RICE UNIVERSITY

Nonlinear Model Reduction via Discrete Empirical Interpolation

by

Saifon Chaturantabut

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Doctor of Philosophy

Approved, Thesis Committee:

Dr. Danny Sorensen, Chair
Noah G. Harding Professor of
Computational and Applied Mathematics

Dr. Matthias Heinkenschloss
Professor of Computational and Applied
Mathematics

Dr. Mark Embree
Professor of Computational and Applied
Mathematics

Dr. Matteo Pasquali
Professor of Chemical and Biomolecular
Engineering and Chemistry

Houston, Texas

May, 2011

Abstract

Nonlinear Model Reduction via Discrete Empirical Interpolation

by

Saifon Chaturantabut

This thesis proposes a model reduction technique for nonlinear dynamical systems based upon combining Proper Orthogonal Decomposition (POD) and a new method, called the Discrete Empirical Interpolation Method (DEIM). The popular method of Galerkin projection with POD basis reduces dimension in the sense that far fewer variables are present, but the complexity of evaluating the nonlinear term generally remains that of the original problem. DEIM, a discrete variant of the approach from [11], is introduced and shown to effectively overcome this complexity issue. State space error estimates for POD-DEIM reduced systems are also derived. These $\mathcal{L}^2$ error estimates reflect the POD approximation property through the decay of certain singular values and explain how the DEIM approximation error involving the nonlinear term comes into play. An application to the simulation of nonlinear miscible flow in a 2-D porous medium shows that the dynamics of a complex full-order system of dimension 15000 can be captured accurately by the POD-DEIM reduced system of dimension 40 with a factor of $\mathcal{O}(1000)$ reduction in computational time.

Acknowledgements

I would like to thank Prof. Dan Sorensen for being my great advisor. I am grateful for his insightful guidance, thoughtful understanding, support, encouragement, and friendship throughout the course of this work. I am indebted to Dan for giving me many invaluable opportunities which truly made a difference in my academic life. Without his help, most of my achievements would not have been possible nor would this thesis have been completed.

I would like to thank the thesis committee members: Prof. Matthias Heinkenschloss for providing some useful background references and the starting model problem; Prof. Mark Embree for his helpful suggestion on the error analysis in this thesis, for being a great instructor and for his incredibly detailed proof-reading with very useful comments; and Prof. Matteo Pasquali for his insightful suggestions on the polymer modeling problem. I also would like to thank Prof. Steve Cox and Anthony Kellems for providing the neuron model to initially test the algorithm in this thesis; Prof. Béatrice Rivière for suggesting the miscible flow model to illustrate an application of the proposed method and for giving related helpful advice and insightful comments; Dr. Jan Hewitt for helping me strengthen the thesis writing and presentation skills. Part of this thesis work was done during my visit at Delft University of Technology, NL, where Dan has spent his sabbatical leave during Fall 2010. I wish to thank Dan for this great opportunity and Prof. Marielba Rojas for being a great

host during my time there.

I would like to thank everyone in the CAAM department for their friendship, especially, Daria Lawrence, Brenda Aune, Fran Moshiri, Ivy Gonzalez, and Eric Aune for their wonderful assistance. Thanks also to all my dear friends in Houston for being part of my good memory during these five years.

My special thanks go to my family, my parents and my two sisters, for their endless encouragement, warm support, and unconditional love. I am greatly indebted to my Mom for always being there for me during my hard times and for the countless phone calls during these years away from home.

Above all, I thank God for the strength and wisdom to accomplish this work.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation and Goal

In many practical applications, such as in optimization, control, and uncertainty analysis, it is often necessary to provide real-time simulations that repeatedly solve discretized systems of differential equations describing the physical phenomena of interest. When the classical grid-based methods are used, the dimension of the resulting discretized systems can get extremely large in order to give highly accurate approximations. This is because each basis function (vector) of these grid-based methods is designed to capture only *local* dynamics around a few grid points, and not *global* characteristics of the system. Hence, performing these simulations can become computationally intensive or possibly infeasible.

Model order reduction can be used to reduce the computational complexity and

computational time of large-scale dynamical systems by approximations of much lower dimension that can produce nearly the same input/output response characteristics. This thesis proposes a method concerned with dimension reduction for high dimensional *nonlinear* ordinary differential equations (ODEs), which will be referred to as *full-order* systems. Although there are numerous important large-scale applications, such as circuit simulation and structural analysis, which are directly described by large systems of ODEs, systems of ODEs arising from discretization of partial differential equations (PDEs) will be primary examples in this thesis. Dimension reduction of discretized time dependent and/or parametrized nonlinear PDEs is of great value in reducing computational times in many applications, including the neuron modeling and two-phase miscible flows in porous media presented here as illustrations.

A common model reduction approach [4] is based on applying the Galerkin projection onto a *low dimensional* subspace, which is expected to contain dominant characteristics of the corresponding solution space. This subspace can be represented by a set of *reduced basis* functions (vectors) with global support which are *"learned"* ; they are constructed from high fidelity classical discretization schemes, such as finite difference (FD), finite volume (FV)[1], or finite element (FE) methods. These reduced basis functions are hence problem dependent. Fine scale detail is *encoded* in these global basis functions and this makes it possible to obtain good approximation with

[1]In the context of FD or FV methods, although there is no explicit notion of using basis functions, it can be thought of as using the standard basis vectors in $\mathbb{R}^n$ to span the solution at all grid points. Also, FD methods can be thought of as local interpolation polynomials.

relatively few basis functions.

Among the various techniques for obtaining a reduced basis, this thesis will focus upon the POD approach. This method constructs a reduced basis from many samples of the trajectories called *snapshots*. The reduced basis from POD is optimal in the sense that a certain approximation error concerning the snapshots is minimized. Thus, the space spanned by the basis from POD often gives an excellent low-dimensional approximation and it therefore has been used extensively in various applications. The POD approach will be used here as a starting point.

However, since the full-order systems of interest are nonlinear, the method of Galerkin projection with any type of reduced basis with global support, including the ones from POD, reduces dimension in the sense that far fewer variables are present, but the complexity of evaluating the nonlinear term generally remains that of the original problem, as explained with more detail in the next chapter. As a result, the computational complexity of the system is not truly reduced.

This thesis introduces a Discrete Empirical Interpolation Method (DEIM) to overcome this complexity issue. In particular, the DEIM is based upon replacing the orthogonal projection of POD with an oblique interpolatory projector. Evaluating the DEIM approximate nonlinear term does not require a prolongation of the reduced state variables back to the original high dimensional state approximation as in the POD-Galerkin approximation. Hence, DEIM improves the efficiency of the POD approximation and achieves a complexity reduction of the nonlinear term with

a complexity proportional to the number of reduced variables. An error bound for the DEIM approximation of a nonlinear vector-valued function is derived in this thesis. An analysis of DEIM is provided and shows that DEIM gives an approximation that is nearly as accurate as orthogonal projection but at greatly reduced cost. This analysis is then further used to develop a state-space error estimate for a reduced-order system constructed from POD-Galerkin approach with DEIM approximation. The derivation of this state-space error bound is based on an error estimate for the POD-Galerkin method given in [94], which shall be discussed in the next section along with other existing techniques for analyzing the accuracy and stability of the POD-Galerkin approach.

Throughout this thesis, a reduced-order system obtained directly from the POD-Galerkin projection will be referred to as the *POD reduced system* and the one obtained from the POD-Galerkin approach with the DEIM approximation will be referred to as the *POD-DEIM reduced system*. The 2-norm in the Euclidean space will be considered and denoted by $\|\cdot\|$. The following gives an overview of the existing work on projection-based model reduction using the reduced basis approach, particularly from POD, as well as the existing nonlinear model reduction techniques.

1.2 Existing Techniques

1.2.1 Techniques for Constructing Reduced Basis

A primary motivation for constructing a reduced basis comes from an observation that the solution space is often embedded in a manifold that has much lower dimension than the dimension of the ODE system derived through classical spatial discretization with a FE, FV, or FD approach. A reduced basis is often empirically derived through samples of trajectories and hence is generally problem dependent. That is, a set of selected solutions of the original full-order system is generally required for reduced-basis methods. The earliest examples of reduced-basis approaches are found in the applications of nonlinear structural analysis [64] and in the context of fluid flow simulations e.g. [66], [46]. The reduced bases used in these works include Lagrange, Taylor, and Hermite bases, which essentially consist of the state solution vectors and their derivatives. These state solutions are often called *snapshots*. Specifically, *a set of snapshots* consists of discrete samples of trajectories (e.g. state variables at certain time instances) associated with a particular set of inputs, initial and boundary conditions.

A number of recent model reduction approaches in the FE context are based on a *Reduced-Basis* (RB) approximation framework, where the basis is a set of solution snapshots specially selected with a greedy selection process [69, 54, 55, 93, 39, 63]. This framework possesses rigorous a posteriori error estimation procedures.

Alternatively, instead of directly using solution snapshots to form a reduced basis, POD can be applied to a set of snapshots to generate an orthonormal reduced basis that is optimal in the sense that a certain approximation error concerning the snapshots is minimized.

Existing Work on POD

POD has been successfully used with a Galerkin projection to provide reduced-order models in numerous applications such as compressible flow [78], computational fluid dynamics [50, 77], aerodynamics [14], and optimal control [48].

Many extensions and modifications of POD are proposed to improve the efficiency and accuracy for particular applications of interest. In [96], Willcox and Peraire proposed a technique which combines POD with the concept of balanced truncation to efficiently construct accurate reduced models for input-output systems in the application of control design. In [95], Willcox applied the *Gappy POD* technique proposed in [31] for handling incomplete ("gappy") data sets to reconstruct unsteady flow from limited available flow measurement data and to determine optimal sensor placement locations. Eftang, Knezevic and Patera proposed an extension of POD to the RB approximation framework in [30] by combining POD with a greedy sampling procedure in parameter space for parametrized parabolic PDEs. In the application of aeronautics where the solutions are sensitive to the changes in parameters, a sophisticated procedure based on *"interpolation"* on the tangent space of the Grassmann manifold

is proposed by Amsallem and Farhat [2, 1] for efficiently constructing an accurate and robust POD reduced system with respect to parameter variations.

The choice of the snapshot ensemble is a crucial factor in constructing a POD basis, and this choice can greatly affect the approximation of the original space of solutions. However, this issue shall not be discussed further in this thesis. The following discussions briefly review some recent techniques concerning snapshot selection. Most of them are developed specifically only for certain applications. Kunisch and Volkwein [51] suggested a way to avoid the dependence on the choice of the snapshots in optimal control applications. A model-constrained adaptive sampling is proposed in [15] for selecting the snapshots for large-scale systems with high-dimensional parametric input spaces. In the optimization application of static systems, Carlberg and Farhat [18] proposed a goal-oriented framework, so-called compact POD, using snapshots from state vectors and their sensitivity derivatives with respect to system input parameters.

Error Estimate and Stability Analysis for POD-Galerkin reduced system

Analyses of stability and accuracy of POD appear in several recent works. Han and Park [65] has shown that POD is robust to noise and can be used in conjunction with empirical data, which is typically characterized by noise. Prajna [68] provided the condition that guarantees preservation of stability and proposed a stability-preserving POD model reduction scheme. In [58], the authors applied the dual-weighted-residual

method, which uses the solution of a dual or adjoint system to obtain an error estimate for the solutions from POD reduced models of nonlinear systems. In [70], the error bounds of solutions from a POD reduced system were derived and the effects of small perturbations on the set of snapshots used for constructing the POD basis were studied. Subsequent work [44] proposed an alternative error estimation based on an adjoint method combined with the method of small sample statistical condition estimation. It also analyzed further the effect of perturbations in both the initial conditions and parameters on the resulting POD reduced system. However, the analysis in [44] is based on linearization, and hence, large perturbations may require some knowledge of the solution of the perturbed system. Some related works on error estimations such as in [88, 32, 58, 43] can be found in the extensive review from [44].

In [49, 50], Kunish and Volkwein derive error estimates for a POD reduced system for a class of nonlinear parabolic PDEs. Their analyses were done in a function-space setting, where the snapshots and the POD basis are in general Hilbert space. Kunish and Volkwein also considered a snapshot set that included finite difference quotients of the snapshots in response to their theoretical error bounds derived for the state solutions from the POD-Galerkin reduced system. The approximation errors were expressed as the contributions from the POD subspace approximation error and from time discretization error. The theoretical results in [50] provide asymptotic error estimates that do not depend on the snapshot set and demonstrate the effect of two different time discretizations used to produce the set of snapshots and for

the numerical integration of the reduced system. Nonlinear problems with Lipschitz continuous nonlinearities are considered in [49] and extended to the Navier-Stokes equations in [50]. Similar approaches for deriving the error estimates in the function space setting from [49, 50] were later applied within a finite dimensional Euclidean space setting in [94].

While the POD-Galerkin method and its extensions discussed above have been quite successful in substantially reducing the number of state variables, they typically fail to reduce the computational complexity involved with evaluating nonlinear terms. Unless there is a special structure, such as a bi-linear form, the evaluation of nonlinear terms has the same complexity as the full order system. Clearly, constructing reduced dimension approximations to the nonlinear terms that actually have complexity proportional to the number of reduced variables is of the highest priority. Several approaches have been proposed to address this fundamental issue.

1.2.2 Techniques for Nonlinearities

In the FE context, this inefficiency of the POD-Galerkin approach arises from the high computational complexity in repeatedly calculating the inner products required to evaluate the weak form of the nonlinearities, as discussed in [11, 38, 62]. In particular, in [62], Nguyen and Peraire discuss the limitations of such approaches and give a number of examples of equations involving non-polynomial nonlinearities. Specifically, they study linear elliptic equations with non-affine parameter dependence, non-

linear elliptic equations and non-linear time dependent convection-diffusion equations. They demonstrate for these examples that the standard POD-Galerkin approach does not admit the sort of pre-computation that is possible with polynomial nonlinearities. They propose a reduced basis approach with a best-points interpolation method (BPIM, see [61]) to selecting interpolation points.

Many nonlinear model reduction techniques have been proposed in the context of FD and FV discretizations, as well as differential-algebraic equations (e.g. in circuit simulation). Missing Point Estimation (MPE) was originally proposed by Astrid [6] to improve the complexity of the POD-Galerkin reduced system from FV discretization, essentially, by solving only a subset of equations of the original model. A reduced system is obtained by first extracting certain equations corresponding to specially chosen spatial grid points and then projecting the extracted system onto the space spanned by the restricted POD with components/rows corresponding to only these selected grid points. This procedure can be viewed as performing the Galerkin projection onto the truncated POD basis via a specially constructed inner product as defined in [9] that evaluates only at selected grid points instead of computing the usual $\mathcal{L}^2$ inner product. Two heuristic methods for selecting these spatial grid points are introduced in the thesis [6] (also in subsequent publications, e.g [5, 8, 7]) by aiming to minimize aliasing effects in using only partial spatial points. This was shown to be equivalent to a criterion for preserving the orthogonality of the restricted POD basis vectors, which is further translated into a criterion for controlling condition number

growth. These grid point selection procedures were later improved by incorporating a greedy algorithm from [95]. The applications of the MPE method are primarily in the context of a linear time varying system arising from FV discretization of a nonlinear computational fluid dynamic model for a glass melting furnace [6, 5, 8, 7]. It has also been used in modeling heat transfer in electrical circuits [89] and in subsurface flow simulation [17].

Alternatively, techniques for approximating a nonlinear function can be used in conjunction with the POD-Galerkin projection method to overcome this computational inefficiency. There are a number of examples that use model reduction approaches with nonlinear approximation based on pre-computation of coefficients defining multi-linear forms of polynomial nonlinearities followed by POD-Galerkin projection [20, 21, 67, 10, 28, 16]. One of these approaches is found in the trajectory piecewise-linear (TPWL) approximation proposed by Rewienski and White [74, 73], which is based on approximating a nonlinear function by a weighted sum of linearized models at selected points along a state trajectory. These linearization points are selected using prior knowledge from a training trajectory (or its approximation) of the full-order nonlinear system [72]. The TPWL approach was successfully applied to several practical nonlinear systems, especially in circuit simulations [71, 72, 73, 89, 12]. However, there are still many nonlinear functions that may not be approximated well by using low degree piecewise polynomials unless there are very many constituent polynomials.

More recently, Galbally et al. [33] applied the techniques of gappy POD, EIM, and BPIM to develop an approach to uncertainty quantification in a nonlinear combustion problem governed by an advection-diffusion-reaction PDE. The nonlinear term involved an exponential nonlinearity of Arrhenius type. In [33], there is a detailed explanation of why POD-Galerkin does not reduce the complexity of evaluating the nonlinear term. They also developed a masked projection framework that is very similar to the projection methodology developed in this thesis. Their work illustrates the similarity of the gappy POD, EIM and BPIM approaches.

Comparison of DEIM to Related Techniques

The DEIM approach proposed in this thesis approximates a nonlinear function by combining projection with interpolation. DEIM constructs specially selected interpolation indices that specify an interpolation based projection to provide a nearly $\mathcal{L}^2$ optimal subspace approximation to the nonlinear term without the expense of orthogonal projection. This approach is a discrete variant of the Empirical Interpolation Method (EIM) introduced by Barrault, Maday, Nguyen and Patera [11], which was originally posed in an empirically derived finite dimensional function space in the FE context. This DEIM variant was initially developed in order to apply to arbitrary systems of ODEs regardless of their origin, including the ones arising from FD and FV methods as well as the ODE system of coefficients derived from FE discretization. The EIM approximation [11] was initially proposed to be used with the *Reduced-*

Basis(RB) framework [39], whose basis functions would be the snapshots selected by an adaptive greedy selection process. In [11], this RB basis is used as an input to the EIM procedure for selecting the spatial interpolation points and each of these input basis functions will get *transformed* during this procedure. It can be shown that a mathematically equivalent approximation can be obtained without this transformation of the input basis [19]. In this thesis, the DEIM procedure for selecting the interpolation indices will instead use a POD basis as an input (although any type of basis would be valid) and will not transform the input basis as done in the EIM procedure.

