Non-intrusive Data-driven Model Reduction for Differential Algebraic Equations Derived from Lifting Transformations

Parisa Khodabakhshi[a,*], Karen E. Willcox[a]

[a] *Oden Institute for Computational Engineering and Sciences, University of Texas at Austin, Texas 78712.*

Abstract

This paper presents a non-intrusive data-driven approach for model reduction of nonlinear systems. The approach considers the particular case of nonlinear partial differential equations (PDEs) that form systems of partial differential-algebraic equations (PDAEs) when lifted to polynomial form. Such systems arise, for example, when the governing equations include Arrhenius reaction terms (e.g., in reacting flow models) and thermodynamic terms (e.g., the Helmholtz free energy terms in a phase-field solidification model). Using the known structured form of the lifted algebraic equations, the approach computes the reduced operators for the algebraic equations explicitly, using straightforward linear algebra operations on the basis matrices. The reduced operators for the differential equations are inferred from lifted snapshot data using operator inference, which solves a linear least squares regression problem. The approach is illustrated for the nonlinear model of solidification of a pure material. The lifting transformations reformulate the solidification PDEs as a system of PDAEs that have cubic structure. The operators of the lifted system for this solidification example have affine dependence on key process parameters, permitting us to learn a parametric reduced model with operator inference. Numerical experiments show the effectiveness of the resulting reduced models in capturing key aspects of the solidification dynamics.

Keywords: Reduced Order Model, Nonlinear Model Reduction, Lifting Transformations, Differential Algebraic Equation, Proper Orthogonal Decomposition, Operator Inference, Additive Manufacturing, Solidification.

1. Introduction

Model reduction is effective in reducing the computational cost of simulating complex systems but remains a challenging task when the governing physics

[*] Please address correspondence to `parisa@austin.utexas.edu`

exhibit nonlinear dynamics that are not readily amenable to low-dimensional approximations. Variable transformations combined with data-driven learning of the reduced-order model (ROM) operators have emerged as one strategy to address the challenges of nonlinear model reduction [1, 2]; however, for a large class of systems, including those that arise in reacting flow and phase-field models, the desired variable transformations lead to systems of partial differential-algebraic equations (PDAEs). This paper considers the form of the PDAEs that arise in lifting a nonlinear system to polynomial form and exploits that structure to extend the Operator Inference (OpInf) approach of [3] to these lifted PDAE systems.

As a driving application, we consider a solidification process in metal additive manufacturing. Additive manufacturing is a process during which a three-dimensional part is built via the layer-by-layer deposition of material according to its digital model. Additive manufacturing's layer-wise process adds value by allowing for the manufacturing of components with complex geometries that are either infeasible or difficult to build by conventional manufacturing processes. However, the additive manufacturing process takes place over a wide range of length scales and time scales, making numerical simulations computationally expensive. Further, uncertainty quantification is essential since the structure and properties of the resulting components are sensitive to process parameter variations [4]. Thus, ROMs are key enablers to making control, optimization, and uncertainty quantification computationally feasible for additive manufacturing. ROMs can be used in multiscale modeling of additively manufactured parts to reduce the computational costs of part-scale simulations while maintaining the desired properties at microscale [5].

Our target problem poses several challenges for existing model reduction methods. First, the transport-dominated physics of the solidification interface result in highly localized changes in the state solution with time. Classical projection-based model reduction methods that seek approximations of the state in a linear subspace (see e.g., [6–9]) require many modes to achieve accuracy in a problem such as this one, rendering the resulting ROMs inefficient. Second, the forward solidification model, a coupled system of nonlinear partial differential equations (PDEs) comprising a phase-field equation and a heat equation, has a strong nonlinear dependence on the process parameters. Classical projection-based model reduction methods that use hyper-reduction methods (such as the Empirical Interpolation Method [10] and the Discrete Empirical Interpolation Method [11]) will require many interpolation points to approximate the nonlinear terms, again rendering the resulting ROMs inefficient.

Methods based on variable transformations are becoming an effective alternative for model reduction of nonlinear systems of PDEs. These approaches draw upon the notion that the introduction of auxiliary variables (often referred to as "lifting") can lead to a reformulation of the governing equations in a structured form. For example, [12] shows how general nonlinear ordinary differential equations (ODEs) can be written as so-called "polynomial ordinary differential systems" through the introduction of additional variables. In biology, variable transformations called "recasting" are used to transform nonlinear

ODEs to the so-called S-system form, a polynomial form that is faster to solve numerically [13]. Approaches based on the Koopman operator lift a nonlinear dynamical system to an infinite-dimensional space in which the dynamics are linear [14, 15]. Ref. [16] introduced the idea of reformulating nonlinear dynamical systems in quadratic form for model reduction and showed that the number of auxiliary variables needed to lift a system to quadratic-bilinear form is linear in the number of elementary nonlinear functions in the original state equations. The work in [16] shows that a large class of nonlinear terms that appear in engineering systems (including monomial, sinusoidal, and exponential terms) may be lifted to quadratic form. Lifting has been extended to model reduction of problems governed by PDEs and shown to be a competitive alternative to hyper-reduction methods [1, 17]. Yet, for many practical applications it is neither feasible nor desirable to explicitly transform the high-fidelity PDE solver, which motivates the use of non-intrusive data-driven model reduction.

Following the definitions in [18], a non-intrusive model reduction method computes the ROM using outputs of the high-fidelity simulation but without access to the full-order operators (or to their action on a vector). This is in contrast to a black-box method, which computes the ROM without using *a priori* or explicit knowledge of the form of the high-fidelity problem definition or its numerical implementation. Ref. [18] notes that black-box methods are non-intrusive, but not all non-intrusive methods are necessarily black-box: a non-intrusive method can exploit knowledge of the high-fidelity problem definition and the corresponding problem structure, even though it does not access the full-order operators themselves. Per this definition, non-intrusive approaches are also different from gray box modeling paradigm which combines aspects of black-box modeling and intrusive approaches. The advantage of non-intrusive approaches is that they compute the ROM directly from simulation data, without needing access to the high-fidelity operators. Methods such as dynamic mode decomposition (DMD) [19–23] and OpInf [3, 18, 24] operate on snapshot data (i.e., simulated state solutions), as do approaches that directly build surrogate models of the proper orthogonal decomposition (POD) modal coefficients [25–29]. Other non-intrusive methods, such as those based on the Loewner framework [30–34], require only compressed data (e.g., input-output measurements or transfer function measurements), and are input-invariant in contrast to OpInf where the reduced model depends on the chosen snapshots. Recent work has recognized the advantages of non-intrusive reduction methods when it comes to exploiting the power of variable transformations as discussed above. In particular, the Lift & Learn method of [2] combines lifting of a nonlinear PDE with data-driven learning of the ROM via the OpInf method of [3], so that variable transformations are applied only to snapshot data and not to the high-fidelity PDE solver itself.

For several classes of nonlinear PDEs, the particular form of the nonlinear terms means that lifting will lead to a system of PDAEs. For example, this is the case for the Arrhenius reaction terms in the tubular reactor example of [1]. It is also the case for the nonlinear thermodynamic dependencies in the solidification model considered in this paper. It is well known that reduction of PDAEs, and, in the semi-discrete case, DAEs, is challenging and that the algebraic equations require special treatment [35, 36]. In some applications the DAEs can be reformulated as a system of differential equations and model reduction techniques are applied to the index-reduced ODEs [37], but these approaches often lead to stiff systems [38]. Another approach taken in the literature is index-aware model reduction in which the nonlinear DAE is linearized about a stationary solution, the linearized DAE is decoupled into the differential and algebraic parts, and model reduction is applied to each part individually [38, 39]. These existing DAE model reduction approaches are intrusive; here we formulate a non-intrusive data-driven approach. When the algebraic equations arise through the lifting process, they take on a particular structured form. In this paper we elicit that structured form and we exploit it to learn the resulting ROM via non-intrusive operator inference. In particular, we show that the reduction of the lifted algebraic equations can be computed explicitly using only manipulations of the low-dimensional basis vectors, while the reduction of the differential equations follows the OpInf approach of [3].

Section 2 of this paper presents the lifting of a nonlinear system of PDEs to polynomial form and discusses the form of the algebraic equations that arise. Section 3 develops the proposed non-intrusive operator inference approach for the lifted system of DAEs. Section 4 presents application of the approach to solidification of a pure metal. Finally, concluding remarks are presented in Section 5.

2. Nonlinear Model Reduction via Lifting to Polynomial Form

This section first presents the general projection-based reduction of a nonlinear system and discusses its computational challenges. We then discuss lifting of nonlinear systems to systems with polynomial terms with particular attention to the differential-algebraic structure that arises for several classes of nonlinear equations. We derive the form of the ROM of the lifted DAE system.

2.1. Projection-based nonlinear model reduction

Our goal is to derive ROMs of systems of nonlinear PDEs. We consider the case where we have a system of d_q PDEs. To keep the presentation of our model reduction approach general, we consider the semi-discrete system of nonlinear ODEs that arises from spatial discretization of the PDEs of interest:

$$\dot{\mathbf{q}} = \mathbf{f}\left(\mathbf{q}\right), \qquad \mathbf{q}\left(0\right) = \mathbf{q}_0, \tag{1}$$

where $\mathbf{q}(t) \in \mathbb{R}^{nd_q}$ is the nd_q-dimensional semi-discrete state vector, with n the number of degrees of freedom in the spatial discretization, and $\mathbf{f} : \mathbb{R}^{nd_q} \rightarrow \mathbb{R}^{nd_q}$ is the discretized nonlinear function. The time interval of interest is $t \in [0, t_f]$ and $\mathbf{q}_0$ is the specified state initial condition. We refer to (1) as the full-order model (FOM).

To construct a projection-based ROM, we define a basis matrix $\mathbf{U} \in \mathbb{R}^{nd_q \times r}$, where $r \ll n$ is the ROM dimension. Using the POD method of snapshots

[40], this is done by constructing a set of K solution snapshots of (1), $\mathbf{Q} = \left[\begin{array}{cccc} \mathbf{q}(t_0) & \mathbf{q}(t_1) & \cdots & \mathbf{q}(t_{K-1}) \end{array} \right]$ where $\mathbf{q}(t_i)$ is the solution of the FOM (1) at $t = t_i$ obtained from a time-stepping scheme. The POD basis is comprised of the r left singular vectors of $\mathbf{Q}$ corresponding to the r largest singular values. That is, given the thin singular value decomposition $\mathbf{Q} = \mathbf{\Theta\Sigma\Psi}^\top$ in which the diagonal matrix $\mathbf{\Sigma}$ contains the singular values of $\mathbf{Q}$ in non-increasing order, then $\mathbf{\Theta}$ contains as its columns the left singular vectors of $\mathbf{Q}$ and the POD basis is given by the first r columns of $\mathbf{\Theta}$, i.e., $\mathbf{U} = \mathbf{\Theta}_{1:r}$. The POD basis is orthonormal, i.e., $\mathbf{U}^\top\mathbf{U} = \mathbf{I}_{r\times r}$, where $\mathbf{I}_{r\times r}$ denotes the identity matrix of dimension $r \times r$.

The POD ROM is derived by forming the POD approximation of the state, $\mathbf{q} \approx \mathbf{U}\hat{\mathbf{q}}$, and then performing a Galerkin projection to yield

$$\dot{\hat{\mathbf{q}}} = \hat{\mathbf{f}}\left(\hat{\mathbf{q}}\right), \qquad \hat{\mathbf{q}}\left(0\right) = \mathbf{U}^\top\mathbf{q}_0, \tag{2}$$

where $\hat{\mathbf{q}} \in \mathbb{R}^r$ is the reduced-order state, and $\hat{\mathbf{f}}\left(\hat{\mathbf{q}}\right) = \mathbf{U}^\top\mathbf{f}\left(\mathbf{U}\hat{\mathbf{q}}\right) \in \mathbb{R}^r$. Although (2) is a low-dimensional system of order $r \ll n$, it is not computationally efficient. The issue lies in the evaluation of the reduced nonlinear function $\hat{\mathbf{f}}$ which still scales with the dimension of the FOM, because to evaluate it we need to transform the reduced state $\hat{\mathbf{q}}$ back to the full-order state space, evaluate the nonlinear function, and then project the full-order nonlinear function $\mathbf{f}$ back to the reduced space.

