# Report - CHE Mobility Discovery

Reese Karo, Sidd Sabata

September 2023

**Abstract**

## 1 Introduction and Problem of interest

In this study, we conduct a computational study on determining an unknown parameter, called *mobility* (see Section 2 for details), of the Cahn-Hilliard (CH) equation, a model from materials science, using synthetic data.

We briefly summarize the history related to the CH equation. Cahn and Hilliard published what is now called Cahn-Hilliard equation in 1958 ([6]) to model dynamics of two liquid phases. Since then, it has been widely generalized and used for many different applications. We refer interested readers in its variants and applications to a recent review paper [18] and the reference therein. It was later understood that the CH equation could also explain two-phase system of solid mixture ([5]). More mathematically oriented formulation came in 1993 using $H^{-1}$–gradient flow framework in [20]. In [4], the CH equation was studied from a different angle. The authors explained the dynamics the interface as the time evolution of the CH equation goes. The existence of equilibrium solution was first established in [19]. More detailed study of the global solutions with constant mobility was conducted in [11]. In [10], the existence of weak solution was proved with variable mobility. The authors of [21] proved that, under certain conditions, the unique classical solution existed. The CH equation is numerically challenging due to its high order, nonlinear, and non-convex nature. There has been tremendous effort put into numerical study. Among the huge amount of literature, the authors of [2] studied the finite element method (FEM) for the CH equation with variable mobility. In [12], a two grid method based on FEM was proposed while the authors of [8] studied a multigrid algorithm based on finite difference method. Many researchers studied convex-splitting technique to tackle nonlinearity of the CH equation, which include the authors of [9]. Also, spectral methods were proposed in [22] for a semi-implicit time discretization.

## 2 Cahn-Hilliard Equation

The Cahn-Hilliard Equation (CHE) plays an important role in phase field modeling and finds applications in diverse areas, ranging from simulating spinodal decomposition of binary alloys, to microstructure evolution that can help medical experts track the growth of tumor and rapid cell growth. Spinodal decomposition is a phase separation process, characterized by a web-like pattern, that shows how different phases grow and separate over time. This web-like structure is unique to spinodal decomposition and offers a way to track the behavior of various phases under different conditions. Phase field models, adept at tracking phenomena like grain growth and coarsening as well as crack propagation, owe its effectiveness in part to the treatment of conserved and non-conserved fields.

There are three different stages of phase separation via spinodal decomposition for a binary mixture of components A and B in thermodynamic phases of either a liquid or solid phase. The "early stage" is defined as the demixing period, where the phases separate until equilibrium concentration. After the demixing event has finished, the evolution of the system will lead into a coarsening period for the separated phases in order to reach further thermodynamic equilibrium. Coarsening is the growth of larger particles in place for smaller particles shrinking or dissolving. This stage is better known as the "intermediate stage", which according to [13], no theory is known for the methodology behind the rapid change into the coarsening. Next is the "late stage" which is the elongated evolution of the phases which can be theorized by the LSW-theory, by Lifshitz, Slyozov and Wagner, [7, 16], describing the physics behind coarsening and phase separation for later times.

We note from [1] that the nucleation - the first step towards the formation or transition into a new thermodynamic phase - mechanism is different from typical phase separating processes in that the homogeneous state is resistant to small changes in composition. However as large fluctuations in phase separation occurs at the nucleation sites, the phase separation becomes unstable.

The evolution of the spinodal decomposition in a system is governed by the Cahn-Hilliard equation. The general form of the CHE is given by

$$u_t = \nabla \cdot (M(u)\nabla w), \tag{1}$$

where $u_t$ is the rate of change of $u$ with respect to time $t$, and $u$ is the order parameter lying in the range $[-1, 1]$ ([15, Eqn. 2]). $w$ is the diffusional chemical potential given by the functional (variational) derivative

$$w = \frac{\delta E}{\delta u} = -\epsilon^2 \Delta u + \psi(u), \tag{2}$$

where $\epsilon$ is the thickness of the phase interface, and $\psi(u)$ represents the derivative of the bulk free energy, with respect to the order parameter ([15, Eqn. 3]). $M$ represents the transport or mobility coefficient which is one of the governing parameter driving the coarsening effect in spinodal decomposition. Both conserved and non-conserved fields are regulated by the Ginzburg-Landau free energy, denoted as $E$, of a system, in conjunction with the functional (variational) derivative with respect to the *phase variable* $u$, representing the concentration field of two phases. However, in the Cahn-Hilliard Equation, since mass is conserved throughout phase separation, we are only considering a conserved system. Phase-field modeling excels in tracking interfaces by minimizing the total free energy of the system while preserving mass. The total free energy functional of the system