The proposed DEIM approach is closely related to MPE in the sense that both methods employ a small selected set of spatial grid points to avoid computing the expensive $\mathcal{L}^2$ inner products at every time step that are required to evaluate the nonlinearities. However, the fundamental procedures for constructing a reduced system and the algorithms for selecting a set of spatial grid points are different. While MPE focuses on reducing the number of equations and using a restricted inner product on the POD basis vectors, DEIM focuses on approximating each nonlinear function, so that a certain coefficient matrix can be precomputed and, as a result, the complexity in evaluating the nonlinear term becomes proportional to the small number of selected spatial indices. Hence, the reduced system from the MPE procedure considers only a POD basis for the state variables, but the one from the DEIM procedure considers both a POD basis for the state variables and a POD basis related to each nonlin-

ear term. The POD-DEIM approach is also closely related to the approach called interpolation of function snapshots suggested in [89] as an alternative to MPE for constructing a reduced system for a nonlinear circuit model. The main steps of both approaches are the same. The nonlinear approximation is computed by using some selected spatial points, and then Galerkin projection is applied to the system. However, a key difference is that in [89] the basis matrices used for spanning the unknowns (state variables) and the nonlinear function in the reduced system are obtained from a least-squares solution of the snapshot matrices in such a way that the unknown coefficients of the resulting reduced system still have the original interpretations of state variables instead of using basis matrices from SVD truncation as done here in the POD-DEIM approach. No concrete algorithm was proposed in [89] for selecting indices (besides the ones used in MPE). However, it was suggested in [89] to select them to minimize an upper bound of the approximation error which is an idea similar to the one leading to our error bound for DEIM approximation (see (2.22) and (2.23) in §2.2.2).

1.3 Thesis Outline and Scope

This thesis is organized as follows. In Chapter 2, the problem formulation is given, with a brief background of POD and a review on model reduction via the POD-Galerkin approach. Then the DEIM approximation, which is the main focus of this thesis, is introduced along with its application for constructing POD-DEIM reduced

systems for nonlinear ODEs. The computational issue of the POD-Galerkin approach and the complexity reduction from applying DEIM are also discussed. Chapter 3 derives a state-space error estimate for POD-DEIM reduced systems introduced in Chapter 2. This derivation is particulary relevant to the nonlinear ODE systems arising from spatial discretizations of parabolic PDEs. Numerical examples are illustrated in Chapter 4 for a 1-D nonlinear PDE arising in neuron modeling and a nonlinear 2-D steady state problem. The purpose of this chapter is to demonstrate how to apply the POD-DEIM model reduction technique to some simple nonlinear problems. A more complex numerical application of the POD-DEIM approach is presented in Chapter 5 through the simulation of nonlinear miscible viscous fingering in a 2-D porous medium. The result in this chapter shows a substantial reduction in computational time of the POD-DEIM reduced system, e.g. by a factor of $\mathcal{O}(1000)$, while the accuracy is still retained. The failure of the POD-Galerkin approach to reduce the complexity of nonlinear terms is demonstrated in both Chapter 4 and Chapter 5. Finally, the conclusions and possible extensions of this thesis are discussed in Chapter 6.

Chapter 2

Nonlinear Model Reduction via Discrete Empirical Interpolations

This chapter presents a model reduction technique for nonlinear ordinary differential equations (ODEs). The problem formulation is first given in §2.1. Dimension reduction via Proper Orthogonal Decomposition (POD) with Galerkin projection is reviewed in §2.1.1 followed by a discussion of its fundamental complexity issue in §2.1.2. The Discrete Empirical Interpolation Method (DEIM) is then introduced in §2.2. The key to complexity reduction is to replace orthogonal projection of POD with the interpolation projection of DEIM. An algorithm for selecting the interpolation indices used in the DEIM approximation is presented in §2.2.1. Section 2.2.2 provides an error bound on this interpolatory approximation, indicating that it is nearly as good as orthogonal projection. The validity of this error bound and the

high quality of the DEIM approximations is illustrated in § 2.2.3 through numerical examples of nonlinear vector-valued functions. Section 2.2.4 explains how to apply the DEIM approximation to nonlinear terms in POD-Galerkin reduced models of FD discretized systems, and then the extension to general nonlinear ODEs will be given in §2.2.5. Finally, the computational complexity will be discussed in §2.2.6.

2.1 Problem Formulation

Although this chapter develops a method for reducing the dimension of general large scale ODE systems regardless of their origin, a considerable source of such systems is the semi-discretization of time dependent or parameter dependent PDEs. In this case, the nonlinearities in the resulting ODEs from the discretization are often in the form of componentwise-evaluation functions, which will be assumed here. Section 2.2.5 will illustrate how to handle general nonlinearities. This method will be developed here in the context of finite difference (FD) discretized systems arising from two types of nonlinear PDEs, which are used for our numerical computations in Chapters 4 and 5. One is time dependent and the other is a parametrized steady state problem. We have considered these two types separately in order to simplify the exposition; however, the two may be merged to address time dependent parametrized systems.

A FD discretization of a scalar nonlinear PDE in one spatial variable results in a system of nonlinear ODEs of the form

$$\frac{d}{dt}\mathbf{y}(t) = \mathbf{A}\mathbf{y}(t) + \mathbf{F}(\mathbf{y}(t)), \tag{2.1}$$

where $t \in [0, T]$ denotes time, $\mathbf{y}(t) = [\mathbf{y}_1(t), \ldots, \mathbf{y}_n(t)]^T \in \mathbb{R}^n$ is a vector of state variables with initial condition $\mathbf{y}(0) = \mathbf{y}_0 \in \mathbb{R}^n$, $\mathbf{A} \in \mathbb{R}^{n \times n}$ is a constant matrix, and $\mathbf{F}$ is a nonlinear function evaluated at $\mathbf{y}(t)$ componentwise, i.e., $\mathbf{F} = [F(\mathbf{y}_1(t)), \ldots, F(\mathbf{y}_n(t))]^T$, with a scalar-valued function $F : \mathcal{I} \mapsto \mathbb{R}$ for $\mathcal{I} \subset \mathbb{R}$. The matrix $\mathbf{A}$ is the discrete approximation of the linear spatial differential operator and F is a nonlinear function of a scalar variable.

Steady nonlinear PDEs (in several spatial dimensions) might give rise similarly to a corresponding FD discretized system of the form

$$\mathbf{A}\mathbf{y}(\mu) + \mathbf{F}(\mathbf{y}(\mu)) = 0, \tag{2.2}$$

with the corresponding Jacobian

$$\mathbf{J}(\mathbf{y}(\mu)) := \mathbf{A} + \mathbf{J}_{\mathbf{F}}(\mathbf{y}(\mu)), \tag{2.3}$$

where $\mathbf{y}(\mu) = [\mathbf{y}_1(\mu), \ldots, \mathbf{y}_n(\mu)]^T \in \mathbb{R}^n$; $\mathbf{A}$ and $\mathbf{F}$ are defined as for (2.1). Note that from (2.3), the Jacobian of the nonlinear function is a diagonal matrix given by

$$\mathbf{J}_{\mathbf{F}}(\mathbf{y}(\mu)) = \mathrm{diag}\{F'(\mathbf{y}_1(\mu)), \ldots, F'(\mathbf{y}_n(\mu))\} \in \mathbb{R}^{n \times n}, \tag{2.4}$$

where F' denotes the first derivative of F. The parameter $\mu \in \mathscr{D} \subset \mathbb{R}^d$, $d = 1, 2, \ldots$, generally represents the system's configuration in terms of its geometry, material properties, etc.

The dimension n of (2.1) and (2.2) reflects the number of spatial grid points used in the FD discretization. As noted, the dimension n can become extremely large

when high accuracy is required. This can lead to substantial increases in storage and computational requirements to solve these systems. Approximate models with much smaller dimensions are needed to recover the efficiency.

Projection-based techniques are commonly used for constructing a reduced-order system. They construct a reduced-order system of order $k \ll n$ that approximates the original system from a subspace spanned by a *reduced basis* of dimension k in $\mathbb{R}^n$. Galerkin projection is used here as the means for dimension reduction. In particular, let $\mathbf{V} \in \mathbb{R}^{n \times k}$ be a matrix whose orthonormal columns are the vectors in the reduced basis. Then by replacing $\mathbf{y}(t)$ in (2.1) by $\mathbf{V}\widehat{\mathbf{y}}(t)$, $\widehat{\mathbf{y}}(t) \in \mathbb{R}^k$ and projecting the system (2.1) onto $\mathbf{V}$, the reduced system of (2.1) is of the form

$$\frac{d}{dt}\widehat{\mathbf{y}}(t) = \underbrace{\mathbf{V}^T\mathbf{A}\mathbf{V}}_{\widehat{\mathbf{A}}}\widehat{\mathbf{y}}(t) + \mathbf{V}^T\mathbf{F}(\mathbf{V}\widehat{\mathbf{y}}(t)). \tag{2.5}$$

Similarly, the reduced-order system of (2.2) is of the form

$$\underbrace{\mathbf{V}^T\mathbf{A}\mathbf{V}}_{\widehat{\mathbf{A}}}\widehat{\mathbf{y}}(\mu) + \mathbf{V}^T\mathbf{F}(\mathbf{V}\widehat{\mathbf{y}}(\mu)) = 0, \tag{2.6}$$

with corresponding Jacobian

$$\widehat{\mathbf{J}}(\widehat{\mathbf{y}}(\mu)) := \widehat{\mathbf{A}} + \mathbf{V}^T\mathbf{J}_{\mathbf{F}}(\mathbf{V}\widehat{\mathbf{y}}(\mu))\mathbf{V}, \tag{2.7}$$

where $\widehat{\mathbf{A}} = \mathbf{V}^T\mathbf{A}\mathbf{V} \in \mathbb{R}^{k \times k}$. The choice of the reduced basis clearly affects the quality of the approximation. The techniques for constructing a set of reduced basis use a common observation that, for a particular system, the solution space is often attracted to a low dimensional manifold. POD constructs a set of global basis functions from

the singular value decomposition (SVD) of *snapshots*, which are discrete samples of trajectories $\mathbf{y}(\cdot)$ associated with a particular set of boundary conditions, parameter values and inputs. It is expected that the samples will be on or near the attractive manifold. Once the reduced model has been constructed from this reduced basis, it may be used to obtain approximate solutions for a variety of initial conditions and parameter settings, provided the set of samples is rich enough. This empirically derived basis is clearly dependent on the sampling procedure.

Among the various techniques for obtaining a reduced basis, POD constructs a reduced basis that is *optimal* in the sense that a certain approximation error concerning the snapshots is minimized. Thus, the space spanned by the basis from POD often gives an excellent low dimensional approximation. The POD approach is therefore used here as a starting point.

2.1.1 Proper Orthogonal Decomposition (POD)

Consider a set of snapshots $\{\mathbf{y}_1, \ldots, \mathbf{y}_{n_s}\} \subset \mathbb{R}^n$ and the corresponding snapshot matrix $\mathbb{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_{n_s}] \in \mathbb{R}^{n \times n_s}$. POD constructs an orthonormal basis that can represent dominant characteristics of the space of expected solutions, which is defined as Range$\{\mathbb{Y}\}$, the span of the snapshots. Let $r = \text{rank}\{\mathbb{Y}\}$. Consider a set of orthonormal basis vectors $\{\mathbf{v}_i\}_{i=1}^k \subset \mathbb{R}^n$ and the corresponding basis matrix $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_k] \in \mathbb{R}^{n \times k}$, for $k < r$. An approximation of a snapshot $\mathbf{y}_j$ in Range$\{\mathbf{V}\}$ is of the form $\mathbf{V}\widehat{\mathbf{y}}_j$ for some coefficient vector $\widehat{\mathbf{y}}_j \in \mathbb{R}^k$. Applying the Galerkin or-

thogonality condition of the residual $\mathbf{y}_j - \mathbf{V}\widehat{\mathbf{y}}_j$ to Range$\{\mathbf{V}\}$ gives $\mathbf{V}^T(\mathbf{y}_j - \mathbf{V}\widehat{\mathbf{y}}_j) = 0$, which implies $\widehat{\mathbf{y}}_j = \mathbf{V}^T\mathbf{y}_j$. That is, the approximation becomes $\mathbf{y}_j \approx \mathbf{V}\mathbf{V}^T\mathbf{y}_j$. POD provides an optimal orthonormal basis $\{\mathbf{v}_i\}_{i=1}^k \subset \mathbb{R}^n$ minimizing the sum of squared errors associated with these approximations for the snapshots. In particular, the POD basis matrix $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_k] \in \mathbb{R}^{n\times k}$ solves the minimization problem:

$$\min_{\text{rank}\{\mathbf{V}\}=k} \sum_{j=1}^{n_s} \|\mathbf{y}_j - \mathbf{V}\mathbf{V}^T\mathbf{y}_j\|^2, \qquad \text{s.t.} \qquad \mathbf{V}^T\mathbf{V} = \mathbf{I}_k, \tag{2.8}$$

where $\mathbf{I}_k \in \mathbb{R}^{k\times k}$ is an identity matrix. More details on POD can be found in, e.g., [50, 70]. Notice that, for $\mathbf{V}^T\mathbf{V} = \mathbf{I}_k$ and Frobenius norm $\|\cdot\|_F$,

$$\min_{\text{rank}\{\mathbf{V}\}=k} \sum_{j=1}^{n_s} \|\mathbf{y}_j - \mathbf{V}\mathbf{V}^T\mathbf{y}_j\|^2 = \min_{\text{rank}\{\mathbf{V}\}=k} \|\mathbb{Y} - \mathbf{V}\mathbf{V}^T\mathbb{Y}\|_F^2 = \min_{\text{rank}\{\mathbb{Y}_k\}=k} \|\mathbb{Y} - \mathbb{Y}_k\|_F^2.$$

The minimization problem (2.8) is therefore equivalent to the problem of low-rank approximation, which is well-known to be solved by the SVD of $\mathbb{Y}$. Hence, POD is essentially the same as a truncated SVD in the Euclidean space setting, which will be considered in this thesis. Specifically, a POD basis of dimension k for (2.8) is just a set of left singular vectors corresponding to the first k dominant singular values of the snapshot matrix $\mathbb{Y}$. The minimum sum of squared errors in the 2-norm from approximating the snapshots using the POD basis is given by

$$\sum_{j=1}^{n_s} \|\mathbf{y}_j - \mathbf{V}\mathbf{V}^T\mathbf{y}_j\|^2 = \sum_{i=k+1}^{r} \sigma_i^2, \tag{2.9}$$

for $k < r$, where $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_k] \in \mathbb{R}^{n\times k}$; $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_r \in \mathbb{R}^n$ are the singular vectors corresponding to the nonzero singular values $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$ of $\mathbb{Y}$. In the

large scale setting, the dominant singular values and vectors of $\mathbb{Y}$ can be efficiently computed by using MATLAB routine **svds** (or ARPACK). If $n \leq n_s$, one only need compute matrix-vector products of the form $\mathbf{w} = \mathbb{Y}(\mathbb{Y}^T\mathbf{v})$, while if $n > n_s$, it is usually more efficient to compute the dominant singular values and vectors of $\mathbb{Y}^T$ which will only require matrix-vector products of the form $\mathbf{w} = \mathbb{Y}^T(\mathbb{Y}\mathbf{v})$.

The choice of the snapshot ensemble is a crucial factor in constructing a POD basis. This choice can greatly affect the approximation of the original solution space, but it is a separate issue and will not be discussed here. POD works well in many applications and often provides an excellent reduced basis. However, as discussed next, when POD is used in conjunction with the Galerkin projection, effective dimension reduction is usually limited to the linear terms or low order polynomial nonlinearities. Systems with general nonlinearities need additional treatment, which will be presented in §2.2.

2.1.2 Complexity Issue of the POD-Galerkin Approach

This section illustrates the computational inefficiency that occurs in solving the reduced-order system that is directly obtained from the POD-Galerkin approach. Equation (2.5) has the nonlinear term

$$\widehat{\mathbf{N}}(\widehat{\mathbf{y}}) := \underbrace{\mathbf{V}^T}_{k\times n}\underbrace{\mathbf{F}(\mathbf{V}\widehat{\mathbf{y}}(t))}_{n\times 1}. \tag{2.10}$$

$\widehat{\mathbf{N}}(\widehat{\mathbf{y}})$ has a computational complexity that depends on n, the dimension of the original full-order system (2.1). It requires on the order of $2nk$ Flops for matrix-vector multiplications and it also requires a full evaluation of the nonlinear function $\mathbf{F}$ at the

n-dimensional vector $\mathbf{V}\widehat{\mathbf{y}}(t)$. In particular, suppose the complexity for evaluating the nonlinear function $\mathbf{F}$ with q components is $\mathcal{O}(\alpha(q))$, where α is some function of q. Then the complexity of the nonlinear term $\mathbf{F}(\mathbf{y}(t))$ in the original system is $\mathcal{O}(n)$ and the complexity for computing (2.10) is roughly $\mathcal{O}(\alpha(n) + 4nk)$. As a result, solving this system might still be as costly as solving the original system. Here, the $4nk$ flops are a result of the two matrix-vector products required to form the argument of $\mathbf{F}$ and then to form the projection. We count both the multiplications and additions as flops.

The same inefficiency occurs when solving the reduced-order system (2.6) for the steady nonlinear PDEs by Newton iteration. At each iteration, besides the nonlinear term of the form (2.10), the Jacobian of the nonlinear term (2.7) must also be computed with a computational cost that still depends on the full-order dimension n. I.e. from (2.7),

$$\widehat{\mathbf{J}}_{\mathbf{F}}(\widehat{\mathbf{y}}(\mu)) := \underbrace{\mathbf{V}^T}_{k\times n}\underbrace{\mathbf{J}_{\mathbf{F}}(\mathbf{V}\widehat{\mathbf{y}}(\mu))}_{n\times n}\underbrace{\mathbf{V}}_{n\times k}, \tag{2.11}$$

has computational complexity roughly $\mathcal{O}(\alpha(n) + 2n^2k + 2nk^2 + 2nk)$ if we treat $\mathbf{J}_{\mathbf{F}}$ as dense. The $2n^2k$ term becomes $\mathcal{O}(nk)$ if $\mathbf{J}_{\mathbf{F}}$ is sparse or diagonal.