To resolve this computational complexity issue arising with nonlinear model reduction, one common approach in the literature is the introduction of another layer of approximation (commonly referred to as hyper-reduction), which limits the evaluation of the nonlinear function to a subselection of sampling points [10, 11, 41–44]. Among existing hyper-reduction methods, the discrete empirical interpolation method [11] (DEIM) has been used broadly in the literature for a wide range of nonlinear model reduction applications [45–49]. It has been shown in [50, 51] that for highly nonlinear functions, the number of sampling points required for hyper-reduction is relatively high compared to the dimension of the FOM, undermining the efficacy of the resulting ROM.

2.2. Lifting transformations

An alternative nonlinear model reduction approach is to employ variable transformations to expose system structure, so that hyper-reduction is not needed [1, 2, 16]. For nonlinear models of polynomial form, Operator Inference (OpInf) [3] is a non-intrusive approach to model reduction where the reduced operators are learned from data through least squares minimization. For nonlinear PDEs with general nonlinearity, one can lift the governing nonlinear PDEs—that is, introduce auxiliary variables—to an equivalent polynomial structure, thus making the lifted system well-suited for OpInf, and then construct the ROM for the lifted form via the OpInf learning scheme [2].

The lifting happens at the PDE level. For some specialized cases, transformations can be found that preserve the number of PDE unknowns (d_q) in the transformed equations, but in general the lifting transformation increases the number of PDE unknowns (and correspondingly the number of equations) in the lifted form, to $d_l > d_q$. This is because auxiliary variables are introduced to recast the nonlinear terms. In what follows, we consider the case where the lifting map introduces additional auxiliary variables (i.e., $d_l > d_q$).

According to [16] many nonlinear systems can be lifted to an exact equivalent polynomial representation; however, for many of the nonlinear terms that appear in scientific and engineering applications, lifting leads to algebraic equations. This means that the resulting lifted system has a PDAE form. Algebraic equations can also arise due to an imposed constraint on either the polynomial degree of the lifted formulation or the number of introduced auxiliary variables. For example, the tubular reactor example of [1] has Arrhenius reaction terms that can be lifted to quartic form as a set of PDEs or to quadratic form as a set of PDAEs. Since the fourth-order operator in the quartic ROM leads to an operator inference problem with $O(r^4)$ degrees of freedom, it is typically desirable to bring the system to a quadratic or cubic form. In this paper, we explicitly consider the form of the PDAE system and its algebraic constraints that arise in such cases, and we formulate an OpInf approach that learns ROMs in this setting.

Following [2], we define a lifting map $\mathbf{T} : \mathbb{R}^{nd_q} \to \mathbb{R}^{nd_l}$ that transforms the native discretized PDE state $\mathbf{q} \in \mathbb{R}^{nd_q}$ into a lifted semi-discrete state of dimension nd_l. We partition the lifted semi-discrete state into its components corresponding to the differential equations, denoted $\mathbf{y}$, and its components corresponding to the algebraic equations, denoted $\mathbf{z}$. That is, the lifted state is

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix} \in \mathbb{R}^{nd_l}.$$

To ease notation, we will present the case where the lifting of the PDEs leads to one algebraic equation (i.e., $\mathbf{z} \in \mathbb{R}^n$) and $d_l - 1$ differential equations (i.e., $\mathbf{y} \in \mathbb{R}^{n(d_l-1)}$); however, it is straightforward to see how the method applies to cases with multiple algebraic equations. We write the state $\mathbf{y}$ as

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}^{(1)} \\ \mathbf{y}^{(2)} \\ \vdots \\ \mathbf{y}^{(m)} \\ \vdots \\ \mathbf{y}^{(d_l-1)} \end{bmatrix} \in \mathbb{R}^{n(d_l-1)}, \quad \mathbf{y}^{(m)} = \begin{bmatrix} y_1^{(m)} \\ \vdots \\ y_l^{(m)} \\ \vdots \\ y_n^{(m)} \end{bmatrix} \in \mathbb{R}^n, \tag{3}$$

where the notation $\mathbf{y}^{(m)}$ denotes the semi-discretization of the mth lifted state component, and thus forms an n-dimensional block component of the vector $\mathbf{y}$. The notation $y_l^{(m)}$ denotes the value of this mth state component at the lth discretization point (e.g., corresponding to the lth spatial point for a finite difference discretization or the lth finite element basis function, etc.).

2.3. Lifting to a system of DAEs with polynomial terms

Consider the case where the lifting map $\mathbf{T}$ leads to cubic form in the lifted equations.[1] The lifted equations can therefore be notionally written as

$$\dot{\mathbf{y}} = \mathbf{C} + \mathbf{A}\begin{bmatrix}\mathbf{y}\\ \mathbf{z}\end{bmatrix} + \mathbf{H}\left(\begin{bmatrix}\mathbf{y}\\ \mathbf{z}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{y}\\ \mathbf{z}\end{bmatrix}\right) + \mathbf{G}\left(\begin{bmatrix}\mathbf{y}\\ \mathbf{z}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{y}\\ \mathbf{z}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{y}\\ \mathbf{z}\end{bmatrix}\right), \tag{4}$$

$$\mathbf{z} = \boldsymbol{\mathcal{C}} + \boldsymbol{\mathcal{A}}\mathbf{y} + \boldsymbol{\mathcal{H}}\left(\mathbf{y}\otimes\mathbf{y}\right) + \boldsymbol{\mathcal{G}}\left(\mathbf{y}\otimes\mathbf{y}\otimes\mathbf{y}\right), \tag{5}$$

although it is important to note that we do *not* arrive at (4) and (5) by discretizing the lifted PDAEs; we present these equations only to motivate the form of the ROM. In (4) and (5), the symbol $\otimes$ denotes the Kronecker product (following the notation from [52]). The operators $\mathbf{C} \in \mathbb{R}^{n(d_l-1)}$, $\mathbf{A} \in \mathbb{R}^{n(d_l-1)\times(nd_l)}$, $\mathbf{H} \in \mathbb{R}^{n(d_l-1)\times(nd_l)^2}$, and $\mathbf{G} \in \mathbb{R}^{n(d_l-1)\times(nd_l)^3}$ are respectively the constant, linear, quadratic, and cubic operators of the differential equations in the lifted system. Similarly, $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{n}$, $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n\times[(d_l-1)n]}$, $\boldsymbol{\mathcal{H}} \in \mathbb{R}^{n\times[n(d_l-1)]^2}$, and $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{n\times[n(d_l-1)]^3}$ are respectively the constant, linear, quadratic, and cubic operators corresponding to the algebraic equation in the lifted system corresponding to the constants of the polynomial algebraic equation. Note that the governing equations are typically sparse with respect to the underlying variables (i.e., the number of one-way, two-way and three-way interactions among the d_l variables is typically quite small, so that $\mathbf{A}$, $\mathbf{H}$, $\mathbf{G}$, $\boldsymbol{\mathcal{A}}$, $\boldsymbol{\mathcal{H}}$, and $\boldsymbol{\mathcal{G}}$ will contain a large number of zero blocks).

The projection-based ROM of the DAE system (4) and (5) preserves the cubic structure, giving

$$\dot{\hat{\mathbf{y}}} = \hat{\mathbf{C}} + \hat{\mathbf{A}}\begin{bmatrix}\hat{\mathbf{y}}\\ \hat{\mathbf{z}}\end{bmatrix} + \hat{\mathbf{H}}\left(\begin{bmatrix}\hat{\mathbf{y}}\\ \hat{\mathbf{z}}\end{bmatrix}\otimes\begin{bmatrix}\hat{\mathbf{y}}\\ \hat{\mathbf{z}}\end{bmatrix}\right) + \hat{\mathbf{G}}\left(\begin{bmatrix}\hat{\mathbf{y}}\\ \hat{\mathbf{z}}\end{bmatrix}\otimes\begin{bmatrix}\hat{\mathbf{y}}\\ \hat{\mathbf{z}}\end{bmatrix}\otimes\begin{bmatrix}\hat{\mathbf{y}}\\ \hat{\mathbf{z}}\end{bmatrix}\right), \tag{6}$$

$$\hat{\mathbf{z}} = \hat{\boldsymbol{\mathcal{C}}} + \hat{\boldsymbol{\mathcal{A}}}\hat{\mathbf{y}} + \hat{\boldsymbol{\mathcal{H}}}\left(\hat{\mathbf{y}}\otimes\hat{\mathbf{y}}\right) + \hat{\boldsymbol{\mathcal{G}}}\left(\hat{\mathbf{y}}\otimes\hat{\mathbf{y}}\otimes\hat{\mathbf{y}}\right), \tag{7}$$

where $\hat{\mathbf{y}} \in \mathbb{R}^{r_1}$, $\hat{\mathbf{z}} \in \mathbb{R}^{r_2}$ are the reduced state vectors for the differential and algebraic equations, respectively. That is, given an r_1-dimensional POD basis $\mathbf{V} \in \mathbb{R}^{n(d_l-1)\times r_1}$ for the differential states $\mathbf{y}$ and another r_2-dimensional POD basis $\mathbf{W} \in \mathbb{R}^{n\times r_2}$ for the algebraic states $\mathbf{z}$, the approximation of the full-order states in the POD subspace is:

$$\begin{bmatrix}\mathbf{y}\\ \mathbf{z}\end{bmatrix} \approx \begin{bmatrix}\mathbf{V} & \mathbf{0}\\ \mathbf{0} & \mathbf{W}\end{bmatrix}\begin{bmatrix}\hat{\mathbf{y}}\\ \hat{\mathbf{z}}\end{bmatrix}. \tag{8}$$

Our task now is to determine the ROM defined by (6) and (7) by inferring the reduced operators $\hat{\mathbf{C}} \in \mathbb{R}^{r_1}$, $\hat{\mathbf{A}} \in \mathbb{R}^{r_1\times(r_1+r_2)}$, $\hat{\mathbf{H}} \in \mathbb{R}^{r_1\times(r_1+r_2)^2}$, $\hat{\mathbf{G}} \in$

[1] Note that we consider the cubic case because it arises in our additive manufacturing example problem in Section 4, but it is straightforward to see how our approach applies to systems with at most quadratic terms, as well as systems with higher-order polynomial terms.

$\mathbb{R}^{r_1\times(r_1+r_2)^3}$, $\hat{\boldsymbol{\mathcal{C}}} \in \mathbb{R}^{r_2}$, $\hat{\boldsymbol{\mathcal{A}}} \in \mathbb{R}^{r_2\times r_1}$, $\hat{\boldsymbol{\mathcal{H}}} \in \mathbb{R}^{r_2\times r_1^2}$, and $\hat{\boldsymbol{\mathcal{G}}} \in \mathbb{R}^{r_2\times r_1^3}$. The forms of (6) and (7) reveal how lifting to polynomial form circumvents the need for hyper-reduction techniques. Whereas in (2) the ROM embeds evaluations that scale with the dimension of the FOM, the polynomial form of (6) and (7) means that once the reduced operators $\hat{\mathbf{C}}$, $\hat{\mathbf{A}}$, $\hat{\mathbf{H}}$, $\hat{\mathbf{G}}$, $\hat{\boldsymbol{\mathcal{C}}}$, $\hat{\boldsymbol{\mathcal{A}}}$, $\hat{\boldsymbol{\mathcal{H}}}$, and $\hat{\boldsymbol{\mathcal{G}}}$ are derived, the ROM is completely decoupled from the FOM; that is, solving the ROM scales only with the reduced dimensions r_1 and r_2 and not with the dimension of the FOM, nor is any hyper-reduction needed.

