TRANSFORMED MODEL REDUCTION FOR PARTIAL DIFFERENTIAL EQUATIONS WITH SHARP INNER LAYERS *

TIANYOU TANG † AND XIANMIN XU ‡

Abstract. Small parameters in partial differential equations can give rise to solutions with sharp inner layers that evolve over time. However, the standard model reduction method becomes inefficient when applied to these problems due to the slowly decaying Kolmogorov N-width of the solution manifold. To address this issue, a natural approach is to transform the equation in such a way that the transformed solution manifold exhibits a fast decaying Kolmogorov N-width. In this paper, we focus on the Allen-Cahn equation as a model problem. We employ asymptotic analysis to identify slow variables and perform a transformation of the partial differential equations accordingly. Subsequently, we apply the Proper Orthogonal Decomposition (POD) method and a qDEIM technique to the transformed equation with the slow variables. Numerical experiments demonstrate that the new model reduction method yield significantly improved results compared to direct model reduction applied to the original equation. Furthermore, this approach can be extended to other equations, such as the convection equation and the Burgers equation.

Key words. Model reduction, Nonlinearity, Small parameter models, Slow variables

1. Introduction.

Numerous physical, chemical, and biological processes can be effectively described by nonlinear partial differential equations (PDEs) that involve small parameters. These small parameters are often associated with the multiscale behavior exhibited by the solutions. However, numerical computations for such problems tend to be time-consuming and demand significant computational resources. This inefficiency becomes particularly pronounced when the equations need to be solved repeatedly or in real-time scenarios to control systems or industrial processes. Consequently, there is a strong motivation to perform model reduction on these systems and solve the resulting reduced models in practical applications.

After spatial discretization of the PDEs with small parameters, a nonlinear parametric dynamic system can be obtained as follows:

$$\mathbf{E}_\varepsilon \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{A}_\varepsilon \mathbf{u} + F_\varepsilon(\mathbf{u}).$$

Here $\mathbf{u} \in \mathbb{R}^n$ and $\varepsilon \ll 1$ is a parameter. $\mathbf{E}_\varepsilon, \mathbf{A}_\varepsilon \in \mathbb{R}^{n\times n}$ are matrices and $F_\varepsilon : \mathbb{R}^n \to \mathbb{R}^n$ is a nonlinear vector-valued function. $\mathbf{E}_\varepsilon$, $\mathbf{A}_\varepsilon$ and F_ε may depend on ε. For simplicity, we will ignore the notation ε and use $\mathbf{E}$, $\mathbf{A}$, $F(\mathbf{u})$ to represent $\mathbf{E}_\varepsilon$, $\mathbf{A}_\varepsilon$ and $F_\varepsilon(\mathbf{u})$ respectively. This leads to a model referred to as the *Full Order Model* (FOM), represented by equation (1.1):

$$\mathbf{E}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{A}\mathbf{u} + F(\mathbf{u}), \tag{1.1}$$

The FOM has very large state-space dimension, i.e. $n \gg 1$, since it originates from the spacial discretization of a PDE system. The objective is to find a low-dimensional representation of the FOM, known as the *Reduced Order Model* (ROM), represented

*The work was partially supported by NSFC 11971469.

† LSEC, ICMSEC, NCMIS, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China, tangtianyou@lsec.cc.ac.cn.

‡ LSEC, ICMSEC, NCMIS, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China, xmxu@lsec.cc.ac.cn. Corresponding author.

by equation (1.2):

$$\tilde{\mathbf{E}}\frac{\mathrm{d}\tilde{\mathbf{u}}}{\mathrm{d}t} = \tilde{\mathbf{A}}\tilde{\mathbf{u}} + \tilde{F}(\tilde{\mathbf{u}}). \tag{1.2}$$

Here, $\tilde{\mathbf{u}} \in \mathbb{R}^r$, $\tilde{\mathbf{E}}, \tilde{\mathbf{A}} \in \mathbb{R}^{r\times r}$, $\tilde{F} : \mathbb{R}^r \to \mathbb{R}^r$, and $r \ll n$. This model is called *Reduced Order Model* (ROM). There exists a function $g : \mathbb{R}^r \to \mathbb{R}^n$ that reconstructs $\mathbf{u}$ from $\tilde{\mathbf{u}}$, for example $g(\tilde{\mathbf{u}}) = \mathbf{G}\tilde{\mathbf{u}}$, where $\mathbf{G} \in \mathbb{R}^{n\times r}$.

There are numerous model reduction methods to derive the ROM in the literature [5]. One category of these methods is the sampling-based approach, which involves constructing low-order models by sampling the parameter space or time domain. Commonly used methods in this category include the Proper Orthogonal Decomposition (POD) method [47], the Reduced Basis method [39], and techniques based on tensor analysis [44, 32]. Another category of methods is motivated by the system's dynamical behavior or theoretical knowledge of a specific model. For example, the balanced truncation method [21] can ensure system stability and provides a priori error estimation. It has been successfully applied to nonlinear control systems [28]. The Loewner interpolation framework has been employed for parameter-independent input and output systems [33]. Furthermore, there has been recent research in model reduction methods that combine machine learning approaches [6, 9].

In our work, we primarily focus on the POD method. This method has been extensively studied and successfully applied to various problems over the past few decades [30, 41, 40, 38]. The POD method is based on sampling, which emphasizes the need to estimate the approximation of the sample manifold and the true solution manifold. The optimal r-dimensional subspace approximation of the manifold $\mathcal{M}$ is characterized by the Kolmogorov N-width [34, 22]. Assuming $\mathcal{M}$ is a subset of H where H is some Banach or Hilbert space with norm $|\cdot|_H$. The Kolmogorov N-width of $\mathcal{M}$ is defined as

$$d_N(\mathcal{M}) := \inf_{\dim Y = N} \sup_{\mathbf{u}\in\mathcal{M}} |\mathbf{u} - P_Y\mathbf{u}|_H,$$

where Y is a N-dimensional linear subspace of H and P_Y denotes the projection onto Y. The efficiency of the POD method relies on the decay rate of the Kolmogorov N-width with respect to N [46, 20]. However, if the solution manifold of a PDE has a slow-decaying width, such as in convection-dominated problems, the standard POD method may not provide satisfactory results, necessitating the development of specialized techniques [37, 35, 36].

In addition to the slow decay of the Kolmogorov N-width, the presence of nonlinear terms in a dynamic system poses additional challenges. This is because the standard POD method may require full order computations when dealing with these terms. Fortunately, there are several techniques available to handle the nonlinearity in model reduction methods. One approach, proposed by Benner et al. [4], is the design of a two-sided projection method specifically tailored for nonlinear terms with quadratic form. Another widely used technique is the empirical interpolation method (EIM) and its discrete form, known as the Discrete Empirical Interpolation Method (DEIM) [2, 10]. These methods utilize a linear combination of low-dimensional basis functions to approximate the nonlinear terms. For a detailed analysis of the error estimation in DEIM, refer to Chaturantabut's work [11]. Building upon the DEIM technique, Drmač et al. introduced an algorithm framework that incorporates a new selection operator called the qDEIM technique [14]. This technique further enhances the accuracy of the approximation.

In this paper, our objective is to develop an efficient model reduction method for nonlinear partial differential equations involving small parameters. To illustrate our approach, we focus on the Allen-Cahn equation, which is commonly used to model phase transitions in material sciences [1]. This equation has found applications in various fields such as fluid dynamics [42, 26], image processing [3], and more. Extensive literature exists on the Allen-Cahn equation, covering aspects such as asymptotic and rigorous analysis, numerical methods, and diverse applications [16, 13, 17, 43, 31, 15]. The equation includes a parameter, denoted by ε, which governs the width of the interface between different phases. As ε becomes smaller, finding accurate numerical solutions becomes increasingly difficult. Additionally, the decay rate of the Kolmogorov N-width for the discrete solution manifold associated with the equation may be remarkably slow, posing a significant obstacle for model reduction techniques.

Our method revolves around the concept of identifying and learning slow latent variables, which allows us to perform model reduction on the transformed system. Specifically, for the Allen-Cahn equation, we conduct asymptotic analysis and leverage the leading order solution to derive an explicit variable transformation. The slow variable exhibits superior regularity compared to the phase field function in the Allen-Cahn equation. Subsequently, we employ a POD method in conjunction with a qDEIM technique to construct a reduced model for the transformed equation. Through numerical experiments, we demonstrate that the transformed system exhibits a significantly faster decaying Kolmogorov N-width, resulting in a highly efficient model reduction approach compared to the original equation. Remarkably, this method's natural and straightforward idea can be extended to other nonlinear PDEs, such as the convection equation and the Burgers equation.

The paper is organized as follows. Section 2 introduces the POD-qDEIM method for nonlinear model reduction, highlighting its key steps and principles. In Section 3, we present a general framework for the transformed model reduction method for second order quasi-linear PDEs with small parameters. In particular, we utilize the 2D Allen-Cahn equation as a model problem, elucidating our motivation and outlining the methodology employed to derive the slow variable. To validate the efficacy of our approach, Section 4 presents numerical results for the model reduction method for the Allen-Cahn equation in various cases. Additionally, Section 5 shows the application of our method to other equations. Finally, in Section 6, we provide conclusions and discuss potential avenues for future research and development.

2. Preliminary: POD-qDEIM Framework.

We utilize the POD method as a basic tool to establish the reduced order model. To overcome the difficulties caused by the nonlinearity of the dynamic system, we apply the DEIM technique and its variant, qDEIM. We briefly introduce the widely used techniques below.

2.1. Proper orthogonal decomposition.

Consider a FOM given by equation (1.1). Let $\mathbf{u}(t)$ denote the solution of the FOM at each time t. Suppose the solutions in a time interval $[0, T]$ form a manifold $\mathcal{M} \subset \mathbb{R}^n$ with $n \gg 1$. By sampling the time variable t over the time interval, we obtain a set of snapshots $\mathbf{u}_1, \ldots, \mathbf{u}_M \in \mathbb{R}^n$, which are the solutions of the FOM computed at different time instances $t_1, \ldots, t_M$. Let $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_M] \in \mathbb{R}^{n \times M}$. The linear space $\operatorname{span}(\mathbf{U})$ spanned by these snapshots may not directly coincide with the solution manifold of the dynamic system. However, it serves as a good representation of the manifold if the number of snapshots is sufficiently large. The POD method aims at determining an r-dimensional subspace within $\operatorname{span}(\mathbf{U})$ and find an approximate solution $\tilde{\mathbf{u}}$ within this subspace.

Suppose that $\mathbf{U}$ admits the singular value decomposition,

$$\mathbf{U} = \mathbf{X}\boldsymbol{\Sigma}\mathbf{Y}^T,$$

where $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_M] \in \mathbb{R}^{n\times M}$, $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1, \cdots, \sigma_M) \in \mathbb{R}^{M\times M}$ satisfies $\sigma_1 \geq \cdots \geq \sigma_M \geq 0$, $\mathbf{Y} \in \mathbb{R}^{M\times M}$. The POD method consist in obtaining a subspace range($\mathbf{V}$), where $\mathbf{V}$ is defined as the first r columns of $\mathbf{X}$ corresponding to the r largest singular values of $\mathbf{U}$. All columns of $\mathbf{V}$ compose a set of *POD basis*. The approximation error of the subspace range($\mathbf{V}$) to range($\mathbf{U}$) can be estimated by the sum of the squares of the singular values corresponding to those left singular vectors not included in the POD basis, i.e.

$$\left\|\mathbf{U} - \mathbf{V}\mathbf{V}^T\mathbf{U}\right\|_F^2 = \sum_{i=1}^{M} \left\|\mathbf{u}_i - \mathbf{V}\mathbf{V}^T\mathbf{u}_i\right\|_2^2 = \sum_{i=r+1}^{M} \sigma_i^2, \tag{2.1}$$

where $\|\cdot\|_F$ is the Frobenius norm.