2.2 Discrete Empirical Interpolation Method (DEIM)

An effective way to overcome the difficulty described in §2.1.2 is to approximate the nonlinear function in (2.5) or (2.6) by projecting it onto a subspace that approximates the space generated by the nonlinear function and that is spanned by a basis of dimension $m \ll n$. This section considers the nonlinear functions $\mathbf{F}(\mathbf{V}\widehat{\mathbf{y}}(t))$ and $\mathbf{F}(\mathbf{V}\widehat{\mathbf{y}}(\mu))$ of the reduced-order systems (2.5) and (2.6), respectively, represented by $\mathbf{f}(\tau)$, where $\tau = t$ or μ. The approximation from projecting $\mathbf{f}(\tau)$ onto the subspace spanned by the basis $\{\mathbf{u}_1, \ldots, \mathbf{u}_m\} \subset \mathbb{R}^n$ is of the form

$$\mathbf{f}(\tau) \approx \mathbf{U}\mathbf{c}(\tau), \tag{2.12}$$

where $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_m] \in \mathbb{R}^{n \times m}$ and $\mathbf{c}(\tau)$ is the corresponding coefficient vector. The vector $\mathbf{c}(\tau)$ can be determined by selecting m distinguished rows from the overdetermined system $\mathbf{f}(\tau) = \mathbf{U}\mathbf{c}(\tau)$. In particular, consider a matrix

$$\mathbf{P} = [\mathbf{e}_{\wp_1}, \ldots, \mathbf{e}_{\wp_m}] \in \mathbb{R}^{n \times m}, \tag{2.13}$$

where $\mathbf{e}_{\wp_i} = [0, \ldots, 0, 1, 0, \ldots, 0]^T \in \mathbb{R}^n$ is the $\wp_i$-th column of the identity matrix $\mathbf{I}_n \in \mathbb{R}^{n \times n}$, for $i = 1, \ldots, m$. Suppose $\mathbf{P}^T\mathbf{U}$ is nonsingular. Then the coefficient vector $\mathbf{c}(\tau)$ can be determined uniquely from

$$\mathbf{P}^T\mathbf{f}(\tau) = (\mathbf{P}^T\mathbf{U})\mathbf{c}(\tau), \tag{2.14}$$

and the final approximation from (2.12) becomes

$$\mathbf{f}(\tau) \approx \mathbf{U}\mathbf{c}(\tau) = \mathbf{U}(\mathbf{P}^T\mathbf{U})^{-1}\mathbf{P}^T\mathbf{f}(\tau). \tag{2.15}$$

Note that pre-multiplying a matrix by $\mathbf{P}^T$ is equivalent to *extracting* the rows $\wp_1, \ldots, \wp_m$ of that matrix, e.g. in MATLAB notation $\mathbf{P}^T\mathbf{U} = \mathbf{U}(\vec{\wp}, :) \in \mathbb{R}^{m \times m}$ with $\vec{\wp} = [\wp_1, \ldots, \wp_m]^T \in \mathbb{R}^m$, and therefore $\mathbf{P}$ should not be constructed explicitly in the actual computation. To obtain the approximation (2.15), we must specify

1. the projection basis $\{\mathbf{u}_1, \ldots, \mathbf{u}_m\}$;

2. the interpolation indices $\{\wp_1, \ldots, \wp_m\}$ used in (2.13).

The projection basis $\{\mathbf{u}_1, \ldots, \mathbf{u}_m\}$ for the nonlinear function $\mathbf{f}$ is constructed by applying the POD on the *nonlinear snapshots* obtained from the original full-order system. These nonlinear snapshots are the sets $\{\mathbf{F}(\mathbf{y}(t_1)), \ldots, \mathbf{F}(\mathbf{y}(t_{n_s}))\}$ and $\{\mathbf{F}(\mathbf{y}(\mu_1)), \ldots, \mathbf{F}(\mathbf{y}(\mu_{n_s}))\}$ obtained from (2.10) and (2.11), respectively. Note, these values are needed to generate the trajectory snapshots in $\mathbb{Y}$ and hence represent no additional cost other than the SVD required to obtain $\mathbf{U}$.

The interpolation indices $\wp_1, \ldots, \wp_m$, used for determining the coefficient vector $\mathbf{c}(\tau)$ in the approximation (2.12), are selected inductively from the basis $\{\mathbf{u}_1, \ldots, \mathbf{u}_m\}$ by the DEIM algorithm introduced in the next section.

2.2.1 DEIM: Algorithm for Interpolation Indices

DEIM is a discrete variant of the Empirical Interpolation Method (EIM) proposed by Barrault, Maday, Nguyen and Patera in [11] for constructing an approximation of a non-affine parametrized function with spatial variable defined in a continuous bounded domain Ω. The *continuous* domain Ω will be treated here as a *finite set*

of discrete points in Ω. The DEIM algorithm selects an index corresponding to one of these discrete spatial points at each iteration to limit growth of an error bound. This provides a derivation of a global error bound as presented in §2.2.2. For general systems of nonlinear ODEs that are not FD approximations to PDEs, this spatial connotation of indices will no longer exist. However, the formal procedure remains unchanged.

Algorithm 1: DEIM

INPUT : $\{\mathbf{u}_\ell\}_{\ell=1}^m \subset \mathbb{R}^n$ linearly independent

OUTPUT: $\vec{\wp} = [\wp_1, \ldots, \wp_m]^T \in \mathbb{R}^m$

1 $[|\rho|,\ \wp_1] = \texttt{max}\{|\mathbf{u}_1|\}$

2 $\mathbf{U} = [\mathbf{u}_1]$, $\mathbf{P} = [\mathbf{e}_{\wp_1}]$, $\vec{\wp} = [\wp_1]$;

3 **for** $\ell \leftarrow 2$ **to** m **do**

4 Solve $(\mathbf{P}^T\mathbf{U})\mathbf{c} = \mathbf{P}^T\mathbf{u}_\ell$;

5 $\mathbf{r} = \mathbf{u}_\ell - \mathbf{U}\mathbf{c}$

6 $[|\rho|\ \ \wp_\ell] = \texttt{max}\{|\mathbf{r}|\}$

7 $\mathbf{U} \leftarrow [\mathbf{U}\ \ \mathbf{u}_\ell]$, $\mathbf{P} \leftarrow [\mathbf{P}\ \ \mathbf{e}_{\wp_\ell}]$, $\vec{\wp} \leftarrow \begin{bmatrix} \vec{\wp} \\ \wp_\ell \end{bmatrix}$

8 **end**

The notation `max` in Algorithm 1 is the same as the function `max` in MATLAB. Thus, $[|\rho|,\ \wp_\ell] = \texttt{max}\{|\mathbf{r}|\}$ implies $|\rho| = |r_{\wp_\ell}| = \max_{i=1,\ldots,n}\{|r_i|\}$, with the smallest index taken in case of a tie. Note that, define $\rho := r_{\wp_\ell}$ in each iteration $\ell = 1, \ldots, m$.

From Algorithm 1, the DEIM procedure constructs a set of indices inductively on the input basis. The order of the input basis $\{\mathbf{u}_\ell\}_{\ell=1}^m$ according to the dominant singular values is important and an error analysis indicates that the POD basis is a suitable choice for this algorithm. The process starts by selecting the first interpolation index $\wp_1 \in \{1, \ldots, n\}$ corresponding to the entry of the first input basis $\mathbf{u}_1$ with largest magnitude. The remaining interpolation indices, $\wp_\ell$ for $\ell = 2, \ldots, m$, are selected so that each of them corresponds to the entry with the *largest* magnitude of the *residual* $\mathbf{r} = \mathbf{u}_\ell - \mathbf{U}\mathbf{c}$ from line 5 of Algorithm 1. The term $\mathbf{r}$ can be viewed as the *residual* or the *error* between the input basis vector $\mathbf{u}_\ell$ and its approximation $\mathbf{U}\mathbf{c}$ from interpolating the basis $\{\mathbf{u}_1, \ldots, \mathbf{u}_{\ell-1}\}$ at the indices $\wp_1, \ldots, \wp_{\ell-1}$ in line 4 of Algorithm 1. Hence, $\mathbf{r}_{\wp_i} = 0$ for $i = 1, \ldots, \ell - 1$. However, the linear independence of the input basis $\{\mathbf{u}_\ell\}_{\ell=1}^m$ guarantees that, in each iteration, $\mathbf{r}$ is a nonzero vector and hence $\rho = \mathbf{r}_{\wp_\ell}$ is also nonzero. Lemma 2.2.3 will demonstrate that $\rho \neq 0$ at each step implies that $\mathbf{P}^T\mathbf{U}$ is always nonsingular and hence the DEIM procedure is well-defined. This also implies that the interpolation indices $\{\wp_i\}_{i=1}^m$ are hierarchical and non-repeated.

Figure 2.1 illustrates the selection procedure in Algorithm 1 for DEIM interpolation indices. To summarize, the DEIM approximation is given formally as follows.

Definition 2.2.1 *Let* $\mathbf{f} : \mathscr{D} \mapsto \mathbb{R}^n$ *be a nonlinear vector-valued function with* $\mathscr{D} \subseteq \mathbb{R}^d$, *for some positive integer* d. *Let* $\{\mathbf{u}_\ell\}_{\ell=1}^m \subset \mathbb{R}^n$ *be a linearly independent set, for*

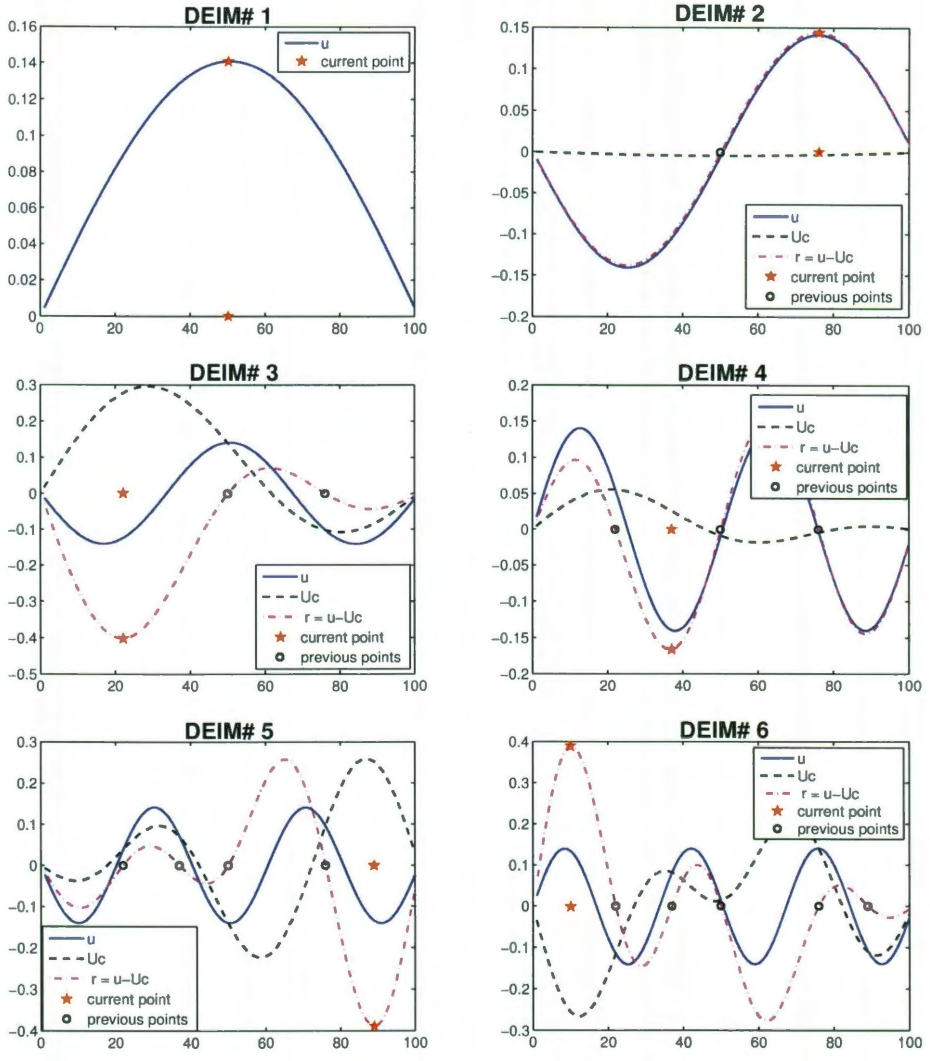


Figure 2.1: Illustration of the selection process of indices in Algorithm 1 for the DEIM approximation. The input basis vectors are the first 6 eigenvectors of the discrete Laplacian. From the plots, $\mathbf{u} = \mathbf{u}_\ell$, $\mathbf{Uc}$ and $\mathbf{r} = \mathbf{u}_\ell - \mathbf{Uc}$ are defined as in iteration ℓ of Algorithm 1.

$m \in \{1, \ldots, n\}$. For $\tau \in \mathscr{D}$, the DEIM approximation of order m for $\mathbf{f}(\tau)$ in the space spanned by $\{\mathbf{u}_\ell\}_{\ell=1}^m$ is given by

$$\hat{\mathbf{f}}(\tau) := \mathbb{P}\ \mathbf{f}(\tau), \qquad \mathbb{P} := \mathbf{U}(\mathbf{P}^T\mathbf{U})^{-1}\mathbf{P}^T, \tag{2.16}$$

where $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_m] \in \mathbb{R}^{n\times m}$ and $\mathbf{P} = [\mathbf{e}_{\wp_1}, \ldots, \mathbf{e}_{\wp_m}] \in \mathbb{R}^{n\times m}$ with $\{\wp_1, \ldots, \wp_m\}$ being the output from Algorithm 1 with the input basis $\{\mathbf{u}_i\}_{i=1}^m$.

Note that the matrix $\mathbf{U}$ used in the DEIM approximation (2.2.1) is not required to have orthonormal columns and also that $\mathbb{P} = \mathbb{P}^2$ and $\mathbb{P} \in \mathbb{R}^{n\times n}$ is an oblique projector onto Span$\{\mathbf{U}\}$. Clearly, $\hat{\mathbf{f}}$ in (2.16) is indeed an interpolation approximation for the original function $\mathbf{f}$, since $\hat{\mathbf{f}}$ is exact at the interpolation as verified with the simple calculation:

$$\mathbf{P}^T\hat{\mathbf{f}}(\tau) = \mathbf{P}^T\left(\mathbf{U}(\mathbf{P}^T\mathbf{U})^{-1}\mathbf{P}^T\mathbf{f}(\tau)\right) = (\mathbf{P}^T\mathbf{U})(\mathbf{P}^T\mathbf{U})^{-1}\mathbf{P}^T\mathbf{f}(\tau) = \mathbf{P}^T\mathbf{f}(\tau).$$

The DEIM approximation is uniquely determined by the projection basis $\{\mathbf{u}_i\}_{i=1}^m$. This basis not only specifies the projection subspace used in the approximation, but also determines the interpolation indices used for computing the coefficient of the approximation. Hence, the choice of projection basis can greatly affect the accuracy of the approximation in (2.16), as shown also in the error bound of the DEIM approximation (2.22) in the next section. As noted, POD introduced in §2.1.1 is an effective method for constructing this projection basis, since it provides an optimal global basis that captures the dynamics of the space generated from snapshots of the nonlinear function.

The selection of the interpolation points is basis dependent. However, once the set of DEIM interpolation indices $\{\wp_\ell\}_{\ell=1}^m$ is determined from $\{\mathbf{u}_i\}_{i=1}^m$, the DEIM approximation is independent of the choice of basis spanning the space Range($\mathbf{U}$). In particular, let $\{\mathbf{q}_\ell\}_{\ell=1}^m$ be any basis for Range($\mathbf{U}$). Then

$$\mathbf{U}(\mathbf{P}^T\mathbf{U})^{-1}\mathbf{P}^T\mathbf{f}(\tau) = \mathbf{Q}(\mathbf{P}^T\mathbf{Q})^{-1}\mathbf{P}^T\mathbf{f}(\tau), \tag{2.17}$$

where $\mathbf{Q} = [\mathbf{q}_1, \ldots, \mathbf{q}_m] \in \mathbb{R}^{n\times m}$. To verify (2.17), note that Range($\mathbf{U}$) = Range($\mathbf{Q}$) so that $\mathbf{U} = \mathbf{QR}$ for some nonsingular matrix $\mathbf{R} \in \mathbb{R}^{m\times m}$. This substitution gives

$$\mathbf{U}(\mathbf{P}^T\mathbf{U})^{-1}\mathbf{P}^T\mathbf{f}(\tau) = (\mathbf{QR})((\mathbf{P}^T\mathbf{Q})\mathbf{R})^{-1}\mathbf{P}^T\mathbf{f}(\tau) = \mathbf{Q}(\mathbf{P}^T\mathbf{Q})^{-1}\mathbf{P}^T\mathbf{f}(\tau).$$

The DEIM index selection procedure in Algorithm 1 can break down only in Step 4 when $\mathbf{P}^T\mathbf{U}$ is not invertible. It can be shown by induction that this will not be the case (i.e. $\mathbf{P}^T\mathbf{U}$ is non-singular for all iterations) as long as the input vectors $\{\mathbf{u}_\ell\}_{\ell=1}^m$ are linearly independent. Moreover, the inverse of $\mathbf{P}^T\mathbf{U}$ can be obtained recursively from the iterations in Algorithm 1.