$$E(u) = \int_{\Omega} [\frac{\epsilon^2}{2} \left|\nabla u\right|^2 + \Psi(u)]dx, \tag{3}$$

over the spatial domain $\Omega$, from $(0, L^2)$, where $(L > 0)$, and $L$ is the length of our square domain ([15, p. 4-5]). Importantly, $\Psi(u)$ often exhibits a double-well potential, with local minima at $u = -1$ and $u = 1$ representing the relative concentration of the two phases. $\Psi(u) = \frac{1}{4}(1 - x^2)^2$ is an approximation of the double well potential of the system. On the left hand side of Eq. (3), we have a gradient energy at the *phase interfaces* with the gradient coefficient indicating intermolecular interactions which is also a penalizing term for interface formation Also, on the left side of the functional , $\epsilon$ represents the thickness of the *phase interface*.

To reiterate these concepts, the evolution for a conserved *phase variable*, $u$, evolves in the direction where the total free energy diminishes the quickest, while still preserving the mass of the system. [14][15]

## 2.1 Mobility

The mobility coefficient denoted as $M$, can occur in two forms of transport. One being *uniform mobility* which is a constant mobility, and the other is a function of the order parameter $u$, defined as *interfacial mobility*. interfacial mobility can be seen as

$$M(u) = M_0 \left|1 - u^2\right| \tag{4}$$

where $M_0$ refers to the mobility factor ([17, Eqn. 3]). Under constant mobility, as seen in [17, Fig. 3b1, 3b2], the phase transport is unrestricted to the interfacial region, whereas under interfacial mobility, the phase transport to the same regions are restricted due to the partial derivative of our concentration parameter, $c$ , with respect to $t$. This causes a weaker inflow of flux, however, no such negative concentration is found under constant mobility, thus there is constant flux flowing through the phase interface. In our study, we will focus on primarily on uniform mobility.

### 2.2 PPGD CHE solver

## 3 Methodology

### 3.1 Ordinary Least Squares (OLS)

In this work, we employ Ordinary Least Squares (OLS) as an optimization technique to approximate the solution of an overdetermined system of equations, which can be represented as:

$$\boldsymbol{\Theta}\xi = \mathbf{x} \tag{5}$$

Here, $\boldsymbol{\Theta}$ is the $m \times n$ coefficient matrix, $\mathbf{x}$ is the observed data vector, and $\xi$ is the target solution vector we aim to approximate. The objective of OLS is to find $\xi$ that minimizes the residual sum of squares, given by:

$$\hat{\xi} = \arg\min_{\xi} \|\mathbf{x} - \boldsymbol{\Theta}\xi\|_2^2 \tag{6}$$

The OLS solution, denoted as $\xi^*$, is computed using the pseudo-inverse of $\boldsymbol{\Theta}$ as:

$$\xi^* = (\boldsymbol{\Theta}^\top\boldsymbol{\Theta})^{-1}\boldsymbol{\Theta}^\top\mathbf{x} \tag{7}$$

This approach provides a straightforward yet effective method for approximating the solution to the problem defined by Eq. (5), making it well-suited for our research.

### 3.2 Sequential Least Squares (SLS)

Sequential Least Squares (SLS), also known as Sequential Thresholded Least Squares (STLS), is an iterative method designed to find a sparse solution to an over determined system of a system of linear equations that takes the form $\Theta\xi = x$, where $\Theta \in \mathbb{R}^{m\times n}$, an unknown vector $\xi \in \mathbb{R}^n$, and our target vector $x \in \mathbb{R}^m$. This algorithm begins with a ordinary least squares regression to find an initial solution vector, which refer back to the previous subsection. Subsequently, terms in the solution vector that are smaller than a threshold provided bu the user are *zeroed* out. The least squares regression is then reapplied to the system. This process is repeated until the solution vector remains unchanged between iterations. Another method is to add to the inputs, "Max Iteration", which will cause the function to break when the iteration count has reached the max iteration integer. The motivation for seeking a sparse solution often comes from the need for simpler, more interpretable models, that include a few terms and can still provide the essentials to the model. This is explained more in depth in ([3]).