Assuming that $\mathbf{u} \approx \mathbf{V}\tilde{\mathbf{u}}$, where $\tilde{\mathbf{u}} \in \mathbb{R}^r$, we substitute this approximation into the equation (1.1) and multiply the left side by $\mathbf{V}^T$. This allows us to obtain a ROM as follows:

$$\mathbf{V}^T\mathbf{E}\mathbf{V}\frac{\mathrm{d}\tilde{\mathbf{u}}}{\mathrm{d}t} = \mathbf{V}^T\mathbf{A}\mathbf{V}\tilde{\mathbf{u}} + \mathbf{V}^T F(\mathbf{V}\tilde{\mathbf{u}}).$$

Define $\tilde{\mathbf{E}} := \mathbf{V}^T\mathbf{E}\mathbf{V}$, $\tilde{\mathbf{A}} := \mathbf{V}^T\mathbf{A}\mathbf{V}$ and $\tilde{F}(\tilde{\mathbf{u}}) := \mathbf{V}^T F(\mathbf{V}\tilde{\mathbf{u}})$. The equation can be written as

$$\tilde{\mathbf{E}}\frac{\mathrm{d}\tilde{\mathbf{u}}}{\mathrm{d}t} = \tilde{\mathbf{A}}\tilde{\mathbf{u}} + \tilde{F}(\tilde{\mathbf{u}}). \tag{2.2}$$

Notice that both $\tilde{\mathbf{E}}, \tilde{\mathbf{A}} \in \mathbb{R}^{r\times r}$ can be pre-computed in many cases when ε is fixed or when $\mathbf{A}$ and $\mathbf{E}$ are homogeneous with respect to ε, e.g. $\mathbf{A} = \varepsilon^{k_0}\mathbf{A}_0$, $\mathbf{E} = \varepsilon^{k_1}\mathbf{E}_0$ with $k_0, k_1 \geq 0$, $\mathbf{A}_0$ and $\mathbf{E}_0$ independent of ε. This leads to a standard POD algorithm(see in Appendix A.).

2.2. Reduction of the nonlinear term. Notice that the evaluation of $\tilde{F}(\tilde{\mathbf{u}}) = \mathbf{V}^T F(\mathbf{V}\tilde{\mathbf{u}})$ in (2.2) contains high order computation due to the nonlinearity of F. We now present two popular techniques, DEIM and qDEIM, for handling the nonlinear term. For more details, we refer the reader to [10, 14].

In DEIM it is assumed that the nonlinear function F can be approximated by

$$F(\mathbf{w}) \approx \mathbf{D}C(\mathbf{w}),$$

where $\mathbf{w} \in \mathbb{R}^n$, $\mathbf{D} \in \mathbb{R}^{n\times m}$, and $C : \mathbb{R}^n \to \mathbb{R}^m$ is a nonlinear function with $m \ll n$. To obtain the low-dimensional nonlinear function C, we equate m rows from both sides of the equation, i.e.

$$\mathbf{P}^T F(\mathbf{w}) = \mathbf{P}^T\mathbf{D}C(\mathbf{w}),$$

where $\mathbf{P} = [\mathbf{e}_{id_1}, \cdots, \mathbf{e}_{id_m}] \in \mathbb{R}^{n\times m}$ is called *selection operator*. Here $\mathbf{e}_{id_k}$ is the id_k-th column of the identity matrix $\mathbf{I}_n \in \mathbb{R}^{n\times n}$. Assuming $\mathbf{P}^T\mathbf{D}$ is non-singular, then $C(\mathbf{w})$ can be uniquely determined by

$$C(\mathbf{w}) = (\mathbf{P}^T\mathbf{D})^{-1}\mathbf{P}^T F(\mathbf{w}). \tag{2.3}$$

Define $\bar{\mathbf{D}} := \mathbf{D}(\mathbf{P}^T\mathbf{D})^{-1}$, $\bar{C}(\mathbf{w}) := \mathbf{P}^T F(\mathbf{w})$, then

$$F(\mathbf{w}) \approx \mathbf{D}C(\mathbf{w}) = \underline{\mathbf{D}(\mathbf{P}^T\mathbf{D})^{-1}}\,\underline{\mathbf{P}^T F(\mathbf{w})} = \bar{\mathbf{D}}\bar{C}(\mathbf{w}). \tag{2.4}$$

Therefore, the approximation of $F(\mathbf{w})$ involves two steps: compute the *DEIM basis* $\mathbf{D}$ and identify the indices $\{id_1, \cdots, id_m\}$.

By defining $\mathbf{F} := [F(\mathbf{w}_1), \cdots, F(\mathbf{w}_N)] \in \mathbb{R}^{n\times N}$ where $\mathbf{w}_1, \cdots, \mathbf{w}_N$ are the samples over the domain of F, the DEIM basis $\mathbf{D}$ is obtained by choosing the first m left singular vectors of $\mathbf{F}$ similarly to the POD process. The indices are determined iteratively based on the choice of $\mathbf{D}$.

The qDEIM improves the error bound of DEIM [10, Lemma 3.2] by using a new selecting operator strategy [14, Theorem 2.1], and shares the same DEIM basis matrix $\mathbf{D}$ with the original DEIM. In this method, $\{id_1, \cdots, id_m\}$ are selected as the first m indices of pivoted QR factorization of $\mathbf{D}^T$. That is to choose indices corresponding to the columns of the leading submatrix of a factorized matrix $\mathbf{D}^T$. The details of the DEIM algorithm and qDEIM algorithm are given in Appendix A.

Combining the POD method and the qDEIM technique, the reduced order model of (1.1) is of this form

$$\underline{\mathbf{V}^T\mathbf{E}\mathbf{V}}\frac{\mathrm{d}\tilde{\mathbf{u}}}{\mathrm{d}t} = \underline{\mathbf{V}^T\mathbf{A}\mathbf{V}}\tilde{\mathbf{u}} + \underline{\mathbf{V}^T\bar{\mathbf{D}}}\bar{C}(\mathbf{V}\tilde{\mathbf{u}}),$$

where $\mathbf{V}$ is the POD basis matrix, $\bar{\mathbf{D}}$ and $\bar{C}$ are defined as in (2.4). Let $\tilde{\mathbf{B}} := \mathbf{V}^T\bar{\mathbf{D}}$, the model can be rewritten as

$$\tilde{\mathbf{E}}\frac{\mathrm{d}\tilde{\mathbf{u}}}{\mathrm{d}t} = \tilde{\mathbf{A}}\tilde{\mathbf{u}} + \tilde{\mathbf{B}}\bar{C}(\mathbf{V}\tilde{\mathbf{u}}). \tag{2.5}$$

Here $\bar{C}(\mathbf{V}\tilde{\mathbf{u}})$ contains at most m nonlinear functions and does not require computing $\mathbf{V}\tilde{\mathbf{u}}$ in every time step.

The error estimate (2.1) implies that the approximation error of the POD method depends on the decay rate of the singular values. The faster the singular values decay, the better the approximation of the original snapshot space for a given reduced order r. The decay property of the singular values is characterized by the Kolmogorov N-width. For many partial differential equations with small parameters, the solution may have small transition layers. The numerical solution may correspond to a slowly decaying Kolmogorov N-width, which poses a major challenge for deriving a ROM for such problems. In the following, we will present a model reduction method for problems with small parameters by learning the intrinsic slow variable in the system which corresponds to fast decaying Kolmogorov N-width.

3. Transformed model reduction method.

In this section, we will introduce the transformed model reduction method for a general nonlinear partial differential equation with small parameters. We first introduce the main idea of the method and the algorithm. Then we apply the method to the Allen-Cahn equation.

3.1. A general framework.

We consider a general second order quasi-linear partial differential equation in a domain $\Omega \subset \mathbb{R}^d$ as follows,

$$u_t = \varepsilon\nabla\cdot(A(\mathbf{x})\nabla u) + b(\mathbf{x}, u)\cdot\nabla u + \varepsilon^{-1} f(\mathbf{x}, u), \tag{3.1}$$

where $u : \Omega\times[0, T] \to \mathbb{R}$ is a scalar unknown function, the coefficiences $A : \Omega \to \mathbb{R}^{d\times d}$, $b : \Omega\times\mathbb{R} \to \mathbb{R}^d$ and $f : \Omega\times\mathbb{R} \to \mathbb{R}$ are given functions. We suppose that ε is a small positive parameter. Notice that the last two terms in the right hand side of (3.1) may be nonlinear with respect to u. The equation covers some widely used models. For example, when $A(\mathbf{x})$ is equal to the identity matrix, $b(\mathbf{x}, u) = 0$ and $f(\mathbf{x}, u) = u - u^3$, (3.1) is reduced to the standard Allen-Cahn equation. When $A(\mathbf{x})$ is the identity matrix, $b(\mathbf{x}, u) = ub_0$ with $b_0 \in \mathbb{R}^d$ and $f = 0$, this is the Burgers equation. If $A(\mathbf{x}) = 0$, $b(\mathbf{x}, u) = b_0$ and $f(\mathbf{x}, u) = 0$, the equation is reduced to a linear convection equation.

When ε is small, the solution u of (3.1) may exhibit an evolving sharp transition layer of order $O(\varepsilon)$ in thickness. In this case, the solution of the equation lies on a manifold with slowly decaying Kolmogorov N-width. Applying the standard model reduction method from the previous section directly to the equation is very inefficient. To improve the efficiency of the model reduction applied to the equation (3.1) with sharp transition layer, we seek a slow variable v in its dynamics. By transforming Eq. (3.1), we obtain an equation for v. We expect that the transformed equation has a faster decaying Kolmogorov N-width and is more amenable to model reduction.

Finding a suitable transformation for a nonlinear PDE is usually challenging [23]. We employ an asymptotic analysis method. The main idea is to find a transformation $u = \phi(v)$ such that the leading order approximation of v is independent of ε. In other words, we want to rewrite $v = v_0 + \varepsilon v_1 + \cdots$ and v_0 does not depend on ε.

Asymptotic analysis. We find the leading order approximation of u in regions far from the transition layer (outer expansion) and in the layer (inner expansion) separately by asymptotic analysis [7, 8]. Firstly, we consider the outer expansions far from the layer. Suppose that the solution u can be expanded with respect to ε as follows,

$$u = u_0 + \varepsilon u_1 + \cdots .$$

Substitute the expansion into (3.1) and equal the same orders. We obtain a series of equations,

$$\begin{aligned} O(\varepsilon^{-1}): &\qquad f(\mathbf{x}, u_0) = 0, \\ O(1): &\qquad u_{0,t} + b(\mathbf{x}, u_0) \cdot \nabla u_0 + \partial_u f(\mathbf{x}, u_0) u_1 = 0, \\ \cdots & \end{aligned}$$

When f and b are given, it is possible to derive the explicit form of the leading order term u_0. For simplicity, we assume there exist two different phases in the system. The values of u_0 in the two phases are denoted as u_0^+ and u_0^-, respectively.