3. Operator Inference for Lifted Differential Algebraic Equations

This section presents our approach for learning the operators of the ROM defined by (6) and (7). The differential equations (6) use the standard OpInf approach from [3], as discussed in Section 3.1. In Section 3.2, we develop a tailored approach for the algebraic equations (7), which exploits the particular structure of the lifted system.

3.1. Operator Inference for differential equations

To determine the ROM for the differential equations in (4), we use the regularized OpInf approach of [53] to infer the reduced operators $\hat{\mathbf{C}} \in \mathbb{R}^{r_1}$, $\hat{\mathbf{A}} \in \mathbb{R}^{r_1\times(r_1+r_2)}$, $\hat{\mathbf{H}} \in \mathbb{R}^{r_1\times(r_1+r_2)^2}$, and $\hat{\mathbf{G}} \in \mathbb{R}^{r_1\times(r_1+r_2)^3}$. The steps of the approach are as follows:

Step 1: Snapshots of the state vector of the original high fidelity model (1) are generated at K time steps to build the state snapshot matrix for the original variables, $\mathbf{Q} \in \mathbb{R}^{nd_q\times K}$.

Step 2: The lifting transformations defined by the lifting map $\mathbf{T}$ are applied to the snapshots. For each snapshot $\mathbf{q}$, we generate

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix} = \mathbf{T}(\mathbf{q}),$$

resulting in the lifted snapshot data for the differential and algebraic equations, contained in the lifted snapshot matrices $\mathbf{Y} \in \mathbb{R}^{n(d_l-1)\times K}$ and $\mathbf{Z} \in \mathbb{R}^{n\times K}$ respectively.

Step 3: Compute the POD basis matrices for the lifted snapshots using thin singular value decomposition:

$$\mathbf{Y} = \mathbf{\Theta}_1\mathbf{\Sigma}_1\mathbf{\Psi}_1^\top, \quad \mathbf{V} = (\mathbf{\Theta}_1)_{1:r_1} \in \mathbb{R}^{n(d_l-1)\times r_1}, \tag{9}$$

$$\mathbf{Z} = \mathbf{\Theta}_2\mathbf{\Sigma}_2\mathbf{\Psi}_2^\top, \quad \mathbf{W} = (\mathbf{\Theta}_2)_{1:r_2} \in \mathbb{R}^{n\times r_2}, \tag{10}$$

where $\mathbf{V}$ and $\mathbf{W}$ are the POD basis matrices for $\mathbf{Y}$ and $\mathbf{Z}$, respectively. The size of the bases r_1 and r_2 are chosen by assigning a threshold for the relative cumulative energy of the POD modes. That is, choose r_1 (respectively r_2) so that $\epsilon = \left(\sum_{i=1}^{r_1} \sigma_i^2\right) / \left(\sum_{i=1}^{K} \sigma_i^2\right)$ is greater than the specified tolerance, where σ_i is the ith singular value of Σ_1 (respectively Σ_2).

Step 4: Project the lifted snapshot matrices $\mathbf{Y}$ and $\mathbf{Z}$ onto their corresponding POD subspaces, to obtain the coordinates of the lifted snapshots in the POD bases. Estimate numerically the time derivative for the snapshots of the differential states:

$$\hat{\mathbf{Y}} = \mathbf{V}^\top \mathbf{Y} = \begin{bmatrix} | & | & & | \\ \hat{\mathbf{y}}\left(t_0\right) & \hat{\mathbf{y}}\left(t_1\right) & \cdots & \hat{\mathbf{y}}\left(t_{K-1}\right) \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{r_1 \times K}, \tag{11}$$

$$\dot{\hat{\mathbf{Y}}} = \begin{bmatrix} | & | & & | \\ \dot{\hat{\mathbf{y}}}\left(t_0\right) & \dot{\hat{\mathbf{y}}}\left(t_1\right) & \cdots & \dot{\hat{\mathbf{y}}}\left(t_{K-1}\right) \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{r_1 \times K}, \tag{12}$$

$$\hat{\mathbf{Z}} = \mathbf{W}^\top \mathbf{Z} = \begin{bmatrix} | & | & & | \\ \hat{\mathbf{z}}\left(t_0\right) & \hat{\mathbf{z}}\left(t_1\right) & \cdots & \hat{\mathbf{z}}\left(t_{K-1}\right) \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{r_2 \times K}. \tag{13}$$

Step 5: Infer the ROM operators for the differential states from snapshot data via OpInf by posing a least squares problem in a minimum residual sense:

$$\min_{\hat{\mathbf{C}}, \hat{\mathbf{A}}, \hat{\mathbf{H}}, \hat{\mathbf{G}}} \left\| \mathbf{1}_K \hat{\mathbf{C}}^\top + \hat{\mathbf{S}}^\top \hat{\mathbf{A}}^\top + \left(\hat{\mathbf{S}} \odot \hat{\mathbf{S}}\right)^\top \hat{\mathbf{H}}^\top + \left(\hat{\mathbf{S}} \odot \hat{\mathbf{S}} \odot \hat{\mathbf{S}}\right)^\top \hat{\mathbf{G}}^\top - \dot{\hat{\mathbf{Y}}}^\top \right\|_F^2, \tag{14}$$

where $\|.\|_F$ is the Frobenius norm, $\hat{\mathbf{S}}^\top = \begin{bmatrix} \hat{\mathbf{Y}}^\top & \hat{\mathbf{Z}}^\top \end{bmatrix} \in \mathbb{R}^{K \times (r_1 + r_2)}$ collects the differential and algebraic snapshot matrices in a single matrix, $\mathbf{1}_K$ is a column vector of length K of values of unity, and $\odot$ denotes the Khatri-Rao product of two matrices (also known as column-wise Kronecker product [52]). It has been shown in [3] that the reduced operators obtained from (14) converge to those obtained from intrusive projection when $dt \to 0$.

As shown in [3], (14) decomposes into r_1 independent least squares problems, one for each row of the reduced system of (6). This reduces the computational cost of the OpInf problem and also lowers the amount of training data required, since the number of coefficients to be inferred in each least squares problem scales with r^3 rather than r^4 (with $r = r_1 + r_2$). For the cubic structure in (14), the number of unknowns for each least squares problem is at most $s = 1 + r + r\left(r+1\right)/2 + r\left(r+1\right)\left(r+2\right)/6$, where we account for the elimination of the redundant terms arising from the commutativity of multiplication within $\hat{\mathbf{H}}$ and $\hat{\mathbf{G}}$. The number of unknown coefficients in the OpInf regression problem can be reduced further by constructing a separate POD basis for each physical variable, which leads to a block diagonal POD basis matrix. As noted before, the governing equations are typically sparse with respect to the underlying variables and a block diagonal POD basis preserves this sparsity. Any sparsity in the ROM can be accounted for in (14) trivially by eliminating from the least squares problem the matrix coefficients that are known to be zero, and solving (14) over the subset of coefficients that are non-zero. This is possible due to the minimum residual formulation of the OpInf least squares problem and the fact that the sparsity pattern is known a priori. Using a separate POD basis also has the advantage of being able to select a different number of modes for each variable, although its sparsity-preserving advantage can be offset by an increase in the total number of modes required to achieve a desired accuracy level. The number of snapshots K must be chosen to be at least as large as the number of unknowns s to avoid an underdetermined system. In addition, the snapshot set must be sufficiently rich to avoid ill-conditioning of (14), which suggests that we should choose $K > s$. The required number of snapshots needed for a certain accuracy is generally problem dependent, but can be checked during computation of the OpInf ROM by evaluating the condition number of the data matrix as described in [53]. The scaling of the computational cost of OpInf with respect to n, K, and r is analyzed in detail in [53].

The least-squares problem (14) is data-driven. There are multiple sources of noise in the data $\hat{\mathbf{S}}$ and $\dot{\hat{\mathbf{Y}}}$, which affect the solution of (14). Sources of noise include the errors introduced due to numerical approximation of the time derivatives $\dot{\hat{\mathbf{Y}}}$, closure error due to the disregarded POD modes, and any potential model mis-specification error. Thus, to avoid overfitting the operators to the data, regularization is needed as discussed in [53].

3.2. Reduced representation of the algebraic equations

We now present the approach for deriving the reduced operators $\hat{\boldsymbol{\mathcal{C}}}$, $\hat{\boldsymbol{\mathcal{A}}}$, $\hat{\boldsymbol{\mathcal{H}}}$, and $\hat{\boldsymbol{\mathcal{G}}}$ of the algebraic equation (7). Our approach leverages the known structure of the algebraic equations that arise during the lifting of the original nonlinear system to polynomial form. In particular, we observe that the algebraic equations have a *pointwise structure* because they arise from lifted equations that specify definitional relationships between lifted continuous variables (this will be illustrated in our example in Section 4). We can expand the terms in the algebraic equations (5) in a component-wise summation format as

$$\begin{aligned}\mathbf{z} = \boldsymbol{\mathcal{C}} &+ \sum_{i=1}^{d_l-1} \boldsymbol{\mathcal{A}}^{(i)}\mathbf{y}^{(i)} + \sum_{i=1}^{d_l-1}\sum_{j=i}^{d_l-1} \boldsymbol{\mathcal{H}}^{(i,j)}\mathbf{y}^{(i)} \otimes \mathbf{y}^{(j)} \\ &+ \sum_{i=1}^{d_l-1}\sum_{j=i}^{d_l-1}\sum_{k=j}^{d_l-1} \boldsymbol{\mathcal{G}}^{(i,j,k)}\mathbf{y}^{(i)} \otimes \mathbf{y}^{(j)} \otimes \mathbf{y}^{(k)}, \end{aligned} \tag{15}$$

where $\mathbf{y}^{(i)}$ denotes the semi-discretization of the ith lifted state component as defined in (3), and $\boldsymbol{\mathcal{A}}^{(i)} \in \mathbb{R}^{n\times n}$, $\boldsymbol{\mathcal{H}}^{(i,j)} \in \mathbb{R}^{n\times n^2}$, and $\boldsymbol{\mathcal{G}}^{(i,j,k)} \in \mathbb{R}^{n\times n^3}$ denote the corresponding blocks within the operators of (5). Since the projection-based ROM inherits the structure of the FOM, the algebraic equations of the ROM (7) would similarly inherit the summation structure of (15). The summation format of (15) accounts for all possible linear, quadratic and cubic terms in the algebraic equation, but in practice, the algebraic equations are typically sparse and only include a few terms (further discussed in Section 4).

