Study of Multi-Point Interactions in PFC Models for Complex Structural Transformations

Matthew Seymour

A thesis presented in partial fulfillment of the degree of

Doctor of Philosophy

Department of Physics

McGill University, Montreal

Canada

October 31, 2017

Dedication

To Conner

Acknowledgements

I am extremely grateful to my advisor, Prof. Nikolas Provatas, for giving me the opportunity to study physics with him and his group. I couldn't have asked for a better graduate school experience. Nik is extremely knowledgable and we have had many fruitful discussions and debates. His advice and guidance have brought me to this point. I thank him also for his patience as I have worked to finish this thesis.

I would like to express my gratitude to Ken Elder for sharing with me his deep knowledge of PFC and computational materials science. Without Ken the contributions I've made in my time here would not have been possible.

I would like to thank Kate Elder. It was a pleasure to work with her on the CVD model. I wish her luck in her future studies.

I would like to thank Niloufar Faghihi for her help in getting me up to speed on PFC and the magnetism model.

To the current and former members of our research group, especially Nana Ofori-Opoku, Gabriel Kocher, Nathan Smith, Hossein Azizi and Bernadine Jugdutt, I would like to thank you for all the help, collaboration, discussions and debates. Without this incredible group of people I'd have never gotten to this point.

I would also like to thank my committee members, Prof. Hong Guo and Prof. Mark Sutton, for providing their perspective and input on my work.

Finally, I would like to thank my friends and family for their patience and support.

Abstract

The phase field crystal (PFC) methodology models material solidification and other transformations through the evolution of an atomic density field driven by dissipative dynamics. PFC permits investigation of atomic scale processes on dissipative time scales. Building on past PFC methodology, we develop and characterize several new models of atomic scale material phase transformations that were not possible with previous PFC modelling approaches. This thesis makes several new contributions to PFC modelling. These are described below.

First, we develop a model of magnetism in materials by extending previous models. This model couples the atomic scale PFC density field to a ferromagnetic order parameter (the magnetization field). We base our model on the structural PFC (XPFC) formalism. Additional terms giving rise to magnetocrystalline anisotropy are also included in the model. We characterize this anisotropy analytically for various crystal structures. Magnetic hysteresis is demonstrated and characterized. Using this model we investigate the effects of external magnetic fields on grain growth in a solidifying system.

Second, we introduce a new, computationally tractable, three-point interaction term for PFC density fields. This interaction term energetically favours particular bond angles between nearest neighbour atoms, while retaining the overall rotational invariance that permits PFC to model crystal formation in any orientation. The primary motivation for this approach is to produce a PFC model of graphene, a material which has proven difficult to model using the PFC formalism. We show that this approach is capable of stabilizing triangle, square, and the aforementioned graphene crystals. Moreover, we show that the defect structure of the resulting polycrystalline graphene structures closely matches experimental

results.

Last, we explore binary PFC models by retaining separate density fields for each atomic species. This differs from the traditional approach of exploring such systems through concentration and total density fields. We demonstrate the versatility of this approach by constructing simple models, including a model of a "salt-like" system, as well as one of an impurity species inhabiting the interstitial sites of a solid lattice. Lastly, we examine a model of the chemical vapour deposition (CVD) of graphene, which is being used to investigate the interaction of carbon and hydrogen on the metal surface during the CVD process.

The major contributions of this work are: the development of new theory to model consistent magneto-crystalline interactions; the introduction into the PFC methodology of rotationally invariant three-point density correlations; and the development of a new formalism to model binary systems based species densities, allowing for local inter-species interaction, leading to specialized binary system models not possible with previous continuum models.

La méthodologie phase field crystal (PFC) modélise les matériaux en transformation (solidification ou autre) à travers un champ de densité qui subit un processus dissipatif. Elle permet d'étudier des processus atomiques à des échelles de temps diffusives. Nous utilisons cette méthodologie pour développer et caractériser plusieurs nouveaux modèles, rendant possible l'étude de phénomènes précédemment inaccessibles. Cette thèse fait plusieures contributions dans le domaine de la modélisation PFC, elles sont listées ci-dessous.

Nous commenons par développer un modèle de magnétisme dans les matériaux, basé sur des travaux antérieurs par Faghihi *et al.*[Faghihi *et al.*, Phy. Rev. E, 88:032407, 2013]. Notre modèle établit un couplage entre la densité PFC à l'échelle atomique et un paramètre d'ordre ferromagnétique (le champ de magnétisation). Basé sur le formalisme XPFC (surnommé 'PFC Structurel') de Greenwood *et al.*[Greenwood *et al.*, Physical Review E, 83:031601, 2011], il inclut des termes supplémentaires donnant lieu à une anisotropie magnétocristalline. Nous caractérisons cette anisotropie analytiquement pour diverses structures cristallines, et démontrons la présence d'une hystérèse magnétique, que nous caractérisons aussi. En utilisant ce modèle, nous étudions les effets de champs magnétiques externes sur la croissance de domaines dans un système en solidification.

Deuxiémement, nous introduisons un nouveau terme d'interaction à trois points viable numériquement pour les champs de densité PFC. Ce terme d'interaction favorise des angles de liaison particuliers entre les atomes voisins les plus proches, tout en conservant l'invariance rotationnelle globale qui permet à la théorie de modéliser la formation de cristaux dans n'importe quelle orientation. La principale motivation de cette approche est de produire un modèle PFC du graphène, un matériau qui s'est avéré difficile à modéliser en utilisant le formalisme PFC

habituel. Nous montrons que cette approche est capable de stabiliser des agencements atomiques triangulaires (réseau hexagonal), carrés (réseau tétragonal) et les cristaux de graphène sus-mentionnés. De plus, nous montrons que la forme des défauts de structure dans notre graphène polycristallin correspond étroitement aux résultats expérimentaux.

Enfin, nous explorons des modèles PFC binaires en conservant des champs de densité distincts pour chaque espèce atomique. Cette approche diffère de la méthode traditionelle de ces systèmes qui privillégie le plus souvent la concentration et le champ de densité totale comme variables. Nous démontrons la polyvalence de notre approche en construisant des modèles simples, y compris un modèle à structure saline, ainsi qu'un modèle pour les impuretés qui habitent les sites interstitiels d'un réseau solide. Enfin, nous examinons un modèle de déposition chimique en phase vapeur (CVD) du graphène. Celui-ci sert à étudier l'interaction du carbone et de l'hydrogène sur une surface métallique pendant le processus CVD.

Contents

Dedication **i**

Acknowledgements **ii**

Abstract **iii**

1 Introduction **1**

1.1 Brief introduction to the phase field crystal model 14

1.1.1 PFC dynamics 16

2 Magneto-Crystalline Interactions **18**

2.1 XPFC model of magneto-crystalline interactions 21

2.1.1 Ferromagnetic free energy density 23

2.1.2 Alloy free energy density 24

2.1.3 Magnetostatics and electrostatics 24

2.1.4 Two-point correlation function 25

2.2 Dynamics 26

2.3 Equilibrium properties 28

2.3.1 Amplitude expansions 28

2.3.2 Phase diagram 30
2.3.3 Magnetic anisotropy 34
2.3.4 Small deformation limit and magnetorestriction 38
2.4 Numerical tests and applications of model 45
2.4.1 Ferromagnetism in square polycrystalline order 45

3 Three Point Correlation Functions 52
3.1 Simplified density functional theory 55
3.2 Two-point correlations 56
3.3 Three-point correlations 59
3.4 Equilibrium Phase diagrams for different crystal structures 63
3.4.1 Triangular crystals 64
3.4.2 Square crystals 66
3.4.3 Graphene crystals 67
3.5 Poisson's ratio of the three-point XPFC model 69
3.6 Dynamics and simulations 73
3.6.1 Polycrystalline 2D materials, defects and coexistence . . . 76

4 Binary Systems 82
4.1 Simplified density functional theory for a two-component system . 84
4.2 Form of the two-point correlation functions 87
4.3 Phase diagrams for two component systems 89
4.4 Special case studies of the binary model 92
4.4.1 "Salt-like" system 92
4.4.2 Solid-gas system 95
4.4.3 Interstitial impurity system 96

4.5 Binary model of graphene-hydrogen surface interactions 98
4.5.1 Properties of the model 99
4.5.2 Dynamics 101

5 Conclusion **104**

A Computing Phase Diagrams **109**

B Fourier Transforms in Polar Coordinates **122**

C Convex Hulls **125**
C.1 Description of convex hull construction algorithm 126
C.2 Complete C++ implementation 127

D Magnetoelastic Coupling Constants **149**

E Three Dimensional Three-point **153**
E.1 3D three point correlation 154

List of Figures

2.1 Organization of the vectors of a reciprocal triangular lattice according to modes. Dots represent reciprocal lattice vectors, circles connect vectors of equal magnitude, i.e. in the same mode. To each mode an amplitude ϕ_k is assigned. 29

2.2 Amplitudes of the first three modes in Eq. 2.19 as a function of ϕ_0 with $\sigma = 0.12$, for the case of a single-peaked XPFC correlation with $k_1 = 2\sqrt{2}\pi$, $1/(2\rho_i\beta_i) = 1/24\sqrt{2}$, and $t = v = 1$. 32

2.3 Free energy as a function of ϕ_0 for $\sigma = 0.12$. The other parameters are the same as described in Fig. 2.2. Note the kink at the liquid-solid transition. 33

2.4 Phase diagram for the magneto-XPFC model, showing coexisting liquid, BCC and FCC phases, as well as paramagnetic and ferromagnetic phases. Parameters are $\alpha_2 = .001$, $r_m = .025$, $\omega_m = 0.25$, and $\gamma_m = 1$. Generated using five modes ($N = 5$). . . 34

2.5 Hysteresis loop for a single two dimensional square crystal. System parameters are as stated at the beginning of the section with $\sigma = 0.04$, $n_0 = 0.05$, and $(\alpha_2, \alpha_4) = (0.001, -0.01)$. 46

2.6 Plot of magnetization components m_x and m_y versus time (t) for the hysteresis loop in Fig. 2.5. Note that the magnetization reverses by rotation of the magnetization vector. 48
2.7 Randomly oriented crystal seeds growing under the influence of an external field. Arrows indicate the orientation of the seed crystals. 49
2.8 Comparison of initially identical systems grown under an external magnetic field (left) and no field (right). Crystal grains aligned with the x-y axes (such as A and G) have their easy axes aligned with the external field, and grow at the expense of those that are not aligned (such as D). 50
2.9 Difference in power spectra of polycrystalline solid grown in an external field (P_{ex}) and no external field (P_0), respectively. 51
2.10 Difference in power spectra of polycrystalline solid grown in an external field at late (P_l) and early (P_e) times, respectively. 51

3.1 Two-point correlation function in real space. 57
3.2 Plot of $-2RJ_1(r_0k)/(r_0k)$ in units of k/r_0. 58
3.3 Triangular-disorder phase diagram using only two-point correlations ($X = 0$). 64
3.4 Triangular-disorder phase diagram using two and three-point correlations, with $m = 6$ and $R = 6$. Here $r_0/a_0 = 0.70785$. 65
3.5 Square-disorder phase diagram using two and three-point correlations with $m = 4$, $R = 5$ and $r_0/a_0 = 0.81736$. 66
3.6 Graphene-disorder coexistence phase diagram, with $m = 3$ and $R = 6$. Here $r_0/a_0 = 1.2259$. 70

3.7 Density fields of triangular (*a* and *b*), square (*c* and *d*) and graphene (*e* and *f*) phase growth, showing early (left) and late (right) times during solidification. Systems are initialized with gaussian density fluctuations. For all three systems $\phi_0 = 0.3$ and values of r_0/a_0 match those in Table 3.1. For (*a* and *b*), $R = 7$ and $X = 0$. For (*c* and *d*), $R = 6$ and $X^{-1} = 0.5$. For (*e* and *f*), $R = 6$ and $X^{-1} = 0.4$. 77

3.8 Comparison of simulated and experimentally determined defect structures of polycrystalline graphene. The defect structure of Figure 3.7(*f*) is highlighted in (*a*). The grain boundary is resolved by a line of 5-7 defect structures. These defect structures match those found experimentally in polycrystalline graphene membranes grown by chemical vapour deposition (CVD) [102]. (*b*) shows an atomic resolution transmission electron microscope (TEM) image of one such graphene membrane; the defect structure is highlighted in (*c*). (*b*) and (*c*) reprinted by permission from Macmillan Publishers Ltd: Nature [102], copyright 2011. 79

3.9 Simulation of coexistence between the ordered and disordered phases of graphene. Density field $n(\mathbf{r})$ of the equilibrium interface between phases shown in (*a*). Smoothed average density along the longitudinal axis depicted in (*b*). Here, $X^{-1} = 0.5$, $R = 6$. Average densities of 0.057 and 0.134 in the disordered and ordered phases respectively match closely the theoretical values from the phase diagram in Figure 3.6. 81

4.1 Density fields of three two component systems, with the component densities A and B coloured red and green, respectively. (a) shows a salt-like system where A and B atoms are attracted to each other but not themselves. (b) shows a system where A atoms form a solid, B atoms favour remaining disordered, but A and B atoms repel each other. (c) shows a system where the disordered B atoms favour dispersing within the lattice of the ordered A atoms. The details of the models used to simulate these three situations are discussed further below. 88

4.2 Free energy plot as a function of density. The red line is the coexistence line, where one phase decomposes into two phases of density n_1 and n_2. As the average density moves from n_1 to n_2, the system remains in coexistence at these endpoints, however the volume fraction changes from $\{v_1, v_2\} = \{1, 0\}$ to $\{v_1, v_2\} = \{0, 1\}$. 91

4.3 Replacing the free energy plot with its convex hull. The convex hull represent the true equilibrium free energy of the system. . . . 92

4.4 Early (left) and late (right) density evolution for the "salt-like" binary system. The density n_A is indicated in red, and n_B is indicated in green. 93

4.5 Phase diagram for the "salt like" toy system (Figures 4.1a and 4.4). The two lines indicate the coexistence region. For all points in the phase diagram $n_A = n_B = n$. The strength of the interaction is determined by s. 94

4.6 Phase diagram for the second system (Figure 4.1b), with $\alpha^{ij} = 1.5$, $a^{ij} = 1$, $R_{BB} = A_{BB} = A_{AB} = 0$, $R_{AA} = A_{AA} = 6$ and $R_{AB} = 2$. Grey lines indicate coexistence tie lines. Species B is always disordered. 96

4.7 Early (left) and late (right) density evolution for the gas-solid system. n_A indicated in red, n_B indicated in green. For this simulation, we took $n_A = n_B = -0.5$. 97

4.8 Separated density fields of the steel-like system. (a) shows both densities together, with n_A in red and n_B in green. (b) shows n_A by itself. (c) shows n_B by itself. Note that the n_B field is attracted to grain boundaries and other defects in the n_A crystal. 98

4.9 Phase diagram for the graphene-hydrogen system, with $a = b = 1$ in the hydrogen-carbon interaction term and $R = 6$, $X^{-1} = 0.4$, $B = 0.125$ in the graphene three-point correlation function. The grey lines indicate coexistence tie lines for the graphene and disordered phases. Species B ("hydrogen") is always disordered. 101

4.10 Results of the graphene-hydrogen model. (a) shows the carbon density, n_A, and (b) shows the hydrogen density, n_B. Note the attraction of the hydrogen atoms to the grain boundaries of the graphene. $n_A = 1$, $n_B = -2$, $R = 4.5$, $X^{-1} = 0.35$, and $a = 2$ and $b = 0$. Results courtesy of Kate Elder. 103

List of Tables

3.1 The ratio of r_0 to a_0 for various two dimensional crystal lattices. K_1 is the reciprocal lattice vector of the first mode of a crystal structure. 59

3.2 Measurement of elastic coefficients for various system parameters for the three-point graphene system. Results obtained using an eight mode amplitude expansion. $R = 6$ and $u = 0.01$ for all entries. 73

4.1 Summary of system parameters for the three systems described. . 98

Chapter 1

Introduction

Advanced materials are a crucial part of modern economies. Manufacturing in sectors such as aerospace, automotive, microelectronics, and energy, to name a few, is critically dependent on the development of advanced materials with improved properties. Examples include metal alloys for green transportation and microelectronic components or metal oxides for improved energy storage. Fundamental to the discipline of material science is that a solid state material's macroscopic properties depend intimately on its underlying microstructure. For example, yield strength of metals depends on grain size and defect distributions. Reliability of metal interconnects in nano-electronics depend on the interaction of electrons with grain boundaries and voids. Coercivity in ferro-magnetic materials is directly controllable by the grain size and crystalline anisotropy of a material. Many more examples abound in the materials science and materials engineering literature. The theme in most cases is the same: the evolution of microstructure in a solid is ultimately the result of non-equilibrium thermodynamic processes at play during its initial phase transformation from the liquid state, and its subsequent

processing. As a result, understanding materials and their properties necessitates understanding the fundamental physics of mesoscale patterning in materials—often inappropriately referred to under the blanket name of "microstructure"—and how this is influenced from microscopic physical principles.

While the final material microstructure of a material can be examined through various forms of microscopy (such as electron, atomic force, transmission electron, or scanning tunnelling microscopes) as well as through X-ray and neutron diffraction, the dynamical evolution of microstructure processes is difficult to examine in-situ experimentally, particularly in metals where the solidification thermo-mechanical processing occurs at high temperatures. Even in cases where experimental observation is achievable, predicting the causal connections between a process and resulting microstructure is typically not possible without some theoretical connections to guide the causal connections between experimental measurements and materials parameters.

Due to this difficulty of examining microstructure evolution experimentally, modelling the process of microstructure evolution plays an important complimentary role to experiments in material science. In order to model microstructure phenomena over the relevant time and length scales—that is, the dynamics of atomic ordering and defect formation on the atomic scale and their emergent effects on the mesoscale, under the influence of thermodynamic forces—theoretical models are required that can bridge the gap between the physics of atomic scale effects and long wavelength pattern formation that defines microstructure over diffusional time scales. The most fundamental level of theoretical modelling revolves around first principles methods, which are concerned primarily with electron band structure and the chemistry of bonding or with electron conduction.

On a slightly higher scale is molecular dynamics (MD), which examines atomic length scales very accurately for a range of nanoscale systems at short time scales (on the order of the Debye frequency). MD has been instrumental in elucidating the kinetic and capillary properties of the interface such as the surface energy and its anisotropy[1, 2]. These are quintessential in free boundary models of solidification[3]. However MD is not presently capable of simulating mesoscale phenomena on diffusive time scales that characterize materials phenomena such as crystallization, grain coarsening and dislocation dynamics out to experimentally relevant time scales.

Another very promising modelling paradigm that has emerged in the past 20 years or so is that of *phase field models*. The original phase field models were extensions of Ginzburg-Landau type models that demonstrated that simple free energies based on symmetries of a material could describe all the essential qualitative physics of first and second order phase transitions [4, 5, 6, 7]. Phase field models describe a system in terms of smooth order parameters that vary spatially on the atomic scale of a solid-liquid or solid-solid interface. They are also coupled to other thermodynamic fields such as temperature, impurity concentration, strain fields, etc. As such, they are excellent at capturing the salient physics of microstructure evolution problems governed by the competition of mass and heat diffusion, interface energy, and elastic strain energy[4, 5, 6, 8, 9, 10, 11]. They have long ago been shown to capture the properties of higher length scale free boundary problems of solidification in the limit where there exists sufficient separation between the model's solid-liquid interface—often treated as a computational parameter—and the capillary length[12, 13]. More recent advances have also used matched asymptotic analysis to quantitatively map phase field models

onto engineering sharp interface models of solidification in the limit when the interface is of the same order or even greater than the capillary length[14, 15, 16]. This made the results of phase field modelling, for the first time, quantitatively comparable to experiments[15, 17, 18, 19, 20].

Since traditional phase field models are usually formulated in terms of fields that are spatially uniform in equilibrium, they suffer from a very important drawback in that their use precludes from study physical features and materials processes that arise on the atomic scale, specifically, due to the periodic nature of crystalline phases. This preclusion includes elastic and plastic deformation, anisotropy and multiple orientations. Traditional phase field models have typically gotten around this problem in Ginzburg-Landau type phase field models by adding one or more auxiliary fields in order to describe atomic scale phenomena effectively, by adding fields describing the distribution of dislocations[21, 22], continuum strain and stress fields through a material[23, 24] or the orientation of crystal grains[25, 26]. Another class of phase field models used in the materials science literature these days defaults to a simple use of multiple order parameters to described a specific crystal orientation. This goes back to the work of Khachaturyan[27], where a discrete set of orientations was to be described, which corresponded to a discrete set of solid-state precipitate orientations in a parent matrix phase. This approach has also worked its way into solidification studies since its inception, where multiple order parameter or "phase fraction" fields are ascribed to different orientations in polycrystalline solidification[22, 25, 28, 26, 29, 30, 31, 32]. Beyond being computationally intensive, this approach is physically inconsistent as it breaks rotational invariance. This approach also precludes a consistent description of nucleation. As a result, such models are only appropriate for the study

of select problems involving a few crystals at pre-defined orientations. Moreover, such so-called multi-phase field or multi-order parameter approaches, like all phase field models, require that one know the free-boundary conditions governing the growth of microstructure in order to tune most of their parameters. This is simple enough to do in traditional solidification problems, but indeed, problems involving the interplay of interface kinetics, elastic and plastic energy, and large density changes don't even have a corresponding sharp interface models.

A new extension to phase field modelling has emerged in the past 12-15 years known collectively as *phase field crystal* (PFC) models[33, 34, 35]. This methodology describes the evolution of the atomic density of a system according to dissipative dynamics driven by free energy minimization. In the PFC approach, the free energy functional of a solid phase is minimized when the density field is periodic. As shown in Refs. [33, 34], the periodic nature of the density field, and the rotational invariance of the free energy, naturally gives rise to elastic energy in solids, multiple crystal orientations and a proper accounting of misorientation energy, the nucleation, motion and interaction of dislocations, and multi-phase nucleation. While these physical features are included in other atomistic approaches (such as molecular dynamics), a significant advantage of the PFC method is that, by construction of its dynamical equations of motion, it operates on diffusive time scales and not on the prohibitively small time scales associated with atomic lattice vibrations. From its earliest applications to pure materials, the PFC approach was shown to model a wide range of phenoma[33, 34] dominated by atomic scale elastic and plastic deformation effects, including grain boundary interactions, epitaxial growth and the yield strength of nano-crystals. Following its phenomenological introduction and demonstration in single component (i.e. pure) materi-

als, the PFC model was somewhat formalized when it was shown to emerge as a simplification of classical density functional theory (cDFT), truncated to two-point particle correlations[35]. In the same work, the application of cDFT to two fields, each representing a different elemental alloy component, was also shown to lead to a simplified alloy PFC model, wherein a PFC density field describing the total mass density was coupled to a concentration field[35] (a similar approach was concurrently proposed by Jin and Khachaturyan[36]). Recently, the original PFC has been augmented with long wavelength multi-point correlations to model triple-point materials and problems relevant to CVD processes in pure materials[37] and alloys[38].

Structure in the original PFC models arises through a single-peaked fourth-order polynomial correlation function in k-space, which allowed only simple structures to be modelled realistically. For example, in two dimensions these models could produce triangular crystal structure and in three dimensions, body-centred cubic (BCC). While it was later found by Jaatinen et. al [39, 40] that face-centred cubic (FCC) and hexagonal closed-packed (HPC) structures were possible in three dimensions, transforming from FCC to HCP was only possible through a change of average density, not temperature. Moreover, the original PFC free energies reproduced equilibrium phase diagrams that were experimentally realistic only in small regions of their phase space (average density for a pure material, concentration for the binary alloy, and the models' temperature parameter). A new type of PFC model called the *structural phase field crystal model* (XPFC) was developed after 2010 to address these shortcomings of the original PFC approach[41]. In XPFC modelling, two-point particle interactions were modelled entirely in Fourier space by using two-point correlation functions consisting of a family of

Gaussian peaks, located at the positions of the reciprocal space peaks of the lattice (or lattices) of interest. Specifically, one Gaussian peak is included for the main reflection from each family of lattice planes for the crystal structure of interest. The height of each peak is modulated by a Debye-Waller like coefficient that scales with a model parameter that represents temperature in some way. By controlling the heights of these peaks, most metallic crystal structure relevant to metals could be stabilized from a liquid. The original work of Ref. [41] was for a pure material. This was later extended binary alloys[41, 42], and then to general multi-component materials by Ofori-Opoku *et al.*[43].

In the XPFC approach, controlling the positions of the two-point correlation peaks makes it possible to very easily model the nucleation and growth from a liquid of triangular and square lattices in two dimensions and FCC and HPC in three dimensions. In a pure material, this can be done either through a temperature quench or by increasing the density of the system. In addition, it is possible to model solid-state transformations. As an example of this feature, it is possible to examine nucleation and growth of a polycrystalline triangular solid, from which square lattice precipitates emerge through heterogeneous nucleation at triple junctions and dislocations. This is possible though processes that change either temperature or density. In the alloy XPFC model, such changes can be induced through by varying the concentration of solute(s). By controlling the width of the Gaussian peaks, it is also possible to model different material parameters, such as elastic coefficients or solid-liquid surface energies. XFPC represented the first time that phase field type models could look at phase transformations that captured both mesoscale structure (grain size distributions and orientations, solute segregation) and atomic scale effects (a wide range of crystal orientations, grain

boundaries, anisotropic elasticity, dislocations and atomic scale nucleation) under one formalism. For example, Berry *et al.* showed that a sufficient number of peaks in the XPFC correlation function can control the stability of stacking faults and partial dislocations in FCC materials[44], in addition to the usual long-lived dislocation structures. Using a two-peaked XPFC model also captured the physics of diffusion-controlled Cobble creep in polycrystalline 3D FCC materials[45]. The multi-component XPFC model of Ofori-Opoku *et al.*[43] also predicted the mechanism for precipitate nucleation through a process of dislocation-mediated solute clustering in two and three dimensions. This work predicted that embryonic precipitates become post-critical in size due to the strain energy of assembled dislocations[46, 47], findings that were later verified by electron microscopy work in Al-Cu and Al-Mg-Si.

The above examples suggest numerous interesting areas of future applications for XPFC modelling. One in particular is in the area of microstructure engineering. In traditional materials this is done almost exclusively by thermal and mechanical processing of grain structure by the processes of solidification, precipitation and recrystallization. This typically leads to indirect control of microstructure. Another candidate avenue for controlling grain structure is to couple the crystalline anisotropy in the lattice of a material to external fields. One of the most promising classes of materials for which this can happen is ferromagnetic and ferroelectric materials. While this can be coupled to molecular dynamics, the short time scale of MD precludes any long-time study of how external fields can influence crystalline anisotropy and grain size selection. The diffusional times scales accessed by PFC models, and the complexity of crystalline structure and elastic anisotropy that can accessed by XPFC in particular, makes a study of mi-

crostructure pattering in ferromagnetic and ferroelectric materials an interesting avenue of future investigation.

Further to new applications of XPFC modelling, it is also instructive to consider avenues of study that lead to improvements this methodology. Specifically, XPFC models, and indeed, the broader class of PFC models, have several shortcomings that remain presently unsolved, and which must resolved for the formalism to be made more physically consistent. For example, it is not clear if it is possible to describe the emergence of complex non-metallic structures solely from two-point interactions, or whether higher-order multi-point correlations are required. A case in point, and one that is quite timely in the materials science community, is that of graphene, a non-metallic two dimensional allotrope of carbon exhibiting a honeycomb lattice. Recent work by Ref. [48] has shown that graphene and Kagomi lattices and grain boundaries can be stabilized from a PFC model which, analogously to XPFC, uses multiple two-point correlation peaks in the free energy. However, that approach precludes the possibility of stable coexistence between graphene and a disordered phases such as a vapour or liquid, a crucial feature for modelling growth of graphene by chemical vapour deposition (CVD). These authors also recently developed a two-component PFC model with two-point interactions wherein two triangular lattices interact to stabilize a suite of very interesting 2D phases/defect structures, including a graphene-like honeycomb lattice[49]. However, this approach only allows graphene-disordered coexistence along a line of equal amounts of A/B components, and represents the honeycomb lattice with equal substitutional constituents of A and B, both conditions being unsuitable to study CVD graphene. Other problem areas with current XPFC models involve the structure of some exotic solid state defects. As shown

in the work of Berry *et al.*[44], the large number of XPFC peaks needed to stabilize stacking faults in FCC materials adversely affect the stability, and thus the efficiency, of dynamical simulations of the XPFC model. Indeed, it was shown that in order to stabilize stacking faults, it was necessary to use a few low-k, but broad, XPFC peaks, but this essentially transformed the XPFC model into the regular PFC model, and thus lost control of the desired features of the XPFC: stability of crystal structure, elastic properties and robustness of phase coexistence.

Another problematic feature of present XPFC models (and also PFC models in general) regards density changes in multi-component or alloy systems. Most XPFC and PFC models have used the atomic density field to distinguish between periodic and uniform phases, and concentration to stimulate the phase changes; local changes of density are typically ignored or at best are present in some implicit but uncontrolled way. Fixing the phase diagram in concentration-density-temperature space is a crucial feature for working in the Helmholtz representation, where pressure changes through the concentration-temperature phase diagram, as shown by Jugdutt *et al.*[50] in their examination of the equilibrium properties of alloy interfaces in the XPFC model. Furthermore, dynamical simulations in the PFC literature have largley ignored local average density change, making them physically inconsistent. The two issues of density change are essential for a proper description of hot tearing during of rapid solidification of confined liquid volumes[38]. Part of the lack of a consistent treatment of average density changes in PFC modelling lies in the lack of an efficient methodology for determining phase diagrams in density-concentration-temperature space. Another lies in in the increased complexity of the equations of motion generated when evolving concentration and density self-consistently. Both the last two problems are much more

effectively addressed if alloy XPFC models are re-designed from a fundamental level as multi-component theories, consistent with cDFT.

The overall goal of this thesis is to extend the PFC methodology beyond its present niche as a model of simple metals and alloys. We focus first on developing a realistic PFC model of magnetic materials, which incorporates magnetocrystalline interactions, including anisotropy and magnetostriction. Next, we focus on the development of a PFC model that employs three-point density correlations, which allows the modelling of 2D materials. We apply this new approach to the modelling of graphene, a material characterized by strong covalent bonds with a distinct defect structure which has proven a challenge for the PFC methodology. Finally, we investigate multicomponent PFC models from a novel perspective that employs atomic species densities directly, permitting the investigation of materials more complex than multicomponent alloys traditionally studied through PFC.

The thesis begins with an application that extends the XPFC method of Refs. [41, 43] to ferromagnetic alloy materials. This work expands on earlier work by Faghihi *et al.*[51]. This is done by coupling a magnetization field to gradients of the XPFC atomic density. The equilibrium properties of the model are examined, showing how the competition of crystalline anisotropy and an external magnetic field can influence magneto-restriction and grain boundary evolution. We also demonstrate this modelling approach as a viable method for microstructure pattering.

The second topic investigated in this thesis is that of modelling complex non-metallic structures, focusing specifically on graphene. We break with the traditional PFC approach for the first time and extend the free energy out to three point-correlations. Here, the two-particle interactions are treated through the correlation

function of a simple repulsive interaction. Structural complexity at the atomic scale is regulated by a novel three-point correlation function. This three-point correlation function favours particular bond angles over others, while remaining invariant with respect to rotations of the system. A significant feature of the form of this three-point correlation function is that it is also computationally tractable, with asymptotic computational complexity no worse than the two-point correlation function of the previous XPFC model. The model's equilibrium properties are examined, demonstrating a robust range of coexistence between graphene and a disordered phase, which can be interpreted as a vapour phase. We also calculate some of the model's elastic properties. Finally, we investigate grain boundaries in graphene as predicted by this latest generation of structural XPFC model, showing excellent agreement with experimental observations.

The final topic of this thesis is an extension of the multi-point XPFC model to multiple components. Alloys in the PFC framework are traditionally constructed in density-concentration (n, c) variables[35, 42, 43]. The concentration is assumed to vary on a much longer length scale than the atomic density fluctuations[43]. This has been relatively successful when describing metallic alloys where the thermodynamic free energies of phases are typically discussed in terms of concentration and temperature. In that approach, it is reasonable to treat the density n as an atomically periodic ordering field whose structure is modulated by the concentration c in a way that conforms to a known phase diagram. This approach, however, does not work well when constructing general multi-component models for which the the outcome (in terms of thermodynamic equilibrium) is not known. An example of this is the phase diagram of CVD graphene wherein carbon and hydrogen appear to be able to coexist in graphene and disordered phases on the

surface of a metal such as copper. In this thesis, we adopt an approach that is more closely linked to cDFT and assign each elemental component its own field (e.g. n_A, n_B for a binary mixture). This approach makes modelling interactions of each component and between components more transparent and controllable. It also allows the diffusion coefficients of each component to differ, as well as eliminating the phenomenological assumption that the concentration must vary on longer length scales than the atomic density variations. Beyond having more fundamental underpinnings than previous approaches, we show that this new flexibility allows unique structures to be stabilized, including mixed (i.e. salt-like) structures. Moreover, employing explicit diffusion coefficients for each component allows for modelling of important systems such as iron-carbon, or surface carbon-hydrogen (vapour deposition of graphene).

The remainder of this thesis is organized as follows. Following a brief introduction to the PFC formalism in the remainder of this chapter, Chapter 2 discusses the new XPFC model that couples magnetization to the atomic density in ferromagnetic materials. Chapter 3 introduces a new XPFC model that employs both two and three-point correlations. This new model is shown to produce complex lattices. Here, we demonstrate its application to polycrystalline graphene and its defects. Finally, Chapter 4 extends the new three-point correlation formalism of Chapter 3 to two components. The robustness of the model is demonstrated for generating different types of materials through control of inter-particle interactions. This section showcases a computationally efficient numerical approach for computing complex two-component phase diagrams. This method is of general validity and of significant enough importance that its details are shown the the Appendix. Several calculations and numerical algorithms used in the main text

are also supported in Appendices. Each chapter in this thesis comes with its own introduction to motivate the topic of that chapter and to emphasize the value and relevance to the topic of XPFC modelling.

Note to the reader: All chapter, section, equation, and figure references, as well as citations, are hyperlinked in the digital PDF version of this work, for convenience.

1.1 Brief introduction to the phase field crystal model

The PFC model can be derived from classical density functional theory (CDFT) through the application of some simplifying assumptions. The starting point is the CDFT free energy functional of T.V. Ramakrishnan and M. Yussouff[52], expanded close to a reference liquid state at coexistence with a solid, of reference density $\bar{\rho}$:

$$\frac{F}{k_B T} = \int d\mathbf{x} \Bigg\{ \rho(\mathbf{x}) \ln\left(\frac{\rho(\mathbf{x})}{\bar{\rho}}\right) + (\rho(\mathbf{x}) - \bar{\rho}) - \frac{\rho(\mathbf{x})}{2} \int d\mathbf{x}' C(|\mathbf{x} - \mathbf{x}'|)\rho(\mathbf{x}') \Bigg\}. \tag{1.1}$$

The first part of the integrand, $\rho \ln(\rho/\bar{\rho}) + (\rho - \bar{\rho})$, is the free energy density of an ideal gas. The second part, $-(\rho(\mathbf{x})/2) \int d\mathbf{x}' C(|\mathbf{x} - \mathbf{x}'|)\rho(\mathbf{x}')$, is the excess free energy due to particle interactions. It is ultimately responsible for solidification.

First, we rescale space by $\mathbf{r} = \mathbf{x}/a$, with $\mathbf{r}$ being dimensionless. a is a length on the order of the lattice spacing of the solid phase. With $V = a^3$, the free energy

is now

$$\frac{F}{k_B T V} = \int d\mathbf{r} \Bigg\{ \rho(\mathbf{r}) \ln\left(\frac{\rho(\mathbf{r})}{\bar{\rho}}\right) + (\rho(\mathbf{r}) - \bar{\rho}) - \frac{\rho(\mathbf{r})}{2} \int d\mathbf{r}' C(|\mathbf{r} - \mathbf{r}'|)\rho(\mathbf{r}') \Bigg\}. \tag{1.2}$$

Next, we recast the free energy in terms of a dimensionless density $n(\mathbf{r}) = (\rho(\mathbf{r}) - \bar{\rho})/\bar{\rho}$. Substituting $\rho = \bar{\rho}(n+1)$ into the ideal free energy and expanding to fourth order we get

$$\bar{\rho}\left(1 + 2n + \frac{n^2}{2} - \frac{n^3}{6} + \frac{n^4}{12}\right). \tag{1.3}$$

The constant term will not affect the physics and can be discarded. The linear term can also be discarded. In what follows, the density will be a conserved parameter:

$$\frac{1}{\Omega}\int_{\Omega} n(\mathbf{r}) d\mathbf{r} = n_0, \tag{1.4}$$

where Ω is the volume of the system and n_0 is the system average density. Under this condition, it should be clear that any linear term in the free energy density (of the form $an(\mathbf{r})$) will integrate to the same value no matter how the density is distributed. Such linear terms can have no effect on a system driven by free energy minimization, so we will discard them.