We then consider the inner expansions near the transition layer between the two phases. Suppose the centering surface of the transition layer is given by $\Gamma(t)$.We define $d(\mathbf{x}, t)$ as the signed distance function from point $\mathbf{x}$ to $\Gamma(t)$. By its definition, we have $\mathbf{n} = \nabla d$ and $\kappa = \Delta d$, where $\mathbf{n}$ is the unit vector normal to Γ and κ is the curvature of Γ. We introduce an inner layer coordinate $\xi(\mathbf{x}, t) := d(\mathbf{x}, t)/\varepsilon$. We represent $u(\mathbf{x}, t)$ in the neighborhood of $\Gamma(t)$ by the function $\tilde{u}(\mathbf{x}, \xi, t)$. We express the derivatives of u as

$$\begin{aligned} \partial_t u &= \frac{1}{\varepsilon} \partial_t d(\mathbf{x}, t) \partial_\xi \tilde{u}(\mathbf{x}, \xi, t) + \partial_t \tilde{u}(\mathbf{x}, \xi, t), \\ \nabla u &= \frac{1}{\varepsilon} \partial_\xi \tilde{u}(\mathbf{x}, \xi, t) \mathbf{n} + \nabla_{\mathbf{x}} \tilde{u}(\mathbf{x}, \xi, t). \end{aligned}$$

The asymptotic expansion of $\tilde{u}$ is

$$\tilde{u}(\mathbf{x}, \xi, t) \sim \tilde{u}_0(\mathbf{x}, \xi, t) + \varepsilon \tilde{u}_1(\mathbf{x}, \xi, t) + \varepsilon^2 \tilde{u}_2(\mathbf{x}, \xi, t) + \cdots .$$

We substitute the expansion into Eq. (3.1). By considering the leading order terms, we obtain

$$O(\varepsilon^{-1}): \qquad (\mathbf{n}^T A(\mathbf{x})\mathbf{n})\tilde{u}_{0,\xi\xi} + (b(\mathbf{x}, \tilde{u}_0)\cdot\mathbf{n})\tilde{u}_{0,\xi} + f(\mathbf{x}, \tilde{u}_0) = 0. \tag{3.2}$$

By matching the outer and inner expansions, we have

$$\lim_{\xi\to\pm\infty} \tilde{u}_0 = u_0^{\pm}.$$

We assume that there exists a unique solution $\tilde{u}_0 = \tilde{\phi}(\xi)$ to the equation (3.2) together with the matching condition.

Transformation and Discretization in space. Motivated by the asymptotic analysis results, we introduce the following transformation that

$$u = \phi(v) := \tilde{\phi}(\frac{v}{\varepsilon}).$$

We substitute the transformation into Eq. (3.1) to derive a model for v,

$$v_t = \varepsilon\nabla\cdot(A(\mathbf{x})\nabla v) + \Big(\varepsilon(\phi')^{-1}\nabla\cdot(\phi' A(\mathbf{x})) + b(\mathbf{x}, \phi(v))\Big)\cdot\nabla v + (\varepsilon\phi')^{-1} f(\mathbf{x}, \phi(v)). \tag{3.3}$$

We expect that the solution of the transformed equation (3.3) has a much faster decaying Kolmogorov width compared to the original model (3.1) for u.

The equation (3.3) can be discretized by standard numerical methods, like the finite difference method etc. We obtain the following discrete model for v,

$$\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = \mathbf{A}_v\mathbf{v} + F_v(\mathbf{v}), \tag{3.4}$$

where $\mathbf{A}_v$ arises from $\varepsilon\nabla(A(\mathbf{x})\nabla v)$ and $F_v(\mathbf{v})$ is obtained from the other terms.

Finally, we apply the POD-qDEIM method introduced in the previous section to perform model reduction for the equation (3.4). This further leads to a model reduction method for the equation (3.1). In summary, the general framework for the transformed model reduction method can be described in the following Algorithm 3.1.

Algorithm 3.1 A POD-qDEIM method based on transformation

Input: model of u;
Output: reduced approximation $\tilde{u}$;
1: Find the slow variable transformation $u = \phi(v)$ by asymptotic analysis;
2: Derive a transformed model for v and discretize;
3: Compute $\tilde{v}$ by applying POD-qDEIM(Algorithm A1 and Algorithm A3) on the discrete model of v;
4: $\tilde{u} := \phi(\tilde{v})$.

3.2. Application to the Allen-Cahn equation. To further demonstrate how we derive a ROM by transforming the PDE, we use the Allen-Cahn equation as a model problem. The equation is given by

$$u_t = \Delta u + \frac{1}{\varepsilon^2} f(u), \qquad (\mathbf{x}, t) \in \Omega\times[0, T]. \tag{3.5}$$

where $f(u) = u - u^3$, $\Omega \subset \mathbb{R}^2$ is a two-dimensional domain, and ε is a small parameter. The Allen-Cahn equation has many applications in phase transitions, wetting problems and image processing (c.f. [12, 31, 3]). Notice that we have re-scaled the time variable in (3.5) in comparison to (3.1) so that the equation approximates a standard mean curvature flow when ε goes to zero.

We seek a leading approximation for $u(\mathbf{x},t)$ by asymptotic analysis. Let $\Gamma(t) \subset \Omega$ be the zero-level set of the solution $u(\mathbf{x},t)$ of the equation at time t. It is known that u has a sharp transition layer around $\Gamma(t)$.

We first consider the outer expansions. We assume that u has the following asymptotic expansion with respect to ε in $\Omega^{\pm}$ away from $\Gamma(t)$:

$$u(\mathbf{x},t) \sim u_0^{\pm}(\mathbf{x},t) + \varepsilon u_1^{\pm}(\mathbf{x},t) + \cdots, \qquad \text{in } \Omega^{\pm}.$$

We substitute the expansion into the equation (3.5). Comparing the leading order terms on both sides, namely the $O(\varepsilon^{-2})$ terms, we obtain

$$u_0^{\pm} - (u_0^{\pm})^3 = 0,$$

which implies $u_0(\mathbf{x},t)^{\pm} = \pm 1$ for $\mathbf{x} \in \Omega^{\pm}$.

We then consider the solution near $\Gamma(t)$. Let $d(\mathbf{x},t)$ be the signed distance function from point $\mathbf{x}$ to $\Gamma(t)$ and introduce an inner layer coordinate $\xi(\mathbf{x},t) := d(\mathbf{x},t)/\varepsilon$. We represent $u(\mathbf{x},t)$ in the neighborhood of $\Gamma(t)$ by the function $\tilde{u}(\mathbf{x},\xi,t)$. By inner expansions, the leading order of $\tilde{u}$ satisfies

$$\partial_{\xi\xi}\tilde{u}_0 + \tilde{u}_0 - (\tilde{u}_0)^3 = 0. \tag{3.6}$$

Matching the leading order term of the outer expansion and the inner expansion gives

$$\lim_{\xi \to \pm\infty} \tilde{u}_0(\mathbf{x},\xi,t) = \pm 1. \tag{3.7}$$

From (3.6) and (3.7), noticing $\tilde{u}(\mathbf{x},\xi,t) = 0$ when $\xi = 0$, we derive an ordinary differential equation in terms of ξ satisfied by $\tilde{u}_0$,

$$\begin{cases} \Phi''(\xi) = -(\Phi - \Phi^3), \ -\infty < z < +\infty, \\ \lim\limits_{\xi \to \pm\infty} \Phi = \pm 1, \\ \Phi(0) = 0. \end{cases}$$

The unique solution of the equation is

$$\Phi(\xi) = \tanh(\frac{1}{\sqrt{2}}\xi),$$

which implies $\tilde{u}_0(\mathbf{x},\xi,t) = \tanh(\frac{\xi}{\sqrt{2}})$.

The leading order term of u in Ω can be obtained by adding the outer and inner approximations together and subtracting the common part:

$$u_0(\mathbf{x},t) = u_0^{\pm}(\mathbf{x},t) + \tilde{u}_0(\mathbf{x}, \frac{d(\mathbf{x},t)}{\varepsilon}, t) - \lim_{\mathbf{x} \to \Gamma(t)} u_0^{\pm}(\mathbf{x},t) = \tanh(\frac{d(\mathbf{x},t)}{\sqrt{2}\varepsilon}). \tag{3.8}$$

Here $d(\mathbf{x},t)$ is a signed distance function to $\Gamma(t)$ that does not depend on ε. The next-order expansion analysis will reveal that the evolution of $d(\mathbf{x},t)$ corresponds to a mean curvature flow. We omit the analysis here because the expression (3.8) suffices to define a transformation for the Allen-Cahn equation.

Motivated by the above asymptotic results, we choose $v = \sqrt{2}\varepsilon \tanh^{-1}(u)$ as a slow variable. Note that the leading order approximation of v is the signed distance function $d(\mathbf{x},t)$ which is independent of ε. Compared to u, the numerical approximation to v should correspond to a faster singular value decaying snapshot matrix. By applying the transformation

$$u(\mathbf{x},t) = \phi(v(\mathbf{x},t)) := \tanh\Big(\frac{v(\mathbf{x},t)}{\sqrt{2}\varepsilon}\Big), \tag{3.9}$$

into (3.5), we obtain a full order model of the new variable v:

$$v_t = \Delta v + \tfrac{\sqrt{2}}{\varepsilon}\phi(v)(1-|\nabla v|^2), \quad \mathbf{x} \in \Omega,\ t \in [0,T]. \tag{3.10}$$

After solving this equation, we can use (3.9) to obtain $u(x,t)$. In the following, we will develop a numerical discretization to (3.10) and perform model reduction on the discrete problem.

We then discretize the modified equation (3.10) in space. The discrete equation has a similar structure to the Allen-Cahn equation (3.5), which also includes a small parameter ε. Many numerical methods for the Allen-Cahn equation, e.g. the finite element method [17, 18], the finite difference method [45, 29] and the spectral method [43], etc, can be adapted to solve the transformed equation. Moreover, the second term is of order $O(\varepsilon^{-1})$ instead of $O(\varepsilon^{-2})$ as in (3.5). This makes the problem (3.10) easier to solve numerically. Since we aim at developing a reduced model for the equation (3.10), we use the finite difference method for simplicity.

Assume that $\Omega = [-1,1]\times[-1,1]$. We uniformly discretize the domain as follows. We introduce a partition along two coordinates, $-1 = x_1 < \cdots < x_K = 1$ and $-1 = y_1 < \cdots < y_K = 1$ with mesh size $h := \frac{2}{K-1}$. This induces a two dimensional partition for Ω with grid points (x_i, y_j), $1 \le i,j \le K$.