Claim 2.2.2 *Let $\{\mathbf{u}_\ell\}_{\ell=1}^m \subset \mathbb{R}^n$ be a linearly independent set of input vectors to Algorithm 1 with output indices $\{\wp_\ell\}_{\ell=1}^m$. Define $\mathbf{M}_\ell := \mathbf{P}_\ell^T\mathbf{U}_\ell \in \mathbb{R}^{\ell\times\ell}$ for $\ell = 1, \ldots, m$ where $\mathbf{P}_\ell = [\mathbf{e}_{\wp_1}, \ldots, \mathbf{e}_{\wp_\ell}] \in \mathbb{R}^{n\times\ell}$, $\mathbf{U}_\ell = [\mathbf{u}_1, \ldots, \mathbf{u}_\ell] \in \mathbb{R}^{n\times\ell}$. Then $\mathbf{M}_\ell$ is nonsingular with $\mathbf{M}_1^{-1} = (\mathbf{p}_1^T\mathbf{u}_1)^{-1}$ and for $\ell = 2, \ldots, m$,*

$$\mathbf{M}_\ell^{-1} = \begin{bmatrix} \mathbf{I} & -\mathbf{c} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{M}_{\ell-1}^{-1} & \mathbf{0} \\ -\rho^{-1}\mathbf{a}^T\mathbf{M}_{\ell-1}^{-1} & \rho^{-1} \end{bmatrix}, \tag{2.18}$$

where $\mathbf{a}^T = \mathbf{p}_\ell^T \mathbf{U}_{\ell-1}$, $\mathbf{c} = \mathbf{M}_{\ell-1}^{-1} \mathbf{P}_{\ell-1}^T \mathbf{u}_\ell$, *and* $\rho = \mathbf{p}_\ell^T \mathbf{u}_\ell - \mathbf{a}^T \mathbf{c} = \mathbf{p}_\ell^T (\mathbf{u}_\ell - \mathbf{U}_{\ell-1} \mathbf{M}_{\ell-1}^{-1} \mathbf{P}_{\ell-1}^T \mathbf{u}_\ell)$ *and* $\mathbf{p}_\ell = \mathbf{e}_{\wp_\ell} \in \mathbb{R}^n$, *which can be obtained directly from Algorithm 1.*

Proof: At the initial step of Algorithm 1, $\mathbf{P}_1 = \mathbf{e}_{\wp_1}$ and $\mathbf{U}_1 = \mathbf{u}_1$. Since $\mathbf{u}_1$ is nonzero, $\mathbf{M}_1 = \mathbf{P}_1^T \mathbf{U}_1 = \mathbf{e}_{\wp_1}^T \mathbf{u}_1 \neq 0$ and $\mathbf{M}_1^{-1} = 1/\mathbf{e}_{\wp_1}^T \mathbf{u}_1$. To simplify notation, for $\ell = 2, \ldots, m$, let $\bar{\mathbf{M}} := \mathbf{M}_{\ell-1} = \bar{\mathbf{P}}^T \bar{\mathbf{U}}$ and $\mathbf{M} := \mathbf{M}_\ell = \mathbf{P}^T \mathbf{U}$ where

$$\begin{aligned} &\mathbf{U} = [\bar{\mathbf{U}} \;\; \mathbf{u}] \in \mathbb{R}^{n \times \ell}, \;\; \bar{\mathbf{U}} = [\mathbf{u}_1, \ldots, \mathbf{u}_{\ell-1}] \in \mathbb{R}^{n \times (\ell-1)}, \;\; \mathbf{u} = \mathbf{u}_\ell \in \mathbb{R}^n, \\ &\mathbf{P} = [\bar{\mathbf{P}} \;\; \mathbf{p}] \in \mathbb{R}^{n \times \ell}, \;\; \bar{\mathbf{P}} = [\mathbf{e}_{\wp_1}, \ldots, \mathbf{e}_{\wp_{\ell-1}}] \in \mathbb{R}^{n \times (\ell-1)}, \;\; \mathbf{p} = \mathbf{e}_{\wp_\ell} \in \mathbb{R}^n. \end{aligned} \tag{2.19}$$

For $\ell = 2$, $\bar{\mathbf{M}} = \mathbf{M}_1 = \mathbf{e}_{\wp_1}^T \mathbf{u}_1$ is invertible, as shown earlier. As an induction hypothesis, assume $\bar{\mathbf{M}} = \bar{\mathbf{P}}^T \bar{\mathbf{U}}$ is invertible for each iteration $\ell \geq 2$. Then, it can be shown that $\mathbf{M}$ used in Step 4 of iteration $\ell + 1$ is invertible as follows. First note that, $\mathbf{M} = \begin{bmatrix} \bar{\mathbf{M}} & \bar{\mathbf{P}}^T \mathbf{u} \\ \mathbf{p}^T \bar{\mathbf{U}} & \mathbf{p}^T \mathbf{u} \end{bmatrix}$ and $\mathbf{M}$ can be factored in the form:

$$\mathbf{M} = \begin{bmatrix} \bar{\mathbf{M}} & \bar{\mathbf{P}}^T \mathbf{u} \\ \mathbf{p}^T \bar{\mathbf{U}} & \mathbf{p}^T \mathbf{u} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{M}} & \mathbf{0} \\ \mathbf{a}^T & \rho \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{c} \\ \mathbf{0} & 1 \end{bmatrix}, \tag{2.20}$$

where $\mathbf{a}^T = \mathbf{p}^T \bar{\mathbf{U}}$, $\mathbf{c} = \bar{\mathbf{M}}^{-1} \bar{\mathbf{P}}^T \mathbf{u}$, and $\rho = \mathbf{p}^T \mathbf{u} - \mathbf{a}^T \mathbf{c} = \mathbf{p}^T (\mathbf{u} - \bar{\mathbf{U}} \bar{\mathbf{M}}^{-1} \bar{\mathbf{P}}^T \mathbf{u})$. Note $|\rho| = \|\mathbf{r}\|_\infty$ where $\mathbf{r}$ is defined at Step 5 of Algorithm 1. Since $\mathbf{u} = \mathbf{u}_\ell$ is not in the span of $\{\mathbf{u}_1, \ldots, \mathbf{u}_{\ell-1}\}$, i.e. $\mathbf{u} \neq \bar{\mathbf{U}} \bar{\mathbf{c}}$ for any $\bar{\mathbf{c}} \in \mathbb{R}^{\ell-1}$, then $\mathbf{r}$ is a nonzero vector, which implies $\rho = \mathbf{r}_{\wp_\ell} \neq 0$. Now, from (2.20), the inverse of $\mathbf{M}$ is given by

$$\mathbf{M}^{-1} = \begin{bmatrix} \mathbf{I} & -\mathbf{c} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \bar{\mathbf{M}}^{-1} & \mathbf{0} \\ -\rho^{-1} \mathbf{a}^T \bar{\mathbf{M}}^{-1} & \rho^{-1} \end{bmatrix}, \tag{2.21}$$

as given in (2.18), which is well-defined since $\rho \neq 0$ and $\bar{\mathbf{M}}$ is invertible by the inductive hypothesis. □

It will be shown next that the norm of $\mathbf{M}_\ell^{-1} = (\mathbf{P}_\ell^T \mathbf{U}_\ell)^{-1}$ from (2.18), for $\ell = 1, \ldots, m$, can be used to derive an error bound for the DEIM approximation.

2.2.2 Error Bound for DEIM

This section provides an error bound in the 2-norm for the DEIM approximation for a nonlinear vector-valued function. This derivation of the error bound provides motivation for the DEIM selection process in Algorithm 1 in terms of recursively limiting the local growth of a certain magnification factor of the best 2-norm approximation error. As before, $\|\cdot\|$ will denote 2-norm. This error bound is given formally as follows.

Lemma 2.2.3 *Let $\mathbf{f} \in \mathbb{R}^n$ be an arbitrary vector. Let $\{\mathbf{u}_\ell\}_{\ell=1}^m \subset \mathbb{R}^n$ be a given orthonormal set of vectors. From Definition 2.2.1, the DEIM approximation of order $m \leq n$ for $\mathbf{f}$ in the space spanned by $\{\mathbf{u}_\ell\}_{\ell=1}^m$ is $\hat{\mathbf{f}} = \mathbb{P}\,\mathbf{f}$, where $\mathbb{P} = \mathbf{U}(\mathbf{P}^T\mathbf{U})^{-1}\mathbf{P}^T$, $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_m] \in \mathbb{R}^{n\times m}$, $\mathbf{P} = [\mathbf{e}_{\wp_1}, \ldots, \mathbf{e}_{\wp_m}] \in \mathbb{R}^{n\times m}$, and $\{\wp_1, \ldots, \wp_m\}$ being the output from Algorithm 1 with the input basis $\{\mathbf{u}_i\}_{i=1}^m$. Then,*

$$\mathbf{f} - \hat{\mathbf{f}} \;=\; (\mathbf{I} - \mathbb{P})\mathbf{w} \quad \textit{and} \quad \|\mathbf{f} - \hat{\mathbf{f}}\| \leq \mathbf{C}_m\, \mathscr{E}_*(\mathbf{f}), \tag{2.22}$$

where $\mathbf{w} := (\mathbf{I} - \mathbf{U}\mathbf{U}^T)\mathbf{f}$,

$$\mathbf{C}_m := \|(\mathbf{P}^T\mathbf{U})^{-1}\| \;\text{ and }\; \mathscr{E}_*(\mathbf{f}) = \|(\mathbf{I} - \mathbf{U}\mathbf{U}^T)\mathbf{f}\|. \tag{2.23}$$

$\mathscr{E}_*(\mathbf{f})$ is the error of the best 2-norm approximation for $\mathbf{f}$ *from the space* Range($\mathbf{U}$) *and the constant* $\mathbf{C}_m$ *is bounded by*

$$\mathbf{C}_m \leq (1+\sqrt{2n})^{m-1}\|\mathbf{u}_1\|_\infty^{-1}. \tag{2.24}$$

Proof: Consider the DEIM approximation $\hat{\mathbf{f}}$ given by (2.15). We wish to determine a bound for the error $\|\mathbf{f}-\hat{\mathbf{f}}\|$ in terms of the *optimal* 2-norm (least-squares) approximation for $\mathbf{f}$ from Range($\mathbf{U}$). This best approximation is given by

$$\mathbf{f}_* = \mathbf{U}\mathbf{U}^T\mathbf{f}, \tag{2.25}$$

which minimizes the error $\|\mathbf{f}-\hat{\mathbf{f}}\|$ over Range($\mathbf{U}$). Consider

$$\mathbf{f} = (\mathbf{f}-\mathbf{f}_*)+\mathbf{f}_* = \mathbf{w}+\mathbf{f}_*, \tag{2.26}$$

where $\mathbf{w} = \mathbf{f}-\mathbf{f}_* = (\mathbf{I}-\mathbf{U}\mathbf{U}^T)\mathbf{f}$. From (2.26) and $\mathbb{P}\mathbf{f}_* = \mathbf{f}_*$,

$$\hat{\mathbf{f}} = \mathbb{P}\mathbf{f} = \mathbb{P}(\mathbf{w}+\mathbf{f}_*) = \mathbb{P}\mathbf{w}+\mathbb{P}\mathbf{f}_* = \mathbb{P}\mathbf{w}+\mathbf{f}_*. \tag{2.27}$$

Equations (2.26) and (2.27) imply $\mathbf{f}-\hat{\mathbf{f}} = (\mathbf{I}-\mathbb{P})\mathbf{w}$ and

$$\|\mathbf{f}-\hat{\mathbf{f}}\| = \|(\mathbf{I}-\mathbb{P})\mathbf{w}\| \leq \|\mathbf{I}-\mathbb{P}\|\|\mathbf{w}\|. \tag{2.28}$$

Note that

$$\|\mathbf{I}-\mathbb{P}\| \;=\; \|\mathbb{P}\| = \|\mathbf{U}(\mathbf{P}^T\mathbf{U})^{-1}\mathbf{P}^T\| = \|(\mathbf{P}^T\mathbf{U})^{-1}\|. \tag{2.29}$$

The first equality in (2.29) follows from the fact that $\|\mathbf{I}-\mathbb{P}\| = \|\mathbb{P}\|$, for any projector $\mathbb{P} \neq \mathbf{0}$ or $\mathbf{I}$ (see [85]).

Note that $\mathscr{E}_*(\mathbf{f}) := \|\mathbf{w}\|$ is the minimum 2-norm error in the least-squares sense for $\mathbf{f}_*$ defined in (2.25). From (2.29), the bound for the error in (2.28) becomes

$$\|\mathbf{f} - \hat{\mathbf{f}}\| \leq \|(\mathbf{P}^T\mathbf{U})^{-1}\| \ \mathscr{E}_*(\mathbf{f}), \tag{2.30}$$

which establishes the error bound (2.22). The magnification factor $\|(\mathbf{P}^T\mathbf{U})^{-1}\|$ depends on the DEIM selection of indices $\wp_1, \ldots, \wp_m$ through the matrix $\mathbf{P}$. It will be shown that each iteration of the DEIM algorithm aims to select an index to limit stepwise growth of $\|(\mathbf{P}^T\mathbf{U})^{-1}\|$ and hence to limit size of the bound for the error $\|\mathbf{f} - \hat{\mathbf{f}}\|$.

The recursive formula for $(\mathbf{P}^T\mathbf{U})^{-1}$ in Claim 2.2.2 will be considered and the notation defined in (2.19) will be used here. That is, let $\bar{\mathbf{M}} = \bar{\mathbf{P}}^T\bar{\mathbf{U}}$ and $\mathbf{M} = \mathbf{P}^T\mathbf{U}$. From Claim 2.2.2, at the initial step of Algorithm 1, $\mathbf{M} = \mathbf{P}^T\mathbf{U} = \mathbf{e}_{\wp_1}^T\mathbf{u}_1$ and hence $\|\mathbf{M}^{-1}\| = \frac{1}{|\mathbf{e}_{\wp_1}^T\mathbf{u}_1|} = \|\mathbf{u}_1\|_\infty^{-1} \geq 1$. That is, for $m = 1$, $\mathbf{C}_m = \|\mathbf{M}^{-1}\| = \|\mathbf{u}_1\|_\infty^{-1}$. Note that the choice of the first interpolation index $\wp_1$ minimizes the matrix norm $\|\mathbf{M}^{-1}\|$ and hence minimizes the error bound (2.22). Now consider a general step $\ell \geq 2$ with matrices defined in (2.19). When $\mathbf{M}$ is written in the form (2.20), from (2.21),

$$\mathbf{M}^{-1} = \begin{bmatrix} \mathbf{I} & -\mathbf{c} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \bar{\mathbf{M}}^{-1} & \mathbf{0} \\ -\rho^{-1}\mathbf{a}^T\bar{\mathbf{M}}^{-1} & \rho^{-1} \end{bmatrix} \tag{2.31}$$

$$= \begin{bmatrix} \mathbf{I} & -\mathbf{c} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\rho^{-1}\mathbf{a}^T & \rho^{-1} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{M}}^{-1} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \tag{2.32}$$

$$= \left\{ \begin{bmatrix} \mathbf{I} & 0 \\ \mathbf{0} & 0 \end{bmatrix} + \rho^{-1} \begin{bmatrix} \mathbf{c} \\ -1 \end{bmatrix} [\mathbf{a}^T, -1] \right\} \begin{bmatrix} \bar{\mathbf{M}}^{-1} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{2.33}$$

A bound for the 2-norm of $\mathbf{M}^{-1}$ is then given by

$$\|\mathbf{M}^{-1}\| \leq \left\{ \left\| \begin{bmatrix} \mathbf{I} & 0 \\ \mathbf{0} & 0 \end{bmatrix} \right\| + |\rho|^{-1} \left\| \begin{bmatrix} \mathbf{c} \\ -1 \end{bmatrix} \left[\mathbf{a}^T, -1\right] \right\| \right\} \left\| \begin{bmatrix} \bar{\mathbf{M}}^{-1} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \right\|. \quad (2.34)$$

Now, observe that

$$\left\| \begin{bmatrix} \mathbf{c} \\ -1 \end{bmatrix} \left[\mathbf{a}^T, -1\right] \right\| = \left\| [\bar{\mathbf{U}}, \mathbf{u}] \begin{bmatrix} \mathbf{c} \\ -1 \end{bmatrix} \left[\mathbf{a}^T, -1\right] \right\| \quad (2.35)$$

$$\leq \left\| \bar{\mathbf{U}}\mathbf{c} - \mathbf{u} \right\| \left\| \left[\mathbf{a}^T, -1\right] \right\| \quad (2.36)$$

$$\leq \sqrt{1 + \|\mathbf{a}\|^2}\sqrt{n} \left\| \bar{\mathbf{U}}\mathbf{c} - \mathbf{u} \right\|_\infty \leq \sqrt{2n}|\rho|. \quad (2.37)$$

Substituting this into (2.34) gives

$$\|\mathbf{M}^{-1}\| \leq [1 + \sqrt{2n}]\|\bar{\mathbf{M}}^{-1}\| \leq (1 + \sqrt{2n})^{m-1}\|\mathbf{u}_1\|_\infty^{-1}, \quad (2.38)$$

with the last inequality obtained by recursively applying this stepwise bound over the m steps. □

Since the DEIM procedure selects the index $\wp_\ell$ that *maximizes* $|\rho|$, it *minimizes* the reciprocal $\frac{1}{|\rho|}$, which controls the increment in the bound of $\|\mathbf{M}^{-1}\|$ at iteration ℓ, as shown in (2.34). Therefore, the selection process for the interpolation index in each iteration of DEIM (line 6 of Algorithm 1) can be explained in terms of limiting growth of the error bound of the approximation $\hat{\mathbf{f}}$. This error bound from Lemma 2.2.3 applies to any nonlinear vector-valued function $\mathbf{f}(\tau)$ approximated by DEIM. However, the bound in (2.24) is not useful as an *a priori* estimate since it is very pessimistic and grows far more rapidly than the actual observed values of $\|(\mathbf{P}^T\mathbf{U})^{-1}\|$. In practice,

we just compute this norm (the matrix is typically small) and use it to obtain an *a posteriori* estimate.

For a given dimension m of the DEIM approximation, the constant C does not depend on $\mathbf{f}$ and hence it applies to the approximation $\hat{\mathbf{f}}(\tau)$ of $\mathbf{f}(\tau)$ from Definition 2.2.1 for any $\tau \in \mathscr{D}$. However, the best approximation error

$$\mathcal{E}_* = \mathcal{E}_*(\mathbf{f}(\tau))$$

is dependent upon $\mathbf{f}(\tau)$ and changes with each new value of τ. This would be quite expensive to compute, so an easily computable estimate is highly desirable. A reasonable estimate is available with the SVD of the nonlinear snapshot matrix

$$\widehat{\mathbf{F}} = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_{n_s}],$$

$\mathbf{f}_i = \mathbf{f}(\tau_i)$, $i = 1, \ldots, n_s$. Let $\mathcal{F} = \text{Range}(\widehat{\mathbf{F}})$ and let $\widehat{\mathbf{F}} = \widehat{\mathbf{U}}\widehat{\mathbf{\Sigma}}\widehat{\mathbf{W}}^T$ be its SVD, where $\widehat{\mathbf{U}} = [\mathbf{U}, \widetilde{\mathbf{U}}]$ and $\mathbf{U}$ represents the leading m columns of the orthogonal matrix $\widehat{\mathbf{U}}$. Partition $\widehat{\Sigma} = \begin{bmatrix} \Sigma & \mathbf{0} \\ \mathbf{0} & \widetilde{\Sigma} \end{bmatrix}$ to conform with the partitioning of $\widehat{\mathbf{U}}$. The singular values are ordered as usual with $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_m \geq \sigma_{m+1} \geq \cdots \geq \sigma_n \geq 0$. The diagonal matrix Σ has the leading m singular values on its diagonal. The orthogonal matrix $\widehat{\mathbf{W}} = [\mathbf{W}, \widetilde{\mathbf{W}}]$ is partitioned accordingly. Any vector $\mathbf{f} \in \mathcal{F}$ may be written in the form

$$\mathbf{f} = \widehat{\mathbf{F}}\hat{\mathbf{g}} = \mathbf{U}\Sigma\mathbf{g} + \widetilde{\mathbf{U}}\widetilde{\Sigma}\tilde{\mathbf{g}},$$

where $\mathbf{g} = \mathbf{W}^T\hat{\mathbf{g}}$ and $\tilde{\mathbf{g}} = \widetilde{\mathbf{W}}^T\hat{\mathbf{g}}$. Thus

$$\|\mathbf{f} - \mathbf{f}_*\| = \|(\mathbf{I} - \mathbf{U}\mathbf{U}^T)\mathbf{f}\| = \|\widetilde{\mathbf{U}}\widetilde{\Sigma}\tilde{\mathbf{g}}\| \leq \sigma_{m+1}\|\tilde{\mathbf{g}}\|.$$

For vectors $\mathbf{f}$ nearly in $\mathcal{F}$, we have $\mathbf{f} = \widehat{\mathbf{F}}\hat{\mathbf{g}} + \mathbf{w}$ with $\mathbf{w}^T\widehat{\mathbf{F}}\hat{\mathbf{g}} = 0$, and thus

$$\mathcal{E}_* = \mathcal{E}_*(\mathbf{f}) \approx \sigma_{m+1} \tag{2.39}$$

is a reasonable approximation so long as $\|\mathbf{w}\|$ is small ($\|\mathbf{w}\|_2 = \mathcal{O}(\sigma_{m+1})$ ideally). The POD approach (and hence the resulting DEIM approach) is most successful when the trajectories are attracted to a low dimensional subspace (or manifold). Hence, the vectors $\mathbf{f}(\tau)$ should nearly lie in $\mathcal{F}$ and this approximation will then serve for all of them.