Substituting the dimensionless density into the excess free energy term yields

$$-\frac{\rho(\mathbf{r})}{2} \int d\mathbf{r}' C(|\mathbf{r} - \mathbf{r}'|)\rho(\mathbf{r}') \rightarrow -\frac{1}{2}(n(\mathbf{r}) + 1) \int d\mathbf{r}' C(|\mathbf{r} - \mathbf{r}'|)(n(\mathbf{r}') + 1). \tag{1.5}$$

Noting that

$$\int d\mathbf{r} \int d\mathbf{r}' C(|\mathbf{r}-\mathbf{r}'|)n(\mathbf{r}') = \int d\mathbf{r} n(\mathbf{r}) \int d\mathbf{r}' C(|\mathbf{r}-\mathbf{r}'|) \tag{1.6}$$

(taking advantage of the rotational symmetry of C and swapping $\mathbf{r}$ and $\mathbf{r}'$), we can manipulate Equation 1.5 into

$$= -\frac{1}{2} n(\mathbf{r}) \int d\mathbf{r}' C(|\mathbf{r}-\mathbf{r}'|)n(\mathbf{r}') + \left(n(\mathbf{r}) + \frac{1}{2}\right) \int d\mathbf{r}' C(|\mathbf{r}-\mathbf{r}'|). \tag{1.7}$$

Since our integrals are over all space, $\int d\mathbf{r}' C(|\mathbf{r}-\mathbf{r}'|)$ evaluates to a constant and the entire second term is of linear order in $n(\mathbf{r})$. As we did with the ideal free energy, we can discard such terms. This leaves us with

$$-\frac{1}{2} n(\mathbf{r}) \int d\mathbf{r}' C(|\mathbf{r}-\mathbf{r}'|)n(\mathbf{r}') \tag{1.8}$$

for the excess free energy density.

Our complete free energy is now

$$\frac{F}{k_B T V} = \int d\mathbf{r} \left\{ \frac{n^2(\mathbf{r})}{2} - \frac{n^3(\mathbf{r})}{6} + \frac{n^4(\mathbf{r})}{12} - \frac{1}{2} n(\mathbf{r}) \int d\mathbf{r}' C(|\mathbf{r}-\mathbf{r}'|)n(\mathbf{r}') \right\}. \tag{1.9}$$

1.1.1 PFC dynamics

As a conserved parameter, the density n obeys *Model B* type dissipative dynamics[53]:

$$\frac{\partial n}{\partial t} = M_n \nabla^2 \left(\frac{\delta F}{\delta n} \right) + \xi. \tag{1.10}$$

Here, $\delta F/\delta n$ is the *functional derivative* of the free energy with respect to the density field, M_n is the diffusivity parameter, and ξ is a noise parameter. The noise parameter should satisfy the fluctuation-dissipation theorem. While important for certain classes of phenomena, for simplicity the noise term will not be included in what follows.

The functional derivative of the ideal free energy, F_{id}, is straightforward:

$$\frac{\delta F_{id}}{\delta n} = n(\mathbf{r}) - \frac{n^2(\mathbf{r})}{2} + \frac{n^3(\mathbf{r})}{3}. \tag{1.11}$$

The excess free energy, F_{ex}, is a convolution integral. Its functional derivative is well known[53] and given by:

$$\frac{\delta F_{ex}}{\delta n} = -\int C_2(|\mathbf{r} - \mathbf{r}'|)n(\mathbf{r}')\, d\mathbf{r}'. \tag{1.12}$$

Combining, we get the equation of motion of the density field:

$$\frac{\partial n}{\partial t} = M_n \nabla^2 \left\{ n(\mathbf{r}) - \frac{n^2(\mathbf{r})}{2} + \frac{n^3(\mathbf{r})}{3} - \int C_2(|\mathbf{r} - \mathbf{r}'|)n(\mathbf{r}')\, d\mathbf{r}' \right\}, \tag{1.13}$$

where the noise term has been dropped here for simplicity.

The remainder of this thesis focuses on extensions of this basic PFC model. In Chapter 2, we add a magnetization field to the free energy; in Chapter 3, we move beyond two-point correlations and include three-point correlations; and in Chapter 4, we consider multicomponent models, expanding the free energy for systems with more than one density field.

Chapter 2

XPFC Modelling of Magneto-Crystalline Interactions

When a solid undergoes a spontaneous transition to a magnetized or electrically polarized state, its electromagnetic properties are strongly controlled by its microstructure because both the magnetization and polarization couple strongly to the crystalline structure [54]. For example, the efficiency of a transformer core is directly influenced by its coercivity, which in turn is a function of the polycrystalline grain size [55, 56, 57, 58, 59]. Ferromagnetic or ferroelectric order can also induce a strain in the lattice. This effect is referred to as magnetostriction in the former case and the piezoelectricity in the latter. The coupling between electric, magnetic and elastic energy in materials has been exploited for many years. For example, the ability to turn magnetic or electric energy into mechanical energy and vice versa is the mechanism behind many sensors and actuators. Materials that contain at least two of the three properties—ferroelasticity, ferromagnetism and ferroelectricity—are referred to as multiferroic.

While there exist some single phase materials that are multiferroic, typically these materials are unsuitable for practical applications as the magneto-electric coupling is either too weak or occurs at low temperatures. In recent years, researchers have been considering composite materials which contain both ferromagnetic and ferroelectric properties (e.g. $YMnO_3$ and $BaNiGF_4$)[60, 61, 62]. Such materials are of interest as the elastic coupling between the materials can be exploited by using either a magnetic field to control ferroelectricity or an electric field to control magnetization. The latter is particularly attractive for non-volatile memory applications in semiconductors where the ability to write and read on magnetic material using current is very advantageous. Heteroepitaxially grown films offer a promising configuration for such composite materials, although the great costs associated with their fabrication makes them difficult and expensive to realize. It is thus important to understand the properties of self-assembled multiferroic materials made by cost effective processes like sintering, embedding particles of one material in another, or eutectic solidification[63, 64, 65].

Modelling multiferroic material properties is challenging because magnetization and polarization couple strongly to the microstructure of a material. For example, the crystal lattice often determines the direction in which the magnetic ($\mathbf{m}$) or electric ($\mathbf{P}$) dipoles align. However, in a real materials, each grain typically has a different crystallographic orientation. In polycrystalline materials, this grain texture can significantly alter coercivity[55, 56, 57, 58, 59], a measure of the strength of the external magnetic field needed to change the direction of magnetization in the material. Furthermore, the presence of grain boundaries, dislocations and magnetic impurities is also expected to influence magnetization and polarization, either directly or through the influence of such topological defects on grain

size. In addition it is expected that the volume fraction of composite components or phases also will be important.

Another difficulty in modelling multiferroics is due to the multiple length and time scales that need to be considered. Molecular dynamics simulations offer an accurate way to simulate atomic interactions. However, they are often limited to nanosecond time scales and nanometer length scales. Traditional phase field models have been developed to describe multiferroics using a free energy functional that couples polarization, magnetization and strain. These models incorporate anisotropy with respect to an *a priori* known crystal orientation and elasticity through the the introduction of auxiliary fields. For example, the Landau-Ginzburg-Devonshire free energy[66] for ferroelectrics includes terms like $\alpha_{11} \sum P_i^4 + \alpha_{12} \sum_{j>i} P_i^2 P_j^2$ where $\alpha_{12} \neq 2\alpha_{11}$ and i and j refer to cartesian coordinates. Grain orientation is incorporated by coupling polarization or magnetization to a local orientation parameter[67], which can be fixed from a grain distribution or, in principle, evolved according kinetic model. A drawback of traditional phase field approaches is that they typically lack an explicit connection to atomic scale crystal structures.

An adaptation of the phase field methodology that can help resolve some of the aforementioned modelling challenges is the structural phase field crystal (XPFC) model. This chapter introduces a binary ferromagnetic XPFC model that couples the PFC crystal density field n and solute concentration field c to a magnetization field $\mathbf{m}$, making it possible to couple ferro-magnetic domain formation to grain orientation, grain boundaries, elastic strain, defects and solute impurities. The coupling of $\mathbf{m}$ to n at the atomic scale naturally leads to the emergence of magnetic anisotropy at the meso-scale, as well as the phenomenon of magne-

tostriction. This model extends the work of Faghihi *et al.*[51], which considered a single component ferromagnetic PFC model without anisotropic ordering of $\mathbf{m}$. One of the important features of the model introduced here is that the form of coupling between n and $\mathbf{m}$ determines the preferred crystallographic directions in which $\mathbf{m}$ orders. This anisotropic ordering is a very important physical feature that has been used to explain the behaviour of the magnetic coercivity in polycrystalline materials [55, 58, 57, 56]. The model presented here does not consider the magneto-electric coupling arising due to magnetic ordering heterogeneity, a phenomenon that occurs in some single phase materials at temperatures much lower than those of the multiferroic materials we wish to address with PFC modelling.

2.1 XPFC model of magneto-crystalline interactions

A dimensionless free energy for the magneto-XPFC model is defined as

$$\Delta F = \frac{\Delta \tilde{F}}{k_B T V \bar{\rho}} = \int d\mathbf{r} \left(f_{id} + f_{ex} + f_m + f_c \right), \tag{2.1}$$

where $\tilde{F}$ represents the dimensional free energy, T denotes temperature, k_B Boltzmann's constant and $\bar{\rho}$ is a reference liquid density around which the free energy functional is expanded. The terms in the integrand are free energy densities: f_{id} is the ideal free energy density, f_{ex} is the excess energy density due to atomic interactions, f_m is the magnetic free energy, and f_c is the alloy free energy density.

Due to the similarities between magnetization and electric polarization, it is straightforward to add an additional term f_P to the free energy which is similar in form to f_m. We do so in Ref. [68], however the present work will focus on magnetism alone.

f_{id} and f_{ex} are based on traditional PFC approaches, in particular the XPFC model of Greenwood *et al.* [69, 70]. These are expressed in terms of the reduced PFC density $n(\mathbf{r}) \equiv (\rho(\mathbf{r}) - \bar{\rho})/\bar{\rho}$, where $\rho(\mathbf{r})$ is related to the atomic number density of the material [33, 70] and $\bar{\rho}$ is a reference density of the cDFT expansion from which the XPFC model emerges. f_{id}, the free energy density of an ideal gas, is given by

$$f_{id} = \frac{n^2}{2} - \eta\frac{n^3}{6} + \chi\frac{n^4}{12}, \tag{2.2}$$

an approximation based on the Taylor series of $\rho \log \rho$. In what follows, $\eta = \chi = 1$.

The excess free energy density is based on the XPFC model of Greenwood *et al.* [69, 70]. In this model the free energy of interactions is expressed via a two-point correlation function:

$$f_{ex} = -\frac{1}{2}n(\mathbf{r})\int C_2(|\mathbf{r} - \mathbf{r}'|)n(\mathbf{r}')\, d\mathbf{r}', \tag{2.3}$$

where C_2 controls the atomic structure of the material. C_2 contains Gaussian peaks in reciprocal space at the reciprocal lattice wave vectors corresponding to the crystal structure to be stabilized. This will be discussed in greater detail below.

2.1.1 Ferromagnetic free energy density

The ferromagnetic free energy density f_m is given by:

$$f_m = \omega_B \left\{ \frac{W_0^2}{2}(\nabla \cdot \mathbf{m})^2 + (r_m - \omega_m n^2)\frac{|\mathbf{m}|^2}{2} + \gamma_m \frac{|\mathbf{m}|^4}{4} - \sum_{k=2,4,\ldots} \frac{\alpha_k}{k}(\mathbf{m} \cdot \nabla n)^k - \mathbf{m} \cdot \mathbf{B} + \frac{|\mathbf{B}|^2}{2} \right\}, \tag{2.4}$$

where $\omega_B = B_0^2/(\mu_0 k_B T \bar{\rho})$ sets the scale of magnetic energy, with B_0 a reference magnetic field and μ_0 the magnetic permeability of free space. Two fields, $\mathbf{m}$ and $\mathbf{B}$ are introduced here. $\mathbf{m}$ is the magnetization, the density of magnetic dipole moments in the material, and $\mathbf{B}$ is the magnetic field. Both are expressed in dimensionless units. The first term, $\frac{W_0^2}{2}(\nabla \cdot \mathbf{m})^2$, represents the exchange free energy, the gradient serving as the lowest order representation of the exchange energy in ferromagnetic systems, where the tendency is for magnetic spins to align. W_0 sets the length scale of the exchange interaction. The second and third terms, $(r_m - \omega_m n^2)\frac{|\mathbf{m}|^2}{2} + \gamma_m \frac{|\mathbf{m}|^4}{4}$, are mean field bulk free energy terms which control the ferromagnetic/paramagnetic transition temperature[51]. r_m, ω_m, and γ_m are parameters which set the magnitude of the equilibrium magnetization of the material. The fourth term, $-\sum_{k=2,4,\ldots} \frac{\alpha_k}{k}(\mathbf{m} \cdot \nabla n)^k$, typically truncated at $k = 4$, couples the magnetization to the atomic lattice in a manner that depends on the relative orientation between the magnetization vector and the atomic lattice, with α_k controlling the strength of the coupling. This term controls the magnetocrystalline anisotropy and magnetostriction in the material. The last two terms, $-\mathbf{m}\cdot\mathbf{B} + \frac{|\mathbf{B}|^2}{2}$, define the magnetostatic free energy.

2.1.2 Alloy free energy density

We can also add a contribution to the model to allow for impurity effects in the base material. For a two component system, an additional field c is introduced into the XPFC model, where $c = C(\vec{r}) - \bar{C}$, with C being the local concentration and $\bar{C}$ a reference concentration. The free energy is given by

$$f_c = \frac{1}{2}(a - bn^2)c^2 + \frac{c^4}{4} + \frac{K}{2}\left|\nabla c\right|^2 + \frac{1}{2}\left(\alpha_m|\mathbf{m}|^2\right)\, c\, n^2 \tag{2.5}$$

where a, b and K are constants. The first three terms of f_c are the standard Cahn-Hilliard type terms that favor phase separation at a particular temperature, which depends on the average density through the n^2 term. The last term in f_c favours a ferromagnetic state in regions of the materials where c is negative and paramagnetic space in regions where c is positive. More complex alloys are possible by using more sophisticated expansions. This component of the model is added formally in the model for completeness, but it will not be considered in any simulations reported later.

2.1.3 Magnetostatics and electrostatics

The total magnetic field $\mathbf{B}$ (scaled by B_0) is given by sum of the external and induced magnetic fields:

$$\mathbf{B} = \mathbf{B}_{ext} + \mathbf{B}_{ind}. \tag{2.6}$$

$\mathbf{B}_{ext}$ results from fields imposed externally on the sample. $\mathbf{B}_{ind}$ is induced by the magnetization within the sample, and is given by

$$\mathbf{B}_{ind} = \nabla \times \mathbf{A}, \tag{2.7}$$

where the vector potential $\mathbf{A}$ is in turn given by

$$\nabla^2 \mathbf{A} = -\nabla \times \mathbf{m}. \tag{2.8}$$

2.1.4 Two-point correlation function

The two-point correlation function in the excess free energy (Eq. 2.3) controls the crystal structure that emerges from the liquid. The form we use here was introduced by Greenwood *et al.* and in reciprocal space is given by the maximal envelope of a set of Gaussian peaks[69, 70]:

$$\hat{C}_2(q) = \max\left(\{\hat{C}_2^i(q)\}\right). \tag{2.9}$$

The peaks $\hat{C}_2^i(q)$ are defined as

$$\hat{C}_2^i(q) = e^{-\sigma^2 k_i^2/(2\rho_i\beta_i)}\, e^{-(q-k_i)^2/(2\xi_i^2)}. \tag{2.10}$$

Here, the values $\{k_i\}$ set the locations of the Gaussians in reciprocal space; these are typically set to correspond to the wavevectors of the desired lattice to be stabilized. The temperature is parameterized by σ. The values $\{\xi_i\}$ set the width of the Gaussians—these control the elastic constants of the material (this will be discussed below). The parameters ρ_i and β_i are, respectively, the planar atomic

density and the number of planes corresponding to the ith wavevector.

If a single peak is used, this correlation function can stabilize triangular lattices in two dimensions, or BCC (body-centered cubic) lattices in three dimensions. If additional peaks are used more structures are possible: square lattices in two dimensions; and in three dimensions FCC (face-centered cubic) and HCP (hexagonal closed packed) lattices.

2.2 Dynamics

The density, n, and concentration, c, are both conserved quantities. As a result, their dynamics are driven by fluxes proportional to the variation of the free energy with respect to changes in these fields. Specifically:

$$\frac{\partial n}{\partial t} = M_n \nabla^2 \left(\frac{\delta F}{\delta n} \right) \tag{2.11}$$

$$\frac{\partial c}{\partial t} = M_c \nabla^2 \left(\frac{\delta F}{\delta c} \right) \tag{2.12}$$

where M_n and M_c are parameters setting the diffusivity of the density and concentration respectively.

The magnetization is not conserved. It follows dynamics of the form

$$\frac{1}{\tau_m} \frac{\partial m_i}{\partial t} = -\frac{\delta F}{\delta m_i}, \tag{2.13}$$

where τ_m sets the relaxation time scale. Note that we have not used the usual Landau-Lifshitz-Gilbert (LLG) type equation to model the magnetization dynamics. LLG equations are valid at microscopic time scales where the conservation of

angular momentum remains relevant. At the diffusional time scales studied here, $\mathbf{m}(\mathbf{r}, t)$ represents a locally time-averaged quantity whose dynamics are driven by free energy relaxation.

In order to simulate the model we will require explicit forms of these equations. For the density field,

$$\begin{aligned}\frac{\partial n}{\partial t} =& M_n \nabla^2 \Bigg[n - \frac{t}{2} n^2 + \frac{v}{3} n^3 - \int C_2(|\mathbf{r} - \mathbf{r}'|) n(\mathbf{r}')\, d\mathbf{r}' \\ &+ \omega_B \Big\{ - \omega_m n m^2 + \alpha_2 \Big((\mathbf{m} \cdot \nabla n)(\nabla \cdot \mathbf{m}) + \mathbf{m} \cdot \nabla (\mathbf{m} \cdot \nabla n) \Big) \\ &+ \alpha_4 \Big((\mathbf{m} \cdot \nabla n)^3 (\nabla \cdot \mathbf{m}) + 3 (\mathbf{m} \cdot \nabla n)^2 \mathbf{m} \cdot \nabla (\mathbf{m} \cdot \nabla n) \Big) \Big\} \\ &+ \alpha_m |\mathbf{m}|^2 c\, n \Bigg]. \end{aligned} \tag{2.14}$$

For the concentration field,

$$\frac{\partial c}{\partial t} = M_c \nabla^2 \Big[\left(a - b n^2 \right) c + c^3 - K \nabla^2 c + \frac{1}{2} \alpha_m |\mathbf{m}|^2 n^2 \Big]. \tag{2.15}$$

And for the components of the magnetization field,

$$\begin{aligned}\frac{1}{\tau_m} \frac{\partial m_i}{\partial t} =& W_0^2 \nabla^2 m_i - \left\{ r_m - (\omega_m - \alpha_m c)\, n^2 + \gamma_m |\mathbf{m}|^2 \right\} m_i \\ &+ \alpha_2 (\mathbf{m} \cdot \nabla n)(\partial_i n) + \alpha_4 (\mathbf{m} \cdot \nabla n)^3 (\partial_i n) + B_i. \end{aligned} \tag{2.16}$$

Note that, additionally, Eqs. 2.7 and 2.8 are solved at each time step of the simulation.

These equations represent relaxation dynamics on a diffusional time scale, driven by a free energy functional. As such, they should, in principle, also include a stochastic noise source to re-introduce the fluctuations washed out in the

implied time-averaging. However, as we will not be considering nucleation from the disordered state, nor interface fluctuations, we will leave noise sources out of the dynamics.

2.3 Equilibrium properties

In this section we investigate the equilibrium properties of the model. Specifically we will look at the phase diagram, anisotropy, and the small deformation limit. For this section we will only be considering single component models. Hence f_c will be omitted in what follows.

2.3.1 Amplitude expansions

We will apply amplitude expansions in order to investigate the equilibrium properties of the model. We begin by expanding the crystal structure in a Fourier series:

$$n(\mathbf{r}) = \sum_{\mathbf{q}} \phi_{\mathbf{q}} e^{i\mathbf{q}\cdot\mathbf{r}}, \tag{2.17}$$

where $\mathbf{q}$ are the reciprocal lattice vectors of the crystal structure and $\phi_{\mathbf{q}}$ are their associated amplitudes. The vectors $\mathbf{q}$ can be grouped by magnitude; these groups are collective referred to as *modes*. Thus $\mathbf{q}$ are replaced by $\mathbf{q}_{k,j}$, which refers to vector j of mode k.

To simplify the analysis we make the assumption that all amplitudes corre-

sponding to $\mathbf{q}_{k,j}$ with j in the mode k are equal and real. That is

$$\phi_{\mathbf{q}_{k,j}} = \phi_k \quad : \quad \forall\, j \text{ in } k. \tag{2.18}$$

Figure 2.1 shows the grouping of reciprocal lattice vectors into modes of equal magnitude (dots represent the lattice vectors indexed by j belonging to the same k; modes k are represented by circles). Each mode is labeled with its amplitude ϕ_k, which is the same for all vectors in that mode.

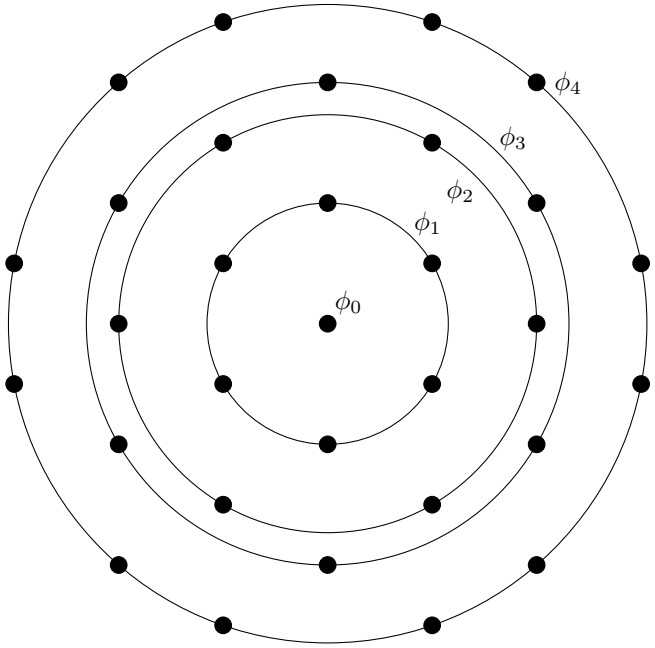


Figure 2.1: Organization of the vectors of a reciprocal triangular lattice according to modes. Dots represent reciprocal lattice vectors, circles connect vectors of equal magnitude, i.e. in the same mode. To each mode an amplitude ϕ_k is assigned.

If we additionally truncate the expansion to the first N modes, our amplitude

expansion becomes:

$$n(\mathbf{r}) \approx \sum_{k=0}^{N} \phi_k \left(\sum_j e^{i\mathbf{q}_{k,j}\cdot\mathbf{r}} \right). \tag{2.19}$$

Using these approximations we may determine the equilibrium properties of the model. Following a well documented approach [71, 72], we begin by inserting Eq. 2.19 into the model free energy density and integrating over a unit cell of the crystal structure of interest. Phase equilibrium is then determined for a given set of system parameters by numerically minimizing the free energy. When $\phi_k = 0, k > 0$ the system is in a liquid state, when $\phi_k \neq 0, k > 0$ the system is in a crystal state. We use the common tangent construction to determine the coexistence region.

2.3.2 Phase diagram

The amplitude expansion of Equation 2.17 is inserted into the free energy (Eq. 2.1). In the mean field limit used to construct the phase diagram, oscillating terms of the form $e^{i(\mathbf{q}_i+\mathbf{q}_j+\ldots)\cdot\mathbf{r}}$ emerge. These integrate to zero unless $\mathbf{q}_i + \mathbf{q}_j + \ldots = 0$; this is known as the *resonance condition*. After integration, the resulting free energy will depend only on the amplitudes $\{\phi_k\}$, the components of the magnetization vector $\{m_i\}$, and the model parameters. This mean field free energy is denoted as $F_{\text{mf}}(\{\phi_k\}, \{m_i\}, \phi_0, \sigma)$, where ϕ_0 has been separated out to signify its role as the system average density.

For now we assume that $\mathbf{m} = 0$. For each set of $\{\phi_0, \sigma\}$, F_{mf} is minimized with respect to the amplitudes $\{\phi_k\}, k > 0$. When the number of modes $N > 1$ it becomes impractical to minimize the free energy analytically so it is minimized numerically. This gives a set of amplitudes $\{\phi_k\}, k > 0$ as a function of the

parameters $\{\phi_0, \sigma\}$. Figure 2.2 shows an example these amplitudes plotted for a fixed σ as a function of ϕ_0. Where the amplitudes are zero (corresponding to a flat density field) the free energy of the liquid phase is lower; where they are non-zero (corresponding to a density field with periodic fluctuations) the free energy of the solid phase is lower. The plot of the free energy (Fig. 2.3) shows a kink at this transition point. The first derivative of the free energy with respect to the density would show a discrete jump at this point, indicating a first order phase transition.

Since ϕ_0 is a conserved parameter, there is a coexistence region between the liquid and solid phases. This region can be found by the common tangent construction. This is done by finding the a secant line that is tangent to the two convex curves on either side of the cusp. The tangent points correspond to the equilibrium densities of the coexisting phases. This procedure is performed numerically.

To construct a phase diagram with more than one solid phases (such as in a BCC/FCC system), we write amplitude expansions of the form of Eq. 2.19 for each potential solid structure. We perform the same procedure as above for each structure and then pick, for each $\{\phi_0, \sigma\}$, the lowest free energy of the structures considered (the free energy of the liquid will be identical for each structure). This produces additional coexistence regions between the various solid phases; these are found by the common tangent construction. Calculation of all coexistence regions at all σ yields the complete phase diagram. Figure 2.4 depicts the full phase diagram. This result closely matches that of Greenwood *et al* in their introduction of the XPFC model [69, 70].

The paramagnetic-ferromagnetic transition line of Fig. 2.4 is calculated from each solid phase using the numerically determined amplitudes computed above. For simplicity it is assumed that the amplitudes are not changed by the introduc-

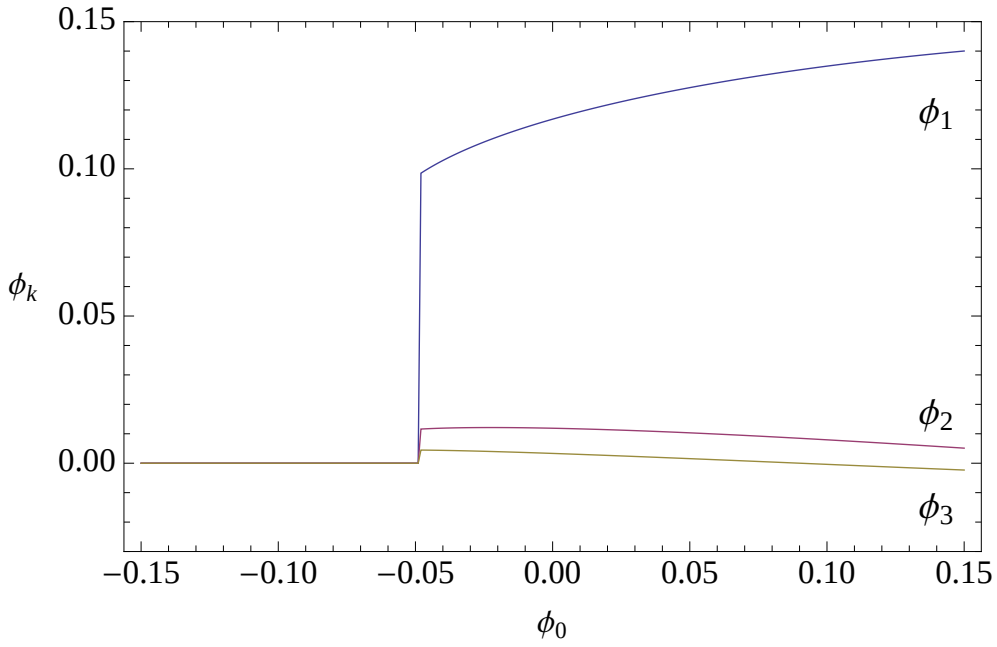


Figure 2.2: Amplitudes of the first three modes in Eq. 2.19 as a function of ϕ_0 with $\sigma = 0.12$, for the case of a single-peaked XPFC correlation with $k_1 = 2\sqrt{2}\pi$, $1/(2\rho_i\beta_i) = 1/24\sqrt{2}$, and $t = v = 1$.

tion of the magnetic free energy. When the amplitude expansion is inserted into the magnetic free energy the result takes on the form of $Am^2 + Bm^4$ with $B > 0$. When $A > 0$ there is a single minimum in the magnetic free energy density at $m = 0$, corresponding to paramagnetic ordering. When $A < 0$ there are two minima at $m \neq 0$, corresponding to ferromagnetic ordering. For each solid phase the value of A depends on the amplitudes, ϕ_k, of the corresponding lattice as well as the average density ϕ_0. For example, the m^2 term for a BCC lattice is

$$\begin{aligned}&\frac{1}{2}\big\{r_m - \omega_m\phi_0^2 + (12k_1^2\alpha_2 - 12\omega_m)\phi_1^2 \\ &+ (12k_1^2\alpha_2 - 6\omega_m)\phi_2^2 + (72k_1^2\alpha_2 - 24\omega_m)\phi_3^2\big\}|\mathbf{m}|^2,\end{aligned} \tag{2.20}$$

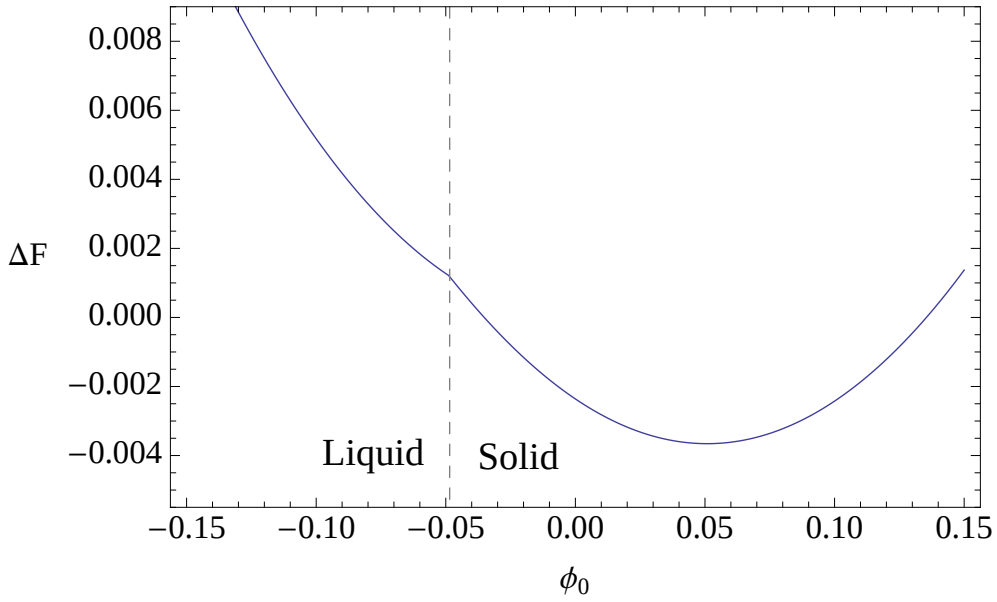


Figure 2.3: Free energy as a function of ϕ_0 for $\sigma = 0.12$. The other parameters are the same as described in Fig. 2.2. Note the kink at the liquid-solid transition.

where $k_1 = |\mathbf{q}_{1,j}|$ is the length of the first set of reciprocal lattice vectors. Inserting the numerically determined amplitudes determines the sign of the coefficient and thus whether the system is in the paramagnetic or ferromagnetic region. As noted above the effect of the magnetic free energy on the amplitudes has been omitted. This assumption is equivalent to assuming that $\mathbf{m} = 0$. At the transition line the *average* magnetization is exactly zero. While there may be spacial variations in the magnetization that average to zero, the magnitude of any such variations would be limited by the gradient term in Eq. 2.4. As a result, the magnetic free energy is unlikely to have a large effect on the phase diagram.

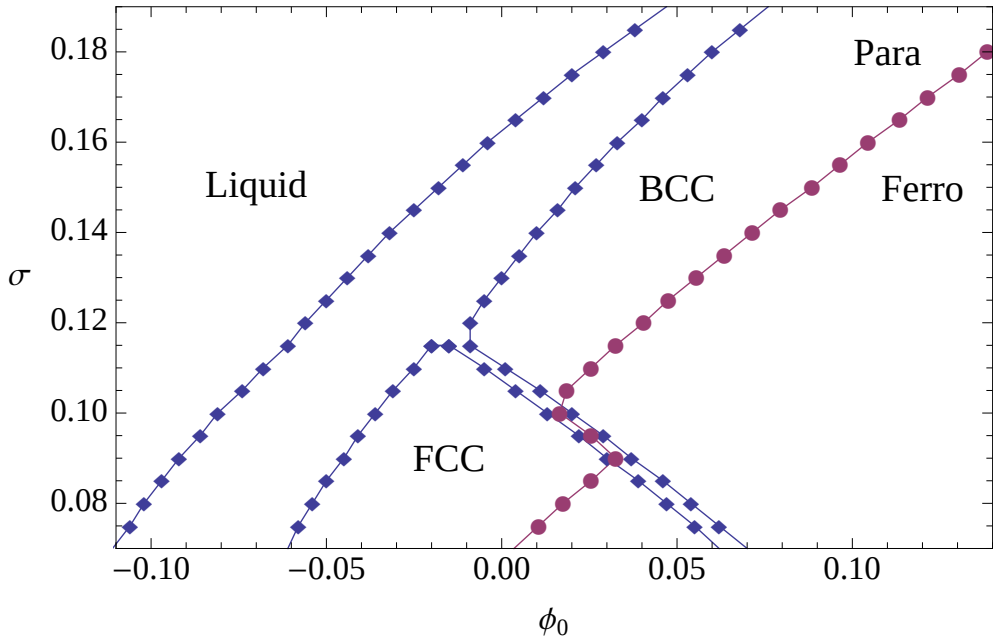


Figure 2.4: Phase diagram for the magneto-XPFC model, showing coexisting liquid, BCC and FCC phases, as well as paramagnetic and ferromagnetic phases. Parameters are $\alpha_2 = .001$, $r_m = .025$, $\omega_m = 0.25$, and $\gamma_m = 1$. Generated using five modes ($N = 5$).

2.3.3 Magnetic anisotropy

Magnetic anisotropy is an essential feature in micromagnetic phenomena. In the literature magnetic anisotropy is generally expressed with respect to a fixed crystal orientation. As an example, for uniaxial anisotropy with symmetry along the z-axis, the lowest order anisotropic free energy per unit volume is given by

$$f_a = K_u m_z^2. \tag{2.21}$$

If $K_u < 0$ there is an *easy axis* along the z direction, along which the magnetization is energetically favoured to lie. If $K_u > 0$ there as a *hard axis* along the z

direction and an *easy plane* in the x-y plane, with the magnetization preferring to line in the plane.

Cubic anisotropy with symmetry along the x, y and z axes is given by

$$f_a = K_c(m_x^2 m_y^2 + m_x^2 m_z^2 + m_y^2 m_z^2), \tag{2.22}$$

with the easy axes along any of the six axes $\pm\hat{x}$, $\pm\hat{y}$ or $\pm\hat{z}$ for $K_c > 0$, or along any of the eight direction vectors $\pm\hat{x} \pm \hat{y} \pm \hat{z}$ for $K_c < 0$.

The above forms cannot be used in the PFC free energy because they assume a given crystal orientation. With PFC the crystal orientation is not *a priori* known, since PFC uses a rotationally invariant free energy which permits crystals to form in any orientation. Instead we must use terms which depend on the density, n. The simplest expression that contains information on the local crystal orientation is ∇n. In order to have magnetic anisotropy that depends on the local crystal orientation, we use terms that couple $\mathbf{m}$ with ∇n in the free energy. Specifically, we use couplings of the form

$$\frac{1}{k}(\mathbf{m} \cdot \nabla n)^k, \tag{2.23}$$

where k is some even integer. To obtain N–fold symmetry it is necessary (but not sufficient) that k be at least of order N. That is, to obtain 2–fold symmetry, $k = 2$, while for 4–fold, $k = 4$, and so on.

Eq. 2.4 contains $\mathbf{m} \cdot \nabla n$ coupling terms of order 2 and 4. The second order term allows for uniaxial anisotropy in systems with HCP ordering. The fourth order term allows for cubic anisotropy in BCC and FCC systems, as well as two dimensional square systems. Two dimensional triangular systems require an ad-

ditional sixth order term.

To obtain an approximation of the long-wavelength form of the anisotropy energy contained in the $(\mathbf{m}\cdot\nabla n)^k$ terms in Eq. 2.4, a single mode expansion,

$$n(\mathbf{r}) \approx \phi_0 + \sum_{\mathbf{q}_j} \phi e^{i\mathbf{q}_j\cdot\mathbf{r}}, \tag{2.24}$$

is inserted into Eq. 2.23, where $\mathbf{q}_j$ are the lowest order reciprocal lattice vectors of a crystal lattice. The result is

$$\frac{1}{k}\Big(\sum_{\mathbf{q}_j}(i\mathbf{m}\cdot\mathbf{q}_j)\phi e^{i\mathbf{q}_j\cdot\mathbf{r}}\Big)^k. \tag{2.25}$$

Expanding out Eq. 2.25 and applying the resonance condition (i.e., terms of the form $e^{i(\mathbf{q}_j+\mathbf{q}_k+\ldots+\mathbf{q}_l)\cdot\mathbf{r}}$ integrate to zero under coarse graining unless $\mathbf{q}_j+\mathbf{q}_k+\ldots+\mathbf{q}_l=0$), an approximation of the anisotropic free energy in the phase field limit is obtained.