Suppose that $v_{i,j}(t)$ approximates $v(x_i, y_j, t)$ for $i,j = 1,\cdots,K$. We can use a finite difference method to discretize the equation (3.10). The Laplacian Δv is discretized by a second-order central difference scheme

$$-\Delta_h(v_{i,j}) \quad = \tfrac{-v_{i-1,j}+2v_{i,j}+v_{i+1,j}}{h^2} + \tfrac{-v_{i,j-1}+2v_{i,j}-v_{i,j+1}}{h^2}.$$

Alternatively, we can use a fourth-order central difference to discretize the Laplacian, which has better numerical properties in solving the original Allen-Cahn equation (3.5), which serves as a reference solution in our numerical experiments. The first-order derivatives in (3.10) are discretized by a second-order central difference

$$(v_{i,j})_x = \frac{v_{i+1,j}-v_{i-1,j}}{2h}, \quad (v_{i,j})_y = \frac{v_{i,j+1}-v_{i,j-1}}{2h}.$$

Let $\mathbf{v}(t) := [v_{1,1}(t)\cdots, v_{1,K}(t), v_{2,1}(t), \cdots, v_{K,K}(t)]^T$. Then the semi-discrete scheme for (3.10) is given by

$$\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = \mathbf{A}_v\mathbf{v} + F_v(\mathbf{v}), \tag{3.11}$$

where $\mathbf{A}_v$ arises from Δv and $F_v(\mathbf{v})$ is obtained from the second term in the right hand side of (3.10). This is a full order model of the partial differential equation (3.10). We finally apply the POD-qDEIM(Algorithm A1 and Algorithm A3) to (3.10). The numerical results will be illustrated in next section.

4. Numerical experiments for the Allen-Cahn equation. In this section, we present some numerical experiments that demonstrate the superior performance of the model reduction method based on the transformation compared to the direct model reduction of the equation (3.5) for u.

To compare with the reference solution for the original equation (3.5), we also introduce a finite difference scheme for the equation. Let

$$\mathbf{u}(t) := [u_{1,1}(t) \cdots, u_{1,K}(t), u_{2,1}(t), \cdots, u_{K,K}(t)]^T.$$

The discrete model for (3.5) is given by

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{A}_u \mathbf{u} + F_u(\mathbf{u}), \tag{4.1}$$

where $\mathbf{A}_u$ arises from the Laplace term Δu and $F_u(\mathbf{u})$ is obtained from the nonlinear term $\frac{1}{\varepsilon^2}(u - u^3)$. Note that $\mathbf{A}_u$ may differ from $\mathbf{A}_v$ in (3.11) even when we use the same finite difference scheme for the Laplace operator, since the boundary condition may vary for u and v.

To solve the equation (3.5), we need set a boundary condition and an initial boundary condition. In the first experiment, we choose a natural boundary condition $\frac{\partial u}{\partial \mathbf{n}} = 0$ on $\partial\Omega$ and an initial boundary condition $u(x, y, 0) = u_0(x, y)$, where

$$u_0(x, y) = \tanh(\frac{\sqrt{x^2 + y^2} - 0.6}{\sqrt{2}\varepsilon}). \tag{4.2}$$

The initial condition $v_0(x, y)$ can be derived directly from the transformation that

$$v_0(x, y) = \phi^{-1}(u_0(x, y)) = \sqrt{x^2 + y^2} - 0.6.$$

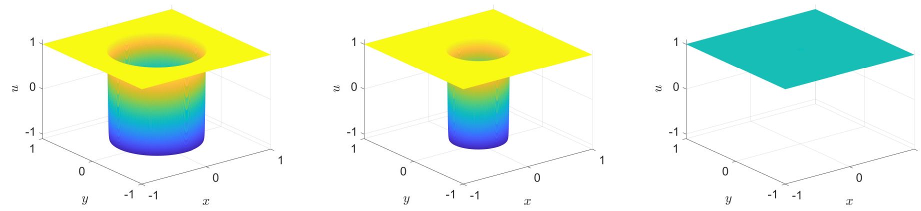
In principle, the boundary condition for $v(x, y)$ can be derived similarly by using the relation $\nabla u = \frac{1}{\sqrt{2}\varepsilon}(1 - \tanh^2(\frac{v}{\sqrt{2}\varepsilon}))\nabla v$. When ε is small, $\frac{1}{\sqrt{2}\varepsilon}(1 - \tanh^2(\frac{v}{\sqrt{2}\varepsilon})) \approx 0$ and v approximates a signed distance function so that we can simply choose $|\nabla v| = 1$ on $\partial\Omega$ in the numerical experiments.

It is well-known that the transition layer of the solution of the Allen-Cahn equation evolves with time, that is a good approximation to a mean curvature flow[16, 19]. Under the initial condition (4.2), the zero level set of u can be described approximately by a shrinking circle with radius

$$r = \sqrt{0.36 - 2t}. \tag{4.3}$$

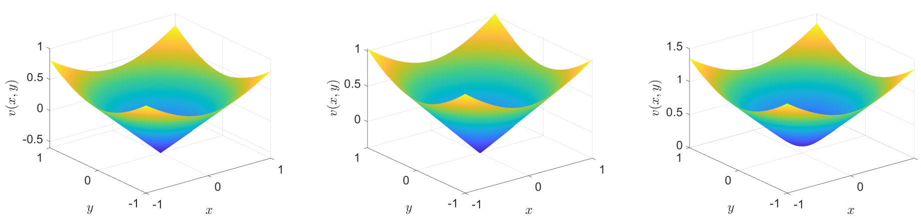
We solve the Allen-Cahn equation (3.5) using the finite difference scheme (4.1). The ordinary differential equation (4.1) is solved by a fourth-order Runge-Kutta method. In the test, we take $T = 0.181$, $K = 1000$, $\varepsilon = 0.005$, and the time step $\Delta t = 5e - 7$. The numerical solution is shown in Figure 1, where we find that the hole disappears almost at $t = 0.18$, just as shown in (4.3). This indicates that the full order model (4.1) is solved correctly.

Similarly, we solve the transformed equation (3.10) using the finite difference scheme (3.11). We choose the same numerical parameters as for u. The numerical solution for $\mathbf{v}$ and the corresponding phase field function $U^v := \phi(\mathbf{v}) = \tanh(\frac{\mathbf{v}}{\sqrt{2}\varepsilon})$ are shown in Figure 2. We can see that $\mathbf{v}$ is a good approximation to a signed distance function while $\phi(\mathbf{v})$ behaves similarly to the solution of the Allen-Cahn equation in Figure 1. This implies that we can use the transformed equation (3.10) instead of the original Allen-Cahn equation (3.5) in numerical simulations. Furthermore, since there

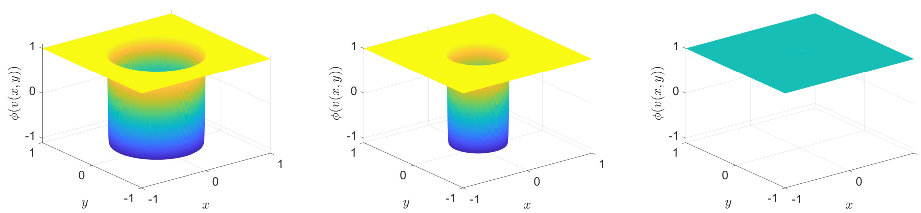


(a) t = 0 (b) t = 0.11 (c) t = 0.18

Figure 1: Numerical results of 2D Allen-Cahn equation. (a) The initial condition (4.2). (b) The reference solution at $t = 0.11$. (c) The reference solution at $t = 0.18$.



(a) $\mathbf{v}$ at $t = 0$ (b) $\mathbf{v}$ at $t = 0.11$ (c) $\mathbf{v}$ at $t = 0.18$

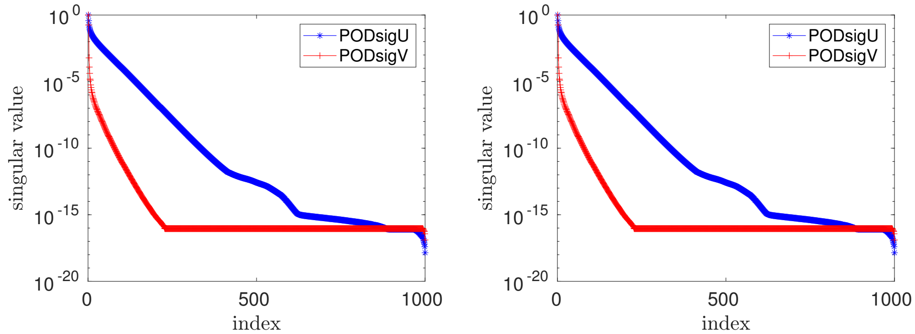


(d) $\phi(\mathbf{v})$ at $t = 0$ (e) $\phi(\mathbf{v})$ at $t = 0.11$ (f) $\phi(\mathbf{v})$ at $t = 0.18$

Figure 2: Numerical results of 2D Allen-Cahn equation. The solutions of V-FOM (3.11) and the corresponding transformed solutions $U^v = \phi(v)$ at different time instants.

is no sharp inner layer for the solution $\mathbf{v}$, we expect that the solution manifold will correspond to a faster decaying Kolmogorov N-width, which will be verified below.

In the following, we will check how the model reduction method (Algorithm 3.1) works well for the transformed equation. In comparison, we also apply a POD-qDEIM method to the original equation (3.5). For convenience, we introduce some notations here. Let U-FOM be the full order model (4.1) for $\mathbf{u}$, and U-ROM be its reduced order model. Similarly, let V-FOM and V-ROM denote the full order model (3.11) for $\mathbf{v}$ and its reduced order model, respectively. We denote the solution of U-FOM as U and the solution of U-ROM as U_{appr}. Furthermore, we transform $\mathbf{u} = \phi(\mathbf{v})$ to the solution of V-FOM to get an approximate solution U^v for the phase field function, i.e., U^v is the approximate FOM solution obtained by first solving the transformed original problem V-FOM to obtain v, then setting $U^v = \phi(v)$. Similarly, we denote



(a) singular values of POD matrices (b) singular values of DEIM matrices

Figure 3: Normalized singular values σ_i/σ_1 for the solution snapshots of the Allen-Cahn equation. (a) PODsigU: singular values of the POD snapshot matrix for *U-FOM* (4.1); PODsigV: singular values of the POD snapshot matrix for *V-FOM* (3.11). (b) DEIMsigU: singular values of the DEIM snapshot matrix for *U-FOM* (4.1); DEIMsigV: singular values of the DEIM snapshot matrix for *V-FOM* (3.11).

by U^v_{appr} the approximate solution of u by first solving the reduced model V-ROM, then transforming it back with ϕ. The relative error of the U-ROM is defined as

$$err(U_{appr}, U) := \frac{\|U_{appr} - U\|_F}{\|U\|_F}.$$

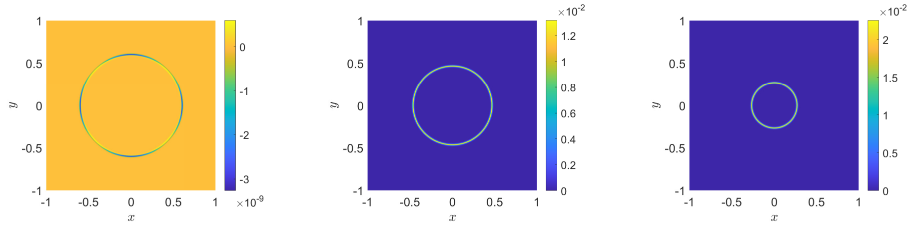
Here $\|\cdot\|_F$ denotes the Frobenius norm. Similarly, the relative error of the V-ROM is defined as

$$err(U^v_{appr}, U) := \frac{\|U^v_{appr} - U\|_F}{\|U\|_F}.$$

v

To derive the reduced order model, we take $M = 1000$ uniform samples $t_1, \cdots, t_M$ over the time interval $[0, T]$ and use $\mathbf{v}_i = \mathbf{v}(t_i)$ (or $\mathbf{u}_i = \mathbf{u}(t_i)$), $i = 1, \cdots, M$, to generate the POD and DEIM bases for V-ROM (or U-ROM). Figure 3(a) shows all the normalized singular values σ_i/σ_1 for the POD snapshot matrices $\mathbf{U} := [\mathbf{u}_1, \cdots, \mathbf{u}_M]$ and $\mathbf{V} := [\mathbf{v}_1, \cdots, \mathbf{v}_M]$. We can see that singular values decay much faster for the slow variable v than for u. This implies that the solution manifold for v has faster decaying Kolmogorov N-width, as expected. Figure 3(b) shows the normalized singular values for the DEIM snapshot matrices for u and v. Similarly, the slow variable v corresponds to faster decaying singular values.