To illustrate the error bound for DEIM approximation, the numerical results will be presented next for nonlinear parametrized functions defined on 1-D and 2-D discrete spatial points. These experiments show that the approximate error bound using σ_{m+1} in place of $\mathcal{E}_*$ is quite reasonable in practice.

2.2.3 Numerical Examples of the DEIM Error Bound

This section demonstrates the accuracy and efficiency of the approximation from DEIM as well as its error bound given in §2.2.2. The examples here use the POD basis in the DEIM approximation. The POD basis is constructed from a set of *snapshots* corresponding to a selected set of elements in $\mathscr{D}$. In particular, define

$$\mathscr{D}^s = \{\mu_1^s, \ldots, \mu_{n_s}^s\} \subset \mathscr{D} \tag{2.40}$$

to be a parameter set for constructing a snapshot matrix $[\mathbf{f}(\mu_1^s), \ldots, \mathbf{f}(\mu_{n_s}^s)]$, which is used for computing the POD basis $\{\mathbf{u}_\ell\}_{\ell=1}^m$ for the DEIM approximation.

To evaluate the accuracy, the DEIM approximation $\hat{\mathbf{f}}$ in (2.16) will be applied to the function at the parameters in the set

$$\bar{\mathscr{D}} = \{\bar{\mu}_1, \ldots, \bar{\mu}_{\bar{n}}\} \subset \mathscr{D}, \tag{2.41}$$

which is different from and larger than the set $\mathscr{D}^s$ used for the snapshots. Then the average error for *DEIM approximation* $\hat{\mathbf{f}}$ will be considered over the elements in $\bar{\mathscr{D}}$, which is given by

$$\bar{\mathcal{E}}(\mathbf{f}) = \frac{1}{\bar{n}} \sum_{i=1}^{\bar{n}} \|\mathbf{f}(\bar{\mu}_i) - \hat{\mathbf{f}}(\bar{\mu}_i)\|_2. \tag{2.42}$$

The average POD error in (2.23) for *POD approximation* $\hat{\mathbf{f}}_*$ from (2.25) over the elements in $\bar{\mathscr{D}}$ is given by

$$\bar{\mathcal{E}}_*(\mathbf{f}) = \frac{1}{\bar{n}} \sum_{i=1}^{\bar{n}} \|\mathbf{f}(\bar{\mu}_i) - \hat{\mathbf{f}}_*(\bar{\mu}_i)\|_2 = \frac{1}{\bar{n}} \sum_{i=1}^{\bar{n}} \mathscr{E}_*(\mathbf{f}(\bar{\mu}_i)). \tag{2.43}$$

From Lemma 2.2.3, the *average error bound* is then given by

$$\bar{\mathcal{E}}(\mathbf{f}) \leq C\bar{\mathcal{E}}_*(\mathbf{f}), \tag{2.44}$$

with the corresponding approximation using (2.39):

$$\bar{\mathcal{E}}(\mathbf{f}) \lesssim C\sigma_{m+1}. \tag{2.45}$$

This estimate is purely heuristic. Although there is little hope for validating this heuristic in general, it does seem to provide a reasonable qualitative estimate of the expected error, as shown next in the following examples.

2.2.3.1 A nonlinear parametrized function with spatial points in 1-D

Consider a nonlinear parametrized function $s : \Omega \times \mathscr{D} \mapsto \mathbb{R}$ defined by

$$s(x;\mu) = (1-x)\cos(3\pi\mu(x+1))e^{-(1+x)\mu}, \tag{2.46}$$

where $x \in \Omega = [-1,1]$ and $\mu \in \mathscr{D} = [1,\pi]$. This nonlinear function is from an example in [61]. Let $\mathbf{x} = [x_1,\ldots,x_n]^T \in \mathbb{R}^n$, with x_i equidistantly spaced points in Ω, for $i = 1,\ldots,n$, $n = 100$. Define $\mathbf{f} : \mathscr{D} \mapsto \mathbb{R}^n$ by

$$\mathbf{f}(\mu) = [s(x_1;\mu),\ldots,s(x_n;\mu)]^T \in \mathbb{R}^n, \tag{2.47}$$

for $\mu \in \mathscr{D}$. This example uses 51 snapshots $\mathbf{f}(\mu_j^s)$ to construct POD basis $\{\mathbf{u}_\ell\}_{\ell=1}^m$ with $\mu_1^s,\ldots,\mu_{51}^s$ selected as equally spaced points in $[1,\pi]$. Figure 2.2 shows the singular values of these snapshots and the corresponding first 6 POD basis vectors with the first 6 spatial points selected from the DEIM algorithm using this POD basis as an input. Figure 2.3 compares the approximate functions from DEIM of dimension 10 with the original function of dimension 100 at different values of $\mu \in \mathscr{D}$. This demonstrates that DEIM gives a good approximation at arbitrary values $\mu \in \mathscr{D}$. Figure 2.4 illustrates the average errors defined in (2.42) and (2.43), with the average error bound and its approximation computed from the right hand side of (2.44) and (2.45), respectively, with $\bar{\mu}_1,\ldots,\bar{\mu}_{\bar{n}} \in \bar{\mathscr{D}}$ selected uniformly over $\mathscr{D}$ and $\bar{n} = 101$.

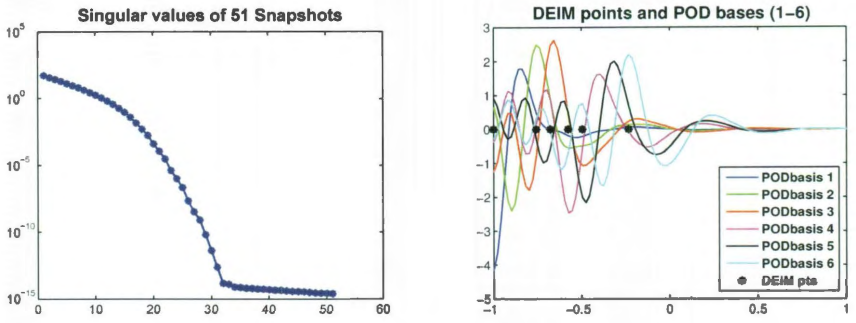


Figure 2.2: Singular values and the corresponding first 6 POD basis vectors with DEIM points of snapshots from (2.47).

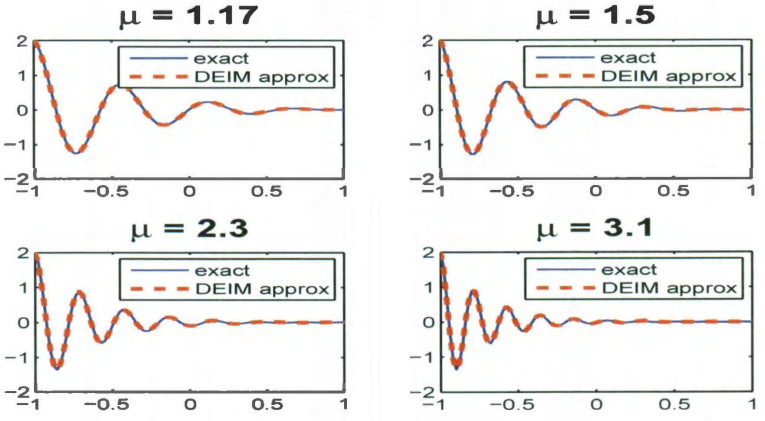


Figure 2.3: The approximate functions from DEIM of dimension 10 compared with the original functions (2.47) of dimension $n = 100$ at $\mu = 1.17, 1.5, 2.3, 3.1$.

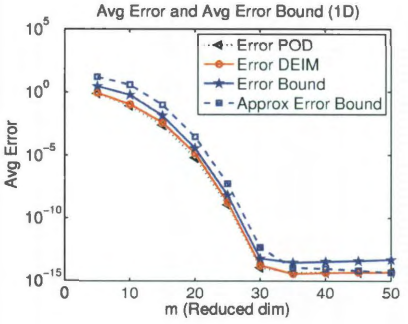


Figure 2.4: Compare average errors of POD and DEIM approximations for (2.47) with the average error bounds and their approximations given in (2.44) and (2.45), respectively.

2.2.3.2 A nonlinear parametrized function with spatial points in 2-D

Consider a nonlinear parametrized function $s : \Omega \times \mathscr{D} \mapsto \mathbb{R}$ defined by

$$s(x, y; \mu) = \frac{1}{\sqrt{(x - \mu_1)^2 + (y - \mu_2)^2 + 0.1^2}}, \tag{2.48}$$

where $(x, y) \in \Omega = [0.1, 0.9]^2 \subset \mathbb{R}^2$ and $\mu = (\mu_1, \mu_2) \in \mathscr{D} = [-1, -0.01]^2 \subset \mathbb{R}^2$. This example is modified from the one given in [38]. Let (x_i, y_j) be uniform grid points in Ω, for $i = 1, \ldots, n_x$ and $j = 1, \ldots, n_y$. Define $\mathbf{s} : \mathscr{D} \mapsto \mathbb{R}^{n_x \times n_y}$ by

$$\mathbf{s}(\mu) = [s(x_i, y_j; \mu)] \in \mathbb{R}^{n_x \times n_y} \tag{2.49}$$

for $\mu \in \mathscr{D}$ and $i = 1, \ldots, n_x$, and $j = 1, \ldots, n_y$. In this example, the full dimension is $n = n_x n_y = 400$ ($n_x = n_y = 20$). Note that a corresponding vector-valued function $\mathbf{f} : \mathscr{D} \mapsto \mathbb{R}^n$ for this problem can be defined by reshaping the matrix $\mathbf{s}(\mu)$ to a vector of length $n = n_x n_y$. The 225 snapshots constructed from uniformly selected parameters $\mu^s = (\mu_1^s, \mu_2^s)$ in the parameter domain $\mathscr{D}$ are used for constructing the POD basis. A different set of 625 pairs of parameters μ are used for testing (error

and CPU time). Figure 2.5 shows the singular values of these snapshots and the corresponding first 6 POD basis vectors. Figure 2.6 illustrates the distribution of the first 20 spatial points selected from the DEIM algorithm using this POD basis as an input. Notice that most of the selected points cluster close to the origin, where the function s increases sharply. Figure 2.7 shows that the approximate functions from DEIM of dimension 6 can reproduce the original function of dimension 400 very well at arbitrarily selected value $\mu \in \mathscr{D}$. Figure 2.8 gives the average errors with the bounds from the last section and the corresponding average CPU times for different dimensions of POD and DEIM approximations. The average errors of POD and DEIM approximations are computed from (2.42) and (2.43), respectively. The average error bounds and their approximations are computed from the right hand side of (2.44) and (2.45), respectively. This example uses $\bar{\mu}_1, \ldots, \bar{\mu}_{\bar{n}} \in \bar{\mathscr{D}}$ selected uniformly over $\mathscr{D}$ and $\bar{n} = 625$. The CPU times are averaged over the same set $\bar{\mathscr{D}}$.

2.2.4 Application of DEIM to Nonlinear Discretized Systems

The DEIM approximation (2.15) developed in the previous section may now be used to approximate the nonlinear term in (2.10) and the Jacobian in (2.11) with nonlinear approximations having computational complexity proportional to the number of reduced variables obtained with POD.

In the case of nonlinear time dependent PDEs in (2.15), by setting $\tau = t$ and $\mathbf{f}(t) = \mathbf{F}(\mathbf{V}\widehat{\mathbf{y}}(t))$, the nonlinear function in (2.5) approximated by DEIM can be

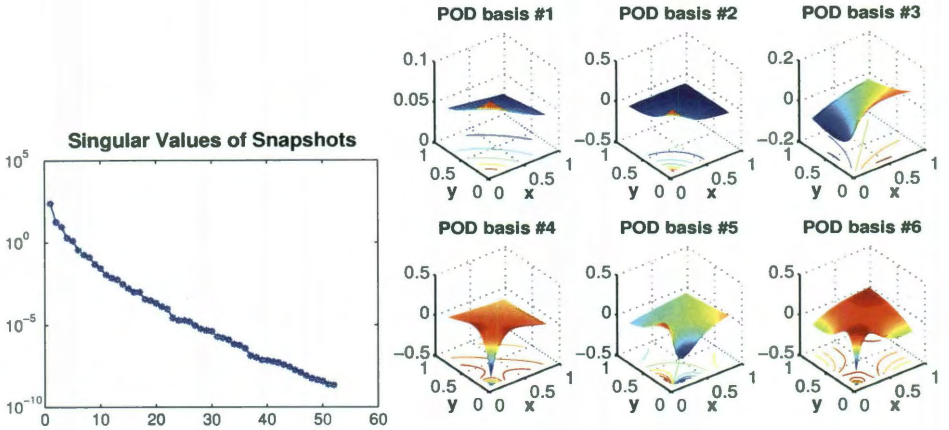


Figure 2.5: Singular values and the first 6 corresponding POD basis vectors of the snapshots of the nonlinear function (2.49).

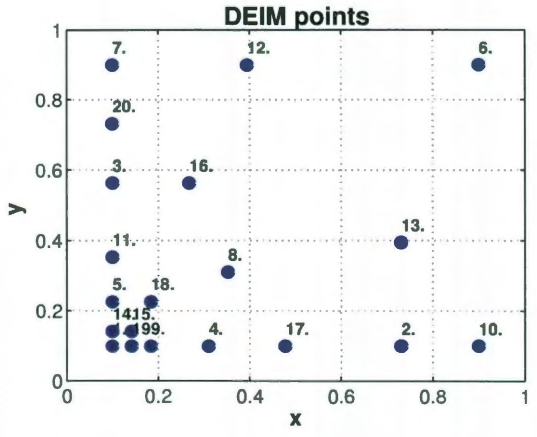


Figure 2.6: First 20 points selected by DEIM for the nonlinear function (2.49).

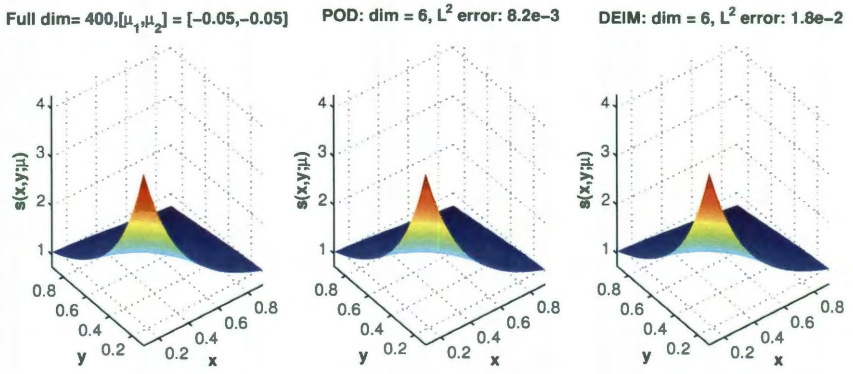


Figure 2.7: Compare the original nonlinear function (2.49) of dimension 400 with the POD and DEIM approximations of dimension 6 at parameter $\mu = (-0.05, -0.05)$.

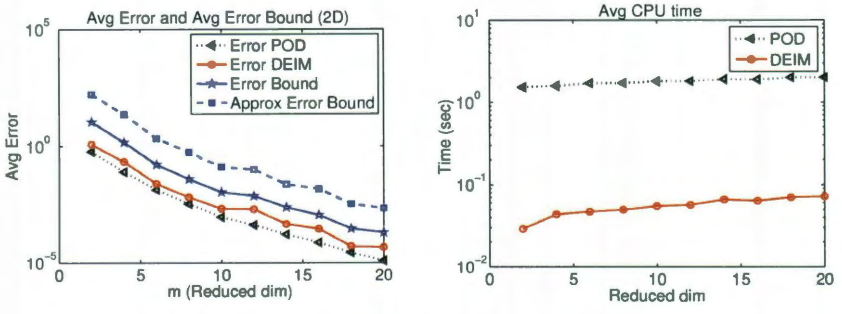


Figure 2.8: Left: Average errors of POD and DEIM approximations for (2.49) with the average error bounds given in (2.44) and their approximations given in (2.45). Right: Average CPU time for evaluating the POD and DEIM approximations.

written as

$$\mathbf{F}(\mathbf{V}\widehat{\mathbf{y}}(t)) \approx \mathbf{U}(\mathbf{P}^T\mathbf{U})^{-1}\mathbf{P}^T\mathbf{F}(\mathbf{V}\widehat{\mathbf{y}}(t)) \tag{2.50}$$

$$= \mathbf{U}(\mathbf{P}^T\mathbf{U})^{-1}\mathbf{F}(\mathbf{P}^T\mathbf{V}\widehat{\mathbf{y}}(t)). \tag{2.51}$$

The last equality in (2.51) follows from the fact that the function $\mathbf{F}$ evaluates componentwise at its input vector. The nonlinear term in (2.10) can thus be approximated by

$$\widehat{\mathbf{N}}(\widehat{\mathbf{y}}) \approx \underbrace{\mathbf{V}^T\mathbf{U}(\mathbf{P}^T\mathbf{U})^{-1}}_{\text{precomputed:}k\times m}\underbrace{\mathbf{F}(\mathbf{P}^T\mathbf{V}\widehat{\mathbf{y}}(t))}_{m\times 1}. \tag{2.52}$$

Note that the term $\mathbf{V}^T\mathbf{U}(\mathbf{P}^T\mathbf{U})^{-1}$ in (2.52) does not depend on t and therefore it can be precomputed before solving the system of ODEs. Note also that $\mathbf{P}^T\mathbf{V}\widehat{\mathbf{y}}(t) \in \mathbb{R}^m$ in (2.52) can be obtained by extracting the rows $\wp_1, \ldots, \wp_m$ of $\mathbf{V}$ and then multiplying against $\widehat{\mathbf{y}}$, which requires $2mk$ operations. Therefore, if $\alpha(m)$ denotes the cost of evaluating m components of $\mathbf{F}$, the complexity for computing this approximation of the nonlinear term roughly becomes $\mathcal{O}(\alpha(m) + 4km)$, which is independent of dimension n of the full-order system (2.1).

