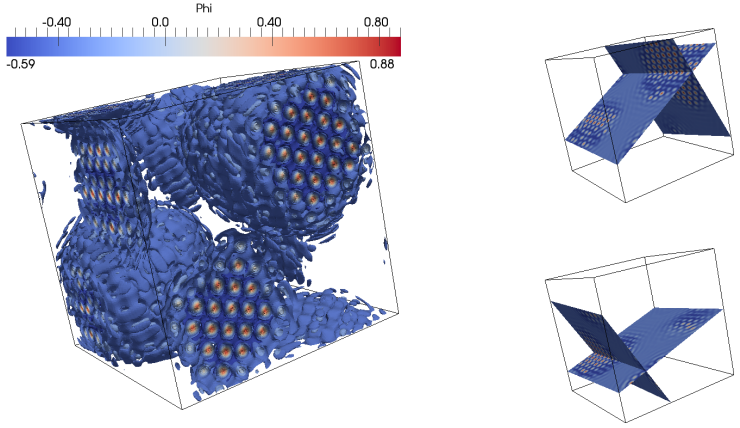
Changing the rotation angle β can have an effect on the free energy of the system, as it influences the grain boundary that is formed. The free energy evolution is plotted in Fig. 3.13 for three different values of β. In the two cases where the rotation angle is relatively small $\left(\text{i.e, } \beta = \frac{\pi}{8} \text{ and } \beta = \frac{\pi}{16}\right)$, the same steady state is reached, as the equation leads both systems to the same energetically minimal state. On the other hand, when the change in β is larger $\left(\frac{\pi}{2}\right)$, the free energy value at steady state differs significantly. The grain boundary formed is considerably different than the ones considered before, as the two grains meet at a significantly different position. Further studies are needed to conclude if the free energy differences are qualitatively accurate and compare well with experiments.



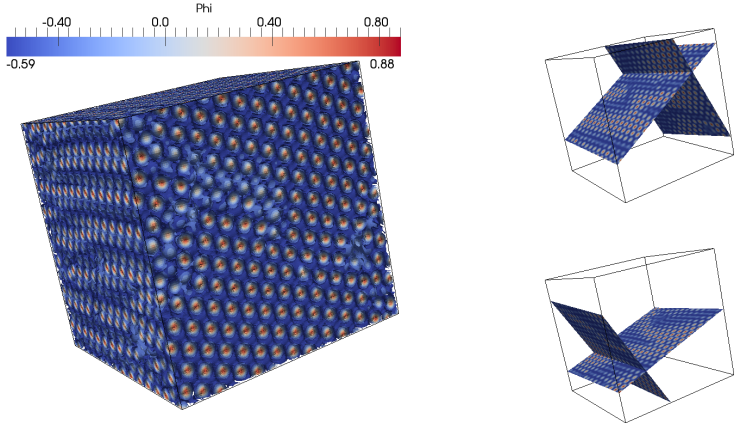
(a) t = 0.



(b) t = 100.



(c) t = 250.



(d) t = 500.

Figure 3.11: Polycrystalline growth in a supercooled liquid in three dimensions. The images show isocontours of the atomistic density field, where two crystallites are initially placed in a domain with different orientations. Grain boundaries emerge once the crystals meet. The labels indicate the computational time, while the mesh used $[150]^3$ linear elements. A time step size of 0.5 was used, with an α_n value of 0.5.

3.7 Discussion

In this chapter, we present a provably stable scheme to solve the phase-field crystal equation. This algorithm conserves mass, guarantees energy stability and is second-order accurate in time. Theoretical proofs are presented, along with numerical results that corroborate them in two spatial dimensions. The method is based on a mixed finite element formulation that involves three coupled, second-order equations. Three-dimensional results involving polycrystalline growth are also presented, showcasing the robustness of the method. The implementation is done using PetIGA, a high-performance isogeometric analysis framework, and the codes are freely available to download. [1]

[1] https://bitbucket.org/dalcinl/petiga

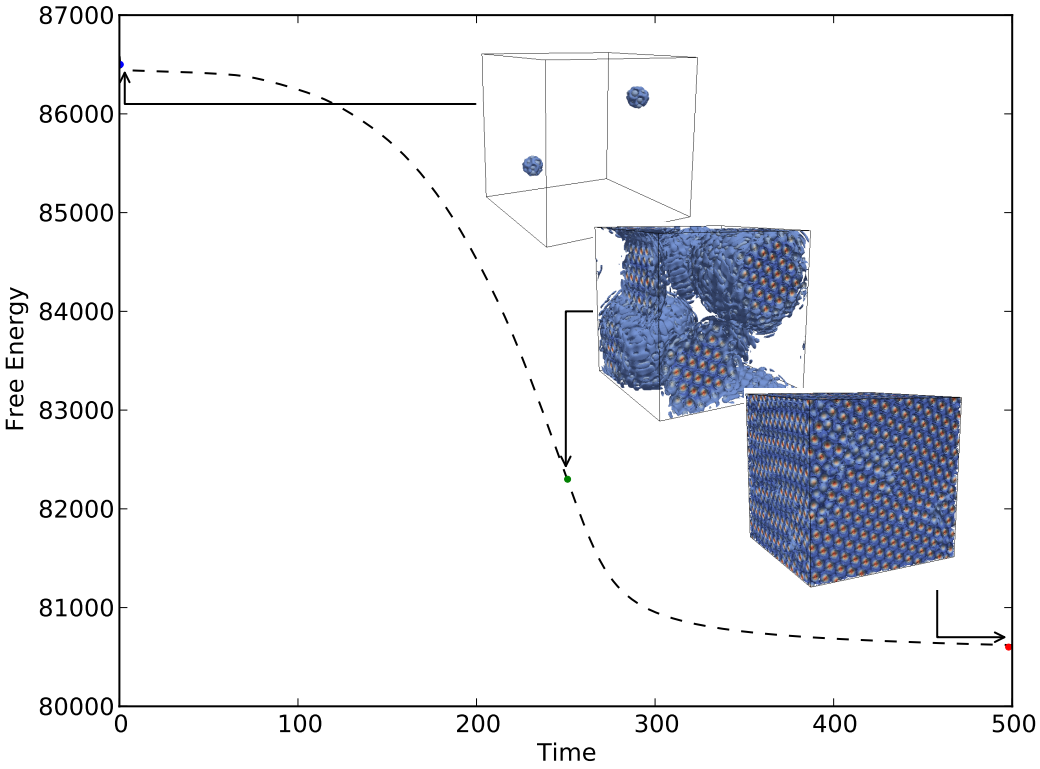


Figure 3.12: Free energy evolution of two crystals. The free energy is monotonically decreasing while the mass remains constant throughout the simulation (the maximum relative error stays below 10^{-9}), which was run using a mesh composed of $150^3\mathcal{C}^0$ linear elements. A time step size of 0.5 was used, with an α_n value of 0.5.

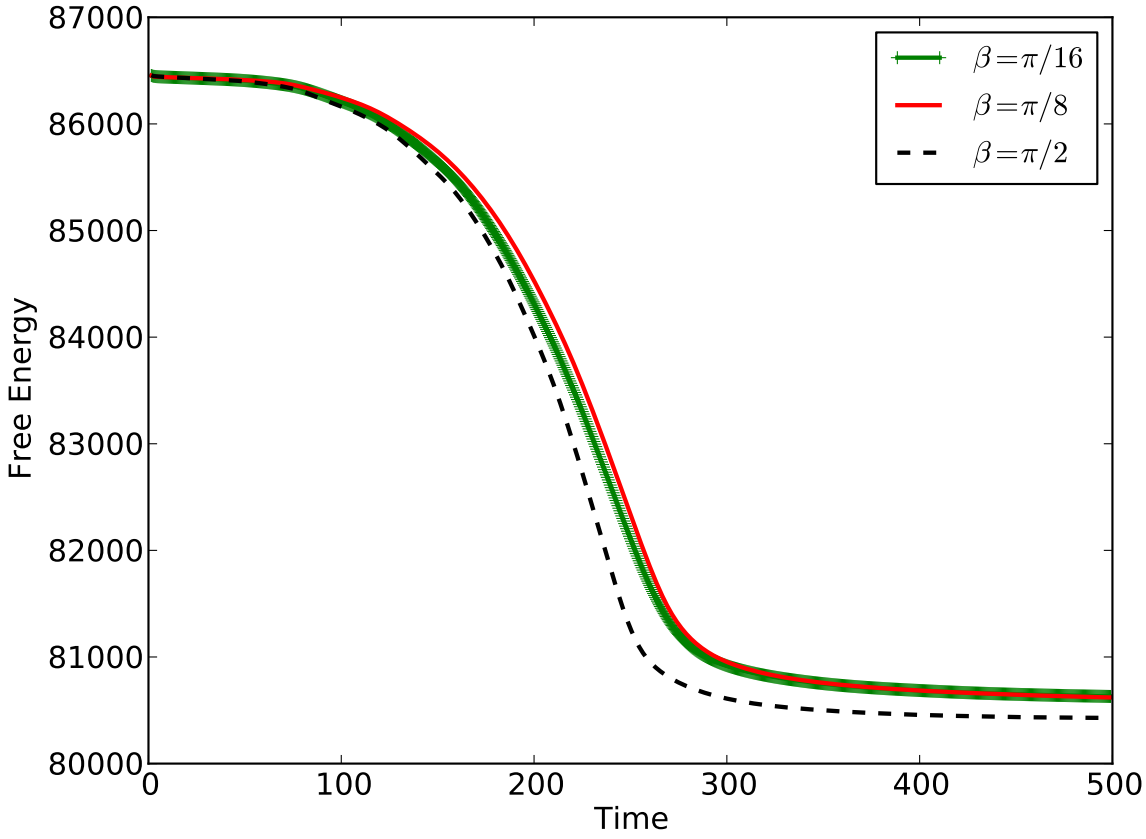


Figure 3.13: Effect of rotation angle on the crystallites. The plotted solutions use a time step size $\Delta t = 0.5$, a stabilization parameter $\alpha_n = 0.5$, and $[150]^3$ linear elements. Three different rotation angles β are considered.

Chapter 4

Linear, Unconditionally Energy-Stable Time-Integrators for Phase-Field Models

In this chapter, we introduce provably energy-stable time-integration methods for general classes of phase-field models. We demonstrate how Taylor series expansions of the nonlinear terms present in the partial differential equations of these models can lead to expressions that guarantee energy-stability implicitly, which are second-order accurate in time. Additionally, the system can be rendered linear, decreasing the computationl time needed to obtain a solution considerably. The spatial discretization relies on a mixed finite element formulation and isogeometric analysis. We also propose an adaptive time-stepping discretization that relies on a first-order backward approximation to give an error-estimator. This methodology can be applied to any second-order accurate time-integration scheme. We present numerical examples in two and three spatial dimensions, which confirm the stability and robustness of the method. The implementation of the numerical schemes is done in PetIGA, a high-performance isogeometric analysis framework.

The chapter is structured as follows: in section 4.1, we give some background on the general problem of numerical energy stability of phase-field problems. In section 4.2, we describe the general class of phase-field models we consider, and present the development of our unconditionally stable time integrator. In section 4.3, we present our spatial discretization strategy, detail the numerical solution process, and present the novel time-adaptive algorithm. In section 4.4, we re-introduce the four phase-field models we briefly described in Chapter 2, which we proceed to solve in section 4.5 using different time-stepping schemes. We conclude with section 4.6.

4.1 Background and Perspective

Numerical techniques that satisfy thermodynamic relations at the discrete level in phase-field models have been developed for both conserved [2, 3] and non-conserved phase-field models [4, 43]. The method we describe in this Chapter generalizes some of these algorithms, and shows that the nonlinear terms present in the equations can be discretized in time using Taylor series to guarantee both second-order accuracy and strong energy stability. Additionally, the numerical schemes developed can be rendered linear in time by keeping appropriate terms in the Taylor series expansions. Given the difference in time scales that can exist in these models, we also present an a posteriori error estimator that can be used to adapt the scheme in time.

Regarding the spatial discretization, we use isogeometric analysis (IGA) [48].The method has successfully been applied in its Galerkin version to solve the Cahn-Hilliard equation [28, 2], the advective [50] and Navier-Stokes-Cahn-Hilliard equations [51, 118], the Swift–Hohenberg equation [4] and the phase-field crystal equation [3, 52, 5]. IGA possesses some advantages over standard finite element methods, which include being able to easily generate high-order, globally continuous basis functions as well as exact geometrical representations as the finite element space is refined. However,

when used to discretize the higher-order partial differential equations, the NURBS-based spaces result in expensive linear systems [53, 54, 55] that need efficient and scalable solvers that mitigate the increase of cost. For this reason, we implemented our algorithm in a high-performance isogeometric analysis framework, PetIGA [56, 57]. Recently, isogeometric collocation methods have also been used to discretize phase-field models [58, 59].

4.2 Energy stability of phase-field models

To model interfacial problems, phase-field methods represent the sharp interfaces between the different phases through thin transition regions that distribute the interfacial forces. In these models, the phase-field or order-parameter varies smoothly (i.e., continuously) over the thin layers, with a key point being that the energy of these models needs to be dissipated as time progresses. The existence of a Lyapunov functional for these diffuse-interface problems implies strong energy stability [44], a property which can be lost if inadequate algorithms and/or spatial/temporal resolutions are used to solve the partial differential equation [44, 80, 4, 81]. This work addresses the nonlinear stability issue for both conserved and non-conserved phase-field variables, and presents a simple process that relies on Taylor series to handle the nonlinear terms present in the partial differential equation.

4.2.1 An Abstract Model Problem

The phase-field framework relies on the minimization of a free energy functional that models a system undergoing a phase transition. This functional depends on a phase-field variable ϕ and its derivatives, and tries to capture both the bulk behavior of the components involved as well as their interfacial behavior. The free energy functional

$\mathcal{F}$ for many important phase-field models can generally be written as [14]

$$\mathcal{F}\big[\phi(\mathbf{x})\big] = \int_{\Omega} \big(f(\phi, \nabla\phi, \Delta\phi) + \Psi(\phi)\big) d\Omega, \tag{4.1}$$

where f is a function of ϕ and its derivatives, such that

$$\begin{aligned} f(\phi, \nabla\phi, \Delta\phi) &= f_1(\phi) + f_2(\nabla\phi) + f_3(\Delta\phi) \\ &= \frac{1}{2}\left(c_0\phi^2 + c_1|\nabla\phi|^2 + c_2\left(\Delta\phi\right)^2\right), \end{aligned} \tag{4.2}$$

where c_0, c_1, and c_2 are real-valued constants, ∇ and Δ represent the gradient and Laplacian operators, respectively, and Ψ, a Landau-type free energy density [14], is a polynomial function of ϕ given by

$$\Psi(\phi) = \frac{1}{4}a_0 - \frac{1}{2}a_2\phi^2 + \frac{1}{4}a_4\phi^4, \tag{4.3}$$

where a_0, a_2 and a_4 are real-valued constants, and a_2 and a_4 are strictly positive. This implies that the fourth derivative of Ψ with respect to ϕ is a constant which verifies

$$\frac{\partial^{iv}\Psi(\phi)}{\partial\phi^{iv}} = \Psi^{iv}(\phi) \geq 0.$$

To simplify notation, we write $\mathcal{F}\left[\phi(\mathbf{x})\right]$ as $\mathcal{F}(\phi)$, and $f(\phi, \nabla\phi, \Delta\phi)$ as $f(\phi)$. The phase field models considered in this work are those obtained through variational arguments, where the evolution in time of the phase-field variable is related to the variational derivative of the free energy functional defined in equation (4.1). In such cases, the initial boundary value problem over the spatial domain Ω and time interval $[0, T]$ is formulated as: given $\phi_0 : \Omega \to \mathbb{R}$, find $\phi : \Omega \times [0, T] \to \mathbb{R}$ such that

$$\begin{cases} \dfrac{\partial \phi}{\partial t} = (-\nabla)^a \cdot \left(-M \nabla^a \dfrac{\delta \mathcal{F}(\phi)}{\delta \phi} \right) & \text{on } \Omega \times]0, T], \\ \phi(\mathbf{x}, 0) = \phi_0(\mathbf{x}) & \text{on } \bar{\Omega}. \end{cases} \tag{4.4}$$

where t stands for a time-like variable, and the parameter a takes a value of one if the phase-field represents a conserved quantity or zero otherwise. Here, $M \equiv M(\phi)$ represents mobility and is defined as either $M = \bar{M}$, with $\bar{M}$ a positive constant, or through the nonlinear expression

$$\begin{cases} M = \bar{M} \left(\dfrac{a_2}{a_4} - \phi^2 \right) \geq 0 \quad \forall \quad \phi \in \left] -\sqrt{\dfrac{a_2}{a_4}}, \sqrt{\dfrac{a_2}{a_4}} \right[, \\ M = 0 \quad \text{otherwise.} \end{cases} \tag{4.5}$$

The initial condition is given by $\phi_0(\mathbf{x})$, and the variational derivative operator $\frac{\delta}{\delta \phi}$ is defined formally as

$$\frac{\delta}{\delta \phi} = \frac{\partial}{\partial \phi} - \nabla \cdot \frac{\partial}{\partial \nabla \phi} + \Delta \frac{\partial}{\partial \Delta \phi}. \tag{4.6}$$

for models including up to Laplacians in $\mathcal{F}$. Higher order models can also be considered by extending equation (4.6). Periodic or no-flux boundary conditions are used in all directions for all the problems considered in this work.

Remark 4. *The discretization can introduce numerical error that makes the phase-field take values outside of its physical range [119], i.e.,* $\phi \in \left] -\sqrt{\dfrac{a_2}{a_4}}, \sqrt{\dfrac{a_2}{a_4}} \right[$. *This issue is simply addressed by setting the mobility to zero as done through equation* (4.5).

Many phase-field models may be derived from Lyapunov functionals using variational arguments [120], which endows them with nonlinear stability. We satisfy this property for a number of models by guaranteeing strong energy stability [44], defined

discretely in time as

$$\mathcal{F}\big(\phi\left(t_{n+1}\right)\big) \leq \mathcal{F}\big(\phi\left(t_{n}\right)\big) \quad \forall\ n = 1, 2, ..., r. \tag{4.7}$$

Thermodynamically consistent algorithms (i.e., energy stable) are necessary. If inadequate discretizations are used to solve the partial differential equations involved, unphysical results where free energy increases can follow [4, 5]. Considering equation (4.7), the jump in free energy $[\![\mathcal{F}]\!]$ is defined as

$$\begin{aligned} [\![\mathcal{F}]\!] &= \mathcal{F}\big(\phi\left(t_{n+1}\right)\big) - \mathcal{F}\big(\phi\left(t_{n}\right)\big), \\ &= \int_{\Omega} \big([\![f]\!] + [\![\Psi]\!]\big) d\Omega \end{aligned} \tag{4.8}$$

where

$$\begin{aligned} [\![f]\!] &= f\big(\phi\left(t_{n+1}\right)\big) - f\big(\phi\left(t_{n}\right)\big), \\ [\![\Psi]\!] &= \Psi\big(\phi\left(t_{n+1}\right)\big) - \Psi\big(\phi\left(t_{n}\right)\big). \end{aligned}$$

With the goal of proving that our method guarantees

$$[\![\mathcal{F}]\!] \leq 0,$$

irrespectively of the mesh and time step-sizes used, the free energy jump can be obtained using the weak form of equation (4.4) as will be shown in section 4.2.3.

4.2.2 A stable scheme

We propose a new second-order accurate, unconditionally energy-stable scheme for phase field models, given by

$$\frac{[\![\phi]\!]}{\Delta t} = (-\nabla)^a \cdot \left(-\tilde{M}\nabla^a \sigma\right), \tag{4.9}$$

$$\sigma = \frac{\partial}{\partial \phi}\left(f_1(\{\phi\}) + \tilde{\Psi}\right) - \nabla \cdot \frac{\partial f_2(\{\phi\})}{\partial \nabla \phi} + \Delta \frac{\partial f_3(\{\phi\})}{\partial \Delta \phi} + (-1)^a \alpha \Delta t \Delta^a [\![\phi]\!], \tag{4.10}$$

where

$$[\![\phi]\!] = \phi_{n+1} - \phi_n; \qquad \Delta t = t_{n+1} - t_n;$$

$$\tilde{M} = M\left(\frac{3}{2}\phi_n - \frac{1}{2}\phi_{n-1}\right); \qquad \{\phi\} = \frac{1}{2}\left(\phi_{n+1} + \phi_n\right).$$

Here, the mobility function $\tilde{M}$ is approximated through a second-order accurate extrapolation from time steps already computed [92], where $M(\cdot)$ is defined by equation (4.5). The parameter α is a stabilization constant whose value depends on the approximation used for $\tilde{\Psi}'$. This parameter guarantees free energy stability. The function $\tilde{\Psi}'$ represents an approximation to Ψ' such that the scheme is stable given sufficient stabilization. We choose the approximation to be either implicit

$$\tilde{\Psi}' = \Psi'_{n+1} - \Psi''_{n+1}\frac{[\![\phi]\!]}{2} + \Psi'''_{n+1}\frac{[\![\phi]\!]^2}{6}, \text{ with } \alpha \geq 0 \tag{4.11}$$

or an explicit one, such that

$$\tilde{\Psi}' = \Psi'_n + \Psi''_n\frac{[\![\phi]\!]}{2}, \text{ with } \alpha \geq \bar{M}\frac{a_2}{a_4}\left(\frac{a_2\Delta t}{6}\right)^2. \tag{4.12}$$

where Ψ'_m is written instead of $\Psi'(\phi_m)$ for the sake of brevity.

4.2.3 Building up $[\![\mathcal{F}]\!]$ from the weak form

The auxiliary variable σ appearing in equations (4.9) and (4.10) is defined as

$$\sigma = \frac{\delta \mathcal{F}(\phi)}{\delta \phi},$$

which in turn modifies the definition of the partial differential equation (4.4), that now reads

$$\frac{\partial \phi}{\partial t} = (-\nabla)^a \cdot (-M \nabla^a \sigma)\,, \tag{4.13}$$

$$\sigma = \frac{\delta \mathcal{F}(\phi)}{\delta \phi}. \tag{4.14}$$

Let us define the functional space $\mathcal{V} \in \mathcal{H}^m$, where $\mathcal{H}^m$ is the Sobolev space of square integrable functions with square integrable derivatives up to order m. In the following, we derive the weak form of equations (4.13)-(4.14), obtained by multiplying them by test functions $w, q \in \mathcal{V}$, respectively, and integrating by parts. The problem then becomes to find $\phi, \sigma \in \mathcal{V}$, such that for all $w, q \in \mathcal{V}$,

$$\left(w, \frac{\partial \phi}{\partial t}\right)_\Omega + (\nabla^a w, M \nabla^a \sigma)_\Omega = 0, \tag{4.15}$$

$$(q, \sigma)_\Omega - \left(q, \frac{\delta \mathcal{F}(\phi)}{\delta \phi}\right)_\Omega = 0, \tag{4.16}$$

where the $\mathcal{L}^2$ inner product over the domain is indicated by $(\cdot,\cdot)_\Omega$. Recalling equations (4.6) and (4.16), we have that

$$\begin{aligned}(q,\sigma)_\Omega &= \left(q, \frac{\delta\mathcal{F}(\phi)}{\delta\phi}\right)_\Omega \\ &= \left(q, \frac{\partial}{\partial\phi}\Big(f_1(\phi)+\Psi(\phi)\Big) - \nabla\cdot\frac{\partial f_2(\phi)}{\partial\nabla\phi} + \Delta\frac{\partial f_3(\phi)}{\partial\Delta\phi}\right)_\Omega \\ &= \left(q, \frac{\partial f_1(\phi)}{\partial\phi}\right)_\Omega + \left(\nabla q, \frac{\partial f_2(\phi)}{\partial\nabla\phi}\right)_\Omega + \left(\Delta q, \frac{\partial f_3(\phi)}{\partial\Delta\phi}\right)_\Omega \\ &+ \left(q, \frac{\partial\Psi(\phi)}{\partial\phi}\right)_\Omega. \end{aligned} \tag{4.17}$$

Recalling the time integration scheme proposed, we substitute equation (4.9) in (4.15) to obtain

$$\left(w, \frac{[\![\phi]\!]}{\Delta t}\right)_\Omega + \left(\nabla^a w, \tilde{M}\nabla^a\sigma\right)_\Omega = 0, \tag{4.18}$$

and substitute equation (4.10) in (4.17) to recover

$$\begin{aligned}(q,\sigma)_\Omega &= \left(q, \frac{\partial f_1(\{\phi\})}{\partial\phi}\right)_\Omega + \left(\nabla q, \frac{\partial f_2(\{\phi\})}{\partial\nabla\phi}\right)_\Omega + \left(\Delta q, \frac{\partial f_3(\{\phi\})}{\partial\Delta\phi}\right)_\Omega \\ &+ \left(q, \tilde{\Psi}'\right)_\Omega + \alpha\Delta t\left(\nabla^a q, \nabla^a[\![\phi]\!]\right)_\Omega. \end{aligned} \tag{4.19}$$

Now, by taking the test function q to be equal to $[\![\phi]\!]$, and taking the test function w as equal to σ, equations (4.18) and (4.19) become

$$(\sigma, [\![\phi]\!])_\Omega = -\Delta t\left(\nabla^a\sigma, \tilde{M}\nabla^a\sigma\right)_\Omega, \tag{4.20}$$

$$\begin{aligned}([\![\phi]\!],\sigma)_\Omega &= \left([\![\phi]\!], \frac{\partial f_1(\{\phi\})}{\partial\phi}\right)_\Omega + \left(\nabla[\![\phi]\!], \frac{\partial f_2(\{\phi\})}{\partial\nabla\phi}\right)_\Omega + \left(\Delta[\![\phi]\!], \frac{\partial f_3(\{\phi\})}{\partial\Delta\phi}\right)_\Omega \\ &+ \left([\![\phi]\!], \tilde{\Psi}'\right)_\Omega + \alpha\Delta t\left(\nabla^a[\![\phi]\!], \nabla^a[\![\phi]\!]\right)_\Omega. \end{aligned} \tag{4.21}$$

Substituting (4.20) in (4.21), we recover that

$$\begin{aligned}-\Delta t\left(\nabla^a\sigma,\tilde{M}\nabla^a\sigma\right)_\Omega &= \left(\llbracket\phi\rrbracket,\frac{\partial f_1(\{\phi\})}{\partial\phi}\right)_\Omega+\left(\nabla\llbracket\phi\rrbracket,\frac{\partial f_2(\{\phi\})}{\partial\nabla\phi}\right)_\Omega\\ &\quad+\left(\Delta\llbracket\phi\rrbracket,\frac{\partial f_3(\{\phi\})}{\partial\Delta\phi}\right)_\Omega+\left(\llbracket\phi\rrbracket,\tilde{\Psi}'\right)_\Omega\\ &\quad+\alpha\Delta t\left(\nabla^a\llbracket\phi\rrbracket,\nabla^a\llbracket\phi\rrbracket\right)_\Omega. \end{aligned}\tag{4.22}$$

Using equation (4.8), it follows that

$$\begin{aligned}\int_\Omega\llbracket f\rrbracket d\Omega &= \llbracket\mathcal{F}(\phi)\rrbracket-\int_\Omega\llbracket\Psi\rrbracket d\Omega\\ &= \left(\llbracket\phi\rrbracket,\frac{\partial f_1(\{\phi\})}{\partial\phi}\right)_\Omega+\left(\nabla\llbracket\phi\rrbracket,\frac{\partial f_2(\{\phi\})}{\partial\nabla\phi}\right)_\Omega+\left(\Delta\llbracket\phi\rrbracket,\frac{\partial f_3(\{\phi\})}{\partial\Delta\phi}\right)_\Omega.\end{aligned}$$

Substituting this result in equation (4.22), we recover that

$$\begin{aligned}\llbracket\mathcal{F}(\phi)\rrbracket &= -\Delta t\left(\nabla^a\sigma,\tilde{M}\nabla^a\sigma\right)_\Omega-\alpha\Delta t\left(\nabla^a\llbracket\phi\rrbracket,\nabla^a\llbracket\phi\rrbracket\right)_\Omega\\ &\quad+\int_\Omega\llbracket\Psi\rrbracket d\Omega-\left(\llbracket\phi\rrbracket,\tilde{\Psi}'\right)_\Omega\\ &= -\Delta t\left(\nabla^a\sigma,\tilde{M}\nabla^a\sigma\right)_\Omega-\alpha\Delta t\left(\nabla^a\llbracket\phi\rrbracket,\nabla^a\llbracket\phi\rrbracket\right)_\Omega\\ &\quad+\int_\Omega\left(\llbracket\Psi\rrbracket-\llbracket\phi\rrbracket\tilde{\Psi}'\right)d\Omega.\end{aligned}\tag{4.23}$$

Since $\tilde{M}$ is always a non-negative quantity within the range where ϕ is defined, i.e., $\left]-\sqrt{\frac{a_2}{a_4}},\sqrt{\frac{a_2}{a_4}}\right[$, the first two terms contribute only to energy decay at each time step. As for the last two terms, the approximation for Ψ' and the stabilization parameter α can be chosen to yield a stable scheme with second order accuracy in time.