## 3.3 Least Absolute Shrinkage and Selection Operator (LASSO)

The Least Absolute Shrinkage and Selection Operator method for sparse regression introduces the L1-norm penalty in light of reducing overfitting for high-dimensionality data, by mitigating coefficients that are not significant in the system. The LASSO method is widely used because of its simplicity in code for computation, and because of the delivery of sparse solutions. Though, many mathematicians in the community agree LASSO can be computationally expensive depending on the amount of data we collect, and unstable. LASSO minimizes sparse vectors by minimizing

$$\hat{\boldsymbol{\Xi}} = \|\mathbf{X} - \boldsymbol{\Theta}\boldsymbol{\Xi}\|_2^2 + \lambda \|\boldsymbol{\Xi}\|_{\mathbf{1}} \tag{8}$$

## 3.4 Sparse Identification of Nonlinear Dynamics (SINDy)

Our goal for this paper is to dive into the complex problem of approximating the mobility coefficient from the Cahn-Hilliard equation. Mobility coefficients are critical parameters in a variety of applications, including material science and fluid dynamics. Accurate estimation of these coefficients can significantly impact the predictive power of mathematical models in these domains. Traditionally, the estimation of mobility coefficients has been a computationally intensive task, often requiring the solution of large-scale optimization problems. However, the abruptness of high-dimensional data and the need for real-time analytics necessitate more efficient methods. This is where sparsity comes into play. A sparse solution not only reduces computational complexity, but also enhances interpretability for other people to learn. To achieve this sparse representation, we turn to the Sparse Identification of Nonlinear Dynamics (SINDy) algorithm. Because SINDy focuses on reducing redundant and noisy data, this is an ideal algorithm for working on the Cahn-Hilliard Equation due to the high dimensionality and noise. We may now consider dynamical systems whose equations takes the form:

$$\dot{\mathbf{X}} = \mathbf{f}(\mathbf{x}(t)) = \boldsymbol{\Theta}(\mathbf{X})\boldsymbol{\Xi} \tag{9}$$

We can define $\dot{\mathbf{X}}$ to be the time derivatives of the state of our system, typically found by numerical differentiation of $\mathbf{x}(\mathbf{t})$, which is a vector representing the state of the system at a specific time, t. The data used is from many time point samples, $t_1, t_2, ..., t_n$ collected via computationally.

$\boldsymbol{\Theta}(\mathbf{X})$ is a data matrix that contains *candidate* functions of the state of the system at each time point. These functions may include: constant, polynomial and trigonometric functions to help describe the system we are trying to recover. It is crucial to find the "correct" coordinate system and function basis for the SINDy algorithm.

$$\mathbf{X} = \begin{pmatrix} x^{\mathbf{T}}(t_1) \\ x^{\mathbf{T}}(t_2) \\ \vdots \\ x^{\mathbf{T}}(t_n) \end{pmatrix} = \begin{pmatrix} x_1(t_1) & x_2(t_1) & \cdots & x_m(t_1) \\ x_1(t_2) & x_2(t_2) & \cdots & x_m(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(t_n) & x_2(t_n) & \cdots & x_m(t_n) \end{pmatrix} \tag{10}$$

$$\dot{\mathbf{X}} \begin{pmatrix} \dot{x}^{\mathbf{T}}(t_1) \\ \dot{x}^{\mathbf{T}}(t_2) \\ \vdots \\ \dot{x}^{\mathbf{T}}(t_n) \end{pmatrix} = \begin{pmatrix} \dot{x}_1(t_1) & \dot{x}_2(t_1) & \cdots & \dot{x}_m(t_1) \\ \dot{x}_1(t_2) & \dot{x}_2(t_2) & \cdots & \dot{x}_m(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \dot{x}_1(t_n) & \dot{x}_2(t_n) & \cdots & \dot{x}_m(t_n) \end{pmatrix} \tag{11}$$

Where, in Eq. (10), we have the state of the system for each dimension in the vector $\mathbf{X(t)}$, at every time point, t. This is the time series data that is required to help us gain a good understanding of the system, we want to model.