Similarly, in the case of steady parametrized nonlinear PDEs, from (2.15), set $\tau = \mu$ and $\mathbf{f}(\mu) = \mathbf{F}(\mathbf{V}\widehat{\mathbf{y}}(\mu))$. Then the nonlinear function in (2.6) approximated by DEIM can be written as

$$\mathbf{F}(\mathbf{V}\widehat{\mathbf{y}}(\mu)) \approx \mathbf{U}(\mathbf{P}^T\mathbf{U})^{-1}\mathbf{F}(\mathbf{P}^T\mathbf{V}\widehat{\mathbf{y}}(\mu)), \tag{2.53}$$

and the approximation for the Jacobian of the nonlinear term (2.11) is of the form

$$\widehat{\mathbf{J}}_{\mathbf{F}}(\widehat{\mathbf{y}}(\mu)) \approx \underbrace{\mathbf{V}^T\mathbf{U}(\mathbf{P}^T\mathbf{U})^{-1}}_{\text{precomputed:}k\times m}\underbrace{\mathbf{J}_{\mathbf{F}}(\mathbf{P}^T\mathbf{V}\widehat{\mathbf{y}}(\mu))}_{m\times m}\underbrace{\mathbf{P}^T\mathbf{V}}_{m\times k}, \tag{2.54}$$

where

$$\mathbf{J}_{\mathbf{F}}(\mathbf{P}^T\mathbf{V}\widehat{\mathbf{y}}(\mu)) = \mathbf{J}_{\mathbf{F}}(\mathbf{y}^r(\mu)) = \mathrm{diag}\{F'(\mathbf{y}^r_1(\mu)),\ldots,F'(\mathbf{y}^r_m(\mu))\},$$

and $\mathbf{y}^r(\mu) = \mathbf{P}^T\mathbf{V}\widehat{\mathbf{y}}(\mu)$, which can be computed with complexity independent of n as noted earlier. Therefore, the computational complexity for the approximation in (2.54) is roughly $\mathcal{O}(\alpha(m) + 2mk + 2\gamma mk + 2mk^2)$, where γ is the average number of nonzero entries per row of the Jacobian.

The approximations from DEIM are now in the form of (2.52) and (2.54) that recover the computational efficiency of (2.10) and (2.11), respectively.

Note that the nonlinear approximation from DEIM in (2.51) and (2.53) are obtained by exploiting the special structure of the nonlinear function $\mathbf{F}$ being evaluated componentwise at $\mathbf{y}$. The next section provides a completely general scheme.

2.2.5 Interpolation of General Nonlinear Functions

The very simple case of componentwise function $\mathbf{F}(\mathbf{y}) = [F(\mathbf{y}_1),\ldots,F(\mathbf{y}_n)]^T$, has been discussed for purposes of illustration and is indeed important in its own right. However, DEIM extends easily to general nonlinear functions. MATLAB notation is used here to explain this generalization.

$$[\mathbf{F}(\mathbf{y})]_i = \mathbf{F}_i(\mathbf{y}) = F_i(\mathbf{y}_{j_1^i},\mathbf{y}_{j_2^i},\mathbf{y}_{j_3^i},\ldots,\mathbf{y}_{j_{n_i}^i}) = F_i(\mathbf{y}(\mathbf{j}_i)), \tag{2.55}$$

where $F_i : \mathcal{Y}_i \to \mathbb{R}$, $\mathcal{Y}_i \subset \mathbb{R}^{n_i}$ and the integer vector $\mathbf{j}_i = [j_1^i, j_2^i, j_3^i, \ldots, j_{n_i}^i]^T$ denotes the indices of the subset of components of $\mathbf{y}$ required to evaluate the i-th component of $\mathbf{F}(\mathbf{y})$ for $i = 1, \ldots, n$.

The nonlinear function of the reduced-order system obtained from the POD-Galerkin method by projecting on the space spanned by columns of $\mathbf{V} \in \mathbb{R}^{n \times k}$ is in the form of $\mathbf{F}(\mathbf{V}\widehat{\mathbf{y}})$, where the components of $\widehat{\mathbf{y}} \in \mathbb{R}^k$ are the reduced variables. Recall that the DEIM approximation of order m for $\mathbf{F}(\mathbf{V}\widehat{\mathbf{y}})$ is given by

$$\mathbf{F}(\mathbf{V}\widehat{\mathbf{y}}) \approx \underbrace{\mathbf{U}(\mathbf{P}^T\mathbf{U})^{-1}}_{k \times m} \underbrace{\mathbf{P}^T\mathbf{F}(\mathbf{V}\widehat{\mathbf{y}})}_{m \times 1}, \tag{2.56}$$

where $\mathbf{U} \in \mathbb{R}^{n \times m}$ is the projection matrix for the nonlinear function $\mathbf{F}$, $\mathbf{P} = [\mathbf{e}_{\wp_1}, \ldots, \mathbf{e}_{\wp_m}] \in \mathbb{R}^{n \times m}$, and $\wp_1, \ldots, \wp_m$ are interpolation indices from the DEIM point selection algorithm. In the simple case when $\mathbf{F}$ is evaluated componentwise at $\mathbf{y}$, we have $\mathbf{P}^T\mathbf{F}(\mathbf{V}\widehat{\mathbf{y}}) = \mathbf{F}(\mathbf{P}^T\mathbf{V}\widehat{\mathbf{y}})$ where $\mathbf{P}^T\mathbf{V}$ can be obtained by extracting rows of $\mathbf{V}$ corresponding to $\wp_1, \ldots, \wp_m$ and hence its computational complexity is independent of n. However, this is clearly not applicable to the general nonlinear vector-valued function.

An efficient method for computing $\mathbf{P}^T\mathbf{F}(\mathbf{V}\widehat{\mathbf{y}})$ in the DEIM approximation (2.56) of a general nonlinear function is possible by using a certain sparse matrix data structure. Notice that, since $\mathbf{y}_j \approx \mathbf{V}(j,:)\widehat{\mathbf{y}}$, an approximation to $\mathbf{F}(\mathbf{y})$ is provided by

$$\mathbf{F}(\mathbf{V}\widehat{\mathbf{y}}) = [F_1(\mathbf{V}(\mathbf{j}_1,:)\widehat{\mathbf{y}}), \ldots, F_n(\mathbf{V}(\mathbf{j}_n,:)\widehat{\mathbf{y}})]^T \in \mathbb{R}^n, \tag{2.57}$$

and thus

$$\mathbf{P}^T\mathbf{F}(\mathbf{V}\widehat{\mathbf{y}}) = [F_{\wp_1}(\mathbf{V}(\mathbf{j}_{\wp_1},:)\widehat{\mathbf{y}}),\ldots,F_{\wp_m}(\mathbf{V}(\mathbf{j}_{\wp_m},:)\widehat{\mathbf{y}})]^T \in \mathbb{R}^m. \tag{2.58}$$

The complexity for evaluating each component $\wp_i$, $i = 1,\ldots,m$, of (2.58):

$$\tilde{F}_{\wp_i}(\widehat{\mathbf{y}}) := F_{\wp_i}(\mathbf{V}(\mathbf{j}_{\wp_i},:)\widehat{\mathbf{y}}) \tag{2.59}$$

is $n_{\wp_i} \times k$ Flops plus the complexity of evaluating the nonlinear scalar valued function $F_{\wp_i}$ of the $n_{\wp_i}$ variables indexed by $\mathbf{j}_{\wp_i}$.

The sparse evaluation procedure may be implemented using a *compressed sparse row* data structure as used in sparse matrix factorizations. Two linear integer arrays are needed: *irstart* is a vector of length $m+1$ containing pointers to locations in the vector *jrow*, which is of length $n_{\vec{\wp}} = \sum_{i=1}^{m} n_{\wp_i}$. The successive n_i entries of $jrow(irstart(i))$ indicate the dependence of the i component of $\mathbf{F}(\mathbf{y})$ on the selected variables from $\mathbf{y}$. In particular,

- *irstart*(i) contains location of the start of the i-th row with $irstart(m+1) = n_{\vec{\wp}} + 1$.

  I.e., $irstart(1) = 1$, and $irstart(i) = 1 + \sum_{j=1}^{i-1} n_{\wp_j}$ for $i = 2,\ldots,m+1$.

- *jrow* contains the indices of the components in $\mathbf{y}$ required to compute the $\wp_i$-th

function $F_{\wp_i}$ in locations $irstart(i)$ to $irstart(i+1)-1$, for $i = 1, \ldots, m$. I.e.,

$$jrow = [\underbrace{\boxed{j_1^{\wp_1}}, \ldots, j_{n_{\wp_1}}^{\wp_1}}_{\mathbf{j}_{\wp_1}}, \underbrace{\boxed{j_1^{\wp_2}}, \ldots, j_{n_{\wp_2}}^{\wp_2}}_{\mathbf{j}_{\wp_2}}, \ \ldots, \ \underbrace{\boxed{j_1^{\wp_m}}, \ldots, j_{n_{\wp_m}}^{\wp_m}}_{\mathbf{j}_{\wp_m}}]^T \in \mathbb{Z}_+^{n_{\vec{\wp}}}.$$

$$\begin{matrix} irstart(1) & & irstart(2) & & & irstart(m) \\ \downarrow & & \downarrow & & & \downarrow \end{matrix}$$

Given $\mathbf{V}$ and $\widehat{\mathbf{y}}$, the following demonstrates how to compute the approximation $\tilde{F}_{\wp_i}(\widehat{\mathbf{y}})$ in (2.59), for $i = 1, \ldots, m$, from the vectors $irstart$ and $jrow$.

for $i = 1 : m$

$\mathbf{j}_{\wp_i} = jrow(irstart(i) : irstart(i+1) - 1)$

$\tilde{F}_{\wp_i}(\widehat{\mathbf{y}}) = F_{\wp_i}(\mathbf{V}(\mathbf{j}_{\wp_i}, :)\widehat{\mathbf{y}})$

end

Typically, the Jacobians of large scale problems are sparse, and this scheme will be very efficient. However, if the Jacobian is dense (or nearly so) the complexity would be on the order of mn, where m is the number of interpolation points.

The next section will discuss the computational complexity used for constructing and solving the reduced-order systems. It will also illustrate, in terms of complexity as well as computation time, that solving the POD reduced system could be more expensive than solving the original full-order system.

2.2.6 Computational Complexity

Recall the POD-DEIM reduced system for the unsteady nonlinear problem (2.1):

$$\frac{d}{dt}\widehat{\mathbf{y}}(t) = \widehat{\mathbf{A}}\widehat{\mathbf{y}}(t) + \mathbf{B}\ \mathbf{F}(\mathbf{V}_{\vec{\wp}}\widehat{\mathbf{y}}(t)), \tag{2.60}$$

and the POD-DEIM reduced system for the steady state problem (2.2):

$$\widehat{\mathbf{A}}\widehat{\mathbf{y}}(t) + \mathbf{B}\ \mathbf{F}(\mathbf{V}_{\vec{\wp}}\widehat{\mathbf{y}}(t)) = 0, \tag{2.61}$$

where $\widehat{\mathbf{A}} = \mathbf{V}^T\mathbf{A}\mathbf{V} \in \mathbb{R}^{k\times k}$, and $\mathbf{B} = \mathbf{V}^T\mathbf{U}\mathbf{U}_{\vec{\wp}}^{-1} \in \mathbb{R}^{k\times m}$ with $\mathbf{U}_{\vec{\wp}} = \mathbf{P}^T\mathbf{U}$ and $\mathbf{V}_{\vec{\wp}} = \mathbf{P}^T\mathbf{V}$. This section summarizes the computational complexity for *constructing (offline)* and *solving (online)* the POD-DEIM reduced system compared to both the original full-order system and the POD reduced system. Table 2.1 gives the *offline* computational complexity for constructing a POD-DEIM reduced system.

Procedure (*offline*)	Complexity (*offline*)
Snapshots	Problem dependent
SVD: POD basis	$\mathcal{O}(nn_s^2)$
DEIM Algorithm: m interpolation indices	$\mathcal{O}(m^4 + mn)$
Pre-compute: $\widehat{\mathbf{A}} = \mathbf{V}^T\mathbf{A}\mathbf{V}$	$\begin{cases} \mathcal{O}(n^2k + nk^2), & \text{for dense } \mathbf{A} \\ \mathcal{O}(nk + nk^2), & \text{for sparse } \mathbf{A} \end{cases}$
Pre-compute: $\mathbf{B} = \mathbf{V}^T\mathbf{U}\mathbf{U}_{\vec{\wp}}^{-1}$	$\mathcal{O}(nkm + m^2n + m^3)$

Table 2.1: Computational complexity for constructing a POD-DEIM reduced-order system.

Note that for large snapshot sets, it is far more efficient to compute the dominant singular values and vectors iteratively via ARPACK (or `svds` in MATLAB) [52]. The

computational work shown in Table 2.1 has to be done only once before solving the POD-DEIM reduced systems. The constant coefficient matrices $\widehat{\mathbf{A}}$ and $\mathbf{B}$ are pre-computed, stored and reused while solving the reduced systems.

The *online* computational complexity for solving the standard POD reduced system can even exceed the complexity for solving the original full-order system due to the orthogonal projection of the nonlinear term at each iteration, especially when $\mathbf{A} \in \mathbb{R}^{n\times n}$ represents the discretization of a linear differential operator and its sparsity is employed in the computation. This section will consider the online computational complexity and online CPU time only for solving the parametrized steady-state problem using Newton's method. More details on the online computational complexity for solving the unsteady nonlinear problem will be given in Appendix A.

Table 2.2 summarizes the complexity (Flops) for computing one Newton iteration of the full-order system (2.1) as well as the POD and POD-DEIM reduced-order systems in (2.6) and (2.61). Notice that, in the case of a sparse full-order system, the complexity $\mathcal{O}(k^3+nk^2)$ used in solving the POD reduced system could become higher than the complexity $\mathcal{O}(n^2)$ used in solving the original system once $\mathcal{O}(k^2)$ becomes proportional to $\mathcal{O}(n)$. In practice, the CPU time may not be directly proportional to these predicted Flops since there are many other factors that might affect the CPU times. However, this analysis does reflect the relative computational requirements and may be useful for predicting expected relative computational times.

The inefficiency of the POD reduced system indeed occurs in this computation.

To illustrate this effect, the nonlinear 2-D steady state problem introduced later in §4.1.4 will be considered. From Figure 2.9, the average CPU time for solving the POD reduced system in each time step exceeds the CPU time for solving the original system as soon as its dimension reaches around 80. Also, Figure 4.9 in §4.1.4 shows that, while the POD reduced system of dimension 15 gives an $\mathcal{O}(10)$ reduction in computation time as compared to the full-order system, the POD-DEIM reduced system with both POD and DEIM having dimension 15 gives an $\mathcal{O}(100)$ reduction in computation time with the same order of accuracy. These demonstrate the inefficiency of the POD reduced system that has been remedied by the introduction of DEIM.

System	Complexity (*online*)
Full (2.1)	Dense $\mathbf{A}$: $\mathcal{O}(n^3)$, Sparse $\mathbf{A}$: $\mathcal{O}(n^2)$
POD (2.6)	$\mathcal{O}(k^3 + nk^2)$
POD-DEIM (2.61)	$\mathcal{O}(k^3 + mk^2)$

Table 2.2: Comparison of the online computational work for each Newton iteration of the steady-state problem.

This chapter has illustrated how the POD-DEIM approach can be used to construct a reduced system as well as discussed its computational complexity reduction. The next chapter will consider the accuracy of the state solution from the POD-DEIM reduced system, particularly for the unsteady nonlinear problem (2.1).

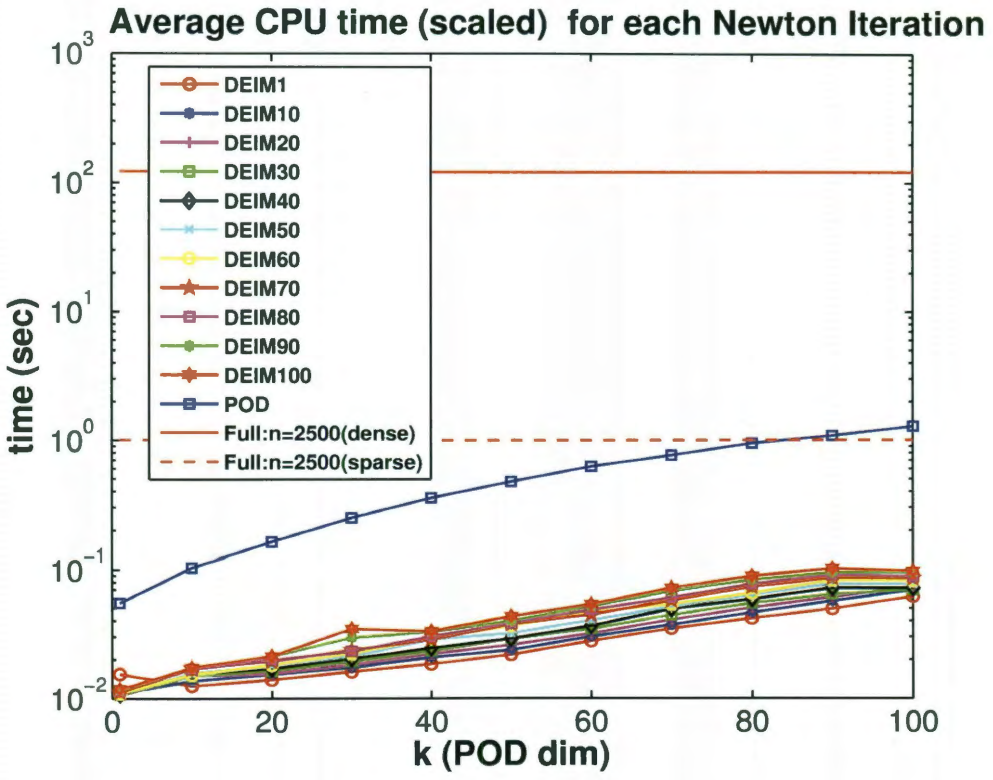


Figure 2.9: Average CPU time (scaled with the CPU time for full-sparse system) in each Newton iteration for solving the steady-state 2-D problem.