Remark 5. *Even though the mixed formulation allows to recover the free energy jump in the case of conserved phase-field models, primal formulations can be used in the case of non-conserved phase-field variables.*

4.2.4 Handling nonlinearities: Taylor expansions guarantee energy-stability

Assuming the free energy functionals to be analytic, the nonlinear term involving Ψ in the free energy $\mathcal{F}$ can be expressed at time levels n or $n+1$ as a Taylor series expansion, from $\Psi_{n+1} = \Psi(\phi_{n+1})$ and $\Psi_n = \Psi(\phi_n)$, respectively. This allows us to write:

$$\Psi_{n+1} = \sum_{a=0}^{\infty} \Psi_n^{(a)} \frac{[\![\phi]\!]^a}{a!} \tag{4.24}$$

$$\Psi_n = \sum_{a=0}^{\infty} \Psi_{n+1}^{(a)} \frac{(-1)^a [\![\phi]\!]^a}{a!} \tag{4.25}$$

where $\Psi_n^{(a)}$ is the a-th derivative of Ψ at t_n, such that

$$\Psi_n^{(a)} = \frac{d^a \Psi(\phi(t_n))}{d\phi^a}.$$

and $\Psi_{n+1}^{(a)}$ is the a-th derivative of Ψ at t_{n+1}, such that

$$\Psi_{n+1}^{(a)} = \frac{d^a \Psi(\phi(t_{n+1}))}{d\phi^a}.$$

The jump in Ψ can then be equivalently expressed as:

$$[\![\Psi]\!] = \Psi_{n+1} - \Psi_n = \sum_{a=1}^{\infty} \Psi_n^{(a)} \frac{[\![\phi]\!]^a}{a!} \tag{4.26}$$

$$= \sum_{a=1}^{\infty} \Psi_{n+1}^{(a)} \frac{(-1)^{a+1} [\![\phi]\!]^a}{a!} \tag{4.27}$$

4.2.4.1 Implicit expansions

Depending on how many terms of the expansions given in equations (4.26) and (4.27) are kept, different approximations can be obtained. Given the definition of Ψ in equation (4.3), $[\![\Psi]\!]$ can be expanded from t_{n+1} using equation (4.27) such that

$$[\![\Psi]\!] = \Psi'_{n+1}[\![\phi]\!] - \Psi''_{n+1}\frac{[\![\phi]\!]^2}{2} + \Psi'''_{n+1}\frac{[\![\phi]\!]^3}{6} - \Psi^{iv}_{n+\epsilon}\frac{[\![\phi]\!]^4}{24}, \tag{4.28}$$

where $\Psi^{iv}_{n+\epsilon}$ represents the exact remainder term [121] of the Taylor series, and is always greater than zero since $a_4 > 0$. Recalling the implicit approximation to the nonlinear function $\tilde{\Psi}'$ given by equation (4.11), $[\![\Psi]\!]$ can be expressed as

$$[\![\Psi]\!] = \tilde{\Psi}'[\![\phi]\!] - \Psi^{iv}_{n+\epsilon}\frac{[\![\phi]\!]^4}{24}.$$

Thus, if equation (4.11) is used to express Ψ' as an approximation to $[\![\Psi]\!]/[\![\phi]\!]$, the jump in free energy of equation (4.23) becomes

$$\begin{aligned}[\![\mathcal{F}(\phi)]\!] &= -\Delta t\left(\nabla^a\sigma, \tilde{M}\nabla^a\sigma\right)_\Omega - \alpha\Delta t\left(\nabla^a[\![\phi]\!], \nabla^a[\![\phi]\!]\right)_\Omega - \int_\Omega \Psi^{iv}_{n+\epsilon}\frac{[\![\phi]\!]^4}{24}d\Omega \\ &\leq 0,\end{aligned}$$

and energy stability is proven. Thus, we see that for the fully implicit expansion no stabilization is needed, and $\alpha = 0$.

4.2.4.2 Explicit expansions

To analyze the stability of this scheme, we now consider the explicit approximation to Ψ defined in (4.26) with Lagrange remainder, more specifically, we let equation (4.12)

be an approximation to $\tilde{\Psi}'$. Thus, we have

$$\begin{aligned}\llbracket\Psi\rrbracket &= \left(\Psi'_n + \Psi''_n\frac{\llbracket\phi\rrbracket}{2}\right)\llbracket\phi\rrbracket + \Psi'''_n\frac{\llbracket\phi\rrbracket^3}{6} + \Psi^{iv}_{n+\epsilon_1}\frac{\llbracket\phi\rrbracket^4}{24}\\ &= \tilde{\Psi}'\llbracket\phi\rrbracket + \Psi'''_n\frac{\llbracket\phi\rrbracket^3}{6} + \Psi^{iv}_{n+\epsilon_1}\frac{\llbracket\phi\rrbracket^4}{24}.\end{aligned}$$

Substituting this result in equation (4.23), we have that

$$\begin{aligned}\llbracket\mathcal{F}(\phi)\rrbracket &= -\Delta t\left(\nabla^a\sigma, \tilde{M}\nabla^a\sigma\right)_\Omega - \alpha\Delta t\left(\nabla^a\llbracket\phi\rrbracket, \nabla^a\llbracket\phi\rrbracket\right)_\Omega\\ &\quad + \int_\Omega\left(\Psi'''_n\frac{\llbracket\phi\rrbracket^3}{6} + \Psi^{iv}\frac{\llbracket\phi\rrbracket^4}{24}\right)d\Omega\\ &= -\Delta t\left(\nabla^a\sigma, \tilde{M}\nabla^a\sigma\right)_\Omega - \alpha\Delta t\left(\nabla^a\llbracket\phi\rrbracket, \nabla^a\llbracket\phi\rrbracket\right)_\Omega\\ &\quad + \int_\Omega\llbracket\phi\rrbracket^2\left(\Psi'''_n\frac{\llbracket\phi\rrbracket}{6} + \Psi^{iv}\frac{\llbracket\phi\rrbracket^2}{24}\right)d\Omega. \qquad (4.29)\end{aligned}$$

Using the fact that $\Psi'''_n = 6a_4\phi_n$, $\Psi^{iv} = 6a_4$ and that $\phi \in \left]-\sqrt{a_2/a_4}, \sqrt{a_2/a_4}\right[$, we can rewrite the last two terms in equation (4.29) such that

$$\begin{aligned}\sup\left(\Psi'''_n\frac{\llbracket\phi\rrbracket}{6} + \Psi^{iv}\frac{\llbracket\phi\rrbracket^2}{24}\right) &= a_4\sup\left(\phi_n\llbracket\phi\rrbracket + \frac{\llbracket\phi\rrbracket^2}{4}\right)\\ &\leq \frac{a_2}{3}.\end{aligned}$$

Applying the previous result to equation (4.29), recalling that $[\![\phi]\!] = \Delta t(-\nabla)^a \cdot (-M\nabla^a\sigma)$, integrating by parts, and using the Cauchy–Schwarz inequality, we obtain

$$\begin{aligned}
[\![\mathcal{F}(\phi)]\!] &= -\Delta t\left(\nabla^a\sigma, \tilde{M}\nabla^a\sigma\right)_\Omega - \alpha\Delta t\left(\nabla^a[\![\phi]\!], \nabla^a[\![\phi]\!]\right)_\Omega \\
&\quad + \int_\Omega [\![\phi]\!]^2\left(-\frac{[\![\phi]\!]^2}{24}\Psi^{iv} + \frac{[\![\phi]\!]}{6}\Psi_n'''\right)d\Omega \\
&\leq -\Delta t\left(\nabla^a\sigma, \tilde{M}\nabla^a\sigma\right)_\Omega - \alpha\Delta t\left(\nabla^a[\![\phi]\!], \nabla^a[\![\phi]\!]\right)_\Omega + \frac{a_2}{3}\int_\Omega [\![\phi]\!]^2 d\Omega \\
&\leq -\Delta t\left(\nabla^a\sigma, \tilde{M}\nabla^a\sigma\right)_\Omega - \alpha\Delta t\left(\nabla^a[\![\phi]\!], \nabla^a[\![\phi]\!]\right)_\Omega \\
&\quad - \frac{a_2}{3}\Delta t\int_\Omega \nabla^a[\![\phi]\!]\tilde{M}\nabla^a\sigma d\Omega \\
&\leq -\Delta t\left(\nabla^a\sigma, \tilde{M}\nabla^a\sigma\right)_\Omega - \alpha\Delta t\left(\nabla^a[\![\phi]\!], \nabla^a[\![\phi]\!]\right)_\Omega \\
&\quad + \frac{a_2}{3}\Delta t\sqrt{\int_\Omega \tilde{M}\left|\nabla^a[\![\phi]\!]\right|^2 d\Omega}\sqrt{\int_\Omega \tilde{M}\left|\nabla^a\sigma\right|^2 d\Omega}.
\end{aligned}$$

Using Young's inequality, $fg \leq \frac{1}{2}(\delta f^2 + \delta^{-1}g^2)$ with $\delta > 0$, as well as the fact that $\sup\left(\tilde{M}\right) = \bar{M}^{a_2}/_{a_4}$, we have that

$$\begin{aligned}
[\![\mathcal{F}(\phi)]\!] &\leq -\Delta t\left(\nabla^a\sigma, \tilde{M}\nabla^a\sigma\right)_\Omega - \alpha\Delta t\left(\nabla^a[\![\phi]\!], \nabla^a[\![\phi]\!]\right)_\Omega \\
&\quad + \frac{a_2\Delta t}{6}\left(\delta\sup\left(\tilde{M}\right)\int_\Omega \left|\nabla^a[\![\phi]\!]\right|^2 d\Omega + \delta^{-1}\int_\Omega \tilde{M}\left|\nabla^a\sigma\right|^2 d\Omega\right) \\
&= -\Delta t\left(1 - \frac{a_2\Delta t}{6\delta}\right)\left(\nabla^a\sigma, \tilde{M}\nabla^a\sigma\right)_\Omega \\
&\quad - \Delta t\left(\alpha - \frac{a_2\Delta t}{6}\sup\left(\tilde{M}\right)\delta\right)\left(\nabla^a[\![\phi]\!], \nabla^a[\![\phi]\!]\right)_\Omega.
\end{aligned}$$

Thus, taking

$$\frac{a_2\Delta t}{6} \leq \delta \quad \text{and} \quad \alpha \geq \sup\left(\tilde{M}\right)\left(\frac{a_2\Delta t}{6}\right)^2, \tag{4.30}$$

energy stability is guaranteed. Contrary to the scheme developed in [5], α is no longer a lagging indicator. The stabilization parameter can be calculated at each time step

as all quantities are known a priori.

4.2.5 Order of accuracy

As we did in section 3.4.2, we compare the method to the Crank–Nicolson scheme to obtain a bound on the local truncation error. We use a second order approximation to M, $\tilde{M}$, which is explicit and thus remains constant for each time step [92]. Therefore, for simplicity of presentation, we suppose that M is constant in this section. When applied to equation (4.4), the scheme reads

$$\begin{aligned}\frac{[\![\phi]\!]}{\Delta t} &= (-\nabla)^a \cdot \left(-M\nabla^a \frac{\delta \mathcal{F}\left(\{\phi\}\right)}{\delta \phi}\right), \\ &= (-\nabla)^a \cdot \left(-M\nabla^a \left(\frac{\delta f(\{\phi\})}{\delta \phi} + \frac{\partial \Psi(\{\phi\})}{\partial \phi}\right)\right),\end{aligned} \tag{4.31}$$

where

$$\frac{\delta f\{\phi\}}{\delta \phi} = \frac{\partial f_1(\{\phi\})}{\partial \phi} - \nabla \cdot \frac{\partial f_2(\{\phi\})}{\partial \nabla \phi} + \Delta \frac{\partial f_3(\{\phi\})}{\partial \Delta \phi}.$$

Substituting the discrete time solution $\{\phi\}$ by the time-continuous solution $\{\phi\left(t_n\right)\}$ into the above equation gives rise to the local truncation error. Indeed, we have

$$\begin{aligned}\frac{[\![\phi]\!]}{\Delta t} &= (-\nabla)^a \cdot \left(-M\nabla^a \frac{\delta \mathcal{F}\left(\{\phi\left(t_n\right)\}\right)}{\delta \phi}\right), \\ &= (-\nabla)^a \cdot \left(-M\nabla^a \left(\frac{\delta f\left(\{\phi(t_n)\}\right)}{\delta \phi} + \frac{\partial \Psi\left(\{\phi(t_n)\}\right)}{\partial \phi}\right)\right) + \tau\left(t_n\right),\end{aligned} \tag{4.32}$$

where $\tau\left(t_n\right)$ represents the global truncation error. Such a scheme gives a bound $\tau\left(t_n\right) \leq C\Delta t^2$ which can be shown using Taylor series.

4.2.5.1 Implicit expansions

To prove second order accuracy in time for our scheme, we compute the next time-step approximation via the scheme applied to the exact solution and compare the result to Taylor expansions. A similar procedure was performed in [92] in the context of the Cahn-Hilliard equations. Using equations (4.9)-(4.10), and reorganizing the splitting into one equation, we have that

$$\phi_{n+1} = \phi\left(t_n\right) + \Delta t \left(-\nabla\right)^a \cdot \Bigg(- M\nabla^a \Bigg(\frac{\delta f\left(\{\phi\left(t\right)\}\right)}{\delta\phi} - \alpha\Delta t \Delta [\![\phi(t)]\!] + \Psi'\left(\phi\left(t_{n+1}\right)\right) - \Psi''\left(\phi\left(t_{n+1}\right)\right)\frac{[\![\phi(t)]\!]}{2} + \Psi'''\left(\phi\left(t_{n+1}\right)\right)\frac{[\![\phi(t)]\!]^2}{6}\Bigg)\Bigg), \tag{4.33}$$

when the first three terms of the implicit approximation (4.11) are used, and $\{\phi(t)\}$ defines the Crank-Nicolson approximation

$$\{\phi(t)\} = \frac{\phi\left(t_{n+1}\right) + \phi\left(t_n\right)}{2} + \mathcal{O}\left(\Delta t^2\right).$$

If at least two terms are used in the implicit approximation of $\tilde{\Psi}'$, the rest of the terms are at least of order (Δt^2), such that equation (4.33) can be written as

$$\phi_{n+1} = \phi\left(t_n\right) + \Delta t \left(-\nabla\right)^a \cdot \Bigg(- M\nabla^a \Bigg(\frac{\delta f\left(\{\phi\left(t\right)\}\right)}{\delta\phi} - \alpha\Delta t \Delta [\![\phi(t)]\!] + \Psi'\left(\phi\left(t_{n+1}\right)\right) - \Psi''\left(\phi\left(t_{n+1}\right)\right)\frac{[\![\phi(t)]\!]}{2} + \mathcal{O}\left(\Delta t^2\right)\Bigg)\Bigg), \tag{4.34}$$

We expand $\Psi'\left(\phi\left(t_{n+1}\right)\right)$ such that

$$\begin{aligned}\Psi'\left(\phi\left(t_{n+1}\right)\right) &= \Psi'\left(\{\phi\left(t\right)\}\right) - \Psi''\left(\phi\left(t_{n+1}\right)\right)\left(\{\phi\left(t\right)\} - \phi\left(t_{n+1}\right)\right) + \mathcal{O}\left(\Delta t^2\right) \\ &= \Psi'\left(\{\phi\left(t\right)\}\right) + \Psi''\left(\phi\left(t_{n+1}\right)\right)\frac{[\![\phi(t)]\!]}{2} + \mathcal{O}\left(\Delta t^2\right).\end{aligned}$$

Thus,

$$\Psi'\left(\{\phi\left(t\right)\}\right)=\Psi'\left(\phi\left(t_{n+1}\right)\right)-\Psi''\left(\phi\left(t_{n+1}\right)\right)\frac{[\![\phi(t)]\!]}{2}+\mathcal{O}\left(\Delta t^{2}\right). \tag{4.35}$$

The stabilization term is of order (Δt^2) and can be written as

$$\begin{aligned}\alpha\Delta t\Delta^{a}[\![\phi(t)]\!] &= \alpha\left(\Delta t\right)^{2}\Delta^{a}\left(\frac{[\![\phi(t)]\!]}{\Delta t}\right)\\ &= \alpha\left(\Delta t\right)^{2}\Delta^{a}\left(\frac{\partial\phi}{\partial t}+\mathcal{O}\left(\Delta t\right)\right)\\ &= \mathcal{O}\left(\Delta t^{2}\right).\end{aligned} \tag{4.36}$$

Using equations (4.35)-(4.36), and substituting them into (4.34), we obtain

$$\begin{aligned}\phi_{n+1} &= \phi\left(t_{n}\right)\\ &\quad+\Delta t\left(-\nabla\right)^{a}\cdot\left(-M\nabla^{a}\left(\frac{\delta f\left(\{\phi\left(t\right)\}\right)}{\delta\phi}+\Psi'\left(\{\phi\left(t\right)\}\right)+\mathcal{O}\left(\Delta t^{2}\right)\right)\right),\end{aligned} \tag{4.37}$$

Alternatively, by Taylor expansion of the solution we have

$$\begin{aligned}\{\phi\left(t\right)\} &= \phi\left(t_{n+1}\right)-\Psi'\left(\{\phi(t)\}\right)\left(\frac{\Delta t}{2}\right)-\Psi''\left(\{\phi(t)\}\right)\frac{1}{2}\left(\frac{\Delta t}{2}\right)+\mathcal{O}\left(\Delta t^{3}\right),\\ \{\phi\left(t\right)\} &= \phi\left(t_{n}\right)+\Psi'(\{\phi\left(t\right)\})\left(\frac{\Delta t}{2}\right)-\Psi''\left(\{\phi(t)\}\right)\frac{1}{2}\left(\frac{\Delta t}{2}\right)+\mathcal{O}\left(\Delta t^{3}\right).\end{aligned}$$

Taking the difference of the above two equations and using equation (4.4) yields

$$\begin{aligned}\phi\left(t_{n+1}\right)-\phi\left(t_{n}\right) &= \Delta t\frac{\partial\{\phi\left(t\right)\}}{\partial t}+\mathcal{O}\left(\Delta t^{3}\right)\\ &= \Delta t\left(-\nabla\right)^{a}\cdot\left(-M\nabla^{a}\left(\frac{\delta f\left(\{\phi(t)\}\right)}{\delta\phi}+\Psi'\left(\{\phi(t)\}\right)+\mathcal{O}\left(\Delta t^{3}\right)\right)\right),\end{aligned}$$

Finally, taking the difference of the above expression with equation (4.37), we obtain the local truncation error

$$\phi\left(t_{n+1}\right)-\phi_{n+1}=\mathcal{O}\left(\Delta t^{3}\right). \tag{4.38}$$

As the global truncation error $\tau\left(t_{n}\right)$ loses an order of accuracy of Δt, the scheme is second-order accurate in time.

4.2.5.2 Explicit expansions

Similarly, for the case of the explicit expansion of Ψ', keeping the first two terms of the approximation (4.12), we have that

$$\begin{aligned}\phi_{n+1}=\phi\left(t_{n}\right)+\Delta t\left(-\nabla\right)^{a}\cdot\Bigg(-M\nabla^{a}\Bigg(&\frac{\delta f\left(\{\phi(t)\}\right)}{\delta\phi}-\alpha\Delta t\Delta[\![\phi(t)]\!]\\&+\Psi'\left(\phi\left(t_{n}\right)\right)+\Psi''\left(\phi\left(t_{n}\right)\right)\frac{[\![\phi(t)]\!]}{2}\Bigg)\Bigg),\end{aligned} \tag{4.39}$$

We expand $\Psi'\left(\{\phi(t)\}\right)$ such that

$$\begin{aligned}\Psi'\left(\{\phi\left(t\right)\}\right)&=\Psi'\left(\phi\left(t_{n}\right)\right)+\Psi''\left(\phi\left(t_{n}\right)\right)\left(\{\phi(t)\}-\phi\left(t_{n}\right)\right)+\mathcal{O}\left(\Delta t^{2}\right).\\&=\Psi'\left(\phi\left(t_{n}\right)\right)+\Psi''\left(\phi\left(t_{n}\right)\right)\frac{[\![\phi(t)]\!]}{2}+\mathcal{O}\left(\Delta t^{2}\right).\end{aligned} \tag{4.40}$$

Following the procedure described previously through equation (4.36)-(4.38), second-order accuracy in time is also proved for the explicit expansion.

4.2.6 Conserving mass when $a=1$

Mass conservation is verified for the case in which $a=1$ by taking the time-discrete version of equation (4.15) and letting the test function w be equal to one while having

test function q equal to zero, such that

$$0 = \left(1, \frac{[\![\phi]\!]}{\Delta t}\right)_\Omega + (0, M\nabla^a\sigma)_\Omega = \int_\Omega \frac{[\![\phi]\!]}{\Delta t} \mathrm{d}\Omega,$$

which implies that mass is conserved at the discrete time levels, that is

$$\int_\Omega \phi_{n+1} \mathrm{d}\Omega = \int_\Omega \phi_n \mathrm{d}\Omega.$$

4.3 Numerical discretization and time-adaptivity

In this section, we discuss the spatial discretization used to solve equations (4.18)-(4.19). We also introduce an adaptive algorithm in time that can be applied together with the stable schemes proposed. This coupling decreases the computational time needed to reach steady state solutions, and guarantees free energy stability during the simulation.

4.3.1 Spatial discretization: semi-discrete formulation

Given the weak form presented in equations (4.18)-(4.19), H^2-conforming spaces are needed if the equation is solved with the Galerkin method. This requirement can be fulfilled by using $\mathcal{C}^1$ finite elements. These finite elements are accessible through isogeometric analysis [48]. Even though the equations addressed in this work are not required to go above $m = 2$, it is possible to generate discrete spaces with a higher degree of continuity [52]. As $m = 2$ allows us to tackle all the problems dealt with in this work, we let $\mathcal{V}_2^h \subset \mathcal{V}_2$ denote the finite dimensional functional space in two or three dimensions. The problem can be stated as follows: find ϕ^h, $\sigma^h \in \mathcal{V}_2^h$ such that

for all w^h, $q^h \in \mathcal{V}_2^h$

$$0 = \left(w^h, \frac{\partial \phi^h}{\partial t} \right)_\Omega + \left(\nabla^a w, M\left(\phi^h\right) \nabla^a \sigma^h \right)_\Omega \tag{4.41}$$

$$\begin{aligned} 0 = & \left(q^h, \sigma^h - \Psi'\left(\phi^h\right) \right)_\Omega \\ & - \left(q^h, \frac{\partial f_1(\phi^h)}{\partial \phi^h} \right)_\Omega - \left(\nabla q^h, \frac{\partial f_2(\phi^h)}{\partial \nabla \phi^h} \right)_\Omega - \left(\Delta q^h, \frac{\partial f_3(\phi^h)}{\partial \Delta \phi^h} \right)_\Omega \end{aligned} \tag{4.42}$$

where the weighting functions w^h and q^h, trial solutions ϕ^h and σ^h, and their respective gradients and Laplacians, are defined as the linear combinations

$$w^h = \sum_A W_A N_A(\boldsymbol{x}), \qquad \sigma^h = \sum_A \Sigma_A N_A(\boldsymbol{x}), \tag{4.43}$$

$$\nabla w^h = \sum_A W_A \nabla N_A(\boldsymbol{x}), \qquad \nabla \sigma^h = \sum_A \Sigma_A \nabla N_A(\boldsymbol{x}), \tag{4.44}$$

$$q^h = \sum_A Q_A N_A(\boldsymbol{x}), \qquad \phi^h = \sum_A \Phi_A N_A(\boldsymbol{x}), \tag{4.45}$$

$$\nabla q^h = \sum_A Q_A \nabla N_A(\boldsymbol{x}), \qquad \nabla \phi^h = \sum_A \Phi_A \nabla N_A(\boldsymbol{x}), \tag{4.46}$$

$$\Delta q^h = \sum_A Q_A \Delta N_A(\boldsymbol{x}), \qquad \Delta \phi^h = \sum_A \Phi_A \Delta N_A(\boldsymbol{x}), \tag{4.47}$$

where N_A are the multidimensional basis functions and W_A, Q_A, Σ_A and Φ_A are the control variables.

Remark 6. *Linear finite elements can be used in phase-field equations where the Laplacian term is absent from the free energy functional (i.e., $c_2 = 0$ in equation (4.2)). Furthermore, the choice to use $\mathcal{C}^1$ finite elements simplifies the presentation of the method, but alternative mixed formulations that only require $\mathcal{C}^0$ finite elements exist [5].*

4.3.2 Fully discrete scheme

The fully discrete version of the general phase-field problem presented in equations (4.13)-(4.14) may be described as follows: Let the time interval $\mathcal{T} =]0, T[$ be divided into r subintervals $\mathcal{T}_n = (t_n, t_{n+1})$ for $n = 0, ..., r-1$. The fully discrete solutions are defined as ϕ_n^h and σ_n^h. The problem can now be defined as follows: given ϕ_{n-1}^h, ϕ_n^h and σ_n^h, find ϕ_{n+1}^h and σ_{n+1}^h such that for all $w^h, q^h \in \mathcal{V}_2^h$

$$0 = \left(w^h, \frac{[\![\phi]\!]}{\Delta t} \right)_\Omega + \left(\nabla^a w^h, \tilde{M}\left(\phi^h\right) \nabla^a \sigma_{n+1}^h \right)_\Omega, \tag{4.48}$$

$$\begin{aligned} 0 = &\left(q^h, \sigma_{n+1}^h - \tilde{\Psi}'\left(\phi^h\right) \right)_\Omega - \alpha \Delta t \Big(\nabla^a q^h, \nabla^a [\![\phi^h]\!] \Big)_\Omega \\ &- \left(q^h, \frac{\partial f_1\left(\{\phi^h\}\right)}{\partial \phi^h} \right)_\Omega - \left(\nabla q, \frac{\partial f_2\left(\{\phi^h\}\right)}{\partial \nabla \phi^h} \right)_\Omega - \left(\Delta q^h, \frac{\partial f_3\left(\{\phi^h\}\right)}{\partial \Delta \phi^h} \right)_\Omega, \end{aligned} \tag{4.49}$$

where $\tilde{\Psi}'\left(\phi^h\right)$ represents the spatially-discrete version of equations (4.11)-(4.12), and $\tilde{M}\left(\phi^h\right) = M\left(\frac{3}{2}\phi_n^h - \frac{1}{2}\phi_{n-1}^h\right)$.

4.3.3 Numerical implementation

With regards to the numerical implementation, we let the global vectors of degrees of freedom associated to ϕ_n^h and σ_n^h be $\mathbf{\Phi}_n^h$ and $\mathbf{\Sigma}_n^h$, respectively. The residual vectors for this formulation are then given by

$$\mathbf{R}_\phi\left(\mathbf{\Phi}_{n-1}, \mathbf{\Phi}_n, \mathbf{\Phi}_{n+1}, \mathbf{\Sigma}_{n+1}\right); \quad \mathbf{R}_\phi = \{R_A^\phi\}; \quad A = 1, ..., n_b,$$

$$\mathbf{R}_\sigma\left(\mathbf{\Phi}_{n-1}, \mathbf{\Phi}_n, \mathbf{\Phi}_{n+1}, \mathbf{\Sigma}_{n+1}\right); \quad \mathbf{R}_\sigma = \{R_A^\sigma\}; \quad A = 1, ..., n_b,$$

where

$$R_A^{\phi} = \left(N_A, \frac{[\![\phi]\!]}{\Delta t}\right)_{\Omega} + \left(\nabla^a N_A, \tilde{M}\left(\phi^h\right)\nabla^a \sigma^h\right)_{\Omega},$$
$$R_A^{\sigma} = \left(N_A, \sigma^h - \tilde{\Psi}'\left(\phi^h\right)\right)_{\Omega} - \alpha\Delta t\left(\nabla^a N_A, \nabla^a [\![\phi^h]\!]\right)_{\Omega}$$
$$- \left(N_A, \frac{\partial f_1\left(\{\phi^h\}\right)}{\partial \phi^h}\right)_{\Omega} - \left(\nabla N_A, \frac{\partial f_2\left(\{\phi^h\}\right)}{\partial \nabla \phi^h}\right)_{\Omega} - \left(\Delta N_A, \frac{\partial f_3\left(\{\phi^h\}\right)}{\partial \Delta \phi^h}\right)_{\Omega}.$$

The resulting system of nonlinear equations for $\mathbf{\Phi}_{n+1}$ and $\mathbf{\Sigma}_{n+1}$ is solved using Newton's method, where $\mathbf{\Phi}_{n+1}^{(i)}$ and $\mathbf{\Sigma}_{n+1}^{(i)}$ correspond to the *i-th* iteration of Newton's algorithm. The iterative procedure is specified in Algorithm 2. The codes are implemented in PetIGA, a high-performance isogeometric analysis framework [56]. The codes can be found in the demo section of the repository[1], and are free to download.

4.3.4 Time adaptivity

Many processes modeled with phase-field models are controlled by different time scales as they evolve, and each regime may be orders of magnitude different from the other. Therefore, efficient numerical solutions for these problems usually involve adaptive time stepping schemes. Different strategies have already been proposed, such as the ones found in [43, 28] for the Allen–Cahn and Cahn–Hilliard equations and in [40] for the phase-field crystal equation. Even though the methods presented therein successfully decrease the computational time taken to reach steady state solutions, they still have room for considerable computational savings. In [28], a solution computed with the generalized-α method is compared against a solution obtained with the backward-Euler method. The difference between the two solutions is used as an error estimator to modify the time step size. Even though shown to be robust and used since in other works [2, 50, 122], this strategy is somewhat inefficient given that the solution must be computed twice at each time step. In [43, 40], no recovery strategies are proposed

[1] https://bitbucket.org/dalcinl/petiga

Algorithm 2 Iterative procedure to solve the mixed form

Taking $\mathbf{\Phi}_{n+1}^{(0)} = \mathbf{\Phi}_n$, $\mathbf{\Sigma}_{n+1}^{(0)} = \mathbf{\Sigma}_n$, for $i = 0, \ldots, \mathrm{i}_{max}$,

(1) Compute the residuals $\mathbf{R}_\phi^{(i)}$, $\mathbf{R}_\sigma^{(i)}$, using $\mathbf{\Phi}_{n+1}^{(i)}$, $\mathbf{\Sigma}_{n+1}^{(i)}$.
(2) Compute the Jacobian matrix $\mathbf{K}^{(i)}$ using the i-th iterates. This matrix is given by

$$\mathbf{K}^{(i)} = \begin{pmatrix} \mathbf{K}^{\phi\phi} & \mathbf{K}^{\phi\sigma} \\ \mathbf{K}^{\sigma\phi} & \mathbf{K}^{\sigma\sigma} \end{pmatrix}^{(i)}, \tag{4.50}$$

where the individual components of each submatrix of the Jacobian are defined in the Appendix in equations (C.1) through (C.4).
(3) Solve the linear system

$$\begin{pmatrix} \mathbf{K}^{\phi\phi} & \mathbf{K}^{\phi\sigma} \\ \mathbf{K}^{\sigma\phi} & \mathbf{K}^{\sigma\sigma} \end{pmatrix}^{(i)} \begin{pmatrix} \Delta\mathbf{\Phi} \\ \Delta\mathbf{\Sigma} \end{pmatrix}^{(i+1)} = \begin{pmatrix} \mathbf{R}_\phi \\ \mathbf{R}_\sigma \end{pmatrix}^{(i)}.$$

(4) Update the solution such that

$$\begin{pmatrix} \mathbf{\Phi}_{n+1} \\ \mathbf{\Sigma}_{n+1} \end{pmatrix}^{(i+1)} = \begin{pmatrix} \mathbf{\Phi}_{n+1} \\ \mathbf{\Sigma}_{n+1} \end{pmatrix}^{(i)} - \begin{pmatrix} \Delta\mathbf{\Phi} \\ \Delta\mathbf{\Sigma} \end{pmatrix}^{(i+1)}.$$

Steps (1) through (4) are repeated until the norms of the global residual vector are reduced to a certain tolerance (10^{-8} in all the examples shown in this work) of their initial value. Convergence is usually achieved in 2 or 3 nonlinear iterations per time step. The linear system is solved with GMRES, to a specified tolerance (10^{-5} in this work) with a restart criterion of 100 (i.e., the total number of Krylov directions to orthogonalize against is set to 100; once this limit is reached, the Krylov space is restarted from zero).

when the numerical solver fails. This implies a period of trial and error is necessary to tune the solver parameters for the specific equation being solved. In the following, we propose an adaptive time-stepping strategy for any second order accurate time integration scheme. In particular, our stable methods. This time adaptive scheme addresses these issues, and is based on the generalized-α method. The method is presented in algorithm 3.