If we apply this procedure for a BCC lattice with $k=4$, we find that

$$\begin{aligned} f_{a,\mathrm{BCC}} =& \frac{1}{V_u}\int d\mathbf{r}\left(-\frac{\alpha_4}{4}(\mathbf{m}\cdot\nabla n)^4\right) \\ \approx& -\frac{27}{2}\alpha_4(k_1\phi)^4\Big(|\mathbf{m}|^4-\frac{7}{9}(m_x^2m_y^2+m_x^2m_z^2+m_y^2m_z^2)\Big), \end{aligned} \tag{2.26}$$

where the integration is over unit cell of volume V_c. Note the similarity of the final term in this formula to Eq. 2.22. Since the result depends on the amplitude, ϕ, the anisotropy depends on both the average density and the temperature parameter through the amplitude's dependance on these parameters. This also ensures that anisotropy vanishes in the liquid region of the phase diagram as expected.

The parameter α_4 can be either positive or negative in order to model different forms of anisotropy. Care must be taken, however, because for a large enough positive α_4, the coefficient of m^4 in Eq. 2.26 will exceed $\gamma_m/4$, leaving no finite global minimum for the free energy as a function of $\mathbf{m}$. A similar issue exists with the ϕ^4 coefficient and the coarse grained term $vn^4/12$ in the free energy. To avoid these issues α_4 must either be sufficiently small, or terms higher order than fourth order in n and $\mathbf{m}$ must be added to f_{id} and f_m, respectively.

Applying the same procedure for an FCC lattice gives

$$f_{a,\mathrm{FCC}} \approx -2\alpha_4(k_1\phi)^4\Big(3|\mathbf{m}|^4 - 4(m_x^2m_y^2 + m_x^2m_z^2 + m_y^2m_z^2)\Big). \tag{2.27}$$

The above derivation considered only a liquid-BCC-FCC system, however the XPFC formalism can also be used to model systems with hexagonal closed-packed (HCP) structure[69]. Magnetically, HCP systems show uniaxial anisotropy, and so only a second order term is required to produce anisotropy. An approximation for this anisotropy is computed in an analogous manner as above, but now using the $(\mathbf{m}\cdot\nabla n)^2$ term, and taking into account the structure factors that arise due to the additional basis atom needed to describe HCP crystals. A single mode expansion is not sufficient in this case because the lowest order (smallest $\mathbf{q}$) mode of the underlying hexagonal Bravais lattice vanishes when the structure factor of the two atom basis is computed. The next lowest order mode lies in the basal plane; we denote it's amplitude by ϕ. With these modifications to Eq. (2.24), the phase field (amplituide) limit of the α_2 term becomes

$$-\frac{\alpha_2}{2}(\mathbf{m}\cdot\nabla n)^2 \approx -\frac{4}{3}\alpha_2\phi^2k_1^2(|\mathbf{m}|^2 - m_z^2), \tag{2.28}$$

which is consistent with the form for uniaxial anisotropy.

2.3.4 Small deformation limit and magnetorestriction

Phase field crystal models naturally contain an elastic response to deformation. An interesting consequence of magnetocrystalline interactions is that crystal phases to deform in the presence of a magnetic field. This effect is known as magnetostriction in ferromagnetic materials. This section derives the magnetostriction coefficients of the present model.

Before proceeding, the density expansion (Eq. 2.17) is modified by assuming that the amplitudes are complex and can vary in space, i.e.

$$n(\mathbf{r}) = \sum_{\mathbf{q}_j} \eta_{\mathbf{q}_j}(\mathbf{r}) e^{i\mathbf{q}_j \cdot \mathbf{r}}. \tag{2.29}$$

The amplitudes are then decomposed in the form of a real magnitude and complex phase. Specifically

$$\eta_{\mathbf{q}_j}(\mathbf{r}) = \phi_{\mathbf{q}_j} e^{i\mathbf{q}_j \cdot \mathbf{u}(\mathbf{r})}, \tag{2.30}$$

where $\mathbf{u}(\mathbf{r})$ is a displacement field that serves to strain the system. Inserting Eq. 2.30 into Eq. 2.29 yields

$$n(\mathbf{r}) = \sum_{\mathbf{q}_j} \phi_{\mathbf{q}_j} e^{i\mathbf{q}_j \cdot (\mathbf{u}(\mathbf{r}) + \mathbf{r})}. \tag{2.31}$$

We denote the spatial derivatives of $\mathbf{u}(\mathbf{r})$ as $\partial_{r_w} u_v = u_{vw}$ ($v, w = 1, 2, 3$ are spatial indices). In the small deformation limit we can write $u_v(\mathbf{r}) = \sum_w u_{vw} r_w$. Then,

by defining $q'_w = \sum_v q_v u_{vw}$, Eq. 2.31 can be re-expressed as

$$n(\mathbf{r}) = \sum_{\mathbf{q}_j} \phi_{\mathbf{q}_j} e^{i(\mathbf{q}_j + \mathbf{q'}_j)\cdot \mathbf{r}}. \tag{2.32}$$

In reciprocal space Eq. 2.32 becomes:

$$\hat{n}(\mathbf{k}) = \sum_{\mathbf{q}_j} \phi_{\mathbf{q}_j} \delta((\mathbf{q}_j + \mathbf{q'}_j) - \mathbf{k}). \tag{2.33}$$

Eq. 2.33 is used below to compute contributions to the elastic energy arising from the relevant terms in the PFC free energy.

Elastic energy density of the excess term

Substituting Eq. 2.33 into the excess term of the free energy, f_{ex} (Eq. 2.3), gives, in reciprocal space,

$$F_{el}[\hat{n}] = -\sum_{\mathbf{q}_j} \phi^2_{\mathbf{q}_j} \hat{C}_2(\mathbf{q}_j + \mathbf{q'}_j). \tag{2.34}$$

The two-point correlation function, $\hat{C}_2$, is rotationally symmetric. For small $\mathbf{q'}_j$, we can write the *magnitude* of $\mathbf{q}_j + \mathbf{q'}_j$ as $(\mathbf{q}_j + \mathbf{q'}_j) \cdot (\mathbf{q}_j/q_j) = q_j + \frac{1}{q_j}(\mathbf{q'}_j \cdot \mathbf{q}_j)$, where $q_j = |\mathbf{q}_j|$. This gives

$$F_{el}[\hat{n}] = -\sum_{\mathbf{q}_j} \phi^2_{\mathbf{q}_j} \hat{C}_2\Big(q_j + \frac{1}{q_j}(\mathbf{q'}_j \cdot \mathbf{q}_j)\Big). \tag{2.35}$$

The peaks in the $\hat{C}_2$ correlation kernel (Eq. 2.9) can be approximated, near the

peak, by the second order expansion

$$\hat{C}_2^i(\mathbf{q}) \approx A\bigg(1 - K(q - k_i)^2 + \dots\bigg), \tag{2.36}$$

where

$$A = e^{-\frac{\sigma^2 k_1^2}{2\beta_1 \rho_1}}, \qquad K = \frac{1}{\xi_1^2}. \tag{2.37}$$

Truncating this expansion at quadratic order and keeping only the peak corresponding to the first mode of the Bravais Fourier series of a BCC lattice (i.e., $i = 1$ with $k_1 = |\mathbf{q}_{1,j}|$) gives, after Eq. 2.36 is substituted into Eq. 2.35,

$$F_{el}[\hat{n}] \approx -\phi^2 A\bigg(12 - 2Kk_1^2\bigg(\sum_i \epsilon_{ii}^2 + \sum_{i<j}(2\epsilon_{ij}^2 + \epsilon_{ii}\epsilon_{jj})\bigg)\bigg), \tag{2.38}$$

where $\epsilon_{ij} = \epsilon_{ji} = (u_{ij} + u_{ji})/2$ is the infinitesimal strain tensor. Or, with the values of A and K inserted,

$$F_{el,\text{XPFC}}[\hat{n}] \approx -\phi^2 e^{-\frac{\sigma^2 k_1^2}{2\beta_1 \rho_1}}\bigg(12 - \frac{2k_1^2}{\xi_1^2}\bigg(\sum_i \epsilon_{ii}^2 + \sum_{i<j}(2\epsilon_{ij}^2 + \epsilon_{ii}\epsilon_{jj})\bigg)\bigg). \tag{2.39}$$

Magneto-elastic energy term

Next we consider the magneto-elastic energy term of the free energy density:

$$f_{me,\text{XPFC}}[\hat{n}] = \frac{1}{V_u}\int d\mathbf{r}\left(-\frac{\alpha_2}{2}(\mathbf{m}\cdot\nabla n)^2\right). \tag{2.40}$$

If we substitute the density expansion of Eq. 2.31 into the the expression $\mathbf{m} \cdot \nabla n$, we get

$$\mathbf{m} \cdot \nabla n = \sum_{\mathbf{q}_j} i\phi_{\mathbf{q}_j}((\mathbf{m} \cdot \nabla)(\mathbf{u} \cdot \mathbf{q}_j) + \mathbf{m} \cdot \mathbf{q}_j)e^{i\mathbf{q}_j \cdot (\mathbf{u}+\mathbf{r})}. \tag{2.41}$$

Substituting into the magneto-elastic energy term, assuming a BCC lattice for the density expansion, and applying the resonance condition, we find that

$$\begin{aligned} f_{me,\text{XPFC}}[\hat{n}] &= \frac{1}{V_u} \int d\mathbf{r} \left(-\frac{\alpha_2}{2} (\mathbf{m} \cdot \nabla n)^2 \right) \\ &\approx -2\alpha_2 \phi^2 k_1^2 \Big(\sum_i m_i^2 (1 + 2\epsilon_{ii}) + 4 \sum_{i<j} m_i m_j \epsilon_{ij} \Big). \end{aligned} \tag{2.42}$$

Calculation of magnetostriction constants

Following Kittel[73], a cubic system is expected to have a magneto-elastic free energy of the form

$$\begin{aligned} F_{me} =& B_1 \sum_i m_i^2 \epsilon_{ii} + B_2 \sum_{i<j} m_i m_j \epsilon_{ij} \\ &+ \frac{1}{2} c_{11} \sum_i \epsilon_{ii}^2 + 2c_{44} \sum_{i<j} \epsilon_{ij}^2 + c_{12} \sum_{i<j} \epsilon_{ii} \epsilon_{jj}. \end{aligned} \tag{2.43}$$

Minimizing this free energy with respect to all the strains we obtain the following stress-free strains:

$$\begin{aligned} \epsilon_{ii} &= \frac{B_1}{c_{11} - c_{12}} \left(\frac{|\mathbf{m}|^2 c_{12}}{c_{11} + 2c_{12}} - m_i^2 \right), \\ \epsilon_{ij} &= -\frac{B_2}{4c_{44}} m_i m_j, \quad i \neq j. \end{aligned} \tag{2.44}$$

The magnetostriction constants $\lambda_{100}, \lambda_{111}$ are defined by [73]:

$$\frac{\delta l}{l} = \frac{3}{2}\lambda_{100}\left(\sum_i \frac{m_i^2\beta_i^2}{|m|^2} - \frac{1}{3}\right) + 3\lambda_{111}\sum_{i<j}\frac{m_i m_j \beta_i \beta_j}{|m|^2}, \tag{2.45}$$

where $\delta l/l$ is the extension of the sample along the direction of the unit vector β due to the magnetization. It can be shown that [73]

$$\frac{\delta l}{l} = \sum_{i\leq j}\epsilon_{ij}\beta_i\beta_j. \tag{2.46}$$

If we substitute the solutions from Eq. 2.44 into Eq. 2.46, drop constant terms, and compare with Eq. 2.45, we obtain the magnetostriction constants[73]:

$$\lambda_{100} = -\frac{2}{3}\frac{B_1|m|^2}{c_{11}-c_{12}}; \qquad \lambda_{111} = -\frac{1}{12}\frac{B_2|m|^2}{c_{44}}. \tag{2.47}$$

Now if we compare Eq. 2.43 to $F_{el,\text{XPFC}} + F_{me,\text{XPFC}}$ we can see that

$$\begin{aligned} B_1 = -4\alpha_2\phi^2k_1^2, \quad B_2 = -8\alpha_2\phi^2k_1^2, \\ c_{11} = 4\phi^2k_1^2AK, \quad c_{44} = 2\phi^2k_1^2AK, \\ c_{12} = 2\phi^2k_1^2AK, \end{aligned} \tag{2.48}$$

which then yields, from Eq. 2.47,

$$\lambda_{100} = \frac{4}{3}\frac{\alpha_2}{AK}|m|^2; \quad \lambda_{111} = \frac{1}{3}\frac{\alpha_2}{AK}|m|^2. \tag{2.49}$$

We can see here that $\lambda_{100} = 4\lambda_{111}$, for a single mode approximation. Hence it is not possible with a single peaked correlation kernel to tune the ratio of λ_{100} to λ_{111}

separately in order to match real materials which exhibit other ratios. In order to tune the magnetostriction separately either the elastic anisotropy, $c_{44}/(c_{11}-c_{12})$, or the ratio of B_1 to B_2 must be tuned.

The ratio of B_1 to B_2 is independent of the correlation function. The elastic anisotropy is not. However, tuning the elastic anisotropy cannot be be achieved with a single peak in the XPFC correlation function. If we wish to tune the elastic anisotropy, and therefore the ratio of λ_{100} to λ_{111}, it is necessary to add a second peak to the correlation function, which corresponds to the second BCC mode. Adding a second peak and following the same general approach as above, but with a two-mode density expansion, gives the elastic free energy

$$F_{el,\text{XPFC2}}[\hat{n}] \approx 2\phi_1^2 k_1^2 \Bigg((A_1 K_1 + 2r^2 A_2 K_2) \sum_i \epsilon_{ii}^2 + A_1 K_1 \sum_{i<j} (2\epsilon_{ij}^2 + \epsilon_{ii}\epsilon_{jj}) \Bigg), \tag{2.50}$$

where we define $r = \phi_1/\phi_2$, and

$$A_i = e^{-\frac{\sigma^2 k_i^2}{2\beta_i \rho_i}}; \qquad K_i = \frac{1}{\xi_i^2}. \tag{2.51}$$

Constant terms have been ignored in the above.

Similarly, expanding f_m with a two mode density expansion yields the two-mode magneto-elastic energy to second order:

$$F_{me,\text{XPFC2}}[\hat{n}] \approx -2\alpha_2 \phi_1^2 k_1^2 (1+r^2) \Big(\sum_i m_i^2 (1+2\epsilon_{ii}) + 4 \sum_{i<j} m_i m_j \epsilon_{ij} \Big). \tag{2.52}$$

Comparing these results to F_{me} in Eq. 2.43 gives

$$\begin{aligned} B_1 = -4\alpha_2\phi^2 k_1^2(1+r^2), \quad B_2 = -8\alpha_2\phi^2 k_1^2(1+r^2), \\ c_{11} = 4\phi^2 k_1^2(A_1K_1 + 2r^2 A_2K_2), \\ c_{44} = 2\phi^2 k_1^2 A_1K_1, \quad c_{12} = 2\phi^2 k_1^2 A_1K_1. \end{aligned} \tag{2.53}$$

The magnetostriction constants now become

$$\lambda_{100} = \frac{4}{3}\frac{\alpha_2(1+r^2)}{A_1K_1 + 4r^2A_2K_2}|m|^2; \tag{2.54}$$

$$\lambda_{111} = \frac{1}{3}\frac{\alpha_2(1+r^2)}{A_1K_1}|m|^2. \tag{2.55}$$

Now it can be seen that $\lambda_{100} = 4\lambda_{111}(1 + 4r^2(A_2K_2)/(A_1K_1))^{-1}$. Although A_1 and A_2 are fixed in the XPFC model, K_1 and K_2 (Eq. 2.51) are tunable parameters. The magnetostriction constants can therefore be tuned individually through their dependence on the elastic anisotropy, which is controlled by the addition of higher order peaks in the XPFC correlation function. Note that A_1, A_2 and r in general depend on the system parameters of temperature (σ) and average density, making the magnetostriction constants functions of temperature and average density.

A similar approach could be followed for other crystal symmetries such as FCC.

An approach for changing the ratio change the ratio of λ_{100} to λ_{111} by changing the ratio of B_1 to B_2 is discussed in Appendix D.

2.4 Numerical tests and applications of model

2.4.1 Ferromagnetism in square polycrystalline order

We use our model to examine ferromagnetism in a single-component XPFC model. Using the XPFC correlation function that gives two-dimensional square crystals, along with a magnetic free energy with 4-fold anisotropy, we tested magnetic hysteresis and the effects of magnetic fields on grain growth in a pure material. Simulations were carried using Euler's method. The numerical mesh spacing was $\Delta x = 0.1$ and the time steps were $\Delta t = 0.001$. Unless otherwise stated, parameters used were: $(t, v) = (1, 1)$, $(W_0, r_m, \omega_m, \gamma_m) = (0.1, 0.01, 0.4, 1)$, $(\xi_1, \xi_2) = (0.9, 1.27279)$, $(\rho_1, \rho_2) = (1, \sqrt{2}/2)$, $(\beta_1, \beta_2) = (4, 4)$, and $(M_n, \tau_m) = (1, 1)$. Other parameters are specified in the text.

Magnetic Hysteresis

We consider here the simplest case of magnetic hysteresis of a perfect crystal, with an alternating external magnetic field, $\mathbf{B}$, aligned to the crystal's magnetocrystalline easy axis. In this situation the magnetic part of the free energy has the form

$$f_m(\mathbf{m}) = f_1(\mathbf{m}) - \mathbf{m} \cdot \mathbf{B}. \tag{2.56}$$

In general this free energy will have a number of stationary points defined by $\partial f_m / \partial m_i = 0$. Those which are also local minima will be stable and the magnetization can come to rest there. When the external field $\mathbf{B}$ is non-zero some local minima can become metastable. When a large enough external field is oriented

opposite to the magnetization, the metastable well in which the magnetization lies becomes unstable and the magnetization abruptly reverses. Reversing the external field repeats this cycle in the opposite direction. Plotting the magnetization against the external field results in a hysteresis loop, such as the one shown in Fig. 2.5 for a two dimensional single square crystal.

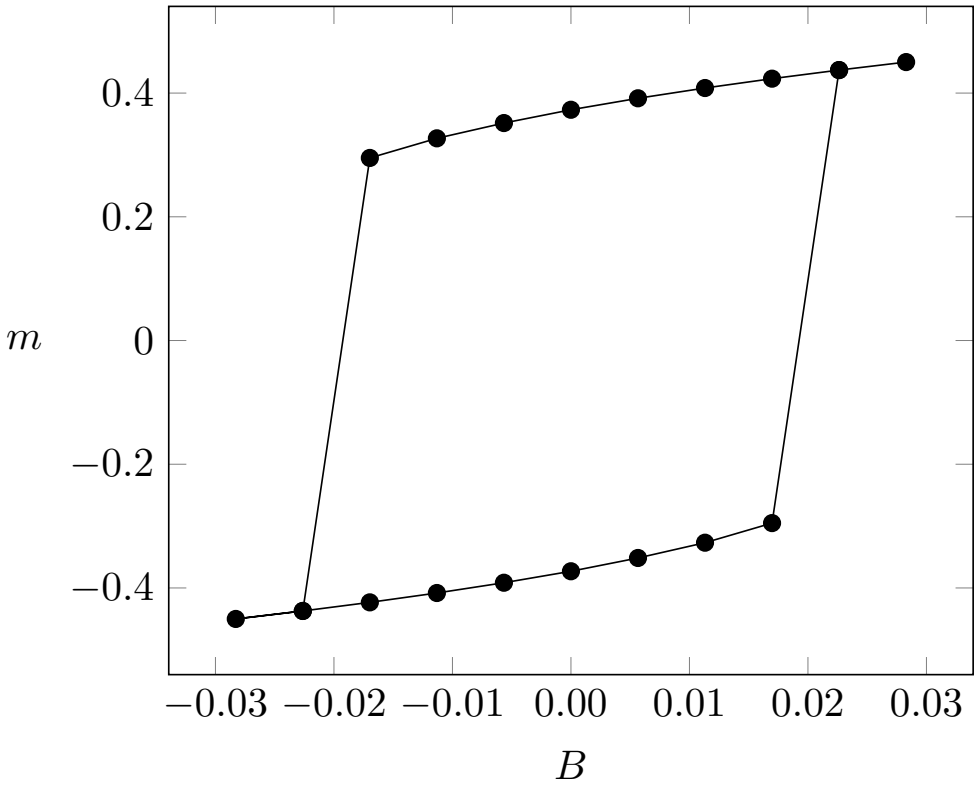


Figure 2.5: Hysteresis loop for a single two dimensional square crystal. System parameters are as stated at the beginning of the section with $\sigma = 0.04$, $n_0 = 0.05$, and $(\alpha_2, \alpha_4) = (0.001, -0.01)$.

To produce the hysteresis loop in Fig. 2.5 the system was first brought to magnetic saturation by an initially strong external field. After a set amount of time the process of measuring the hysteresis loop proceeded by sweeping the external field

through successively decreasing values. For each external field value the magnetization was allowed to relax until $d\mathbf{m}/dt$ was smaller than a convergence value (10^{-5} in Fig. 2.5), ensuring that the magnetization had reached a local minima corresponding to that value of the external field. Once the magnetization had relaxed the value was recorded and the external field was then decreased to its next value. As the strength of the external field increases in the negative direction the magnetization eventually reverses direction. This reversal gives the drop from the upper left quadrant to the lower left quadrant of the loop in Fig. 2.5. Once the external field was at its negative extreme the process was reversed and the external field increased at each step back towards its initial value, causing the magnetization to reverse again. This reversal produces the raise from the lower right to upper right quadrants of the loop in Fig. 2.5. Fig. 2.6 plots the magnetization (both m_x and m_y) versus time for the entire process. Note that the magnetization reverses by rotation of the magnetization vector.

Solidification under an external field

The influence of external magnetic fields on the growth of crystal grains has potential applications in microstructural engineering. To investigate the role that external magnetic fields play on crystal growth, we ran simulations with identical initial conditions and pseudorandom seeds, differing only in presence of an external magnetic field. Crystal seeds of random orientations grow, impinge and coarsen with or without the influence of an external field. This process is depicted schematically in Fig. 2.7.

A large two dimensional system ($4000\Delta x \times 4000\Delta x$) with 160 randomly oriented seeds of square lattice crystals was simulated. Solidification was initiated

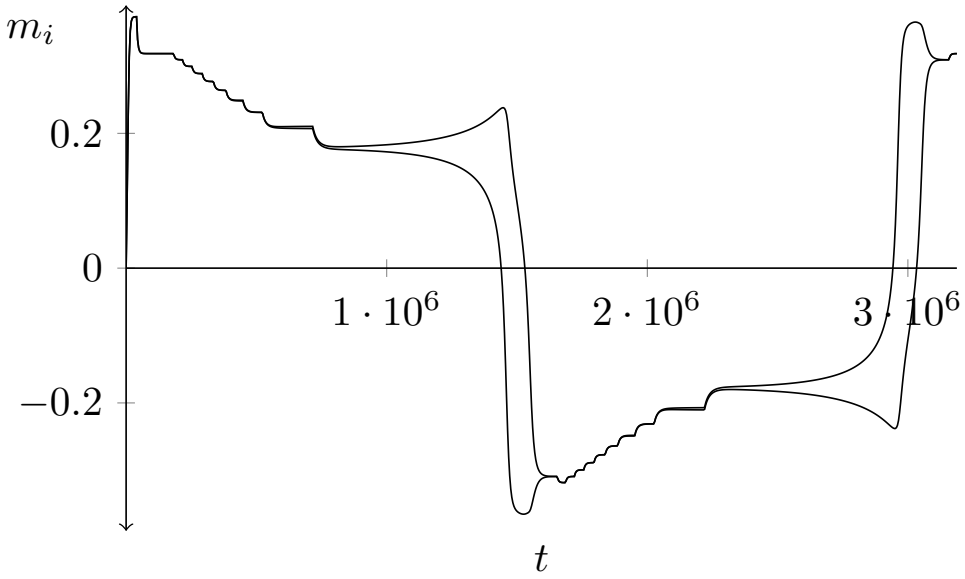


Figure 2.6: Plot of magnetization components m_x and m_y versus time (t) for the hysteresis loop in Fig. 2.5. Note that the magnetization reverses by rotation of the magnetization vector.

following a quench to $\sigma = .04$ and $n_0 = .05$. We set $(\alpha_2, \alpha_4) = (0.001, -0.0005)$; with these values the magnetic easy axes lie along the diagonals of the unit cell. Other parameters are as discussed in the introduction to this section. Two simulations are run with the same pseudorandom number seed: one with an external field and one without. A snapshot in time of the density field $n(\vec{r}, t)$ for the two cases is shown in Fig. (2.8) for $t = 5.75 \times 10^4 \Delta t$. The red lines mark the polycrystalline grain boundaries in the two cases. Qualitatively it appears that the grains whose magnetic easy axes are aligned favourably with the external field are favoured in the final, solidified structure.

In order to investigate more quantitatively the effects of the external field on grain orientation, we computed the power spectrum of the density field n as a function of orientation (θ) at the distance in k-space of the first Bragg peak. The differ-

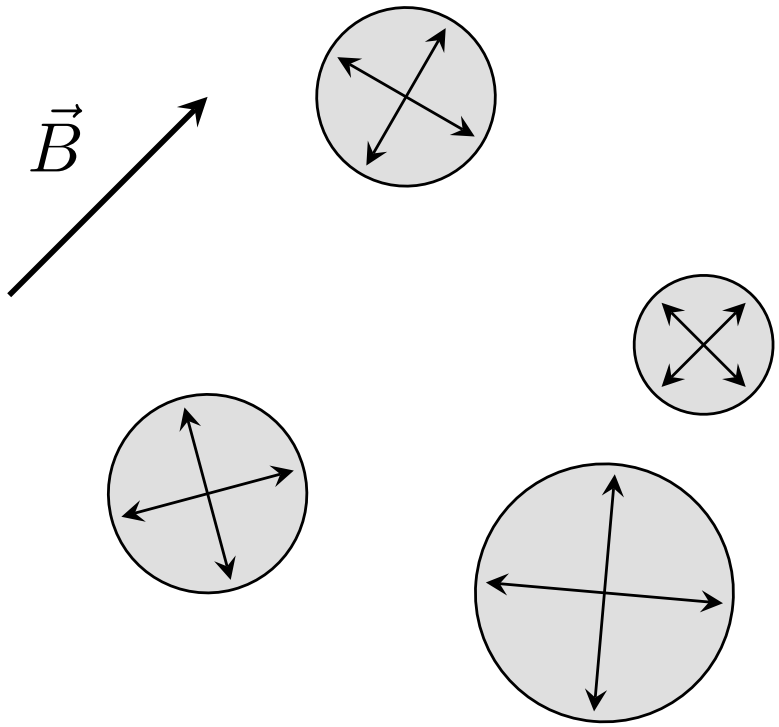


Figure 2.7: Randomly oriented crystal seeds growing under the influence of an external field. Arrows indicate the orientation of the seed crystals.

ence in these power spectra is depicted in Fig. 2.9 after 1.4×10^5 time steps. The external field is oriented at an angle of $\theta = \pi/4$; thus a crystal grain oriented with its principle reciprocal lattice vectors along the directions $\theta = 0, \pi/2, \pi, 3\pi/2$ will have its easy axes aligned to the external field. Fig. 2.9 shows that a greater portion of grains evolve to become aligned along these angles when the external field is present.

We also compared the power spectra of the same system at late and early times. Fig. 2.10 shows such a comparison for the system with an external field present, where the density field is evaluated at $t_{\rm early} = 1 \times 10^4 \Delta t$ and $t_{\rm late} = 1.4 \times 10^5 \Delta t$. From this figure we can draw the same conclusion as in Fig. 2.9: the presence

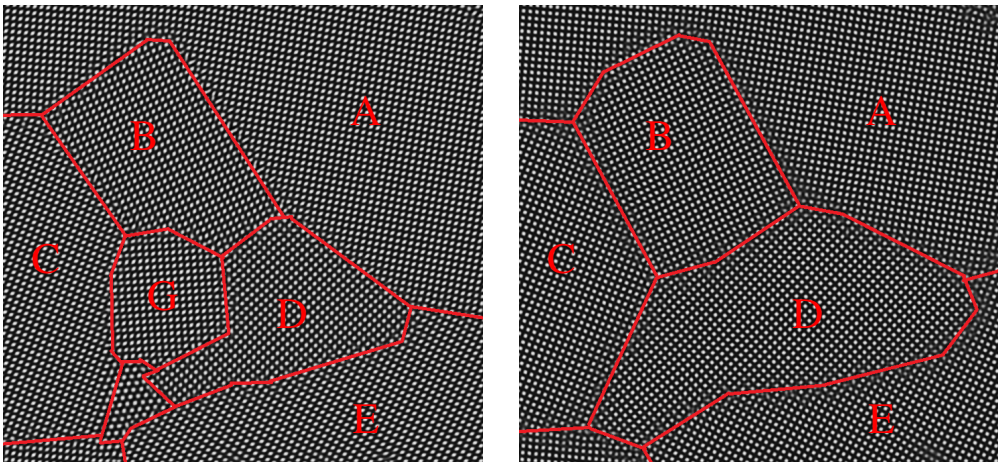


Figure 2.8: Comparison of initially identical systems grown under an external magnetic field (left) and no field (right). Crystal grains aligned with the x-y axes (such as A and G) have their easy axes aligned with the external field, and grow at the expense of those that are not aligned (such as D).

of the external field results in preferential growth for those grains with easy axes which are aligned to it.

A potentially interesting application of the magneto-XPFC model introduced in this chapter is the prediction of how to control microstructure of ferromagnetic materials. Specifically, our results show that it is possible to use external magnetic fields to influence the domain orientation and anisotropy of grains starting from solidification and continuing into grain growth. It would be interesting as a future direction of research to growth ferromagnetic materials experimentally in an external magnetic field, and compare the statistics of grain growth in these materials to predictions of the present XPFC model.

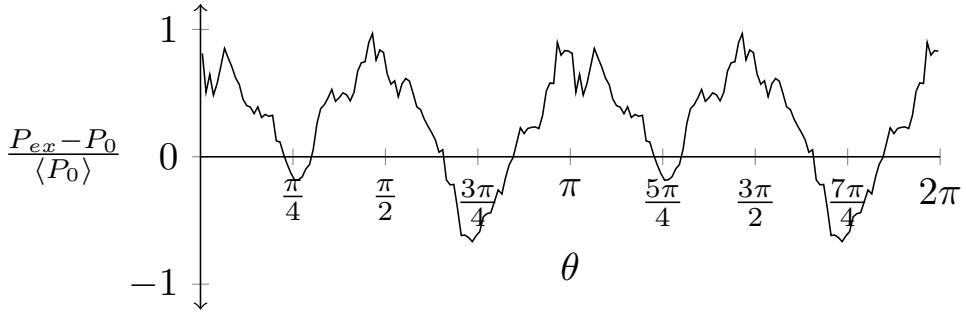


Figure 2.9: Difference in power spectra of polycrystalline solid grown in an external field (P_{ex}) and no external field (P_0), respectively.

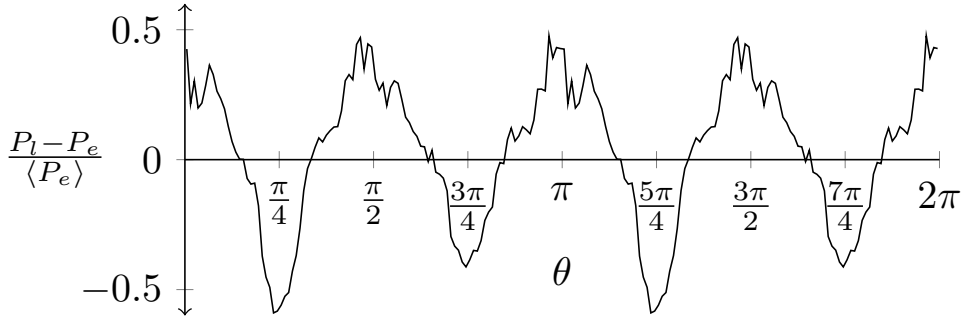


Figure 2.10: Difference in power spectra of polycrystalline solid grown in an external field at late (P_l) and early (P_e) times, respectively.

Chapter 3

Three Point Correlation Functions for XPFC Modelling of Graphene

Graphene is an exciting, and recently discovered, material consisting of a two dimensional, single atomic layer of carbon atoms arranged in a hexagonal lattice. It exhibits interesting electrical[74, 75] and mechanical properties[76, 77]. Crystals of graphene can be obtained by exfoliating cleaved graphite samples onto an oxidized silicon wafer to produce flakes of graphene[78]. A scalable method for obtaining graphene is through chemical vapour deposition (CVD) [79, 80], a method that produces a polycrystalline material. While theoretically graphene can be about a hundred times stronger than steel, the properties of graphene realized in experiments typically reveal a wide variability, up to an order of magnitude from their theoretical predictions[81, 82]. Variability in the strength properties of graphene, and in particular their relationship to the defect structure, still remains largely unexplored, particularly theoretically. Recent work suggests that it is linked to the defect microstructure at grain boundaries[81, 83, 82]. The topolog-

ical defects in graphene typically take the form of periodic patterns of heptagonal and pentagonal disclinations, the patterning of which are dictated by the tessellation requirements of atoms in adjacent grains[84, 85, 86]. The complexity of forming and measuring graphene, however, make it challenging to experimentally isolate and examine the role of specific defects and grain boundaries on the growth and properties of this material.

Computational modelling can serve as a route for theoretically understanding the difficult to measure properties of graphene. First principles studies are useful in examining the adsorption process of carbon onto metal surfaces during graphene formation[87]. Molecular dynamics (MD) studies of graphene have been successful at predicting the anisotropy of graphene morphologies on metal surfaces[88] or the energy of specific defect structures[86]. On the continuum scale, phase field models have been used to study how anisotropic diffusion of carbon on a surface can yield the formation of the dendritic graphene structures[89]. To date, there has not been a model that can address both the atomically varying defect microstructures of graphene alongside its nucleation and diffusional growth kinetics from a disordered state on a surface.

As mentioned in the introduction, phase field crystal (PFC) modelling is a promising approach for modelling many microstructure phenomena. PFC models naturally capture many of the salient physics of nucleation, polycrystalline solidification, grain boundaries[90, 91, 41, 92, 93] and multi-component, multi-phase solidification[35, 43, 72, 37]. PFC models also capture, in the context of a single order parameter, elasticity and plasticity phenomena relevant to solid state processes such as dislocation source creation, dislocation stability[44, 94] and creep[45]. The most important feature of PFC-type models is that they incor-

porate the above phenomena from atomic to micron length scales and over diffusional times scales, where the emergent properties of non-equilibrium phase transformations are typically manifested.

The original PFC model was predominately used for the study of two dimensional triangular and three dimensional BCC crystal symmetries[95, 90]. Later XPFC models introduced multi-peaked two-point correlation kernels in the non-local part of the free energy that allowed for a simple yet robust means to simulate most of the common metallic crystal structures (2D square, BCC, FCC, HCP) in phase transformations [17, 96, 70]. More recently, a new multi-peaked two-point correlation was introduced to stabilize more exotic 2D structure such as graphene and Kagome lattices, and a morphological phase diagram distinguishing the stability ranges of the two solid phases was explored numerically[48]. However this model can leads to unstable structures within the graphene part of is phase diagram and does not support phase coexistence between the solid and any disordered phase, precluding it from CVD type studies. Indeed, it has proven quite difficult to stabilize highly anisotropic 2D and 3D materials in PFC modelling using only two-point correlations.

This chapter introduces a new structural PFC theory that breaks with the tradition of previous PFC models and expands the free energy up to three-point correlations in the PFC density field. Unlike previous PFC theories, the two-point excess term is based on a simple repulsive interaction. It is shown that this allows for stabilization of triangular lattices in two dimensions. In this formalism, more complex crystal structures that are describable by a particular bond angle are stabilized using a new, rotationally invariant, three-point correlation function we introduce for the excess free energy. This term allows for the stabilization of

triangular, square and graphene lattices in two dimensions. It is noteworthy that beyond stabilizing the aforementioned structures, this formalism also allows for stable coexistence of these structures with a disordered phase, a feature crucial for modelling nucleation and growth of polycrystalline 2D materials from a vapour or a disordered arrangement of atoms on a surface. In preparation for future applications, this chapter highlights the derivation of our new XPFC free energy and examines its equilibrium properties. It then uses dynamical simulations to demonstrate defect structures produced in polycrystalline samples and compares these to experimental observations.

3.1 Simplified density functional theory

The dimensionless density, n, is defined in the usual way as $n = (\rho - \bar{\rho})/\bar{\rho}$, where ρ is the PFC density field and $\bar{\rho}$ is the reference density around which a functional expansion of the free energy is carried out. With n as an order parameter, we expand the free energy around the reference density $\bar{\rho}$ in a functional series as

$$\frac{\Delta F}{k_B T \bar{\rho}} = F_{id}[n] + F_{ex,2}[n] + F_{ex,3}[n]. \tag{3.1}$$

where F_{id} is the ideal free energy of the system, which ignores particle interactions. Its form is expanded from its fundamental logarithmic form ($\rho \log \rho$) based on a Landau expansion in the order parameter n according to the usual fourth order form,

$$F_{id} = \int d\mathbf{r} \left\{ \frac{n^2}{2} - \eta \frac{n^3}{6} + \chi \frac{n^4}{12} \right\}, \tag{3.2}$$

where η and ξ are dimensionless parameters. In what follows, we will set $\eta = \xi = 1$ for simplicity. Their form can be used in general to tune the quantitative form of the bulk free energy.