We define a potential library, denoted as the data matrix $\mathbf{\Theta}(\mathbf{X})$, to aid in the approximation of the system's dynamics. This library consists of candidate functions, including polynomials and trigonometric functions, defined as follows:

$$\mathbf{\Theta}(\mathbf{X}) = \begin{bmatrix} | & | & | & | & | & | & | & | \\ \mathbf{1} & \mathbf{X_1} & \mathbf{X_2} & \mathbf{X_3} & \cdots & \sin(\mathbf{X}) & \cos(\mathbf{X}) & \cdots \\ | & | & | & | & | & | & | & | \end{bmatrix} \tag{12}$$

Here, $\mathbf{X_1}$, $\mathbf{X_2}$, and $\mathbf{X_3}$ represent the linear, quadratic, and cubic terms of the polynomials involved, respectively. The trigonometric functions $\sin(\mathbf{X})$ and $\cos(\mathbf{X})$ are also included as candidate functions. The primary purpose of this library is to identify the combination of these candidate functions that best approximates the system's dynamics. Hence, converting a nonlinear model into a linear system problem. Once we have calculated the derivatives, $\dot{X}$, we employ sparse regression techniques to select terms from this library that form a minimal yet accurate representation of the system's dynamics. This results in a coefficient matrix $\mathbf{\Xi} = \begin{bmatrix} \xi_1 & \xi_2 & \xi_3 & \cdots & \xi_n \end{bmatrix}$ where each column of $\Xi$, $\xi_{\mathbf{k}}$, describes which terms are active in approximating the right-hand side of our governing equations.

$$\dot{\mathbf{x}}_{\mathbf{k}} = \mathbf{f}(\mathbf{x}) = \mathbf{\Theta}(\mathbf{x}^{\mathbf{T}})\xi_{\mathbf{k}} \tag{13}$$

which is actually equivalent to

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) = \mathbf{\Xi}^{\mathbf{T}}(\mathbf{\Theta}(\mathbf{x}^{\mathbf{T}}))^{\mathbf{T}} \tag{14}$$

because the $\Theta$ is a data

### 3.5 Chaotic Lorenz System

As a preparatory step for our main study, we employed the Lorenz System as a testing ground for the SINDy algorithm we will use later for the Cahn-Hilliard Equation data. The Lorenz system is a well-known ODE showcasing chaotic behavior and is given by:

$$\begin{aligned}\dot{x} &= \sigma(y - x),\\ \dot{y} &= x(\rho - z) - y,\\ \dot{z} &= xy - \beta z.\end{aligned}$$

We used this system to test the viability of using a basis function design matrix of polynomials up to degree 5, computed via the Runge-Kutta of order 4 (RK4) method.

#### 3.5.1 Runge-Kutta of Order 4 Method (RK4)

The RK4 is employed to solve for the slopes to solve for the Lorenz ODE. It uses a weighted average of various slopes for its computations:

$$w_{i+1} = w_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4),$$

where,

$$\begin{aligned}k_1 &= f(t_i, w_i)\\ k_2 &= f\left(t_i + \frac{h}{2}, w_i + \frac{h}{2}k_1\right)\\ k_3 &= f\left(t_i + \frac{h}{2}, w_i + \frac{h}{2}k_2\right)\\ k_4 &= f\left(t_i + h, w_i + hk_3\right)\end{aligned}$$

where the $k_i$ terms are computed using different slope approximations at different time steps. Its effectiveness and simplicity make it ideal for our Lorenz system setup, initialized at $t_0, w_0 = (8, -8, 27)$.

After state computation, we focus on calculating the system's derivatives, $\dot{\mathbf{X}}$, usually via central difference methods:

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}. \tag{15}$$

To capture the system's dynamics, we include polynomial terms up to 5th order in our Theta matrix. Optimization techniques like Least Squares, LASSO, and SLS are then applied to finalize the model, and return the original parameters $\sigma = 10, \beta = 8/3$, and $\rho = 28$ .

In many numerica analysis textbooks, the RK4 method can be explained in more detail, as well as many other methods for better approximation for the derivative

# 4 Theory

# 5 Numerical Implementation

## 5.1 Settings

In this section we take a further look into the settings of our solver that produces the Cahn-Hilliard data for our computations to find the mobility using basis functions in our design matrix, and regression to solve for the parameters.

Our parameter settings for generating data on a 64 × 64 square domain using the PPGD Cahn-Hilliard equation solver are as follows: The spatial domain of our simulations were set to $\Omega = (0, L)^2$, where the length of the spatial square domain $L = 1$, and the resolution of our spatial discretization $N = 2^6$ giving us our 64 × 64 discretized spatial grid. The thickness of our phase interface, $\epsilon = 0.01$. The number of time steps is defined as $K = 100$ , The final time of the microstructure evolution, $T = 0.1$. By tuning our "number of saves", we set this to 20, which results in a data save every 5 iterations. This will give us 21 total time intervals including the initial conditions as the first save. Lastly we set our step size of the outer loop of the PPGD solver for a single time march, $\beta = 0.2$.