Then we can generate a reduced order model for v by the POD-qDEIM method as described in Algorithm 3.1. Here we choose $r = 50$ for the POD basis and $m = 100$ for the DEIM basis in the V-ROM. The V-ROM is also solved numerically by a fourth-order Runge-Kutta method. By transforming the numerical solution of the V-ROM, we get an approximate solution U^v_{appr}. Figure 4 shows the error $U - U^v_{appr}$ at various time instants. We can see that the maximum error is of order $O(10^{-2})$, which occurs only near the transition layer of the phase field equation. We also show the radius of the zero level set for U^v_{appr} in Figure 5. The "exact radius" is given



(a) $t = 0$ (b) $t = 0.07$ (c) $t = 0.14$

Figure 4: Numerical errors of the transformed model reduction method for the 2D Allen-Cahn equation. We use 50 POD bases and 100 DEIM bases. (a) The pointwise error $U - U^v_{appr}$ at $t = 0$. (b) The pointwise error $U - U^v_{appr}$ at $t = 0.07$. (c) The pointwise error $U - U^v_{appr}$ at $t = 0.14$.

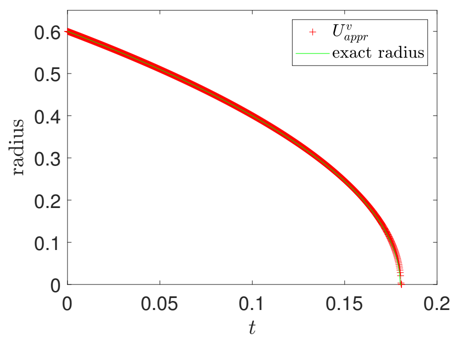


Figure 5: Numerical results of the 2D Allen-Cahn equation. We use 50 POD bases and 100 DEIM bases, and compare the exact radius and U^v_{appr}.

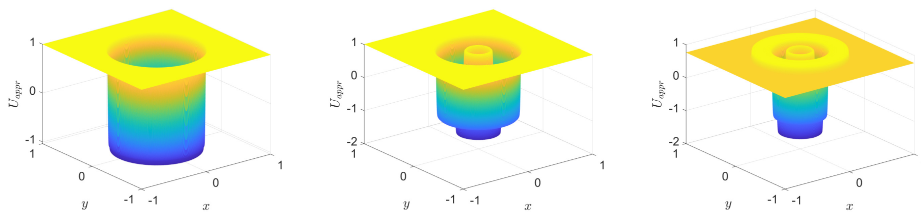
by the equation (4.3), corresponding to that for a mean curvature flow. We can see that the radius computed by the V-ROM is very close to the exact radius for all time $t \in [0, T]$.

For comparison, we also study the numerical behavior of the reduced order model U-ROM for the original Allen-Cahn equation. That is to apply the POD-qDEIM method to (4.1). We still choose $r = 50$ for the POD basis and $m = 100$ for the DEIM basis in the U-ROM. The numerical solution U_{appr} and the pointwise error $U - U_{appr}$ are shown in Figure 6. We can see that the U-ROM gives a totally wrong solution. This indicates that the V-ROM performs much better than the U-ROM. To make the comparison more clear, we show the relative error in Frobenius norm in Figure 7. It is easy to see that the relative error for the V-ROM for the transformed problem is much smaller than that for the U-ROM for the original Allen-Cahn equation.

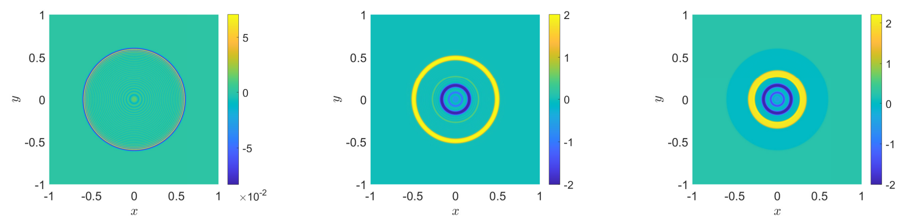
In the next experiment, we also use the V-ROM to solve problems with different initial conditions. We slightly change the initial values for u and consider

$$u_0(x, y) = \tanh\left(\frac{\sqrt{x^2 + y^2} - r_0}{\sqrt{2}\varepsilon}\right).$$

with $r_0 = 0.8$ and 0.4 respectively. We find that the V-ROM trained using the data



(a) $t = 0$ (b) $t = 0.07$ (c) $t = 0.14$



(d) $t = 0$ (e) $t = 0.07$ (f) $t = 0.14$

Figure 6: Numerical results of the 2D Allen-Cahn equation. We use 50 POD bases and 100 DEIM bases. (a) U_{appr} at $t = 0$. (b) U_{appr} at $t = 0.07$. (c) U_{appr} at $t = 0.14$. (d) The pointwise error $U - U_{appr}$ at $t = 0$. (e) The pointwise error $U - U_{appr}$ at $t = 0.07$. (f) The pointwise error $U - U_{appr}$ at $t = 0.14$.

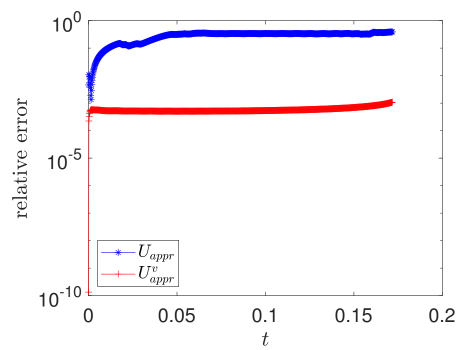
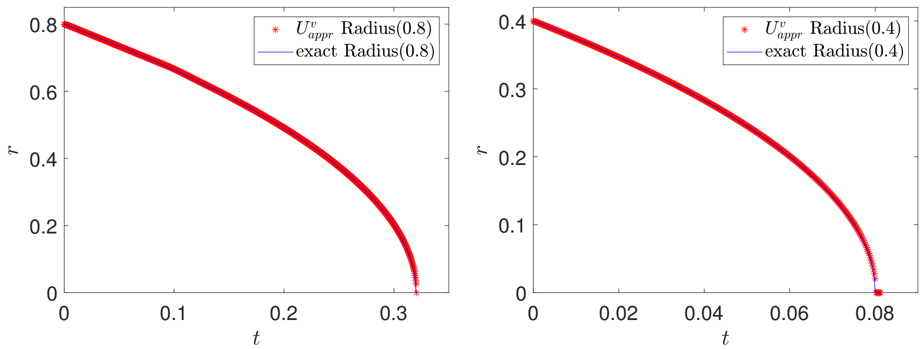


Figure 7: Relative errors in Frobenius norm of the 2D Allen-Cahn equation. We use 50 POD bases and 100 DEIM bases. U_{appr}: the relative errors of U_{appr} for $t \in [0, T]$; U^v_{appr}: the relative errors of U^v_{appr} for $t \in [0, T]$.

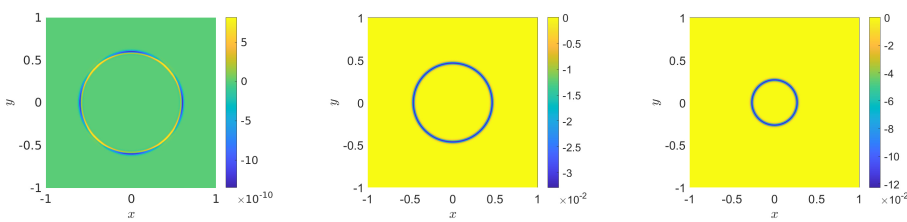
for $r_0 = 0.6$ still works well in the two cases. The reason is that the solution for the new initial values may still be in the solution manifold for v when we only change the radius of the transition layer. The radius of the zero level set of the solutions for the reduced order models are shown in Figure 8(a) and Figure 8(b), which agrees very well with the exact solutions obtained from the mean curvature flow.

In the next experiment, we study how the transformed model reduction method works when we change the value of the parameter ε. We choose the above V-ROM trained in the case of $\varepsilon = 0.005$. Then we simply change the value of ε to approximate



(a) radius of U^v_{appr} for new initial condition (b) radius of U^v_{appr} for new initial condition

Figure 8: Numerical results of the 2D Allen-Cahn equation. (a) We apply the reduced model to a new initial function with $r_0 = 0.8$, and compare the radius of the new exact solution and the reduced approximate solution. (b) We apply the reduced model to a new initial function with $r_0 = 0.4$, and compare the radius of the new exact solution and the reduced approximate solution.

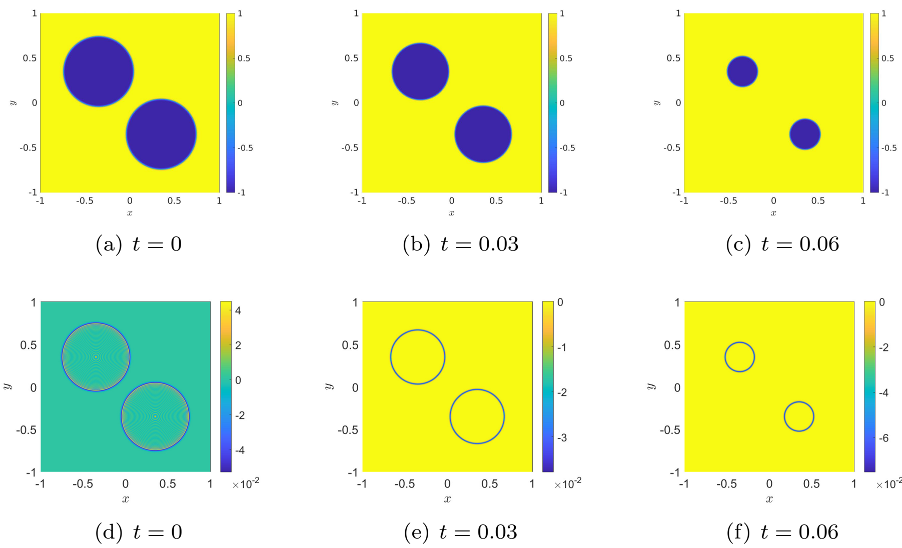


(a) $t = 0$ (b) $t = 0.07$ (c) $t = 0.14$

Figure 9: Numerical errors of the 2D Allen-Cahn equation with a different ε. We use 50 POD bases and 100 DEIM bases. (a) The pointwise error $\hat{U} - \hat{U}^v_{appr}$ at $t = 0$. (b) The pointwise error $\hat{U} - \hat{U}^v_{appr}$ at $t = 0.07$. (c) The pointwise error $\hat{U} - \hat{U}^v_{appr}$ at $t = 0.14$.

the Allen-Cahn equation with a different parameter. For example, we change ε from 0.005 to 0.01. The pointwise error $\hat{U} - \hat{U}^v_{appr}$ is shown in Figure 9, where $\hat{U}$ denotes the reference solution of the Allen-Cahn equation with $\varepsilon = 0.01$, and $\hat{U}^v_{appr}$ denotes the approximation solution of the transformed model reduction method. We see that the errors are relatively small. This implies that the V-ROM still works well when we change the value of ε. This reason might be that the leading order of the solution v of the transformed equation does not depend on ε.