The terms $F_{ex,2}$ and $F_{ex,3}$ constitute the excess free energy, incorporating two and three body interactions respectively. The excess free energy due to two body interactions, $F_{ex,2}$, takes the form

$$F_{ex,2} = -\frac{1}{2}\int n(\mathbf{r})\int C_2(\mathbf{r}-\mathbf{r}')n(\mathbf{r}')\,d\mathbf{r}'\,d\mathbf{r}, \tag{3.3}$$

where C_2 is the two-point correlation function, discussed below. The excess free energy due to three body interactions, $F_{ex,3}$, takes the form

$$F_{ex,3} = -\frac{1}{3}\int n(\mathbf{r})\int C_3(\mathbf{r}-\mathbf{r}', \mathbf{r}-\mathbf{r}'')n(\mathbf{r}')n(\mathbf{r}'')\,d\mathbf{r}'d\mathbf{r}''d\mathbf{r}, \tag{3.4}$$

where C_3 is the three-point correlation function, also discussed below. The forms of $F_{ex,2}$ and $F_{ex,3}$ are translationally invariant.

3.2 Two-point correlations

The two-point correlation function, C_2, is defined using a simple repulsive term. The lattice spacing will be determined by the length scale in this term. We first define the circ function as a circular step function of unit radius and height, centred on the origin:

$$\mathrm{circ}(r) = \begin{cases} 1 & : r \leq 1, \\ 0 & : r > 1. \end{cases} \tag{3.5}$$

Using this function we define C_2 in two dimensions according to

$$C_2(\mathbf{r}) = -\frac{R}{\pi r_0^2}\mathrm{circ}\left(\frac{r}{r_0}\right), \tag{3.6}$$

where R sets the strength of the repulsion and r_0 sets the cutoff length scale. The normalization factor, πr_0^2, has been set such that the total integral of C_2 is $-R$. Figure 3.1 depicts $C_2(r)$ schematically.

It is a straightforward matter to add an additional positive XFPC type Gaussian peak at some $r > r_0$ to further influence interactions at the two-point correlation level as well. That will not be studied in this work, and only the repulsive part of the two-point correlation function will be retained for simplicity.

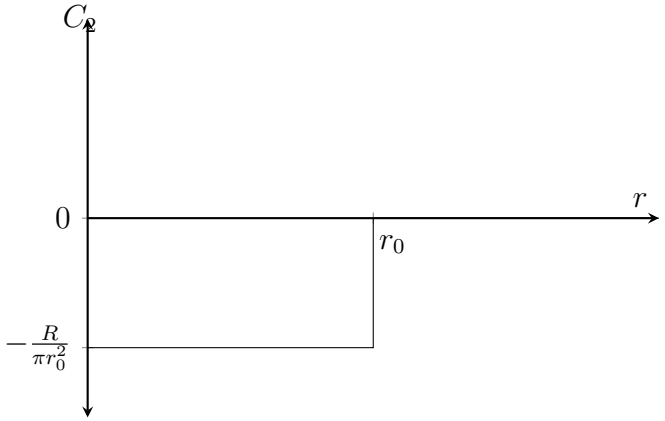


Figure 3.1: Two-point correlation function in real space.

It is convenient to have the Fourier space representation of C_2. The Fourier transform of Eq. 3.6 is given by

$$\hat{C}_2(\mathbf{k}) = -2R\frac{J_1(r_0 k)}{r_0 k}, \tag{3.7}$$

where J_m is the mth Bessel function of the first kind. Figure 3.2 shows a plot of $\hat{C}_2(k)$. Using the Fourier space representation of C_2 we can use the convolution theorem to write $F_{ex,2}$ as

$$F_{ex,2} = -\frac{1}{2}\int n(\mathbf{r})\mathcal{F}^{-1}\left\{\hat{C}_2(\mathbf{k})\hat{n}(\mathbf{k})\right\} d\mathbf{r}. \tag{3.8}$$

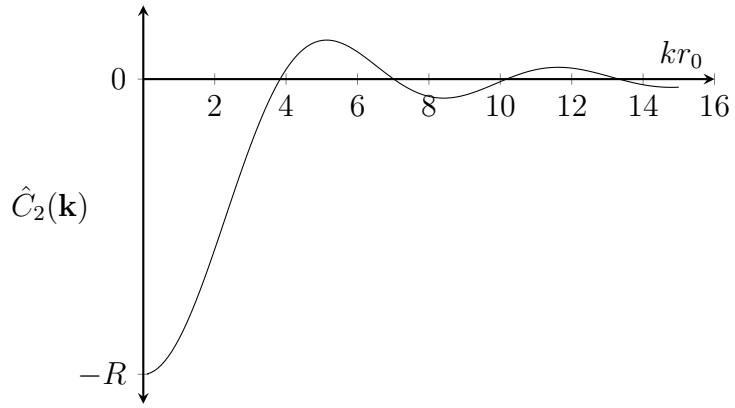


Figure 3.2: Plot of $-2RJ_1(r_0k)/(r_0k)$ in units of k/r_0.

While there is no attraction between the atoms in the two-point correlation function, the system can still undergo a phase transition and solidify at high enough density. It can be seen from the Fourier space representation of C_2 that there are some wavelengths (the peaks of $\hat{C}_2(\mathbf{k})$) which are energetically favoured. The relationship between the lattice constant, a_0, and the cutoff length, r_0, is non-trivial. In a one-mode approximation the energy of the system will be minimized when the first reciprocal space mode (K_1) of the density lies on the peak of $\hat{C}_2(\mathbf{k})$ (Eq. 3.7). This occurs at $K_1 \approx 5.13562/r_0$. Table 3.1 gives the ratio r_0/a_0 for various two dimensional lattices.

The two-point correlation function is completely isotropic and favours only

Lattice	$K_1 a_0$	r_0/a_0
Triangular	$4\pi/\sqrt{3}$	$0.707854\ldots$
Square	2π	$0.81736\ldots$
Graphene	$4\pi/3$	$1.22604\ldots$

Table 3.1: The ratio of r_0 to a_0 for various two dimensional crystal lattices. K_1 is the reciprocal lattice vector of the first mode of a crystal structure.

one equilibrium distance. It therefore strongly favours a lattice with the highest packing fraction. In two dimensions this is the triangular lattice, consistent with the classical result of hard sphere theory[97]. In order to stabilize more complex solid phases we must include either additional length scales, as in other XPFC models[69], or break the isotropy of the interactions. However, since we require the free energy to be rotationally invariant, the isotropy can only be broken relative to some local density configuration. This is not possible with two-point correlations—in order to accomplish this we have to move to higher order three-point correlation functions. Doing so will permit us to energetically favour particular relative angles between nearest neighbour atoms in order to produce a greater variety of crystal structures, particularly those of non-metals.

3.3 Three-point correlations

Three-point correlations in the model are included through the excess energy of Eq. 3.4. This term is computationally expensive relative to the two-point excess term of Eq. 3.3. The convolution theorem allows the two-point correlation to be computed by transforming to reciprocal space and multiplying point-wise. The computational complexity of this approach is $O(N \log N)$, the computational

complexity of the fast Fourier transform (N is the total number of grid points in the system). To our knowledge there is not such reduction in complexity available for Eq. 3.4. It therefore exhibits a prohibitive $O(N^3)$ computational complexity.

However we can remedy this problem by restricting C_3 to those functions that can be separated in the following manner:

$$C_3(\mathbf{r}-\mathbf{r}',\mathbf{r}-\mathbf{r}'') = \sum_i C_s^{(i)}(\mathbf{r}-\mathbf{r}')C_s^{(i)}(\mathbf{r}-\mathbf{r}''), \tag{3.9}$$

where the $C_s^{(i)}$ will be defined below. While this limits the possible forms C_3 may take, it remains flexible enough to produce a wide variety of crystal structures, including graphene and those previously modelled by XPFC and similar 2D models. When we insert Eq. 3.9 into Eq. 3.4, it reduces the three-point term to

$$F_{ex,3} = -\frac{1}{3}\int n(\mathbf{r}) \sum_i \left(\int C_s^{(i)}(\mathbf{r}-\mathbf{r}')n(\mathbf{r}')\,d\mathbf{r}' \right)^2 d\mathbf{r}. \tag{3.10}$$

How to approach this term computationally will be discussed below. However it should be apparent that reducing the double integration over $\mathbf{r}'$ and $\mathbf{r}''$ to just a single integration over $\mathbf{r}'$ considerably reduces the computational complexity of the three-point term.

We now define the $C_s^{(i)}$ functions. We work in polar coordinates and separate $\mathbf{r}$ into r and θ. We define $C_s^{(i)}$ as

$$C_s^{(1)}(r,\theta) = C_r(r)C_\theta^{(1)}(\theta) = C_r(r)\cos(m\theta); \tag{3.11}$$

$$C_s^{(2)}(r,\theta) = C_r(r)C_\theta^{(2)}(\theta) = C_r(r)\sin(m\theta); \tag{3.12}$$

$$C_r(r) = \frac{X}{2\pi a_0}\delta(r-a_0). \tag{3.13}$$

X defines the strength of the interaction, a_0 corresponds to the lattice spacing, and m defines bond order (discussed below) of the crystal phase.

Despite the angular factors present in Eqs. 3.11-3.12, the total excess free energy remains rotationally invariant. To demonstrate this, we begin with the sum in Eq. 3.9 and expand the square to get

$$\sum_i \left(\int C_s^{(i)}(\mathbf{r}-\mathbf{r}') C_s^{(i)}(\mathbf{r}-\mathbf{r}'') n(\mathbf{r}') n(\mathbf{r}'')\, d\mathbf{r}'\, d\mathbf{r}'' \right). \tag{3.14}$$

Defining $\mathbf{r_1} \equiv \mathbf{r}-\mathbf{r}'$, $\mathbf{r_2} \equiv \mathbf{r}-\mathbf{r}''$ and pulling the sum inside the integral gives

$$\int \left(\sum_i C_s^{(i)}(\mathbf{r_1}) C_s^{(i)}(\mathbf{r_2}) \right) n(\mathbf{r}-\mathbf{r_1}) n(\mathbf{r}-\mathbf{r_2})\, d\mathbf{r_1}\, d\mathbf{r_2}. \tag{3.15}$$

Considering the sum in the brackets and changing to polar coordinates, $\mathbf{r_i} \rightarrow (r_i, \theta_i)$, gives

$$\sum_i C_s^{(i)}(\mathbf{r_1}) C_s^{(i)}(\mathbf{r_2}) = \sum_i C_s^{(i)}(r_1,\theta_1) C_s^{(i)}(r_2,\theta_2) \tag{3.16}$$

$$= C_r(r_1) C_r(r_2) \sum_i C_\theta^{(i)}(\theta_1) C_\theta^{(i)}(\theta_2). \tag{3.17}$$

Next we insert the $C_\theta^{(i)}$ from Eqs. (3.11) and (3.12) to find that

$$\sum_i C_s^{(i)}(\mathbf{r_1}) C_s^{(i)}(\mathbf{r_2}) = C_r(r_1) C_r(r_2) \Big\{ \cos(m\theta_1)\cos(m\theta_2) + \sin(m\theta_1)\sin(m\theta_2) \Big\}. \tag{3.18}$$

And finally, applying the identity

$$\cos(\theta_1)\cos(\theta_2) + \sin(\theta_1)\sin(\theta_2) = \cos(\theta_2 - \theta_1) \tag{3.19}$$

gives us

$$\sum_i C_s^{(i)}(\mathbf{r_1}) C_s^{(i)}(\mathbf{r_2}) = C_r(r_1) C_r(r_2) \cos\big(m(\theta_2 - \theta_1)\big). \tag{3.20}$$

Since Eq. 3.20 depends only on the difference $\theta_2 - \theta_1$, the free energy remains rotationally invariant. By selecting different values for m we can favour certain crystal structures over others. $m = 6$ favours the six-fold triangular lattice, $m = 4$ the four-fold square lattice, and $m = 3$ favours three-fold graphene crystals.

It is noted that an analogous mathematical approach as the one above can be used in 3D by replacing the $C_s^i(r, \theta)$ by spherical harmonics $Y_{lm}(r, \theta)$ and replacing the sum over i in Eq. (3.17) by a sum over m (see Appendix E). However, this will not be pursued here further in order to keep the scope of this thesis tractable and limited to three-point interactions in two dimensional systems. We have communicated this result to colleagues with whom we are collaborating and who are presently pursuing this approach for 3D materials[98].

We will later have need for the Fourier transforms of $C_s^{(i)}(r, \theta)$. For $C_s^{(1)}(r, \theta)$ we begin by taking the multipole expansion

$$C_s^{(1)}(r, \theta) = \frac{X}{2\pi a_0} \delta(r - a_0) \frac{e^{im\theta} + e^{-im\theta}}{2}, \tag{3.21}$$

which transforms as (see Appendix B):

$$\hat{C}_s^{(1)}(k,\theta_k) = X\frac{i^m e^{im\theta_k} J_m(ka_0) + i^{-m} e^{-im\theta_k} J_{-m}(ka_0)}{2} \tag{3.22}$$

$$= Xi^m \frac{e^{im\theta_k} + e^{-im\theta_k}}{2} J_m(ka_0) \tag{3.23}$$

$$= Xi^m \cos(m\theta_k) J_m(ka_0), \tag{3.24}$$

where (k, θ_k) are the Fourier space polar coordinates. In Eq. (3.23) we have used the fact that $J_{-m}(r) = (-1)^m J_m(r)$. Applying the same proceedure for $C_s^{(2)}(r,\theta)$ gives

$$\hat{C}_s^{(2)}(k,\theta_k) = Xi^m \sin(m\theta_k) J_m(ka_0). \tag{3.25}$$

3.4 Equilibrium Phase diagrams for different crystal structures

Using the approach outlined in Section 2.3.1, we can produce phase diagrams for various crystal structures stabilized by the two-point and three-point correlation function terms in our model. Here we will present phase diagrams for triangular, square and honeycomb (graphene) phases coexisting with a disordered phase. The mathematical procedure of constructing the phase diagram will not be shown here as again as it was already shown in Chapter 2. The interested reader can contact the author for access to the Mathematical files used to generate the phase diagrams that follow.

3.4.1 Triangular crystals

The triangular phase is notable because we can produce it even with no three-point correlations. The rejection term R in the two-point correlation function is sufficient to produce a triangular crystal phase. Figure 3.3 shows an order-disorder phase diagram in $\{\phi_0, R\}$ space with $X = 0$, i.e. without the three-point interaction term. Note that we have plotted R^{-1} on the vertical axis, since decreasing R corresponds to increasing temperature in this model. This phase diagram was generated with a five mode amplitude expansion for the density.

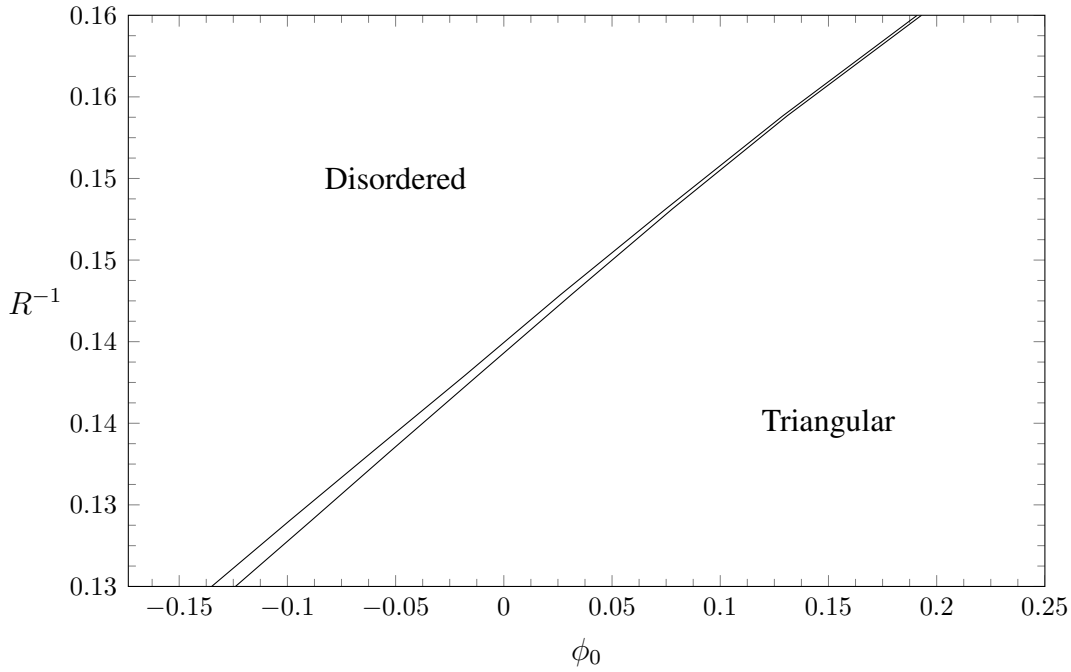


Figure 3.3: Triangular-disorder phase diagram using only two-point correlations ($X = 0$).

We can also generate a phase diagram for disordered-triangular phase coexis-

tence by including three-point correlations. Using $m = 6$ for six-fold symmetry and fixing $R = 6$, we obtain the phase diagram in $\{\phi_0, X\}$ space shown in Fig. 3.4. It is noted that the use of the three-point term gives rise to an expanded region of

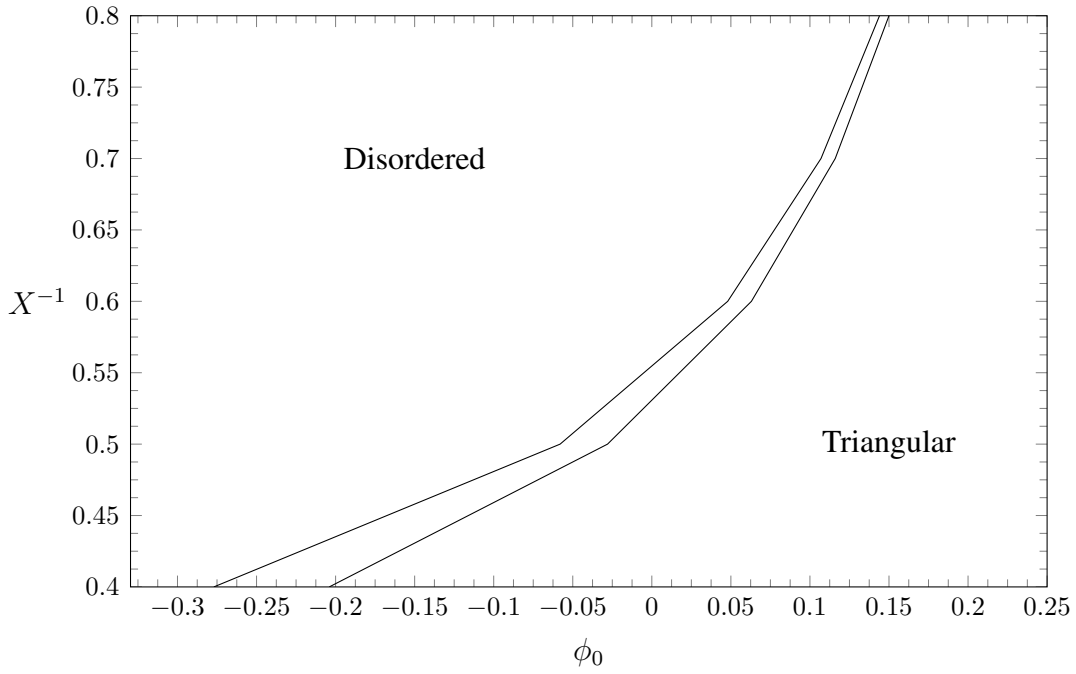


Figure 3.4: Triangular-disorder phase diagram using two and three-point correlations, with $m = 6$ and $R = 6$. Here $r_0/a_0 = 0.70785$.

coexistence between the disordered and triangular phases.

As was mentioned earlier, it is also possible to modify the correlations in our model by adding a Guassian peak to the two-point correlation function, with its peak position at $r > r_0$. In addition, we could also replace the delta function in Eq. 3.13 by a similar Gaussian peak. We do not expect that these replacements would alter the behaviour of the phase diagrams in Figures 3.3 and 3.4, although the resulting forms are a little mode cumbersome to deal with algebraically. How-

ever, the added flexibility of these interaction terms would likely make it possible to further expand the coexistence range of the phase diagram.

3.4.2 Square crystals

For four-fold symmetry we must include the three-point correlations into the free energy, with $m = 4$. Using a five mode amplitude expansion and setting $R = 5$, we obtain the square-disorder phase diagram depicted in Fig. 3.5.

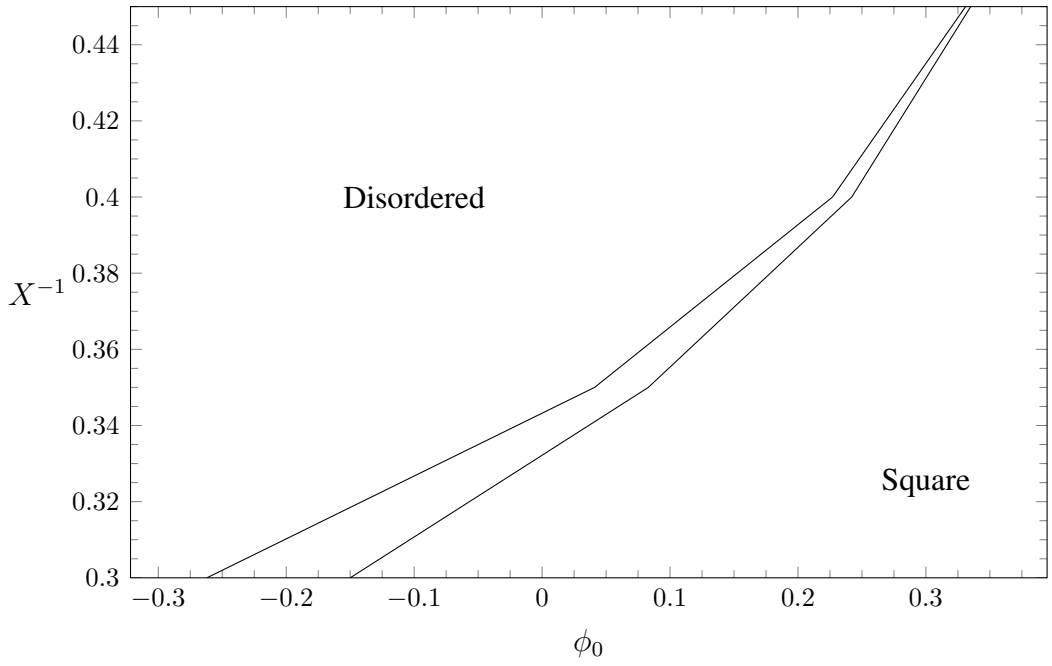


Figure 3.5: Square-disorder phase diagram using two and three-point correlations with $m = 4$, $R = 5$ and $r_0/a_0 = 0.81736$.

3.4.3 Graphene crystals

Graphene crystals can be described using a triangular Bravais lattice with a two atom basis. As a result, graphene crystals differ from triangular crystals only in their structure factors. Starting with the primitive vectors of the triangular lattice (in terms of the coordinate vectors $\mathbf{x}$ and $\mathbf{y}$):

$$\mathbf{a_0} = \sqrt{3}\mathbf{x}, \qquad \mathbf{a_1} = \frac{\sqrt{3}}{2}\mathbf{x} + \frac{3}{2}\mathbf{y}, \tag{3.26}$$

we locate the basis atoms at

$$\mathbf{d}_0 = -\frac{\sqrt{3}}{2}\mathbf{x} - \frac{1}{2}\mathbf{y}, \qquad \mathbf{d}_1 = -\frac{\sqrt{3}}{2}\mathbf{x} + \frac{1}{2}\mathbf{y}. \tag{3.27}$$

Now, if we have a density expansion for a triangular lattice n_{T} then we can write the expansion of a graphene lattice by translating n_{T} to the positions of the basis atoms. That is

$$n_{\mathrm{G}}(\mathbf{r}) = \sum_i n_{\mathrm{T}}(\mathbf{r} - \mathbf{d}_i). \tag{3.28}$$

If n_{T} was expanded as in Eq. 2.19, then n_{G} becomes

$$n_{\mathrm{G}}(\mathbf{r}) = \sum_{k=0}^{N} \phi_k^T \left(\sum_j S_{k,j} e^{i\mathbf{q}_{k,j}\cdot\mathbf{r}} \right); \tag{3.29}$$

$$S_{k,j} = \sum_i e^{-i\mathbf{q}_{k,j}\cdot\mathbf{d_i}}; \tag{3.30}$$

where $S_{k,j}$ are the graphene structure factors.

The free energy is invariant under a translation of the basis atoms (that is

$\mathbf{d_i} \to \mathbf{d_i} + \mathbf{t}$). In selecting the positions of the basis atoms, we have taken advantage of this freedom: the basis atoms have been positioned such that all structure factors are real and equal for a given mode. That is, $S_{k,j} = S_k$. Noting this, we can pull the structure factors out of the second sum:

$$n_{\mathrm{G}}(\mathbf{r}) = \sum_{k=0}^{N} S_k \phi_k^T \left(\sum_j e^{i\mathbf{q}_{k,j}\cdot\mathbf{r}} \right) \tag{3.31}$$

$$= \sum_{k=0}^{N} \phi_k^G \left(\sum_j e^{i\mathbf{q}_{k,j}\cdot\mathbf{r}} \right); \tag{3.32}$$

$$\phi_k^G = S_k \phi_k^T. \tag{3.33}$$

Thus, although both triangular and graphene phases are described by the same underlying lattice, the difference will be detectable in their amplitudes. When we compute the first four structure factors we find that

$$S_0 = 2; \quad S_1 = -1; \quad S_2 = 2; \quad S_3 = -1. \tag{3.34}$$

In contrast with the triangular phase, where all structure factors are equal to unity, we can see that some of the structure factors for graphene are negative. The distinguishing feature of graphene amplitude expansions is therefore that certain amplitudes are negative. The crudest test to distinguish the phases is to simply examine the first amplitude of thr density expansion: if $\phi_1 < 0$ we have graphene, while if $\phi_1 > 0$ we have the triangular phase.

These negative amplitudes provide some insight into the difficulty of producing graphene phases using two-point correlations alone. If we insert the amplitude expansion of a triangular lattice into the two-point excess free energy (Eq. 3.3) and

integrate over a unit cell we find that

$$F_{ex,2} = -\frac{1}{2}\left(\hat{C}_2(q_0)\phi_0^2 + 6\hat{C}_2(q_1)\phi_1^2 + 6\hat{C}_2(q_2)\phi_2^2 + \ldots\right), \tag{3.35}$$

where q_i are the lengths of the reciprocal space modes (recall that all vectors in a mode are of the same magnitude). Note that since all amplitudes ϕ_k are squared, the excess free energy due to two-point correlations is symmetrical with respect to the signs of ϕ_k. No matter what the function $\hat{C}_2(q)$ looks like, it cannot energetically favour negative amplitudes over positive amplitudes. The ideal free energy of Eq. 3.2 is of no help either, as with the standard values $\eta = \xi = 1$ it favours positive amplitudes as well. [1] Three-point correlations don't suffer this limitation, since they produce terms of odd powers in ϕ_k.

To produce the phase diagram for the graphene system we use a five mode triangular density expansion with $m = 3$. Minimizing the free energy with respect to the amplitudes and applying the common tangent construction we obtain a graphene-disorder phase diagram in $\{\phi_0, X\}$ space. Figure 3.6 shows such a phase diagram for the case $R = 6$. Although not shown in the diagram, it was confirmed that the amplitudes followed the pattern established in Eq. 3.34, with the first mode negative and the second positive.

3.5 Poisson's ratio of the three-point XPFC model

In this section we calculate the elastic constant of graphene crystals formed with our model at various values of the temperature parameter X^{-1}. The approach we

[1] It is noted that if we change the sign of η it is possible to generate graphene structure. However, it has been found that the elastic response of this type of graphene is not correct and also the density of the ordered phases becomes lower than that of the disordered phase [99].

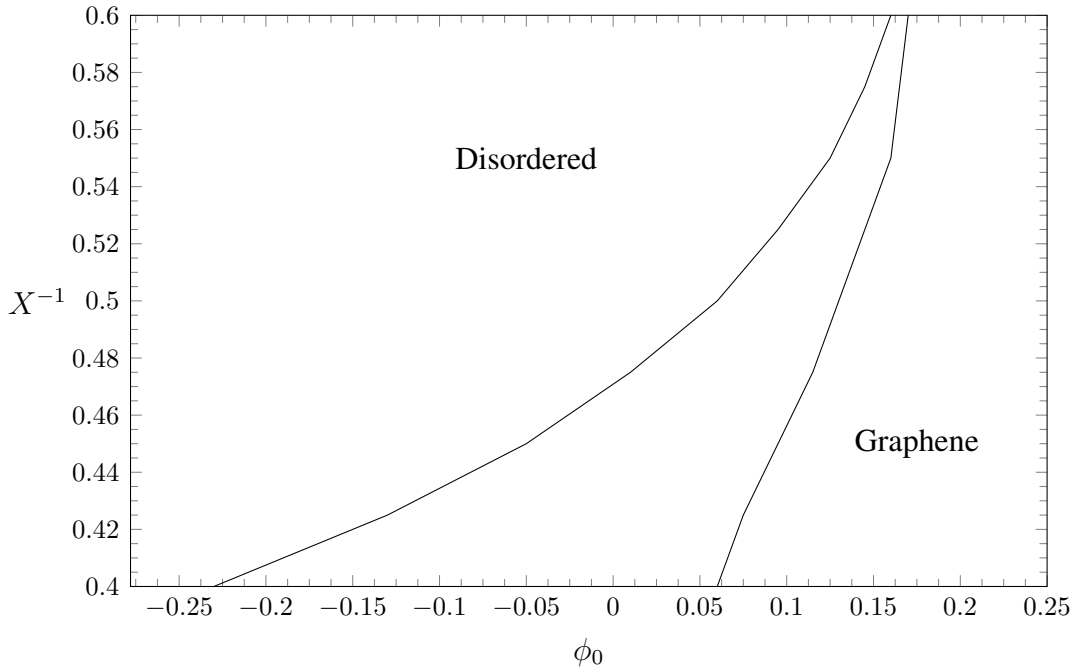


Figure 3.6: Graphene-disorder coexistence phase diagram, with $m = 3$ and $R = 6$. Here $r_0/a_0 = 1.2259$.

take here is numerical, wherein we examine the variations of the free energy of our system and from this extract the corresponding elastic constants. The free energy is approximated by numerical minimization of a finite mode amplitude expansion.

To proceed, we write the elastic energy of a strained system as

$$F = \frac{1}{2} \sum C_{ij} \epsilon_i \epsilon_j + F_0, \tag{3.36}$$

where C_{ij} is the elasticity tensor, ϵ_i are the strains (in Voigt notation) and F_0 is the unstrained free energy. In terms of the derivatives of the displacement vectors $\mathbf{u}$, $\epsilon_1 = u_{xx} = \partial u_x/\partial x$, $\epsilon_2 = u_{yy} = \partial u_y/\partial y$, etc. For the simplest anisotropic

material is it can be shown that $C_{11} = C_{22}$ [100]. This simplifies the elastic energy in Eq. 3.36 to

$$F = \frac{1}{2}C_{11}(u_{xx}^2 + u_{yy}^2) + C_{12}u_{xx}u_{yy} + F_0. \tag{3.37}$$

We next let $u_{xx} = u\cos(t)$ and $u_{yy} = u\sin(t)$. Noting that $\cos(t)\sin(t) = \sin(2t)/2$, we have

$$F = \frac{1}{2}C_{11}u^2 + \frac{1}{2}C_{12}u^2\sin(2t) + F_0. \tag{3.38}$$

The free energy of such a strained system can be measured either through amplitude expansions or through simulations. As mentioned above, we take the former route. If we measure the free energy as a function of t from $t = 0$ to $t = 2\pi$, we can compute the Fourier series of this function. The values of this series can be used to derive the parameters C_{11} and C_{12} appearing in Eq. 3.38. The constant term gives us $F = \frac{1}{2}C_{11}u^2 + F_0$ and can be used to find C_{11}. The third term gives us the $C_{12}u^2\sin(2t)$ term, which can be used to find C_{12}.

We can proceed to compute the bulk modulus, shear modulus and Poison ratio of the graphene generated by our model. These are given explicitly in term of the

elasticity constants of our system as[53]:

$$C_{11} = C_{12} + 2C_{44}, \tag{3.39}$$

$$B = \frac{C_{11} + C_{12}}{2}, \tag{3.40}$$

$$\mu = C_{44}, \tag{3.41}$$

$$Y = \frac{4B\mu}{B + \mu}, \tag{3.42}$$

$$\tag{3.43}$$

and

$$\nu = \frac{B - \mu}{B + \mu}, \tag{3.44}$$

where B is the bulk modulus, μ is the shear modulus, Y is Young's modulus and ν is Poisson's ratio, respectively. From this we can see that

$$\nu = \frac{C_{12}}{C_{11}}. \tag{3.45}$$

Thus, Poisson's ratio for our graphene in our model can be computed from C_{11} and C_{12}. Table 3.2 shows the values obtained for various system parameters using an eight mode amplitude expansion. It is seen that the value of the elastic moduli depends on the system parameters used. Specifically, the Poisson ratio becomes positive for the temperature parameter $X^{-1} \geq 0.5$ in our model, which indicates an upper limit of reliability of X^{-1}, at least with the tested values of the parameters R in the two-point correlation function. In a recent study that compared this and other PFC type models of graphene, the present one gave a range of parameters

System parameters		Elastic coefficients			Bulk moduli		
n_0	X^{-1}	C_{11}	C_{12}	C_{44}	B	μ	ν
0.2	0.4	43.5673	−1.00583	22.2866	21.2807	22.2866	−0.0230869
0.2	0.5	7.6481	0.838848	3.40462	4.24347	3.40462	0.109681
0.3	0.4	49.7526	−1.4694	25.611	24.1416	25.611	−0.0295341
0.3	0.5	10.6984	1.01357	4.84243	5.85599	4.84243	0.0947399

Table 3.2: Measurement of elastic coefficients for various system parameters for the three-point graphene system. Results obtained using an eight mode amplitude expansion. $R = 6$ and $u = 0.01$ for all entries.

that features a negative range of Poisson ratios[101]. It would be instructive to map out the elastic constants in a more comprehensive way as functions of R, X^{-1} and n_0 for our system. The model parameters would have to be matched to graphene properties more quantitatively before a comparison could be made of the elastic constants to experiments.

3.6 Dynamics and simulations

The density field $n(\mathbf{r})$ is a conserved order parameter. It therefore follows model B dynamics for conserved fields[7]. This implies that

$$\frac{\partial n}{\partial t} = M_n \nabla^2 \left(\frac{\delta F}{\delta n} \right), \tag{3.46}$$

where M_n is a mobility parameter that sets the diffusion rate, and $\delta F/\delta n$ is the functional derivative of F with respect to n. Realistic simulations of n would also include a noise source in Eq. 3.46, however for simplicity these will be ignored in the present work.

In order to simulate the evolution of n we will require the functional deriva-

tives of F. We derive the variational of our free energy term by term in what follows. The most straightforward is the functional derivative of F_{id} (Eq. 3.2). Since F_{id} is an integral of a simple function of n, we need only take the derivative of that function:

$$\frac{\delta F_{id}}{\delta n} = n - \eta\frac{n^2}{2} + \chi\frac{n^3}{3}. \tag{3.47}$$

The term $F_{ex,2}$ (Eq. 3.3) is a convolution integral, the variational of which is well known[53] and given by:

$$\frac{\delta F_{ex,2}}{\delta n} = -\int C_2(\mathbf{r}-\mathbf{r}')n(\mathbf{r}')\,d\mathbf{r}' \equiv -C_2 * n, \tag{3.48}$$

where $*$ indicates convolution. In general, terms such as this can be computed efficiently using the convolution theorem.