In principle, one could learn the reduced operators of the ROM corresponding to (15), $\hat{\boldsymbol{\mathcal{A}}}^{(i)} \in \mathbb{R}^{r\times r}$, $\hat{\boldsymbol{\mathcal{H}}}^{(i,j)} \in \mathbb{R}^{r\times r^2}$, and $\hat{\boldsymbol{\mathcal{G}}}^{(i,j,k)} \in \mathbb{R}^{r\times r^3}$, via least squares with a reformulation of (14) to reflect the residual of the algebraic equation (which does not have a time derivative term). However, this indirect learning from data is unnecessary: due to the structure of the algebraic equation, its reduced operators can be determined explicitly in a straightforward manner. In particular, due to the pointwise structure of the algebraic equation (which as noted above arises because, at the continuous level, the algebraic equations specify definitional relationships between lifted continuous variables), the operators $\boldsymbol{\mathcal{A}}^{(i)}$, $\boldsymbol{\mathcal{H}}^{(i,j)}$, and $\boldsymbol{\mathcal{G}}^{(i,j,k)}$ each will have only one non-zero term at each row corresponding respectively to the one-way, two-way, and three-way definitional relationship encoded in the algebraic equation. Therefore, we have

$$\begin{aligned}
\boldsymbol{\mathcal{C}} &= \mathscr{c}\mathbf{1}_n, && (16)\\
\boldsymbol{\mathcal{A}}^{(i)} &= \mathscr{a}^{(i)}\mathbf{I}_{n\times n}, && (17)\\
\boldsymbol{\mathcal{H}}^{(i,j)} &= \mathscr{h}^{(i,j)}\begin{bmatrix} \mathbf{e}_1 & \mathbf{0}_{n\times n} & \mathbf{e}_2 & \mathbf{0}_{n\times n}\cdots & \mathbf{0}_{n\times n} & \mathbf{e}_n \end{bmatrix}_{n\times n^2}, && (18)\\
\boldsymbol{\mathcal{G}}^{(i,j,k)} &= \mathscr{g}^{(i,j,k)}\begin{bmatrix} \mathbf{e}_1 & \mathbf{0}_{n\times(n^2+n)} & \mathbf{e}_2 & \mathbf{0}_{n\times(n^2+n)}\cdots & \mathbf{0}_{n\times(n^2+n)} & \mathbf{e}_n \end{bmatrix}_{n\times n^3}, && (19)
\end{aligned}$$

where $\mathbf{0}_{\alpha\times\beta}$ represents a zero block with α rows and β columns, $\mathbf{e}_p$ is a column vector of length n with all zeros and a unity value at the pth row, and $\mathscr{c}$, $\mathscr{a}^{(i)}$, $\mathscr{h}^{(i,j)}$ and $\mathscr{g}^{(i,j,k)}$ are scalar constants in the algebraic equation. We can then exploit this form to determine the ROM operators

$$\begin{aligned}
\hat{\boldsymbol{\mathcal{C}}} &= \mathbf{W}^\top\boldsymbol{\mathcal{C}} = \mathscr{c}\mathbf{W}^\top\mathbf{1}_n, && (20)\\
\hat{\boldsymbol{\mathcal{A}}}^{(i)} &= \mathbf{W}^\top\boldsymbol{\mathcal{A}}^{(i)}\mathbf{V}^{(i)} = \mathscr{a}^{(i)}\mathbf{W}^\top\mathbf{V}^{(i)}. && (21)\\
\hat{\boldsymbol{\mathcal{H}}}^{(i,j)} &= \mathbf{W}^\top\boldsymbol{\mathcal{H}}^{(i,j)}\left(\mathbf{V}^{(i)}\otimes\mathbf{V}^{(j)}\right), && (22)\\
\hat{\boldsymbol{\mathcal{G}}}^{(i,j,k)} &= \mathbf{W}^\top\boldsymbol{\mathcal{G}}^{(i,j,k)}\left(\mathbf{V}^{(i)}\otimes\mathbf{V}^{(j)}\otimes\mathbf{V}^{(k)}\right), && (23)
\end{aligned}$$

where $\mathbf{V}^{(i)}$ denotes the partitioning of the POD basis $\mathbf{V}$ (whether it is block diagonal or not) in the same format as (3):

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}^{(1)} \\ \mathbf{V}^{(2)} \\ \vdots \\ \mathbf{V}^{(m)} \\ \vdots \\ \mathbf{V}^{(d_l-1)} \end{bmatrix}, \tag{24}$$

where each block $\mathbf{V}^{(i)}$ has dimension $n\times r_1$. Observe that the definition of $\hat{\boldsymbol{\mathcal{C}}}$ in (20) and $\hat{\boldsymbol{\mathcal{A}}}$ in (21) use products of the POD basis matrices $\mathbf{V}$ and $\mathbf{W}$ that are straightforward to compute and then combine with the known constants $\mathscr{c}$ and

$\mathscr{a}^{(i)}$. Similarly, since each row of the operators $\boldsymbol{\mathcal{H}}^{(i,j)}$, and $\boldsymbol{\mathcal{G}}^{(i,j,k)}$ contains only one non-zero element as shown in (18) and (19), computation of the reduced operators $\hat{\boldsymbol{\mathcal{H}}}^{(i,j)}$ and $\hat{\boldsymbol{\mathcal{G}}}^{(i,j,k)}$ can be achieved efficiently and without having to first construct the corresponding full-order operators. Given (18) and (22) for the quadratic operator, and (19) and (23) for the cubic operator, $\hat{\boldsymbol{\mathcal{H}}}^{(i,j)}$ and $\hat{\boldsymbol{\mathcal{G}}}^{(i,j,k)}$ can be written as

$$\hat{\boldsymbol{\mathcal{H}}}^{(i,j)} = \mathscr{h}^{(i,j)}\mathbf{W}^{\top} \begin{bmatrix} \mathbf{V}_1^{(i)} \otimes \mathbf{V}_1^{(j)} \\ \mathbf{V}_2^{(i)} \otimes \mathbf{V}_2^{(j)} \\ \vdots \\ \mathbf{V}_l^{(i)} \otimes \mathbf{V}_l^{(j)} \\ \vdots \\ \mathbf{V}_n^{(i)} \otimes \mathbf{V}_n^{(j)} \end{bmatrix}, \qquad \hat{\boldsymbol{\mathcal{G}}}^{(i,j,k)} = \mathscr{g}^{(i,j,k)}\mathbf{W}^{\top} \begin{bmatrix} \mathbf{V}_1^{(i)} \otimes \mathbf{V}_1^{(j)} \otimes \mathbf{V}_1^{(k)} \\ \mathbf{V}_2^{(i)} \otimes \mathbf{V}_2^{(j)} \otimes \mathbf{V}_2^{(k)} \\ \vdots \\ \mathbf{V}_l^{(i)} \otimes \mathbf{V}_l^{(j)} \otimes \mathbf{V}_l^{(k)} \\ \vdots \\ \mathbf{V}_n^{(i)} \otimes \mathbf{V}_n^{(j)} \otimes \mathbf{V}_n^{(k)} \end{bmatrix}, \tag{25}$$

where $\mathbf{V}_l^{(i)}$ denotes the lth row of $\mathbf{V}^{(i)}$.

4. Application: Solidification Process in Additive Manufacturing

We test the efficacy of the proposed method for a solidification process in metal additive manufacturing. The additive manufacturing process takes place over a wide range of length scales and time scales, and it is sensitive to variations in process parameters. Numerical simulations of the additive manufacturing process are computationally expensive, making it a challenge to achieve control, optimization, and uncertainty quantification. Reduced-order modeling is thus a critical enabler for achieving models that are sufficiently accurate and computationally efficient. Section 4.1 presents the target problem of phase-field simulation of the solidification process. Section 4.2 derives the lifting transformation map to expose a cubic polynomial PDAE structure in the solidification model, and discusses the different choices and possibilities that exist in the lifting process. Section 4.3 discusses the construction of the POD basis, and the determination of the ROM operators for the algebraic equation of the lifted formulation. Finally, section 4.4 presents numerical experiments that test the efficacy of the proposed model reduction method.

4.1. Phase-Field Simulation of Solidification

This study uses Kobayashi's solidification model [54] for pure materials in which the evolution of the order parameter is represented by the Allen-Cahn equation. The governing equations are

$$\begin{cases} \dot{u} = \nabla . \left(K\left(\phi\right)\nabla u\right) + L_H\dot{\phi}, & \text{on } \left(0,t_f\right] \times \left(0,\ell\right) \\ \tau\dot{\phi} = \xi^2\Delta\phi - p'\left(\phi\right) - q\left(u,\phi\right), & \text{on } \left(0,t_f\right] \times \left(0,\ell\right), \end{cases} \tag{26}$$

where the phase-field order parameter, $\phi\left(x,t\right)$, characterizes the phase at spatial location x at time t and $u\left(x,t\right)$ is the temperature. The phase-field equation accounts for the evolution of the diffuse interface in an implicit manner where $\phi = 0$ and 1 represent the liquid and solid phases, respectively, and $\phi = 0.5$ is considered to be the location of the solid-liquid interface [55–57]. L_H is the non-dimensional latent heat, K is the thermal diffusivity, τ is the relaxation parameter, ξ represents the width of the diffuse interface, ℓ is the length of the one-dimensional physical domain, and t_f is the final time. The terms $p'\left(\phi\right)$ and $q\left(u,\phi\right)$ appearing in (26) arise respectively from the differentiation of the first and second terms of the free energy density with respect to ϕ. The Helmholtz free energy density used in this study is [54]

$$f\left(u,\phi\right) = \frac{1}{4}\phi^2\left(1-\phi\right)^2 + \left(\frac{1}{3}\phi^3 - \frac{1}{2}\phi^2\right)m\left(u\right), \tag{27}$$

where

$$m\left(u\right) = \frac{\beta}{2}m_0\left(u\right), \quad \text{with} \;\; m_0\left(u\right) = \tanh\left[\gamma\left(u_M - u\right)\right], \tag{28}$$

where $\beta < 1$ is the parameter that controls the magnitude of m and γ is the parameter controlling the rate of change of m about the melting temperature u_M. The constraint $\beta < 1$ is to enforce $|m\left(u\right)| < 1/2$ which guarantees that the Helmholtz free energy f acquires local minimum at liquid ($\phi = 0$) and solid ($\phi = 1$) phases, and a local maximum at $\phi = 1/2 - m$. Given the free energy density (27), we have

$$p'\left(\phi\right) = \frac{1}{2}\phi\left(1-\phi\right)\left(1-2\phi\right), \quad q\left(u,\phi\right) = \phi\left(\phi-1\right)m\left(u\right), \tag{29}$$

where $p\left(\phi\right) = f\left(\phi,u_M\right) = \frac{1}{4}\phi^2\left(1-\phi\right)^2$, and the prime denotes differentiation with respect to ϕ. The thermodynamical driving term is controlled by m with $m > 0$ driving the system towards solidification, and $m < 0$ resulting in the liquid phase as the stable phase. Note that the definition of m in (28) is not a unique choice [54]; other monotonically decreasing continuous functions of u can be used for m as long as $|m\left(u\right)| < 1/2$.

Homogeneous Neumann boundary conditions are imposed on the temperature and order parameter. In this study the initial conditions are chosen to be:

$$\phi(x,0) = \phi_0 = \frac{1}{2}\left(1 - \tanh\left(1000(x-x_0)\right)\right), \quad u(x,0) = u_0 = u_M\phi_0 \tag{30}$$

where ϕ_0 corresponds to having a solid phase in $0 < x < x_0$ and a liquid phase in $x_0 < x < \ell$ with x_0/ℓ as the initial solid fraction, and the temperature in the solid phase is considered to be equal to the melting temperature u_M. The initial conditions are chosen such that $u \leq u_M$, hence the system will be driven towards solidification.

In this study, the thermal diffusivity is taken to depend on the phase of the material through an interpolation function h (which satisfies $h(0) = 0$ and $h(1) = 1$):

$$K(\phi) = K_0 (1 - h(\phi)) + K_1 h(\phi), \tag{31}$$

where K_0, and K_1 represent the thermal diffusivity of the liquid and solid phases, respectively. The selected interpolation function is $h(\phi) = 6\phi^5 - 15\phi^4 + 10\phi^3$ where $h'(\phi) = 120p(\phi)$ [4].

Figure 1 shows an example result of the phase-field simulation of the solidification phenomena. The figure depicts the evolution of the temperature and order parameter in time over the one-dimensional spatial domain ($\ell = 1$). Sharp transitions in the order parameter occur along the interface (the dashed line), and the interface moves in time. Because of the existence of the latent heat term in the heat equation (26), abrupt transitions along the interface are also evident in the temperature field. The moving front nature of the solidification phenomena makes it a challenging problem for model reduction.