Rather than computing the solution twice as done in [28], the solutions at t_n and t_{n-1} can be stored and used to estimate a posteriori the local truncation error of a lower-order method once the solution at t_{n+1} is obtained with the second-order accurate generalized-α method (see Appendix C.1.2). The estimation of the local truncation error is done through variable step-size backward differentiation. Borrowing techniques from embedded Runge–Kutta methods [123], we consider the second-order accurate generalized-α method as a high-order scheme, and the first-order backward-Euler method as a lower-order scheme. Our time-adaptive scheme is based on controlling the local truncation error for the lower-order method, i.e., the backward-Euler method. By using Taylor series expansions, it can be shown [124] that the local truncation error of the backward-Euler method is

$$\tau^{BE}\left(t_{n+1}\right) = -\frac{\Delta t^2}{2}\mathbf{U}''\left(t_{n+1}\right) + \mathcal{O}\left(\Delta t^3\right). \tag{4.51}$$

Given the approximate solutions $\mathbf{U}_{n+1}$, $\mathbf{U}_n$ and $\mathbf{U}_{n-1}$, at times t_{n+1}, t_n and t_{n-1}, respectively, we can express the scaled second derivative $1/2\Delta t^2\mathbf{U}''\left(t_{n+1}\right)$ using the second order backward difference formula

$$\frac{\Delta t^2}{2}\mathbf{U}''\left(t_{n+1}\right) = \frac{1}{\eta}\mathbf{U}_{n+1} - \frac{1}{\eta-1}\mathbf{U}_n + \frac{1}{\eta(\eta-1)}\mathbf{U}_{n-1} + \mathcal{O}\left(\Delta t^3\right), \tag{4.52}$$

where $\eta = (\Delta t + \Delta t_p)/\Delta t = 1 + \Delta t_p/\Delta t$, $\Delta t = t_{n+1} - t_n$ and $\Delta t_p = t_n - t_{n-1}$.

From equations (4.51)-(4.52) the truncation error can be conveniently estimated

as

$$\mathbf{E}_{n+1} = -\frac{1}{\eta}\mathbf{U}_{n+1} + \frac{1}{\eta-1}\mathbf{U}_n - \frac{1}{\eta(\eta-1)}\mathbf{U}_{n-1} \approx \tau^{BE}\left(t_{n+1}\right).$$

Then, we can calculate a weighted local truncation error (WLTE) [123, equation 4.11] as

$$\text{WLTE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{E_i}{\tau_i^{\text{abs}} + \tau_i^{\text{rel}}\max\left(|U_i|, |U_i + E_i|\right)}\right)^2}, \tag{4.53}$$

where $E_i = (\mathbf{E}_{n+1})_i$, $U_i = (\mathbf{U}_{n+1})_i$, τ_i^{abs} and τ_i^{rel} define tunable absolute and relative tolerances, respectively. The WLTE can be used to control the error at each time step, by updating the time step size through

$$\Delta t_{\text{next}} = \rho\ \text{WLTE}^{-1/2}\ \Delta t,$$

with ρ a safety factor. The default values for the safety factor ρ, the absolute and relative tolerances τ_i^{abs} and τ_i^{rel} are set to 0.9, 10^{-4} and 10^{-4}, respectively.

An initial time derivative $\mathbf{V}_0$ needs to be provided given the non-self-starting nature of the algorithm. By using a finite difference strategy, a reasonable initial time derivative can be estimated (i.e., $\mathcal{O}(\Delta t^2)$). To do this, the backward-Euler method on the system $\frac{d\mathbf{U}}{dt} = F(t, \mathbf{U})$ is used to estimate the value $\mathbf{U}^*_{1/2}$ at time $t_{1/2} = t_0 + {}^1\!/_2\Delta t$

$$\frac{\mathbf{U}^*_{1/2} - \mathbf{U}_0}{\Delta t/2} = F\left(\Delta t/2, \mathbf{U}^*_{1/2}\right).$$

Using the approximation to $\mathbf{U}^*_{1/2}$, a second backward-Euler solve is done to get the value $\mathbf{U}^*_1$ at time $t_1 = t_0 + \Delta t$

$$\frac{\mathbf{U}^*_1 - \mathbf{U}^*_{1/2}}{\Delta t/2} = F\left(\Delta t, \mathbf{U}^*_1\right).$$

Given $\mathbf{U}_0$, $\mathbf{U}^*_{1/2}$, $\mathbf{U}^*_1$, the initial time derivative can be approximated as

$$\mathbf{V}_0 = \frac{1}{\Delta t}\left(-3\mathbf{U}_0 + 4\mathbf{U}^*_{1/2} - \mathbf{U}^*_1\right). \tag{4.54}$$

Now, given a solution $\mathbf{U}_0$ at time t_0, it is possible to compute a solution $\mathbf{U}^\alpha_1$ at $t_1 = t_0 + \Delta t$ using the standard generalized-α method [125] for first-order systems.

Algorithm 3 Pseudocode for adaptive time-stepping.
$k_{\max} = 10$, $k_{\min} = 0.1$, $\rho = 0.9$.

1: Compute solution $\mathbf{U}^\alpha_{n+1}$ using Δt through algorithm 2
2: **if** nonlinear solve failed to converge **then**
3: $k = 0.25$, update time step size $\Delta t \Leftarrow k\Delta t$
4: **goto** 1
5: **end if**
6: Calculate local truncation error $\mathbf{E}_{n+1}$ using equation (4.51)
7: Calculate WLTE using equation (4.53)
8: Compute $k_{lte} = \rho \mathrm{WLTE}^{-1/2}$
9: Set $k = \max\big(k_{\min}, \min\left(k_{lte}, k_{\max}\right)\big)$
10: **if** WLTE > 1 **then**
11: Reject step, update time step size $\Delta t \Leftarrow k\Delta t$
12: **goto** 1
13: **end if**
14: Accept step, update time step size $\Delta t \Leftarrow k\Delta t$

The same strategy developed here for the generalized-α method can be applied to any second-order accurate scheme, and, in particular, to both methods proposed in section 4.2.2.

4.4 Phase-field models: weak forms

In this section, we focus on the four phase-field models we described in Chapter 2, that have had considerable success in modeling different phenomena [14, 1, 12]. These are the Allen–Cahn [65], the Cahn–Hilliard [27], the Swift–Hohenberg [66] and the phase-field crystal [30] equations. The Allen–Cahn and Cahn–Hilliard equations, two of the most successful phase-field models to date, are derived as gradient flows of the

same free energy functional [43]. The same statement can be made regarding the Swift–Hohenberg and the phase-field crystal equation, which are also derived from the same free energy functional.

These equations were selected because their stable time integration has garnered considerable interest in recent years [78, 2, 4, 5], and this work generalizes some of the numerical schemes that have been put forth in the context of the Cahn-Hilliard [82, 43] and phase-field crystal equations [5]. Additionally, the discretization in space is done using isogeometric analysis [48], which allows to easily generate high-order and globally continuous basis functions. Given the higher-order nature of the equations and the complicated geometries that can arise in phase-field problems [126], a finite element discretization with improved geometrical properties [127] which allows for a straightforward discretization of the high-order differential operators, seems to be the most flexible and robust choice among the myriad of methods available.

4.4.1 The Allen–Cahn equation

The Allen–Cahn equation governs the behavior of the Ising ferromagnet [13], where the competition between energy and entropy gives way to a phase transition below some critical temperature [65]. The free energy functional for the system is given by

$$\mathcal{F}_{AC} = \int_{\Omega} \left(\Psi_{AC} + \frac{1}{2} \left| \nabla \phi \right|^2 \right) d\Omega,$$

where $\Psi_{AC} = \dfrac{1}{4\epsilon^2} \left(\phi^2 - 1 \right)^2$ and ϵ is a parameter related to interface thickness. The phase-field parameter is related to the ordering of the phase, which translates to it being a non-conserved quantity. The partial differential equation that describes this

order-disorder transition is given by

$$\frac{\partial \phi}{\partial t} = -\sigma,$$
$$\sigma = \Psi'_{AC}(\phi) - \Delta\phi,$$

where a constant mobility of value one is considered. The weak formulation, following equations (4.18)-(4.19), is given by

$$0 = \left(w, \frac{\partial \phi}{\partial t} + \sigma \right)_\Omega, \tag{4.55}$$
$$0 = \left(q, \sigma - \Psi'_{AC}(\phi) \right)_\Omega - \left(\nabla w, \nabla \phi \right)_\Omega. \tag{4.56}$$

The time discretization, using the stable scheme proposed in section 4.2.2, becomes

$$0 = \left(w, \frac{\partial \phi}{\partial t} + \sigma \right)_\Omega,$$
$$0 = \left(q, \sigma - \tilde{\Psi}'_{AC}(\phi) \right)_\Omega - \left(\nabla w, \nabla \{\phi\} \right)_\Omega - \alpha_{AC} \Delta t \, \left(q, [\![\phi]\!] \right)_\Omega.$$

4.4.2 The Cahn–Hilliard equation

The Cahn–Hilliard equation governs the evolution of an immiscible binary mixture undergoing phase separation [27]. This is the equation that popularized the use of the phase-field method [14]. In this equation, the phase field represents the concentration of one of the components of the mixture. The dimensionless free energy functional is given by

$$\mathcal{F}_{CH} = \mathcal{F}_{AC}$$
$$= \int_\Omega \left(\Psi_{CH} + \frac{1}{2} |\nabla \phi|^2 \right) d\Omega \tag{4.57}$$

where $\Psi_{CH} = \frac{1}{4\epsilon^2}\left(\phi^2 - 1\right)^2$. The parameter ϵ is related to the thickness of the interface. The partial differential equation, given that the phase-field is in this case a conserved quantity, is given by

$$\begin{aligned}\frac{\partial \phi}{\partial t} &= \nabla \cdot \left(M_{CH} \nabla \sigma\right), \\ \sigma &= \Psi'_{CH}\left(\phi\right) - \Delta\phi,\end{aligned}$$

with $M_{CH} = \bar{M}\left(1 - \phi^2\right)$. The variational formulation is then given by

$$0 = \left(w, \frac{\partial \phi}{\partial t}\right)_\Omega + \left(\nabla w, M_{CH} \nabla \sigma\right)_\Omega, \tag{4.58}$$

$$0 = \left(q, \sigma - \Psi'_{CH}(\phi)\right)_\Omega - \left(\nabla w, \nabla \phi\right)_\Omega. \tag{4.59}$$

The time discretization, using the stable scheme proposed, becomes

$$\begin{aligned}0 &= \left(w, \frac{\partial \phi}{\partial t}\right)_\Omega + \left(\nabla w, \tilde{M}_{CH} \nabla \sigma\right)_\Omega, \\ 0 &= \left(q, \sigma - \tilde{\Psi}'_{CH}(\phi)\right)_\Omega - \left(\nabla w, \nabla \{\phi\}\right)_\Omega - \alpha_{CH} \Delta t \, \left(\nabla q, \nabla [\![\phi]\!]\right)_\Omega,\end{aligned}$$

where the nonlinear mobility $\tilde{M}_{CH}$ is defined using the extrapolation method presented in section 4.2.2, taken from [92].

4.4.3 The Swift–Hohenberg equation

The Swift–Hohenberg equation is derived from a phenomenon known as Rayleigh–Bénard convection, in which a fluid is trapped between a hot and a cold plate. The equation describes a convective instability, which occurs when the difference in temperature between the plates becomes large enough. The Swift–Hohenberg equation, a fourth order, nonlinear partial differential equation, is derived as a gradient flow of

the dimensionless free energy functional $\mathcal{F}_{SH}$ defined by [66, 4]

$$\mathcal{F}_{SH} = \int_{\Omega} \left[\frac{1}{2} \left(\phi^2 - 2|\nabla \phi|^2 + (\Delta \phi)^2 \right) + \Psi_{SH}(\phi) \right] d\Omega, \tag{4.60}$$

where $\Psi_{SH}(\phi) = -\frac{\epsilon}{2}\phi^2 + \frac{1}{4}\phi^4$. Using equations (4.4), (4.13), (4.14) along with the definition of free energy given by equation (4.60) defines the evolution in time of ϕ

$$\frac{\partial \phi}{\partial t} = -\sigma, \tag{4.61}$$

$$\sigma = (1 + \Delta)^2 \phi + \Psi'_{SH}(\phi), \tag{4.62}$$

where $M_{SH} = \bar{M}_{SH} = 1$, $(1 + \Delta)^2 = 1 + 2\Delta + \Delta\Delta$, Ψ'_{SH} is defined as $\Psi'_{SH}(\phi) = \frac{\partial \Psi(\phi)}{\partial \phi} = \phi^3 - \epsilon\phi$, as well as the fact that the order parameter ϕ is not a conserved quantity in the case of this equation such that $a = 0$. The corresponding weak form, following equations (4.18)-(4.19), is given by

$$0 = \left(w, \frac{\partial \phi}{\partial t} + \sigma \right)_{\Omega}, \tag{4.63}$$

$$0 = \left(q, \sigma - \phi - \Psi'_{SH}(\phi) \right)_{\Omega} + 2\left(\nabla w, \nabla \phi \right)_{\Omega} - \left(\Delta w, \Delta \phi \right)_{\Omega}. \tag{4.64}$$

The time discretization, following section 4.2.2, is given by

$$\begin{aligned} 0 &= \left(w, \frac{[\![\phi]\!]}{\Delta t} + \sigma \right)_{\Omega}, \\ 0 &= \left(q, \sigma - \{\phi\} - \tilde{\Psi}'_{SH} \right)_{\Omega} + 2\left(\nabla q, \nabla \{\phi\} \right)_{\Omega} - \left(\Delta q, \Delta \{\phi\} \right)_{\Omega} \\ &\quad - \alpha_{SH} \Delta t \ (q, [\![\phi]\!])_{\Omega}. \end{aligned}$$

where $\tilde{\Psi}'_{SH}$ represents an approximation to $\Psi'_{SH}(\phi)$, as defined in equations (4.11) and (4.12).

4.4.4 The phase-field crystal equation

The phase-field crystal equation is used to model the evolution of microstructures at atomistic length scales and diffusive time scales. It is a sixth-order, nonlinear partial differential equation. It can be derived by considering a conservative description of the Rayleigh Bénard convection problem [30, 14]. It uses the same dimensionless free energy functional as the Swift–Hohenberg equation, such that

$$\begin{aligned}\mathcal{F}_{PFC} &= \mathcal{F}_{SH} \\ &= \int_\Omega \left[\frac{1}{2}\left(\phi^2 - 2|\nabla\phi|^2 + (\Delta\phi)^2\right) + \Psi_{PFC}(\phi)\right] d\Omega,\end{aligned}$$

with $\Psi_{PFC}(\phi) = -\dfrac{\epsilon}{2}\phi^2 + \dfrac{1}{4}\phi^4$. The evolution equation of the phase field ϕ, which now represents an atomistic density field that is a conserved quantity, is given by

$$\begin{aligned}\frac{\partial\phi}{\partial t} &= \Delta\sigma, \\ \sigma &= (1+\Delta)^2\,\phi + \Psi'_{PFC}\,(\phi)\,,\end{aligned}$$

where the mobility $M = 1$. This leads to the variational formulation

$$0 = \left(w, \frac{\partial\phi}{\partial t}\right)_\Omega + (\nabla w, \nabla\sigma)_\Omega\,, \tag{4.65}$$

$$0 = \big(q, \sigma - \phi - \Psi'_{PFC}(\phi)\big)_\Omega + 2\big(\nabla q, \nabla\phi\big)_\Omega - \big(\Delta q, \Delta\phi\big)_\Omega. \tag{4.66}$$

with a time discretization given by

$$\begin{aligned}0 &= \left(w, \frac{[\![\phi]\!]}{\Delta t}\right)_\Omega + (\nabla w, \nabla\sigma)_\Omega\,, \\ 0 &= \Big(q, \sigma - \{\phi\} - \tilde{\Psi}'_{PFC}\Big)_\Omega + 2\big(\nabla q, \nabla\{\phi\}\big)_\Omega - (\Delta q, \Delta\{\phi\})_\Omega \\ &\quad - \alpha_{PFC}\Delta t\ \big(\nabla q, \nabla[\![\phi]\!]\big)_\Omega.\end{aligned}$$

4.5 Numerical examples

In this section, we study the temporal approximation properties of the algorithms resulting from the implicit and explicit expansions of Ψ' presented in section 4.2.2. We compare them with state-of-the-art algorithms developed by Gomez et al. for the equations presented in section 4.4, as well as the backward-Euler scheme and the generalized-α method [125] (using a spectral radius $\rho_\infty = 0.5$). As previously mentioned, the spatial discretization is done using isogeometric analysis, which allows to easily generate the non-standard basis functions needed in the case of the Swift–Hohenberg and phase-field crystal equations. The parameters for the equations that are solved are summarized in table 4.1.

To focus on temporal error only, we use a uniform mesh with $[512]^2$-$\mathcal{C}^1$ quadratic elements. This is an overkill spatial resolution for the examples solved in this work [128, 2, 4, 5]. An exception is made with regards to the Cahn–Hilliard equation, which uses $[64]^2$-$\mathcal{C}^1$-quadratic elements, as the degenerate mobility increases the stiffness of the system considerably. The solver requires a time step size on the order of $\Delta t = \mathcal{O}\left(10^{-10}\right)$ to be able to handle the fine mesh (i.e., $[512]^2$-$\mathcal{C}^1$ quadratic elements).

The initial conditions specified in the following are only defined for ϕ, not for the auxiliary variable σ presented through equation (4.14). To correctly initialize σ, a nonlinear L_2 projection is performed to solve equation (4.41), the semidiscrete version of equation (4.14).

Table 4.1: Summarizing the phase-field models from section 4.4 within the context of equations (4.9) and (4.10).

Eq.	$\tilde{M}$	$\Psi = \frac{1}{4}a_0 - \frac{1}{2}a_2\phi^2 + \frac{1}{4}a_4\phi^4$	ϵ	f	α	Mass conservation
AC	1	$a_0 = a_2 = a_4 = \frac{1}{\epsilon^2}$	0.01	$\frac{1}{2}\lvert\nabla\phi\rvert^2$	$\frac{9}{4}\Delta t^2$	No ($a = 0$)
CH	$\frac{1}{2}\left(1-\phi^2\right)$	$a_0 = a_2 = a_4 = \frac{1}{\epsilon^2}$	0.03	$\frac{1}{2}\lvert\nabla\phi\rvert^2$	$\frac{9}{4}\Delta t^2$	Yes ($a = 1$)
SH	1	$a_0 = 0, a_2 = \epsilon, a_4 = 1$	2	$\frac{1}{2}\left(\phi^2 - 2\lvert\nabla\phi\rvert^2 + (\Delta\phi)^2\right)$	$\frac{9}{4}(\epsilon\Delta t)^2$	No ($a = 0$)
PFC	1	$a_0 = 0, a_2 = \epsilon, a_4 = 1$	0.325	$\frac{1}{2}\left(\phi^2 - 2\lvert\nabla\phi\rvert^2 + (\Delta\phi)^2\right)$	$\frac{9}{4}(\epsilon\Delta t)^2$	Yes ($a = 1$)

4.5.1 Allen–Cahn equation

We simulate the evolution of a star-shaped interface in a curvature-driven flow [128]. The initial condition is given by

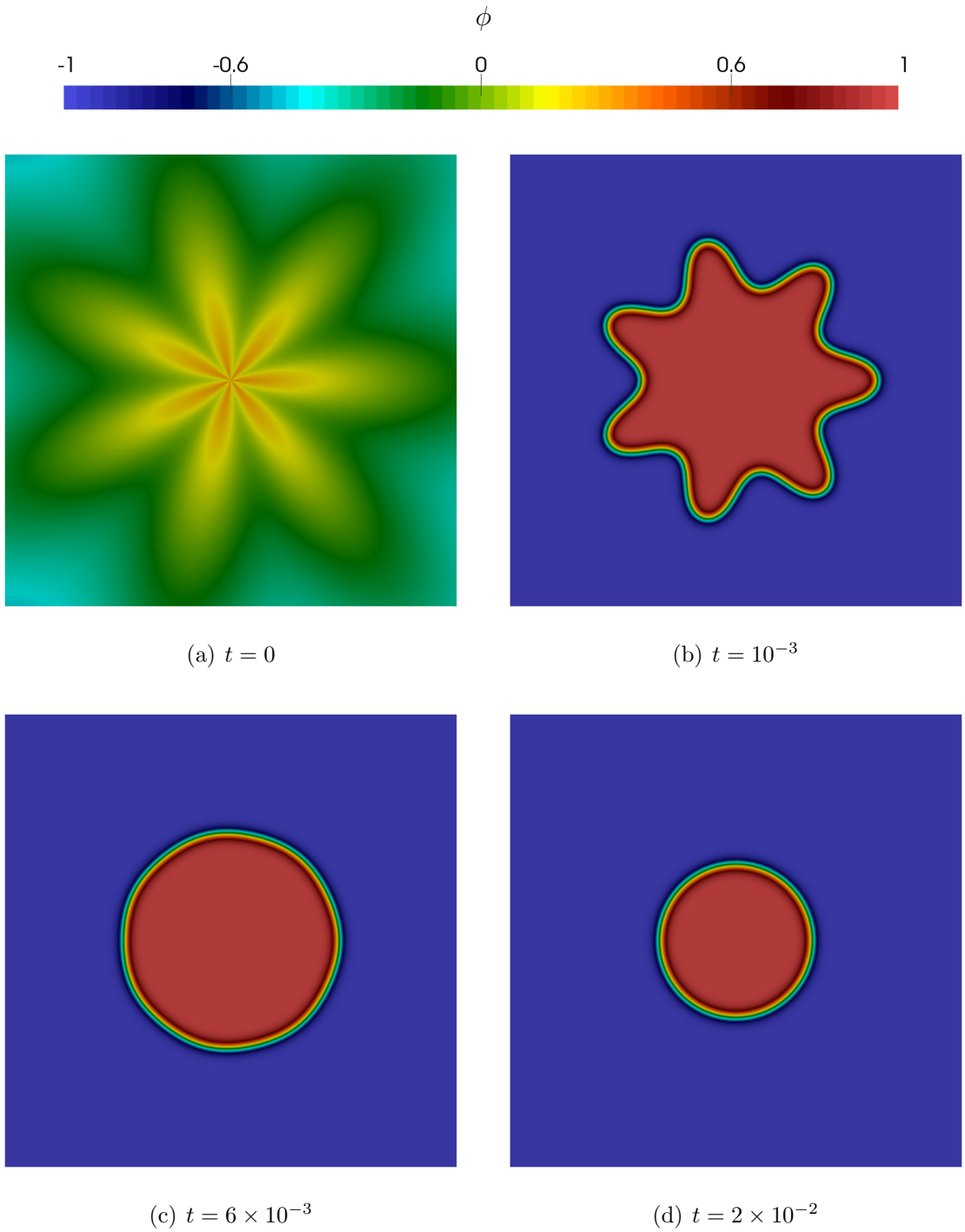
$$\phi_{\text{star}}(\mathbf{x}) = \tanh\left(\frac{1/4 + 0.1\cos(7\theta) - \sqrt{(x-0.5)^2 + (y-0.5)^2}}{(3/4)\sqrt{2}}\right), \tag{4.67}$$

where x and y represent Cartesian coordinates, θ is defined as

$$\theta = \begin{cases} \tan^{-1}\left(\dfrac{y-0.5}{x-0.5}\right) & \text{if } x > 0.5, \\ \pi + \tan^{-1}\left(\dfrac{y-0.5}{x-0.5}\right) & \text{otherwise,} \end{cases}$$

The computational domain is the unit square $\bar{\Omega} = [0,1]^2$, and no-flux boundary conditions are imposed. For the spatial discretization, we use a uniform mesh of $[512]^2$-$\mathcal{C}^1$-quadratic elements. The initial condition is shown in figure 4.1, along with snapshots of the solution calculated with an overkill temporal resolution ($\Delta t = 10^{-7}$). The tips of the star move inward, while the gaps between the tips move outward. Once the surface becomes circular, the radius of the circle shrinks with increasing speed due to the increasing curvature.

To evaluate the performance of the schemes developed in this work, we compare



(a) $t = 0$

(b) $t = 10^{-3}$

(c) $t = 6 \times 10^{-3}$

(d) $t = 2 \times 10^{-2}$

Figure 4.1: Evolution of a star-shaped interface in a curvature-driven flow using the Allen–Cahn equation. The initial condition prescribed by equation (4.67) is shown, as well as the solution evolution calculated with our implicit algorithm. The simulation is run in a unit square domain $\bar{\Omega} = [0, 1]^2$ using a computational mesh composed of 512×512 $\mathcal{C}^1$-quadratic elements, with a time step size $\Delta t = 10^{-7}$. We consider this our reference solution.

the the provably-stable implicit (Implicit) and explicit expansions (Linear) against the backward-Euler method (BE), the generalized-α method (Generalized-α) and a second-order, provably-stable, state-of-the-art algorithm of Gomez et al. adapted from [4] to solve the Allen–Cahn equation. The temporal integration schemes are applied to the variational formulation presented in equations (4.55)-(4.56). The fully discrete formulations for the provably-stable methods can be defined as: given ϕ_n^h, σ_n^h, find ϕ_{n+1}^h, σ_{n+1}^h such that for all w^h, $q^h \in \mathcal{V}_2^h$

$$0 = \left(w^h, \frac{[\![\phi]\!]}{\Delta t}\right)_\Omega + \left(w^h, \sigma_{n+1}^h\right)_\Omega, \tag{4.68}$$

$$0 = \left(q^h, \sigma_{n+1}^h - \tilde{\Psi}'\right)_\Omega - \left(\nabla q^h, \nabla\{\phi\}\right)_\Omega - \alpha_{AC}\Delta t \left(q^h, [\![\phi]\!]\right)_\Omega, \tag{4.69}$$

where

- the implicit expansion scheme is recovered with

$$\tilde{\Psi}' = \Psi'_{n+1} - \Psi''_{n+1}\frac{[\![\phi]\!]}{2} + \Psi'''_{n+1}\frac{[\![\phi]\!]^2}{6}, \quad \alpha_{AC} = 0,$$

- the explicit expansion scheme is recovered with

$$\tilde{\Psi}' = \Psi'_n + \Psi''_n\frac{[\![\phi]\!]}{2}, \qquad \alpha_{AC} \geq \left(\frac{a_2\Delta t}{6}\right)^2,$$

- the scheme by Gomez et al. [4] is recovered with

$$\tilde{\Psi}' = \frac{1}{2}\left(\Psi'_n + \Psi'_{n+1}\right) - \Psi'''_n\frac{[\![\phi]\!]^2}{12}, \qquad \alpha_{AC} = 0.$$

The function Ψ is defined in equation (4.3) with the coefficients given in row AC of table 4.1. The solution evolution is shown in figure 4.2 with time step size $\Delta t = 10^{-4}$. In the figure, rows correspond to a particular method (indicated in the leftmost

column) while columns indicate the moment in time at which the solution snapshots are taken. Free energy evolutions are plotted in figures 4.3(a)-4.3(c). The only discernible difference between the simulations is observed with the backward-Euler method, which seems to reach the circular state faster. Even though counter intuitive, as the scheme is first order accurate in nature, this behavior can be explained by the inherent numerical dissipation introduced by the method. Simulations were also run for $\Delta t = 10^{-5}$ and $\Delta t = 10^{-6}$, but the snapshots are not shown: this example converges by $\Delta t = 10^{-5}$, confirmed by the free energy evolution seen in figure 4.3(b). The computed solutions are correct, given that they respect the energy stability property of the Allen–Cahn equation, namely,

$$\mathcal{F}_{AC} \leq 0.$$

All the second-order accurate methods follow the same behavior for the time step sizes studied, seemingly making the linear method the best choice. Given the explicit nature of its Jacobian, the solution can be obtained in less than half the computational time of the other second-order accurate methods considered, and also guarantees the monotonic decrease of free energy.