The PPGD solver uses the implicit Euler scheme method to solve for each iteration

$$\frac{u^{k+1} - u^k}{T/K} = \nabla \cdot (M(u_k)\nabla w^{k+1})$$

where $w$ is our chemical potential described in (2)

$$w^{k+1} = -\epsilon^2 \Delta u^{k+1} + \psi(u^{k+1})$$

Lastly we focus on the mobility settings, $M$, allowing us to tune our data in order to build our design matrix, $\Theta(x)$. an example of our setting for $M$ in a constant form would be to set $M = 1.5$ if we were to want constant mobility. If a linear mobility is desired, we would set our mobility to $M = 0.2$x or something similar.

## 5.2 Results

Our Cahn-Hilliard data was generated with different functions consisting utilizing the PPGD solver with the following equations:

$$y_c = 1$$
$$y_a = 0.5x + 1$$
$$y_q = 0.25x^2 + 0.5$$

where $y_c$ represents our data from a constant mobility setting in the programming. $y_l$ represents an affine function and $y_q$ is a quadratic function for mobility, providing 3 columns for our basis function matrix shown below. Each function of different mobility's resulted in the output of a $21 \times 64 \times 64$ tensor. Each $64 \times 64$ array in the tensor corresponds to a single point in time of the dynamics. On top of the initial condition, there are twenty more points of time series data. In order to fit the SINDy framework described above, we took each mobility functions data generated Cahn-Hilliard, and we flatten each into a single column vector, allowing easier interpretation of the results in the optimization that will follow.

Assuming we are targeting the constant function of mobility data $(y_c)$ , we can set up the simple optimization problem as follows:

$$\begin{bmatrix} | & | & | \\ y_c & y_a & y_q \\ | & | & | \end{bmatrix} \begin{bmatrix} c_c \\ c_a \\ c_q \end{bmatrix} = \begin{bmatrix} | \\ y_c \\ | \end{bmatrix}$$

With this framework, we can begin to apply regression techniques such as least squares, sequential threshold least squares, LASSO, and Elastic Net. All of these methods result in the following "$\Xi$" matrix:

$$\Xi = \begin{bmatrix} c_c \\ c_a \\ c_q \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

This result indicates that we have successfully estimated our target data using the SINDy framework, with .

### 5.3 Interpretation

## 6 Conclusion

### 6.1 Work in the Future

## 7 Appendix

### 7.1 Gateaux Derivative

### 7.2 Proof of a functional derivative

The chemical potential in the Cahn-Hilliard equation can be derived using the concept of a functional derivative. Consider an energy functional $E$ that depends on a field variable $u(x)$ and its first order derivatives, $\nabla u(x)$. The functional derivative of $E$ with respect to $u$ is defined as:

$$\frac{\delta E}{\delta u} = \frac{\partial E}{\partial u} - \nabla \cdot \frac{\partial E}{\partial \nabla u}$$

Proof:

$$
\begin{aligned}
\langle \nabla E(u), v \rangle &= \frac{d}{dt} E(u + vt)\bigg|_{t=0} && (16)\\
&= \left[\int_\Omega \frac{d}{dt} E(x, u + vt, \nabla u + \nabla vt)\right]_{t=0} && (17)\\
&= \int_\Omega \left( \frac{\partial E}{\partial u} v + \frac{\partial E}{\partial \nabla u} \cdot \nabla v \right) dx && (18)\\
&= \int_\Omega \left( \frac{\partial E}{\partial u} v - (\nabla \cdot \frac{\partial E}{\partial \nabla u}) v \right) dx + \int_{\partial\Omega} \left( \frac{\partial E}{\partial \nabla u} \cdot n \right) v \, ds && (19)\\
&= \int_\Omega \left( \frac{\partial E}{\partial u} - \nabla \cdot \frac{\partial E}{\partial \nabla u} \right) v dx = \int_\Omega \frac{\delta E}{\delta u} v && (20)
\end{aligned}
$$

which shows that the directional derivative and gradient of the energy functional relate by inner product.

# References

[1] Reza Abazari, Hadi Rezazadeh, Lanre Akinyemi, and Mustafa Inc. Numerical simulation of a binary alloy of 2d cahn–hilliard model for phase separation. *Computational and Applied Mathematics*, 41:389, 2022.