Finally, we do experiments for the Allen-Cahn equation with two transition layers. Suppose that the initial value u_0 has two circular layers as shown in Figure 10. We develop a ROM by Algorithm 3.1. We choose the end time $T = 0.1$, $K = 1000$, and $\varepsilon = 0.005$. The numerical solution and the approximation errors for the transformed model reduction method are shown in Figure 10. We see that the numerical errors



(a) $t = 0$ (b) $t = 0.03$ (c) $t = 0.06$

(d) $t = 0$ (e) $t = 0.03$ (f) $t = 0.06$

Figure 10: Numerical results of the 2D Allen-Cahn equation with two layers. We use 50 POD bases and 100 DEIM bases. (a) The solution U^v_{appr} at $t = 0$. (b) The solution U^v_{appr} at $t = 0.03$. (c) The solution U^v_{appr} at $t = 0.06$. (d) The pointwise error $U - U^v_{appr}$ at $t = 0$. (e) The pointwise error $U - U^v_{appr}$ at $t = 0.03$. (f) The pointwise error $U - U^v_{appr}$ at $t = 0.06$.

are relatively small and locate mainly in the vicinity of layers. This is similar to the case with one transition layer.

5. Applications to other equations. Although we use the Allen-Cahn equation as a model problem to illustrate the model reduction method in the previous two sections, Algorithm 3.1 is quite general and applies to many other problems. In this section, we show applications of the algorithm to two other equations with slowly decaying Kolmogorov N-width. The two equations are a linear convection equation and a nonlinear Burgers equation.

5.1. A linear convection equation. We consider a linear convection problem as follows

$$u_t + au_x = 0,$$

with $u_0 = \psi(\frac{x}{\varepsilon})$. It is known that the solution of the equation is $u(x,t) = \psi(\frac{x-at}{\varepsilon})$. Here the initial function ψ is chosen as a pulse function such as

$$\psi(\frac{x}{\varepsilon}) = \exp\Big(-\frac{(x-x_0)^2}{\varepsilon}\Big),$$

where x_0 denotes the initial location of the pulse function. We can show that the solution manifold of the problem corresponds to slowly decaying Kolmogorov N-width when ε is small. This causes troubles for the standard POD method.

For this problem, we consider a function transformation $u = \psi(\frac{v(x,t)}{\varepsilon})$. We can see that v satisfies the same equation

$$v_t + av_x = 0,$$

with $v_0 = x$. Its solution is $v(x,t) = x - at$, which corresponds to a solution manifold with a basis set $\{1, x\}$. This implies a very narrow Kolmogorov N-width for the solution manifold. The standard POD method will work well for the transformed problem.

We remark that Algorithm 3.1 reduces to the method developed in [37] for the linear convection equation. We will not present numerical tests here and refer to [37] for numerical examples and detailed discussions.

5.2. Burgers equation. We then present an application of Algorithm 3.1 to a one-dimensional Burgers equation, which reads

$$\begin{cases} u_t = -uu_x + \varepsilon u_{xx}, & (x,t) \in [0,L] \times [0,T], \\ u_x(0,t) = 0,\ u_x(L,t) = 0 & t \in [0,T], \\ u(x,0) = u_0(x), & x \in [0,L]. \end{cases} \tag{5.1}$$

where ε is the viscosity coefficient. We choose the initial condition as

$$u_0(x) = \frac{1}{2}(1 - \tanh(\frac{x - x_0}{4\varepsilon})).$$

where x_0 is the position of the transition layer. According to [25], such an initial condition may lead to a traveling wave solution with a moving inner layer. We use $s(t)$, $s(0) = x_0$, to represent the position of the inner layer. The smaller the viscosity coefficient ε, the sharper the layer. When $\varepsilon = 0$, the Burgers equation reduces to a nonlinear hyperbolic equation and the inner layer becomes a shock with a jump solution.

We apply our method to the Burgers equation (5.1). We can also perform asymptotic analysis on the equation (see Appendix). The traveling wave solution of the Burgers equation motivates us to define a transformation as follows

$$u(x,t) = \phi(v(x,t)) := \frac{1}{2}(1 - \tanh\frac{v(x,t)}{4\varepsilon}). \tag{5.2}$$

Here $v(x,t)$ is a slow variable. Substituting the transformation (5.2) into (5.1), we get a model for the new variable v:

$$\begin{cases} v_t = -\frac{1}{2}(1 - \psi(v))v_x - \frac{1}{2}\psi(v)v_x^2 + \varepsilon v_{xx}, & (x,t) \in [0,L] \times [0,T], \\ v_x(0,t) = 0,\ v_x(L,t) = 0 & t \in [0,T], \\ v(x,0) = v_0(x), & x \in [0,L], \end{cases} \tag{5.3}$$

where $\psi(v) := \tanh(\frac{v}{4\varepsilon})$. The initial function $v_0(x) = x - x_0$ comes from the inverse of the transformation (5.2).

We uniformly partition the spatial interval into n points: $0 = x_1 < \cdots < x_n = L$ with a mesh size $h := \frac{L}{n-1}$. Let $u_i(t)$ denote $u(x_i,t)$ and $v_i(t)$ denote $v(x_i,t)$ for $i = 1, \cdots, n$. We apply the finite difference scheme to discretize both the Burgers equation and the transformed equation, and discretize the term $-uu_x$ in (5.1) using a WCNS scheme [27] and the viscosity term εu_{xx} using a fourth-order central difference

$$(u_i)_{xx} = \frac{-u_{i-2} + 16u_{i-1} - 30u_i + 16u_{i+1} - u_{i+2}}{12h^2}.$$

Let $\mathbf{u}(t) := [u_1(t), \cdots, u_n(t)]^T$. The full order model corresponding to (5.1) is

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{A}_u\mathbf{u} + F_u(\mathbf{u}), \tag{5.4}$$

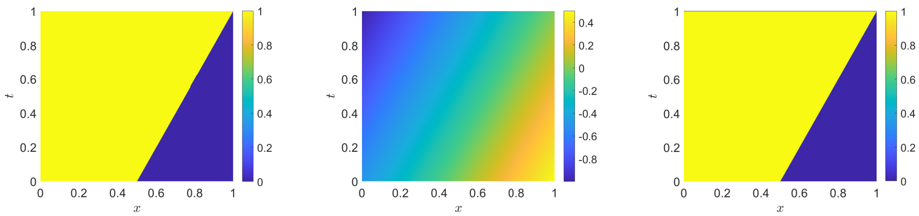


Figure 11: Numerical results of Burgers equation. (a) The reference state solution of Burgers equation (5.1) obtained by (5.4). (b) The state solution of V-FOM(5.5). (c) The state transformation $U^v = \phi(v)$.

where $\mathbf{A}_u$ comes from εu_{xx} and $F_u(\mathbf{u})$ comes from the nonlinear term $-uu_x$. We use a second-order central difference scheme to discretize the first order derivative v_x in the transformed equation (5.3), i.e.

$$(v_i)_x = \frac{v_{i+1} - v_{i-1}}{2h}.$$

Let $\mathbf{v}(t) := [v_1(t), \cdots, v_n(t)]^T$ denote the vector of the dependent variable. The full order model corresponding to (5.3) can be written as

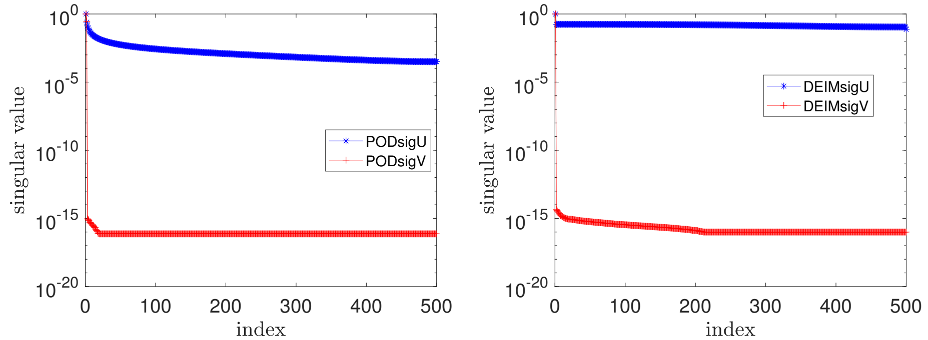
$$\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = \mathbf{A}_v\mathbf{v} + F_v(\mathbf{v}), \tag{5.5}$$

where $\mathbf{A}_v$ is the matrix arising from the diffusion term εv_{xx} and $F_v(\mathbf{v})$ is the vector function representing the other terms. Similar to the Allen-Cahn equation in the previous section, we can apply a POD-qDEIM method to derive reduced order models for the full order model.

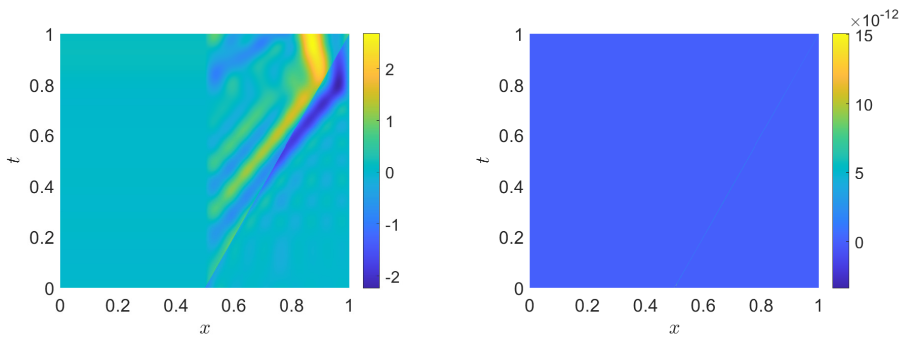
In our numerical experiments, we set $L = 1$, $T = 1$, $x_0 = 0.5$, and the number of spatial points to $n = 1000$. The viscosity parameter is $\varepsilon = 0.0001$, which is very small. We use the ode15s solver in MATLAB to solve the two full order models and their reduced models, compute 1000 timesteps, i.e. $\Delta T = 1/999$. The reference solution is obtained by solving U-FOM (5.4), as shown in Figure 11(a), which clearly displays a traveling wave with a very thin inner layer. The numerical solution for the full order model (V-FOM) (5.5) and its transformation U^v are shown in Figure 11. We observe that the solution for V-FOM is quite smooth and that the solution U^v is almost identical to the reference solution obtained by solving U-FOM. The moving inner layer structure of the reference solution makes it difficult to obtain a good reduction for U-FOM, while it is much easier to develop a reduced order model for V-FOM.

To derive reduced-order models, we take $M = 1000$ uniform samples $t_1, \cdots, t_M$ over the time interval $[0, 1]$ and use the solution snapshots at these time points to generate the POD and DEIM basis. Figure 12(a) shows the normalized singular values of the POD snapshot matrices for the solutions of U-FOM and V-FOM. Figure 12(b) shows the normalized singular values of the corresponding DEIM snapshot matrices. Clearly, the singular values for the slow variable v decay much faster than those for the original variable u.