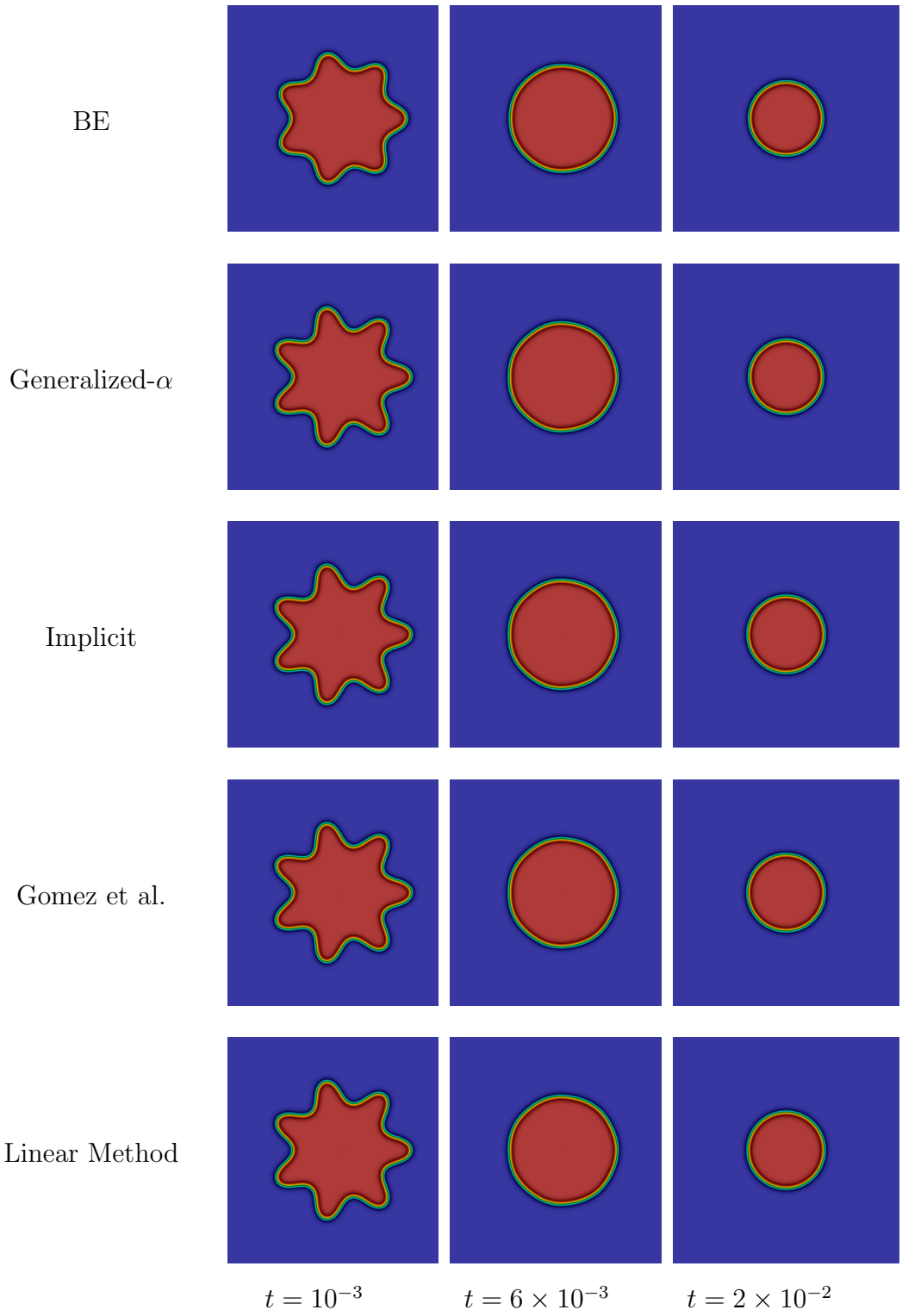
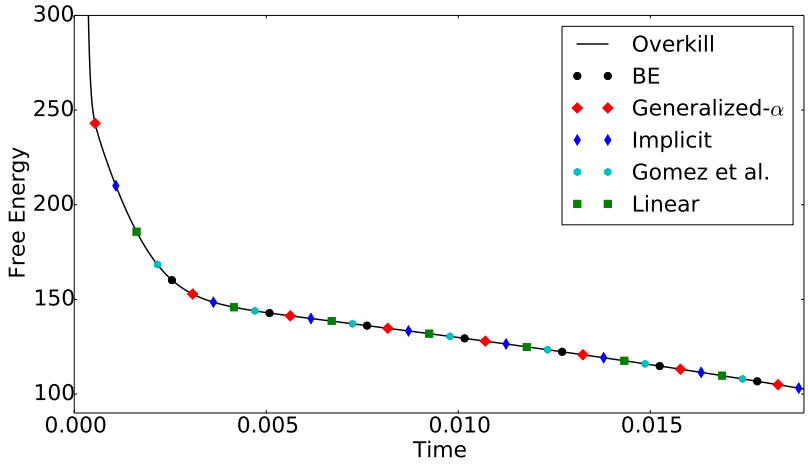
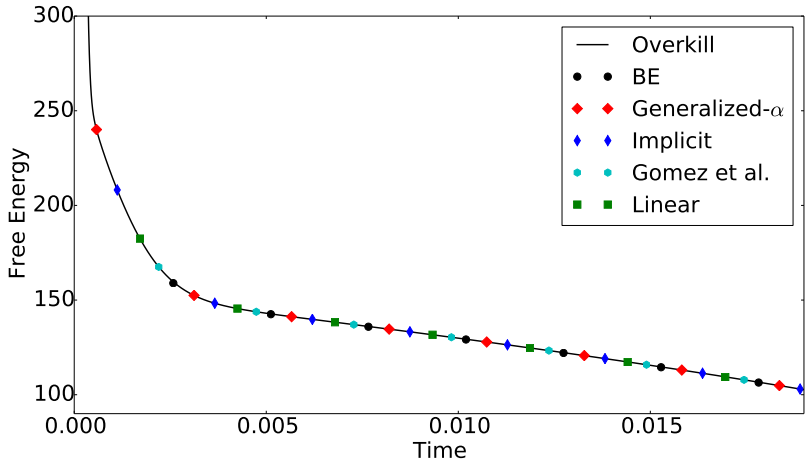


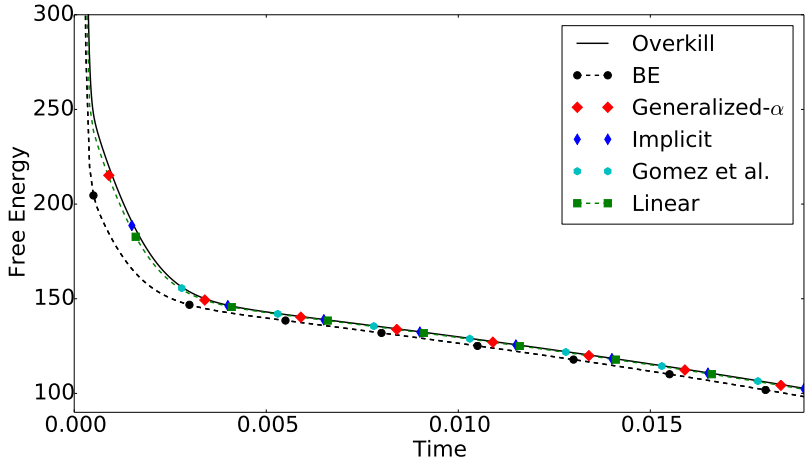
Figure 4.2: Evolution of a star-shaped interface in a curvature-driven flow. The simulation is run in a unit square domain $\bar{\Omega} = [0, 1]$ using a computational mesh composed of 512×512 $\mathcal{C}^1$-quadratic elements, with a time step size $\Delta t = 10^{-4}$.



(a) $\Delta t = 10^{-6}$



(b) $\Delta t = 10^{-5}$



(c) $\Delta t = 10^{-4}$

Figure 4.3: Free energy evolution comparison for the different methods analyzed to solve the Allen–Cahn equation.

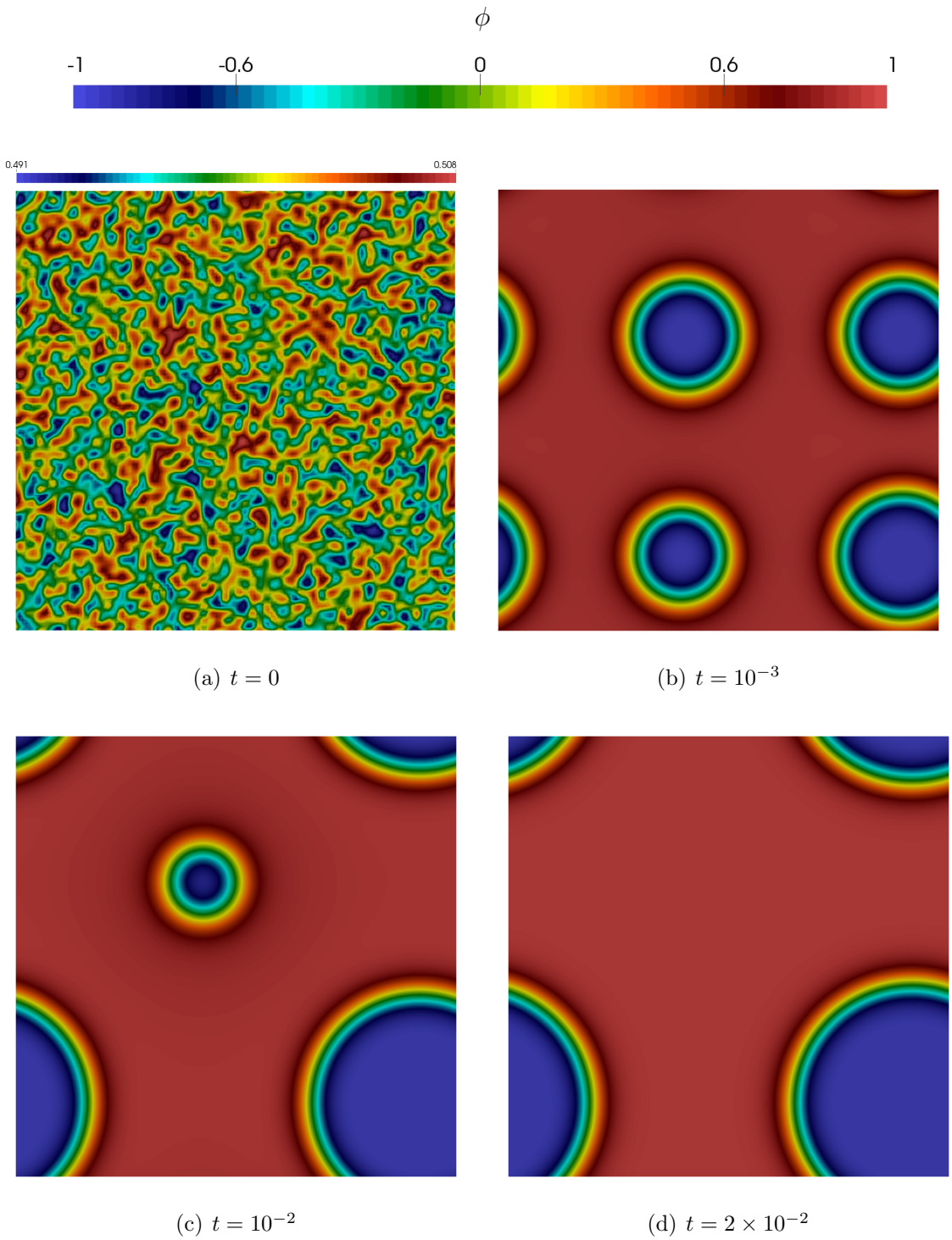
4.5.2 Cahn–Hilliard equation

The phase-separation process we simulate with the Cahn–Hilliard equation starts from the initial condition

$$\phi(\mathbf{x}) = \bar{\phi} + r,$$

where $\bar{\phi} = 0.5$ and r is a random variable with uniform distribution in $[-0.01, 0.01]$. Periodic boundary conditions are considered in both directions, in a unit square domain $\bar{\Omega} = [0, 1]^2$. The uniform mesh uses $[64]^2$-$\mathcal{C}^1$-quadratic elements. The initial condition is shown in figure 4.4(a), and represents a mixture which is unstable under those conditions [28]. The reference evolution of the phase is shown in figures 4.4(b) through 4.4(d). The mixture goes through an initial spinodal decomposition, which results in two coexisting phases being formed, and is followed by coarsening in later stages, whereby the two distinct phases grow with time to decrease the total interfacial area. The final state is reached when the phase separation is complete: a single bubble remains immersed within the other phase (4.4(d)). This solution minimizes the interfacial area for the average concentration proposed, a requirement stemming from the free energy functional $\mathcal{F}_{CH}$ (see equation (4.57)).

The performance of our provably-stable time-stepping algorithms is once again compared with (BE) and (Generalized-α), as well as the second-order, provably-stable, state-of-the-art algorithm presented in [2] by Gomez et al.. The temporal integration schemes are applied to the variational formulation presented in equations (4.58)-(4.59). The state-of-the-art algorithm we compare against [2] can be reinterpreted as a stabilized scheme. That is, this method can be stated as, given ϕ_n^h,



(a) $t = 0$

(b) $t = 10^{-3}$

(c) $t = 10^{-2}$

(d) $t = 2 \times 10^{-2}$

Figure 4.4: Spinodal decomposition using the Cahn–Hilliard equation. System evolves from a randomly perturbed initial condition that results in a single bubble at steady state. This solution minimizes the surface area of the interface. The unit square domain considered, $\bar{\Omega} = [0, 1]^2$, is meshed using $[64]^2\mathcal{C}^1$-quadratic elements. The solution evolution is calculated with our implicit algorithm, with a time step size $\Delta t = 10^{-7}$. We consider this our reference solution.

σ_n^h, find ϕ_{n+1}^h, σ_{n+1}^h such that for all w^h, $q^h \in \mathcal{V}_2^h$

$$0 = \left(w^h, \frac{[\![\phi]\!]}{\Delta t} \right)_\Omega + \left(\nabla w^h, M\left(\phi_{n+\alpha}^h\right) \nabla \sigma_{n+1}^h \right)_\Omega,$$
$$0 = \left(q^h, \sigma_{n+1}^h - \frac{1}{2}\left(\Psi_n' + \Psi_{n+1}'\right) + \Psi_n''' \frac{[\![\phi]\!]^2}{12} \right)_\Omega - \left(\nabla q^h, \nabla \phi_{n+\alpha}^h \right)_\Omega.$$

where

$$\phi_{n+\alpha}^h = \phi_n^h + \alpha [\![\phi]\!], \tag{4.70}$$
$$\alpha = \frac{1}{2} + \frac{1}{2}\tanh\left(\frac{\Delta t}{C\epsilon^2}\right), \tag{4.71}$$
$$C = 10^3. \tag{4.72}$$

Substituting equations (4.71) and (4.72) in equation (4.70), we have that

$$\begin{aligned} \phi_{n+\alpha}^h &= \phi_n^h + \alpha [\![\phi]\!], \\ &= \{\phi\} + \frac{1}{2}\tanh\left(\frac{\Delta t}{10^3\epsilon^2}\right) [\![\phi]\!]. \end{aligned} \tag{4.73}$$

Using equation (4.73), the fully discrete formulations for the provably-stable methods to solve the Cahn–Hilliard equation can now be expressed as: given ϕ_n^h, σ_n^h, find ϕ_{n+1}^h, σ_{n+1}^h such that for all w^h, $q^h \in \mathcal{V}_2^h$

$$0 = \left(w^h, \frac{[\![\phi]\!]}{\Delta t} \right)_\Omega + \left(\nabla w^h, M_{CH} \nabla \sigma_{n+1}^h \right)_\Omega,$$
$$0 = \left(q^h, \sigma_{n+1}^h - \tilde{\Psi}' \right)_\Omega - \left(\nabla q^h, \nabla \{\phi\} \right)_\Omega - \alpha_{CH} \left(\nabla q^h, \nabla [\![\phi]\!] \right)_\Omega,$$

where

- the implicit expansion scheme is recovered with

$$M_{CH} = M(\{\phi\}), \quad \tilde{\Psi}' = \Psi_{n+1}' - \Psi_{n+1}'' \frac{[\![\phi]\!]}{2} + \Psi_{n+1}''' \frac{[\![\phi]\!]^2}{6}, \quad \alpha_{CH} = 0,$$

- the explicit expansion scheme is recovered with

$$M_{CH} = M\left(\frac{3}{2}\phi_n - \frac{1}{2}\phi_{n-1}\right),\ \tilde{\Psi}' = \Psi'_n + \Psi''_n\frac{[\![\phi]\!]}{2},\ \alpha_{CH} \geq \Delta t\left(\frac{a_2\Delta t}{6}\right)^2,$$

- the scheme by Gomez et al. [2] is recovered with

$$\begin{aligned} M_{CH} &= M\left(\{\phi\} + \frac{1}{2}\tanh\left(\frac{\Delta t}{C\epsilon^2}\right)\right), \quad C = 1000, \\ \tilde{\Psi}' &= \frac{1}{2}\left(\Psi'_n + \Psi'_{n+1}\right) - \Psi'''_n\frac{[\![\phi]\!]^2}{12}, \quad \alpha_{CH} = \frac{1}{2}\tanh\left(\frac{\Delta t}{C\epsilon^2}\right). \end{aligned} \tag{4.74}$$

The value of the constant C can be tuned to either increase the accuracy or the robustness of the method [2]. The function Ψ is defined in equation (4.3) with the coefficients given in row CH of table 4.1.

The solution evolution with the different methods considered is shown in figure 4.5 with time step size $\Delta t = 10^{-5}$. Free energy evolutions are shown in figure 4.6. All methods seem to be converged at $\Delta t = 5 \times 10^{-6}$, and the linear method seems to be underestimating the dissipation rate when $\Delta t = 10^{-5}$. This is confirmed in figure 4.6(c). None of the methods present free energy increases, confirming a correct physical evolution.

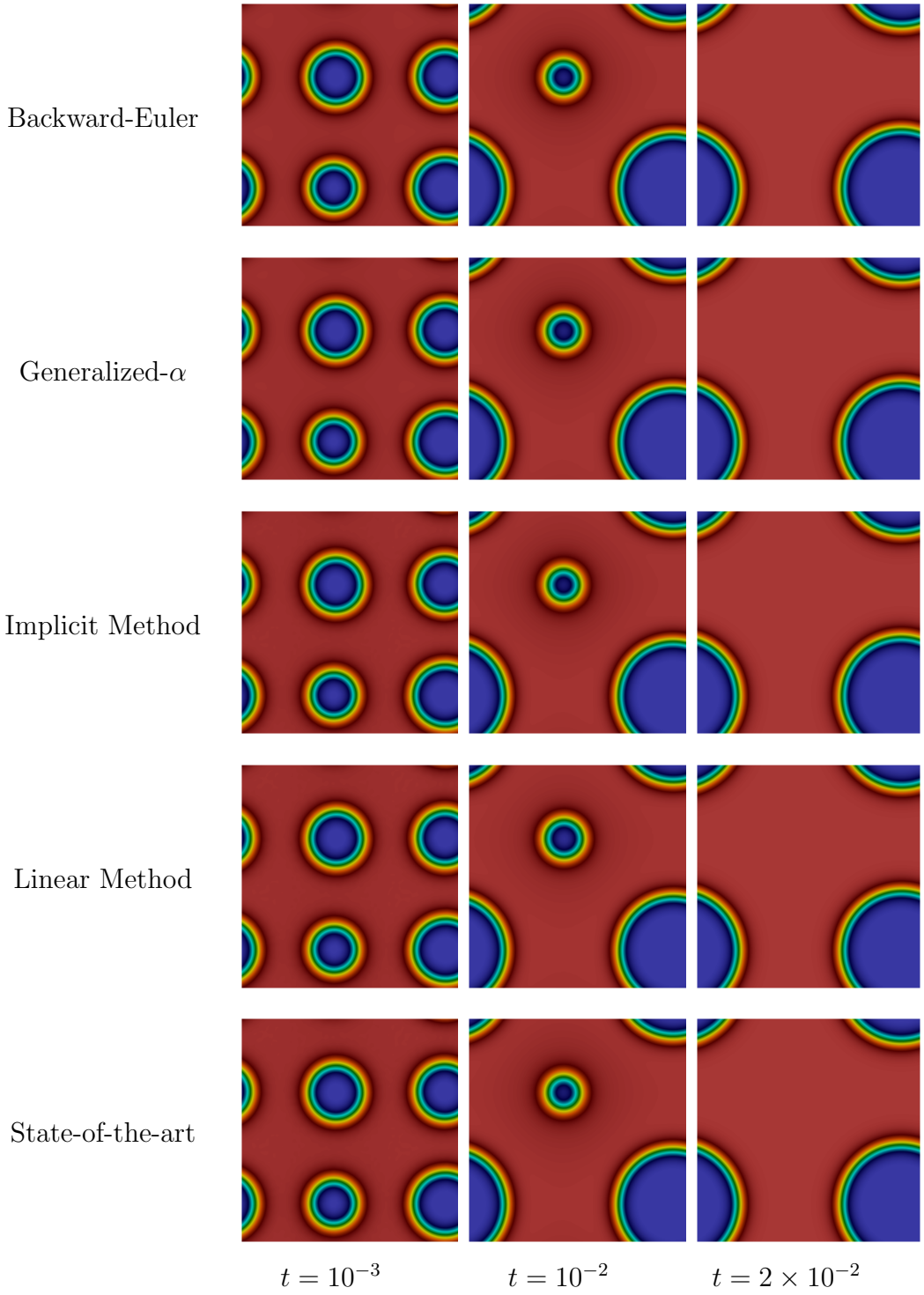
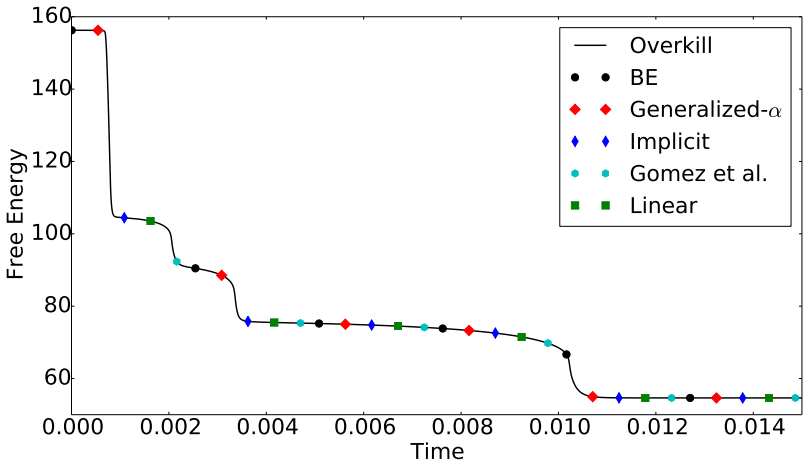
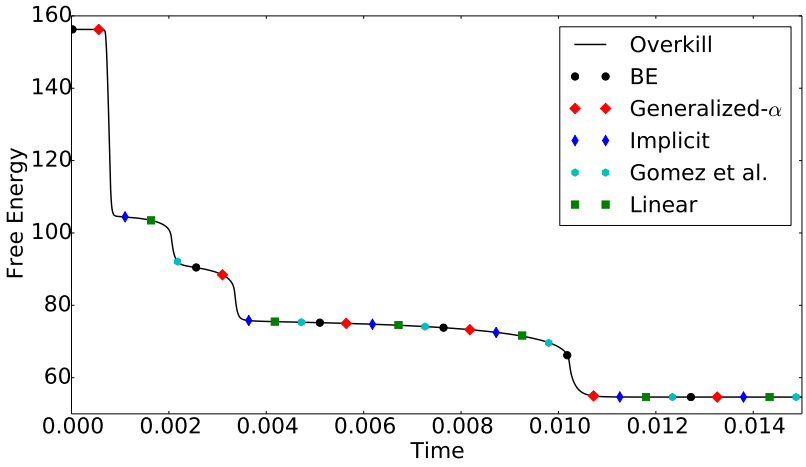


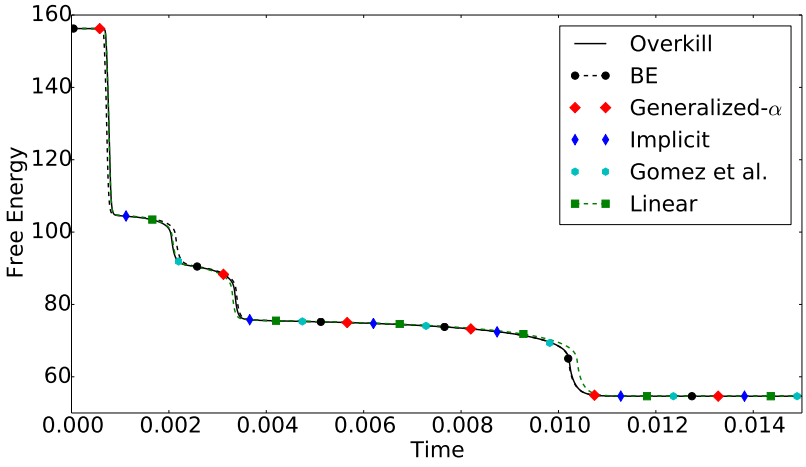
Figure 4.5: Evolution of the concentration ϕ from a randomly perturbed initial condition. The simulation is run in a unit square domain $\bar{\Omega} = [0, 1]^2$ using a computational mesh comprised of 64×64 $\mathcal{C}^1$-quadratic elements, with a time step size $\Delta t = 10^{-5}$.



(a) $\Delta t = 10^{-6}$



(b) $\Delta t = 5 \times 10^{-6}$



(c) $\Delta t = 10^{-5}$

Figure 4.6: Free energy evolution comparison for the different methods analyzed to solve the Cahn–Hilliard equation.

4.5.3 Swift–Hohenberg equation

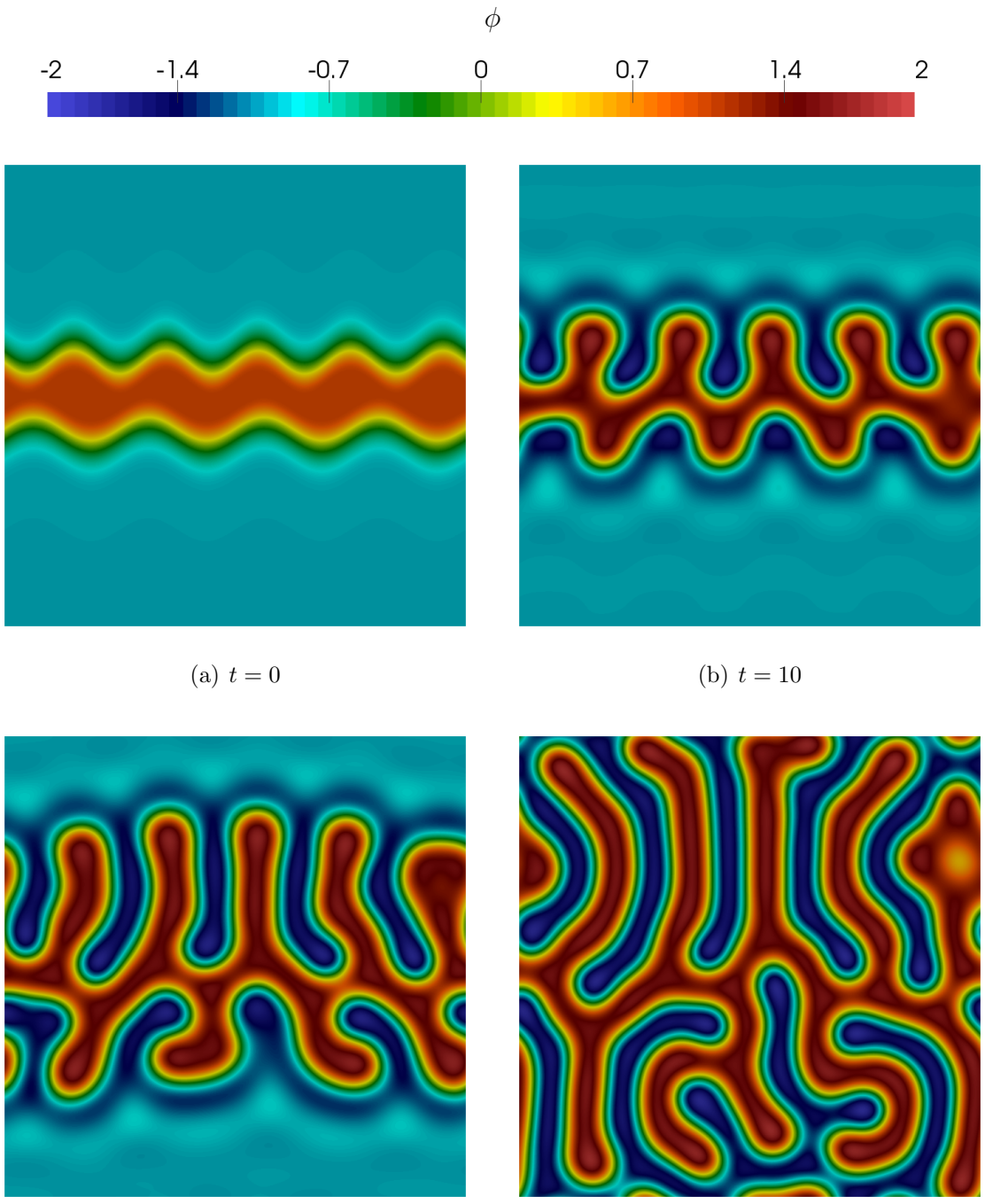
For this example, we modify an initial condition found in [4] that can be used to study pattern formation. Initially, a curvy horizontal stripe is embedded into a constant state. A square domain $\bar{\Omega} = [0, 40]^2$ is used, and we mesh it uniformly with $[512]^2$-$\mathcal{C}^1$-quadratic elements. The stripes are created using two sinusoidal waves defined by

$$C_1(\mathbf{x}) = \sin\left(\frac{2\pi}{40}4x + 19\right),$$
$$C_2(\mathbf{x}) = -\sin\left(\frac{2\pi}{40}5x + 21\right).$$

The phase field variable initially takes a value equal to one within the space created by the two stripes, and a value of minus one outside of it. To create a smooth interface between these two values, a rescaled Gaussian function is used, which leads to the initial condition

$$\phi_0 = \begin{cases} 1 & \text{if} \quad C_1 < y < C_2, \\ 2\exp\left(-\dfrac{(y - C_1)^2}{2\nu}\right) - 1 & \text{if} \quad y < C_1, \\ 2\exp\left(-\dfrac{(y - C_2)^2}{2\nu}\right) - 1 & \text{if} \quad y < C_2, \end{cases}$$

with $\nu = 4$. To avoid oscillations generated by the projection of ϕ and ensure the monotonic behavior of the phase field, we use a mass lumping technique [129]. This procedure also avoids spurious oscillations generated by the temporal integration schemes encountered in [4] for large time step sizes. The initial condition is shown in 4.7(a), and snapshots of the phase field ϕ at three instances in time can be seen in figures 4.7(b), 4.7(c) and 4.7(d). The pattern evolves developing vertical fingers that can bifurcate.



(a) $t = 0$

(b) $t = 10$

(c) $t = 30$

(d) $t = 70$

Figure 4.7: Pattern formation using the Swift–Hohenberg equation. System evolves from a curvy stripe embedded into a constant state. The pattern develops vertical fingers that can bifurcate throughout the evolution. The square domain considered, $\bar{\Omega} = [0, 40]^2$, is uniformly meshed using $[512]^2\mathcal{C}^1$-quadratic elements. The boundary conditions are periodic. The solution evolution is calculated with our implicit algorithm, with a time step size $\Delta t = 10^{-2}$. We consider this our reference solution.

The performance of our provably-stable time-stepping algorithms is once again compared against (BE) and (Generalized-α), as well as the second-order, provably-stable, state-of-the-art algorithm presented by Gomez et al. in [4]. The temporal integration schemes are applied to the variational formulation presented in equations (4.63)-(4.64). The fully discrete formulations for the provably-stable methods to solve the Swift–Hohenberg equation can be defined as: given ϕ_n^h, σ_n^h, find ϕ_{n+1}^h, σ_{n+1}^h such that for all w^h, $q^h \in \mathcal{V}_2^h$

$$
\begin{aligned}
0 &= \left(w^h, \frac{[\![\phi]\!]}{\Delta t}\right)_\Omega + \left(w^h, \sigma_{n+1}^h\right)_\Omega, \\
0 &= \left(q^h, \sigma_{n+1}^h - \tilde{\Psi}'\right)_\Omega - 2\left(\nabla q^h, \nabla\{\phi\}\right)_\Omega + \left(\Delta q^h, \Delta\{\phi\}\right)_\Omega \\
&\quad - \alpha_{SH}\Delta t \left(q^h, [\![\phi]\!]\right)_\Omega,
\end{aligned}
$$

- the implicit expansion scheme is recovered with

$$
\tilde{\Psi}' = \Psi'_{n+1} - \Psi''_{n+1}\frac{[\![\phi]\!]}{2} + \Psi'''_{n+1}\frac{[\![\phi]\!]^2}{6}, \qquad \alpha_{SH} = 0,
$$

- the explicit expansion scheme is recovered with

$$
\tilde{\Psi}' = \Psi'_n + \Psi''_n\frac{[\![\phi]\!]}{2}, \qquad \alpha_{SH} \geq \left(\frac{a_2\Delta t}{6}\right)^2,
$$

- the scheme by Gomez et al. [4] is recovered with

$$
\tilde{\Psi}' = \frac{1}{2}\left(\Psi'_n + \Psi'_{n+1}\right) - \Psi'''_n\frac{[\![\phi]\!]^2}{12}, \qquad \alpha_{SH} = 0.
$$

The function Ψ is defined in equation (4.3) with the coefficients given in row SH of table 4.1. Snapshots of the phase field ϕ calculated with the different methods and time step sizes $\Delta t = 0.1, 1$, are shown in figures 4.8 and 4.9, respectively. The free

energy evolutions for $\Delta t = 0.1$, 0.5 and 1, respectively, are plotted in figures 4.10(a), 4.10(b) and 4.10(c). As observed in previous cases, the underestimation of the dissipation rate increases in the linear scheme with time step size. The opposite effect is observed in the case of the backward-Euler method, which shows more dissipation of the free energy with Δt. The solver breaks-down for the case of backward-Euler with $\Delta t = 1$. The provably-stable methods also start separating from the reference solution at that time step size. The explicit expansion of the free energy, which leads to a linear method, separates the most from the reference solution.