The three-point term (Eq. 3.10) is more complex. We begin by expanding the square for each term in the summation:

$$F^{(i)}_{ex,3}[n] \;=\; -\frac{1}{3}\int C^{(i)}_s(\mathbf{r}-\mathbf{r}')C^{(i)}_s(\mathbf{r}-\mathbf{r}'')n(\mathbf{r})n(\mathbf{r}')n(\mathbf{r}'')\,d\mathbf{r}'\,d\mathbf{r}''\,d\mathbf{r}. \tag{3.49}$$

To find the functional derivative we next make use of the well known formula [53]

$$F[n+\delta n] - F[n] \equiv \int \delta n \frac{\delta F}{\delta n}\,d\mathbf{r}. \tag{3.50}$$

Applying this to Eq. (3.10) and discarding terms of order $\mathcal{O}\big((\delta n)^2\big)$ we are left

with three terms,

$$
\begin{aligned}
&F^{(i)}_{ex,3}[n+\delta n] - F^{(i)}_{ex,3}[n] = \\
&-\frac{1}{3}\int C^{(i)}_s(\mathbf{r}-\mathbf{r}')C^{(i)}_s(\mathbf{r}-\mathbf{r}'')\delta n(\mathbf{r})n(\mathbf{r}')n(\mathbf{r}'')\,d\mathbf{r}'\,d\mathbf{r}''\,d\mathbf{r} \\
&-\frac{1}{3}\int C^{(i)}_s(\mathbf{r}-\mathbf{r}')C^{(i)}_s(\mathbf{r}-\mathbf{r}'')n(\mathbf{r})\delta n(\mathbf{r}')n(\mathbf{r}'')\,d\mathbf{r}'\,d\mathbf{r}''\,d\mathbf{r} \\
&-\frac{1}{3}\int C^{(i)}_s(\mathbf{r}-\mathbf{r}')C^{(i)}_s(\mathbf{r}-\mathbf{r}'')n(\mathbf{r})n(\mathbf{r}')\delta n(\mathbf{r}'')\,d\mathbf{r}'\,d\mathbf{r}''\,d\mathbf{r}. \qquad (3.51)
\end{aligned}
$$

Swapping $\mathbf{r}$ and $\mathbf{r}'$ in the second term and $\mathbf{r}$ and $\mathbf{r}''$ in the third gives:

$$
\begin{aligned}
&F^{(i)}_{ex,3}[n+\delta n] - F^{(i)}_{ex,3}[n] = \\
&\quad -\frac{1}{3}\int \delta n(\mathbf{r}) \int \Big(C^{(i)}_s(\mathbf{r}-\mathbf{r}')C^{(i)}_s(\mathbf{r}-\mathbf{r}'')n(\mathbf{r}')n(\mathbf{r}'') \\
&\quad + C^{(i)}_s(\mathbf{r}'-\mathbf{r})C^{(i)}_s(\mathbf{r}'-\mathbf{r}'')n(\mathbf{r}')n(\mathbf{r}'') \\
&\quad + C^{(i)}_s(\mathbf{r}''-\mathbf{r}')C^{(i)}_s(\mathbf{r}''-\mathbf{r})n(\mathbf{r}')n(\mathbf{r}'')\Big)\,d\mathbf{r}'\,d\mathbf{r}''\,d\mathbf{r}. \qquad (3.52)
\end{aligned}
$$

Combining the last two terms (by swapping $\mathbf{r}'$ and $\mathbf{r}''$ in the third term) gives:

$$
\begin{aligned}
&F^{(i)}_{ex,3}[n+\delta n] - F^{(i)}_{ex,3}[n] = \\
&\quad -\frac{1}{3}\int \delta n(\mathbf{r}) \int \Big(C^{(i)}_s(\mathbf{r}-\mathbf{r}')C^{(i)}_s(\mathbf{r}-\mathbf{r}'')n(\mathbf{r}')n(\mathbf{r}'') \\
&\quad + 2C^{(i)}_s(\mathbf{r}'-\mathbf{r})C^{(i)}_s(\mathbf{r}'-\mathbf{r}'')n(\mathbf{r}')n(\mathbf{r}'')\Big)\,d\mathbf{r}'\,d\mathbf{r}''\,d\mathbf{r}. \qquad (3.53)
\end{aligned}
$$

Finally, noting from Eqs. (3.11) and (3.12) that $C^{(i)}_s(-\mathbf{r}) = (-1)^m C^{(i)}_s(\mathbf{r})$ we find

$$
\frac{\delta F^{(i)}_{ex,3}}{\delta n} = -\frac{1}{3}\Big(\big[C^{(i)}_s * n\big]^2 + 2(-1)^m C^{(i)}_s * \big[n\cdot(C^{(i)}_s * n)\big]\Big). \qquad (3.54)
$$

Having expressed the functional derivative in terms of convolutions we can repeatedly apply the convolution theorem in order to compute the result efficiently. The first term can be computed by performing the convolution in reciprocal space and then returning to real space in order to calculate the square. The second term can be computed in three steps: computing the inner convolution in reciprocal space; returning to real space for the multiplication by n; and finally transforming to reciprocal space once again to compute the outer convolution.

3.6.1 Polycrystalline 2D materials, defects and coexistence

In this section, we demonstrate our model by simulating the evolution of triangular, square and graphene phases by selecting model parameters and average densities corresponding to the ordered regions of their respective phase diagrams studied in this chapter. The density field is initialized with Gaussian noise on a two dimensional system of size $N \times M$ with a numerical grid spacing of $\Delta x = 0.1$ and time steps of $\Delta t = 0.0001$, both of which were small enough to assure stability and accuracy of the following results presented. From the initial conditions, the system subsequently solidifies into a polycrystalline solid. Figure 3.7 shows snapshots of the density field at early (left) and late (right) times, for triangular, square and graphene systems.

Of particular interest are the defect structures of the polycrystalline graphene phase. Close examination of Fig. 3.7f reveals that along the grain boundaries separating misaligned crystal grains we see rings of five (pentagons) and seven (heptagons) atoms. These are highlighted in Fig. 3.8a. These rings are frequently paired and referred to as 5-7 defects. The overall pattern of 5-7 defects along the grain boundary appears to be aperiodic in nature. These aperiodic, 5-7 defect

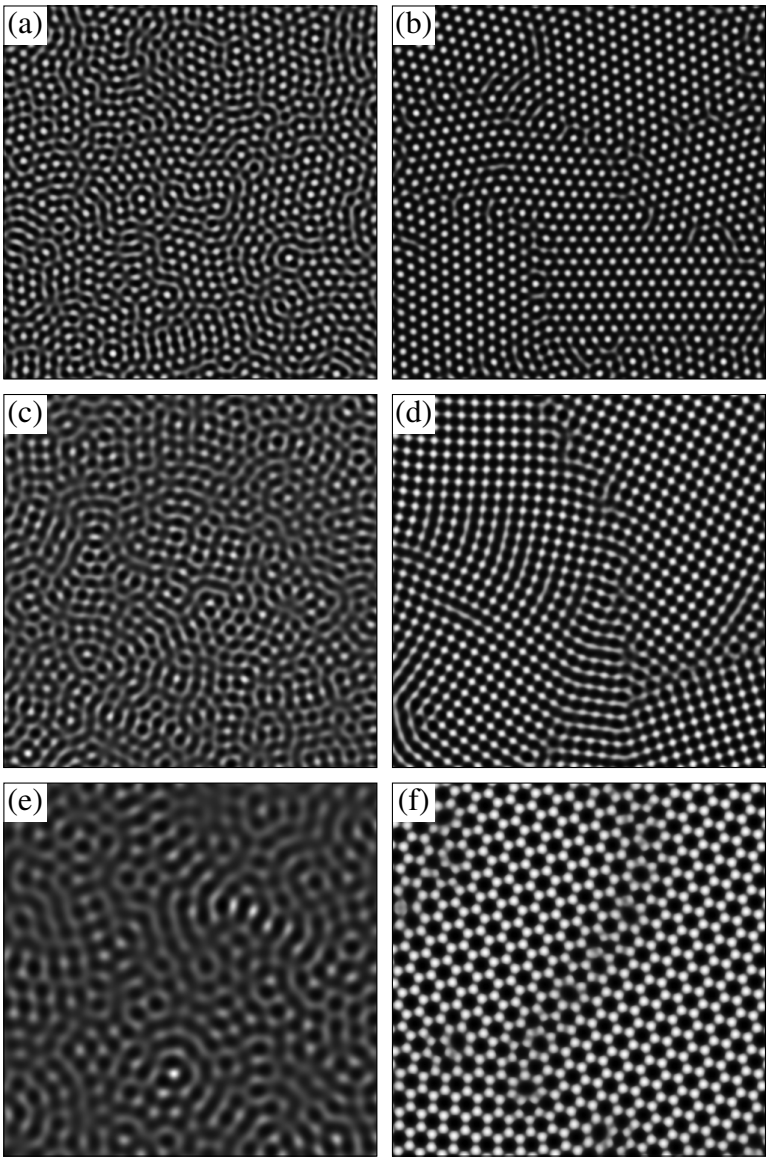


Figure 3.7: Density fields of triangular (*a* and *b*), square (*c* and *d*) and graphene (*e* and *f*) phase growth, showing early (left) and late (right) times during solidification. Systems are initialized with gaussian density fluctuations. For all three systems $\phi_0 = 0.3$ and values of r_0/a_0 match those in Table 3.1. For (*a* and *b*), $R = 7$ and $X = 0$. For (*c* and *d*), $R = 6$ and $X^{-1} = 0.5$. For (*e* and *f*), $R = 6$ and $X^{-1} = 0.4$.

grain boundaries are in excellent agreement with experimental results of polycrystalline graphene[102]. Figure 3.8b and c show grain boundaries in polycrystalline graphene samples grown experimentally by chemical vapour deposition (CVD). It is seen that both the simulation and experiment exhibit similar morphological features.

In order to demonstrate coexistence between the ordered and disordered phases of our graphene system, we simulate a 2000×100 system with periodic boundary conditions, $\Delta x = 0.10392$ (to match the lattice to the box), and the same numerical time step as used to generate the data in Fig. 3.7. A large initial seed is placed extending through the system transverse to the long dimension. This configuration ensures that the order-disorder interface is a straight line, eliminating curvature effects on the equilibrium densities. The initial average densities of the solid and disordered phases are selected according to the phase diagram (Fig. 3.6) and the system is allowed to come to equilibrium. Figure 3.9a shows the density field at equilibrium, while Fig. 3.9b shows the smoothed density profile through the interface. It can be seen that the resulting equilibrium bulk values agree well with those in Fig. 3.6.

As has been mentioned at above, and in the introduction of this thesis, other methods exist for stabilizing graphene symmetries using the PFC methodology, but each with its own shortcoming. The methods Makhonta *et al.* [48, 49] do not generate appropriate phase coexistence between graphene and a disordered state, which is important to simulate a CVD process. Another method is that wherein one inverts the cubic term in the ideal free energy, first applied to the standard PFC in [101] and by us in the single-peak XPFC [99, 103]. These so-called "inverted triangle models" have the deficiency that they place the solid (graphene) phase

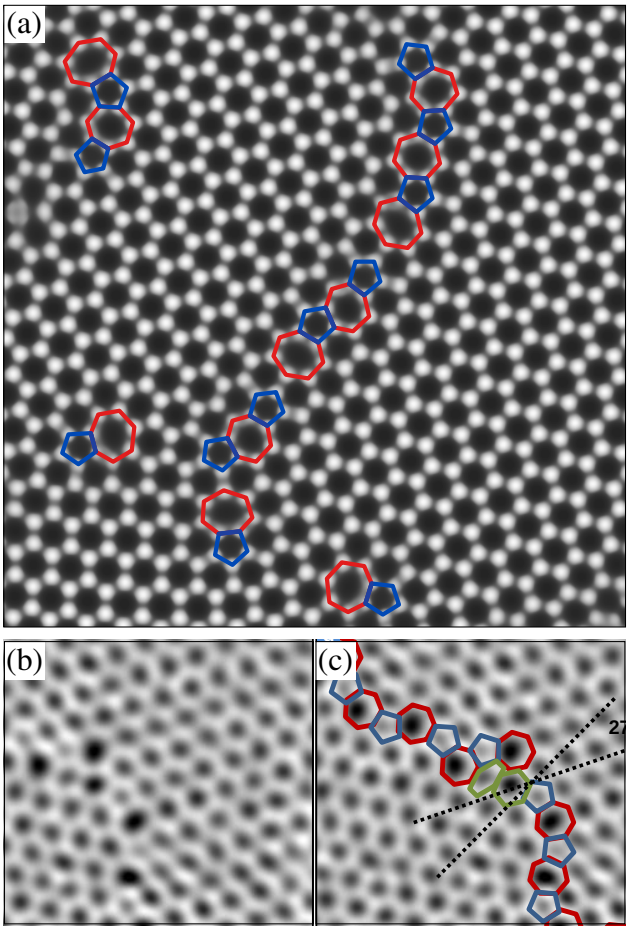


Figure 3.8: Comparison of simulated and experimentally determined defect structures of polycrystalline graphene. The defect structure of Figure 3.7(*f*) is highlighted in (*a*). The grain boundary is resolved by a line of 5-7 defect structures. These defect structures match those found experimentally in polycrystalline graphene membranes grown by chemical vapour deposition (CVD) [102]. (*b*) shows an atomic resolution transmission electron microscope (TEM) image of one such graphene membrane; the defect structure is highlighted in (*c*). (*b*) and (*c*) reprinted by permission from Macmillan Publishers Ltd: Nature [102], copyright 2011.

below that of the disordered phase, and they also give the wrong Poisson ratio for graphene. An amplitude model of single-component graphene was also developed in [101]. It under-predicts the grain boundary energy (compared to MD simulations presented in [101]), and it fails to capture defect structure and morphology of graphene grain boundaries. An interesting future project emerging from the study of this chapter is to develop a complex amplitude model of graphene from the present model. It would be instructive to determine if this model, which would have contributions from the three-point and two-point correlation terms, would fare any better than the amplitude model developed in [101].

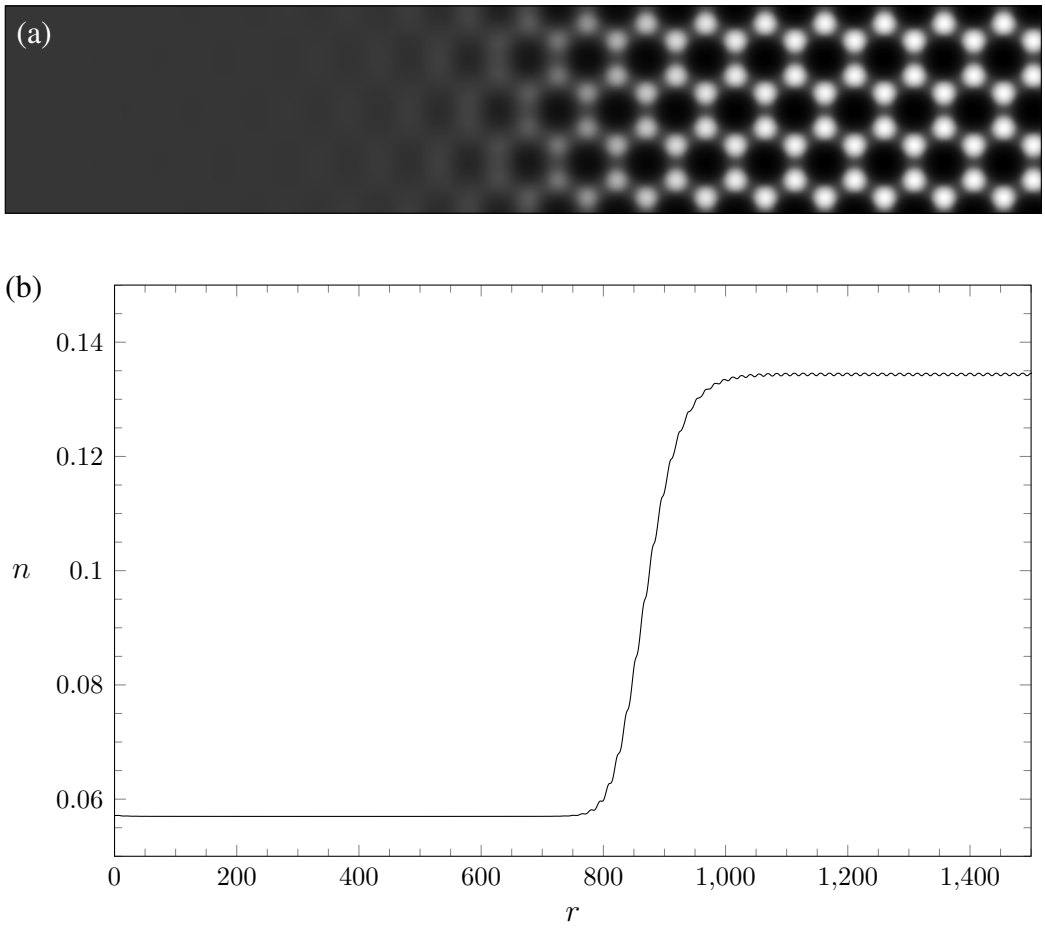


Figure 3.9: Simulation of coexistence between the ordered and disordered phases of graphene. Density field $n(\mathbf{r})$ of the equilibrium interface between phases shown in (*a*). Smoothed average density along the longitudinal axis depicted in (*b*). Here, $X^{-1} = 0.5$, $R = 6$. Average densities of 0.057 and 0.134 in the disordered and ordered phases respectively match closely the theoretical values from the phase diagram in Figure 3.6.

Chapter 4

XPFC Modelling of Binary Systems

This chapter is concerned with modelling multicomponent—specifically binary—systems. Traditionally in XPFC, binary systems have been modelled in the density-concentration space, where the density n represents the total density of all components (relative to a reference total density), and the concentration c represents the fraction of one component, ranging from 0 to 1 [35, 43]. In this chapter we break with this tradition and construct the free energy of the system on the basis of the individual densities n_A, n_B, etc. There are several reasons for proceeding in this way.

In the density-concentration description, the simplifying assumption is often made that the concentration varies on a longer length scale than the density[69, 43]. This works well for systems where the two (or more) species separate into their own crystal grains, but it precludes systems where atoms of differing species neighbour each other. Of lesser importance is also the fact that even in metal alloys where the former approximation holds, it can be shown that the assumption that c varies on a longer length scale than n is not strictly true[104]. The density-

concentration description also forces a single diffusion constant on each solute concentration. This effectively means that only solute (minority) species diffuse relative to a background lattice. By having each component represented by its own field, we can model different species' diffusions by setting different diffusion constants for each species.

Another reason for describing a binary system in terms of n_A and n_B is that for many binary systems the phase diagram is not known in the (n, c) variables, or indeed any system of variables. A case in point in graphene and hydrogen on a surface of a metal at certain pressure and temperature: it is not clear under what conditions these two elements mix to form a graphene, a disordered "surface state," or both. SnSe, popular for its applications in thermoelectric devices, is another two-dimensional material whose equilibrium phase diagram is not known. Essentially, when the thermodynamics of a mixture—or alloy—are unknown, building up a theory for such a material is probably most effectively done by considering the physics of the individual species and their mutual interactions.

In this chapter, we use the above approach to construct new XPFC models of several binary materials, and demonstrate some basic properties of each system. The first system is a "salt-like" model where the two species self-repel but are mutually attracted. This produces a solid with a square crystal lattice where the nearest neighbours are of opposite species. The second system is a disordered-solid model, where one species has no self-interaction term and behaves like an ideal gas, but is repelled from a second, solid-forming species. The third system has a solid forming species with a diffuse gas-like species that fills the interstitial sites of the solid forming species. In this model the interstitial species has a high diffusion constant compared to the solid-forming species. This could be adopted

into a model for steel, with carbon diffusing into the interstitial sites of iron. Or, including long-range screened interactions can lead to a PFC model for ionic fluids, where the two-component system represents the ionic cores (nuclei and their core electrons), and the valence electrons, which can play the role of a classical electron "gas," interacting with the ionic cores. Finally, we introduce a model of chemical vapour deposition of graphene and hydrogen. This model extends the model of the previous chapter to add a gas-like hydrogen density.

Working in the space of two densities requires methods of constructing phase diagrams of coexistence that keep the all densities conserved. It is a significantly more challenging task to deal with two conserved components than it is to deal only with concentration, as in standard models. Another key contribution of this thesis is the development of an efficient computational tool used for calculating two component phase diagrams for the models proposed here, as well as for future models. We develop theory of these phase diagram construction tools in the text of this chapter, and show the details of the algorithm in Appendix C.

4.1 Simplified density functional theory for a two-component system

We begin by introducing the ideal free energy density for a multi-component system consisting of N species. As in the case of single component materials, we will express a species' true ideal free energy as a Taylor series expanded close to the free energy of a reference liquid at coexistence with the solid phase we are trying to model. Specifically, in dimensional units, the true ideal free energy of

component i near coexistence is given by[52]

$$\frac{\mathcal{F}_i^{id}}{k_B T} = \rho_i \log\left(\frac{\rho_i}{\bar{\rho}_i}\right) - (\rho_i - \bar{\rho}_i)\,, \tag{4.1}$$

where ρ_i is the local number density of species i, $\bar{\rho}_i$ represents the number density of species i in the reference liquid, k_B Boltzmann's constant and T is the temperature. Defining a rescaled species density by $n_i = (\rho_i - \bar{\rho}_i)/\bar{\rho}_i$, and expanding the density-rescaled form of Eq. 4.1 in a Taylor series around $n_i = 0$ to fourth order, gives

$$\frac{\mathcal{F}_i^{id}}{k_B T} = \bar{\rho}_i \left(\frac{n_i^2}{2} - \eta_i \frac{n_i^3}{6} + \chi_i \frac{n_i^4}{12}\right), \tag{4.2}$$

where the series expansion yields $\eta_i = \chi_i = 1$. However, it is possible that higher order correlation interactions could produce bulk polynomial terms of third and fourth order, in which case η_i and χ_i could take on other values. In terms of Eq. 4.2 the total ideal free energy becomes

$$f_{id} = \sum_i \frac{\mathcal{F}_i^{id}}{k_B T \bar{\rho}} = \sum_i w_i \left(\frac{n_i^2}{2} - \eta_i \frac{n_i^3}{6} + \chi_i \frac{n_i^4}{12}\right) \tag{4.3}$$

where $\bar{\rho} = \bar{\rho}_1 + \bar{\rho}_2 + \cdots$ is the total average number density of the system and $w_i = \bar{\rho}_i/\bar{\rho}$. The weights w_i come out formally from the rescaling of the total ideal free energy by $k_B T \bar{\rho}$ (these also formally appear in the excess terms discussed below when written in terms of the densities n_i). In what follows, we consider only two components and consider the simple case where $w_1 \approx w_2 = 1$, and $\eta_i = \chi_i = 1$ for each species. We make these simplification here as the aim of this work is to investigate new physical properties that come out the *form* of of

our new model. Future work will explore the use of such constants on specific quantities. With these simplifications, the ideal free energy of our binary model is taken as

$$f_{id} = \sum_i \left(\frac{n_i^2}{2} - \frac{n_i^3}{6} + \frac{n_i^4}{12} \right). \tag{4.4}$$

For the excess free energy density, truncating at two-point correlations, there are in general self-interaction terms for each component, as well as between every pair of components [42]:

$$f_{ex} = -\frac{1}{2} \sum_{i,j} n_i(\mathbf{r}) \int C_2^{ij}(\mathbf{r} - \mathbf{r}') n_j(\mathbf{r}')\, d\mathbf{r}'. \tag{4.5}$$

Specifically, for a two component system (n_A and n_B), we have:

$$\begin{aligned} f_{ex} = &-\frac{1}{2} n_A(\mathbf{r}) \int C_2^{AA}(\mathbf{r} - \mathbf{r}') n_A(\mathbf{r}')\, d\mathbf{r}' \\ &-\frac{1}{2} n_B(\mathbf{r}) \int C_2^{BB}(\mathbf{r} - \mathbf{r}') n_B(\mathbf{r}')\, d\mathbf{r}' \\ &- n_A(\mathbf{r}) \int C_2^{AB}(\mathbf{r} - \mathbf{r}') n_B(\mathbf{r}')\, d\mathbf{r}'. \end{aligned} \tag{4.6}$$

As mentioned above, Greenwood *et al.* [69] take a different approach and transform these equations into a density-concentration space, where the density is taken to be the total density of all components and the concentration is the fraction of B atoms. Additionally, they take the inter-species correlation function C_2^{AB} to be an interpolation between the self-interaction correlation functions, C_2^{AA} and C_2^{BB}, with the interpolation modulated by the concentration field. Here we take an alternative approach, expressing the model in terms of the individual component

densities and retaining all three correlation functions explicitly. Doing so allows us to produce a greater variety of structures.

4.2 Form of the two-point correlation functions

The essence of structure in PFC models—and indeed all DFT models—comes from the structure of the two-point (and higher) correlation functions. For the new XPFC binary model defined above, we define a general class of correlation functions that have the following properties:

1. bounded in real space;
2. exhibit short-ranged repulsion and longer ranged attraction, each characterized by a strength and length scale; and
3. have analytical forms for their k-space representations.

To this end we propose the following form of the correlation functions:

$$C_2^{ij}(r) = -R_{ij}\frac{\alpha^{ij}}{\pi}e^{-\alpha^{ij}r^2} + \frac{A_{ij}}{2\pi a^{ij}}\delta(r - a^{ij}), \tag{4.7}$$

where the parameters α^{ij} and a^{ij} are given by

$$\alpha^{ij} = (\alpha^i + \alpha^j)/2; \quad a^{ij} = (a^i + a^j)/2 \tag{4.8}$$

It is noted that in our computational code we use $\alpha^{ij} = (\alpha^i + \alpha^j)/2$ and $a^{ij} = (a^i + a^j)/2$. The Fourier transform of this correlation function is give by (see

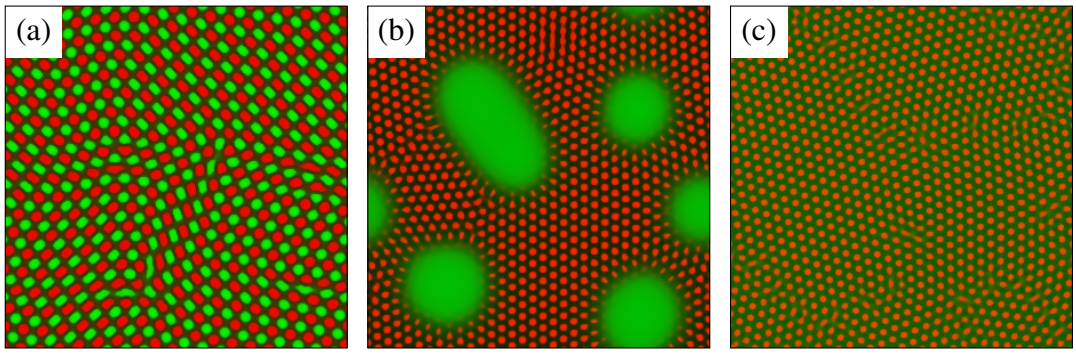


Figure 4.1: Density fields of three two component systems, with the component densities A and B coloured red and green, respectively. (a) shows a salt-like system where A and B atoms are attracted to each other but not themselves. (b) shows a system where A atoms form a solid, B atoms favour remaining disordered, but A and B atoms repel each other. (c) shows a system where the disordered B atoms favour dispersing within the lattice of the ordered A atoms. The details of the models used to simulate these three situations are discussed further below.

Appendix B for details):

$$\hat{C}_2^{ij}(k) = -R_{ij}e^{-k^2/(4\alpha^{ij})} + A_{ij}J_0(a^{ij}k), \tag{4.9}$$

where J_m are the Bessel functions. The parameters $\{R_{ij}, A_{ij}, \alpha^{ij}, a^{ij}\}$ control the phases produced by the system.

Figure 4.1 shows "snapshots" of three example systems generated with the above model using the general form of $C_2^{ij}(r)$ in Eq. 4.7. Later sections of this chapter will investigate these three systems in further detail. These binary systems will serve as case studies that can be expended upon in future work to model three important material classes that have not been studied previously with PFC modelling.

4.3 Phase diagrams for two component systems

In a single component system, if there are two phases coexisting at densities n_1 and n_2, with volume fractions v_1 and v_2, then the average density and average free energy density (denoted n and $f'(n)$, respectively) will be

$$n = n_1 v_1 + n_2 v_2; \tag{4.10}$$

$$f'(n) = f(n_1)v_1 + f(n_2)v_2. \tag{4.11}$$

Since $v_1 + v_2 = 1$, we can set $v_2 = v$ and $v_1 = 1 - v$. This gives

$$n = n_1(1 - v) + n_2 v; \tag{4.12}$$

$$f'(n) = f(n_1)(1 - v) + f(n_2)v. \tag{4.13}$$

With some rearrangement we can obtain:

$$f'(n) = \frac{(f(n_2) - f(n_1))n + f(n_1)n_2 - f(n_2)n_1}{n_2 - n_1}, \tag{4.14}$$

which for a given n_1, n_2 defines an equation of a line (see red line in Fig. 4.2). If $f'(n) < f(n)$ then the minimization of total free energy of the system favours separation of a single phase into regions of density n_1 and n_2; otherwise, minimization favours the entire system remaining at the average density n. Naturally, through fluctuations, the system will select separation into densities n_1, n_2 to minimize the free energy.

The above conditions imply that if the free energy as a function of the density is convex (a line segment between any two points on a graph of the function lies

above the graph), then the entire system will stay at that density and there will be no points of coexistence. If, on the other hand, the free energy as a function of the density is concave over any region, then that region is termed a coexistence region and the endpoints of the region are the densities n_1 and n_2 in the formulae above. In this case the system will separate into regions of density n_1 and n_2 in order to achieve a system average free energy that is lower than the free energy of the system with uniform density. This lower free energy of the coexisting system is its true free energy. As a result, the plot of the true equilibrium free energy of the system as a function of the system density is always convex; regions of coexistence are straight lines. Therefore free energy "landscapes" generated by assuming a single system-wide density must be corrected by taking the convex hull constructed from the free energy landscape, as illustrated in Fig. 4.3.

For two component systems the situation is more complicated. Here we can in general have three coexisting regions of volume fractions v_1, v_2 and v_3, with the constraint that $v_1 + v_2 + v_3 = 1$ allowing us to eliminate v_3 as an independent variable. v_1, v_2 can then be determined by the solution to the following formulae:

$$n^A = n_1^A v_1 + n_2^A v_2 + n_3^A (1 - v_1 - v_2); \tag{4.15}$$

$$n^B = n_1^B v_1 + n_2^B v_2 + n_3^B (1 - v_1 - v_2). \tag{4.16}$$

Meanwhile, the average free energy density of the system is given by

$$\begin{aligned} f'(n^A, n^B) = &\big(f(n_1^A, n_1^B) - f(n_3^A, n_3^B)\big) v_1 \\ &+ \big(f(n_2^A, n_2^B) - f(n_3^A, n_3^B)\big) v_2 \\ &+ f(n_3^A, n_3^B), \end{aligned} \tag{4.17}$$

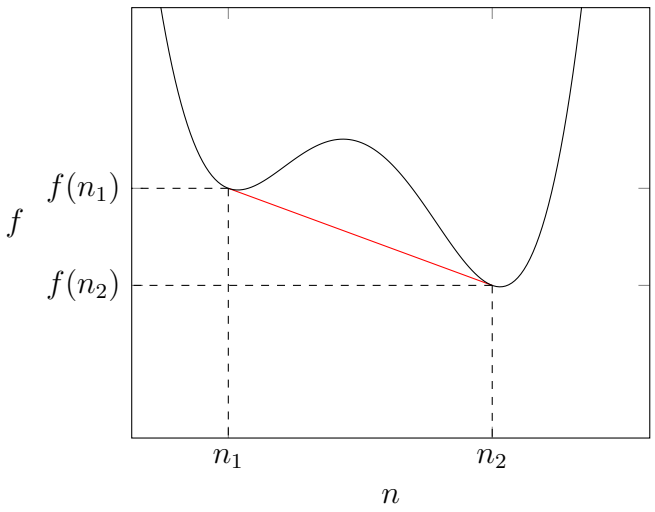


Figure 4.2: Free energy plot as a function of density. The red line is the coexistence line, where one phase decomposes into two phases of density n_1 and n_2. As the average density moves from n_1 to n_2, the system remains in coexistence at these endpoints, however the volume fraction changes from $\{v_1, v_2\} = \{1, 0\}$ to $\{v_1, v_2\} = \{0, 1\}$.

which is the formula for a plane. It should be noted that sometimes $n_1^A, n_1^B = n_3^A, n_3^B$ (or equivalent) and we have a line. Similarly to the case of single component system, the convexity of the free energy landscape of the system is critical in defining coexistence. If the system can achieve a lower free energy by separating into two or three regions then it will, and so a plot of the true free energy of the system will always be convex as a function of the densities. Thus, finding the equilibrium free energy "landscape" of a system as a function of n_A and n_B requires mapping the local free energy to its convex hull. A computational procedure and algorithm for doing so is presented in Appendix C. This algorithm was used in the phase diagram examples of the binary model proposed in this chapter and demonstrated below.

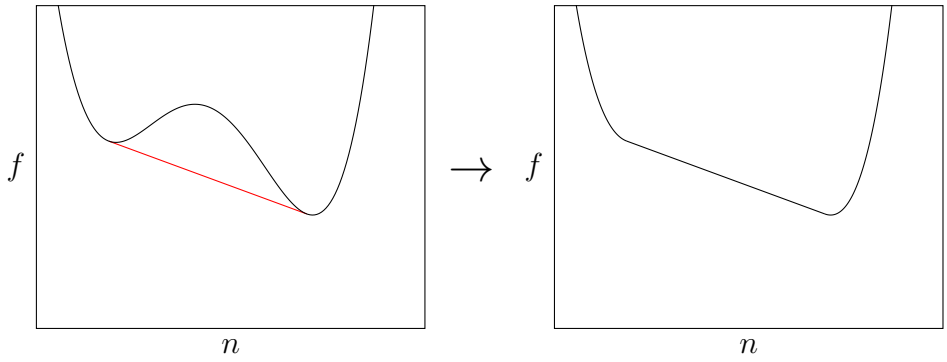


Figure 4.3: Replacing the free energy plot with its convex hull. The convex hull represent the true equilibrium free energy of the system.

4.4 Special case studies of the binary model

This section specializes the above binary XPFC theory to three types of systems modelled by the class of two-point correlation functions in Eq (4.7). These are a salt-like binary system, a solid-gas system and an interstitial alloy system. These systems are demonstrated mostly to illustrate the robustness of the new binary model, and indeed, the salt-like system is merely a toy model of the real system it mimics here. It is expected that these three example systems will serve as templates to build upon for modelling more complex materials in the above three classes of materials.

4.4.1 "Salt-like" system

In an ionic salt, positively charged cations are attracted to negatively charged anions. Both species repel themselves; hence the resulting structure has each cation surrounded by anions, and each anion surrounded by cations. Motivated by this

behaviour, we generated a "salt-like" system like the one shown in Figure 4.1a by using $\{\alpha^{ij}, a^{ij}\} = \{1.5, 1\}$. We also set the-short ranged rejection parameters to $R_{AA} = R_{BB} = R_{AB} = 6s$, where s is a parameter setting the overall strength of the interaction ($s = 1$ in Figure 4.4). We want the two components to attract each other at the lattice spacing distance, so we set $A_{AB} = 6s$. However, we want each component to reject itself at that distance as well, so we set $A_{AA} = A_{BB} = -6s$. When the average densities are equal, the result is a system that solidifies into a square lattice where each atom neighbours atoms of the other species. Two snapshots of the early and late time evolution of our model forming a solid with A-B atoms arranged into interlacing square lattices are shown in Figure 4.4.

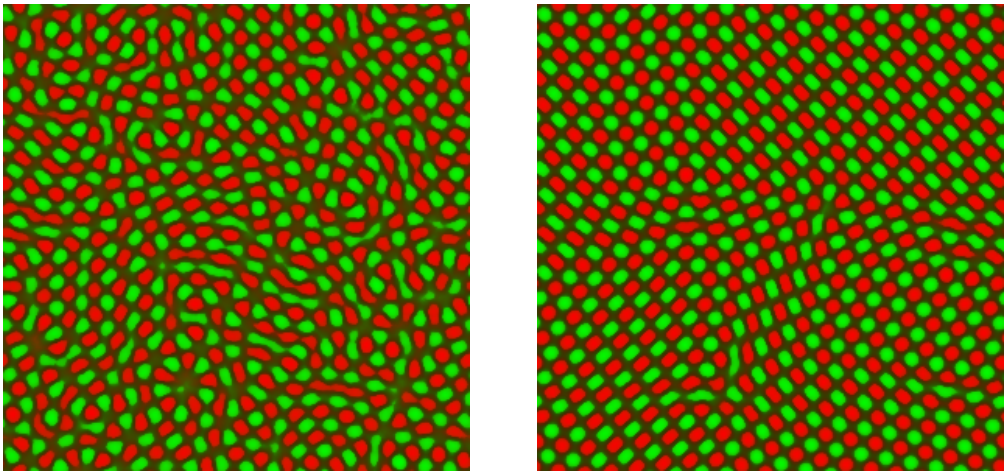


Figure 4.4: Early (left) and late (right) density evolution for the "salt-like" binary system. The density n_A is indicated in red, and n_B is indicated in green.

A numerical phase diagram for this toy "salt-like" system is depicted in Figure 4.5. This specific phase diagram shows a cut along $n_A = n_B$ line in (n_A, n_B) space.

To properly describe an actual salt like system requires the use of screened

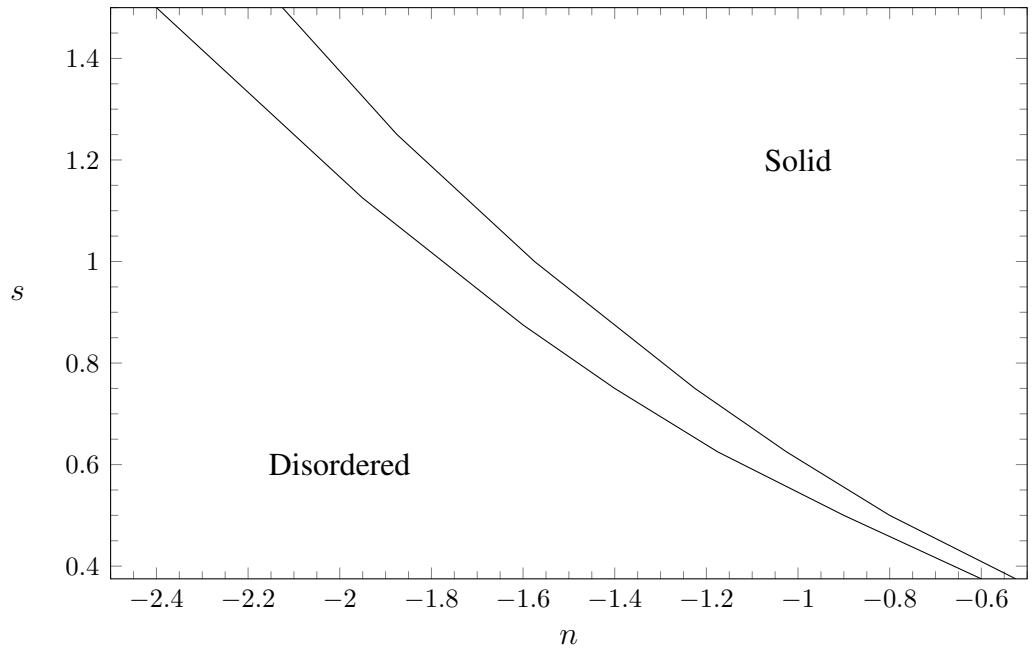


Figure 4.5: Phase diagram for the "salt like" toy system (Figures 4.1a and 4.4). The two lines indicate the coexistence region. For all points in the phase diagram $n_A = n_B = n$. The strength of the interaction is determined by s.