[2] John W. Barrett and James F. Blowey. Finite element approximation of the Cahn-Hilliard equation with concentration dependent mobility. *Math. Comp.*, 68(226):487–517, 1999.

[3] Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113:3932 – 3937, 2015.

[4] J. W. Cahn, C. M. Elliott, and A. Novick-Cohen. The Cahn-Hilliard equation with a concentration dependent mobility: motion by minus the Laplacian of the mean curvature. *European J. Appl. Math.*, 7(3):287–301, 1996.

[5] John W Cahn. On spinodal decomposition. *Acta Metallurgica*, 9(9):795–801, 1961.

[6] John W. Cahn and John E. Hilliard. Free energy of a nonuniform system. i. interfacial free energy. *The Journal of Chemical Physics*, 28(2):258–267, 1958.

[7] Wagner Carl. Theorie der alterung von nierdeerschlägen durch ulmösen (ostwald-reifung). *Berichte der Bunsengesellschaft für physikalische Chemie*, 65:581–591, 1961.

[8] Wenbin Chen, Cheng Wang, Xiaoming Wang, and Steven M. Wise. Positivity-preserving, energy stable numerical schemes for the Cahn-Hilliard equation with logarithmic potential. *J. Comput. Phys. X*, 3:100031, 29, 2019.

[9] Xiaochun Chen, Cheng Wang, and Steven M. Wise. A preconditioned steepest descent solver for the Cahn-Hilliard equation with variable mobility. *Int. J. Numer. Anal. Model.*, 19(6):839–863, 2022.

[10] Charles M. Elliott and Harald Garcke. On the Cahn–Hilliard equation with degenerate mobility. *SIAM journal on mathematical analysis*, 27(2):404–423, 1996.

[11] Charles M. Elliott and Zheng Songmu. On the Cahn-Hilliard equation. *Archive for rational mechanics and analysis*, 96(4):339–357, 1986.

[12] Hongen Jia, Yang Li, Guorui Feng, and Kaitai Li. An efficient two-grid method for the Cahn-Hilliard equation with the concentration-dependent mobility and the logarithmic Flory-Huggins bulk potential. *Appl. Math. Comput.*, 387:124548, 15, 2020.

[13] Björn König, Olivier J. J. Ronsin, and Jens Harting. Two-dimensional cahn–hilliard simulations for coarsening kinetics of spinodal decomposition in binary mixtures. *Physical Chemistry Chemical Physics*, 23:24823–24833, 2021.

[14] Dongsun Lee, Joo Youl Huh, Darae Jeong, Jaemin Shin, Ana Yun, and Junseok Kim. Physical, mathematical, and numerical derivations of the cahn–hilliard equation. *Computational Materials Science*, 81:216–225, 2014.

[15] Wei Li, Martin Z. Bazant, and Juner Zhu. Phase-field deeponet: Physics-informed deep operator neural network for fast simulations of pattern formation governed by gradient flows of free-energy functionals. *ArXiv*, abs/2302.13368, 2023.

[16] Lifshitz I. M. and Slyozov V. V. The kinetics of precipitation from supersaturated solid solutions. *Journal of Physics and Chemistry of Solids*, 19(1-2):35–50, 1961.

[17] Zirui Mao and Michael J. Demkowicz. Mobility inference of the cahn–hilliard equation from a model experiment. *Journal of Materials Research*, 36:2830 – 2842, 2021.

[18] Alain Miranville. *The Cahn-Hilliard equation*, volume 95 of *CBMS-NSF Regional Conference Series in Applied Mathematics.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2019. Recent advances and applications.

[19] Amy Novick-Cohen and Lee A. Segel. Nonlinear aspects of the Cahn-Hilliard equation. *Phys. D*, 10(3):277–298, 1984.

[20] Jean E. Taylor and John W. Cahn. Linking anisotropic sharp and diffuse surface motion laws via gradient flows. *J. Statist. Phys.*, 77(1-2):183–197, 1994.

[21] Jingxue Yin and Changchun Liu. Regularity of solutions of the Cahn-Hilliard equation with concentration dependent mobility. *Nonlinear Anal.*, 45(5):543–554, 2001.

[22] Jingzhi Zhu, Long-Qing Chen, Jie Shen, and Veena Tikare. Coarsening kinetics from a variable-mobility cahn-hilliard equation: Application of a semi-implicit fourier spectral method. *Phys. Rev. E*, 60:3564–3572, Oct 1999.