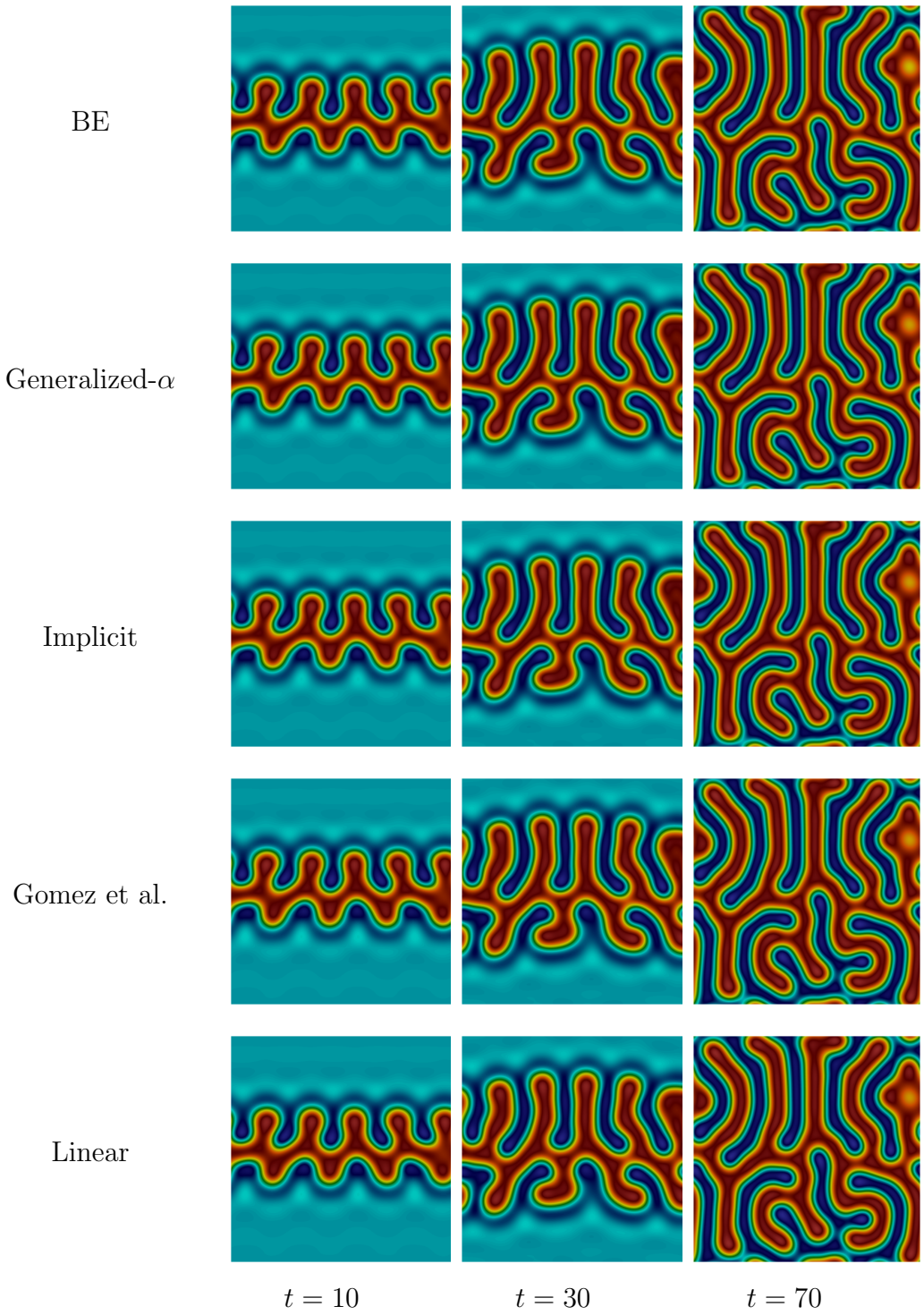


Figure 4.8: Pattern formation using the Swift–Hohenberg equation. The simulation is run in a square domain $\bar{\Omega} = [0, 40]^2$ using a computational mesh comprised of 512×512 $\mathcal{C}^1$-quadratic elements, pediodic boundary conditions along both directions, with a time step size $\Delta t = 0.1$.

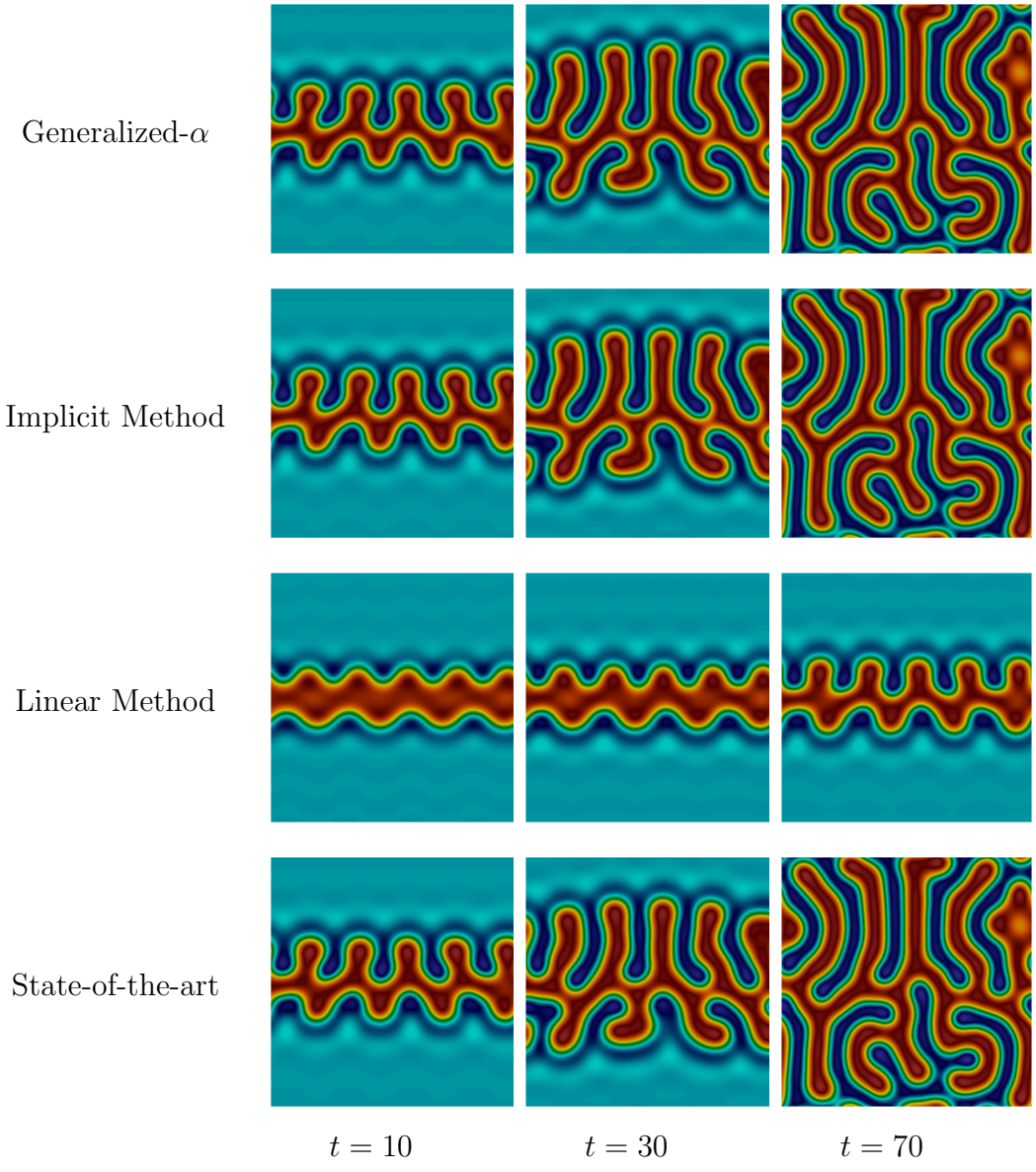
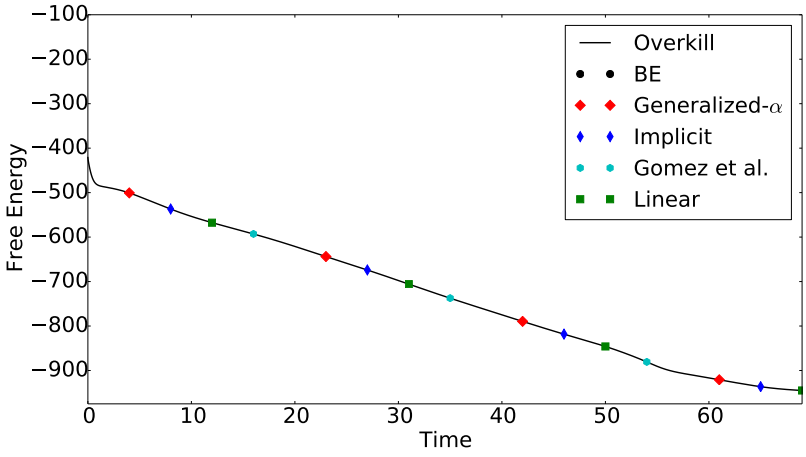
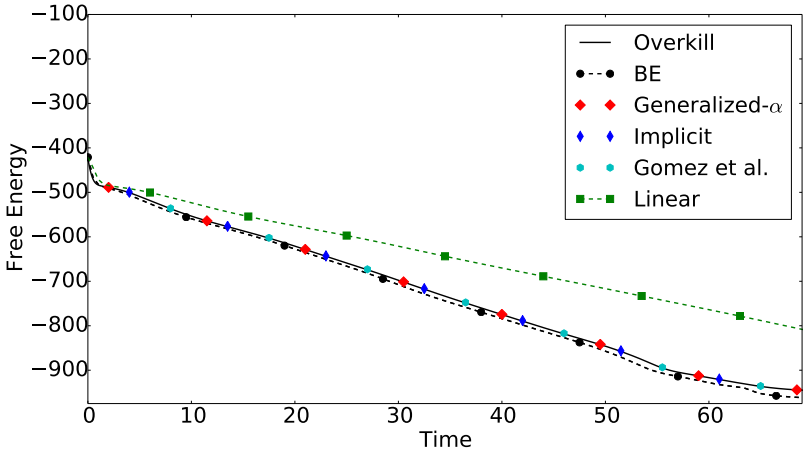


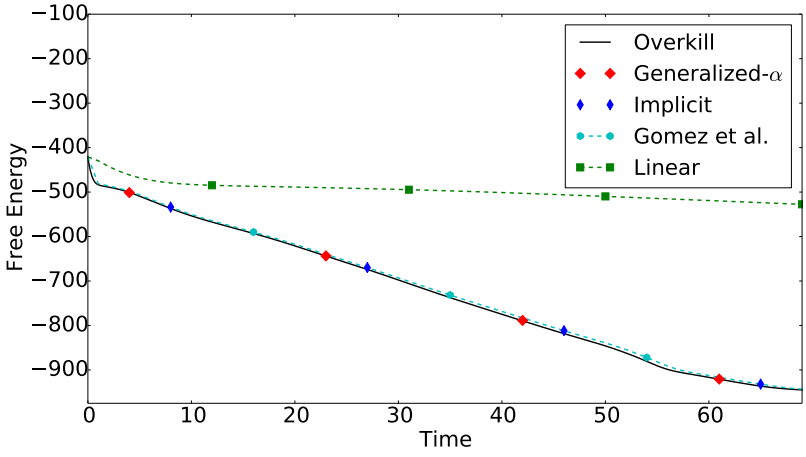
Figure 4.9: Pattern formation using the Swift–Hohenberg equation. The simulation is run in a square domain $\bar{\Omega} = [0, 40]^2$ using a computational mesh comprised of 512×512 $\mathcal{C}^1$-quadratic elements, pediodic boundary conditions along both directions, with a time step size $\Delta t = 1$.



(a) $\Delta t = 0.1$



(b) $\Delta t = 0.5$



(c) $\Delta t = 1$

Figure 4.10: Free energy evolution comparison for the different methods analyzed to solve the Swift–Hohenberg equation. The case for backward-Euler with $\Delta t = 1$ is not shown as the solver broke down.

4.5.4 Phase-field crystal equation

We simulate the two-dimensional growth of a crystal in a supercooled liquid, using a one-mode approximation for the density profile of the crystalline structure [5]. We use a triangular configuration, defined by

$$\phi_S(\mathbf{x}) = \cos\left(\frac{q}{\sqrt{3}}y\right)\cos(qx) - \frac{1}{2}\cos\left(\frac{2q}{\sqrt{3}}y\right), \tag{4.75}$$

where q represents a wavelength related to the lattice constant [14], and x and y represent the Cartesian coordinates. A solid crystallite is initially placed in the center of a liquid domain, which is assigned an average density $\bar{\phi}$. The initial condition becomes

$$\phi_0(\mathbf{x}) = \bar{\phi} + \omega(\mathbf{x})\left(A\phi_S(\mathbf{x})\right), \tag{4.76}$$

where A represents an amplitude of the fluctuations in density, and the scaling function $\omega(\mathbf{x})$ is defined as

$$\omega(\mathbf{x}) = \begin{cases} \left(1 - \left(\dfrac{||\mathbf{x}-\mathbf{x}_0||}{d_0}\right)^2\right)^2 & \text{if} \quad ||\mathbf{x}-\mathbf{x}_0|| \leq d_0, \\ 0 & \text{otherwise,} \end{cases}$$

where $\mathbf{x}_0$ is the coordinate of the center of the domain, and d_0 is $\dfrac{1}{6}$ of the domain length in the x-direction. In order to avoid mismatches on the boundaries when the grain boundaries meet, the computational domain $\bar{\Omega}$ is dimensioned in such a way as to make it periodic along both directions. To do this while keeping the problem within a reasonable size, we use the frequency present in equation (4.75) to define the

domain $\bar{\Omega}$ as

$$\bar{\Omega} = \left[0, \frac{2\pi}{q}a\right] \times \left[0, \frac{\sqrt{3}\pi}{q}b\right],$$

where a and b are assigned values of 10 and 12, respectively. These numbers make the domain almost square. The number of elements in the y-direction, N_y, is then defined as

$$N_y = \left\lfloor \frac{b\sqrt{3}}{2a}N_x + \frac{1}{2} \right\rfloor,$$

where N_x represents the number of elements in the x-direction. This adjustment is made to account for the difference in length between both directions, and to have the element size h in both directions be approximately equal. The variables q and A are assigned their corresponding equilibrium values, obtained by minimizing the free energy presented in table 4.1, with respect to both A and q, while using the approximation of equation (4.75) to define the atomistic density. For the results presented in this example, the values used are

$$\epsilon = 0.325, \quad \bar{\phi} = \frac{\sqrt{\epsilon}}{2}, \quad A = \frac{4}{5}\left(\bar{\phi} + \frac{\sqrt{15\epsilon - 36\bar{\phi}^2}}{3}\right), \quad q = \frac{\sqrt{3}}{2}.$$

The parameter ϵ is chosen such that the triangular structure is stable [14, 103]. The initial crystallite placed in the center of the domain grows at the expense of the supercooled liquid. This is enforced by the degree of undercooling ϵ.

The performance of our provably-stable time-stepping algorithms is once again compared against (BE) and (Generalized-α), as well as the second-order, provably-stable, state-of-the-art algorithm presented by Gomez et al. in [3]. The temporal integration schemes are applied to the variational formulation presented in equations (4.65)-(4.66). The fully discrete formulations for the provably-stable methods

to solve the phase-field crystal equation can be defined as: given ϕ_n^h, σ_n^h, find ϕ_{n+1}^h, σ_{n+1}^h such that for all w^h, $q^h \in \mathcal{V}_2^h$

$$
\begin{aligned}
0 &= \left(w, \frac{\partial \phi}{\partial t}\right)_\Omega + (\nabla w, \nabla \sigma)_\Omega \,, \\
0 &= \left(q, \sigma - \phi - \tilde{\Psi}'\right)_\Omega + 2\left(\nabla q^h, \nabla \phi\right)_\Omega - \left(\Delta q^h, \Delta \phi\right)_\Omega \\
&\quad - \alpha_{PFC} \Delta t \left(\nabla q^h, \nabla [\![\phi]\!]\right)_\Omega .
\end{aligned}
$$

- the implicit expansion scheme is recovered with

$$
\tilde{\Psi}' = \Psi'_{n+1} - \Psi''_{n+1} \frac{[\![\phi]\!]}{2} + \Psi'''_{n+1} \frac{[\![\phi]\!]^2}{6}, \qquad \alpha_{PFC} = 0,
$$

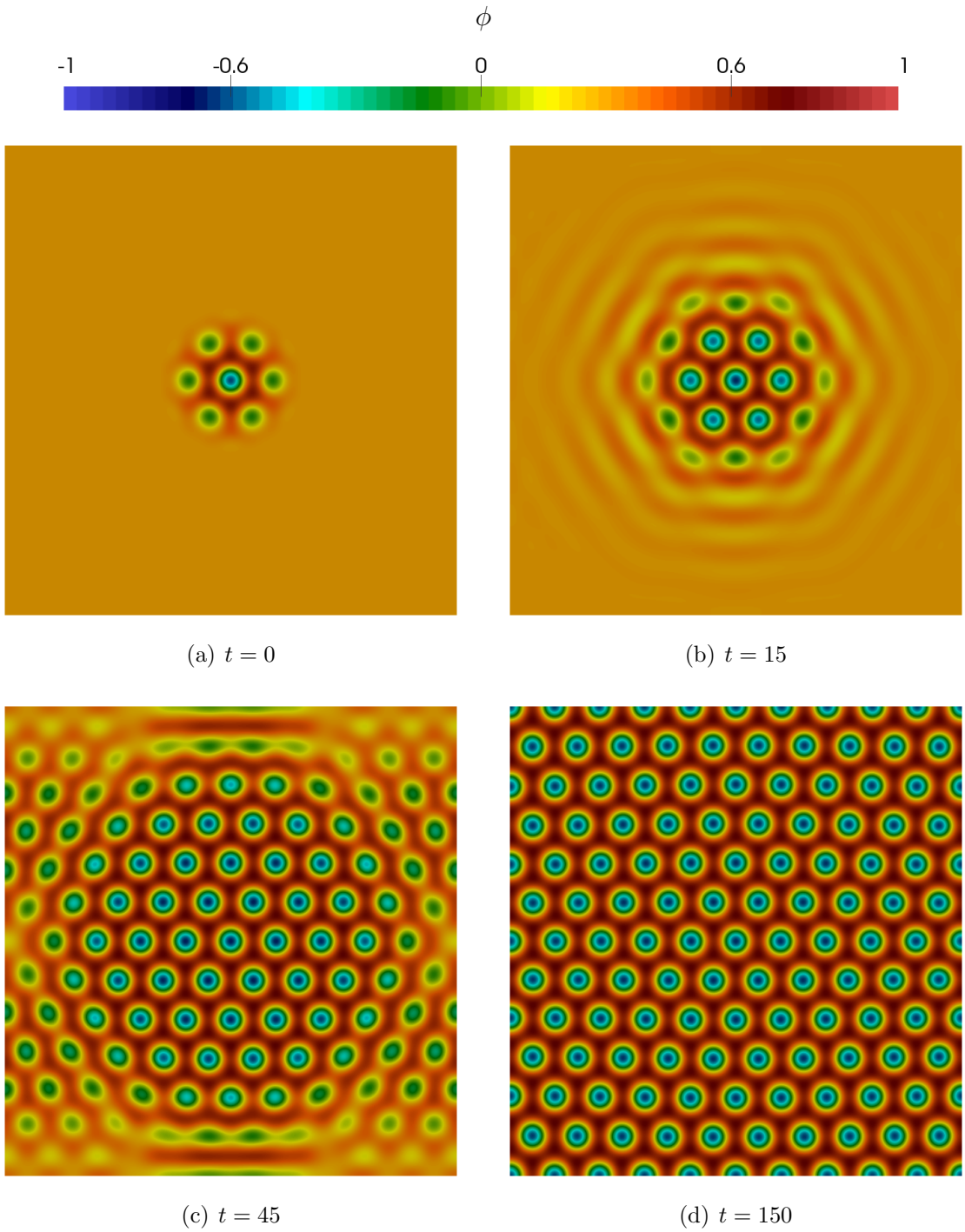
- the explicit expansion scheme is recovered with

$$
\tilde{\Psi}' = \Psi'_n + \Psi''_n \frac{[\![\phi]\!]}{2}, \qquad \alpha_{PFC} \geq \left(\frac{a_2 \Delta t}{6}\right)^2,
$$

- the scheme by Gomez et al. [3] is recovered with

$$
\tilde{\Psi}' = \frac{1}{2}\left(\Psi'_n + \Psi'_{n+1}\right) - \Psi'''_n \frac{[\![\phi]\!]^2}{12}, \qquad \alpha_{PFC} = 0.
$$

The function Ψ is defined in equation (4.3) with the coefficients given in row PFC of table 4.1. The simulations with the different temporal schemes are shown in figures 4.12 and 4.13, for time step sizes $\Delta t = 0.1$ and 1, respectively. Free energy evolutions are shown in figure 4.14. This example follows the same trend as the one shown for the Swift–Hohenberg equation, with the backward-Euler method dissipating more free energy with the increase in time step size than the other schemes. Also with the increase in time size, the effectivity of the time step size for the linear case decreases, as seen in figures 4.13 and 4.14(c). This effect is also observed in [130, 5].



(a) $t = 0$

(b) $t = 15$

(c) $t = 45$

(d) $t = 150$

Figure 4.11: Crystal growth using the phase-field crystal equation. A small crystallite with a stable triangular structure is placed in the centre of the domain. Given the undercooling enforced by ϵ, the crystallite grows and replicates the initial pattern throughout the domain. The rectangular domain considered, $\bar{\Omega} = [0, {}^{40\pi}/_{\sqrt{3}}] \times [0, 24\pi]$, is uniformly meshed using $[512] \times [532]$ $\mathcal{C}^1$-quadratic elements. Periodic boundary conditions are considered along both directions. The solution evolution is calculated with our implicit algorithm, with a time step size $\Delta t = 10^{-2}$. We consider this our reference solution.

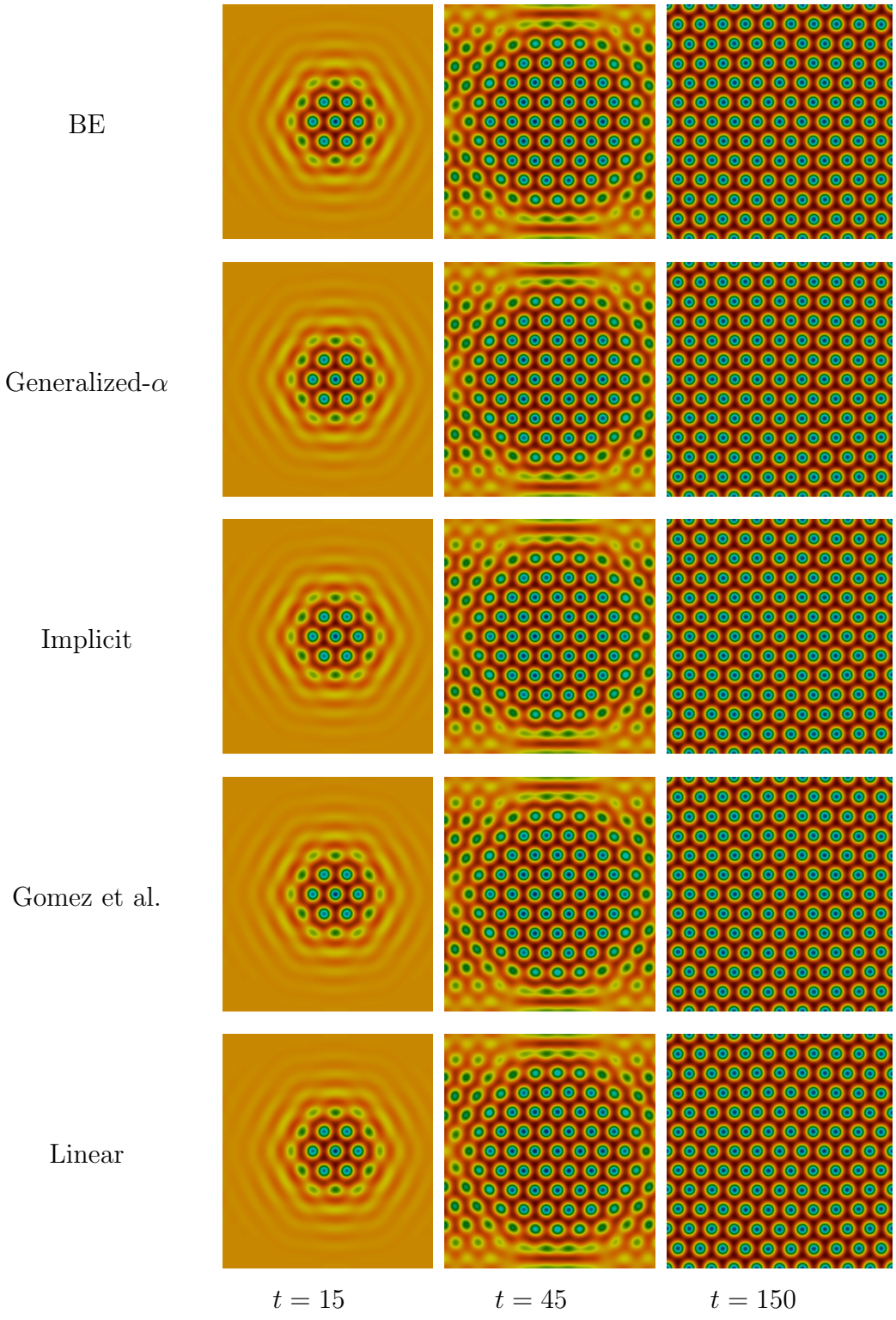


Figure 4.12: Snapshots of the approximate dimensionless atomistic density field showing its evolution throughout the simulation, which was run using a computational mesh composed of 512×532 C^1-quadratic elements, with a time step size $\Delta t = 0.1$.

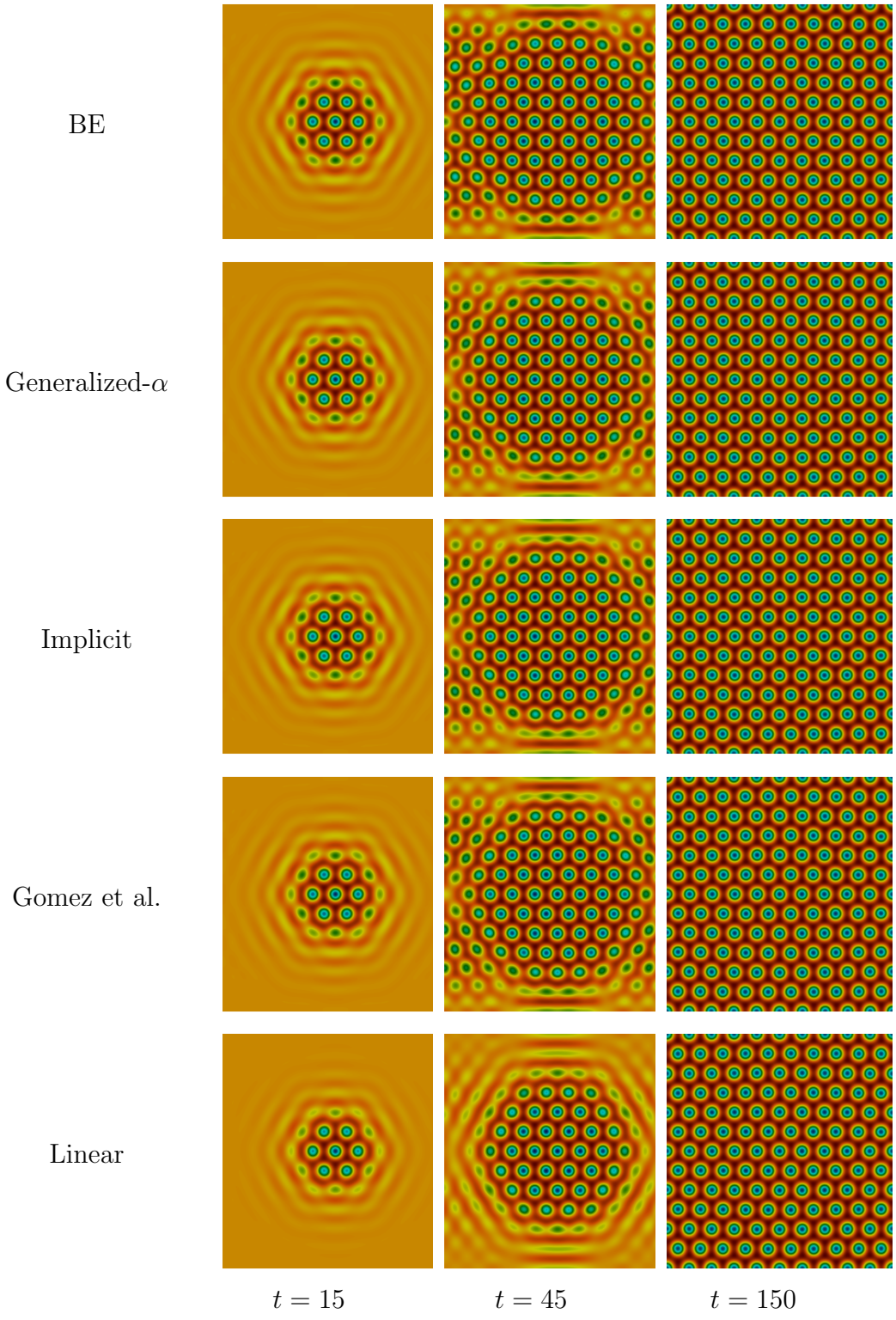
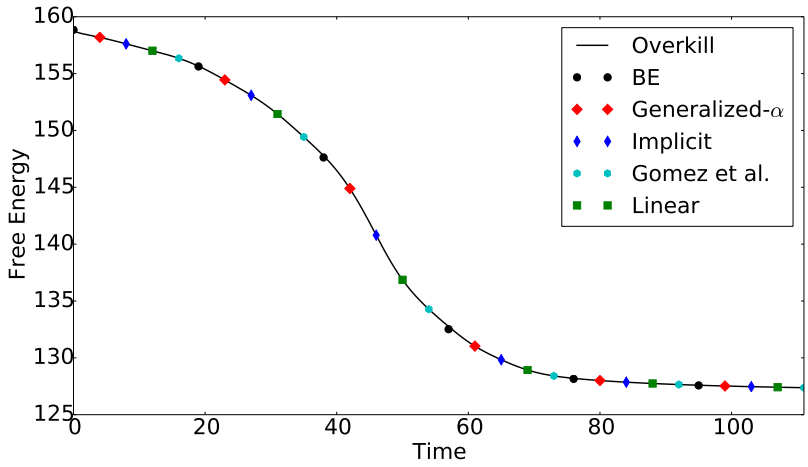
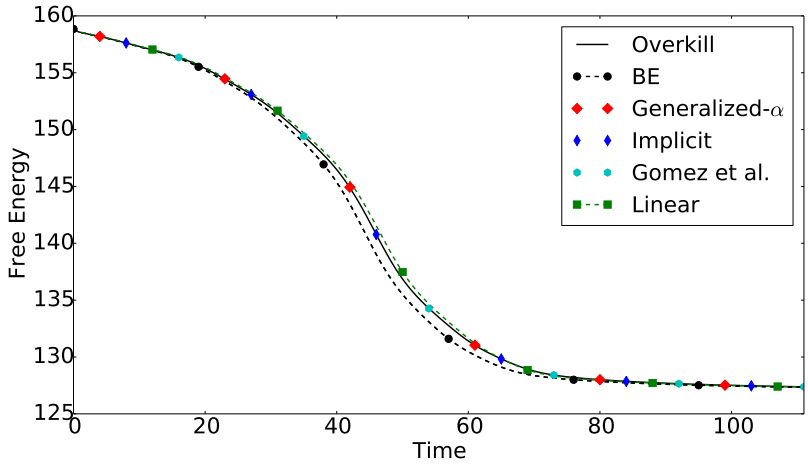


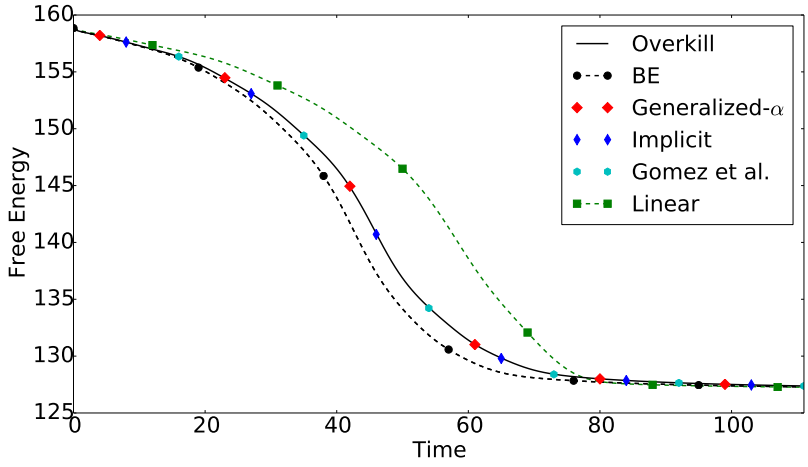
Figure 4.13: Snapshots of the approximate dimensionless atomistic density field showing its evolution throughout the simulation, which was run using a computational mesh composed of 512×532 C^1-quadratic elements, with a time step size $\Delta t = 1$.