Coulomb interactions as done in the theory of ionic liquids [105]. The model illustrated here is a crude toy model analogy of this type of material, which is made using the interactions defined in Eq. (4.7). It would be an interesting future application of this binary model to include more detailed screened ionic interactions into the correlation functions and to use these to investigate defects in ionic materials.

4.4.2 Solid-gas system

The solid-gas like system in Figure 4.1b represents a binary material where the A species favours a solid phase, and the B species favours to be a gas (or a disordered) phase; however the A-rich solid can still contain a fraction of B atoms within it. To simulate this situation with the binary model, we again set $\{\alpha^{ij}, a^{ij}\} = \{1.5, 1\}$. In this system the B atoms are "gas-like" and so have no self-interaction, and so we set $R_{BB} = A_{BB} = 0$. The A atoms form a solid, and so these exhibit an attraction at the lattice spacing length and repulsion at smaller length scales. Thus, we set $R_{AA} = A_{AA} = 6$. Lastly, we set the two components to be weakly repulsed by each other by setting $R_{AB} = 2$, $A_{AB} = 0$. This system favours in the A atoms solidifying with pockets of the disordered B atoms between them.

Figure 4.6 shows a phase diagram of the solid-gas system constructed in (n_A, n_B) space. The phase diagram was made by the same method outlined in Section 4.3 of this chapter using a new computational algorithm developed for two component materials, which is discussed in Appendix (A). The lines in the phase diagram indicate tie lines. The lines connect any pair of average densities $(\langle n_A \rangle, \langle n_B \rangle)$ in the coexistence region to the unique points at the boundaries of the coexistence region of the phase diagram. Note that the discrete nature of the coexistence boundaries in Figure 4.6 are due to numerical resolution. These boundaries become smother as the resolution of the convex hull algorithm described in Appendix C increases, at the cost of course of increased time to compute the phase diagram.

Figure 4.7 shows "snapshots" of a simulation of the time evolution of a solid-gas system. The system is prepared with Gaussian noise with average densities $n_A = -0.55$ and $n_B = -0.55$. This puts the system in the coexistence part of the

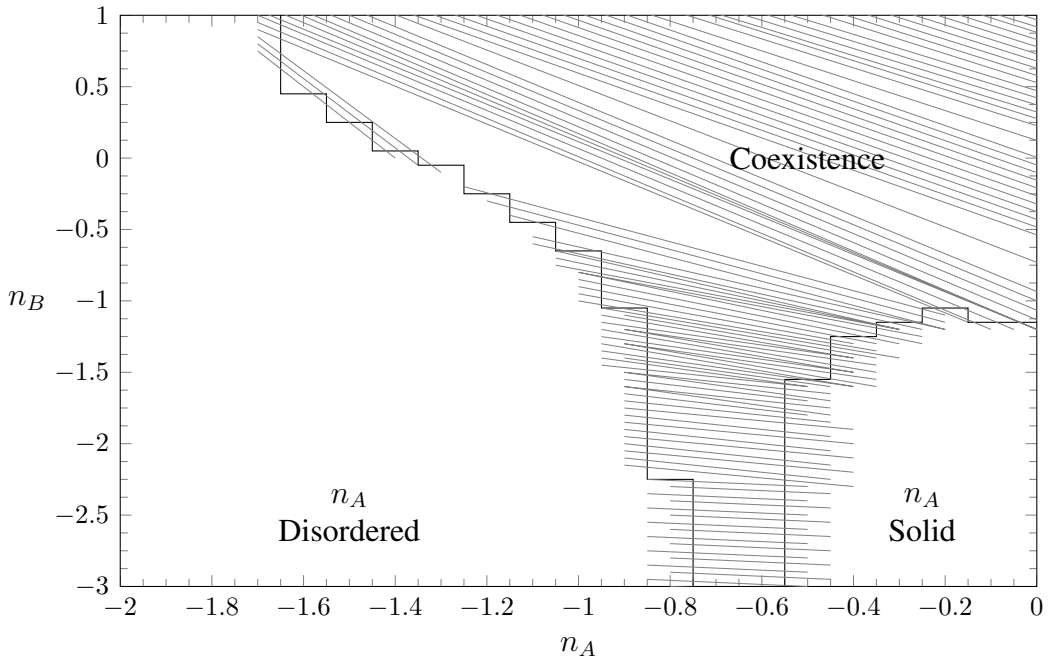


Figure 4.6: Phase diagram for the second system (Figure 4.1b), with $\alpha^{ij} = 1.5$, $a^{ij} = 1$, $R_{BB} = A_{BB} = A_{AB} = 0$, $R_{AA} = A_{AA} = 6$ and $R_{AB} = 2$. Grey lines indicate coexistence tie lines. Species B is always disordered.

phase diagram of Figure 4.6. After an initial solidification into a polycrystalline solid, over time B-rich gas pockets develop at grain boundaries. The latter develop at grain boundaries as these are higher energy locations that mediate the nucleation of such gas pockets, as expected.

4.4.3 Interstitial impurity system

In the system shown in Figure 4.1c, species B, which on its own favours a gas phase, is also able to exist within the interstitial regions of an A-rich solid phase. This can serve as the first PFC paradigm to our knowledge of an interstitial species

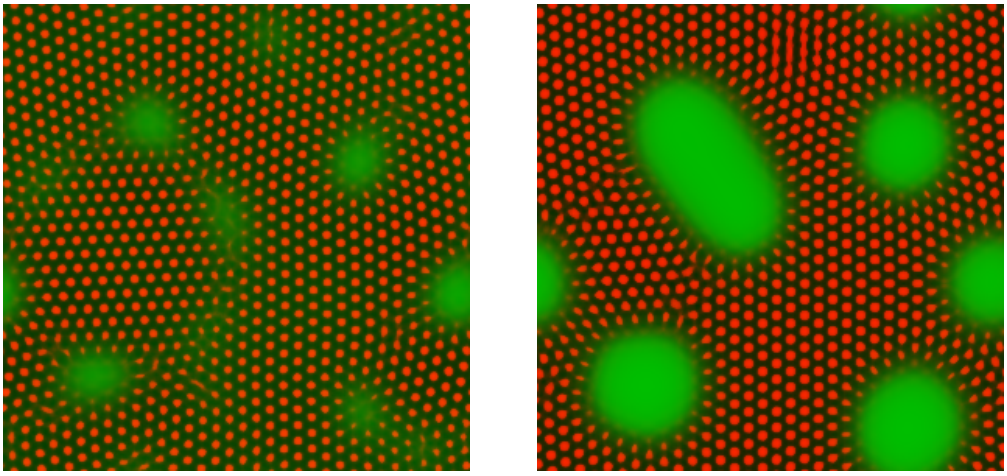


Figure 4.7: Early (left) and late (right) density evolution for the gas-solid system. n_A indicated in red, n_B indicated in green. For this simulation, we took $n_A = n_B = -0.5$.

diffusing within a host solid of another species. One of the most common real-world examples of this is steel, where carbon is an interstitial element in an iron lattice.

To simulate this interstitial system, system, we set $\alpha^{ij} = 2.5$, $a^A = 1$, $a^B = .5$ (see Eq. 4.8). In this system the A atoms form a solid while the B atoms form a diffuse gas in the interstitial sites. $R_{AA} = A_{AA} = 6$ produces a solid for the A component. $R_{BB} = A_{BB} = 0$ results in the B atoms remaining diffuse with no self-interactions. The B atoms are however attracted to the spaces between the A atoms by $R_{AB} = 3$, $A_{AB} = 1$. We additionally change the diffusion constant for the B atoms to be ten times larger than that of the A atoms, reflecting their smaller size.

Figure 4.8 shows a polycrystalline sample of A-rich phase with interstitial B atoms that are attracted to the interstitial spaces between the A atoms. The n_B density is also greater near defects in the lattice, as expected.

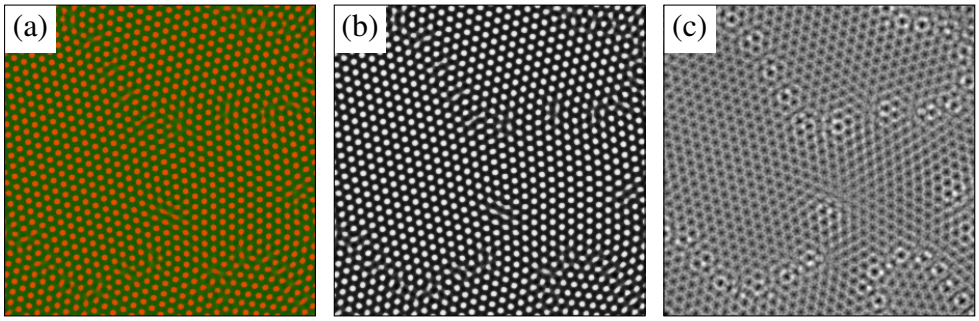


Figure 4.8: Separated density fields of the steel-like system. (a) shows both densities together, with n_A in red and n_B in green. (b) shows n_A by itself. (c) shows n_B by itself. Note that the n_B field is attracted to grain boundaries and other defects in the n_A crystal.

System	A-A $\{R_{AA}, A_{AA}\}$	B-B $\{R_{BB}, A_{BB}\}$	A-B $\{R_{AB}, A_{AB}\}$	$\{\alpha^A, \alpha^B\}$	$\{a^A, a^B\}$
Salt-like	$\{6, -6\}$	$\{6, -6\}$	$\{6, 6\}$	$\{1.5, 1.5\}$	$\{1, 1\}$
Gas-solid	$\{6, 6\}$	$\{0, 0\}$	$\{2, 0\}$	$\{1.5, 1.5\}$	$\{1, 1\}$
Interstitial impurity	$\{6, 6\}$	$\{0, 0\}$	$\{3, 1\}$	$\{2.5, 2.5\}$	$\{1, 0.5\}$

Table 4.1: Summary of system parameters for the three systems described.

For completeness, Table 4.1 summarizes all the parameters used to demonstrate the three systems described in the three subsections of the present section.

4.5 Binary model of graphene-hydrogen surface interactions

A common way to grow graphene in experiments is through chemical vapour deposition (CVD) onto a surface of some metal immersed in controlled gaseous environment, where one gas controls the transfer of carbon to the surface and the other

the overall pressure of the sample[106, 107]. A particular example is that used at McGill in the lab of professor Michael Hilke. In his group's experimental set up, graphene grows on the surface of a copper sample that is in a two-component gas of molecular hydrogen (H_2) and methane (CH_4). It is conjectured that these two gasses, which are maintained at some low pressure, dissociate and adsorb as carbon (C) and hydrogen (H) onto the copper surface. Adsorption of each species is proportional its carrier gas' partial pressure. As the surface concentration of carbon (or the pressure of CH_4) increases, graphene flakes start to grow. While the role of the hydrogen on the surface is presently not well understood, its presence appears to interfere with the growth of the carbon. It is plausible that the carbon and hydrogen on the surface form a mixture out of which graphene grows in regions of low enough hydrogen density, assuming a locally high carbon density.

4.5.1 Properties of the model

To model CVD graphene as a binary system, we add a second species to the graphene model described in Chapter 3. We use n_A to represent the carbon surface density and add n_B to represent the hydrogen surface density. The total free energy density of the system (in dimensionless form) is $f(n_A, n_B) = f_A(n_A) + f_B(n_B) + f_{AB}(n_A, n_B)$, while, as usual, the total free energy of the system is $F = \int f(n_A, n_B)\, dV$, where the volume is over the system. We take the free energy of the graphene forming carbon species (A) to be given by Equations 3.2, 3.6, 3.10 and 3.11–3.13. There is no self-interaction of the hydrogen atoms, which implies that the B species on their own favour an ideal gas state.

Thus, we take the hydrogen component of the total free energy to be

$$f_B = \int \left(\frac{n_B(\boldsymbol{r})^2}{2} - \frac{n_B(\boldsymbol{r})^3}{6} + \frac{n_B(\boldsymbol{r})^4}{12} \right) d\boldsymbol{r} \tag{4.18}$$

To this we add an additional term to model an interaction between the two components:

$$f_{AB}(\mathbf{r}) = \big(a n_A(\mathbf{r}) + b n_A^2(\mathbf{r}) \big) \int \chi(\mathbf{r}' - \mathbf{r}) n_B(\mathbf{r}')\, d\mathbf{r}' \tag{4.19}$$

where the function $\chi(r' - r)$ here comprises a long-range two-point correlation function. This form essentially causes an A atom (carbon) to feel the effect of hydrogen in an averaged way in some region around It. The form of $\chi(r' - r)$ in Fourier space is given by

$$\hat{\chi}(k) = B e^{-k^2/(4\alpha)}. \tag{4.20}$$

The constants a and b in Eq. (4.19) influence the type of interactions. For example, using $b = 0$ mostly couples the average of n_A to the local average of n_B , while the case $b \neq 0$ couples the ordering parameters (amplitudes) of n_A to the local average of n_B. The constant B controls the magnitude of interaction and α adjusts the interface energy between the fields. In the results shown below, $B = 2$ and $\alpha = 0.5$.

Figure 4.9 shows a constant temperature the phase diagram of the above model for the case where the interaction parameters are set to $a = b = 1$. The same two-component phase diagram algorithm discussed previously was used, only now the amplitude expansions included the more complex three-point interactions in

$f_A(n_A)$. In the figure, the temperature parameter of the three-point model was set to $X^{-1} = 0.4$ and the two-point interaction strength was set to $R = 6$.

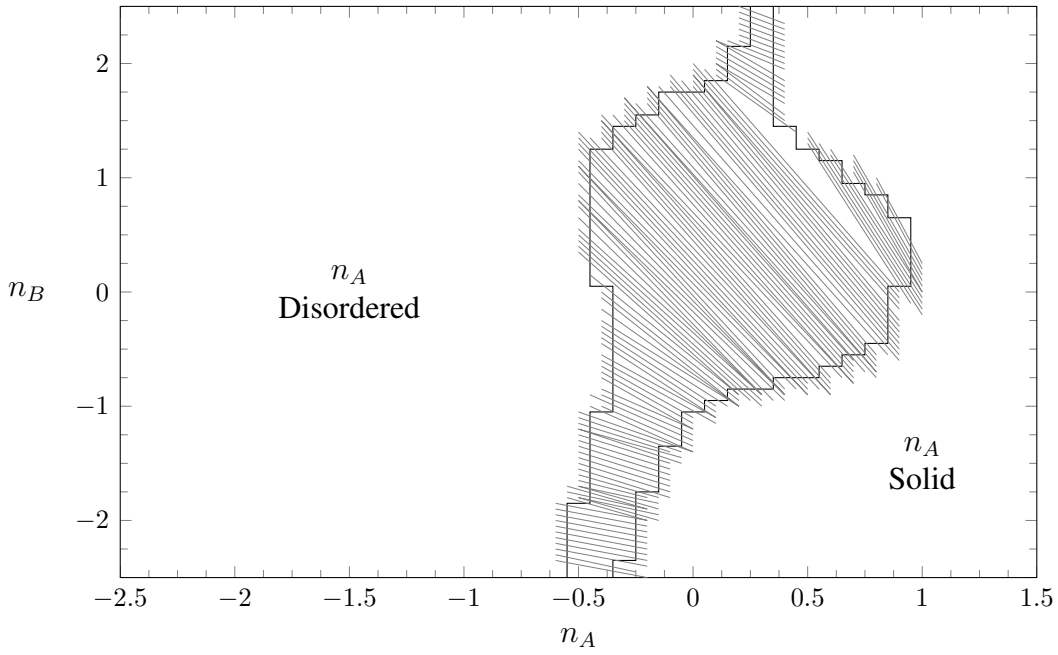


Figure 4.9: Phase diagram for the graphene-hydrogen system, with $a = b = 1$ in the hydrogen-carbon interaction term and $R = 6$, $X^{-1} = 0.4$, $B = 0.125$ in the graphene three-point correlation function. The grey lines indicate coexistence tie lines for the graphene and disordered phases. Species B ("hydrogen") is always disordered.

4.5.2 Dynamics

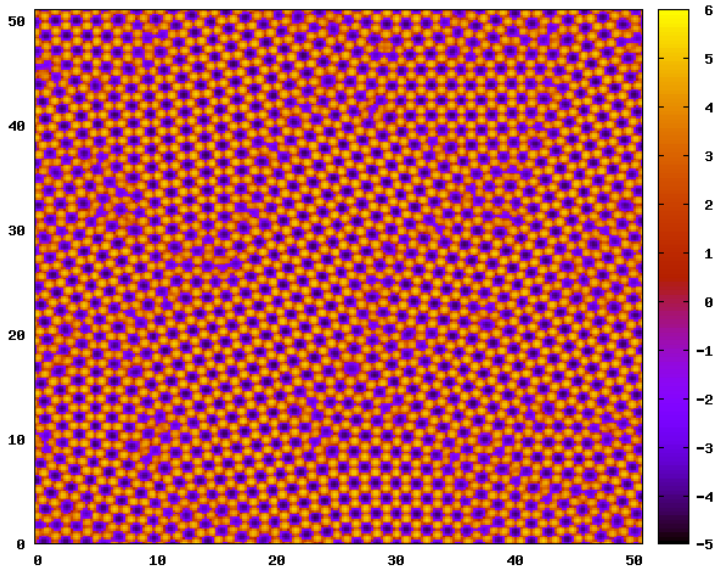
To simulate dynamics in the above model, each species follows the usual model B type equations of motion given by Eq. 3.46. To mimic the CVD process, however, we add to each species' equations of motion a flux term to represent the flux of carbon and hydrogen onto the surface from the vapour phase surrounding the 2D

surface onto which graphene grows. As a result, the equations of motion become, for each species n_i, $i = A, B$:

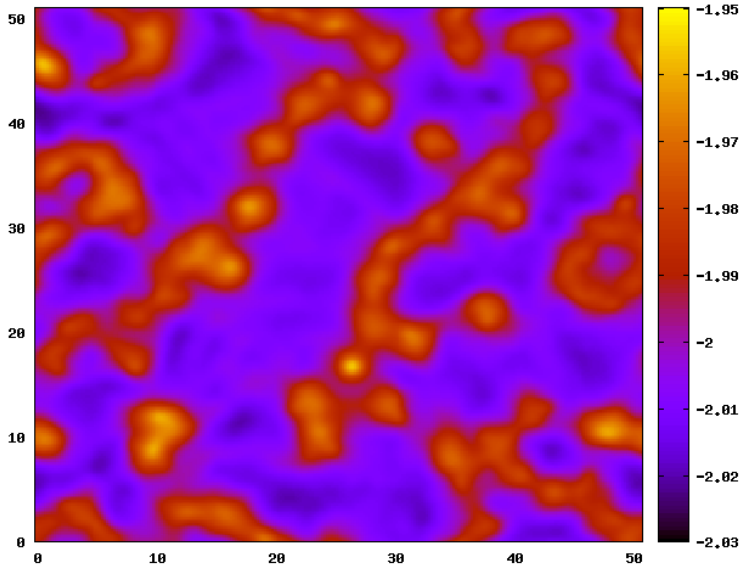
$$\frac{\partial n_i}{\partial t} = M_n \nabla^2 \left(\frac{\delta F}{\delta n_i} \right) - f(n_i - n_i^*), \tag{4.21}$$

In Eq. 4.21, n_i^* is the "target" density at which the flux onto the surface reduces to zero. In the results presented below, $f = 1.95$ was used.

Figure 4.10 shows a graphene polycrystalline grown with the graphene-hydrogen binary model. The temperature parameter was set to $X^{-1} = 0.4$, and the interaction parameters in f_{AB} were set to $a = 1$ and $b = 0$. The densities n_A and n_B were initialized uniformly with Gaussian fluctuations of zero mean and unit variance. Their averages $(\langle n_B \rangle - \langle n_A \rangle)$ were chosen in the solid region of the phase diagram corresponding to $a = 1$ and $b = 0$ in Figure 4.9. The top image shows the graphene structure set by n_A (carbon), while the bottom image shows the distribution of n_B (hydrogen). The hydrogen is seen to be disordered, with larger accumulations in the grain boundaries, consistent with experimental observations [108, 109], which suggests that hydrogen chemically mixes (binds) with carbon within the bulk of graphene, but has a higher propensity for binding on defects. This simulation begins with the nucleation of different crystal orientations from the disordered phase, followed by growth and impingement leading to grain boundaries.



(a)



(b)

Figure 4.10: Results of the graphene-hydrogen model. (a) shows the carbon density, n_A, and (b) shows the hydrogen density, n_B. Note the attraction of the hydrogen atoms to the grain boundaries of the graphene. $n_A = 1$, $n_B = -2$, $R = 4.5$, $X^{-1} = 0.35$, and $a = 2$ and $b = 0$. Results courtesy of Kate Elder.

Chapter 5

Conclusion

Using the PFC methodology, we developed and characterized several new models of atomic scale material phase transformations. In doing so, we have extended the reach of PFC to investigate new areas of material science. New models have been developed to: (1) model consistent magneto-crystalline interactions; (2) introduce into the PFC methodology rotationally invariant three-point density correlations; and (3) develop a new formalism to model binary systems based on species densities and local inter-species interactions.

In Chapter 2 we expanded the PFC model of Faghihi *et al.*[51], which was only valid for isotropic magneto-crystalline coupling, to include anisotropy and use the XPFC two-point correlation functions as the underlying model of the density evolution. We derived analytically the form of the anisotropy for various crystal structures. We also calculated the model's magnetostriction coefficients. We demonstrated that a single-peak correlation function does not give enough degrees of freedom to separately tune the magnetostriction constants, however multi-peaked XFPC correlation functions do allow for separate control of these

constants. Magnetic hysteresis was also quantified in the model through direct simulation. We simulated the effects of external magnetic fields on the resulting grain structure of a solidifying crystal, showing that grains in anisotropically favourable orientations grow preferentially at the expense of those in unfavourable orientations. This points to potentially interesting applications of using applied external magnetic fields to effect microstructure control in materials manufacturing. A potentially interesting avenue of future work with this model would include the study of using low-level magnetic fields to bias magnetic impurities to grain boundaries of polycrystalline materials. This can be of use in thermoelectric materials, where it is favourable to form many grain boundaries in order to increase the dispersion of lattice phonons (thus decreasing thermal conductivity), while minimizing the impact on electrical conductivity due to the enhanced presence of grain boundaries. The migration of certain elements over others can have demonstrable effects in the latter electronic effect[110].

Chapter 3 introduced a new type of PFC model that included, for the first time in the PFC literature, a three point correlation function. These three point interactions favour particular bond angles while being constructed so as to be rotationally invariant. The aim of this was to increase control of complex crystallographic lattices beyond the standard structures found in metals, without compromising the ability to control previously studied structures as well. In this study, a simple repulsive interaction was used as the two-point correlation function. These new interaction forms were then applied to the modelling of graphene. We derived phase diagrams and simulated systems for 60°, 90° and 120° bond angles, producing triangular, square, and honeycomb (graphene) ordering, respectively. Simulations demonstrated that the defect structures at grain boundaries, consisting of

rings of five and seven atoms, compared excellently with those found experimentally. We also calculated the elastic constants of this model and found ranges in the space of average density and the model's temperature parameter where these elastic constants are consistent with theory; this is in contrast to recent other models of graphene in the literature which are less stable. An interesting avenue of future work will be to extend the model of Chapter 3 to include three-point interactions in three dimensions. One potential approach is to replace the $C_s^i(r,\theta)$ by spherical harmonics. The idea is sketched out in Appendix E. We are also collaborating with a previous Postdoc of the Provatas group (Dr. David Montiel) at the University of Michigan who is exploring this approach extending the three-point interaction to three dimensions in order to produce graphene sheets. Also, by applying this 3D idea, the Michigan group and their own collaborators have recently yielded 3D perovskite structures[111].

Another avenue of future work with the new three-point structural phase field crystal (XPFC) model introduced in Chapter 3 is investigating to what extent adding a Gaussian peak in the two-point correlation function, as well as the softening the delta function peak by a Gaussian function in the three-point interaction, would make the model less numerically stiff without compromising the salient structural benefits introduced in this current model introduced herein.

Chapter 4 studied binary systems. We introduced several new binary XPFC models, formulated in terms of two independent densities, n_A and n_B (as opposed to previous formulations based on density n and concentration c). This representation has several benefits, including more robust control over atomic self-interactions and inter-atomic interactions, since we can explicitly control the three types of two-point correlation functions, C_2^{AA}, C_2^{BB} and C_2^{AB}. It also allows us

the ability to assign each component a unique diffusion coefficient, which is not possible for models formulated based on density and concentration fields. In this framework, we proposed a new class of two-point correlation functions for C_2^{ij} that featured both long and short-ranged interactions. We used these correlation functions to model more exotic material structures such as salt-like substances, solid-gas coexistence, and carbon diffusing into the interstitial sites of iron. Future work along these lines could modify the latter model into a model of free electrons in metals by including long-range screened Coulomb interactions that can lead to a PFC model for ionic fluids, where the two components of the system represent the ionic cores (nuclei and their core electrons) and the valence electrons, which can play the role of a classical electron "gas" that interacts with the ionic cores. By coupling the classical electron field to a local voltage, this new two-component XPFC model could have potentially interesting applications in electromigration. Another future direction of research is to more thoroughly examine the interstitial binary model presented here to map out the parallels to real steel (iron-carbon alloy). Preliminary simulations we conduced have shown, for example, that excess amount of the interstitial segregate at grain boundaries, a promising property of real steel.

Chapter 4 also developed a new model of graphene vapour deposition. By taking the graphene model developed in Chapter 3 and adding a second field representing hydrogen, we developed a simple model of graphene vapour deposition on a copper surface. In this model, the A species (carbon) favours the graphene state and the B species (hydrogen) favours a gas phase. Kate Elder, a Masters student in the Provatas group with who I have collaborated, has run chemical vapour deposition experiments of the deposition of carbon (from $\mathrm{CH_4}$) and hydrogen (from $\mathrm{H_2}$

and the aforementioned CH_4) onto copper surfaces to examine the effects of the gas flow rates on graphene dendrite formation. She has also run simulations of the graphene-hydrogen model to investigate etching of the graphene grain boundaries by hydrogen. Her simulations are along the lines shown in Chapter 4, but have characterized the role of hydrogen levels in grain boundaries more thoroughly. Future work and a collaboration with the Hilke group at McGill will conduct further experiments on full coverage graphene to compare the role of hydrogen in experiments to Kate Elder's simulation with the present graphene-hydrogen model.

Appendix A

Numerical Approaches for Computing Phase Diagrams

In $N = 3$ dimensions, given a set of primitive lattice vectors $\mathbf{a_1}, \mathbf{a_2}, \mathbf{a_3}$ defining a Bravais lattice, one finds the primitive vectors of the reciprocal lattice vectors by[112, p. 29]:

$$\mathbf{b_1} = 2\pi \frac{\mathbf{a_2} \times \mathbf{a_3}}{\mathbf{a_1} \cdot (\mathbf{a_2} \times \mathbf{a_3})}; \tag{A.1}$$

$$\mathbf{b_2} = 2\pi \frac{\mathbf{a_3} \times \mathbf{a_1}}{\mathbf{a_1} \cdot (\mathbf{a_2} \times \mathbf{a_3})}; \tag{A.2}$$

$$\mathbf{b_3} = 2\pi \frac{\mathbf{a_1} \times \mathbf{a_2}}{\mathbf{a_1} \cdot (\mathbf{a_2} \times \mathbf{a_3})}. \tag{A.3}$$

However these formula don't generalize to $N \neq 3$. For a general formula, place the N column vectors corresponding to the primitive lattice vectors next to each other to construct an $N \times N$ matrix, $[\mathbf{a_1 a_2} \ldots \mathbf{a_N}]$. The primitive reciprocal lattice

vectors can then by found by matrix inversion[112, p. 29]:

$$[\mathbf{b_1}\mathbf{b_2}\ldots\mathbf{b_N}]^T = 2\pi[\mathbf{a_1}\mathbf{a_2}\ldots\mathbf{a_N}]^{-1}. \tag{A.4}$$

The volume of the unit cell can be found using the absolute value of the determinate:

$$V_{\text{cell}} = |\det[\mathbf{a_1}\mathbf{a_2}\ldots\mathbf{a_N}]|. \tag{A.5}$$

This translates succinctly into Mathematica code, with `primitiveVectors` being a list of vectors:

```
GenerateReciprocalLatticeVectors[primitiveVectors_]:=
        2π Inverse[Transpose[primitiveVectors]];
```

For example, if we test our function using the primitive lattice vectors for a triangular lattice:

```
TriPrimitiveVectors = {{1, 0}, {1/2, Sqrt[3]/2}};
GenerateReciprocalLatticeVectors[TriPrimitiveVectors]
>>> {{2 π, -((2 π)/Sqrt[3])}, {0, (4 π)/Sqrt[3]}}
```

Or, for the 2d square lattice:

```
SqPrimitiveVectors = {{1, 0}, {0, 1}};
GenerateReciprocalLatticeVectors[SqPrimitiveVectors]
>>> {{2 π, 0}, {0, 2 π}}
```

Note: the term "plane" is used in the code below due to the equivalence between the indices indicating reciprocal space vectors and Miller indices indicating real space planes.

We now define a function, `Planes`, to generate groups of reciprocal lattice planes. This function takes a set of primitive lattice vectors (`a`) and the range of

indices to generate over (`size`). It will generate every reciprocal lattice vector of the form $k_1\mathbf{a_1} + k_2\mathbf{a_2} + \ldots + k_n\mathbf{a_n}$, with k_i ranging from `-size` to `size`. The sets $\{k_1, k_2, \ldots, k_n\}$ are then grouped by their norm. For example:

```
Planes[TriPrimitiveVectors, 1]
>>> {{0, {{0, 0}}},
        {(4 π)/Sqrt[3], {{1, 1}, {1, 0}, {0, 1}, {0, -1}, {-1, 0}, {-1, -1}}},
        {4 π, {{1, -1}, {-1, 1}}}}
```

We can see that `Planes` has produced three groups with norms `0`, `(4 π)/Sqrt[3]` and `4 π`. Each group has a list of the sets of indices belonging to it (for example, the second group has the sets of indices `{{1, 1}, {1, 0}, {0, 1}, {0, -1}, {-1, 0}, {-1, -1}}`). Note that since the indices are restricted to the range `-size` to `size`, the third group is incomplete here. This will not be a problem in what follows; we will rely on the fact that the first `size + 1` groups are complete.

We implement `Planes` as:

```
Planes[a_,size_]:=Module[
        {b,
         groupedPlanes,
         dim,
         tuples,
         norms
        },
        b=GenerateReciprocalLatticeVectors[a];
        dim=Length[b];
        tuples=Tuples[Range[-size,size],dim];
        norms=Map[
            Function[tuple, Simplify[Norm[Sum[b[[i]] tuple[[i]],{i,1,dim}]]]],
            tuples];
        groupedPlanes=GatherBy[
                Sort[
                        Transpose[{norms,tuples}],
                        #1[[1]]<#2[[1]] &
```

```
                ],
                First
        ];

        Table[
                {
                        groupedPlanes[[i,1,1]],
                        Table[groupedPlanes[[i,j,2]],
                              {j,1,Length[groupedPlanes[[i]]]}]
                },
                {i,1,Length[groupedPlanes]}
        ]
];
```

Next we define two helper functions, `ReciprocalLatticeVectors` and `ReciprocalLatticeVectorsIndices`. The first takes the result from `Planes` and produces a flattened list of vectors, the second does the same but produces the indices. For example:

```
ReciprocalLatticeVectors[TriPrimitiveVectors, 1]
>>> {{0, 0}, {2 π, (2 π)/Sqrt[3]}, {2 π, -((2 π)/Sqrt[3])}, {0, (4 π)/Sqrt[3]},
          {0, -((4 π)/Sqrt[3])}, {-2 π, (2 π)/Sqrt[3]}, {-2 π, -((2 π)/Sqrt[3])},
          {2 π, -2 Sqrt[3] π}, {-2 π, 2 Sqrt[3] π}}
ReciprocalLatticeVectorsIndices[TriPrimitiveVectors, 1]
>>> {{0, 0}, {1, 1}, {1, 0}, {0, 1}, {0, -1}, {-1, 0}, {-1, -1}, {1, -1}, {-1, 1}}
```

These are implemented as follows:

```
ReciprocalLatticeVectors[a_,size_]:=Module[{b,flattenedPlanes,dim},
        b=GenerateReciprocalLatticeVectors[a];
        dim=Length[b];
        flattenedPlanes=ReciprocalLatticeVectorsIndices[a, size];
        Table[
                Sum[
                        flattenedPlanes[[i,j]] b[[j]],
                        {j,1,dim}
                ],
                {i,1,Length[flattenedPlanes]}
```

```
59          ]
60  ];
```

```
61  ReciprocalLatticeVectorsIndices[a_,size_]:=Module[{b,planes,dim},
62          b=GenerateReciprocalLatticeVectors[a];
63          dim=Length[b];
64          planes=Planes[a,size];
65          Flatten[
66                  Table[
67                          planes[[i,2]],
68                          {i,1,Length[planes]}
69                  ],
70                  1
71          ]
72  ];
```

Now we define a subroutine to generate common lattice information that will be used by the functions that follow. `GenerateLatticeInfo` takes a set of primitive lattice vectors and a list of amplitude names. It returns an object containing the following information:

- `dim`: the number of dimensions of the system.
- `VCell`: volume of the primitive unit cell.
- `numPlanes`: number of families of vectors.
- **`K`**: for each family of vectors, the norm common to those vectors.
- β: for each family of vectors, the number of vectors in that family.
- ρ: for each family of vectors, the planar atomic density.
- `Vectors`: all reciprocal lattice vectors generated.
- `VectorsIndices` all reciprocal vectors generated, in indices of the primitive reciprocal vectors.
- `Amplitudes` corresponding amplitudes for all vectors.

For example:

```
TriLatticeInfo = GenerateLatticeInfo[TriPrimitiveVectors, {ϕ[0], ϕ[1], ϕ[2]}];
TriLatticeInfo["dim"]
>>> 2
TriLatticeInfo["numPlanes"]
>>> 3
TriLatticeInfo["VCell"]
>>> Sqrt[3]/2
TriLatticeInfo["K"]
>>> {0, (4 π)/Sqrt[3], 4 π}
TriLatticeInfo["β"]
>>> {1, 6, 6}
TriLatticeInfo["ρ"]
>>> {0, 1, 1/Sqrt[3]}
TriLatticeInfo["Vectors"]
>>> {{0, 0}, {2 π, (2 π)/Sqrt[3]}, {2 π, -((2 π)/Sqrt[3])}, {0, (4 π)/Sqrt[3]},
     {0, -((4 π)/Sqrt[3])}, {-2 π, (2 π)/Sqrt[3]}, {-2 π, -((2 π)/Sqrt[3])},
     {4 π, 0}, {2 π, 2 Sqrt[3] π}, {2 π, -2 Sqrt[3] π}, {-2 π, 2 Sqrt[3] π},
     {-2 π, -2 Sqrt[3] π}, {-4 π, 0}}
TriLatticeInfo["VectorsIndices"]
>>> {{0, 0}, {1, 1}, {1, 0}, {0, 1}, {0, -1}, {-1, 0}, {-1, -1}, {2, 1}, {1, 2},
     {1, -1}, {-1, 1}, {-1, -2}, {-2, -1}}
TriLatticeInfo["Amplitudes"]
>>> {ϕ[0], ϕ[1], ϕ[1], ϕ[1], ϕ[1], ϕ[1], ϕ[1], ϕ[2], ϕ[2], ϕ[2], ϕ[2], ϕ[2], ϕ[2]}
```

This subroutine is implemented as follows:

```
GenerateLatticeInfo[primitiveVectors_,ampList_]:=Module[
        {info,
         numPlanes=Length[ampList],
         VCell,
         β, ρ, K,
         Vectors,
         VectorsIndices,
         Amplitudes
        },
        VCell=Abs[Det[primitiveVectors]];
        K=Map[
```

```
        #1[[1]] &,
        Planes[primitiveVectors,numPlanes][[1;;numPlanes]]
];
β=Map[
        Length[#1[[2]]] &,
        Planes[primitiveVectors,numPlanes][[1;;numPlanes]]
];
ρ=Map[
        Function[k,
                If[k!=0,
                        (2 π)/(k VCell),
                        0
                ]
        ],
        K
];
Vectors=ReciprocalLatticeVectors[
    primitiveVectors, numPlanes][[1;;Total[β]]];
VectorsIndices=ReciprocalLatticeVectorsIndices[
    primitiveVectors, numPlanes][[1;;Total[β]]];
Amplitudes=Flatten[
        Table[
                Table[
                        ampList[[i]],
                        {j,1,β[[i]]}
                ],
                {i,1,Length[β]}
        ]
];
info["dim"]=Length[primitiveVectors];
info["numPlanes"]=numPlanes;
info["VCell"]=VCell;
info["K"]=K;
info["β"]=β;
info["ρ"]=ρ;
info["Vectors"]=Vectors;
info["VectorsIndices"]=VectorsIndices;
info["Amplitudes"]=Amplitudes;
```

```
145         info
146 ]
```

We can now use this information in order to construct amplitude expansions of the free energy of various systems.