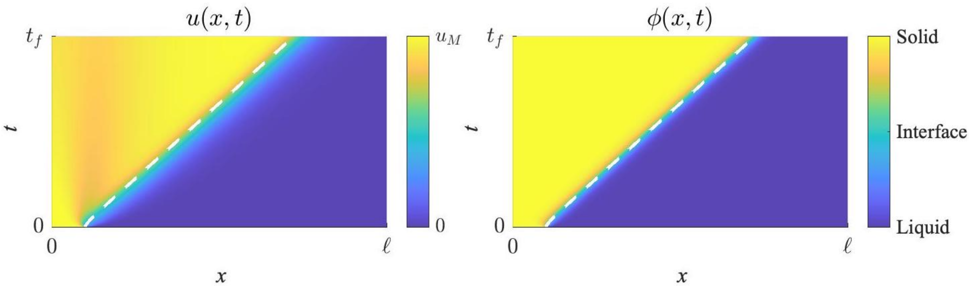

Figure 1: Phase-field simulation of the solidification phenomena of a pure material ($L_H = 1.0$, $K_0 = 1.0$, $K_1 = 0.1$, $\xi = 0.01$, $\tau = 0.0003$, $u_M = 1.0$, $\beta = 0.9$, $x_0 = 0.1$, $\ell = 1$). The dashed line represents the location of the interface. Note: the horizontal axis corresponds to the physical domain, and the vertical axis represents the time.

4.2. Lifting Transformation

We define a lifting map that lifts the nonlinear governing equations (26) to a polynomial system with cubic form. We define the auxiliary variables

$$\begin{cases} K = K_0 + (K_1 - K_0)\left(6\phi^5 - 15\phi^4 + 10\phi^3\right) \\ p = \dfrac{1}{4}\phi^2 (1-\phi)^2 \\ p' = \dfrac{1}{2}\phi(1-\phi)(1-2\phi) \\ p'' = 3\phi(\phi - 1) + \dfrac{1}{2} \\ m_0 = \tanh\left[\gamma (u_M - u)\right] \\ z = -\gamma\left(1 - m_0^2\right). \end{cases} \tag{32}$$

The lifted states are then defined with $y = \begin{bmatrix} u & \phi & K & p & p' & p'' & m_0 \end{bmatrix}^\top$, and $z = -\gamma(1 - m_0^2)$. This leads to the lifted equations

$$\left\{ \begin{aligned} &\dot{u} = \nabla . \{K \nabla u\} + \frac{L_H}{\tau}\left[\xi^2 \Delta\phi - p' - \frac{\beta}{6} m_0 \left(p'' - \frac{1}{2}\right)\right] \\ &\tau\dot{\phi} = \xi^2 \Delta\phi - p' - \frac{\beta}{6} m_0 \left(p'' - \frac{1}{2}\right) \\ &\dot{K} = \frac{120\,(K_1 - K_0)}{\tau} p \left[\xi^2 \Delta\phi - p' - \frac{\beta}{6} m_0 \left(p'' - \frac{1}{2}\right)\right] \\ &\dot{p} = \frac{1}{\tau} p' \left[\xi^2 \Delta\phi - p' - \frac{\beta}{6} m_0 \left(p'' - \frac{1}{2}\right)\right] \\ &\dot{p}' = \frac{1}{\tau} p'' \left[\xi^2 \Delta\phi - p' - \frac{\beta}{6} m_0 \left(p'' - \frac{1}{2}\right)\right] \\ &\dot{p}'' = \frac{3}{\tau}\,(2\phi - 1) \left[\xi^2 \Delta\phi - p' - \frac{\beta}{6} m_0 \left(p'' - \frac{1}{2}\right)\right] \\ &\dot{m}_0 = z\Big(\nabla . \{K \nabla u\} + \frac{L_H}{\tau}\left[\xi^2 \Delta\phi - p' - \frac{\beta}{6} m_0 \left(p'' - \frac{1}{2}\right)\right]\Big) \\ &z = -\gamma\left(1 - m_0^2\right) \end{aligned} \right. \tag{33}$$

These equations comprise a PDAE system of seven PDEs and one algebraic equation. The lifted system has cubic form; that is, all terms on the right-hand side of (33) are constant or have only linear, quadratic, or cubic dependence on the lifted states y and z. The lifted system is sparse in the sense that only a limited number of the possible interactions among auxiliary variables appear in the equations. For example, in (33), the equation for $\dot{\phi}$ includes on its right-hand side only four terms: linear in ϕ, linear in p', linear in m_0, and bilinear in m_0 and p''. The lifting introduces no approximations—it is simply a rewriting of the equation set (26), but since it employs the chain rule to derive the equations for the auxiliary variables, the lifting assumes that the necessary derivatives exist. An important point to emphasize is that the lifted equations (33) will *not* be discretized or solved, but rather provide the guiding structure for defining the variables over which the POD basis is computed and in formulating the operator inference problem.

The lifting transformation is not unique and the choice employed in (32) is just one of a number of choices available for this example. A different lifting transformation will lead to a different lifted system. For example, in the solidification problem of (26) an alternative choice for the auxiliary variables would be to replace $\begin{bmatrix} K & p & p' & p'' \end{bmatrix}^\top$ with $\begin{bmatrix} \phi^5 & \phi^4 & \phi^3 & \phi^2 \end{bmatrix}^\top$. We chose the former set of auxiliary variables, because they each represent some physical aspect of the solidification model (i.e., diffusion coefficient, double-well potential, and the first and the second derivative of the double-well potential which appear explicitly in (26)). However, in principle one could choose the latter set of auxiliary variables. This would also lead to a PDAE system with cubic structure, but the sparsity structure of the system of equations would be different, because the relationships among auxiliary variables are different.

Yet another choice for the lifting transformations is to eliminate the variable

z by replacing it with $z = -\gamma(1 - m_0^2)$ in the evolution equation for $\dot{m}_0$ in (33). This would lead to a lifted system of seven PDEs with quartic structure, with the advantage of eliminating the algebraic equation. A similar observation was made in [1] for the Arrhenius reaction term of a tubular reactor model, with different lifting transformations resulting in either a quartic system of PDEs or a quadratic PDAE system. The disadvantage of the quartic PDE lifting choice for our solidification example is that the number of reduced operator coefficients to be learned for the cubic structure of (33) scales with r^3, whereas for the quartic structure it would scale with r^4. Consequently, even though the PDAE structure requires a special treatment, it reduces the number of reduced operator coefficients to be learned, which is important for the numerical conditioning, robustness and snapshot data requirements of the OpInf implementation.

In general the lifting map $\mathbf{T}$ is not bijective. However, with our chosen lifting transformations the lifted state in (33) subsumes the original state variables u and ϕ. It is therefore straightforward to recover ROM estimates of the original variables without defining a reverse lifting map. Approximations introduced by the ROM will destroy the one-to-one mapping present in the lifted snapshot data. For example, using a ROM prediction of ϕ to compute a corresponding prediction of $K(\phi)$ will in general not give the same result as computing K directly from the lifted POD basis.

This discussion highlights the richness of possibilities for choosing lifting transformations, and the corresponding tradeoffs in computational complexity and numerical well-posedness (i.e., the polynomial degree of the lifted system and number of coefficients to be inferred), lifted system structure (i.e., PDEs versus PDAEs, sparsity structure), directly approximating physical quantities of interest (e.g., the primary variables of the original governing equations) through the choice of basis variables, and overall cost-accuracy performance of the resulting ROMs. The choices made in this paper balance these considerations as determined through exploring several different options. The optimal choices will be highly problem-dependent; automated strategies to learn/guide such choices are an important area of future work.

4.3. POD Basis and ROM Structure

With the lifting transformations chosen, the next step is to compute the POD basis. As already discussed in Section 3.1, one could use a separate POD basis for each variable to preserve the PDAE sparsity structure with respect to the physical variables. In addition, this choice allows for different POD basis dimensions for each variable, which gives more control over ROM accuracy. On the other hand, one could construct a single POD basis for all physical variables concatenated together, which can reduce the overall dimension of the entire POD basis. After exploring several options, we choose here to construct the POD basis matrix in a block-diagonal manner

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}^{(u)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}^{(\phi)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{V}^{(\mathrm{aux})} \end{bmatrix}, \quad \mathbf{W} = \mathbf{W}^{(\mathrm{aux})}, \tag{34}$$

with a separate basis for temperature ($\mathbf{V}^{(u)} \in \mathbb{R}^{n\times r_u}$), the order parameter ($\mathbf{V}^{(\phi)} \in \mathbb{R}^{n\times r_\phi}$), the five auxiliary differential variables ($\mathbf{V}^{(\mathrm{aux})} \in \mathbb{R}^{5n\times r^{\mathrm{d}}_{\mathrm{aux}}}$), and the auxiliary algebraic variable ($\mathbf{W}^{(\mathrm{aux})} \in \mathbb{R}^{n\times r^{\mathrm{a}}_{\mathrm{aux}}}$). We use reduced dimensions of r_u to approximate the temperature, r_ϕ to approximate the order parameter, $r^{\mathrm{d}}_{\mathrm{aux}}$ to approximate the auxiliary differential variables, and $r^{\mathrm{a}}_{\mathrm{aux}}$ to approximate the auxiliary algebraic variable. The dimension of bases $\mathbf{V}$ and $\mathbf{W}$ is $r_1 = r_u + r_\phi + r^{\mathrm{d}}_{\mathrm{aux}}$ and $r_2 = r^{\mathrm{a}}_{\mathrm{aux}}$, respectively. This choice resulted in a good tradeoff between accuracy and cost. It preserves the block sparsity that exists between u, ϕ, the auxiliary differential variables, and the auxiliary algebraic variable, although we lose any sparsity of interactions among the five auxiliary differential variables. It also has the advantage that we can more easily control accuracy of the representations for the original state variables u and ϕ, since they each have a separate basis.

Using the approximation (8) together with the basis choice in (34), we seek a ROM with the cubic structure (6)–(7). As described in Section 3, the reduced operators for the differential equations are inferred from the projected snapshot data using OpInf, while the reduced operators for the algebraic equations are computed explicitly as follows. For the particular lifting transformations that arise here, the only non-zero terms in the summations of (15) are the constant term $\boldsymbol{\mathcal{C}}$ with scalar $\mathscr{c} = -\gamma$, and one quadratic term $\boldsymbol{\mathcal{H}}^{(7,7)}$ with scalar $\mathscr{h}^{(7,7)} = \gamma$. This latter term corresponds to the coefficient multiplying m_0^2 (recall that m_0 is the seventh state variable in the lifted formulation of (33)). Hence the reduced-space representation of the algebraic equation is

$$\hat{\mathbf{z}} = \hat{\boldsymbol{\mathcal{C}}} + \hat{\boldsymbol{\mathcal{H}}}^{(7,7)}\left(\hat{\mathbf{y}}\otimes\hat{\mathbf{y}}\right), \quad \hat{\boldsymbol{\mathcal{C}}} = -\gamma\mathbf{W}^T\mathbf{1}_n, \;\; \hat{\boldsymbol{\mathcal{H}}}^{(7,7)} = \gamma\mathbf{W}^T\begin{bmatrix} \mathbf{V}^{(7)}_1 \otimes \mathbf{V}^{(7)}_1 \\ \mathbf{V}^{(7)}_2 \otimes \mathbf{V}^{(7)}_2 \\ \vdots \\ \mathbf{V}^{(7)}_n \otimes \mathbf{V}^{(7)}_n \end{bmatrix}. \tag{35}$$

This illustrates that the terms $\hat{\boldsymbol{\mathcal{C}}}$ and $\hat{\boldsymbol{\mathcal{H}}}^{(7,7)}$ can be computed explicitly from the basis matrices $\mathbf{V}$ and $\mathbf{W}$ in the offline stage. That is, once the basis matrices are determined, the reduced-space representation of the algebraic equation can be computed directly and does not need to be learned via operator inference. This is due to the known structure of the algebraic equations, which encode the pointwise relationships in (32) that arise during lifting.