(a) $\Delta t = 0.1$



(b) $\Delta t = 0.5$



(c) $\Delta t = 1$

Figure 4.14: Free energy evolution comparison for the different methods analyzed to solve the phase-field crystal equation.

4.5.5 Methods: comparison and performance

The cases studied show similar features. The backward-Euler method is the first to break down in every case with the increase in time step size, while the time step effectivity of the linear method decreases with time step size. Still, to obtain overkill solutions for the high-order equations studied, the linear method is a reasonable choice. It offers significant computational savings as the expression for the Jacobian is explicit (see Appendix C.1.1). Another feature that is worthwhile pointing out involves the striking resemblance between the implicit method and the state-of-the-art methods developed in [2, 3, 4]. This can be explained by analyzing the structure of the approximation proposed for Ψ' in equation (4.74). Recalling equation (4.25) and using the Lagrange remainder, the terms Ψ'_n and Ψ'''_n can be rewritten as

$$\Psi'_n = \Psi'_{n+1} - \Psi''_{n+1}[\![\phi]\!] + \Psi'''_{n+1}\frac{[\![\phi]\!]^2}{2} - \Psi^{iv}_{n+\epsilon_3}\frac{[\![\phi]\!]^3}{6}, \tag{4.77}$$

$$\Psi'''_n = \Psi'''_{n+1} - \Psi^{iv}_{n+\epsilon_4}[\![\phi]\!], \tag{4.78}$$

where $\epsilon_3,\ \epsilon_4 \in [0,1]$. Realizing that $\Psi^{iv}_{n+\epsilon_3} = \Psi^{iv}_{n+\epsilon_4} = \Psi^{iv}_{n+1} = \Psi^{iv}$, and substituting equations (4.77)-(4.78) in (4.74) results in

$$\begin{aligned}\tilde{\Psi}' &= \Psi'_{n+1} + \left(-\Psi''_{n+1}\frac{[\![\phi]\!]}{2} + \Psi'''_{n+1}\frac{[\![\phi]\!]^2}{4} - \Psi^{iv}\frac{[\![\phi]\!]^3}{12}\right) \\ &\quad - \left(\Psi'''_{n+1}\frac{[\![\phi]\!]^2}{12} - \Psi^{iv}\frac{[\![\phi]\!]^3}{12}\right), \\ &= \Psi'_{n+1} - \Psi''_{n+1}\frac{[\![\phi]\!]}{2} + \Psi'''_{n+1}\frac{[\![\phi]\!]^2}{6}.\end{aligned}$$

Thus, the approximations proposed in [3, 4] and in (4.11) for $\tilde{\Psi}'$ are equivalent. This expansion, to the best of our knowledge, was initially proposed in the context of the Navier–Stokes–Cahn–Hilliard equation [82] but the connection between the two had yet to be made. The linear expansion proposed through (4.12) was also previ-

ously proposed in the context of the Allen–Cahn and Cahn–Hilliard equations [43] and the Navier–Stokes–Cahn–Hilliard equation [131]. Nonetheless, the case with the degenerate mobility was not treated in that work, nor was the value for the stabilization parameter α determined. With regards to the scheme proposed in [2] for the Cahn–Hilliard equation, recalling equation (4.73), considering $\frac{\Delta t}{10^3\epsilon^2} \geq 0$ and $\frac{1}{2}\tanh(x) \leq \frac{x}{2}$, we have the inequality

$$0 \leq \frac{1}{2}\tanh\left(\frac{\Delta t}{10^3\epsilon^2}\right) \leq \frac{\Delta t}{2 \times 10^3\epsilon^2}.$$

Thus, the scheme proposed in [2] can be reinterpreted as a stabilized scheme that uses the implicit approximation of equation (4.11). The value of the stabilization parameter proposed in [2] verifies the bound presented in equation (4.30) for the linear case given the Δt considered. Still, as concluded in section 4.2.4.1, the stabilization parameter can have a value of 0 for the polynomial version of the double well potential when the implicit expansion is used. In the case of the logarithmic version of the potential considered for Ψ in [2], it must have a value greater than 0 to guarantee free energy stability.

Regarding computational performance, results are shown in in table 4.2. The linear method takes less computational time than all the other methods for a given time step size. The equivalence of the implicit method and the methods developed by Gomez et al. is also confirmed numerically, given that the number of nonlinear iterations is the same for all cases studied.

Table 4.2: Computational performance of the different methods. We show the computational time of the different simulations with respect to the linear method in each one of the different cases, as well as the number of nonlinear and linear iterations. The linear method is faster than all the other methods for a given time step size. The number of nonlinear iterations is the same for the implicit and the methods developed in [2, 3, 4] by Gomez et al., which numerically confirms the result of section 4.5.5: the methods are equivalent.

		AC			CH			SH			PFC		
	Simulation time T	0.02			0.02			70			150		
	Δt	10^{-6}	10^{-5}	10^{-4}	10^{-6}	5×10^{-6}	10^{-5}	0.1	0.5	1	0.1	0.5	1
BE	Ref. comp. time	1.68	1.70	2.63	2.02	2.60	2.70	4.7	9.8	-	3.99	4.15	7.5
	Nonlinear iterations	40002	4068	617	40872	9352	5096	2114	566	-	4502	900	518
	Linear iterations	314378	32594	9362	289936	106169	71627	736547	414644	-	2668526	806003	562104
G-α	Ref. comp. time	1.82	1.71	2.44	2.09	2.36	2.26	3.68	7.38	14.60	3.63	3.51	5.4
	Nonlinear iterations	40002	4036	608	40623	8652	4748	2103	561	286	4499	900	450
	Linear iterations	343589	30661	7684	251562	80766	54966	582343	312996	216314	2230008	691960	400426
Implicit	Ref. comp. time	1.98	1.93	2.83	2.12	2.62	3.26	3.71	5.05	13.08	2.67	3.39	5.33
	Nonlinear iterations	40002	4034	608	40987	9098	6028	2101	441	282	3752	897	450
	Linear iterations	342864	30415	7675	246534	82039	68429	565271	207023	198043	1695227	649567	384345
Gomez et al.	Ref. comp. time	2.07	2.02	2.93	2.30	2.78	3.48	3.73	5.02	12.97	2.67	3.40	5.26
	Nonlinear iterations	40002	4034	608	40987	9098	6028	2101	441	282	3752	897	450
	Linear iterations	342864	30415	7675	246531	82119	68451	565269	206931	198048	1695239	649553	384126
Linear	Ref. comp. time	1	1	1	1	1	1	1	1	1	1	1	1
	Nonlinear iterations	-	-	-	-	-	-	-	-	-	-	-	-
	Linear iterations	156535	15580	2589	123315	35977	22400	142246	38155	12564	625111	190375	70346

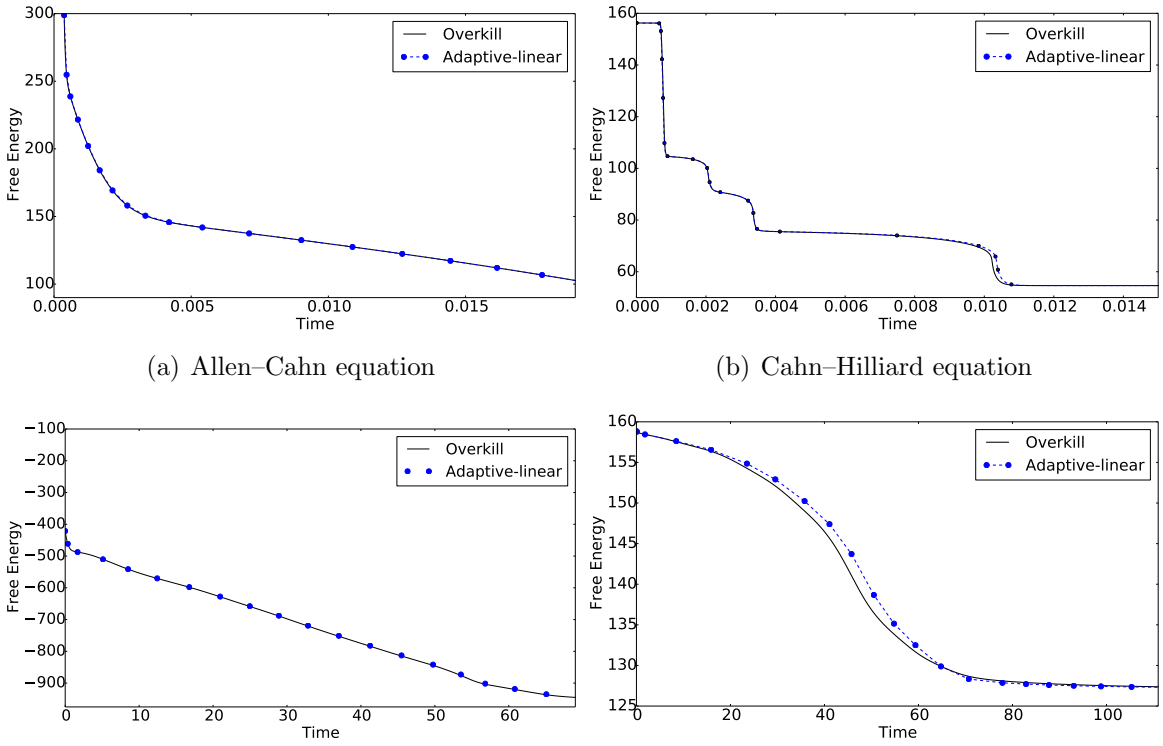
Table 4.3: Computational gains using the linear-adaptive algorithm to solve the phase-field problems of sections 4.5.1-4.5.4. By using the linear-adaptive strategy, the number of iterations taken to reach a high-quality solution is drastically reduced, and on the order of the coarser simulations presented in table 4.2 for each one of the four cases studied.

	AC	CH	SH	PFC
Simulation time T	0.02	0.02	70	150
Total number of steps	319	659	478	351
Linear iterations	3166	7992	95752	205931
Rejected steps	0	45	0	82
Maximum Δt	1.59×10^{-4}	7.94×10^{-3}	2.48×10^{-1}	7.12
Minimum Δt	10^{-8}	10^{-11}	10^{-5}	10^{-6}

4.5.6 Performance of the time-adaptive scheme and three-dimensional simulation of the Cahn–Hilliard equation

The two-dimensional examples presented in sections 4.5.1-4.5.4 are solved using the linear scheme within the context of algorithm 3, with the results shown in terms of the free energy evolution in figure 4.15. With the exception of the phase-field crystal equation, the linear-adaptive solution compares well with the overkill solution for each one of the cases, and can be computed with fewer iterations as seen in table 4.3. The number of iterations for the Swift–Hohenberg and phase-field crystal equations can be reduced further by splitting the equation through the use of another auxiliary variable, resulting in a similar weak form to the one presented in [5]. We plan to study this point in future work.

Finally, to showcase the robustness of the linear-adaptive method, the three-dimensional counterpart of the example solved in two dimensions for the Cahn–Hilliard equation in section 4.5.2 is presented in figure 4.17. Periodic boundary conditions are considered in all directions, in a unit cube domain $\bar{\Omega} = [0, 1]^3$. The uniform mesh uses $[64]^3$-$\mathcal{C}^1$ cubic elements. Similarly to what happens in the two-dimensional case, the mixture initially undergoes a fast spinodal decomposition, which results in the formation of two coexisting phases. Once completed, this physical phenomenon



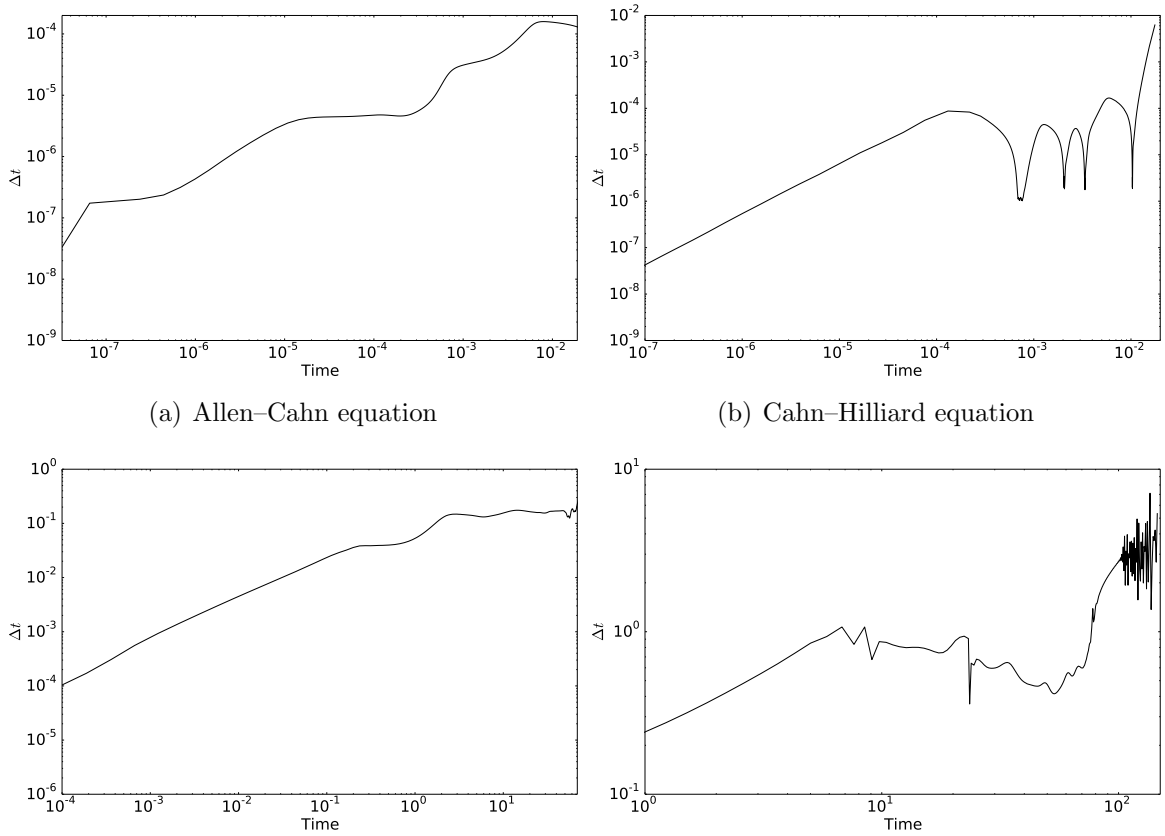
(a) Allen–Cahn equation

(b) Cahn–Hilliard equation

(c) Swift–Hohenberg equation

(d) Phase-field crystal equation

Figure 4.15: Evaluating the adaptive-linear time stepping scheme: free energy evolution. The evolution of the free energy is accurately represented for the four phase-field models studied.



(a) Allen–Cahn equation

(b) Cahn–Hilliard equation

(c) Swift–Hohenberg equation

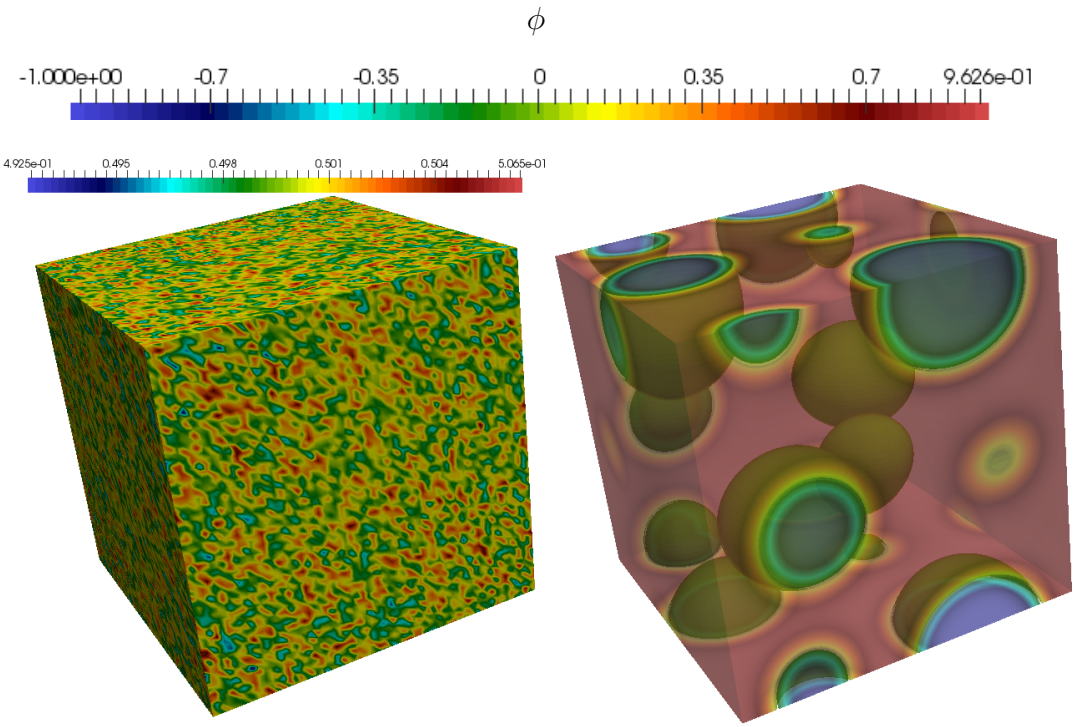
(d) Phase-field crystal equation

Figure 4.16: Evaluating the adaptive-linear time stepping scheme: time-adaptivity.

Table 4.4: Computational performance of the three-dimensional Cahn–Hilliard simulation. Less than 5% of the steps are rejected, while the time-step size changes nine orders of magnitude during the simulation.

Simulation time T	0.02
Total number of steps	1033
Linear iterations	85443
Rejected steps	30
Maximum Δt	1.422×10^{-4}
Minimum Δt	10^{-13}

is followed by a coarsening stage, that ends when the interfacial area is minimized. For the initial condition considered, the steady state is achieved when a single bubble remains, as seen in figure 4.17(d). The free energy evolution of the solution is shown in figure 4.18(a), and the evolution of the time step size is shown in 4.18(b). Less than 5% of the steps are rejected, while the time-step size changes nine orders of magnitude during the simulation, as seen in table 4.4.



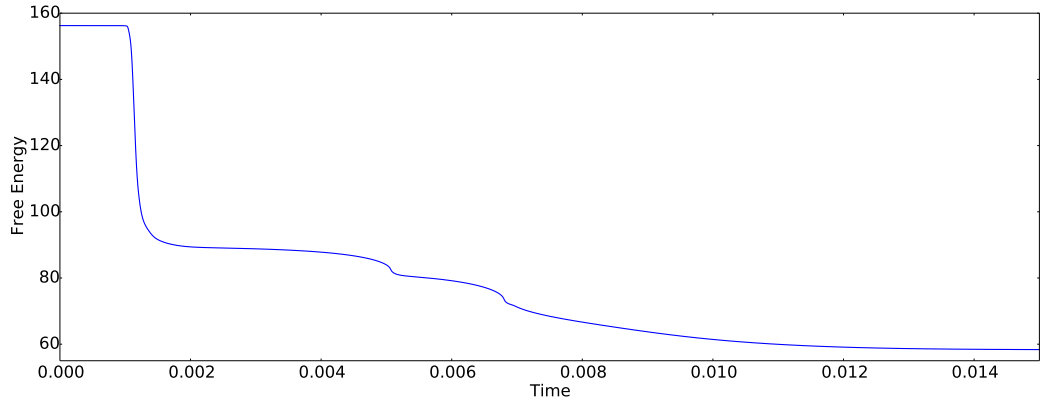
(a) $t = 0$

(b) $t \simeq 1.174 \times 10^{-3}$

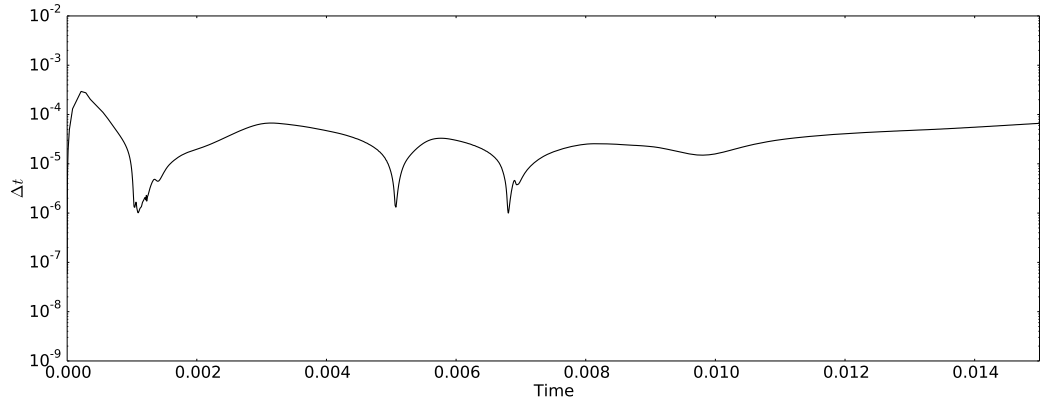
(c) $t \simeq 4.654 \times 10^{-3}$

(d) $t \simeq 2.118 \times 10^{-2}$

Figure 4.17: Three-dimensional simulation of the Cahn–Hilliard equation. System evolves from a randomly perturbed initial condition that results in a single bubble at steady state. This solution minimizes the surface area of the interface. The unit cube domain considered, $\bar{\Omega} = [0, 1]^3$, is meshed using $[64]^2\mathcal{C}^1$-cubic elements. The solution evolution is calculated with our linear-adaptive algorithm.



(a) Free energy evolution



(b) Evolution of the time step size

Figure 4.18: Three-dimensional simulation of the Cahn–Hilliard equation. The equation is solved using the linear-adaptive algorithm. The free energy is decreasing monotonically as seen in figure 4.18(a). The evolution of the time step size Δt its shown in figure 4.18(b), and seems reasonable: the fast changes in free energy match the times at which the time size decreases. The mesh is comprised of $[64]^3$-$\mathcal{C}^1$ cubic elements.

4.6 Discussion

We introduce an unconditionally energy-stable, second-order time accurate, mixed variational method for general classes of phase-field models. The numerical strategy relies on a mixed finite element formulation for the space discretization and on a second-order accurate algorithm based on Taylor expansions for the time discretization. This time-integration algorithm preserves mass by construction if needed, and guarantees strong energy stability. The class of time integrators we introduce is implicit by nature, and can be rendered linear to accelerate the solution process. Additionally, an adaptive time-stepping strategy that can be combined with our method is proposed, which overcomes some of the problems of adaptive time-stepping schemes used in the literature. The time step size is controlled by approximating a local truncation error a posteriori through a backward differentiation procedure. Several two-dimensional numerical examples are shown, for both uniform and adaptive time-step sizes, that deal with some of the more popular phase-field models available, such as the Allen–Cahn, Cahn–Hilliard, Swift–Hohenberg, and phase-field crystal equations. A three dimensional example involving the Cahn–Hilliard equation is also presented. The examples support our theoretical findings, and show the efficiency, stability and robustness of the new method, which is able to solve several high-order, nonlinear partial using the same setting. The implementation uses PetIGA, a high-performance isogeometric analysis framework, and the codes are freely available to download.

Chapter 5

Concluding Remarks

5.1 Summary

The main goal of this thesis was to develop thermodynamically-consistent algorithms for time integration of phase-field models. The numerical challenges faced when dealing with phase-field equations are common to several models. Thus, studying stable, efficient and robust algorithms for these high-order partial differential equations is an important goal in computational materials science.

Given the firm mathematical and physical footings of the diffuse-interface approach, detailed in chapter 2 through the derivation of the Cahn–Hilliard equation, the broad range of applications should not come as a surprise. The phase-field crystal equation, for example, arose as a possible solution to problems in molecular dynamics to study solidification phenomena. Chapter 3 describes a numerical method to solve the PFC equation that guarantees mass conservation, is energy-stable, and second order accurate in time. The method, based on a mixed finite element formulation, involves a system of three second-order equations. We ensure energy stability through a convex splitting of the nonlinear term present in the equation and the use of a stabilization parameter. Numerical examples implemented in our in-house,

high-performance finite element framework PetIGA showcase the utilization of the equation in simulating polycrystalline growth and the effect of the rotation angle on free energy.

We go further in chapter 4 and develop an implicit method that generalizes many methods published in literature on the topic of stable methods for both conserved and non-conserved phase-field models. We rely on Taylor expansions to accomplish this goal. What is more, we show how to render the system linear with the help of a stabilization parameter. We also develop an adaptive time-stepping strategy that overcomes many of the previously encountered problems in the literature. This approach relies on the approximation of a local truncation error a posteriori through a backward differentiation procedure to control the time step size. Several two-dimensional numerical examples are shown, for both uniform and adaptive time-step sizes, that deal with some of the more popular phase-field models available, such as the Allen–Cahn, Cahn–Hilliard, Swift–Hohenberg and phase-field crystal equations.

It is common practice in research groups to write code from scratch. Still, reusable software should be the norm in science. We hope PetIGA and our codes help the community in this way, as they are freely available to download [113]. As knowledge of physical phenomena keeps on being pushed forward, technological applications will demand accurate and robust algorithms able to handle complicated geometries and boundary conditions. Through this work, we hope to have addressed some of the challenges encountered when tackling phase-field problems.

5.2 Future work

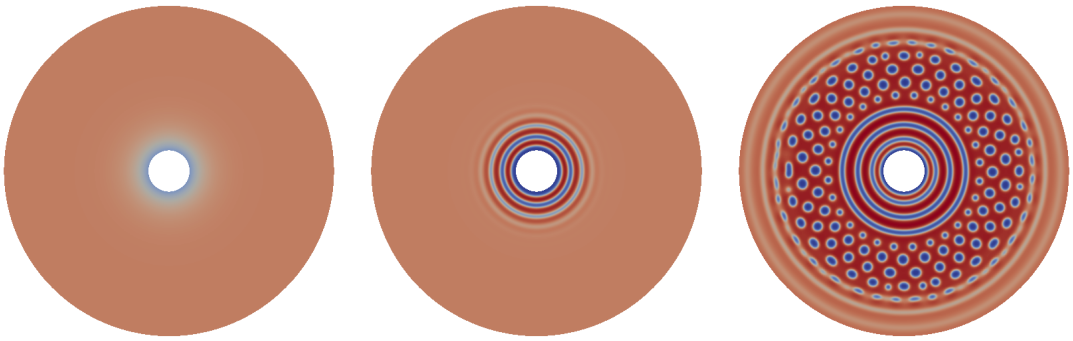
Even though the discretization method developed in this work is robust, challenges remain. As shown in chapters 3 and 4, ensuring energy stability requires the use of a stabilization parameter, whose effect on the solution remains unclear. Although we

consider the simplified version of the polynomial approximation to the potential, a connection with the logarithmic potential exists (see section 4.5.5). Also missing is work comparing the different methods developed, as well as the ones available in the literature. The linear method, the clear winner when running overkill resolutions, loses accuracy as time step size increases. We hope to answer these questions in the future [81, 132].

Given that our method is based on a midpoint approximation, it does not introduce any numerical dissipation. Nonetheless, this feature may be critical depending on the problem. Fortunately, our scheme can be modified to include it [80].

The time-adaptive strategy proposed needs to be studied further too. It possesses many advantages over current methods. Given that we can use it in conjunction with any second-order accurate method translates to a broad range of applicability. Our time-adaptive method is by no means limited to phase-field models.