Any reciprocal lattice vector in N dimensions can be written in terms of the N primitive reciprocal lattice vectors, $\mathbf{q_i}$, by:

$$\mathbf{q} = \sum_{i=1}^{N} a_i \mathbf{q_i}, \tag{A.6}$$

where the integers a_i are the *vector indices*. The set $\{a_i\}$ is unique for a given vector. To simplify what follows, we work with $N = 3$ and expand the vector as

$$\mathbf{q} = h\mathbf{q_1} + k\mathbf{q_2} + l\mathbf{q_3}. \tag{A.7}$$

A function periodic on a Bravais lattice can be expressed as a Fourier series through the expansion:

$$n(\mathbf{r}) = \sum_{\mathbf{q}} \phi_{\mathbf{q}} e^{i\mathbf{q}\cdot\mathbf{r}}. \tag{A.8}$$

Any function f of n will also be expressible as a Fourier series with its own set of amplitudes $\phi^f_{\mathbf{q}}$. Our goal is to compute the amplitudes $\phi^f_{\mathbf{q}}$ in terms of the amplitudes of n, $\phi_{\mathbf{q}}$. Of particular interest are the zero order amplitudes, $\phi_{\mathbf{0}}$, corresponding to the system average.

If we insert Eq. A.7 into Eq. A.8, then we obtain:

$$n(\mathbf{r}) = \sum_{h,k,l} \phi_{hkl} e^{ih\mathbf{q_1}\cdot\mathbf{r}} e^{ik\mathbf{q_2}\cdot\mathbf{r}} e^{il\mathbf{q_3}\cdot\mathbf{r}} \tag{A.9}$$

$$= \sum_{h,k,l} \phi_{hkl} (e^{i\mathbf{q_1}\cdot\mathbf{r}})^h (e^{i\mathbf{q_2}\cdot\mathbf{r}})^k (e^{i\mathbf{q_3}\cdot\mathbf{r}})^l. \tag{A.10}$$

For simplicity, define $e^{i\mathbf{q_i}\cdot\mathbf{r}} = X_i$, yielding

$$n(\mathbf{r}) = \sum_{h,k,l} \phi_{hkl} X_1^h X_2^k X_3^l. \tag{A.11}$$

In other words, amplitude expansions can be mapped onto polynomials. This allows us to use Mathematica's inbuilt polynomial manipulation to compute amplitude expansion coefficients efficiently.

We will be focusing on polynomial functions of n. For example, consider $f[n] = n^2$:

$$f[n] = n^2(\mathbf{r}) = \left(\sum_{h,k,l} \phi_{hkl} X_1^h X_2^k X_3^l \right)^2 \tag{A.12}$$

$$= \sum_{\substack{h_1,k_1,l_1 \\ h_2,k_2,l_2}} \phi_{h_1k_1l_1} \phi_{h_2k_2l_2} X_1^{h_1+h_2} X_2^{k_1+k_2} X_3^{l_1+l_2}. \tag{A.13}$$

Expanding f as

$$f(\mathbf{r}) = \sum_{h,k,l} \phi^f_{hkl} X_1^h X_2^k X_3^l, \tag{A.14}$$

we can see the amplitudes ϕ^f_{hkl} correspond to the coefficients of the terms of the same powers in Eq. A.13.

We can now translate this approach into Mathematica code. We are only interested in the ϕ_{000} term. There is one important caveat to the above: Mathematica manipulates polynomials most efficiently with terms of powers that are positive. Therefore, we will multiply both sides of the above by $X_1^{-m}X_2^{-m}X_3^{-m}$, where m is the smallest of the values of h, k, l (m is referred to as `minIndex` below). This will ensure all powers are greater than or equal to zero.

```
ExpandTermsFast[vectorsIndices_,ampsList_]:=Module[
        {dim,minIndex,BuildTerm,BuildExpr,exprList, exprExpanded,indexList,n},
        n=Length[ampsList];
        dim=Length[First[vectorsIndices]];
        minIndex=Min[Flatten[vectorsIndices]];
        BuildTerm[hkl_]:=Product[X[i]^(hkl[[i]]-minIndex),{i,1,dim}];
        BuildExpr[amps_]:=Total[
                MapThread[Times,{amps,Map[BuildTerm,vectorsIndices]}]
        ];
        exprList=Map[BuildExpr,ampsList];
        exprExpanded=Apply[Times,exprList];
        indexList=ConstantArray[1-minIndex n ,dim];
        Extract[CoefficientList[exprExpanded,Table[X[i],{i,1,dim}]],indexList]
]
```

We can use this subroutine as follows:

```
TriPrimitiveVectors = {{1, 0}, {1/2, Sqrt[3]/2}};
TriLatticeInfo = GenerateLatticeInfo[HexPrimitiveVectors, {ϕ[0], ϕ[1], ϕ[2]}];
Amps = TriLatticeInfo["Amplitudes"];
VecIndices = TriLatticeInfo["VectorsIndices"];
ExpandTermsFast[VecIndices, {Amps, Amps}]
>>> ϕ[0]^2 + 6 ϕ[1]^2 + 6 ϕ[2]^2
ExpandTermsFast[VecIndices, {Amps, Amps, Amps}]
>>> ϕ[0]^3 + 18 ϕ[0] ϕ[1]^2 + 12 ϕ[1]^3 + 36 ϕ[1]^2 ϕ[2] + 18 ϕ[0] ϕ[2]^2
    + 12 ϕ[2]^3
```

In the first instance we expanded n^2 to obtain $\phi_0^2+6\phi_1^2+6\phi_2^2$. In the second instance we expanded n^3 to obtain $\phi_0^3+18\phi_0\phi_1^2+12\phi_1^3+36\phi_1^2\phi_2+18\phi_0\phi_2^2+12\phi_2^3$. Note that

since our function takes lists of amplitudes (rather than specific powers) it allows us to mix amplitude lists (so long as they correspond to the same expansion). This allows us to obtain expansions of products of different variables. This will be especially helpful in obtaining expansions of correlation functions, described below.

We commonly wish to expand the standard PFC ideal free energy term, $n^2/2 - \nu n^3/6 + \xi n^4/4$.

```
IdealFreeEnergy[latticeInfo_,ν_,ξ_]:=Module[
        {amps = latticeInfo["Amplitudes"],
         vecs = latticeInfo["VectorsIndices"]},
        Expand[FullSimplify[
                Expand[ExpandTermsFast[vecs,{amps, amps}, 2]/2
                -ν ExpandTermsFast[vecs,{amps, amps, amps}, 3]/6
                +ξ ExpandTermsFast[vecs,{amps, amps, amps, amps}, 4]/12]
        ]]
];
```

We can use this as follows:

```
IdealFreeEnergyTri = IdealFreeEnergy[TriLatticeInfo, 1, 1]
>>> ϕ[0]^2/2 - ϕ[0]^3/6 + ϕ[0]^4/12 + 3 ϕ[1]^2 - 3 ϕ[0] ϕ[1]^2 + 3 ϕ[0]^2 ϕ[1]^2
    - 2 ϕ[1]^3 + 4 ϕ[0] ϕ[1]^3 + (15 ϕ[1]^4)/2 - 6 ϕ[1]^2 ϕ[2]
    + 12 ϕ[0] ϕ[1]^2 ϕ[2] + 12 ϕ[1]^3 ϕ[2] + 3 ϕ[2]^2 - 3 ϕ[0] ϕ[2]^2
    + 3 ϕ[0]^2 ϕ[2]^2 + 30 ϕ[1]^2 ϕ[2]^2 - 2 ϕ[2]^3 + 4 ϕ[0] ϕ[2]^3
    + (15 ϕ[2]^4)/2
```

Say we have a function `C2[k]` which gives the two point correlation as a function of `k`. For example, a very simple correlation function based on XPFC would be (where σ is a temperature parameter)

```
C2[K_] := Exp[-σ] Exp[-(K - 4 \pi/Sqrt[3])^2];
```

By the convolution theorem, we can calculate the amplitudes of the convolution of C_2 with n by multiplying each amplitude $\phi_{\mathbf{q}}$ by $C_2(\mathbf{q})$, where $\mathbf{q}$ is the recip-

rocal lattice vector associated with that amplitude. For rotationally symmetric correlation functions, we can do this in Mathematica in the following manner:

```
C2n = MapThread[
   Function[{amp, vec}, C2[Norm[vec]] amp],
   {TriLatticeInfo["Amplitudes"], TriLatticeInfo["Vectors"]}
];
```

`C2n` is an amplitude list. From this we can calculate the full amplitude expansion of the two-point excess term by expanding the product of these amplitudes with the density amplitudes:

```
ExcessFreeEnergyTri = -(1/2) ExpandTermsFast[TriLatticeInfo["VectorsIndices"],
    {TriLatticeInfo["Amplitudes"], C2n}]
>>> (-(ϕ[0]^2 Exp[(16 π^2)/3 - σ]) - (6 ϕ[1]^2) Exp[-σ]
     - (6 ϕ[2]^2) Exp[(4 π - (4 π)/Sqrt[3])^2 - σ])/2
```

We now have the amplitude expansion of the ideal and excess free energies of this simple system. The next step is to minimize the free energy with respect to the amplitudes. To do this we will create a function `CreateAmplitudesMinimizer`. We will define this function below, first we will show how to use it:

```
AmpsMinimizer = CreateAmplitudesMinimizer[IdealFreeEnergyTri + ExcessFreeEnergyTri,
                  {ϕ[1], ϕ[2]}, {ϕ[0], σ}];
```

The first argument is the free energy. The second argument is the list of amplitudes which the free energy will be minimized with respect to. The third argument is the list of system parameters, in this case the average density ϕ`[0]` and the temperature parameter σ. We can use the `AmpsMinimizer` by giving it a list of the values of the system parameters, specified in the same order:

```
AmpsMinimizer[{.3, .05}]
>>> {{0.22452, -0.00190909}, 0.0267824}
```

The output is a list of the amplitudes (so $\phi_1 = 0.22452$ and $\phi_2 = -0.00190909$) and the minimized free energy (0.0267824). At a higher temperature, the amplitudes are zero and the system is in the disordered state:

```
AmpsMinimizer[{.3, .3}]
>>> {{0, 0}, 0.041175}
```

We now define the `CreateAmplitudesMinimizer` function, using the built-in Mathematica function **`NMinimize`** to numerically minimize the free energy:

```
CreateAmplitudesMinimizer[freeEnergy_,amplitudeList_,params_]:=Function[{paramVals},
        Module[{ParamMap,F,Result},
                ParamMap=MapThread[#1->#2&,{params,paramVals}];
                F=freeEnergy/.ParamMap;
                Result=Last[NMinimize[F,amplitudeList,Method->"RandomSearch"]];
                {Chop[amplitudeList/.Result],F/.Result}
        ]
];
```

With these utilities we can find free energies over a range of values and then plot phase diagrams.

Appendix B

Notes on Fourier Transforms in Polar Coordinates

This appendix shows the mathematical steps required to write Fourier transforms in polar coordinates. Note that we use the following convention for the Fourier transform:

$$\mathcal{F}\{f(\mathbf{r})\} \equiv \hat{f}(\mathbf{k}) \equiv \int f(\mathbf{r}) e^{i\mathbf{k}\cdot\mathbf{r}}\, d\mathbf{r} \tag{B.1}$$

To begin, we rewrite Eq. B.1 in polar form:

$$\hat{f}(k,\theta_k) = \int_{r=0}^{\infty}\int_{\theta=0}^{2\pi} f(r,\theta) e^{ikr\cos(\theta-\theta_k)} r\, d\theta\, dr \tag{B.2}$$

Expressing f as a multipole expansion

$$f(r,\theta) = \sum_m f_m(r) e^{im\theta} \tag{B.3}$$

and substituting, we get

$$\hat{f}(k,\theta_k) = \sum_m \int_{r=0}^{\infty} f_m(r) r \int_{\theta=0}^{2\pi} e^{im\theta} e^{ikr\cos(\theta-\theta_k)}\, d\theta\, dr \tag{B.4}$$

Since we integrate over the full range of θ we can substitute $\theta \to \theta + \theta_k$ with no loss of generality. This leaves us with:

$$\hat{f}(k,\theta_k) = \sum_m e^{im\theta_k} \int_{r=0}^{\infty} f_m(r) r \left(\int_{\theta=0}^{2\pi} e^{im\theta} e^{ikr\cos(\theta)}\, d\theta \right) dr \tag{B.5}$$

We make use of following form of the Bessel functions J_m

$$J_m(z) = \frac{i^{-m}}{\pi} \int_0^{\pi} e^{iz\cos(\theta)} \cos(m\theta)\, d\theta \tag{B.6}$$

With some manipulation this becomes:

$$J_m(z) = \frac{i^{-m}}{2\pi} \int_0^{2\pi} e^{iz\cos(\theta)} \cos(m\theta)\, d\theta \tag{B.7}$$

$$= \frac{i^{-m}}{2\pi} \int_0^{2\pi} e^{iz\cos(\theta)} \frac{1}{2}(e^{im\theta} + e^{-im\theta})\, d\theta \tag{B.8}$$

$$= \frac{i^{-m}}{2\pi} \int_0^{2\pi} e^{iz\cos(\theta)} e^{im\theta}\, d\theta \tag{B.9}$$

The term in parenthesis in Eq. B.5 can therefore be replaced by $2\pi i^m J_m(kr)$:

$$\hat{f}(k,\theta_k) = 2\pi \sum_m i^m e^{im\theta_k} \int_0^{\infty} f_m(r) r J_m(kr)\, dr \tag{B.10}$$

Eq. B.10 gives the Fourier transform of a two dimensional function in polar coordinates, provided f is expressed in the form of Eq. B.3.

We will be interested particularly in the transforms of the Dirac delta and circ functions. For the Dirac delta we set $f_0(r) = \delta(r - a_0)$:

$$\mathcal{F}\big\{\delta(r-a_0)\big\} = 2\pi \int_0^\infty \delta(r-a_0) r J_0(kr)\, dr \tag{B.11}$$

$$= 2\pi a_0 J_0(ka_0) \tag{B.12}$$

For circ we set $f_0(r) = \mathrm{circ}(r/r_0)$:

$$\mathcal{F}\left\{\mathrm{circ}\left(\frac{r}{r_0}\right)\right\} = 2\pi \int_0^\infty \mathrm{circ}\left(\frac{r}{r_0}\right) r J_0(kr)\, dr \tag{B.13}$$

$$= 2\pi \int_0^{r_0} r J_0(kr)\, dr \tag{B.14}$$

using the coordinate transform $r' = kr$, $dr' = kdr$ this becomes

$$= \frac{2\pi}{k^2} \int_0^{kr_0} r' J_0(r')\, dr' \tag{B.15}$$

Finally, noting the identity:

$$\int_0^a x J_0(x)\, dx = a J_1(a) \tag{B.16}$$

we get

$$\mathcal{F}\left\{\mathrm{circ}\left(\frac{r}{r_0}\right)\right\} = \frac{2\pi r_0}{k} J_1(r_0 k) \tag{B.17}$$

Lastly, for the Gaussian centered on the origin:

$$\mathcal{F}\left\{\left(\frac{\alpha}{\pi}\right) e^{-\alpha r^2}\right\} = e^{-k^2/4\alpha} \tag{B.18}$$

Appendix C

Numerical Approach for Computing Convex Hulls of Free Energy Data

In determining the phase diagram for a system of two components we usually begin by numerically approximating the free energy for a finite set of density values. That is, we obtain a set F of three dimensional points $\{n_A, n_B, f(n_A, n_B)\}$. The points n_A, n_B for which we obtain the free energy are usually spaced over a regular discrete grid covering the range of values of interest. In determining coexistence regions for such systems we need to use this set of points to approximate the*convex hull* of the free energy function.

First, some definitions. A set is *convex* if every line between every pair of points in the set consists solely of points in that set. The *convex hull* of a set of points is the smallest convex set which contains all the points. In three dimensions, the convex hull of a set of discrete points will be a polyhedron.

The set of all points above the graph of a function is termed its *epigraph*. A *convex function* is a function with the property that its epigraph is a convex set.

Note that the convex hull of a function has no upper bound.

We are interested in obtaining the convex hull of the free energy function, which we have approximated by the set of points F. We can find an approximation of the convex hull of the free energy function by finding only the *lower bound* of the convex hull of F (we don't need the upper bound, since the epigraph of f has no upper bound). The convex hull we obtain will be a triangular mesh over the domain of n_A, n_B with the property that any line joining any two points of F will lie on or above the mesh.

In the sections that follow, for simplicity we will use the phase "convex hull" to refer to the lower bound of the convex hull. A point is said to be "included in" or "inside" a hull if it lies above above the hull.

C.1 Description of convex hull construction algorithm

I developed for this thesis a new algorithm for constructing the convex hull of any free energy density of the form $f(n_A, n_B)$ that satisfies the properties described in the text. This algorithm is based on the QuickHull approach and is shown here in its full C++ implementation. Access to the code can be obtained by contacting the author.

The input will consist of a set of values over a regular two dimensional grid.

We start with an initial convex hull that may not include all the points. We will expand this hull iteratively until all points are included. At all stages the hull consists of a set of triangles that form a mesh covering the entire domain of the original points.

For each point p we consider if it is inside the hull. If it is we may discard it and

move on. If the point p is not in the hull then the hull must be extended to include the point. We find all faces of the hull visible to the point. The boundary of these faces (consisting of a set of edges forming a polygon) is termed the "horizon" h. We remove these visible faces. Then, for each edge e in the horizon h, we add a new triangular face constructed from e and p.

C.2 Complete C++ implementation

This subsection provides a complete implementation of the algorithm, written in C++. To invoke this program, the input data must be a text file containing a comma separated list of the data points, in column major order. The program is invoked as:

```
convex [columns] [rows] [file for data values] [output prefix] [xmin] [ymin] [xstep] [ystep]
```

Where `[columns]` and `[rows]` denote the number of columns and rows of the input data. If the number of data points provided in the data file doesn't match an error will be displaced. `[file for data values]` is the name of the input data file, described above. `[output prefix]` is a prefix for the generated output files. Four output files will be generated, each with this prefix. These are: `"prefix_hull_tris.txt"` which lists all the 3d triangles making up the convex hull; `"prefix_on_hull.txt"` which outputs a crude text representation of which points of the input data are or are not on the resulting convex hull; `"prefix_coexist_region.txt"` which outputs a set of lines outlining the coexistence region; and `"prefix_coexist_lines.txt"` which outputs the tie lines connecting points of coexistence. `[xmin]` and `[ymin]` are the x and y

coordinates of the first data point. Lastly, `[xstep]` and `[ystep]` indicate the spacing between the data points.

We now provide the implication, with commentary. We start with the included libraries:

```
#include <fstream>
#include <iostream>
#include <sstream>
#include <string>
#include <algorithm>
#include <functional>
#include <vector>
#include <set>
#include <stdlib.h>
#include <time.h>
#include <tuple>
#include <deque>

using namespace std;
```

Define some basic geometry classes:

```
class point {
public:
    double x, y, z;
    point(double x, double y, double z) : x(x), y(y), z(z) {};
};

class gridPoint {
public:
    int x, y;
    double z;
    gridPoint(int x, int y, double z) : x(x), y(y), z(z) {};
    operator point() {return point(x,y,z);};
    bool operator ==(gridPoint p) {return x == p.x && y == p.y && z == p.z;};
};

class edge {
```

```
31 public:
32     gridPoint p1, p2;
33     tuple<int,int,int,int> sortedXY;
34     edge(gridPoint p1, gridPoint p2) : p1(p1), p2(p2) {
35         //Implement so that edges can be sorted/tested for equality. We want
               edges to be considered equal regardless of the order
36         //the endpoints are specified, hence getSortedXY, which orders the
               endpoints into a tuple. z-values are ignored.
37
38         //Sort endpoints into a tuple
39         tuple<int,int> t1(p1.x, p1.y), t2(p2.x, p2.y);
40
41         //sort the tuples by which is "lower"
42         tuple<int,int> tLower, tUpper;
43         if(t1 > t2) {
44             tLower = t2;
45             tUpper = t1;
46         } else {
47             tLower = t1;
48             tUpper = t2;
49         }
50
51         sortedXY = tuple<int,int,int,int>(get<0>(tLower), get<1>(tLower), get<0>(
               tUpper), get<1>(tUpper));
52     };
53
54 };
55
56 class triangle {
57 public:
58     gridPoint p1, p2, p3;
59     triangle(gridPoint p1, gridPoint p2, gridPoint p3) : p1(p1), p2(p2), p3(p3)
           {};
60     triangle(gridPoint p, edge e) : p1(p), p2(e.p1), p3(e.p2) {};
61 };
```

We need to be able to sort and compare instances of `edge` such that the order in which points are specified does not matter:

```
62 //For use with std::sort. Equivalent edges will be sorted next to each other
63 bool sortEdges (edge lhs, edge rhs) {
64     return lhs.sortedXY < rhs.sortedXY;//leftTuple < rightTuple;
65 }
66
67 //Equality check
68 bool operator==(const edge &lhs, const edge &rhs) {
69     return lhs.sortedXY == rhs.sortedXY;
70 }
```

Basic geometry operations:

```
71 double dotProduct(point v1, point v2) {
72     return v1.x * v2.x + v1.y * v2.y + v1.z * v2.z;
73 }
74
75 double dotProduct2d(point v1, point v2) {
76     return v1.x * v2.x + v1.y * v2.y;
77 }
78
79 point crossProduct(point v1, point v2) {
80     return point(v1.y * v2.z - v1.z * v2.y, v1.z * v2.x - v1.x * v2.z, v1.x * v2.
          y - v1.y * v2.x);
81 }
82
83 gridPoint subtract(gridPoint p1, gridPoint p2) {
84     return gridPoint(p1.x - p2.x, p1.y - p2.y, p1.z - p2.z);
85 }
86
87 point subtract(point p1, point p2) {
88     return point(p1.x - p2.x, p1.y - p2.y, p1.z - p2.z);
89 }
```

Some geometry predicates:

```
90 //is the 2d projection of the triangle onto the xy plane degenerate (a line or
       point)?
91 bool isDegenerate2d(triangle tri) {
92     gridPoint v1 = subtract(tri.p2, tri.p1);
```

```
93      gridPoint v2 = subtract(tri.p3, tri.p1);
94      return (v1.x * v2.y - v1.y * v2.x == 0);
95  }
96
97  point planeNormalUp(triangle plane) {
98      point norm = crossProduct( subtract(plane.p2, plane.p1), subtract(plane.p3,
            plane.p1));
99      if(norm.z >= 0) {
100         return norm;
101     } else {
102         return subtract(point(0,0,0), norm);
103     }
104 }
105
106 //Above or on
107 bool pointAbovePlane(gridPoint p, triangle plane) {
108     if(p == plane.p1 || p == plane.p2 || p == plane.p3) {
109         return true;
110     }
111
112     point normal = planeNormalUp(plane);
113
114     point v = subtract(p, plane.p1);
115
116     double dot = dotProduct(normal, v);
117
118     return dot >= 0;
119 }
120
121 bool triangleNotVisible(gridPoint p, triangle tri) {
122     point normal = planeNormalUp(tri);
123     point vectorToPlane = subtract(tri.p1, p);
124     double dot = dotProduct(normal, vectorToPlane);
125     return dot <= 0;
126 }
127
128 bool pointAboveHull(gridPoint point, vector<triangle> hull) {
129     //assuming the hull is convex, if the point is below any plane it is below
```

```
        the hull:
    for(auto h = hull.begin(); h != hull.end(); h++) {
        if(!pointAbovePlane(point, *h)) {
            return false;
        }
    }
    return true;
}
```

We can now implement the basic algorithm iteration.

```
//removes the found triangles from the hull
vector<triangle> extractVisibleTriangles(gridPoint p, vector<triangle> &hull) {
    //partition the list so that the visible are at the end and the not visible
        are at the beginning
    //(it's computationally cheaper to remove from the end of the list):
    vector<triangle>::iterator bound = partition (hull.begin(), hull.end(), bind(
        triangleNotVisible,p,std::placeholders::_1));
    //copy out visible to be returned:
    vector<triangle> visible(bound, hull.end());
    //remove visible from the hull:
    hull.erase(bound, hull.end());

    return visible;
}

vector<edge> removeDuplicateEdges(vector<edge> edges) {
    //To find the duplicates we sort the list of edges so that
    //equal edges will be next to each other in the list.

    //sort them
    sort(edges.begin(), edges.end(), sortEdges);

    //copy out non-duplicates:
    vector<edge> edgesOut;
    unsigned int i = 0;
    while(i + 1 < edges.size()) { //i < size - 1, but in case size is zero (it's
        unsigned) we use i + 1 < size
        if(edges[i] == edges[i + 1]) {  //it's a duplicate
```

```
162             i += 2;
163         } else { //it's not a duplicate, copy it out:
164             edgesOut.push_back(edges[i]);
165             i++;
166         }
167     }
168 
169     //if there's still one entry left, it isn't a duplicate:
170     if(i + 1 == edges.size()) {
171         edgesOut.push_back(edges[i]);
172     }
173 
174     return edgesOut;
175 }
176 
177 vector<edge> getEdgesOfTriangles(vector<triangle> group) {
178     //We only want the edges forming the boundray of the triangle group.
179     //All interior edges will show up twice, so if we take all edges and
180     //remove the duplicates we will be left with the boundary.
181 
182     //find all the edges in the group:
183     vector<edge> edges;
184     for(auto tri = group.begin(); tri != group.end(); tri++) {
185         edges.push_back(edge((*tri).p1, (*tri).p2));
186         edges.push_back(edge((*tri).p2, (*tri).p3));
187         edges.push_back(edge((*tri).p3, (*tri).p1));
188     }
189 
190     return removeDuplicateEdges(edges);
191 }
192 
193 void iterate(vector<gridPoint> &points, vector<triangle> &hull) {
194     //select a point at random
195     int pointToRemoveIndex = rand() % points.size();
196     gridPoint thePoint = points[pointToRemoveIndex];
197 
198     //remove it from points:
199     points[pointToRemoveIndex] = points.back();
```

```
200     points.pop_back();
201
202     if(pointAboveHull(thePoint, hull)) {
203         //if it's above the hull, do nothing
204         return;
205     } else {
206         //otherwise we must extend the hull to include the point:
207
208         //find and remove all triangles in the hull visible to the point
209         vector<triangle> visibleTriangles = extractVisibleTriangles(thePoint,
                hull);
210
211         //find the edges of the visible triangles
212         vector<edge> edges = getEdgesOfTriangles(visibleTriangles);
213
214         //add new triangles from the point to the edges
215         for(auto edge = edges.begin(); edge != edges.end(); ++edge) {
216             triangle newTri(thePoint, *edge);
217             //remove denerate (vertically oriented) triangles:
218             if(!isDegenerate2d(newTri)) {
219                 hull.push_back(newTri);
220             }
221         }
222     }
223 }
```

Using `iterate` we can now find the convex hull from the initial data set:

```
224 vector<triangle> getInitialHull(vector<gridPoint> &points, gridPoint topLeft,
        gridPoint topRight, gridPoint bottomLeft, gridPoint bottomRight) {
225     //we take the four corners plus the minimum point in the set to generate the
            initial convex hull:
226
227     gridPoint minPoint = points[0];
228     for(auto p = points.begin(); p != points.end(); p++) {
229         if((*p).z < minPoint.z) {
230             minPoint = *p;
231         }
232     }
```

```
233
234     //remove the corners from the points list:
235     points.erase(std::remove(points.begin(), points.end(), topLeft), points.end()
            );
236     points.erase(std::remove(points.begin(), points.end(), topRight), points.end
            ());
237     points.erase(std::remove(points.begin(), points.end(), bottomLeft), points.
            end());
238     points.erase(std::remove(points.begin(), points.end(), bottomRight), points.
            end());
239
240     //remove the minPoint
241     points.erase(std::remove(points.begin(), points.end(), minPoint), points.end
            ());
242
243     triangle tri1 = triangle(topLeft, topRight, minPoint);
244     triangle tri2 = triangle(topRight, bottomRight, minPoint);
245     triangle tri3 = triangle(bottomRight, bottomLeft, minPoint);
246     triangle tri4 = triangle(bottomLeft, topLeft, minPoint);
247
248     vector<triangle> hull;
249     //remove denerate (vertically oriented) triangles:
250     if(!isDegenerate2d(tri1)){ hull.push_back(tri1); };
251     if(!isDegenerate2d(tri2)){ hull.push_back(tri2); };
252     if(!isDegenerate2d(tri3)){ hull.push_back(tri3); };
253     if(!isDegenerate2d(tri4)){ hull.push_back(tri4); };
254
255     return hull;
256 }
257
258 vector<triangle> generateConvexHull(vector<gridPoint> points, gridPoint topLeft,
        gridPoint topRight, gridPoint bottomLeft, gridPoint bottomRight) {
259     vector<triangle> hull = getInitialHull(points, topLeft, topRight, bottomLeft,
            bottomRight);
260
261     while(points.size() > 0) {
262         cout << "points␣remaining␣" << points.size() << "\n";
263         iterate(points, hull);
```

```
264     }
265
266     return hull;
267 }
268
269 vector<triangle> generateConvexHullFromData(int width, int height, vector<double>
        data) {
270     vector<gridPoint> points;
271     int dataIndex = 0;
272
273     //column major order
274     for(int i = 0; i < width; i++) {
275         for(int j = 0; j < height; j++) {
276             points.push_back(gridPoint(i, j, data[dataIndex]));
277             dataIndex++;
278         }
279     }
280
281     vector<triangle> hull = generateConvexHull(points, points[0], points[height -
            1], points[height * (width - 1)], points[width * height - 1]);
282     return hull;
283 }
```

With the convex hull we can extract some relevant data. `getCoexistLine` is used to generate the tie lines seen in the phase diagrams in Chapter 4. While the algorithm generates a hull consisting of three dimensional triangles, in practice the systems we consider demonstrate two-point coexistence over most of the coexistence region. The finite resolution of the input data means that these regions of two-point coexistence will be spanned by long, thin triangles in the generated convex hull. The `getCoexistLine` function makes the simplifying assumption that the triangles generated can be replaced by lines. It does so by replacing the shortest edge in each triangle with its midpoint.

```
284 //Sets the boolean array isOnHull to true or false for each point.
285 //A point is on the hull if it is the endpoint of one of the hull triangles.
```

```
286 void setIsOnHull(bool* isOnHull, int width, int height, vector<triangle> hull) {
287     for(int i = 0; i < width * height; i++) {
288         isOnHull[i] = false;
289     }
290 
291     for(auto tri = hull.begin(); tri != hull.end(); tri++) {
292         triangle t = *tri;
293         isOnHull[t.p1.y * width + t.p1.x] = true;
294         isOnHull[t.p2.y * width + t.p2.x] = true;
295         isOnHull[t.p3.y * width + t.p3.x] = true;
296     }
297 }
298 
299 //Determines if a point is on the edge of the domain
300 bool notOnEdge(gridPoint p, int width, int height) {
301     if(p.x == 0 || p.y == 0 || p.x == width - 1 || p.y == height - 1) {
302         return false;
303     } else {
304         return true;
305     }
306 }
307 
308 //Removes edges that lie entirely on the edge of the domain
309 vector<edge> removeAreaEdges(vector<edge> edges, int width, int height) {
310     vector<edge> edgesOut;
311     for(auto e = edges.begin(); e != edges.end(); e++) {
312         if(notOnEdge((*e).p1, width, height) || notOnEdge((*e).p2, width, height)
                ) {
313             edgesOut.push_back(*e);
314         }
315     }
316     return edgesOut;
317 }
318 
319 //"large" triangles are those of area greater than 1/2. The convex areas of the
        input
320 //values will be tiled by "small" triangles of area 1/2. Thus the "large"
        triangles
```

```
321 //outline the areas of coexistence.
322 vector<triangle> getLargeTriangles(vector<triangle> tris) {
323     vector<triangle> outTris;
324     for(auto t = tris.begin(); t != tris.end(); t++) {
325         triangle tri = *t;
326         gridPoint u = subtract(tri.p3, tri.p1);
327         gridPoint v = subtract(tri.p2, tri.p1);
328         int twiceArea2d = abs(u.x * v.y - u.y * v.x);
329         if(twiceArea2d != 1) {
330             outTris.push_back(tri);
331         }
332     }
333     return outTris;
334 }
335
336 //Coexistence lines are determined by assuming the triangles are relatively long.
337 //We take the shortest edge and find its midpoint, and then return an edge from
        that
338 //midpoint to the point opposite the short side.
339 tuple<double, double, double, double> getCoexistLine(triangle tri) {
340     //vectors for each edge:
341     gridPoint u = subtract(tri.p3, tri.p1);
342     gridPoint v = subtract(tri.p2, tri.p1);
343     gridPoint w = subtract(tri.p3, tri.p2);
344
345     //norm^2 for each edge:
346     int uu = dotProduct2d(u, u);
347     int vv = dotProduct2d(v, v);
348     int ww = dotProduct2d(w, w);
349
350     //find the edge:
351     if(uu <= vv && uu <= ww) { //u is the short edge
352         return tuple<double, double, double, double>((tri.p3.x + tri.p1.x) / 2.0,
                (tri.p3.y + tri.p1.y) / 2.0, tri.p2.x, tri.p2.y);
353     }
354     if(vv <= uu && vv <= ww) { //v is the short edge
355         return tuple<double, double, double, double>((tri.p2.x + tri.p1.x) / 2.0,
                (tri.p2.y + tri.p1.y) / 2.0, tri.p3.x, tri.p3.y);
```

```
356     }
357     if(ww <= uu && ww <= vv) { //w is the short edge
358         return tuple<double, double, double, double>((tri.p3.x + tri.p2.x) / 2.0,
                (tri.p3.y + tri.p2.y) / 2.0, tri.p1.x, tri.p1.y);
359     }
360
361     //should never get here
362     throw "No shortest edge";
363 }
364
365 //Connects a set of edges into paths. Paths are sequences of points, an edge is a
        path of length 2.
366 //Sequences of edges sharing common endpoints are converted into equivalent paths
        .
367 vector<vector<gridPoint>> connectEdges(vector<edge> edges) {
368     deque<vector<gridPoint>> paths;
369
370     //Push all edges into the list of paths:
371     for(auto e = edges.begin(); e != edges.end(); e++) {
372         vector<gridPoint> path;
373         path.push_back((*e).p1);
374         path.push_back((*e).p2);
375         paths.push_back(path);
376     }
377
378     //It will take at most this many iterations to connect all paths:
379     int iterations = paths.size();
380
381     while(iterations >= 0) {
382         iterations--;
383
384         //We take the first path, then search for any paths that connect to it at
                the front or back.
385         //If we find one we merge the paths into one.
386         //The resulting path (whether or not it was merged with another one) is
                pushed to the back of
387         //the queue.
388
```

```
389         vector<gridPoint> path = paths[0];
390         paths.pop_front();
391 
392         //search for connections and merge:
393         for(int i = 0; i < (int)(paths.size()); i++) {
394             gridPoint first = path[0];
395             gridPoint last  = path.back();
396 
397             //front to back
398             if(paths[i][0] == last) {
399                 auto foundPath = paths[i];
400 
401                 //copy found path to end of path
402                 path.insert(path.end(), foundPath.begin() + 1, foundPath.end());
403                 //remove found path
404                 paths.erase(paths.begin() + i);
405                 i--;
406                 continue;
407             }
408 
409             //back to front
410             if(paths[i].back() == first) {
411                 auto foundPath = paths[i];
412 
413                 //copy path to end of found path:
414                 foundPath.insert(foundPath.end(), path.begin() + 1, path.end());
415                 //this is now our path:
416                 path = foundPath;
417                 //remove the found path from the list:
418                 paths.erase(paths.begin() + i);
419                 i--;
420                 continue;
421             }
422 
423             //front to front
424             if(paths[i][0] == first) {
425                 auto foundPath = paths[i];
426 
```

```
427                 //reverse the path:
428                 reverse(path.begin(), path.end());
429                 //now append the found path to it:
430                 path.insert(path.end(), foundPath.begin() + 1, foundPath.end());
431                 //remove found path
432                 paths.erase(paths.begin() + i);
433                 i--;
434                 continue;
435             }
436 
437             //back to back
438             if(paths[i].back() == last) {
439                 auto foundPath = paths[i];
440 
441                 //reverse the found path:
442                 reverse(foundPath.begin(), foundPath.end());
443                 //now append the found path to path:
444                 path.insert(path.end(), foundPath.begin() + 1, foundPath.end());
445                 //remove found path
446                 paths.erase(paths.begin() + i);
447                 i--;
448                 continue;
449             }
450         }
451 
452         paths.push_back(path);
453     }
454 
455     return vector<vector<gridPoint>>(paths.begin(), paths.end());
456 }
```