Chapter 3

A State-Space Error Estimate for POD-DEIM Reduced Systems

This chapter derives state space error bounds for the solutions of reduced-order systems constructed using Proper Orthogonal Decomposition (POD) together with the Discrete Empirical Interpolation Method (DEIM) introduced in Chapter 2. The analysis is particularly relevant to nonlinear ODE systems arising from spatial discretizations of parabolic PDEs. The resulting error estimates in 2-norm reflect the approximation property of the POD based scheme through the decay of the corresponding singular values. The derivation clearly identifies where the parabolicity is crucial. It also explains how the DEIM approximation error involving the nonlinear term comes into play.

The error bound for the DEIM approximation for a nonlinear vector-valued func-

tion given in Lemma 2.2.3 from Chapter 2 is used in this chapter to establish the global accuracy of state solution from the POD-DEIM reduced system. The derivation given here extends the error analysis of Kunish and Volkwein in [94] for POD reduced systems to the POD-DEIM reduced systems for ODEs with Lipschitz continuous nonlinearities. As before, $\|\cdot\|$ shall be used to denote the 2-norm in Euclidean space throughout this chapter. The 2-norm error estimates presented here are shown to be proportional to the sums of the singular values corresponding to neglected POD basis vectors both in Galerkin projection of the reduced system and in DEIM approximation of the nonlinear term. The separate POD basis used in DEIM to approximate the nonlinearity is very closely related Kunish-Volkwein's inclusion of finite difference snapshots [49][1].

3.1 Problem formulation

Consider systems of nonlinear ODEs of the form:

$$\frac{d}{dt}\mathbf{y}(t) = \mathbf{A}\mathbf{y}(t) + \mathbf{F}(t, \mathbf{y}(t)), \qquad \mathbf{y}(0) = \mathbf{y}_0, \qquad \text{for } t \in [0, T], \tag{3.1}$$

where the matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ is constant and the nonlinear function $\mathbf{F} : [0, T] \to \mathcal{Y}$ is assumed to be uniformly Lipschitz continuous with respect to the second argument

[1] In [49], the finite difference snapshots of the form $(\mathbf{y}_{j+1} - \mathbf{y}_j)/h$ are included into the snapshot set. This is related to the POD-DEIM approach in this thesis which considers also the nonlinear snapshots, since $(\mathbf{y}_{j+1} - \mathbf{y}_j)/h \approx \dot{\mathbf{y}}(t_j) = \mathbf{F}(\mathbf{y}_j)$, where $\mathbf{y}_j \approx \mathbf{y}(t_j)$ and $\dot{\mathbf{y}} = \mathbf{F}(\mathbf{y})$ for time stepsize h.

with Lipschitz constant $L_f > 0$ and $\mathcal{Y} \subseteq \mathbb{R}^n$. I.e., for $\mathbf{y}_1, \mathbf{y}_2 \in \mathcal{Y}$ and for all $t \in [0, T]$,

$$\|\mathbf{F}(t, \mathbf{y}_1) - \mathbf{F}(t, \mathbf{y}_2)\| \leq L_f \|\mathbf{y}_1 - \mathbf{y}_2\|. \tag{3.2}$$

Recall that, in the POD-DEIM approach, two POD bases are derived. One is the POD basis matrix $\mathbf{V} \in \mathbb{R}^{n \times k}$ of the solution $\mathbf{y}(t)$ and the other is the POD basis matrix $\mathbf{U} \in \mathbb{R}^{n \times m}$ of the nonlinear function $\mathbf{F}(t, \mathbf{y}(t))$. The corresponding POD-DEIM reduced system is constructed by applying Galerkin projection on the column space of the POD basis matrix $\mathbf{V}$, and then applying DEIM approximation to the nonlinear function using interpolation projection onto the column space of the POD basis matrix $\mathbf{U}$. The resulting reduced system is then given by

$$\frac{d}{dt}\widehat{\mathbf{y}}(t) = \widehat{\mathbf{A}}\widehat{\mathbf{y}}(t) + \mathbf{V}^T\mathbb{P}\mathbf{F}(t, \mathbf{V}\widehat{\mathbf{y}}(t)), \qquad \widehat{\mathbf{y}}(0) = \mathbf{V}^T\mathbf{y}_0, \qquad \text{for } t \in [0, T], \tag{3.3}$$

where $\widehat{\mathbf{A}} := \mathbf{V}^T\mathbf{A}\mathbf{V} \in \mathbb{R}^{k \times k}$, $\mathbb{P} := \mathbf{U}(\mathbf{P}^T\mathbf{U})^{-1}\mathbf{P}^T \in \mathbb{R}^{n \times n}$, and $\mathbf{P} \in \mathbb{R}^{n \times m}$ is a matrix whose columns come from some selected columns of the identity matrix corresponding to the DEIM indices, as defined in § 2.2 of Chapter 2. Note that in actual computation, the quantity $\mathbf{V}^T\mathbf{U}(\mathbf{P}^T\mathbf{U})^{-1} \in \mathbb{R}^{m \times m}$ in the nonlinear term would be precomputed and stored, so that the computational cost in solving (3.3) is only proportional to the reduced dimensions k and m (and not the original dimension n) as explained in the previous chapter. However, for the purpose of error analysis, this chapter will consider the nonlinear term written in the form as given in (3.3). Notice that if $m = n$, then $\mathbb{P}$ is equal to the n-by-n identity matrix and the system in (3.3) is just a reduced system constructed by the standard POD-Galerkin approach. Hence,

the error analyses given in this chapter will also apply to the POD reduced system. Recall that the Lipschitz continuity assumption on $\mathbf{F}$ in the original system (3.1) will guarantee the existence and uniqueness of the solution from the original system (by, e.g., Picard-Lindelöf theorem). The Lipschitz continuity of $\mathbf{F}$ is inherited by the reduced order nonlinear term $\widehat{\mathbf{F}}(t, \widehat{\mathbf{y}}(t)) := \mathbf{V}^T \mathbb{P} \mathbf{F}(t, \mathbf{V}\widehat{\mathbf{y}}(t))$, since

$$\begin{aligned} \|\widehat{\mathbf{F}}(t, \widehat{\mathbf{y}}_1(t)) - \widehat{\mathbf{F}}(t, \widehat{\mathbf{y}}_2(t))\| &= \|\mathbf{V}^T \mathbb{P} \mathbf{F}(t, \mathbf{V}\widehat{\mathbf{y}}_1(t)) - \mathbf{V}^T \mathbb{P} \mathbf{F}(t, \mathbf{V}\widehat{\mathbf{y}}_2(t))\| \\ &\leq L_f \|\mathbb{P}\| \|\widehat{\mathbf{y}}_1(t) - \widehat{\mathbf{y}}_2(t)\|, \end{aligned}$$

for all $t \in [0, T]$, where $\|\mathbb{P}\|$ is a bounded constant as shown in Lemma 2.2.3 and the fact that $\mathbf{V}$ has orthonormal columns is also used. Thus, existence and uniqueness of the solution to the POD-DEIM reduced system (3.3) will also be inherited.

The solution $\mathbf{y}(t)$ of the original full-order system (3.1) is then approximated by $\mathbf{V}\widehat{\mathbf{y}}$, where $\widehat{\mathbf{y}}$ is the solution from the POD-DEIM reduced system (3.3). The accuracy of this approximation therefore can be measured by considering the error $\|\mathbf{y}(t) - \mathbf{V}\widehat{\mathbf{y}}(t)\|$ for $t \in [0, T]$. The bounds for this DEIM state space error will be the main focus in this chapter. Note that the derivation for the error bounds presented later in this chapter can be applied to the case when other matrices with orthonormal columns are used in place of these POD basis matrices. This derivation also can be extended to a more general class of parametrized ODE systems.

The error bounds in discrete setting will be also considered in §3.2.2 where implicit Euler time integration is used for both full-order system (3.1) and the POD-DEIM

reduced system (3.3) as shown below:

$$\frac{1}{\triangle t}(Y_j - Y_{j-1}) = \mathbf{A}Y_j + \mathbf{f}(t_j, Y_j), \quad Y_0 = \mathbf{y}_0, \tag{3.4}$$

$$\frac{1}{\triangle t}(\widehat{Y}_j - \widehat{Y}_{j-1}) = \widehat{\mathbf{A}}\widehat{Y}_j + \mathbf{V}^T\mathbb{P}\mathbf{F}(t_j, \mathbf{V}\widehat{Y}_j), \quad \widehat{Y}_0 = \mathbf{V}^T\mathbf{y}_0, \tag{3.5}$$

where $\triangle t = T/n_t$, Y_j and $\widehat{Y}_j$ are the approximations of $\mathbf{y}(t_j)$ and $\widehat{\mathbf{y}}(t_j)$, $t_j = j\triangle t$, $j = 1, \ldots n_t$ for a given n_t. The accuracy of the POD-DEIM discretized system (3.5) will be considered through the discrete state space errors: $\|Y_j - \mathbf{V}Y_j\|$. Similar error bounds can be obtained for other discretization schemes.

The goal here is to compare the accuracy of the POD-DEIM approximate solutions with the best approximation in the least-square sense. In particular, the resulting $\mathcal{L}^2$-norm error bounds derived in this chapter will be expressed in terms of the errors $\mathcal{E}_{\mathbf{y}}$ and $\mathcal{E}_{\mathbf{f}}$ in the continuous setting (or $\bar{\mathcal{E}}_{\mathbf{y}}$ and $\bar{\mathcal{E}}_{\mathbf{f}}$ in the discrete setting) where

$$\mathcal{E}_{\mathbf{y}} := \int_0^T \|\mathbf{y}(t) - \mathbf{V}\mathbf{V}^T\mathbf{y}(t)\|^2 dt, \quad \mathcal{E}_{\mathbf{f}} := \int_0^T \|\mathbf{f}(t) - \mathbf{U}\mathbf{U}^T\mathbf{f}(t)\|^2 dt, \tag{3.6}$$

$$\bar{\mathcal{E}}_{\mathbf{y}} := \sum_{j=0}^{n_t} \|Y_j - \mathbf{V}\mathbf{V}^T Y_j\|^2, \qquad \bar{\mathcal{E}}_{\mathbf{f}} := \sum_{j=0}^{n_t} \|F_j - \mathbf{U}\mathbf{U}^T F_j\|^2, \tag{3.7}$$

with $\mathbf{f}(t) = \mathbf{F}(t, \mathbf{y}(t))$, $F_j = \mathbf{F}(t_j, Y_j)$. Note that, the least square approximation of $\mathbf{y}(t)$ in the span of $\mathbf{V}$ is given by $\mathbf{V}\mathbf{V}^T\mathbf{y}(t)$ for $\mathbf{V}^T\mathbf{V} = \mathbf{I}$, $t \in [0, T]$. Hence, $\mathcal{E}_{\mathbf{y}}$ can be viewed as the least-square error for a given basis matrix $\mathbf{V}$. The error $\mathcal{E}_{\mathbf{y}}$ is minimized when $\mathbf{V}$ is chosen to be the POD basis of the snapshot set $\{\mathbf{y}(t)|t \in [0, T]\}$. I.e., by definition [49, 50, 94], $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_k] \in \mathbb{R}^{n\times k}$ is the POD basis for $\{\mathbf{y}(t)|t \in [0, T]\}$ if it solves the following minimization problem:

$$\min_{\text{rank}\{\Phi\}=k} \int_0^T \|\mathbf{y}(t) - \Phi\Phi^T\mathbf{y}(t)\|^2 dt, \qquad \text{s.t.} \qquad \Phi^T\Phi = \mathbf{I}_k. \tag{3.8}$$

It is well known [50] that the POD basis which solves (3.8) is the set of first k dominant eigenvectors of the symmetric matrix $\mathbf{R} := \int_0^T \mathbf{y}(t)\mathbf{y}(t)^T dt \in \mathbb{R}^{n\times n}$. Using the notation established in [50], let $r = \text{rank}\{\mathbf{R}\}$ and let $\lambda_1^\infty \geq \lambda_2^\infty \geq \cdots \geq \lambda_r^\infty > 0$ be the nonzero eigenvalues of $\mathbf{R}$ with the corresponding eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_r \in \mathbb{R}^n$. Then, the minimum 2-norm error of (3.8) is given by[2]

$$\int_0^T \|\mathbf{y}(t) - \mathbf{V}\mathbf{V}^T\mathbf{y}(t)\|^2 dt = \sum_{i=k+1}^{r} \lambda_i^\infty. \tag{3.9}$$

Similarly, $\mathcal{E}_\mathbf{f}$ is minimized when $\mathbf{U} \in \mathbb{R}^{n\times m}$ is the POD basis matrix of nonlinear snapshots $\mathbf{f}(t) = \mathbf{F}(t, \mathbf{y}(t))$ for time t on the entire time interval $[0, T]$, and the minimum value is given by

$$\int_0^T \|\mathbf{f}(t) - \mathbf{U}\mathbf{U}^T\mathbf{f}(t)\|^2 dt = \sum_{i=m+1}^{r_s} s_i^\infty, \tag{3.10}$$

where $s_1^\infty \geq s_2^\infty \geq \ldots \geq s_{r_s}^\infty > 0$ are the r_s nonzero eigenvalues of $\int_0^T \mathbf{f}(t)\mathbf{f}(t)^T dt \in \mathbb{R}^{n\times n}$. Analogously, the errors $\bar{\mathcal{E}}_\mathbf{y}$ and $\bar{\mathcal{E}}_\mathbf{f}$ in the discrete setting are minimized when $\mathbf{V}$ is the POD basis of $\mathbb{Y} = [Y_1, \ldots, Y_{n_t}]$ and $\mathbf{U}$ is the POD basis of $\mathbb{F} = [F_1, \ldots, F_{n_t}]$ with the minimum values given by, respectively,

$$\sum_{j=1}^{n_s} \|Y_j - \mathbf{V}\mathbf{V}^T Y_j\|^2 = \sum_{i=k+1}^{\bar{r}} \lambda_i \tag{3.11}$$

$$\sum_{j=1}^{n_s} \|F_j - \mathbf{U}\mathbf{U}^T F_j\|^2 = \sum_{i=m+1}^{\bar{r}_s} s_i, \tag{3.12}$$

[2]The connection between (3.9) and (2.8) in Chapter 2 was demonstrated in [50] when the sampled snapshots used for (2.8) are sufficiently dense in $[0, T]$. In particular, $\sum_{i=k+1}^{r} \lambda_i \leq 2\sum_{i=k+1}^{r} \lambda_i^\infty$ when $n_s > \bar{n}_s$ for some sufficient large value $\bar{n}_s$.

where $\{\lambda_i\}_{i=1}^{\bar{r}}$ and $\{s_i\}_{i=1}^{\bar{r}_s}$ are eigenvalues of $\mathbb{Y}\mathbb{Y}^T$ and $\mathbb{F}\mathbb{F}^T$, indexed in decreasing order as defined similarly for the POD basis in Chapter 2.

3.2 Error analysis of POD-DEIM reduced system

This section develops a bound on the state approximation error for numerical solutions obtained from the POD-DEIM reduced system. The derivation will involve an application of the logarithmic norm [24] and the integral form of Gronwall's lemma [40, 13]. The logarithmic norm of $\mathbf{A} \in \mathbb{C}^{n\times n}$ with respect to the 2-norm is defined as [24]

$$\mu(A) := \lim_{h\to 0^+} \frac{\|\mathbf{I} + h\mathbf{A}\|_2 - 1}{h}, \tag{3.13}$$

which has an explicit expression suitable for calculation given by

$$\mu(A) = \max\{\mu : \mu \in \sigma\left([\mathbf{A} + \mathbf{A}^*]/2\right)\}, \tag{3.14}$$

where $\sigma\left([\mathbf{A} + \mathbf{A}^*]/2\right)$ is the set of eigenvalues of the Hermitian part $[\mathbf{A} + \mathbf{A}^*]/2$ of $\mathbf{A}$. Note that the quantity in (3.14) is also known as *numerical abscissa* of $\mathbf{A}$. A well-known property of logarithmic norm that will be used here is

$$\|e^{\mathbf{A}t}\| \leq e^{\mu(\mathbf{A})t}, \tag{3.15}$$

for $t \geq 0$ (see, e.g.[24, 83, 53]). By using (3.14), it is straightforward to show that

$$\mu(\widehat{\mathbf{A}}) \leq \mu(\mathbf{A}), \tag{3.16}$$

where $\widehat{\mathbf{A}} = \mathbf{V}^T\mathbf{A}\mathbf{V} \in \mathbb{R}^{k\times k}$ and $\mathbf{V} \in \mathbb{R}^{n\times k}$, $\mathbf{V}^T\mathbf{V} = \mathbf{I}$. Hence, (3.15) and (3.16) give

$$\|e^{\widehat{\mathbf{A}}t}\| \leq e^{\mu(\mathbf{A})t}. \tag{3.17}$$

The logarithmic norm was introduced by Dahlquist [24] to provide a mechanism for bounding the growth of the solution to a linear dynamical system of the form

$$\dot{\mathbf{y}}(t) = \mathbf{A}\mathbf{y}(t) + \mathbf{r}(t)$$

whenever $\mathbf{r}$ is a bounded function of t. For $t \geq 0$ the norm of $\mathbf{y}$ satisfies the differential inequality

$$\frac{d}{dt}\|\mathbf{y}(t)\| \leq \mu(\mathbf{A})\|\mathbf{y}(t)\| + \|\mathbf{r}(t)\|, \tag{3.18}$$

As explained by Söderlind [82], the bound (3.18) is able to distinguish between forward and reverse time and it may also be able to distinguish between stable and unstable systems. In fact, $\mu(\mathbf{A})$ may be negative and when it is, the system is certain to be stable. The opposite assertion ($\mathbf{A}$ stable implies $\mu(\mathbf{A}) < 0$) is *not* true. The non-normal matrix $\begin{bmatrix} \lambda & 1 \\ 0 & \lambda \end{bmatrix}$ provides a counterexample when $-.5 < \text{Real}(\lambda) < 0$. More details on logarithmic norms can be found in e.g. [24, 83, 25, 82]. Next, bounds on the state approximation error provided by POD-DEIM solutions will be derived in two different settings: one for the ideal case involving the full trajectory of the ODE system, as presented in §3.2.1, while the other one applies to the reduced system derived from snapshots obtained via numerical solution of the ODE system, as presented in §3.2.2.