Lastly, we would like to incorporate these algorithms into a real experiment, and have found a good candidate in [77]. In this work, the authors report a transition from rings to spots with hexagonal symmetry in a periodic precipitation system, which consists of sulfide/hydroxide ions diffusing into a gel matrix containing dissolved cadmium(II) ions. The authors show that a scenario analogous to the Cahn-Hilliard equation can capture the experimental results. However, this model can be improved, and the phase-field framework can be used to understand some of the new physics behind the experimental system. Some preliminary results solving this problem are shown in figure 5.1.



(a) (b) (c)

Figure 5.1: Representing a diffusion-reation problem through the Cahn-Hilliard equation. The phase-field, representing a rescaled concentration, is used to represent the precipitate that gets formed when the diffusing solvent reacts with the gel support. As can be seen, rings emerge and give way to spots in the domain.

REFERENCES

[1] H. Emmerich, *The Diffuse Interface Approach in Materials Science: Thermodynamic Concepts and Applications of Phase-Field Models*. Springer Berlin Heidelberg, 2003, vol. 73.

[2] H. Gómez and T. Hughes, "Provably unconditionally stable, second-order time-accurate, mixed variational methods for phase-field models," *Journal of Computational Physics*, vol. 230, no. 13, pp. 5310 – 5327, 2011.

[3] H. Gómez and X. Nogueira, "An unconditionally energy-stable method for the phase field crystal equation," *Computer Methods in Applied Mechanics and Engineering*, vol. 249-252, no. 0, pp. 52–61, 2012.

[4] ——, "A new spacetime discretization for the Swift-Hohenberg equation that strictly respects the lyapunov functional," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 12, pp. 4930 – 4946, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1007570412002286

[5] P. Vignal, L. Dalcin, D. Brown, N. Collier, and V. Calo, "An energy-stable convex splitting for the phase-field crystal equation," *Computers & Structures*, vol. 158, pp. 355–368, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045794915001753

[6] J. N. Israelachvili, "15 - solvation, structural, and hydration forces," in *Intermolecular and Surface Forces (Third Edition)*, 3rd ed. San Diego: Academic Press, 2011, pp. 341–380. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780123751829100156

[7] J. Contreras-García, E. R. Johnson, S. Keinan, R. Chaudret, J.-P. Piquemal, D. N. Beratan, and W. Yang, "Nciplot: A program for plotting noncovalent interaction regions," *Journal of Chemical Theory and Computation*, vol. 7, no. 3, pp. 625–632, 2011, pMID: 21516178. [Online]. Available: http://dx.doi.org/10.1021/ct100641a

[8] B. J. Alder and T. E. Wainwright, "Studies in Molecular Dynamics. I. General Method," *The Journal of Chemical Physics*, vol. 31, no. 2, pp. 459–466, 1959. [Online]. Available: http://scitation.aip.org/content/aip/journal/jcp/31/2/10.1063/1.1730376

[9] G. Fitzgerald, J. DeJoannis, and M. Meunier, "1-Multiscale modeling of nanomaterials: recent developments and future prospects," in *Modeling, Characterization, and Production of Nanomaterials*, ser. Woodhead Publishing Series in Electronic and Optical Materials, V. K. T. Zhang, Ed. Woodhead Publishing, 2015, pp. 3–53. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9781782422280000016

[10] W. Callister, *Materials Science and Engineering: an introduction*, 5th ed. Wiley, 2007.

[11] J. Rowlinson, "Translation of j.d. van der Waals' the thermodynamik theory of capillarity under the hypothesis of a continuous variation of density," *Journal of Statistical Physics*, vol. 20, no. 2, pp. 197–200, 1979. [Online]. Available: http://dx.doi.org/10.1007/BF01011513

[12] H. Emmerich, H. Löwen, R. Wittkowski, T. Gruhn, G. Tóth, G. Tegze, and L. Gránásy, "Phase-field-crystal models for condensed matter dynamics on atomic length and diffusive time scales: an overview," *ArXiv e-prints*, Jul. 2012.

[13] M. Plapp, "Phase-field models," in *Multiphase Microfluidics: The Diffuse Interface Model*, ser. CISM Courses and Lectures, R. Mauri, Ed. Springer Vienna, 2012, vol. 538, pp. 129–175. [Online]. Available: http://dx.doi.org/10.1007/978-3-7091-1227-4_4

[14] N. Provatas and K. Elder, *Phase-Field Methods in Materials Science and Engineering*, 1st ed. Wiley-VCH, 2010.

[15] J. Stefan, "*Ü*ber die theorie der eisbildung," *insbesondere uber die Eisbildung im Polarmeere*, vol. 42, pp. 269–286, 1891.

[16] N. Moelans, B. Blanpain, and P. Wollants, "An introduction to phase-field modeling of microstructure evolution," *Calphad*, vol. 32, no. 2, pp. 268 – 294, 2008.

[17] M. Haataja, J. Mahon, N. Provatas, and F. Léonard, "Scaling of domain size during spinodal decomposition: Dislocation discreteness and mobility effects," *Applied Physics Letters*, vol. 87, no. 25, p. 251901, Dec. 2005.

[18] C. Hirt and B. Nichols, "Volume of fluid (VOF) method for the dynamics of free boundaries," *Journal of Computational Physics*, vol. 39, no. 1, pp. 201 – 225, 1981. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0021999181901455

[19] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations," *Journal of Computational Physics*, vol. 79, no. 1, pp. 12–49, 1988.

[20] D. Sun and W. Tao, "A coupled volume-of-fluid and level set (VOSET) method for computing incompressible two-phase flows," *International Journal of Heat and Mass Transfer*, vol. 53, no. 4, pp. 645–655, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0017931009005717

[21] M. Sussman and E. G. Puckett, "A Coupled Level Set and Volume-of-Fluid Method for Computing 3D and Axisymmetric Incompressible Two-Phase Flows," *Journal of Computational Physics*, vol. 162, no. 2, pp. 301–337, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999100965379

[22] G. Caginalp and X. Chen, "Convergence of the phase field model to its sharp interface limits," *European Journal of Applied Mathematics*, vol. null,

pp. 417–445, 8 1998. [Online]. Available: http://journals.cambridge.org/article_S0956792598003520

[23] K. R. Elder, M. Grant, N. Provatas, and J. M. Kosterlitz, "Sharp interface limits of phase–field models," *Phys. Rev. E*, vol. 64, p. 021604, Jul 2001. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevE.64.021604

[24] G. Barenblatt, *Scaling, Self-similarity, and Intermediate Asymptotics: Dimensional Analysis and Intermediate Asymptotics*. Cambridge University Press, 2009.

[25] Y. Efendiev and T. Hou, *Multiscale Finite Element Methods: Theory and Applications*, 1st ed. Springer-Verlag New York, 2009.

[26] L. Onsager, "Reciprocal relations in irreversible processes. i." *Phys. Rev.*, vol. 37, pp. 405–426, Feb 1931. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRev.37.405

[27] J. W. Cahn and J. E. Hilliard, "Free energy of a nonuniform system. I. interfacial free energy," *J. Chem. Phys*, vol. 28, no. 258, 1958.

[28] H. Gomez, V. Calo, Y. Bazilevs, and T. Hughes, "Isogeometric analysis of the Cahn–Hilliard phase-field model," *Computer Methods in Applied Mechanics and Engineering*, vol. 197, no. 49–50, pp. 4333–4352, 2008.

[29] X. Wu, G. van Zwieten, and K. van der Zee, "Stabilized second-order convex splitting schemes for Cahn-Hilliard models with application to diffuse-interface tumor-growth models," *Int. J. Numer. Meth. Biomed. Engng.*, vol. 30, no. 2, pp. 180–203, 2014. [Online]. Available: http://dx.doi.org/10.1002/cnm.2597

[30] K. Elder, M. Katakowski, M. Haataja, and M. Grant, "Modeling elasticity in crystal growth," *Phys. Rev. Lett.*, vol. 88, p. 245705, Jun 2002.

[31] K. Elder and M. Grant, "Modeling elastic and plastic deformations in nonequilibrium processing using phase field crystals," *Phys. Rev. E*, vol. 70, p. 051605, Nov 2004.

[32] W. Boettinger, J. Warren, C. Beckermann, and A. Karma, "Phase-field simulation of solidification," *Annual Review of Materials Research*, vol. 32, no. 1, pp. 163–194, 2002.

[33] G. Vilanova, I. Colominas, and H. Gomez, "Coupling of discrete random walks and continuous modeling for three-dimensional tumor-induced angiogenesis," *Computational Mechanics*, vol. 53, no. 3, pp. 449–464, 2014. [Online]. Available: http://dx.doi.org/10.1007/s00466-013-0958-0

[34] ——, "Capillary networks in tumor angiogenesis: From discrete endothelial cells to phase-field averaged descriptions via isogeometric analysis," *International Journal for Numerical Methods in Biomedical Engineering*, vol. 29, no. 10, pp. 1015–1037, 2013. [Online]. Available: http://dx.doi.org/10.1002/cnm.2552

[35] L. Cueto-Felgueroso and R. Juanes, "Macroscopic phase-field model of partial wetting: Bubbles in a capillary tube," *Phys. Rev. Lett.*, vol. 108, p. 144502, Apr 2012. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevLett.108.144502

[36] E. Merkurjev, T. Kosti, and A. Bertozzi, "An MBO scheme on graphs for classification and image processing," *SIAM Journal on Imaging Sciences*, vol. 6, no. 4, pp. 1903–1930, 2013.

[37] H. Gomez, L. Cueto-Felgueroso, and R. Juanes, "Three-dimensional simulation of unstable gravity-driven infiltration of water into a porous medium," *Journal of Computational Physics*, vol. 238, no. 0, pp. 217 – 239, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999112007498

[38] D. J. Eyre, "An unconditionally stable one-step scheme for gradient systems," 1997.

[39] O. Wodo and B. Ganapathysubramanian, "Computationally efficient solution to the Cahn-Hilliard equation: Adaptive implicit time schemes, mesh sensitivity analysis and the 3D isoperimetric problem," *Journal of Computational Physics*, vol. 230, no. 15, pp. 6037 – 6060, 2011.

[40] Z. Zhang, Y. Ma, and Z. Qiao, "An adaptive time-stepping strategy for solving the Phase Field Crystal model," *Journal of Computational Physics*, vol. 249, pp. 204–215, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.jcp.2013.04.031

[41] P. Yue, J. Feng, C. Liu, and J. Shen, "A diffuse–interface method for simulating two–phase flows of complex fluids," *Journal of Fluid Mechanics*, vol. 515, pp. 293–317, 9 2004. [Online]. Available: http://journals.cambridge.org/article_S0022112004000370

[42] Z. Hu, S. Wise, C. Wang, and J. Lowengrub, "Stable and efficient finite-difference nonlinear-multigrid schemes for the phase field crystal equation," *Journal of Computational Physics*, vol. 228, no. 15, pp. 5323 – 5339, 2009.

[43] F. Guillén-González and G. Tierra, "Second order schemes and time-step adaptivity for Allen–Cahn and Cahn–Hilliard models," *Computers & Mathematics with Applications*, vol. 68, no. 8, pp. 821–846, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0898122114003320

[44] S. Wise, C. Wang, and J. Lowengrub, "An Energy-Stable and Convergent Finite-Difference Scheme for the Phase Field Crystal Equation," *SIAM Journal on Numerical Analysis*, vol. 47, no. 3, pp. 2269–2288, 2009.

[45] L. Chen and J. Shen, "Applications of semi-implicit Fourier-spectral method to phase field equations," *Computer Physics Communications*, vol. 108, no. 23, pp. 147 – 158, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S001046559700115X

[46] G. Tegze, G. Bansel, G. Tóth, T. Pusztai, Z. Fan, and L. Gránásy, "Advanced operator splitting-based semi-implicit spectral method to solve the binary phase-field crystal equations with variable coefficients," *Journal of Computational Physics*, vol. 228, no. 5, pp. 1612 – 1623, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999108005858

[47] T. Hughes, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. New York: Dover, 2000.

[48] J. Cottrell, T. Hughes, and Y. Bazilevs, *Isogeometric Analysis: Toward Unification of CAD and FEA*. John Wiley & Sons, Ltd, 2009.

[49] L. Piegl and W. Tiller, *The NURBS Book*, ser. Monographs in Visual Communication. New York: Springer, 1995.

[50] J. Liu, L. Dedè, J. Evans, M. Borden, and T. Hughes, "Isogeometric analysis of the advective Cahn-Hilliard equation: Spinodal decomposition under shear flow," *Journal of Computational Physics*, vol. 242, pp. 321–350, 2013.

[51] P. Vignal, A. Sarmiento, A. Côrtes, L. Dalcin, and V. Calo, "Coupling Navier-Stokes and Cahn-Hilliard equations in a two-dimensional annular flow configuration," *Procedia Computer Science*, vol. 51, no. 0, pp. 934–943, 2015, international Conference On Computational Science, {ICCS} 2015 Computational Science at the Gates of Nature.

[52] P. Vignal, N. O. Collier, and V. M. Calo, "Phase Field Modeling Using PetIGA," *Procedia Computer Science*, vol. 18, no. 0, pp. 1614–1623, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050913004729

[53] N. Collier, D. Pardo, L. Dalcin, M. Paszynski, and V. Calo, "The cost of continuity: A study of the performance of isogeometric finite elements using direct solvers," *Computer Methods in Applied Mechanics and Engineering*, vol. 213–216, pp. 353–361, 2012.

[54] N. Collier, L. Dalcin, D. Pardo, and V. Calo, "The cost of continuity: Performance of iterative solvers on isogeometric finite elements," *SIAM Journal on Scientific Computing*, vol. 35, no. 2, pp. A767–A784, 2013.

[55] N. Collier, L. Dalcin, and V. Calo, "On the computational efficiency of isogeometric methods for smooth elliptic problems using direct solvers," *International Journal for Numerical Methods in Engineering*, vol. 100, no. 8, pp. 620–632, 2014.

[56] ——, "PetIGA: High-performance isogeometric analysis," *arxiv*, no. 1305.4452, 2013, http://arxiv.org/abs/1305.4452.

[57] A. Côrtes, P. Vignal, A. Sarmiento, D. García, N. Collier, L. Dalcin, and V. Calo, "Solving nonlinear, high-order partial differential equations using a high-performance isogeometric analysis framework," in *CARLA, CCIS 485*, 2014, pp. 236–247.

[58] D. Schillinger, M. J. Borden, and H. K. Stolarski, "Isogeometric collocation for phase-field fracture models," *Computer Methods in Applied Mechanics and Engineering*, vol. 284, no. 0, pp. 583–610, 2014, isogeometric Analysis Special Issue. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045782514003557

[59] H. Gomez, A. Reali, and G. Sangalli, "Accurate, efficient, and (iso)geometrically flexible collocation methods for phase-field models," *Journal of Computational Physics*, vol. 262, no. 0, pp. 153–171, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999113008528

[60] L. Gao and V. Calo, "Fast isogeometric solvers for explicit dynamics," *Computer Methods in Applied Mechanics and Engineering*, vol. 274, no. 0, pp. 19–41, 2014.

[61] L. Gao, "Kronecker Products on Preconditioning." [Online]. Available: http://hdl.handle.net/10754/303766

[62] M. Woźniak, K. Kuźnik, M. Paszyński, V. Calo, and D. Pardo, "Computational cost estimates for parallel shared memory isogeometric multi-frontal solvers," *Computers & Mathematics with Applications*, vol. 67, no. 10, pp. 1864–1883, 2014.

[63] M. Woźniak, M. Paszyński, D. Pardo, L. Dalcin, and V. M. Calo, "Computational cost of isogeometric multi-frontal solvers on parallel distributed memory machines," *Computer Methods in Applied Mechanics and Engineering*, vol. 284, pp. 971–987, 2015.

[64] W. C. R.B. Ballufi, S.M. Allen, *Kinetics of materials*. Wiley, 2005.

[65] S. Allen and J. Cahn, "A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening," *Acta Metallurgica*, vol. 27,

no. 6, pp. 1085–1095, 1979. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0001616079901962

[66] J. Swift and P. C. Hohenberg, "Hydrodynamic fluctuations at the convective instability," *Phys. Rev. A*, vol. 15, pp. 319–328, Jan 1977. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.15.319

[67] J. W. Cahn and J. E. Hilliard, "Free energy of a nonuniform system. III. Nucleation in a Two-Component Incompressible Fluid," *The Journal of Chemical Physics*, vol. 31, no. 3, pp. 688–699, 1959. [Online]. Available: http://scitation.aip.org/content/aip/journal/jcp/31/3/10.1063/1.1730447

[68] A. Bertozzi, S. Esedoglu, and A. Gillette, "Inpainting of Binary Images Using the Cahn-Hilliard Equation," *Image Processing, IEEE Transactions on*, vol. 16, no. 1, pp. 285–291, Jan 2007.

[69] S. Tremaine, "On the origin of irregular structure in saturn's rings," *The Astronomical Journal*, vol. 125, no. 2, p. 894, 2003. [Online]. Available: http://stacks.iop.org/1538-3881/125/i=2/a=894

[70] T. Ursell, "Cahn–Hilliard kinetics and spinodal decomposition in a diffuse system," 2007.

[71] D. Lee, J.-Y. Huh, D. Jeong, J. Shin, A. Yun, and J. Kim, "Physical, mathematical, and numerical derivations of the cahnhilliard equation," *Computational Materials Science*, vol. 81, pp. 216–225, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0927025613004801

[72] D. Gaskell, *Introduction to the Thermodynamics of Materials*, 5th ed., ser. Introduction to the Thermodynamics of Materials. Taylor & Francis, 2003. [Online]. Available: https://books.google.com.sa/books?id=my8hPO-JFPoC

[73] O. Penrose and P. Fife, "Thermodynamically consistent models of phase-field type for the kinetic of phase transitions," *Physica D: Nonlinear Phenomena*, vol. 43, no. 1, pp. 44 – 62, 1990.

[74] I. Loginova, G. Amberg, and J. gren, "Phase-field simulations of non-isothermal binary alloy solidification," *Acta Materialia*, vol. 49, no. 4, pp. 573 – 581, 2001.

[75] L. Goudenège, D. Martin, and G. Vial, "High Order Finite Element Calculations for the Cahn–Hilliard equation," *Journal of Scientific Computing*, vol. 52, no. 2, pp. 294–321, 2012. [Online]. Available: http://dx.doi.org/10.1007/s10915-011-9546-7

[76] A. Hawkins-Daarud, K. G. van der Zee, and J. T. Oden, "Numerical simulation of a thermodynamically consistent four-species tumor growth model," *International Journal for Numerical Methods in Biomedical Engineering*, vol. 28, no. 1, pp. 3–24, 2012. [Online]. Available: http://dx.doi.org/10.1002/cnm.1467

[77] M. Dayeh, M. Ammar, and M. Al-Ghoul, "Transition from rings to spots in a precipitation reaction-diffusion system," *RSC Adv.*, vol. 4, pp. 60 034–60 038, 2014. [Online]. Available: http://dx.doi.org/10.1039/C4RA11223G

[78] G. Tierra and F. Guillén-González, "Numerical Methods for Solving the Cahn–Hilliard Equation and Its Applicability to Related Energy-Based Models," *Archives of Computational Methods in Engineering*, vol. 22, no. 2, pp. 269–289, 2015. [Online]. Available: http://dx.doi.org/10.1007/s11831-014-9112-1

[79] R. Guo and Y. Xu, "Local Discontinuous Galerkin Method and High Order Semi-Implicit Scheme for the Phase Field Crystal Equation," *SIAM Journal on Scientific Computing*, vol. 38, no. 1, pp. A105–A127, 2016. [Online]. Available: http://dx.doi.org/10.1137/15M1038803

[80] A. Sarmiento, P. Vignal, L. Dalcin, and V. M. Calo, "Generalized-α method for the Swift–Hohenberg equation," 2016, in preparation.

[81] P. Vignal, L. Dalcin, D. Brown, N. Collier, and V. Calo, "Energy-stable time discretizations for the phase-field crystal equation," *in preparation*, 2016.

[82] J. Kim, K. Kang, and J. Lowengrub, "Conservative multigrid methods for Cahn–Hilliard fluids," *Journal of Computational Physics*, vol. 193, no. 2, pp. 511–543, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999103003966

[83] T. Takaki, T. Shimokawabe, M. Ohno, A. Yamanaka, and T. Aoki, "Unexpected selection of growing dendrites by very-large-scale phase-field simulation," *Journal of Crystal Growth*, vol. 382, no. 0, pp. 21 – 25, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022024813005241

[84] M. Kluge and J. Ray, "Velocity versus temperature relation for solidification and melting of silicon: A molecular-dynamics study," *Phys. Rev. B*, vol. 39, pp. 1738–1746, Jan 1989. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevB.39.1738

[85] L. Dalcin, N. Collier, P. Vignal, A. Côrtes, and V. Calo, "Petiga: A Framework for High-Performance Isogeometric Analysis," *Computer Methods in Applied Mechanics and Engineering*, 2015, submitted.

[86] A. Jaatinen and T. Ala-Nissila, "Extended phase diagram of the three-dimensional phase field crystal model," *Journal of Physics: Condensed Matter*, vol. 22, no. 20, p. 205402, 2010.

[87] A. Karma and W.-J. Rappel, "Phase-field method for computationally efficient modeling of solidification with arbitrary interface kinetics," *Phys. Rev. E*, vol. 53, pp. R3017–R3020, Apr 1996. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevE.53.R3017

[88] K. Elder, N. Provatas, J. Berry, P. Stefanovic, and M. Grant, "Phase-field crystal modeling and classical density functional theory of freezing," *Phys. Rev. B*, vol. 75, p. 064107, Feb 2007.

[89] N. Provatas, J. Dantzig, B. Athreya, P. Chan, P. Stefanovic, N. Goldenfeld, and K. Elder, "Using the phase-field crystal method in the multi-scale modeling of microstructure evolution," *Journal of Minerals*, vol. 59, no. 7, pp. 83–90, 2007. [Online]. Available: http://dx.doi.org/10.1007/s11837-007-0095-3

[90] E. Asadi and M. Asle Zaeem, "A review of quantitative phase-field crystal modeling of solidliquid structures," *JOM*, vol. 67, no. 1, pp. 186–201, 2015. [Online]. Available: http://dx.doi.org/10.1007/s11837-014-1232-4

[91] M. Elsey and B. Wirth, "A simple and efficient scheme for phase field crystal simulation," *European Series in Applied and Industrial Mathematics: Mathematical Modelling and Numerical Analysis*, vol. 47, pp. 1413–1432, 9 2013. [Online]. Available: http://www.esaim-m2an.org/action/article_S0764583X13000745

[92] X. Wu, G. van Zwieten, and K. van der Zee, "Stabilized second-order convex splitting schemes for Cahn-Hilliard models with application to diffuse-interface tumor-growth models," *Int. J. Numer. Meth. Biomed. Engng.*, vol. 30, no. 2, pp. 180–203, 2014. [Online]. Available: http://dx.doi.org/10.1002/cnm.2597

[93] T. Ramakrishnan and M. Yussouff, "First-principles order-parameter theory of freezing," *Phys. Rev. B*, vol. 19, pp. 2775–2794, Mar 1979.

[94] R. Backofen, A. Rätz, and A. Voigt, "Nucleation and growth by a phase field crystal (PFC) model," *Philosophical Magazine Letters*, vol. 87, no. 11, pp. 813–820, 2007.

[95] R. Backofen and A. Voigt, "A phase-field-crystal approach to critical nuclei," *Journal of Physics: Condensed Matter*, vol. 22, no. 36, p. 364104, 2010.

[96] G. Tóth, G. Tegze, T. Pusztai, G. Tóth, and L. Gránásy, "Polymorphism, crystal nucleation and growth in the phase-field crystal model in 2D and 3D," *Journal of Physics: Condensed Matter*, vol. 22, no. 36, p. 364101, 2010. [Online]. Available: http://stacks.iop.org/0953-8984/22/i=36/a=364101

[97] N. Guttenberg, N. Goldenfeld, and J. Dantzig, "Emergence of foams from the breakdown of the phase field crystal model," *Phys. Rev. E*, vol. 81, p. 065301, Jun 2010. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevE.81.065301

[98] J. Berry, K. Elder, and M. Grant, "Simulation of an atomistic dynamic field theory for monatomic liquids: Freezing and glass formation," *Phys. Rev. E*, vol. 77, p. 061506, Jun 2008. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevE.77.061506

[99] H. Löwen, "A phase-field-crystal model for liquid crystals," *Journal of Physics: Condensed Matter*, vol. 22, no. 36, p. 364105, 2010. [Online]. Available: http://stacks.iop.org/0953-8984/22/i=36/a=364105

[100] N. Pisutha-Arnond, V. Chan, K. Elder, and K. Thornton, "Calculations of isothermal elastic constants in the phase-field crystal model," *Phys. Rev. B*, vol. 87, p. 014103, Jan 2013. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevB.87.014103

[101] K.-A. Wu and A. Karma, "Phase-field crystal modeling of equilibrium bcc-liquid interfaces," *Phys. Rev. B*, vol. 76, p. 184107, Nov 2007.

[102] K.-A. Wu, A. Adland, and A. Karma, "Phase-field-crystal model for fcc ordering," *Phys. Rev. E*, vol. 81, no. 6, p. 061601, Jun. 2010.

[103] A. Jaatinen, C. Achim, K. Elder, and T. Ala-Nissila, "Thermodynamics of bcc metals in phase-field-crystal models," *Phys. Rev. E*, vol. 80, p. 031602, Sep 2009.

[104] A. Baskaran, Z. Hu, J. S. Lowengrub, C. Wang, S. M. Wise, and P. Zhou, "Energy stable and efficient finite-difference nonlinear multigrid schemes for the modified phase field crystal equation," *Journal of Computational Physics*, vol. 250, pp. 270–292, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999113002891

[105] E. Asadi, M. Asle Zaeem, and M. Baskes, "Phase-Field Crystal Model for Fe Connected to MEAM Molecular Dynamics Simulations," *JOM*, vol. 66, no. 3, pp. 429–436, 2014. [Online]. Available: http://dx.doi.org/10.1007/s11837-013-0845-3

[106] B. Wu, Q. Chen, and Z. Wang, "Uniqueness and stability of an inverse problem for a phase field model using data from one component," *Computers & Mathematics with Applications*, vol. 66, no. 10, pp. 2126 – 2138, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0898122113005464

[107] M. Jamshidian and T. Rabczuk, "Phase Field Modelling of Stressed Grain Growth: Analytical Study and the Effect of Microstructural Length Scale," *Journal of Computational Physics*, vol. 261, pp. 23–35, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.jcp.2013.12.022

[108] P. Thamburaja and M. Jamshidian, "A multiscale taylor model-based constitutive theory describing grain growth in polycrystalline cubic metals," *Journal of the Mechanics and Physics of Solids*, vol. 63, no. 0, pp. 1 – 28, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022509613002172

[109] Y. Bazilevs, L. Beirão Da Veiga, J. A. Cottrell, T. J. R. Hughes, and G. Sangalli, "Isogeometric Analysis: Approximation, stability and error estimates for h-refined meshes," *Mathematical Models and Methods in Applied Sciences*, vol. 16, no. 07, pp. 1031–1090, 2006. [Online]. Available: http://www.worldscientific.com/doi/abs/10.1142/S0218202506001455

[110] V. Calo, H. Gómez, Y. Bazilevs, G. Johnson, and T. Hughes, "Simulation of engineering applications using isogeometric analysis," *TeraGrid*, 2008.

[111] M. Borden, C. Verhoosel, M. Scott, T. Hughes, and C. Landis, "A phase-field description of dynamic brittle fracture," *Computer Methods in Applied Mechanics and Engineering*, vol. 217, pp. 77–95, 2012.

[112] P. Vignal, L. Dalcin, N. O. Collier, and V. M. Calo, "PetIGA: Solution of higher-order partial differential equations," in *4th International Congress of Serbian Society of Mechanics*, 2013, pp. 71–80.

[113] L. Dalcin and N. Collier. (2015) PetIGA: High performance isogeometric analysis. https://bitbucket.org/dalcinl/petiga. [Online]. Available: https://bitbucket.org/dalcinl/petiga

[114] S. Balay, K. Buschelman, W. Gropp, D. Kaushik, M. Knepley, L. C. McInnes, B. Smith, and H. Zhang, "PETSc Web page," 2014, http://www.mcs.anl.gov/petsc.

[115] S. Balay, K. Buschelman, V. Eijkhout, W. Gropp, D. Kaushik, M. Knepley, L. C. McInnes, B. Smith, and H. Zhang, "PETSc users manual," Argonne National Laboratory, Tech. Rep. ANL-95/11 - Revision 3.4, 2014.

[116] N. Collier and L. Dalcin. (2013, Sep.) PetIGA and igakit tutorial. https://petiga-igakit.readthedocs.org/en/latest/.

[117] W. Levine, *The Control Handbook*, 2nd ed. CRC Press, 2010.

[118] L. F. R. Espath, A. Sarmiento, P. Vignal, B. Varga, A. M. A. Cortes, L. Dalcin, and V. M. Calo, "Energy Exchange Analysis in Droplet Dynamics via the Navier–Stokes–Cahn–Hilliard Model," *Journal of Fluid Mechanics*, 2015, submitted.

[119] M. Copetti and C. Elliott, "Numerical analysis of the Cahn–Hilliard equation with a logarithmic free energy," *Numerische Mathematik*, vol. 63, no. 1, pp. 39–65, 1992. [Online]. Available: http://dx.doi.org/10.1007/BF01385847

[120] M. Peletier, "Energies, gradient flows, and large deviations: a modelling point of view," 2012. [Online]. Available: http://www.win.tue.nl/~mpeletie/Onderwijs/Pisa2011/PeletierLectureNotesPisa2011.pdf

[121] J. Stewart, "Additional topics: Formulas for the Remainder Term in Taylor series," in *Calculus: Early Trascendentals*, 8th ed. Brooks Cole, 2015.

[122] M. Kästner, P. Metsch, and R. de Borst, "Isogeometric analysis of the Cahn–Hilliard equation - a convergence study," *Journal of Computational Physics*, pp. –, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S002199911500724X

[123] E. Hairer, S. Norsett, and G. Wanner, *Solving Ordinary Differential Equations: Nonstiff problems. 1*, ser. Solving ordinary differential equations. Springer London, Limited, 1987. [Online]. Available: https://books.google.com.sa/books?id=xmA5QgAACAAJ

[124] R. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems (Classics in Applied*

Mathematics Classics in Applied Mathemat). Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2007.

[125] K. Jansen, C. Whiting, and G. Hulbert, "A generalized-α method for integrating the filtered Navier–Stokes equations with a stabilized finite element method," *Computer Methods in Applied Mechanics and Engineering*, vol. 190, no. 3–4, pp. 305–319, 2000.

[126] F. H. Fenton, E. M. Cherry, A. Karma, and W.-J. Rappel, "Modeling wave propagation in realistic heart geometries using the phase-field method," *Chaos*, vol. 15, no. 1, pp. –, 2005. [Online]. Available: http://scitation.aip.org/content/aip/journal/chaos/15/1/10.1063/1.1840311

[127] T. Hughes, J. Cottrell, and Y. Bazilevs, "Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement," *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 39–41, pp. 4135–4195, 2005.