Generating and outputting info from the hull:

```
457 //Writes out a grid of 1s and 0s indicating which points in the input were on the
        resulting
458 //convect hull. (On hull is "1", not on hull is "0").
459 void writeOnHull(string filename, vector<triangle> hull, int width, int height) {
460     ofstream outfile( filename.c_str(), ofstream::out );
461 
```

```
462     bool onHull[width * height];
463 
464     setIsOnHull(onHull, width, height, hull);
465 
466     int idx = 0;
467     for(int j = 0; j < height; j++) {
468         for(int i = 0; i < width; i++) {
469             outfile << (onHull[idx] ? "1" : "0");
470             idx++;
471         }
472         outfile << "\n";
473     }
474 
475 }
476 
477 //Outputs all triangles.
478 void writeTris(string filename, vector<triangle> tris, double xMin, double yMin,
        double xStep, double yStep) {
479     ofstream outfile( filename.c_str(), ofstream::out );
480 
481     for(unsigned int i = 0; i < tris.size(); i++) {
482         auto tri = tris[i];
483         outfile << "{{"  << tri.p1.x * xStep + xMin << "," << tri.p1.y * yStep +
                yMin << "," << tri.p1.z
484                 << "},{" << tri.p2.x * xStep + xMin << "," << tri.p2.y * yStep +
                        yMin << "," << tri.p2.z
485                 << "},{" << tri.p3.x * xStep + xMin << "," << tri.p3.y * yStep +
                        yMin << "," << tri.p3.z << "}}";
486 
487         if(i + 1 < tris.size()) {
488             outfile << ",\n";
489         }
490 
491     }
492 }
493 
494 //Outputs the coexisting lines (as determined by getCoexistLine) of the given
        convex hull.
```

```
495 void writeCoexistLines(string filename, vector<triangle> hull, double xMin,
        double yMin, double xStep, double yStep) {
496     auto largeTris = getLargeTriangles(hull);
497 
498     vector<tuple<double, double, double, double>> edges;
499     for(auto tri = largeTris.begin(); tri != largeTris.end(); tri++) {
500         edges.push_back(getCoexistLine(*tri));
501     }
502 
503     ofstream outfile( filename.c_str(), ofstream::out );
504 
505     for(unsigned int i = 0; i < edges.size(); i++) {
506         outfile << "{{"
507             << get<0>(edges[i]) * xStep + xMin << ","
508             << get<1>(edges[i]) * yStep + yMin << "},{"
509             << get<2>(edges[i]) * xStep + xMin << ","
510             << get<3>(edges[i]) * yStep + yMin << "}}";
511 
512         if(i + 1 < edges.size()) {
513             outfile << ",\n";
514         }
515     }
516 
517 }
518 
519 //Writes the paths outlining the coexistence region
520 void writeCoexistOutline(string filename, vector<triangle> hull, int width, int
        height, double xMin, double yMin, double xStep, double yStep) {
521     ofstream outfile( filename.c_str(), ofstream::out );
522 
523 
524     //Use setIsOnHull to find the areas of coexistence
525     bool onHull[width * height];
526 
527     setIsOnHull(onHull, width, height, hull);
528 
529     vector<edge> edges;
530 
```

```
531     //For each point not on the hull, add the four edges corresponding to a
            square surrounding that point:
532     int idx = 0;
533     for(int j = 0; j < height; j++) {
534         for(int i = 0; i < width; i++) {
535             if(!onHull[idx]) {
536                 //0 for the z values, it won't be printed anyway
537                 edges.push_back(edge(gridPoint(i    , j    , 0), gridPoint(i + 1,
                        j    , 0)));
538                 edges.push_back(edge(gridPoint(i    , j    , 0), gridPoint(i    ,
                        j + 1, 0)));
539                 edges.push_back(edge(gridPoint(i + 1, j    , 0), gridPoint(i + 1,
                        j + 1, 0)));
540                 edges.push_back(edge(gridPoint(i    , j + 1, 0), gridPoint(i + 1,
                        j + 1, 0)));
541             }
542             idx++;
543         }
544     }
545
546     //Remove all duplicate edges and edges on the perimeter of the domain, then
            connect them into paths:
547     auto paths = connectEdges(removeAreaEdges(removeDuplicateEdges(edges), width,
             height));
548
549     //Print the paths:
550     for(unsigned int i = 0; i < paths.size(); i++) {
551         outfile << "{";
552         for(unsigned int j = 0; j < paths[i].size(); j++) {
553             //shift the point by -.5, -.5 to account for placement of squares at
                    (i,j) to (i+1,j+1) above
554             double x = (paths[i][j].x - .5) * xStep + xMin;
555             double y = (paths[i][j].y - .5) * yStep + yMin;
556             outfile << "{" << x << "," << y << "}";
557             if(j + 1 < paths[i].size()) {
558                 outfile << ",";
559             }
560         }
```

```
561         outfile << "}";
562         if(i + 1 < paths.size()) {
563             outfile << ",";
564         }
565         outfile << "\n";
566     }
567 }
```

Some functions for reading input data:

```
568 vector<string> readFile(string filename) {
569     vector <string> data;
570     ifstream infile( filename.c_str() );
571 
572     while (infile)
573     {
574         string line;
575         if (!getline( infile, line ))
576             break;
577 
578         istringstream ss( line );
579 
580         while (ss)
581         {
582             string s;
583             if (!getline( ss, s, ',' ))
584                 break;
585             data.push_back( s );
586         }
587     }
588 
589     if (!infile.eof()) {
590         cerr << "??\n";
591     }
592 
593     return data;
594 }
595 
596 std::string trim(const std::string& str, const std::string& whitespace = " \t")
```

```
{
    const unsigned int strBegin = str.find_first_not_of(whitespace);
    if (strBegin == std::string::npos)
        return ""; // no content

    const int strEnd = str.find_last_not_of(whitespace);
    const int strRange = strEnd - strBegin + 1;

    return str.substr(strBegin, strRange);
}

template <class T> T string_convert(const std::string& s) {
        std::istringstream i(s);
        T x;
        if (!(i >> x)) {
                //return NAN;
        }
   return x;
}

vector<double> convertStringData(vector<string> dataIn) {
    vector<double> dataOut;
    for(auto datum = dataIn.begin(); datum != dataIn.end(); datum++) {
        double dataNum = string_convert<double>(trim(*datum));
        dataOut.push_back(dataNum);
    }
    return dataOut;
}
```

Finally, tying the program together:

```
void run(vector<string> args) {

    //Check that the user had provided the correct number of arguments:
    int numArgs = args.size();

    if(numArgs != 8) {
        cout << "Arguments:␣[size␣x]␣[size␣y]␣[file␣for␣data␣values]␣[output␣
            prefix]␣[xmin]␣[ymin]␣[xstep]␣[ystep]\n";
```

```
632         return;
633     }
634 
635     //Convert input arguments:
636     unsigned int width  = string_convert<unsigned int>(trim(args[0]));
637     unsigned int height = string_convert<unsigned int>(trim(args[1]));
638     string inputFileName = args[2];
639     string outPrefix = args[3];
640     double xmin  = string_convert<double>(trim(args[4]));
641     double ymin  = string_convert<double>(trim(args[5]));
642     double xstep = string_convert<double>(trim(args[6]));
643     double ystep = string_convert<double>(trim(args[7]));
644 
645     //Get the input data in string format:
646     vector<string> stringData = readFile(inputFileName);
647 
648     //Check that the data size matches the expected size from the width and
            height:
649     if(stringData.size() != width * height) {
650         cout << "Data␣size␣(" << stringData.size() << ")␣doesn't␣match␣expected␣
                size␣(" << width << "␣*␣" << height << ")\n";
651         return;
652     }
653 
654     //Convert the string data into doubles:
655     vector<double> data = convertStringData(stringData);
656 
657     //Generate the convex hull:
658     auto hull = generateConvexHullFromData(width, height, data);
659 
660     //Output data:
661     writeTris(outPrefix + "_hull_tris.txt", hull, xmin, ymin, xstep, ystep);
662 
663     writeOnHull(outPrefix + "_on_hull.txt", hull, width, height);
664 
665     writeCoexistOutline(outPrefix + "_coexist_region.txt", hull, width, height,
            xmin, ymin, xstep, ystep);
666 
```

```
667     writeCoexistLines(outPrefix + "_coexist_lines.txt", hull, xmin, ymin, xstep,
            ystep);
668 }
669
670 int main(int argc, char *argv[]) {
671     srand (time(NULL));
672
673     int numArgs = argc - 1;
674     vector<string> args;
675     if (numArgs > 0) {
676         args.assign(argv + 1, argv + argc);
677     }
678
679     run(args);
680 }
```

Appendix D

Tuning the Magnetoelastic Coupling Constants

In Chapter 2 we showed how to tune the magnetostriction constants by controlling the elastic anisotropy factor. While this is useful, it has a few drawbacks. First it can only be done by adding a second peak to the correlation function. This requires either using XPFC correlations or adding higher orders of ∇^2 to the PFC free energy. It also denies us the freedom to tune the elastic anisotropy independent of the magnetostriction should we want to do that. We might like to attempt to tune the magnetostriction constants instead by altering the ratio of B_1 to B_2. This is not as easy as it might appear. With just one term, $(\mathbf{m} \cdot \nabla n)^2$, $B_1/B_2 = 1/2$ is clearly fixed. It might appear that by including the $|\mathbf{m} \times \nabla n|^2$

term we could affect this ratio. An amplitude expansion of this term yields:

$$-\frac{\beta_{2,m}}{2}|\mathbf{m}\times\nabla n|^2 \approx -4\beta_{2,m}\phi^2\Big(\sum_i m_i^2 - 2\sum_{i<j} m_i m_j \epsilon_{ij} + \sum_i m_i^2 \sum_{i\neq j}\epsilon_{jj}\Big) \tag{D.1}$$

This last term is not included in Kittel's free energy. Modifying Equation 2.43 to include a term of this form yields:

$$\begin{aligned} F_{me} =& B_1\sum_i m_i^2\epsilon_{ii} + B_2\sum_{i<j} m_i m_j\epsilon_{ij} + B_3\sum_i m_i^2\sum_{i\neq j}\epsilon_{jj} \\ &+ \frac{1}{2}c_{11}\sum_i \epsilon_{ii}^2 + \frac{1}{2}c_{44}\sum_{i<j}\epsilon_{ij}^2 + c_{12}\sum_{i<j}\epsilon_{ii}\epsilon_{jj} \end{aligned} \tag{D.2}$$

and following the procedure outlined in the paper [73] we obtain:

$$\lambda_{100} = -\frac{2}{3}\frac{B_1 - B_3}{c_{11}-c_{12}}; \qquad \lambda_{111} = -\frac{1}{3}\frac{B_2}{c_{44}} \tag{D.3}$$

We are now interested in the ratio of $B_1 - B_3$ to B_2. With Equations D.1 and D.1 we get:

$$B_1 = -4\alpha_{2,m}\phi^2; \quad B_2 = -8(\alpha_{2,m}-\beta_{2,m})\phi^2; \quad B_3 = -4\beta_{2,m}\phi^2 \tag{D.4}$$

However this still leads to $(B_1 - B_3)/B_2 = 1/2$ and so $\lambda_{100}/\lambda_{111} = c_{44}/(c_{11}-c_{12})$.

Perhaps there could be another term that can work? Let's write $(\mathbf{m}\cdot\nabla n)^2$ in

a more suggestive way:

$$(\mathbf{m}\cdot\nabla n)^2 = (m_i\delta_{ij}(\partial_j n))^2$$
$$= m_i m_k \delta_{ij}\delta_{kl}(\partial_j n)(\partial_l n) = \sigma_{ijkl} m_i m_k (\partial_j n)(\partial_l n) \tag{D.5}$$

where δ_{ij} is the Kronecker delta and the rank 4 tensor $\sigma_{ijkl} \equiv \delta_{ij}\delta_{kl}$ has been defined. Similarly for $|\mathbf{m}\times\nabla n|^2$:

$$|\mathbf{m}\times\nabla n|^2 = (\mathbf{m}\times\nabla n)\cdot(\mathbf{m}\times\nabla n) \tag{D.6}$$
$$= (\epsilon_{mij} m_i(\partial_j n))\delta_{mn}(\epsilon_{nkl} m_k(\partial_l n)) \tag{D.7}$$
$$= \epsilon_{mij}\epsilon_{mkl} m_i m_k (\partial_j n)(\partial_l n) \tag{D.8}$$
$$= (\delta_{ik}\delta_{jl} - \delta_{il}\delta_{jk}) m_i m_k (\partial_j n)(\partial_l n) = \tau_{ijkl} m_i m_k (\partial_j n)(\partial_l n) \tag{D.9}$$

where the rank 4 tensor $\tau_{ijkl} \equiv \delta_{ik}\delta_{jl} - \delta_{il}\delta_{jk}$ has been defined. It can be seen that both terms can be written using rank 4 tensors. The most general isotropic rank 4 tensor is[113]:

$$\alpha\delta_{ij}\delta_{kl} + \beta\delta_{ik}\delta_{jl} + \gamma\delta_{il}\delta_{jk} \tag{D.10}$$

from which we can see that σ_{ijkl} corresponds to $\alpha = 1$, $\beta = \gamma = 0$ while τ_{ijkl} corresponds to $\alpha = 0$, $\beta = 1$ and $\gamma = -1$. An amplitude expansion with the form of this most general tensor (i.e. that replaces τ_{ijkl} or σ_{ijkl} by the form in Eq. (D.10) yields (ignoring any additional coefficients):

$$B_1 = 8(\alpha+\beta+\gamma)\phi^2; \quad B_2 = 16(\alpha+\gamma)\phi^2; \quad B_3 = 8\beta\phi^2; \tag{D.11}$$

and so $(B_1 - B_3)/B_2 = 1/2$ leaving us again with $\lambda_{100}/\lambda_{111} = c_{44}/(c_{11} - c_{12})$. No rank 4 tensor will do the job of allowing us to tune λ_{100} and λ_{111} via B_1 and B_2.

The next step would be to consider going to a higher order rank 6 tensor in the magnetizations and density amplitudes. If we want to keep our term to second order in $\mathbf{m}$, this amounts to going to a fourth order term in ∇n. There are 15 linearly independent rank 6 tensors[113]. Clearly we don't need this much freedom; let's try just one of them, $\delta_{ij}\delta_{kl}\delta_{pq}$, through which we can write

$$(\delta_{ij}\delta_{kl}\delta_{pq})m_i m_l(\partial_j n)(\partial_k n)(\partial_p n)(\partial_q n) = (\mathbf{m}\cdot\nabla n)^2(\nabla n\cdot\nabla n) \tag{D.12}$$

An amplitude expansion with the form of this term yields (truncating, as before, to lowest order in terms of the strain tensor):

$$\begin{aligned}\gamma_m(\mathbf{m}\cdot\nabla n)^2(\nabla n\cdot\nabla n) \approx & \gamma_m\phi^4\Bigg(76\sum_i m_i^2 + 260\sum_i m_i^2\epsilon_{ii} \\ & + 392\sum_{i<j} m_i m_j\epsilon_{ij} + 22\sum_i m_i^2\sum_{i\neq j}\epsilon_{jj}\Bigg)\end{aligned} \tag{D.13}$$

giving

$$B_1 = 260\gamma_m\phi^4; \quad B_2 = 392\gamma_m\phi^4; \quad B_3 = 22\gamma_m\phi^4. \tag{D.14}$$

So $(B_1 - B_3)/B_2 = 17/28$. Combining this term with one of the terms already present allows for some tuning of the ratio of the magnetoelastic coupling constants, in principle allowing us to tune the magnetostriction constants independently of the elastic anisotropy.

Appendix E

Three Dimensional Analogue of the Three-point Correlation Interaction

This appendix shows the main ideas behind generalizing the three-point correlation function introduced in Chapter 3 to 3D. The three point term of Eq. 3.4 is given by:

$$F_{ex,3} = -\frac{1}{3}\int n(\mathbf{r})\int C_3(\mathbf{r}-\mathbf{r}', \mathbf{r}-\mathbf{r}'')n(\mathbf{r}')n(\mathbf{r}'')\, d\mathbf{r}' d\mathbf{r}'' d\mathbf{r} \tag{E.1}$$

In Chapter 3 C_3, is rewritten as

$$C_3(\mathbf{r}-\mathbf{r}', \mathbf{r}-\mathbf{r}'') = \sum_i C_s^{(i)}(\mathbf{r}-\mathbf{r}')C_s^{(i)}(\mathbf{r}-\mathbf{r}'') \tag{E.2}$$

where

$$C_s^{(1)}(r,\theta) = C_r(r)C_\theta^{(1)}(\theta) = C_r(r)\cos(m\theta) \tag{E.3}$$

$$C_s^{(2)}(r,\theta) = C_r(r)C_\theta^{(2)}(\theta) = C_r(r)\sin(m\theta) \tag{E.4}$$

$$C_r(r) = \frac{X}{2\pi a_0}\delta(r-a_0) \tag{E.5}$$

Later we use

$$\cos(\theta_1)\cos(\theta_2) + \sin(\theta_1)\sin(\theta_2) = \cos(\theta_2 - \theta_1) \tag{E.6}$$

to show that

$$\sum_i C_s^{(i)}(\mathbf{r_1})C_s^{(i)}(\mathbf{r_2}) = C_r(r_1)C_r(r_2)\cos\left(m(\theta_2-\theta_1)\right) \tag{E.7}$$

E.1 3D three point correlation

For three dimensions the idea is to use the three dimensional "analogue" of Eq. E.6:

$$P_l\left(\frac{\mathbf{r_1}\cdot\mathbf{r_2}}{r_1 r_2}\right) = \frac{4\pi}{2l+1}\sum_{m=-l}^{l} Y_{lm}(\theta_1,\phi_1)Y_{lm}^*(\theta_2,\phi_2) \tag{E.8}$$

Where P_l are the Legendre polynomials and Y_{lm} are spherical harmonics. This is called the *Addition theorem*. Note that the result depends only on the dot product of the two vectors so it is independent of global rotations. For real valued spherical

harmonics (R_{lm}) this is just:

$$P_l\left(\frac{\mathbf{r_1}\cdot\mathbf{r_2}}{r_1 r_2}\right) = \frac{4\pi}{2l+1}\sum_{m=-l}^{l} R_{lm}(\theta_1,\phi_1)R_{lm}(\theta_2,\phi_2) \tag{E.9}$$

This would suggest for the individual $C_s^{(m)}$:

$$C_s^{(m)}(r,\theta,\phi) = C_r(r)C_{\theta,\phi}^{(m)}(\theta,\phi) = C_r(r)R_{lm}(\theta,\phi) \tag{E.10}$$

with m running from $-l$ to l, we use the form of Eq. 3.9. We pick the l we want to get the bond angle we want to produce.

Bibliography

[1] J.J. Hoyt, M. Asta, and A. Karma. Method for computing the anisotropy of the solid-liquid interfacial free energy. *Physical Review Letters*, 86(24):5530–5533, 2001.

[2] T. Haxhimali, A. Karma, F. Gonzales, and M. Rappaz. Orientation selection in dendritic evolution. *Nature Materials*, 5:660–664, 2006.

[3] A. Karma and W.-J. Rappel. *Phys. Rev. E*, 60:3614, 1999.

[4] John W Cahn. Nucleation on dislocations. *Acta Metallurgica*, 5(3):169 – 172, 1957.

[5] J. W. Cahn and J. E. Hilliard. *J. Chem. Phys.*, 28:258, 1958.

[6] J. W. Cahn and J. E. Hilliard. *J. Chem. Phys.*, 31:688, 1959.

[7] P. C. Hohenberg and B. I. Halperin. Theory of dynamic critical phenomena. *Reviews of Modern Physics*, 49(3):435–479, 1977.

[8] L. Q. Chen and W. Yang. *Phy. Rev. B*, 50:15752, 1994.

[9] D. Fan and L.-Q. Chen. *Acta Metallurgica*, 45:3297, 2002.

[10] A. Kazaryan, Y. Wang, S. A. Dregia, and B. R. Patton. *Phys. Rev. B*, 63:184102, 2001.

[11] D. Fan, S. P. Chen, L.-Q. Chen, and P. W. Voorhees. *Acta Materialia*, 50:1897, 2002.

[12] G. Caginalp. *Phys. Rev. A*, 39:5887, 1989.

[13] G. Caginalp and X. Chen. *On On The Evolution Of Phase Boundaries, edited by E. Gurtin and G.B. McFadden*, volume 1. Springer-Verlag, New York, 1992.

[14] A. Karma and W. .J Rappel. Quantitative phase-field modeling of dendritic growth in two and three dimensions. *Phys. Rev. E*, 57:4323–4349, 1998.

[15] B. Echebarria, R. Folch, A. Karma, and M. Plapp. *Phys. Rev. E.*, 70:061604–1, 2004.

[16] C. Tong, M. Greenwood, and N. Provatas. *Phys. Rev. E*, 77:1, 2008.

[17] M. Greenwood, N. Provatas, and J. Rottler. Free energy functionals for efficient phase field crystal modeling of structural phase transformations. *Physical Review Letters*, 105:045702, 2010.

[18] I. Steinbach. *Modelling Simul. Mater. Sci. Eng.*, 17:073001, 2009.

[19] Blas Echebarria, Alain Karma, and Sebastian Gurevich. *Phys. Rev. E*, 81:021608, 2010.

[20] M. Amoorezai, S. Gurevich, and Nikolas Provatas. pacing characterization in al-cu alloys directionally solidified under transient growth conditions. *Acta Materialia*, 58:6115, 2010.

[21] Y. U. Wang, Y. M. Jin, A. M. Cuitino, and A. G. Khachaturyan. *Appl. Phys. Lett.*, 78:2324, 2001.

[22] Y. M. Jin and A. G. Khachaturyan. *Philos. Mag. Lett.*, 81:607, 2001.

[23] L. Q. Chen and A. G. Khachaturyan. *Script. Metall. et Mater.*, 25:61, 1991.

[24] Y. Wang and A. G. Khachaturyan. *Acta. Mater.*, 45:759, 1997.

[25] L.-Q Chen and W. Yang. *Phys. Rev. B*, 50:15752, 1994.

[26] J. A. Warren, W. C. Carter, and R. Kobayashi. *Physica (Amsterdam)*, 261A:159, 1998.

[27] A. G. Khachaturyan. *Theory of structural transformations in solids*. Wiley-Interscience Publications (New York), 1983.

[28] H. Garcke, B. Nestler, and B. Stoth. *SIAM J. Appl. Math*, 60:295, 1999.

[29] J. A. Warren, R. Kobayashi, A. E. Lobkovsky, and W. C. Carter. *Acta Materialia*, 51:6035, 2003.

[30] L. GrŽanŽasy, T. Pusztai, and J. A. Warren. *J. Phys.: Condens. Matter*, 16:R1205, 2004.

[31] H. Garcke, B. Nestler, and B. Stoth. *SIAM J. Appl. Math*, 60:295, 1999.

[32] Lei Wang, Nan Wang, and Nikolas Provatas. *Acta Materialia*, 126:302, 2017.

[33] K. R. Elder, M. Katakowski, M. Haataja, and M. Grant. *Phys. Rev. Lett.*, 88:245701, 2002.

[34] K. R. Elder and Martin Grant. Modeling elastic and plastic deformations in nonequilibrium processing using phase field crystals. *Phys. Rev. E*, 70:051605, Nov 2004.

[35] K.R. Elder, N. Provatas, J. Berry, P. Stefanovic, and M. Grant. Phase-field crystal modeling and classical density functional theory of freezing. *Physical Review B*, 75:064107, 2007.

[36] Y. M. Jin and A. G. Khachaturyan. *J. Appl. Phys.*, 100:013519, 2006.

[37] Gabriel Kocher and Nikolas Provatas. New density functional approach for solid-liquid-vapor transitions in pure materials. *Phys. Rev. Lett.*, 114:155501, 2015.

[38] Nan Wang and Nikolas Provatas. 2017.

[39] A Jaatinen, C. V Achim, K. R Elder, and T Ala-Nissila. *Phy. Rev. E*, 80:031602, 2009.

[40] A Jaatinen and T Ala-Nissila. *Journal of Physics: Condensed Matter*, 22(20):205402, 2010.

[41] M. Greenwood, J. Rottler, and N. Provatas. Phase-field-crystal methodology for modeling of structural transformations. *Physical Review E*, 83:031601, 2011.

[42] M. Greenwood, N. Ofori-Opoku, J. Rottler, and N. Provatas. Modeling structural transformations in binary alloys with phase field crystals. *Physical Review B*, 84:064104, 2011.

[43] Nana Ofori-Opoku, Vahid Fallah, Michael Greenwood, Shahrzad Esmaeili, and Nikolas Provatas. Multicomponent phase-field crystal model for structural transformations in metal alloys. *Phys. Rev. B*, 87:134105, 2013.

[44] J. Berry, N. Provatas, J. Rottler, and C. W. Sinclair. Defect stability in phase field crystal models: Stacking faults and partial dislocations. *Phys. Rev. B*, 86:224112, 2012.

[45] J. Berry, J. Rottler, C. W. Sinclair, and N. Provatas. An atomistic study of diffusion-mediated plasticity and creep using phase field crystal methods. Under Review, 2015.

[46] V. Fallah, A. Korinek, N. Ofori-Opoku, N. Provatas, and S. Esmaeili. Atomistic investigation of clustering phenomenon in the al-cu system: three-dimensional phase-field crystal simulation and hrtem/hrstem characterization. *Acta Materialia*, 61(17):6372–6386, 2013.

[47] Vahid Fallah, Andreas Korinek, Nana Ofori-Opoku, Babak Raeisinia, Mark Gallerneault, Nikolas Provatasc, and Shahrzad Esmaeili. *Acta Materialia*, 82:457, 2015.

[48] S. K. Mkhonta, K. R. Elder, and Z. F. Huang. Exploring the complex world of two-dimensional ordering with three modes. *Phys. Rev. Lett.*, 111:035501, 2013.

[49] Doaa Taha, S. K. Mkhonta, K. R. Elder, and Zhi-Feng Huang. *Phys. Rev. Lett*, 118:255501, 2017.

[50] Bernadine A. Jugdutt, Nana Ofori-Opoku, and Nikolas Provatas. *Phy. Rev. E*, 92:042405, 2015.

[51] Niloufar Faghihi, Nikolas Provatas, Ken Elder, Martin Grant, and Mikko Karttunen. *Phy. Rev. E*, 88:032407, 2013.

[52] T.V. Ramakrishnan and M. Yussouff. First-principles order-parameter theory of freezing. *Physical Review B*, 19(5):2775–2794, 1979.

[53] P. M. Chaikin and T. C. Lubensky. *Principles of condensed matter physics*. Cambridge University Press, 1995.

[54] Alex Hubert and Rudolf Schafer. *Magnetic Domains The Analysis of Magnetic Microstructures*. Springer-Verlag, Berlin, 1998.

[55] G. Herzer. *Nanocrystalline soft magnetic alloys*, volume 10 of *Handbook of Magnetic Materials, Ed. K. H. J. Buschow*, chapter 3. Elsevier Science B.V., 1997.

[56] G. Herzer. Grain-size dependence of coercivity and permeability in nanocrystalline ferromagnets. *IEEE Trans. Magn.*, 26:1397, 1990.

[57] G. Herzer. Soft-magnetic nanocrystalline materials. *Scripta Metel. et Mater.*, 33:1741, 1995.

[58] G. Herzer and H. Warlimont. Nanocrystalline soft magnetic materials by partial crystallization of amorphous alloys. *Nanostruct. Mater.*, 1:263, 1992.

[59] F. Pfeifer and C. Radeloff. Soft magnetic ni-fe and co-fe alloys -some physical and metallurgical aspects. *J. Magn. Magn. Mater.*, 19:190, 1980.

[60] D. Chiba, M. Sawicki, Y. Nishitani, Y. Nakatani, F. Matsukura, and H. Ohno. Magnetization vector manipulation by electric fields. *Nature Lett.*, 455:515, 2008.

[61] T. H. E. Lahtinen, J. O. Tuomi, and S. van Dijken. Pattern transfer and electric-field-induced magnetic domain formation in multiferroic heterostructures. *Adv. Mater.*, 23:3187, 2011.

[62] T. H. E. Lahtinen, J. O. Tuomi, and S. van Dijken. Electrical writing of magnetic domain pattern in ferromagnetic/ferroelectric heterostructures. *IEEE Trans. Magn.*, 47:3768, 2011.

[63] J. Boomgaard, D. R. Terrell, R. A. J. Born, and H. F. J. I. Giller. An in situ grown eutectic magnetoelectric composite material. part 1 composition and unidirectional solidification. *J. Mater. Sci.*, 9:1705, 1974.

[64] A. M. J. G. Run, D. R. Terrell, and J. H. Scholing. An in situ grown eutectic magnetoelectric composite material. part 2 physical properties. *J. Mater. Sci.*, 9:1710, 1974.

[65] J. Boomgaard, A. M. J. G. Run, and J. Suchtelen. Magnetoelectricity in piezoelectric-magnetostrictive composites. *Ferroelectrics*, 10:295, 1976.

[66] A. F Devonshire. Theory of ferroelectrics. *Philos. Mag., Suppl.*, 3:85, 1954.

[67] S. Choudhury, Y. L. Li, C. Krill III, and L. Q. Chen. Effect of grain orientation and grain size on ferroelectric domain switching and evolution: Phase field simulations. *Acta. Mater.*, 55:1415, 2007.

[68] Matthew Seymour, F. Sanches, Ken Elder, and Nikolas Provatas. Phase-field crystal approach for modeling the role of microstructure in multiferroic composite materials. *Phys. Rev. B*, 92:184109, Nov 2015.

[69] M. Greenwood, N. Provatas, and J. Rottler. *Phys. Rev. Lett.*, 105(4):045702, 2010.

[70] Michael Greenwood, Jörg Rottler, and Nikolas Provatas. Phase-field-crystal methodology for modeling of structural transformations. *Phys. Rev. E*, 83:031601, Mar 2011.

[71] K. R. Elder, Z.-F. Huang, and N. Provatas. Amplitude expansion of the binary phase-field-crystal model. *Physical Review E*, 81(1):011602, 2010.

[72] Nana Ofori-Opoku, Jonathan Stolle, Zhi-Feng Huang, and Nikolas Provatas. Complex order parameter phase-field models derived from structural phase-field-crystal models. *Phys. Rev. B*, 88:104106, 2013.

[73] C. Kittel. *Rev. Mod. Phys.*, 21:541, 1949.

[74] A. K. Geim and K. S. Novoselov. The rise of graphene. *Nature Materials*, 6:183–191, 2007.

[75] Weiwei Cai, Yanwu Zhu, Xuesong Li, Richard D. Piner, and Rodney S. Ruoff. Large area few-layer graphene/graphite films as transparent thin conducting electrodes. *Applied Physics Letters*, 95:123115, 2009.

[76] S. Stankovich, D. A. Dikin, G. H. B. Dommett, K. M. Kohlhaas, E. J. Zimney, E. a. Stach, R. D. Piner, S. T. Nguyen, and R. S. Ruoff. *Nature*, 442:282, 2006.

[77] C. Lee, X. Wei, J. W. Kysar, and J. Hone. Measurement of the elastic properties and intrinsic strength of monolayer graphene. *Science*, 321:385, 2008.

[78] K. S. Novoselov, A. K. Geim, S. V. Morozov, D. Jiang, Y. Zhang, S. V. Dubonos, I. V. Grigorieva, and A. A. Firsov. Electric field effect in atomically thin carbon films. *Science*, 306:666, 2004.

[79] Xuesong Li, Carl W. Magnuson, Archana Venugopal, Rudolf M. Tromp, James B. Hannon, Eric M. Vogel, Luigi Colombo, and Rodney S. Ruoff. Large-area graphene single crystals grown by low-pressure chemical vapor deposition of methane on copper. *J. Am. Chem. Soc.*, 133:2816–2819, 2011.

[80] H. Tetlow, J. Posthuma de Boer, I. J. Ford, D. D. Vvedensky, J. Coraux, and L. Kantorovich. Growth of epitaxial graphene: Theory and experiment. *Nat. Nanotechnol.*, 9:755, 2014.

[81] C. S. Ruiz-Vargas, H. L. Zhuang, P. Y. Huang, A. M. van der Zande, S. Garg, P. L. McEuen, D. A. Muller, R. G. Hennig, and J. Park. *Nano Lett.*, 11:2259, 2011.

[82] G.-H. Lee, R. C. Cooper, S. J. An, S. Lee, A. van der Zande, N. Petrone, A. G. Hammerberg, C. Lee, B. Crawford, W. Oliver, J. W. Kysar, and J. Hone. *Science*, 340:1073, 2013.

[83] H. I. Rasool, C. Ophus, W. S. Klug, A. Zettl, and J. K. Gimzewski. *Nature Commun.*, 4:2811, 2013.

[84] O. V. Yazyev and S. G. Louie. *Phy. Rev. B*, 81:195420, 2010.

[85] Y. Wei, J. Wu, H. Yin, X. Shi, R. Yang, and M. Dresselhaus. *Nature Materials*, 11:759, 2012.

[86] Matthew Daly and Chandra Veer Singh. A kinematic study of energy barriers for crack formation in graphene tilt boundaries. *Journal of Applied Physics*, 115:223513, 2014.

[87] Natasha M. Galea, Daniel Knapp, and Tom Ziegler. Density functional theory studies of methane dissociation on anode catalysts in solid-oxide fuel cells: Suggestions for coke reduction. *Journal of Catalysis*, 247:20–33, 2007.

[88] Vasilii I. Artyukhov, Yufeng Hao, Rodney S. Ruoff, and Boris I. Yakobson. Breaking of symmetry in graphene growth on metal substrates. *Phys. Rev. Lett.*, 114:115502, 2015.

[89] Esteban Meca, John Lowengrub, Hokwon Kim, Cecilia Mattevi, and Vivek B. Shenoy. Epitaxial graphene growth and shape dynamics on copper: Phase- epitaxial graphene growth and shape dynamics on copper: Phase- field modeling and experiments. *Nano Lett.*, 13:5692–5697, 2013.

[90] K.R. Elder and M. Grant. Modeling elastic and plastic deformations in non equilibrium processing using phase field crystals. *Physical Review E*, 70:051605, 2004.

[91] J. Mellenthin, A. Karma, and M. Plapp. Phase-field crystal study of grain-boundary pre melting. *Physical Review B*, 78:184110, 2008.

[92] G. I. Toth, T. Pusztai, G. Tegze, G. Toth, and L. Granasy. *Phys. Rev. Lett.*, 107:175702, 2011.

[93] L. Granasy, F. Podmaniczky, G. I. Toth, G. Tegze, and T. Pusztai. *Chem. Soc. Rev.*, 43:2159–2173, 2014.

[94] J. Berry, N. Provatas, J. Rottler, and C. W. Sinclair. Phase field crystal modeling as a unified atomistic approach to defect dynamics. *Phys. Rev. B*, 89:214117, 2014.

[95] K.R. Elder, M. Katakowski, M. Haataja, and M. Grant. Modeling elasticity in crystal growth. *Physical Review Letters*, 88(24):245701, 2002.

[96] Kuo-An Wu, Mathis Plapp, and Peter W Voorhees. Controlling crystal symmetries in phase-field crystal models. *Journal of Physics: Condensed Matter*, 22(36):364102, 2010.

[97] V. Kalikmanov. *Statistical physics of fluids: basic concepts and applications*. Springer Berlin, 2001.

[98] Matthew Seymour. Private communication of spherical harmonics idea to Dr. David Montiel, University of Michigan., November 2015.

[99] Kate L. M. Elder. New structural phase field crystal model for graphene symmetries in alloys. Master's thesis, McGill University, 2017.

[100] L. D. Landau and E. M. Lifshitz. *Theory of Elasticity (3rd ed.)*. ButterWorth-Heinemann, 1986.

[101] P. Hirvonen, M. M. Ervasti, Z. Fan, M. Jalavand, M. Seymour, S. Mehdi Vaez Allaei, N. Provatas, A. Harju, K. R. Elder, and T. Ala-Nissila. *Phys. Rev. B*, 94(3):035414, 2016.

[102] Pinshane Y Huang, Carlos S Ruiz-Vargas, Arend M van der Zande, William S Whitney, Mark P Levendorf, Joshua W Kevek, Shivank Garg, Jonathan S Alden, Caleb J Hustedt, Ye Zhu, et al. Grains and grain boundaries in single-layer graphene atomic patchwork quilts. *Nature*, 469(7330):389–392, 2011.

[103] K.L.M Elder, M. Seymour, M. Lee, M. Hilke, and N. Provatas. 2017.

[104] Zhi-Fang Huang, Ken Elder, and Nikolas Provatas. *Phy. Rev. E*, 82:021605, 2010.

[105] Jean-Pierre Hansen and I.R. McDonald. *Theory of Simple Liquids With Applications to Soft Matter*. Elsevier, fourth edition) edition, 2013.

[106] A A. Drabińska, M. Kamińska A. Wołoś, W. Strupinski, A. Wysmolek, W. Bardyszewski, R. Bożek, and J. .M. Baranowski. Enhancement of elastic and inelastic scattering lengths in quasi-free-standing graphene measured with contactless microwave spectroscopy. *Phy. Rev. B*, 88:165413, 2013.

[107] B. Jabakhanji, N. Camara, A. Caboni, C. Consejo, B. Jouault, P. Godignon, and J. Camassel. Almost free standing graphene on sic(000-1) and sic(11-20). *Materials Science Forum*, 711:235–241, 2012.

[108] T. Balgar, H. Kim, and E. Hasselbrink. *J. Phys. Chem. Lett*, 4:2094, 2013.

[109] F. Banhart, J. Kotakoski, and A. Krasheninnikov. *ACS NANO*, 5(1):26, 2010.

[110] Dalhousie University Jesse Maassen. Private Communication, July 2017.

[111] Eli Alster, David Montiel, Katsuyo Thornton, and Peter W. Voorhees. Simulating complex crystal structures using the phase-field crystal model. *arXiv:1707.03044v1 [cond-matt.mtrl-sci]*, July 2017.

[112] Dragica Vasileska, Stephen M. Goodnick, and Gerhard Klimeck. *Computational Electronics: Semiclassical and Quantum Device Modeling and Simulation*. CRC Press Taylor and Francis Group, 6000 Broken Sound Parkway NW, Suite 300, Boca Raton, FL 33487-2742, April 2010.

[113] Eliot A. Kearsley and Jeffrey T. Fong. *JOURNAL OF RESEARCH of the National Bureau of Standards - B. Mathematical Sciences*, 79B(1 and 2):49.