4.4. Parametric ROM simulations

We conduct parametric studies considering variations in the latent heat, L_H, and the parameter γ appearing in the thermodynamic term (28). Both L_H and γ appear explicitly in the lifted equations (33). In addition, γ appears in the initial conditions for the evolution of auxiliary variable m_0. Other parameters are set to the values $\tau = 0.0003$, $\xi = 0.01$, $\beta = 0.9$, $u_M = 1.0$, $K_0 = 1$, and $K_1 = 0.1$. The one-dimensional spatial domain has length $\ell = 1$ and is discretized into $n = 1000$ cells (chosen to ensure that the diffuse interface is sufficiently resolved with $1/n = \xi/10)$. The timestep for numerical simulations is chosen to be $\Delta t = 8.33 \times 10^{-7}$ based on a numerical convergence study. Simulations are initialized by setting the initial solid fraction to cover 10% of the spatial domain (i.e., $\phi(x, 0) = 1$ for $x < 0.1$ and $\phi(x, 0) = 0$ otherwise) and the initial temperature profile is such that the solidified region is at melting temperature, and the remainder of the spatial domain is at the undercooling temperature of zero as defined in (30). The final time for the simulations is chosen to be $t_f = 0.03$ resulting in $K = 36000$ snapshots. The time-stepping scheme for the heat equation and the phase-field equation are Crank-Nicolson, and Strang splitting, respectively.

The OpInf minimization problem (14) is augmented with Tikhonov regularization to guard against ill-conditioning and over-fitting. Tikhonov regularization is chosen following the studies in [53]. This choice maintains the linearity of the regression problem while addressing the numerical ill-posedness that appears in the OpInf optimization problems. Other choices of regularization are possible, but not yet explored in the literature. The criterion for choosing the optimal regularization parameter is to minimize the relative error between the ROM predictions and the FOM snapshots of the original variables (i.e., temperature and order parameter) over a set of regularization parameters. This approach leads to stable ROMs. The L-curve criterion [58], which is often used in the literature to select the optimal regularization parameter, resulted here in unstable ROMs for a number of the studied cases.

Parametric studies are conducted on variations of the two process parameters L_H and γ. A global POD basis is constructed by performing SVD on the global snapshot matrix obtained from concatenating the snapshots corresponding to all parameters sampled in the training set [7]. The time derivatives $\dot{\mathbf{Y}}$ needed for OpInf are estimated numerically from the snapshots using a five-point central difference scheme (as in [59]) which is fourth-order accurate. Reduced operators are inferred for each parametric condition in the training set, so that we compute multiple ROMs each defined at a single parametric condition. ROM solutions for parameter values not in the training set are computed using linear interpolation of ROM solutions corresponding to the neighboring samples in the training set. The lifting map of (32) gives rise to an affine parametric dependence in (33). The affine structure can be utilized in a systematic way for a more efficient parametric model reduction strategy [7], which is the subject of future studies.

The efficacy of the method is tested by evaluating the relative error of ROM versus FOM solutions for every sample in the test set. The errors are also evaluated for the samples in the training set. For the target problem in this study, error metrics are defined for ROM predictions of temperature, order parameter, and interface location. The interface location is determined in a post-processing step, defined by the point at which $\phi = 0.5$ and computed by interpolating the estimated order parameter field. The error metrics for temperature and order parameter are defined by the norm of the difference between full and reduced solutions divided by the norm of the full model solution (i.e., $\left\|\widetilde{\mathbf{U}} - \mathbf{U}\right\|_F^2 / \|\mathbf{U}\|_F^2$

and $\left\|\widetilde{\boldsymbol{\Phi}} - \boldsymbol{\Phi}\right\|_F^2 / \left\|\boldsymbol{\Phi}\right\|_F^2$ for temperature and order parameter, respectively, where $\mathbf{U}$ and $\boldsymbol{\Phi}$ are the FOM solutions, and $\widetilde{\mathbf{U}}$ and $\widetilde{\boldsymbol{\Phi}}$ are the reconstructed ROM solutions for temperature and order parameter, respectively). The interface location prediction error is defined to be $\|\widetilde{\mathbf{x}}_{\text{interface}} - \mathbf{x}_{\text{interface}}\|_2 / \|\mathbf{x}_{\text{interface}}\|_2$ where $\mathbf{x}_{\text{interface}}$ and $\widetilde{\mathbf{x}}_{\text{interface}}$ are the FOM and ROM predictions of the location of the interface, respectively, determined by postprocessing on $\boldsymbol{\Phi}$ and $\widetilde{\boldsymbol{\Phi}}$.

For the first set of numerical results, we set $\gamma = 10$ and vary the latent heat between $L_H = 0$ and $L_H = 1$. The training and test sets are:

$$L_H^{\text{train}} = \{0.0,\ 0.1,\ \cdots,\ 0.9,\ 1.0\}, \quad L_H^{\text{test}} = \{0.05,\ 0.15,\ \cdots,\ 0.85,\ 0.95\}.$$

The global basis matrices $\mathbf{V}^{(u)}$, $\mathbf{V}^{(\phi)}$, $\mathbf{V}^{(\text{aux})}$, and $\mathbf{W}^{(\text{aux})}$ are obtained by performing SVD on a snapshot matrix constructed by concatenating snapshots of the respective variables from all training simulations with Δt intervals. Figure 2 shows the POD singular values (normalized by the largest singular value in each case) and the energy in the neglected modes ($1 - \frac{\sum_{i=1}^{r} \sigma_i^2}{\sum_{i=1}^{n} \sigma_i^2}$) for a given basis dimension. Due to the moving front nature of the solidification phenomena, the decay of the singular values is slow, which makes this a challenging problem for model reduction.

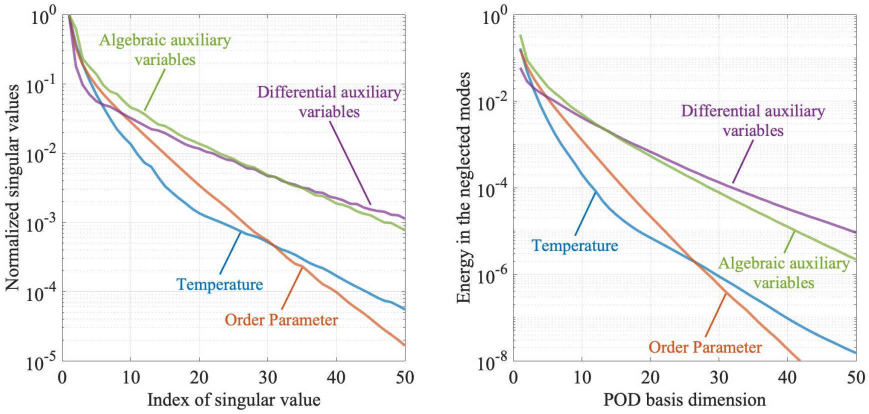

Figure 2: Left: the decay in the singular values of the snapshot matrices for temperature, order parameter, and auxiliary states in differential and algebraic form, varying the latent heat. Right: energy in the neglected POD modes for a given basis dimension.

Figure 3 shows the dimensions of each component of the basis matrices in (34) for increasing retained POD energy. As indicated in Figure 2, the singular value decay rate for the auxiliary variables is slower than that for the temperature and order parameter, and therefore the auxiliary states have a larger contribution to the total basis dimensions in Figure 3.

We assess the ROM performance by plotting relative errors for each of temperature, order parameter, and interface location in Figure 4. The reduced basis dimension r for the data points corresponds to the retained energies listed in Figure 3. Figure 4 shows the mean relative errors averaged over all samples in

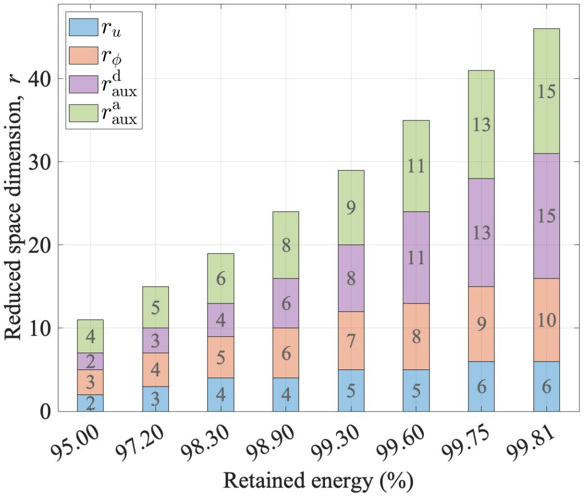

Figure 3: POD basis dimensions for variations in the latent heat at different levels of retained POD energy.

the test set, as well as the relative errors for the L_H values at which the ROM performs the best (minimum error) and the worst (maximum error). Mean, minimum, and maximum relative errors are also plotted for the training set. The relative errors in temperature and order parameter decrease with the increase in the dimension of the POD basis, as expected. The interface location is not directly approximated by the POD basis, and while its accuracy improves, the error reduction is less than that observed for temperature and order parameter.

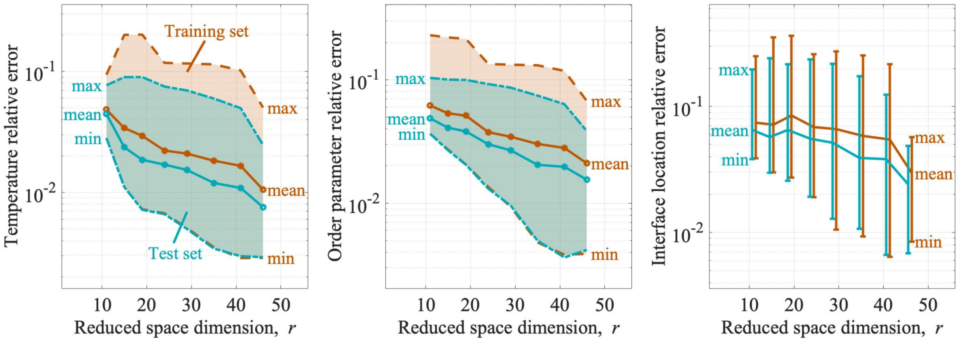

Figure 4: Relative errors in temperature (left), order parameter (middle), and interface location (right) for varying latent heat.

Figure 4 shows that the spread of the relative error for the test set is smaller than that of the training set, and that (unexpectedly) the test set has lower mean relative error than the training set. This is an artifact of the latent heat values selected in L_H^{train} and L_H^{test}. As Figure 5 shows for two different basis dimensions, the error is larger for the training set endpoint values $L_H = 0$ and $L_H = 1$. Following typical reduced modeling best practices, the test set was chosen to interpolate the training set and so has lower errors at its endpoints.

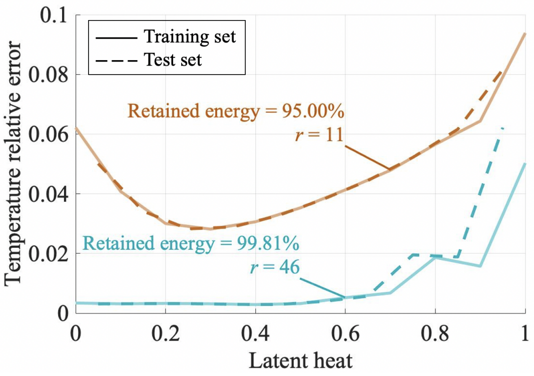

Figure 5: Variation of the temperature relative error versus L_H for two values of the retained energy over the training set (solid line) and the test set (dashed line).

Table 1 tabulates the online runtime of the ROM for different retained energies. The following time measurements are wall-clock time obtained from a MATLAB 2020 implementation on a local machine with 1.7 GHz Quad-Core Intel Core i7 CPU and 16GB RAM. The runtimes are averaged over 10 runs. The last row of the table reports the speedup achieved by the ROM, which is determined as the relative ratio of the runtime of the FOM to the runtime of the ROM. The timestep is the same for both the FOM and the ROM solver; however, the FOM is solved implicitly whereas the ROM is solved explicitly. An explicit scheme for the FOM with the current timestep would result in an unstable system. For total reduced basis dimensions smaller than 35, more than one order of magnitude of speedup is achieved. As expected, the magnitude of the speedup decreases with increase in the reduced basis dimension.

retained energy	95.00	97.20	98.30	98.90	99.30	99.60	99.75	99.81
r	11	15	19	24	29	35	41	46
runtime (sec)	1.05	1.63	2.14	3.11	8.11	11.48	17.03	24.34
speedup	109.71	70.57	53.63	36.86	14.15	9.99	6.73	4.71

Table 1: Online runtimes for the ROM for different reduced space dimensions r, and the speedup of the ROM over the FOM for the numerical example on variations of the latent heat. The online runtimes are averaged over 10 runs. The runtime for the FOM is 114.69 sec. Speedups of more than one order of magnitude are achieved for smaller reduced basis dimensions.