[128] Y. Li, H. G. Lee, D. Jeong, and J. Kim, "An unconditionally stable hybrid numerical method for solving the Allen–Cahn equation," *Computers & Mathematics with Applications*, vol. 60, no. 6, pp. 1591–1606, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0898122110004554

[129] O. Zienkiewicz, R. Taylor, and J. Zhu, "Chapter 12 - the time dimension: Semi-discretization of field and dynamic problems," in *The Finite Element Method: its Basis and Fundamentals (Seventh Edition)*, 7th ed., O. Zienkiewicz, R. Taylor, and J. Zhu, Eds. Oxford: Butterworth-Heinemann, 2013, pp. 379–405. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9781856176330000125

[130] H. G. Lee, J. Shin, and J.-Y. Lee, "First and second order operator splitting methods for the phase field crystal equation," *Journal of Computational Physics*, vol. 299, pp. 82–91, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999115004362

[131] S. Aland, "Time integration for diffuse interface models for two-phase flow," *Journal of Computational Physics*, vol. 262, pp. 58–71, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999114000102

[132] P. Vignal, L. Dalcin, N. Collier, and V. Calo, "High-performance isogeometric analysis of energy-stable time integrators for cahn–hilliard models," *in preparation*, 2016.

[133] R. Yokota, J. Pestana, H. Ibeid, and D. Keyes, "Fast multipole preconditioners for sparse matrices arising from elliptic equations," *Computing Research Repository*, pp. 1–32., 2014.

[134] N. Collier, A.-L. Haji-Ali, F. Nobile, E. von Schwerin, and R. Tempone, "A continuation multilevel Monte Carlo algorithm," *BIT Numerical Mathematics*, vol. 55, no. 2, pp. 399–432, 2015.

[135] H. Casquero, C. Bona-Casas, and H. Gomez, "A NURBS-based immersed methodology for fluid-structure interaction," *Computer Methods in Applied Mechanics and Engineering*, vol. 284, no. 0, pp. 943–970, 2015, isogeometric Analysis Special Issue.

[136] S. Rudraraju, A. V. der Ven, and K. Garikipati, "Three-dimensional isogeometric solutions to general boundary value problems of toupins gradient elasticity theory at finite strains," *Computer Methods in Applied Mechanics and Engineering*, vol. 278, no. 0, pp. 705–728, 2014.

[137] M. Bercovier and G. Berold, "Solving design problems by integration of CAD and FEM software," in *Proceedings of IFIP TC 2/WG 2.5 Working Conference on Problem Solving Environments for Scientific Computing, Sophia Antipolis, France, 17–21 June, 1985*, B. Ford and F. Chatelin, Eds. North Holland, 1987, p. 309.

[138] A. Sheffer, T. Blacker, and M. Bercovier, "Steps towards smooth CAD–FEM integration," in *Proceedings of 6th International Conference on Numerical Grid Generation in Computational Field Simulations*, 1998, pp. 705–714.

[139] A. Sheffer and M. Bercovier, "CAD model editing and its applications," Ph.D. dissertation, Hebrew University of Jerusalem, 1999.

[140] K. Höllig, *Finite Element Methods with B-splines*, ser. Frontiers in Applied Mathematics. Society for Industrial and Applied Mathematics, 2003.

[141] H. Gomez, T. Hughes, X. Nogueira, and V. Calo, "Isogeometric analysis of the isothermal Navier–Stokes–Korteweg equations," *Computer Methods in Applied Mechanics and Engineering*, vol. 199, no. 25–28, pp. 1828–1840, 2010.

[142] D. Benson, Y. Bazilevs, M.-C. Hsu, and T. Hughes, "A large deformation, rotation-free, isogeometric shell," *Computer Methods in Applied Mechanics and Engineering*, vol. 200, no. 13–16, pp. 1367–1378, 2011.

[143] J. Kiendl, M.-C. Hsu, M. Wu, and A. Reali, "Isogeometric Kirchhoff–Love shell formulations for general hyperelastic materials," *Computer Methods in Applied Mechanics and Engineering*, vol. 291, no. 0, pp. 280–303, 2015.

[144] R. Bouclier, T. Elguedj, and A. Combescure, "An isogeometric locking–free NURBS-based solid–shell element for geometrically nonlinear analysis," *International Journal for Numerical Methods in Engineering*, vol. 101, no. 10, pp. 774–808, 2015.

[145] I. Akkerman, Y. Bazilevs, V. Calo, T. Hughes, and S. Hulshoff, "The role of continuity in residual-based variational multiscale modeling of turbulence," *Computational Mechanics*, vol. 41, no. 3, pp. 371–378, 2008.

[146] I. Akkerman, Y. Bazilevs, C. Kees, and M. Farthing, "Isogeometric analysis of free-surface flow," *Journal of Computational Physics*, vol. 230, no. 11, pp. 4137–4152, 2011, special issue High Order Methods for CFD Problems.

[147] A. Buffa, J. Rivas, G. Sangalli, and R. Vázquez, "Isogeometric discrete differential forms in three dimensions," *SIAM Journal on Numerical Analysis*, vol. 49, no. 2, pp. 818–844, 2011.

[148] J. Evans and T. Hughes, "Isogeometric divergence-conforming B-splines for the unsteady Navier–Stokes equations," *Journal of Computational Physics*, vol. 241, pp. 141–167, 2013.

[149] L. B. da Veiga, A. Buffa, J. Rivas, and G. Sangalli, "Some estimates for h–p–k-refinement in Isogeometric Analysis," *Numerische Mathematik*, vol. 118, pp. 271–305, 2011.

[150] J. Evans, Y. Bazilevs, I. Babuška, and T. Hughes, "n-Widths, sup–infs, and optimality ratios for the k-version of the isogeometric finite element method," *Computer Methods in Applied Mechanics and Engineering*, vol. 198, no. 21–26, pp. 1726–1741, 2009, advances in Simulation-Based Engineering Sciences – Honoring J. Tinsley Oden.

[151] S. Balay, S. Abhyankar, M. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. Gropp, D. Kaushik, M. Knepley, L. C. McInnes, K. Rupp, B. Smith, S. Zampini, and H. Zhang. (2015) PETSc Web page. http://www.mcs.anl.gov/petsc. [Online]. Available: http://www.mcs.anl.gov/petsc

[152] ——, "PETSc users manual," Argonne National Laboratory, Tech. Rep. ANL-95/11 - Revision 3.6, 2015. [Online]. Available: http://www.mcs.anl.gov/petsc/petsc-current/docs/manual.pdf

[153] S. Balay, W. Gropp, L. C. McInnes, and B. Smith, "Efficient management of parallelism in object oriented numerical software libraries," in *Modern Software Tools in Scientific Computing*, E. Arge, A. Bruaset, and H. Langtangen, Eds. Birkhäuser Press, 1997, pp. 163–202.

[154] R. Falgout and U. Yang, "*hypre*: A library of high performance preconditioners," in *Computational Science – ICCS 2002*, ser. Lecture Notes in Computer Science, P. Sloot, A. Hoekstra, C. K. Tan, and J. Dongarra, Eds. Springer Berlin Heidelberg, 2002, vol. 2331, pp. 632–641.

[155] M. Heroux, R. Bartlett, V. Howle, R. Hoekstra, J. Hu, T. Kolda, R. Lehoucq, K. Long, R. Pawlowski, E. Phipps, A. Salinger, H. Thornquist, R. Tuminaro, J. Willenbring, A. Williams, and K. Stanley, "An overview of the Trilinos

project," *ACM Transactions on Mathematical Software*, vol. 31, no. 3, pp. 397–423, 2005.

[156] P. Amestoy, A. Guermouche, J.-Y. L'Excellent, and S. Pralet, "Hybrid scheduling for the parallel solution of linear systems," *Parallel Computing*, vol. 32, no. 2, pp. 136–156, 2006, parallel Matrix Algorithms and Applications.

[157] W. Bangerth, R. Hartmann, and G. Kanschat, "deal.II – a general purpose object oriented finite element library," *ACM Transactions on Mathematical Software*, vol. 33, no. 4, pp. 24/1–24/27, 2007.

[158] A. Logg, K.-A. Mardal, and G. Wells, *Automated Solution of Differential Equations by the Finite Element Method*. Springer Berlin Heidelberg, 2012.

[159] B. Kirk, J. Peterson, R. Stogner, and G. Carey, "`libMesh`: A C++ Library for Parallel Adaptive Mesh Refinement/Coarsening Simulations," *Engineering with Computers*, vol. 22, no. 3–4, pp. 237–254, 2006.

[160] V. Sonzogni, A. Yommi, N. Nigro, and M. Storti, "A parallel finite element program on a beowulf cluster," *Advances in Engineering Software*, vol. 33, no. 7–10, pp. 427–443, 2002, engineering Computational Technology & Computational Structures Technology.

[161] J. Jacobsen and K. Schmitt, "The Liouville-Bratu-Gelfand Problem for Radial Operators," *Journal of Differential Equations*, vol. 184, no. 1, pp. 283–298, 2002.

[162] W. Schroeder, K. Martin, and B. Lorensen, *The Visualization Toolkit, 4th edition*. Kitware Inc., 2006.

[163] "VTK File Formats," www.vtk.org/VTK/img/file-formats.pdf, 2015.

APPENDICES

Appendix A

A.1 PetIGA

PetIGA is an in-house, scalable implementation of isogeometric analysis for linear/nonlinear and static/transient problems [85, 113]. The framework, which is built on top of PETSc, gives users a robust and versatile platform to solve partial differential equations. The framework scales well on thousands of cores, and is well suited for large scale applications. Even though primarily conceived for distributed-memory computing environments, PetIGA is also able to extract excellent performance on nowadays shared memory multicore laptop and desktop computers. Our software is already being used by members of the community to tackle challenging problems related to parallel multifrontal direct solvers [62], fast multipole-based preconditioners for elliptic equations [133], multilevel Monte Carlo algorithms geared towards the approximation of stochastic models [134], fluid-structure interaction [135], and finite strain gradient elasticity [136].

The numerical examples presented in this work are all implemented in PetIGA, and can be found online.[1]

[1] https://bitbucket.org/dalcinl/petiga

A.1.1 Motivation for the software

Isogeometric analysis, a finite element method originally proposed in 2005 [127, 48], was originally motivated by the desire to find a technique for solving partial differential equations which would simplify or remove the problem of converting geometric descriptions for discretizations in the engineering design process. Once a design is born inside a Computer Aided Design (CAD) program, converting the CAD representation to an analysis-suitable form usually is a bottleneck of the engineering analysis process. Isogeometric methods aim to use CAD representations directly by using the Non-Uniform Rational B-spline (NURBS) basis, circumventing the need to generate an intermediate geometrical description. The term *isogeometric* reflects that as the finite element space is refined, the geometrical representation can be preserved exactly. NURBS technologies have been used in CAD for decades due to their properties, particularly the smoothness and ability to represent conic sections. The key insight of isogeometric analysis is to use the geometrical map of the NURBS representation as a basis for the push forward used in analysis. This allows isogeometric modeling to advance where predescessors have found limitations [137, 138, 139, 140].

In addition to the geometrical benefits, the basis is also well-suited to solving higher-order partial differential equations, such as the ones related to phase-field problems [28, 141, 5] or large deformation shell formulations [142, 143, 144]. Classical finite element spaces use basis functions which are C^0 continuous across element boundaries, making them unsuitable for higher-order problems using a primal Galerkin formulation. The NURBS-based spaces may be constructed to possess arbitrary degrees of inter-element continuity for any spatial dimension. These higher-order continuous basis functions have been numerically [145, 146, 147, 148] and theoretically [149, 150] observed to possess superior approximability per degree of freedom when compared to their C^0 counterparts. However, when used to discretize a Galerkin weak form,

the higher-order continuous basis functions have also been shown to result in linear systems which are more expensive to solve with multifrontal direct solvers [55, 53] and iterative solvers [54]. These results motivate the development of efficient, scalable software frameworks which can mitigate the increase of cost.

A.1.2 PETSc and PetIGA

PETSc [151, 152, 153], the *Portable Extensible Toolkit for Scientific Computation*, is a collection of algorithms and data structures for the solution of scientific problems, particularly those modeled by partial differential equations. PETSc is applicable to a wide range of problem sizes, including extreme large-scale simulations, where high-performance parallel computation is a must. PETSc uses the message-passing interface (MPI) model for communication, but provides high-level interfaces with collective semantics so that typical users rarely have to make message-passing calls directly.

PETSc provides a rich environment for modeling scientific problems as well as for rapid algorithm design and prototyping. The library enables easy customization and extension of both algorithms and implementations. This approach promotes code reuse and flexibility. PETSc is object-oriented in style, with components that may be changed via a command-line interface at runtime. These components include:

- Index sets to describe permutations, indexing, renumbering, and communication patterns;
- Matrices and vectors that provide basic linear algebra abstractions;
- Krylov subspace methods and preconditioners that include multigrid and sparse direct solvers;
- Nonlinear solvers and time stepping algorithms; and

- Distributed arrays for parallelizing structured grid-based problems.

PETSc is also designed to be highly modular, enabling the interoperability with specialized parallel libraries like Hypre [154], Trilinos/ML [155], MUMPS [156], and others through a unified interface. Other scientific packages geared towards solving partial differential equations use components from PETSc (for example deal.II [157], FEniCS [158], libMesh [159], and PETSc-FEM [160]).

PetIGA reuses PETSc algorithms and data structures to obtain a high-performance framework designed for isogeometric analysis. Parallel vector and matrix assembly are implemented within PetIGA using PETSc data structures and interface into PETSc's wide range of solvers. More details on implementation, applications, and performance can be found in [85]. The software is freely available [113] and under active development.

A.1.3 Solving the Bratu equation with PetIGA

In this section we solve the Bratu equation as a model application to highlight some of the useful features users have access to when using our framework. The Bratu equation is a nonlinear second-order boundary value problem [161]. The strong form of the equation can be stated as: find $u : \bar{\Omega} \to \mathbb{R}$ such that

$$-\Delta u = \lambda \exp(u), \quad \boldsymbol{x} \in \Omega, \tag{A.1}$$

$$u = 0, \qquad \boldsymbol{x} \in \partial\Omega, \tag{A.2}$$

where $\bar{\Omega}$ is the unit square in $[0,1]^2$, $\partial\Omega$ denotes the domain boundary, Δ represents the Laplace operator, $u \equiv u(\boldsymbol{x})$ is a scalar field defined in Ω and λ is a positive constant. No solution exists when λ goes above $\lambda_{max} = 6.80812$ as the equation possesses a bifurcation point. This equation models the steady-state of a nonlinear reaction and heat conduction problem, and results from a simplification of the solid-

fuel ignition model.

Within the framework of Galerkin finite elements, let $\mathcal{V}$ denote the trial and weighting function spaces, where $\mathcal{V}$ belongs to $\mathcal{H}_0^1$, i.e., the Sobolev space of square-integrable functions with square-integrable first derivatives and zero value on $\partial\Omega$. The weak form is obtained by multiplying equation (A.1) by a test function w and integrating by parts. The variational problem can then be defined as that of finding $u \in \mathcal{V}$ such that for all $w \in \mathcal{V}$

$$(\nabla w, \nabla u)_\Omega - (w, \lambda \exp(u))_\Omega = 0, \tag{A.3}$$

where $(\cdot,\cdot)_\Omega$ denotes the $\mathcal{L}^2$ inner product on domain Ω. The finite-dimensional problem can then be formulated as: find $u^h \in \mathcal{V}^h$, where $\mathcal{V}^h \subset \mathcal{V}$, such that for all $w^h \in \mathcal{V}^h$

$$\left(\nabla w^h, \nabla u^h\right)_\Omega - \left(w^h, \lambda \exp\left(u^h\right)\right)_\Omega = 0, \tag{A.4}$$

where w^h, u^h and their respective gradients are defined as the linear combinations

$$w^h = \sum_A W_A N_A(\boldsymbol{x}), \qquad u^h = \sum_A U_A N_A(\boldsymbol{x}), \tag{A.5}$$

$$\nabla w^h = \sum_A W_A \nabla N_A(\boldsymbol{x}), \qquad \nabla u^h = \sum_A U_A \nabla N_A(\boldsymbol{x}), \tag{A.6}$$

where N_A are the basis functions and W_A, U_A are the control variables. Denoting $\mathbf{U} = \{U_A\}$ the vector of coefficients, we define the residual vector as $\mathbf{R}(\mathbf{U}) = \{R_A\}$, where R_A is obtained from (A.4)–(A.6) as

$$R_A = \left(\nabla N_A, \nabla u^h\right)_\Omega - \left(N_A, \lambda \exp\left(u^h\right)\right)_\Omega, \tag{A.7}$$

As the problem is nonlinear, solving it with Newton's method requires the specifi-

cation of the Jacobian $\mathbf{J} = \partial\mathbf{R}/\partial\mathbf{U}$. In this particular problem, its entries are defined as

$$J_{AB} = (\nabla N_A, \nabla N_B)_\Omega - \left(N_A, \lambda \exp\left(u^h\right) N_B\right)_\Omega . \tag{A.8}$$

A.1.4 Implementation

To code a program within the framework, the user first needs to include the PetIGA C header file

```
#include <petiga.h>
```

For the sake of simplicity, two macros are defined to set the dimensionality of the problem to two and implement the dot product of two-vectors

```
#define dim 2
#define dot(a,b) (a[0]*b[0]+a[1]*b[1])
```

To change the dimensionality of the problem to one or three, these two lines should be modified accordingly. Then, a C structure is used to handle the problem-specific parameter λ. This approach is preferred to the alternative of using global variables.

```
typedef struct {
  double lambda;
} Params;
```

The residual routine implementing equation (A.7) reads

```
int Residual(IGAPoint p,const double U[],double R[],void *ctx) {
  int    a,nen      = p->nen;
  double (*N0)      = (typeof(N0)) p->shape[0];
  double (*N1)[dim] = (typeof(N1)) p->shape[1];
  double u,grad_u[dim],lambda = ((Params*)ctx)->lambda;
  IGAPointFormValue(p,U,&u);
  IGAPointFormGrad (p,U,grad_u);
  for (a=0; a<nen; a++)
    R[a] = dot(N1[a],grad_u) - lambda*exp(u)*N0[a];
  return 0;
}
```

The residual routine constitutes the integrand to be evaluated at each quadrature point to compute local contributions to the global residual vector. The `Residual` routine has the following arguments

- an input pointer `p`, of type `IGAPoint`, used as a quadrature point context holding discretization data required to perform the residual evaluation,
- an input floating point array `U`, which contains the local control variables gathered from the global vector $\mathbf{U}$, i.e., the coefficients corresponding to basis functions whose support contains the quadrature point, recall equations (A.5) and (A.6),
- an output floating point array `R`, where the routine is expected to return local contributions to be assembled in the global residual vector $\mathbf{R}$, recall equation (A.7),
- an input opaque pointer `ctx`, used to pass problem-specific information to the residual routine.

The code proceeds to declare the following local variables

- two integers `a` and `nen`, the first to be used as a loop index, while the second is initialized to the number of local basis functions,
- two pointers `N0` and `N1`, initialized from data within the `IGAPoint`, used in the following for convenient access to the values of local basis functions (0th derivatives) and their first derivatives,
- two floating point variables `u` and `grad_u` to store the values of u^h and ∇u^h at the quadrature point
- a floating point variable `lambda` to store the value of λ, initialized from the `Params` structure through the opaque pointer `ctx`.

Next, the routines `IGAPointFormValue` and `IGAPointFormGrad` are invoked to compute the values of u^h and ∇u^h at the quadrature point, following equations (A.5) and (A.6). Finally, local residual contributions are computed in a loop following equation (A.7)

by using the previously defined `dot` macro as well as the `exp` routine from the standard library of the C programming language. The quadrature weights and Jacobian determinant of the geometry mapping are handled internally, which slightly simplifies the coding.

The Jacobian routine implementing equation (A.8) reads

```
22 int Jacobian(IGAPoint p,const double U[],double J[],void *ctx) {
23   int    a,b,nen    = p->nen;
24   double (*N0)      = (typeof(N0)) p->shape[0];
25   double (*N1)[dim] = (typeof(N1)) p->shape[1];
26   double u,lambda = ((Params*)ctx)->lambda;
27   IGAPointFormValue(p,U,&u);
28   for (a=0; a<nen; a++)
29     for (b=0; b<nen; b++)
30       J[a*nen+b] = dot(N1[a],N1[b]) - lambda*exp(u)*N0[a]*N0[b];
31   return 0;
32 }
```

The `Jacobian` routine is almost the same as the previous `Residual` routine. The main differences are the output floating point array `J` where the routine is expected to return local contributions to be assembled in the global Jacobian matrix **J**, and the double-loop computing these contributions following equation (A.8).

The code of the main program routine then begins

```
34 int main(int argc, char *argv[]) {
35   PetscInitialize(&argc,&argv,NULL,NULL);
```

The first statement invokes the `PetscInitialize` routine, which internally handles the initialization of the PETSc library. This routine is fed with the command line arguments to the program. PETSc uses these arguments to build a database of options. These options are queried later.

The code proceeds to create and initialize an `iga` object of type `IGA`. The `IGA` type is a key component in PetIGA. This core data structure contains all the information related to discretization and parallel communication.

```
37   IGA iga;
38   IGACreate(PETSC_COMM_WORLD,&iga);
39   IGASetDim(iga,dim);
40   IGASetDof(iga,1);
41   IGASetFromOptions(iga);
```

```
42 ||  IGASetUp(iga);
```

To properly setup the `iga` object, the user only needs to hardwire in code a couple of problem-specific parameters, and ask the framework to handle the rest at runtime through the options database

- the routine `IGASetDim` specifies the number of space dimensions, in this particular example it is set to two,
- the routine `IGASetDof` specifies the number of components in the solution, in this particular example it is set to one as the code is dealing with a scalar problem,
- the routine `IGASetFromOptions` queries the options database to let users change different properties of the discretization such as number of elements, polynomial degree and regularity of the approximation space, type of basis functions to use, quadrature rules, number of quadrature points, among others. When values are not specified by the user, PetIGA selects default values such as 16 elements per direction, unit domain, uniform refinement, quadratic C^1 spaces, and Gauss–Legendre quadrature. Even though other ways to initialize an IGA object are available, this is the simplest one. When more complex discretizations and/or geometries are required, the IGA object can be initialized by loading binary datafiles,
- finally, the routine `IGASetUp` prepares the data structure to be used in what follows. This step handles the parallel partitioning of the domain and prepares internal data structures that manage parallel communication.

Boundary conditions are handled in the code snippet that follows

```
44 ||  int direction,side;
45 ||  for (direction=0; direction<dim; direction++) {
46 ||    for (side=0; side<2; side++) {
47 ||      int field = 0; double value = 0.0;
```

```
48          IGASetBoundaryValue(iga,direction,side,field,value);
49        }
50      }
```

The routine `IGASetBoundaryValue` is used to set the Dirichlet boundary conditions of this problem, as specified in equation (A.2), and takes as arguments

- the `IGA` context in which the boundary condition is to be set,
- the parametric direction, which takes values `0` or `1` to specify either the first or second parametric directions, respectively,
- the boundary side, which takes values `0` or `1` to specify either the left or right side along the parametric direction, respectively,
- the component index, which is `0` for scalar problems,
- finally, the value to be enforced at the boundary.

Although not highlighted in this example, PetIGA allows users to specify more general and possibly nonlinear boundary conditions. These boundary conditions should be handled in the user-defined residual and Jacobian routines.

The final step to configure the `IGA` context requires the specification of the user-defined residual and Jacobian callbacks, as shown in the following lines of code

```
52      Params params;
53      params.lambda = IGAGetOptReal(NULL,"-lambda",6.80);
54      IGASetFormFunction(iga,Residual,&params);
55      IGASetFormJacobian(iga,Jacobian,&params);
```

A local variable `params` of type `Params` is declared, the `lambda` member is initialized from a user-specified command line option, or from a hardwired default value otherwise. The calls to `IGASetFormFunction` and `IGASetFormJacobian` register within the `IGA` context the user-defined residual and Jacobian routines. They also store the pointer to the `Params` instance which is used in `Residual` and `Jacobian` to access the λ parameter, as explained previously.

The code then proceeds to solve the nonlinear problem

```
57    SNES snes;
58    IGACreateSNES(iga,&snes);
59    SNESSetFromOptions(snes);
60
61    Vec U;
62    IGACreateVec(iga,&U);
63    SNESSolve(snes,NULL,U);
```

The routine `IGACreateSNES` creates a nonlinear solver context and performs additional initialization such as associating the global vector to form the residual and the global matrix in which to form the Jacobian within the solver context. The routine `SNESSetFromOptions` enables users to further configure the nonlinear solver through command line options. Finally, a global vector is created to store the solution coefficients and the routine `SNESSolve` is invoked to solve the problem.

A call to the routine `VecViewFromOptions` creates a viewer context which can be used to visualize the solution vector in real-time, or store the solution as a VTK datafile [162, 163]

```
65    VecViewFromOptions(U,NULL,"-output");
```

Lastly, all PETSc and PetIGA objects created through the code are destroyed to free resources and the routine `PetscFinalize` is invoked to properly shutdown the framework

```
67    VecDestroy(&U);
68    SNESDestroy(&snes);
69    IGADestroy(&iga);
70    PetscFinalize();
71    return 0;
72  }
```

Appendix B

B.1 Energy stable formulation for the PFC equation

B.1.1 Jacobian for the $2+2+2$ mixed form

The Jacobian components of $\mathbf{K}$ in equation (4.50) are defined as

$$K_{AB}^{\phi\phi} = \left(N_A, \frac{N_B}{\Delta t}\right)_\Omega , \tag{B.1}$$

$$K_{AB}^{\phi\sigma} = \left(\nabla N_A, \nabla N_B\right)_\Omega , \tag{B.2}$$

$$K_{AB}^{\phi\theta} = 0, \tag{B.3}$$

$$\begin{aligned} K_{AB}^{\sigma\phi} = & \frac{1}{2}\left(N_A, N_B\right)_\Omega \left(\Psi_c'''\left(\phi_{n+1}^h\right)[\![\phi_n^h]\!] - \Psi_c''\left(\phi_{n+1}^h\right) + \Psi_e''\left(\phi_n^h\right)\right) \\ & - \alpha_n \Delta t \left(\nabla N_A, \nabla N_B\right)_\Omega , \end{aligned} \tag{B.4}$$

$$K_{AB}^{\sigma\sigma} = \left(N_A, N_B\right)_\Omega , \tag{B.5}$$

$$K_{AB}^{\sigma\theta} = -\left(N_A, N_B\right)_\Omega + \left(\nabla N_A, \nabla N_B\right)_\Omega , \tag{B.6}$$

$$K_{AB}^{\theta\phi} = -\frac{1}{2}\left(N_A, N_B\right)_\Omega + \frac{1}{2}\left(\nabla N_A, \nabla N_B\right)_\Omega . \tag{B.7}$$

$$K_{AB}^{\theta\sigma} = 0, \tag{B.8}$$

$$K_{AB}^{\theta\theta} = \left(N_A, N_B\right)_\Omega . \tag{B.9}$$

B.1.2 Running the code

Both PETSc and PetIGA are regularly maintained and updated, so it is worthwhile to download their respective repositories through the version control systems Git[1] and Mercurial[2]. These tools can be used to clone the PETSc and PetIGA repositories with the following commands

- `git clone https://bitbucket.org/petsc/petsc`
- `hg clone https://bitbucket.org/dalcinl/PetIGA`

PETSc must be configured and installed before installing PetIGA. After completing the PetIGA installation, the igakit repository can be cloned.

- `hg clone https://bitbucket.org/dalcinl/igakit`

Igakit is a Python-based pre-processing and post-processing tool for PetIGA. Further information on these software packages can be found in [114, 115, 56, 116, 113], and the discretization proposed in this work can be found in the `demo/` directory of the PetIGA sources.

[1] http://git-scm.com
[2] http://mercurial.selenic.com/

Appendix C

C.1 Generalized formulation for phase-field problems

C.1.1 Jacobian for the mixed form

The Jacobian components of $\mathbf{K}$ in equation (4.50) are defined as

$$K_{AB}^{\phi\phi} = \left(N_A, \frac{N_B}{\Delta t} \right)_\Omega , \tag{C.1}$$

$$K_{AB}^{\phi\sigma} = \left(\nabla^a N_A, \tilde{M}\left(\phi^h\right) \nabla^a N_B \right)_\Omega , \tag{C.2}$$

$$K_{AB}^{\sigma\phi} = \left(N_A, \left(-\tilde{\Psi}'' - \frac{c_0}{2} \right) N_B \right)_\Omega - \frac{c_1}{2} \left(\nabla N_A, \nabla N_B \right)_\Omega - \frac{c_2}{2} \left(\Delta N_A, \Delta N_B \right)_\Omega - \alpha \Delta t \left(\nabla^a N_A, \nabla^a N_B \right)_\Omega , \tag{C.3}$$

$$K_{AB}^{\sigma\sigma} = \left(N_A, N_B \right)_\Omega , \tag{C.4}$$

recalling the expression for the function f defined through equation (4.2).

C.1.2 Generalized-α method for first order systems

The generalized-α method for first-order differential equations in time [125]

$$\mathbf{R}\left(\mathbf{U}, \dot{\mathbf{U}} \right) = 0,$$

is defined as: given $\dot{\mathbf{U}}_n$, $\mathbf{U}_n$, find $\dot{\mathbf{U}}_{n+1}$, $\mathbf{U}_{n+1}$ such that

$$\begin{aligned}
\mathbf{R}\left(\dot{\mathbf{U}}_{n+\alpha_m}, \mathbf{U}_{n+\alpha_f}\right) &= 0 \\
\dot{\mathbf{U}}_{n+\alpha_m} &= \dot{\mathbf{U}}_n + \alpha_m \left(\dot{\mathbf{U}}_{n+1} - \dot{\mathbf{U}}_n\right) \\
\mathbf{U}_{n+\alpha_f} &= \mathbf{U}_n + \alpha_f \left(\mathbf{U}_{n+1} - \mathbf{U}_n\right) \\
\mathbf{U}_{n+1} &= \mathbf{U}_n + \Delta t \left(\gamma \dot{\mathbf{U}}_{n+1} + (1-\gamma)\dot{\mathbf{U}}_n\right)
\end{aligned}$$

where $\Delta t = t_{n+1} - t_n$ is the time-step, and α_f, α_m and γ are parameters which control numerical dissipation and define the method. These parameters can be chosen using the spectral radius $\rho_\infty \in [0, 1]$

$$\begin{aligned}
\alpha_m &= \frac{1}{2}\left(\frac{3-\rho_\infty}{1+\rho_\infty}\right), \\
\alpha_f &= \frac{1}{1+\rho_\infty}, \\
\gamma &= \frac{1}{2} + \alpha_m - \alpha_f,
\end{aligned}$$

to obtain a second-order accurate, unconditionally stable method.