We set the POD basis number to $r = 10$ and the DEIM basis number to $m = 20$. The error $U - U_{appr}$ is shown in Figure 12(c). In comparison, the error $U - U^v_{appr}$



(a) normalized singular values of POD matrices (b) normalized singular values of DEIM matrices



(c) error of U_{appr} in the space time domain (d) error of U^v_{appr} in the space time domain

Figure 12: Numerical results of 1D Burgers equation. (a) PODsigU: normalized singular values of POD snapshot matrix for *U-FOM* (5.4); PODsigV: normalized singular values of POD snapshot matrix for *V-FOM* (5.5). (b) DEIMsigU: normalized singular values of DEIM snapshot matrix for *U-FOM* (5.4); DEIMsigV: normalized singular values of DEIM snapshot matrix for *V-FOM* (5.5). (c) The error between the exact solution and U_{appr} with 10 POD bases and 20 DEIM bases. (d) The error between the exact solution and U^v_{appr} with 10 POD bases and 20 DEIM bases.

is shown in Figure 12(d). Clearly, we observe that V-ROM generates much smaller errors than U-ROM.

The relative error of the solution of the reduced models at each time point is shown in Figure 13. It seems that the relative error of U^v_{appr} does not increase with time and is much smaller than that of U_{appr}. The relative error is of order $O(10^{-12})$ in this case.

We can also change the initial position of the transition layer. For example, we modify the initial function by changing $x_0 = 0.5$ to $\tilde{x}_0 = 0.3$. The difference between the reference solution and the solution of V-FOM obtained above is shown in Figure 14. It seems that V-ROM can be applied directly to these new initial values. This is similar to the Allen-Cahn equation, since the solution is still in the solution manifold for v when we only change the initial position of the transition layer.

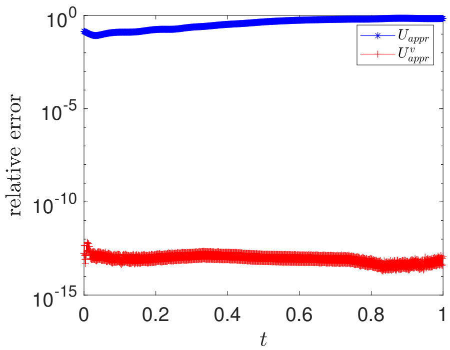


Figure 13: Numerical results of 1D Burgers equation. With 10 POD bases and 20 DEIM bases, U_{appr}: the relative errors between the exact solution and U_{appr} over $t \in [0, T]$; U_{appr}^v: the relative errors between the exact solution and U_{appr}^v over $t \in [0, T]$.

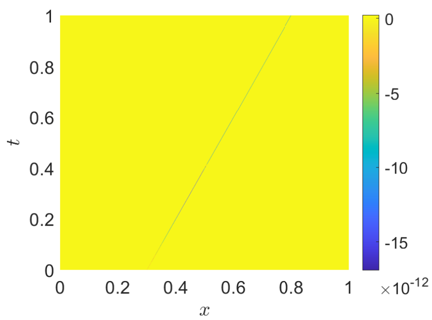


Figure 14: Numerical results of 1D Burgers equation. The error in the space-time domain between the reference solution and U_{appr}^v obtained by applying the V-ROM to a new initial function.

6. Conclusions. In this paper, we present a novel model reduction method for partial differential equations involving a small parameter. The solution manifold of these equations exhibits a slowly decaying Kolmogorov N-width, making the standard model reduction method highly inefficient. Our method revolves around learning slow variables for the dynamic problem and performing a transformation of the partial differential equation. We derive equations for these slow variables and apply well-established model reduction techniques, such as the POD-qDEIM method. Notably, the Kolmogorov N-width of the solution manifold for the slow variables decays much faster than that of the original function. Consequently, the model reduction method applied to the transformed equation demonstrates significantly improved performance compared to the original method. We validate our approach through numerous numerical experiments focusing on the Allen-Cahn equation. Furthermore, we demonstrate the applicability of our model reduction method to other equations, such as the convection equation and the Burgers equation.

There are still some aspects that require further consideration in our future work. Firstly, in this paper, we primarily select slow variables through asymptotic analysis.

However, it would be intriguing to explore the possibility of learning slow variables directly from data using machine learning algorithms. Secondly, it is crucial to conduct theoretical analysis for the model reduction method. Extending the standard analysis applied to the POD and qDEIM methods to the transformed equations should not pose significant challenges. Thirdly, the qDEIM method appears to be less effective for certain complex nonlinear problems. Therefore, improvements to the model reduction methods for nonlinear problems are still necessary. Lastly, it would be of great interest to investigate the applicability of our method to other problems, such as the Fokker-Planck equation and the Cahn-Hilliard equation.

Appendix A: Some standard model reduction algorithms. The standard POD algorithm for (1.1) is given in Algorithm A1.

Algorithm A1 A POD method

Input: $\mathbf{E}$, $\mathbf{A}$, $F(\mathbf{u})$, $\mathbf{U}$, r;

Output: $\mathbf{V}$, $\tilde{\mathbf{E}}$, $\tilde{\mathbf{A}}$, $\tilde{F}(\tilde{\mathbf{u}})$;

$[\mathbf{X}, \mathbf{\Sigma}, \mathbf{Y}] = \mathrm{svd}(\mathbf{U})$;

$\mathbf{V} = \mathbf{X}(:, 1:r)$; {Taking the first r columns of $\mathbf{X}$.}

$\tilde{\mathbf{E}} = \mathbf{V}^T\mathbf{E}\mathbf{V}$;

$\tilde{\mathbf{A}} = \mathbf{V}^T\mathbf{A}\mathbf{V}$;

$\tilde{F}(\tilde{\mathbf{u}}) = \mathbf{V}^T F(\mathbf{V}\tilde{\mathbf{u}})$.

The standard DEIM algorithm is given in Algorithm A2.

Algorithm A2 The DEIM technique

Input: $F(\mathbf{u})$, $\{\mathbf{w}_1, \cdots, \mathbf{w}_M\}$, m;

Output: $\mathbf{D}$, $\{id_1, \cdots, id_m\}$;

$\mathbf{F} = [F(\mathbf{w}_1), \cdots, F(\mathbf{w}_M)]$;

$[\mathbf{X}, \mathbf{\Sigma}, \mathbf{Y}] = \mathrm{svd}(\mathbf{F})$;

$[\mathbf{d}_1, \cdots, \mathbf{d}_m] =: \mathbf{D} = X(:, 1:m)$; {Taking the first m columns of $\mathbf{X}$.}

$[|\rho_1|, id_1] = \max_{i=1,\cdots,n}\{|\mathbf{d}_{1,i}|\}$;

$\mathbf{D}_1 := [\mathbf{d}_1], \mathbf{P}_1 := [\mathbf{e}_{id_1}]$;

for $i = 2$ to m **do**

  $\mathbf{r} = \mathbf{d}_i - \mathbf{D}_{i-1}(\mathbf{P}_{i-1}^T\mathbf{D}_{i-1})^{-1}\mathbf{P}_{i-1}^T\mathbf{d}_i$;

  $[|\rho_i|, id_i] = \max_{k=1,\cdots,n}\{|\mathbf{r}_k|\}$;

  $\mathbf{D}_i := [\mathbf{D}_{i-1}\ \mathbf{d}_i], \mathbf{P}_i := [\mathbf{P}_{i-1}\ \mathbf{e}_{id_i}]$;

end for

The qDEIM algorithm is given in Algorithm A3.

Algorithm A3 The qDEIM technique

Input: $F(\mathbf{u})$, $\{\mathbf{w}_1, \cdots, \mathbf{w}_M\}$, m;

Output: $\mathbf{D}$, $\{id_1, \cdots, id_m\}$;

$\mathbf{F} = [F(\mathbf{w}_1), \cdots, F(\mathbf{w}_M)]$;

$[\mathbf{X}, \mathbf{\Sigma}, \mathbf{Y}] = \mathrm{svd}(\mathbf{F})$;

$\mathbf{D} = \mathbf{X}(:, 1:m)$;{Taking the first m columns of $\mathbf{X}$.}

$[\mathbf{Q}, \mathbf{R}, \mathbf{Order}] = \mathrm{qr}(\mathbf{D}^T)$; {**Order** denotes the column indices after pivoting.}

$\{id_1, \cdots, id_m\} = \mathbf{Order}(1:m)$. {Taking the first m indices.}

Appendix B: Asymptotic analysis for the Burgers equation.

We do asymptotic analysis for the following Burgers equation

$$u_t = -uu_x + \varepsilon u_{xx}, \qquad x \in (-\infty, \infty), t > 0 \tag{B1}$$

with an initial value

$$u(x,0) = \begin{cases} 1 & x < x_0, \\ 0 & x > x_0, \end{cases} \tag{B2}$$

The analysis is inspired by that in [24]. Assume that there is an inner layer centered at $s(t)$. Using the matched asymptotic expansions technique [24], we analyze the outer expansion of $u(x,t)$ at x far from $s(t)$ and the inner expansion in the neighborhood of $s(t)$. We consider only the leading order term of $u(x,t)$.

Outer expansion. When ε is small, we assume the following expansion of u far from $s(t)$,

$$u(x,t) \sim u_0(x,t) + \varepsilon u_1(x,t) + \varepsilon^2 u_2(x,t) + \cdots .$$

Substituting this expansion into the equation (B1), and comparing the leading order term on both sides, we can obtain

$$(u_0)_t = -u_0(u_0)_x$$

This is the inviscid Burgers equation. Using the characteristic line method, we have $u_0(x,t) = u_0(\bar{x}, 0)$ when $x = \bar{x} + u_0(\bar{x}, 0)t$. Since the initial value $u(x,0)$ is a piecewise constant function, $u_0(x,t)$ is

$$u_0(x,t) = \begin{cases} 1 & x < s(t), \\ 0 & x > s(t). \end{cases}$$

Inner expansion. Next, we consider the situation in the neighborhood of $s(t)$. Let the *interior layer coordinate* be

$$z(x,t) := \frac{x - s(t)}{\varepsilon^\alpha} \in (-\infty, +\infty),$$

where $\alpha > 0$. The variable z stretches the neighborhood of $s(t)$. Use $U(z,t)$ to represent $u(x,t)$ in this neighborhood, then u_t, u_x, u_{xx} are represented accordingly as

$$\begin{aligned} &u_t = U_z z_s s_t + U_t = -\tfrac{1}{\varepsilon^\alpha} U_z s', \\ &u_x = U_z z_x = \tfrac{1}{\varepsilon^\alpha} U_z, \\ &u_{xx} = \tfrac{1}{\varepsilon^{2\alpha}} U_{zz}. \end{aligned}$$

Substituting the relations into the equation (B1), U satisfies

$$\varepsilon^0 U_t + \varepsilon^{-\alpha}(UU_z - s'U_z) - \varepsilon^{1-2\alpha} U_{zz} = 0.$$

Balancing the orders of ε of terms in the above equation:

1. $-\alpha = 1 - 2\alpha \Rightarrow \alpha = 1$, in this case, the last two terms are balanced, and the order of the first term is zero satisfies $0 > -\alpha$, so the first term is a higher order term which is reasonable.