We present a final numerical example that studies variation of the γ term appearing in the thermodynamical driving force q. The latent heat is set to $L_H = 1.0$. The snapshots generated for training and testing use the following values:

$$\gamma^{\text{train}} = \{0.25,\ 0.75,\ \cdots,\ 4.75,\ 5.25\}, \quad \gamma^{\text{test}} = \{0.5,\ 1.0,\ \cdots,\ 4.5,\ 5.0\}.$$

The global basis matrix is determined from concatenating the snapshots for all samples from the training set.

Figure 6 shows the normalized singular values and energy in the neglected POD modes as we increase the dimension of the basis components of (34) and 520 the dimensions of these basis components are shown in Figure 7. Again, we see the slower decay in the singular values for the auxiliary variables, requiring their basis dimensions to be larger than those for the temperature and order parameter. The ROM relative errors are plotted in Figure 8 where the reduced basis dimension of the data points corresponds to the retained energies listed 525 in Figure 7. The ROM error decays more rapidly for this case than it did for variations in the latent heat. This is consistent with the singular value decay rates, which illustrate that varying the latent heat leads to a more complicated set of snapshot dynamics.

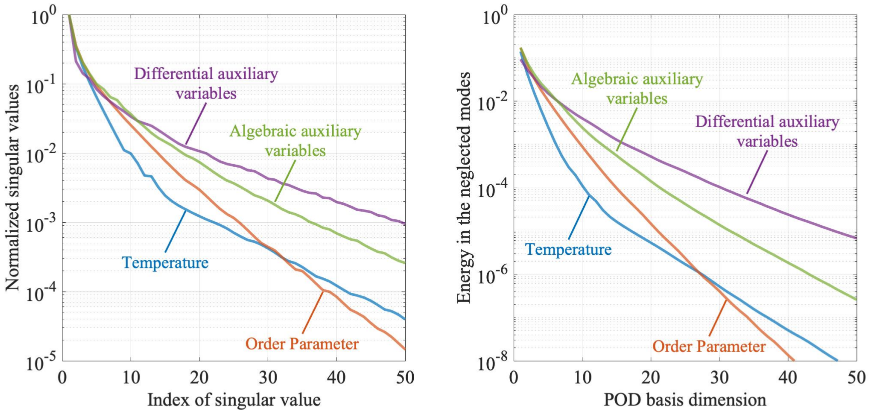

Figure 6: Left: the decay in the singular values of the snapshot matrices for temperature, order parameter, and auxiliary states in differential and algebraic form, varying γ. Right: energy in the neglected POD modes for a given basis dimension.

Figure 9 compares the FOM and ROM solutions for the samples from the 530 test set with maximum and minimum error. The retained POD energy for the ROM is 99.90%, giving a total POD basis dimension of 47. We see that the ROM is capable of capturing the solidification dynamics. Oscillations about the solid–liquid interface noticed in the ROM solution for ϕ are due to the Gibbs phenomenon induced by sharp transitions in the order parameter field. 535 Table 2 reports the online runtimes for the ROM and the speedup over the FOM for different magnitudes of the retained POD energy. The runtime for a one-dimensional solidification example is 114.17 sec, and speedups of more than an order of magnitude can be achieved by the ROM for most basis dimensions.

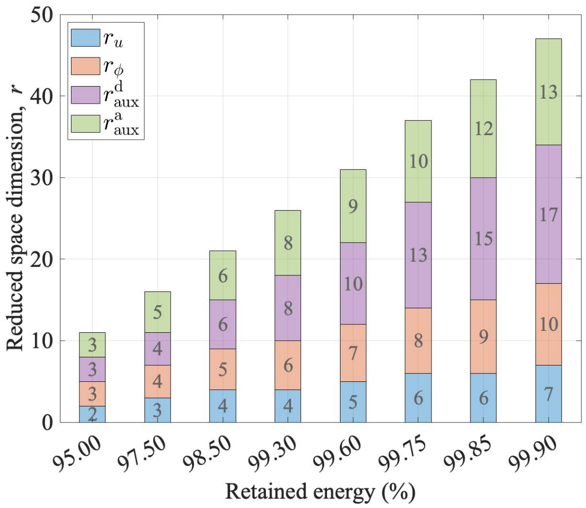

Figure 7: POD basis dimensions for variations in γ at different levels of retained POD energy.

retained energy	95.00	97.50	98.50	99.30	99.60	99.75	99.85	99.90
r	11	16	21	26	31	37	42	47
runtime (sec)	1.17	1.54	2.59	5.58	8.86	13.61	18.71	27.17
speedup	97.29	74.36	44.02	20.45	12.89	8.39	6.10	4.20

Table 2: Runtimes of the ROM for different retained POD energies, and the speedup achieved by the ROM over the FOM for the numerical example on variations in γ. The runtime for the FOM is 114.17 sec.

5. Concluding Remarks

This paper has proposed and demonstrated a non-intrusive data-driven model reduction method that addresses the PDAE structure arising in lifting nonlinear systems to polynomial form. The approach provides a new alternative for model reduction of highly nonlinear systems for which more classical hyper-reduction techniques may be ineffective. The approach is effective for the studied solidification problem; however, the relatively slow decay of the POD singular values points to the inefficiencies of representing transport-dominated dynamics in a static linear basis. A fruitful direction of future work is to combine the approach proposed here with a localized [60, 61] and/or adaptive basis [62–64], although it remains an open question how to achieve this in a non-intrusive way. The solidification model also highlights the interesting question of how to optimally define the low-dimensional basis. Using a separate basis for each physical quantity preserves the sparsity of the lifted PDAEs, but may come at a cost of increased total ROM dimension. The results presented here used four separate bases—one for temperature, one for order parameter, one for the auxiliary differential variables, and one for the auxiliary algebraic variable. Through numerical experiments, this choice was found to provide a good tradeoff between

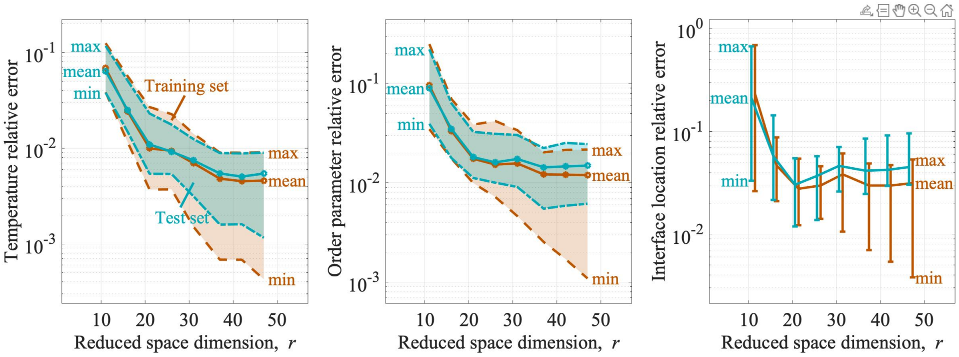

Figure 8: Relative errors in temperature (left), order parameter (middle), and interface location (right) for varying γ.

block-sparsity and overall ROM dimension, leading to efficiency in the resulting ROMs. Formalizing the process of optimal basis design is another necessary area of future research. In this study, the parametric dependencies have been accounted for through interpolation which is efficient for the one-dimensional parametric space studied in this paper. However, for high-dimensional parametric spaces, interpolation of the ROM solutions becomes challenging and inefficient.

6. Acknowledgments

This work has been supported in part by the U.S. Department of Energy AEOLUS MMICC center under award DE-SC0019303, program manager W. Spotz. The authors acknowledge helpful discussions with Y. Bao, G. Biros, S. DeWitt and B. Radhakrishnan in creating the solidification model implementation.

References

[1] B. Kramer, K. E. Willcox, Nonlinear model order reduction via lifting transformations and proper orthogonal decomposition, AIAA Journal 57 (6) (2019) 2297–2307.

[2] E. Qian, B. Kramer, B. Peherstorfer, K. Willcox, Lift & Learn: Physics-informed machine learning for large-scale nonlinear dynamical systems, Physica D: Nonlinear Phenomena 406 (2020) 132401.

[3] B. Peherstorfer, K. Willcox, Data-driven operator inference for nonintrusive projection-based model reduction, Computer Methods in Applied Mechanics and Engineering 306 (2016) 196 – 215.

[4] B. Radhakrishnan, S. B. Gorti, J. A. Turner, R. Acharya, J. A. Sharon, A. Staroselsky, T. El-Wardany, Phase field simulations of microstructure evolution in IN718 using a surrogate Ni–Fe–Nb alloy during laser powder bed fusion, Metals 9 (1) (2019) 14.

[5] M. Redeker, B. Haasdonk, A POD–EIM reduced two-scale model for crystal growth, Advances in Computational Mathematics 41 (5) (2015) 987–1013.

[6] A. Antoulas, Approximation of large-scale dynamical systems, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2005.

[7] P. Benner, S. Gugercin, K. Willcox, A survey of projection-based model reduction methods for parametric dynamical systems, SIAM Review 57 (4) (2015) 483 – 531.

[8] P. Benner, M. Ohlberger, A. Cohen, K. Willcox, Model Reduction and Approximation: Theory and Algorithms, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2017.

[9] P. Benner, M. Ohlberger, A. Patera, G. Rozza, K. Urban, Model Reduction of Parametrized Systems, Springer, Cham, Switzerland, 2017.

[10] M. Barrault, Y. Maday, N. C. Nguyen, A. T. Patera, An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations, Comptes Rendus Mathematique 339 (9) (2004) 667 – 672.

[11] S. Chaturantabut, D. C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, SIAM Journal on Scientific Computing 32 (5) (2010) 2737 – 2764.

[12] E. H. Kerner, Universal formats for nonlinear ordinary differential systems, Journal of Mathematical Physics 22 (7) (1981) 1366–1371.

[13] M. A. Savageau, E. O. Voit, Recasting nonlinear differential equations as S-systems: A canonical nonlinear form, Mathematical Biosciences 87 (1) (1987) 83–115.

[14] I. Mezić, Analysis of fluid flows via spectral properties of the Koopman operator, Annual Review of Fluid Mechanics 45 (2013) 357–378.

[15] M. Korda, I. Mezić, Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control, Automatica 93 (2018) 149–160.

[16] C. Gu, QLMOR: A projection-based nonlinear model order reduction approach using quadratic-linear representation of nonlinear systems, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 30 (9) (2011) 1307 – 1320.

[17] P. Benner, T. Breiten, Two-sided projection methods for nonlinear model order reduction, SIAM Journal on Scientific Computing 37 (2) (2015) B239–B260.

[18] O. Ghattas, K. Willcox, Learning physics-based models from data: perspectives from inverse problems and model reduction, Acta Numerica 30 (2021) 1–111.

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

[20] S. L. Brunton, J. L. Proctor, J. H. Tu, J. N. Kutz, Compressed sensing and dynamic mode decomposition, Journal of Computational Dynamics 2 (2) (2015) 165.

[21] J. N. Kutz, S. L. Brunton, B. W. Brunton, J. L. Proctor, Dynamic mode decomposition: data-driven modeling of complex systems, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2016.

[22] H. Arbabi, I. Mezic, Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the Koopman operator, SIAM Journal on Applied Dynamical Systems 16 (4) (2017) 2096–2126.

[23] I. V. Gosea, I. Pontes Duff, Toward fitting structured nonlinear systems by means of dynamic mode decomposition, in: P. Benner, T. Breiten, H. Faßbender, M. Hinze, T. Stykel, R. Zimmermann (Eds.), Model Reduction of Complex Dynamical Systems, Springer International Publishing, Cham, Switzerland, 2021, pp. 53–74.

[24] P. Benner, P. Goyal, B. Kramer, B. Peherstorfer, 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) 113433.

[25] H. V. Ly, H. T. Tran, Modeling and control of physical processes using proper orthogonal decomposition, Mathematical and Computer Modelling 33 (1-3) (2001) 223–236.

[26] C. Audouze, F. De Vuyst, P. Nair, Reduced-order modeling of parameterized pdes using time–space-parameter principal component analysis, International journal for numerical methods in engineering 80 (8) (2009) 1025–1057.

[27] F. Alsayyari, Z. Perkó, M. Tiberga, J. L. Kloosterman, D. Lathouwers, A fully adaptive nonintrusive reduced-order modelling approach for parametrized time-dependent problems, Computer Methods in Applied Mechanics and Engineering 373 (2021) 113483.

[28] W. Chen, J. S. Hesthaven, B. Junqiang, Y. Qiu, Z. Yang, Y. Tihao, Greedy nonintrusive reduced order model for fluid dynamics, AIAA Journal 56 (12) (2018) 4927–4943.

[29] R. Swischuk, L. Mainini, B. Peherstorfer, K. Willcox, Projection-based model reduction: Formulations for physics-based machine learning, Computers & Fluids 179 (2019) 704–717.

[30] A. C. Ionita, A. C. Antoulas, Data-driven parametrized model reduction in the Loewner framework, SIAM Journal on Scientific Computing 36 (3) (2014) A984–A1007.

[31] B. Peherstorfer, S. Gugercin, K. Willcox, Data-driven reduced model construction with time-domain Loewner models, SIAM Journal on Scientific Computing 39 (5) (2017) A2152–A2178.

[32] I. V. Gosea, A. C. Antoulas, Data-driven model order reduction of quadratic-bilinear systems, Numerical Linear Algebra with Applications 25 (6) (2018) e2200.

[33] A. C. Antoulas, I. V. Gosea, M. Heinkenschloss, On the Loewner framework for model reduction of Burgers' equation, in: Active Flow and Combustion Control 2018, Springer, Cham, Switzerland, 2019, pp. 255–270.

[34] D. S. Karachalios, I. V. Gosea, A. C. Antoulas, On bilinear time-domain identification and reduction in the loewner framework, in: P. Benner, T. Breiten, H. Faßbender, M. Hinze, T. Stykel, R. Zimmermann (Eds.), Model Reduction of Complex Dynamical Systems, Springer International Publishing, Cham, Switzerland, 2021, pp. 3–30.

[35] V. Mehrmann, T. Stykel, Balanced truncation model reduction for large-scale systems in descriptor form, in: Dimension Reduction of Large-Scale Systems, Springer, Berlin, Heidelberg, 2005, pp. 83–115.

[36] S. Gugercin, T. Stykel, S. Wyatt, Model reduction of descriptor systems by interpolatory projection methods, SIAM Journal on Scientific Computing 35 (5) (2013) B1010–B1033.

[37] S. Grundel, L. Jansen, N. Hornung, T. Clees, C. Tischendorf, P. Benner, Model order reduction of differential algebraic equations arising from the simulation of gas transport networks, in: Progress in Differential-Algebraic Equations, Springer, Berlin, Heidelberg, 2014, pp. 183–205.

[38] N. Banagaaya, S. Grundel, P. Benner, Index-aware MOR for gas transport networks, in: IUTAM Symposium on Model Order Reduction of Coupled Systems, Stuttgart, Germany, May 22–25, 2018, Springer, Cham, Switzerland, 2020, pp. 191–207.

[39] N. Banagaaya, G. Alì, W. Schilders, Index-aware Model Order Reduction Methods, Atlantis Press, Paris, France, 2016.

[40] L. Sirovich, Turbulence and the dynamics of coherent structures. I. coherent structures, Quarterly of Applied Mathematics 45 (3) (1987) 561–571.

[41] P. Astrid, S. Weiland, K. Willcox, T. Backx, Missing point estimation in models described by proper orthogonal decomposition, IEEE Transactions on Automatic Control 53 (10) (2008) 2237–2251.

[42] N. C. Nguyen, J. Peraire, An efficient reduced-order modeling approach for non-linear parametrized partial differential equations, International Journal for Numerical Methods in Engineering 76 (1) (2008) 27 – 55.

[43] N. C. Nguyen, A. T. Patera, J. Peraire, A 'best points' interpolation method for efficient approximation of parametrized functions, International Journal for Numerical Methods in Engineering 73 (4) (2008) 521 – 543.

[44] D. Galbally, K. Fidkowski, K. Willcox, O. Ghattas, Non-linear model reduction for uncertainty quantification in large-scale inverse problems, International Journal for Numerical Methods in Engineering 81 (12) (2010) 1581 – 1608.

[45] S. Chaturantabut, D. C. Sorensen, Application of POD and DEIM on dimension reduction of non-linear miscible viscous fingering in porous media, Mathematical and Computer Modelling of Dynamical Systems 17 (4) (2011) 337 – 353.

[46] D. Xiao, F. Fang, A. G. Buchan, C. C. Pain, I. M. Navon, J. Du, G. Hu, Non-linear model reduction for the Navier–Stokes equations using residual DEIM method, Journal of Computational Physics 263 (2014) 1–18.

[47] A. Radermacher, S. Reese, POD-based model reduction with empirical interpolation applied to nonlinear elasticity, International Journal for Numerical Methods in Engineering 107 (6) (2016) 477 – 495.

[48] P. Tiso, D. J. Rixen, Discrete empirical interpolation method for finite element structural dynamics, in: Topics in Nonlinear Dynamics, Volume 1, Springer, New York, NY, 2013, pp. 203 – 212.

[49] Y. Wang, I. M. Navon, X. Wang, Y. Cheng, 2D Burgers equation with large Reynolds number using POD/DEIM and calibration, International Journal for Numerical Methods in Fluids 82 (12) (2016) 909 – 931.

[50] M. A. Cardoso, L. J. Durlofsky, P. Sarma, Development and application of reduced-order modeling procedures for subsurface flow simulation, International Journal for Numerical Methods in Engineering 77 (9) (2009) 1322 – 1350.

[51] C. Huang, J. Xu, K. Duraisamy, C. Merkle, Exploration of reduced-order models for rocket combustion applications, in: 2018 AIAA Aerospace Sciences Meeting, 2018, p. 1183.

[52] T. G. Kolda, B. W. Bader, Tensor decompositions and applications, SIAM Review 51 (3) (2009) 455–500.

[53] S. A. McQuarrie, C. Huang, K. E. Willcox, Data-driven reduced-order models via regularized operator inference for a single-injector combustion process, Journal of the Royal Society of New Zealand 51 (2021) 194–211.

[54] R. Kobayashi, Modeling and numerical simulations of dendritic crystal growth, Physica D: Nonlinear Phenomena 63 (3-4) (1993) 410–423.

[55] A. A. Wheeler, W. J. Boettinger, G. B. McFadden, Phase-field model for isothermal phase transitions in binary alloys, Physical Review A 45 (10) (1992) 7424.

[56] W. J. Boettinger, J. A. Warren, The phase-field method: simulation of alloy dendritic solidification during recalescence, Metallurgical and Materials Transactions A 27 (3) (1996) 657–669.

[57] W. J. Boettinger, J. A. Warren, C. Beckermann, A. Karma, Phase-field simulation of solidification, Annual Review of Materials Research 32 (1) (2002) 163 – 194.

[58] P. C. Hansen, The L-curve and its use in the numerical treatment of inverse problems, in: Computational Inverse Problems in Electrocardiology. Edited by: Johnston P. WIT Press, Citeseer, 2001, pp. 119–142.

[59] R. Swischuk, B. Kramer, C. Huang, K. Willcox, Learning physics-based reduced-order models for a single-injector combustion process, AIAA Journal 58 (6) (2020) 2658–2672. doi:10.2514/1.J058943.

[60] K. Smetana, A. T. Patera, Optimal local approximation spaces for component-based static condensation procedures, SIAM Journal on Scientific Computing 38 (5) (2016) A3318–A3356.

[61] A. Buhr, L. Iapichino, M. Ohlberger, S. Rave, F. Schindler, K. Smetana, Localized model reduction for parameterized problems, in: P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. Schindlers, L. Sileira (Eds.), In Model Order Reduction: Volume 2 Snapshot-Based Methods and Algorithms, De Gruyter, Berlin, 2020, pp. 245–306.

[62] C. Gräßle, M. Hinze, N. Scharmacher, POD for optimal control of the Cahn-Hilliard system using spatially adapted snapshots, in: European Conference on Numerical Mathematics and Advanced Applications, Springer, Cham, Switzerland, 2017, pp. 703–711.

[63] C. Gräßle, M. Hinze, J. Lang, S. Ullmann, POD model order reduction with space-adapted snapshots for incompressible flows, Advances in Computational Mathematics 45 (5) (2019) 2401–2428.

[64] B. Peherstorfer, Model reduction for transport-dominated problems via online adaptive bases and adaptive sampling, SIAM Journal on Scientific Computing 42 (5) (2020) A2803–A2836.

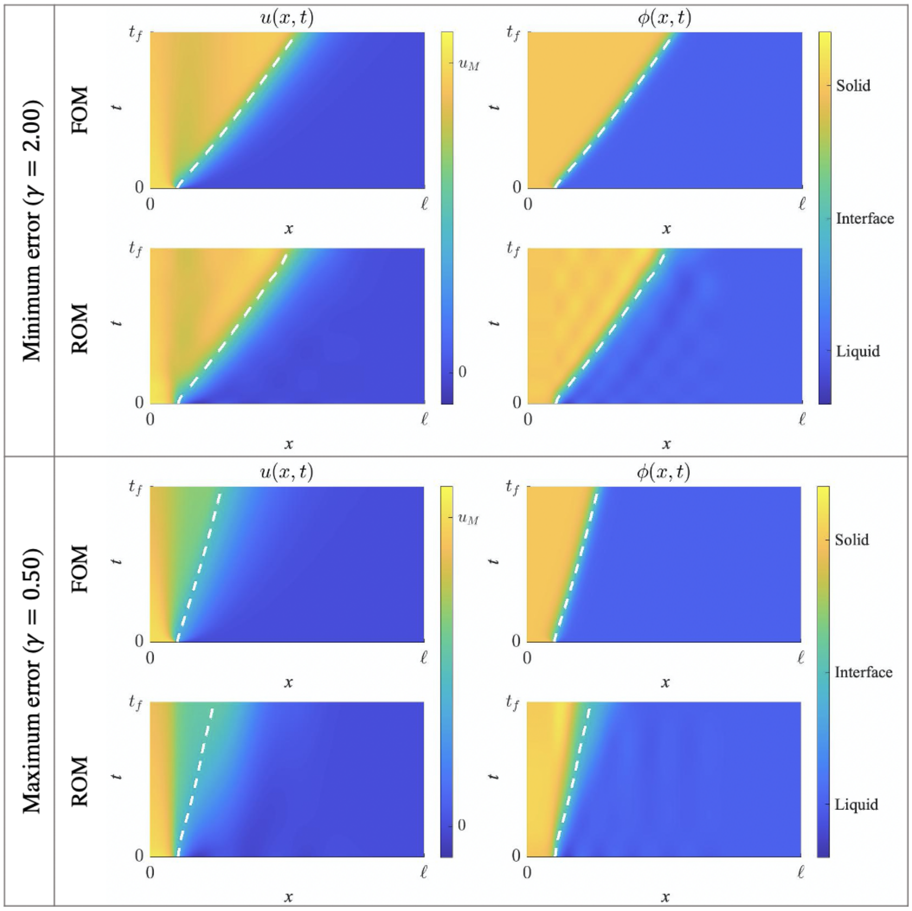

Figure 9: A comparison of the ROM and FOM solutions for the samples of the test set with minimum and maximum error ($\gamma = 2.0$ and $\gamma = 0.5$, respectively). The dashed line represents the location of the interface. The ROM solutions are shown for the retained POD energy of 99.90%.