2. $1-2\alpha = 0 \Rightarrow \alpha = 1/2$, in this case, the first term and the third term are balanced, but the second term $-\alpha = -1/2 < 0$ is a lower order term which is not possible.

Assume the asymptotic expansion of $U(x,t)$ is

$$U(z,t) \sim U_0(z,t) + \varepsilon U_1(z,t) + \varepsilon^2 U_2(z,t) + \cdots,$$

Substituting this and $\alpha = 1$ into the equation (B1), comparing the leading order term on both sides, we can get

$$-s'(U_0)_z + U_0(U_0)_z = (U_0)_{zz}.$$

Integrating the above formula with respect to z on $\mathbb{R}$, we get

$$A(t) - s'U_0 + \frac{1}{2}(U_0)^2 = (U_0)_z, \tag{B3}$$

where $A(t)$ is a undefined function independent of z.

Matching. Matching the leading order term of outer expansion and inner expansion gives

$$\begin{aligned} \lim_{z\to-\infty} U_0 &= \lim_{x\to s(t)^-} u_0 = (u_0)_- = 1, \\ \lim_{z\to+\infty} U_0 &= \lim_{x\to s(t)^+} u_0 = (u_0)_+ = 0. \end{aligned}$$

The derivative definition leads to $\lim_{z\to\pm\infty}(U_0)_z = 0$. Applying this limitation on (B3) gives

$$\begin{cases} 0 = A(t) - s'(u_0)_- + \frac{1}{2}((u_0)_-)^2 = A(t) - s' + \frac{1}{2}, \\ 0 = A(t) - s'(u_0)_+ + \frac{1}{2}((u_0)_+)^2 = A(t). \end{cases}$$

Then $s'(t) = 1/2$. Combined with $s(0) = x_0$, $s(t)$ is solved to be

$$s(t) = \frac{1}{2}t + x_0.$$

Substituting the above results back into the inner expansion (B3), U_0 satisfies the following ordinary differential equation about z

$$\begin{cases} \Phi'(z) = \frac{1}{2}(\Phi^2 - \Phi), \\ \lim_{z\to-\infty} \Phi = 1, \ \lim_{z\to+\infty} \Phi = 0. \end{cases}$$

The solution to the above equation is $\Phi(z) = \frac{1}{2}(1 - \tanh(\frac{1}{4}z))$, Therefore

$$U_0 = \frac{1}{2}(1 - \tanh(\frac{x - (\frac{1}{2}t + x_0)}{4\varepsilon})).$$

The leading order term of u on Ω can be obtained by adding the approximations together and subtracting the common part:

$$u_0(x,t) + U_0(x,t) - \lim_{x\to s(t)} u_0(x,t) = U_0(x,t).$$

REFERENCES

[1] S. M. Allen and J. W. Cahn, A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening, Acta metallurgica, 27 (1979), pp. 1085–1095.

[2] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera, An 'empirical interpolation' method: Application to efficient reduced-basis discretization of partial differential equations, Comptes Rendus Mathematique, 339 (2004), pp. 667–672.

[3] M. Beneš, V. Chalupeckỳ, and K. Mikula, Geometrical image segmentation by the allen–cahn equation, Applied Numerical Mathematics, 51 (2004), pp. 187–205.

[4] P. Benner and T. Breiten, Two-sided projection methods for nonlinear model order reduction, SIAM Journal on Scientific Computing, 37 (2015), pp. B239–B260.

[5] P. Benner, A. Cohen, M. Ohlberger, and K. Willcox, eds., Model Reduction and Approximation: Theory and Algorithms, no. 15 in Computational Science and Engineering, SIAM, Society for Industrial and Applied Mathematics, Philadelphia, 2017.

[6] P. Benner, P. Goyal, B. Kramer, B. Peherstorfer, and K. Willcox, Operator inference for non-intrusive model reduction of systems with non-polynomial nonlinear terms, Computer Methods in Applied Mechanics and Engineering, 372 (2020), p. 113433.

[7] G. Caginalp, An analysis of a phase field model of a free boundary, Archive for rational mechanics and analysis, 92 (1986), pp. 205–245.

[8] G. Caginalp and P. C. Fife, Dynamics of layered interfaces arising from phase boundaries, SIAM Journal on Applied Mathematics, 48 (1988), pp. 506–518.

[9] S. Cao, Choose a transformer: Fourier or galerkin, Advances in neural information processing systems, 34 (2021), pp. 24924–24940.

[10] S. Chaturantabut and D. C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, SIAM Journal on Scientific Computing, 32 (2010), pp. 2737–2764.

[11] S. Chaturantabut and D. C. Sorensen, A state space error estimate for pod-deim nonlinear model reduction, SIAM Journal on Numerical Analysis, 50 (2012), pp. 46–63.

[12] L.-Q. Chen, Phase-field models for microstructure evolution, Annual review of materials research, 32 (2002), pp. 113–140.

[13] X. Chen, Spectrum for the allen-chan, chan-hillard, and phase-field equations for generic interfaces, Communications in partial differential equations, 19 (1994), pp. 1371–1395.

[14] Z. Drmač and S. Gugercin, A new selection operator for the discrete empirical interpolation method—improved a priori error bound and extensions, SIAM Journal on Scientific Computing, 38 (2016), pp. A631–A648.

[15] Q. Du and X. Feng, The phase field method for geometric moving interfaces and their numerical approximations, Handbook of Numerical Analysis, 21 (2020), pp. 425–508.

[16] L. C. Evans, H. M. Soner, and P. E. Souganidis, Phase transitions and generalized motion by mean curvature, Communications on Pure and Applied Mathematics, 45 (1992), pp. 1097–1123.

[17] X. Feng and A. Prohl, Numerical analysis of the allen-cahn equation and approximation for mean curvature flows, Numerische Mathematik, 94 (2003), pp. 33–65.

[18] X. Feng and H.-j. Wu, A posteriori error estimates and an adaptive finite element method for the allen–cahn equation and the mean curvature flow, Journal of Scientific Computing, 24 (2005), pp. 121–146.

[19] J. Fischer, T. Laux, and T. M. Simon, Convergence rates of the allen–cahn equation to mean curvature flow: A short proof based on relative entropies, SIAM Journal on Mathematical Analysis, 52 (2020), pp. 6222–6233.

[20] C. Greif and K. Urban, Decay of the kolmogorov N-width for wave problems, Applied Mathematics Letters, 96 (2019), pp. 216–222.

[21] S. Gugercin and A. C. Antoulas, A survey of model reduction by balanced truncation and some new results, International Journal of Control, 77 (2004), pp. 748–766.

[22] B. Haasdonk, Reduced basis methods for parametrized pdes–a tutorial introduction for stationary and instationary problems, Model reduction and approximation: theory and algorithms, 15 (2017), p. 65.

[23] J. S. Hesthaven, C. Pagliantini, and G. Rozza, Reduced basis methods for time-dependent problems, Acta Numerica, 31 (2022), pp. 265–345.

[24] M. H. Holmes, Introduction to Perturbation Methods, Texts in Applied Mathematics, Springer New York, New York, NY, 2013.

[25] M. H. Holmes, Introduction to the Foundations of Applied Mathematics, Texts in Applied Mathematics, Springer International Publishing, Cham, 2019.

[26] Z. Huang, G. Lin, and A. M. Ardekani, Consistent and conservative scheme for incompressible two-phase flows using the conservative allen-cahn model, Journal of Com-

putational Physics, 420 (2020), p. 109718.

[27] Y. Jiang, X. Chen, R. Fan, and X. Zhang, High order semi-implicit weighted compact nonlinear scheme for viscous Burgers' equations, Mathematics and Computers in Simulation, 190 (2021), pp. 607–621.

[28] S. Lall, J. E. Marsden, and S. Glavaški, A subspace approach to balanced truncation for model reduction of nonlinear control systems, International Journal of Robust and Nonlinear Control, 12 (2002), pp. 519–535.

[29] C. Lee, Y. Choi, and J. Kim, An explicit stable finite difference method for the allen–cahn equation, Applied Numerical Mathematics, 182 (2022), pp. 87–99.

[30] K. Lu, K. Zhang, H. Zhang, X. Gu, Y. Jin, S. Zhao, C. Fu, and Y. Yang, A review of model order reduction methods for large-scale structure systems, Shock and Vibration, 2021 (2021), pp. 1–19.

[31] S. Lu and X. Xu, An efficient diffusion generated motion method for wetting dynamics, Journal of Computational Physics, 441 (2021), p. 110476.

[32] A. V. Mamonov and M. A. Olshanskii, Interpolatory tensorial reduced order models for parametric dynamical systems, 2022.

[33] A. Mayo and A. Antoulas, A framework for the solution of the generalized realization problem, Linear Algebra and its Applications, 425 (2007), pp. 634–662.

[34] D. Mumford, J. Fogarty, and F. Kirwan, Ergebnisse der mathematik und ihrer grenzgebiete (2)[results in mathematics and related areas (2)], 1994.

[35] M. Nonino, F. Ballarin, G. Rozza, and Y. Maday, A reduced basis method by means of transport maps for a fluid–structure interaction problem with slowly decaying kolmogorov n-width, Advances in Computational Science and Engineering, 1 (2023), pp. 36–58.

[36] B. Peherstorfer, Model reduction for transport-dominated problems via online adaptive bases and adaptive sampling, SIAM Journal on Scientific Computing, 42 (2020), pp. A2803–A2836.

[37] J. Reiss, P. Schulze, J. Sesterhenn, and V. Mehrmann, The shifted proper orthogonal decomposition: A mode decomposition for multiple transport phenomena, SIAM Journal on Scientific Computing, 40 (2018), pp. A1322–A1344.

[38] C. W. Rowley, Model reduction for fluids, using balanced proper orthogonal decomposition, International Journal of Bifurcation and Chaos, 15 (2005), pp. 997–1013.

[39] G. Rozza, D. B. P. Huynh, and A. T. Patera, Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: Application to transport and continuum mechanics, Archives of Computational Methods in Engineering, 15 (2008), pp. 229–275.

[40] P. J. Schmid, Dynamic mode decomposition of numerical and experimental data, Journal of Fluid Mechanics, 656 (2010), pp. 5–28.

[41] H. L. Shang, A survey of functional principal component analysis, AStA Advances in Statistical Analysis, 98 (2014), pp. 121–142.

[42] J. Shen and X. Yang, An efficient moving mesh spectral method for the phase-field model of two-phase flows, Journal of computational physics, 228 (2009), pp. 2978–2992.

[43] J. Shen and X. Yang, Numerical approximations of allen-cahn and cahn-hilliard equations, Discrete Contin. Dyn. Syst, 28 (2010), pp. 1669–1691.

[44] X. Shi, H. Yan, Q. Huang, J. Zhang, L. Shi, and L. He, Meta-model based high-dimensional yield analysis using low-rank tensor approximation, in Proceedings of the 56th Annual Design Automation Conference 2019, Las Vegas NV USA, June 2019, ACM, pp. 1–6.

[45] T. Tang and J. Yang, Implicit-explicit scheme for the allen-cahn equation preserves the maximum principle, Journal of Computational Mathematics, (2016), pp. 451–461.

[46] B. Unger and S. Gugercin, Kolmogorov n-widths for linear dynamical systems, Advances in Computational Mathematics, 45 (2019), pp. 2273–2286.

[47] S. Volkwein, Proper orthogonal decomposition: Theory and reduced-order modelling, Lecture Notes, University of Konstanz, 4 (2013), pp. 1–29.