3.2.1 Error bounds in ODE setting

This section compares the solution $\mathbf{y}(t)$ from the original full-order system (3.1) to the approximation $\mathbf{V}\widehat{\mathbf{y}}(t)$ where $\widehat{\mathbf{y}}$ is the solution of the POD-DEIM reduced system (3.3). Define the pointwise error $\mathbf{e}(t) := \mathbf{y}(t) - \mathbf{V}\widehat{\mathbf{y}}(t)$, and write

$$\mathbf{e}(t) = \rho(t) + \theta(t),$$

where $\rho(t) := \mathbf{y}(t) - \mathbf{V}\mathbf{V}^T\mathbf{y}(t)$, and $\theta(t) := \mathbf{V}\mathbf{V}^T\mathbf{y}(t) - \mathbf{V}\widehat{\mathbf{y}}(t)$. Notice that $\int_0^T \|\rho(t)\|dt = \mathcal{E}_{\mathbf{y}}$ is the minimum $\mathcal{L}^2$-norm error of the approximation on Span$\{\mathbf{U}\}$, as defined in (3.6). It therefore only remains to find a bound for $\|\theta(t)\|$ which can be done through the application of Gronwall's lemma. Define $\widehat{\theta}(t) := \mathbf{V}^T\theta(t)$. Then $\theta(t) = \mathbf{V}\widehat{\theta}(t)$. Consider $\dot{\widehat{\theta}}(t) = \mathbf{V}^T\dot{\mathbf{y}}(t) - \dot{\widehat{\mathbf{y}}}(t)$ with $\dot{\mathbf{y}}(t)$ and $\dot{\widehat{\mathbf{y}}}(t)$ satisfying (3.1) and (3.3). That is,

$$\begin{aligned} \frac{d}{dt}\widehat{\theta}(t) &= \mathbf{V}^T\left[\mathbf{A}\left[\rho(t) + \theta(t)\right] + \mathbf{F}(t, \mathbf{y}(t)) - \mathbb{P}\mathbf{F}(t, \mathbf{V}\widehat{\mathbf{y}}(t))\right] \\ &= \widehat{\mathbf{A}}\widehat{\theta}(t) + \mathbf{G}(t), \end{aligned} \tag{3.19}$$

where $\mathbf{G}(t) := \mathbf{V}^T\mathbf{A}\rho(t) + \mathbf{V}^T\left[\mathbf{F}(t, \mathbf{y}(t)) - \mathbb{P}\mathbf{F}(t, \mathbf{V}\widehat{\mathbf{y}}(t))\right]$. Note that $\theta(0) = 0$ since $\widehat{\mathbf{y}}_0 = \mathbf{V}^T\mathbf{y}(0)$. Hence, the solution to (3.19) can be written as

$$\widehat{\theta}(t) = \int_0^t e^{\widehat{\mathbf{A}}(t-s)}\mathbf{G}(s)ds. \tag{3.20}$$

To find a bound for $\|\mathbf{G}(t)\|$, write

$$\mathbf{G}(t) = \mathbf{V}^T\mathbf{A}\rho(t) + \mathbf{V}^T\Big[(\mathbf{I} - \mathbb{P})\mathbf{F}(t, \mathbf{y}(t)) + \mathbb{P}[\mathbf{F}(t, \mathbf{y}(t)) - \mathbf{F}(t, \mathbf{V}\widehat{\mathbf{y}}(t))]\Big].$$

The Lipschitz continuity of $\mathbf{F}$, together with $(\mathbf{I}-\mathbb{P})\mathbf{F}(t,\mathbf{y}(t)) = (\mathbf{I}-\mathbb{P})\mathbf{w}(t)$ from (2.22) in Lemma 2.2.3, where $\mathbf{w}(t) := \mathbf{F}(t,\mathbf{y}(t)) - \mathbf{U}\mathbf{U}^T\mathbf{F}(t,\mathbf{y}(t))$, implies

$$\begin{aligned}\|\mathbf{G}(t)\| &\leq \|\mathbf{V}^T\mathbf{A}\rho(t)\| + \|\mathbf{V}^T(\mathbf{I}-\mathbb{P})\,\mathbf{F}(t,\mathbf{y}(t))\| + \|\mathbf{V}^T\mathbb{P}\|L_f\|\mathbf{y}(t)-\mathbf{V}\widehat{\mathbf{y}}(t)\| \\ &\leq \alpha\|\rho(t)\| + \beta\|\mathbf{w}(t)\| + \gamma\|\theta(t)\|, \qquad (3.21)\end{aligned}$$

where $\alpha := \|\mathbf{V}^T\mathbf{A}\| + \|\mathbf{V}^T\mathbb{P}\|L_f$, $\beta := \|\mathbf{V}^T(\mathbf{I}-\mathbb{P})\|$, $\gamma := \|\mathbf{V}^T\mathbb{P}\|L_f$. Since $\|\widehat{\theta}(t)\| = \|\theta(t)\|$ and $\|e^{\widehat{\mathbf{A}}(t-s)}\| \leq e^{\mu(t-s)}$ where $\mu := \mu(\mathbf{A})$, (3.20) and (3.21) imply

$$\begin{aligned}\|\theta(t)\| &\leq \int_0^t \|e^{\widehat{\mathbf{A}}(t-s)}\|\Big(\alpha\|\rho(s)\| + \beta\|\mathbf{w}(s)\| + \gamma\|\theta(s)\|\Big)ds \\ &\leq \eta + \gamma\int_0^t e^{\mu(t-s)}\|\theta(s)\|ds, \qquad (3.22)\end{aligned}$$

where η satisfies $\eta \geq \widehat{\eta}(t) := \int_0^t e^{\mu(t-s)}\Big(\alpha\|\rho(s)\| + \beta\|\mathbf{w}(s)\|\Big)ds$, for all $t \in [0,T]$. Applying the integral form of Gronwall's inequality [13] to (3.22) gives

$$\|\theta(t)\| \leq \eta e^{\gamma b_\mu(t)}, \qquad (3.23)$$

where $b_\mu(t) := \int_0^t e^{\mu(t-s)}ds = \begin{cases} \frac{1}{\mu}(e^{\mu t}-1) & , \ \mu \neq 0 \\ t & , \ \mu = 0 \end{cases}$. Now, η can be specified by applying the Cauchy-Schwarz inequality to $\widehat{\eta}(t)$ so that we can put

$$\eta := \Big[a_\mu(T)\Big(\alpha^2\ \mathcal{E}_{\mathbf{y}} + \beta^2\ \mathcal{E}_{\mathbf{f}}\Big)\Big]^{1/2},$$

where $a_\mu(t) := 2\int_0^t e^{2\mu(t-s)}ds = \begin{cases} \frac{1}{\mu}(e^{2\mu t}-1) & , \ \mu \neq 0 \\ 2t & , \ \mu = 0 \end{cases}$, with $\mathcal{E}_{\mathbf{y}} = \int_0^T \|\rho(t)\|^2 dt$ and $\mathcal{E}_{\mathbf{f}} = \int_0^T \|\mathbf{w}(t)\|^2 dt$, as defined in (3.6). Using $b_\mu(t) \leq b_\mu(T)$ for all $t \in [0,T]$ and (3.23), a bound for $\|\theta(t)\|^2$ is given by

$$\|\theta(t)\|^2 \leq a_\mu(T)e^{2\gamma b_\mu(T)}\Big[\alpha^2\ \mathcal{E}_{\mathbf{y}} + \beta^2\ \mathcal{E}_{\mathbf{f}}\Big], \qquad (3.24)$$

for all $t \in [0, T]$. Finally, since $\rho(t)^T \theta(t) = 0$, then

$$\int_0^T \|e(t)\|^2 dt = \int_0^T \|\rho(t)\|^2 dt + \int_0^T \|\theta(t)\|^2 dt \;\leq\; \mathbf{C}\left(\mathcal{E}_{\mathbf{y}} + \mathcal{E}_{\mathbf{f}}\right), \qquad (3.25)$$

where $\mathbf{C} = \max\{1 + c_\mu \alpha^2 T, c_\mu \beta^2 T\}$ and $c_\mu = a_\mu(T) e^{2\gamma b_\mu(T)}$. Notice that when $\mu < 0$, $a_\mu(t), b_\mu(t) < \frac{1}{|\mu|}$ for all $t > 0$ and hence $c_\mu < \frac{e^{2\gamma/|\mu|}}{|\mu|}$, which does not depend on the final integration time T. In this case, the error bound in (3.25) is linear in T as well as the least-square errors $\mathcal{E}_{\mathbf{y}}$ and $\mathcal{E}_{\mathbf{f}}$.

In practice, the exact solutions of dynamical systems are not available and the numerical solutions from their discretized systems are often required. The next section will apply an analogous derivation to analyze the accuracy of a discretized POD-DEIM reduced system compared to the discretized full-order system.

3.2.2 Error bounds in discrete setting

This section compares the solutions of (3.4) and (3.5) obtained from implicit Euler time discretization for the full-order system (3.1) and the POD-DEIM reduced system (3.3), respectively. Define the error at time step t_j as $E_j := Y_j - \mathbf{V}\widehat{Y}_j$, and write

$$E_j = \rho_j + \theta_j,$$

where $\rho_j := Y_j - \mathbf{V}\mathbf{V}^T Y_j, \quad \theta_j := \mathbf{V}\mathbf{V}^T Y_j - \mathbf{V}\widehat{Y}_j$. Since $\rho_j^T \theta_j = 0$, $\|\mathbf{E}_j\|^2 = \|\rho_j\|^2 + \|\theta_j\|^2$. Note that, from (3.7), $\sum_{j=0}^{n_t} \|\rho_j\|^2 = \bar{\mathcal{E}}_{\mathbf{y}}$, and it therefore remains to determine a bound for the norm of θ_j. Analogous to the continuous case, the *discrete Gronwall's lemma* can be used to obtain a bound for $\|\theta_j\|$. Define $\widehat{\theta}_j := \mathbf{V}^T \theta_j = \mathbf{V}^T Y_j - \widehat{Y}_j$ for

$\mathbf{V}^T\mathbf{V} = \mathbf{I}$. Then $\theta_j = \mathbf{V}\widehat{\theta}_j$. Consider

$$\frac{1}{\triangle t}(\widehat{\theta}_j - \widehat{\theta}_{j-1}) = \mathbf{V}^T\Big[\frac{1}{\triangle t}(Y_j - Y_{j-1})\Big] - \Big[\frac{1}{\triangle t}(\widehat{Y}_j - \widehat{Y}_{j-1})\Big].$$

That is, using $Y_j - \mathbf{V}\widehat{Y}_j = \rho_j + \theta_j$ gives

$$\begin{aligned}\frac{1}{\triangle t}(\widehat{\theta}_j - \widehat{\theta}_{j-1}) &= \mathbf{V}^T\mathbf{A}[\rho_j + \theta_j] + \mathbf{V}^T[\mathbf{F}(t_j, Y_j) - \mathbb{P}\mathbf{F}(t_j, \mathbf{V}\widehat{Y}_j)], &(3.26)\\ &= \widehat{\mathbf{A}}\widehat{\theta}_j + \mathbf{G}_j, &(3.27)\end{aligned}$$

where $\mathbf{G}_j := \mathbf{V}^T\mathbf{A}\rho_j + \mathbf{V}^T[\mathbf{F}(t_j, Y_j) - \mathbb{P}\mathbf{F}(t_j, \mathbf{V}\widehat{Y}_j)]$. From successive substitution of $\widehat{\theta}_{j-1}, \ldots, \widehat{\theta}_0$,

$$\begin{aligned}\widehat{\theta}_j &= (\mathbf{I} - \triangle t\widehat{\mathbf{A}})^{-1}\Big[\widehat{\theta}_{j-1} + \triangle t\mathbf{G}_j\Big] &(3.28)\\ &= (\mathbf{I} - \triangle t\widehat{\mathbf{A}})^{-j}\widehat{\theta}_0 + \triangle t\sum_{i=1}^{j}\Big[(\mathbf{I} - \triangle t\widehat{\mathbf{A}})^{-i}\mathbf{G}_{j-i+1}\Big]. &(3.29)\end{aligned}$$

To find a bound for $\|\mathbf{G}_j\|$, first rewrite

$$\mathbf{G}_j = \mathbf{V}^T\mathbf{A}\rho_j + \mathbf{V}^T\Big[(\mathbf{I} - \mathbb{P})\mathbf{F}(t_j, Y_j) + \mathbb{P}[\mathbf{F}(t_j, Y_j) - \mathbf{F}(t_j, \mathbf{V}\widehat{Y}_j)]\Big].$$

Then, by using Cauchy-Schwarz inequality, Lipschitz continuity of $\mathbf{F}$, and $(\mathbf{I}-\mathbb{P})\mathbf{F}(t_j, Y_j) = (\mathbf{I} - \mathbb{P})\mathbf{w}_j$ for $\mathbf{w}_j = (\mathbf{I} - \mathbf{U}\mathbf{U}^T)\mathbf{F}(t_j, Y_j)$ from Lemma 2.2.3,

$$\|\mathbf{G}_j\| \leq \alpha\|\rho_j\| + \beta\|\mathbf{w}_j\| + \gamma\|\theta_j\|, \qquad (3.30)$$

where $\alpha = \|\mathbf{V}^T\mathbf{A}\| + \|\mathbf{V}^T\mathbb{P}\|L_f$, $\beta = \|\mathbf{V}^T(\mathbf{I}-\mathbb{P})\|$, $\gamma = \|\mathbf{V}^T\mathbb{P}\|L_f$. Let $\mu = \mu(\mathbf{A})$ and assume $\mu\triangle t < 1$, so that $\mathbf{I} - \triangle t\widehat{\mathbf{A}}$ is invertible. Then $\|(\mathbf{I} - \triangle t\widehat{\mathbf{A}})^{-1}\| \leq (1 - \triangle t\mu)^{-1}$ [83]. Let $\zeta := (1 - \triangle t\mu)^{-1}$. Since $\|\theta_j\| = \|\widehat{\theta}_j\|$, then (3.29) and (3.30) give

$$\|\theta_j\| \leq \zeta^j\|\theta_0\| + \triangle t\sum_{i=1}^{j}\zeta^i\|\mathbf{G}_{j-i+1}\| \leq \bar{\eta} + \triangle t\gamma\sum_{\ell=1}^{j}\widehat{\zeta}^{\ell}\|\theta_\ell\|, \qquad (3.31)$$

where $\widehat{\zeta}^{\ell} := \zeta^{j-\ell+1}$ and $\bar{\eta}$ satisfies $\bar{\eta} \geq \eta_j := \zeta^j \|\theta_0\| + \triangle t \sum_{\ell=1}^{j} \left[\widehat{\zeta}^{\ell}(\alpha\|\rho_\ell\| + \beta\|\mathbf{w}_\ell\|)\right]$, for all $j = 1, \ldots, n_t$. By using the Cauchy-Schwarz inequality, we can put $\bar{\eta}$ as

$$\bar{\eta} \;:=\; \left[\triangle t \bar{a}_\mu \left(\alpha^2\, \bar{\mathcal{E}}_{\mathbf{y}} + \beta^2\, \bar{\mathcal{E}}_{\mathbf{f}}\right)\right]^{1/2}, \tag{3.32}$$

where $\bar{a}_\mu := 2\triangle t \sum_{\ell=1}^{n_t} \zeta^{2\ell} = 2\triangle t \zeta^2 \left(\frac{1-\zeta^{2n_t}}{1-\zeta^2}\right)$ and $\bar{\mathcal{E}}_{\mathbf{y}} = \sum_{\ell=1}^{n_t} \|\rho_\ell\|^2$, $\bar{\mathcal{E}}_{\mathbf{f}} = \sum_{\ell=1}^{n_t} \|\mathbf{w}_\ell\|^2$ as defined in (3.7). Note that $\theta_0 = 0$, since $Y_0 = \mathbf{y}_0$ and $\widehat{Y}_0 = \mathbf{V}^T\mathbf{y}_0$. Now we can apply the discrete Gronwall lemma (e.g. [22]) on (3.31) to obtain

$$\|\theta_j\| \;\leq\; \bar{\eta} \exp\left\{\triangle t \gamma \sum_{\ell=1}^{j} \widehat{\zeta}^{\ell}\right\}. \tag{3.33}$$

Let $\bar{b}_\mu := \triangle t \sum_{\ell=1}^{n_t} \zeta^\ell = \triangle t \zeta \left(\frac{1-\zeta^{n_t}}{1-\zeta}\right)$. Then, using $T = n_t \triangle t$ gives

$$\sum_{j=1}^{n_t} \|\theta_j\|^2 \;\leq\; \sum_{j=1}^{n_t} \bar{\eta}^2 e^{2\gamma\bar{b}_\mu} \;\leq\; T\bar{a}_\mu e^{2\gamma\bar{b}_\mu} \left(\alpha^2\, \bar{\mathcal{E}}_{\mathbf{y}} + \beta^2\, \bar{\mathcal{E}}_{\mathbf{f}}\right). \tag{3.34}$$

Finally, since $\rho_j^T\theta_j = 0$, then $\sum_{j=0}^{n_t} \|Y_j - \mathbf{V}\widehat{Y}_j\|^2 = \sum_{j=0}^{n_t} \|\rho_j\|^2 + \sum_{j=0}^{n_t} \|\theta_j\|^2$ and

$$\sum_{j=0}^{n_t} \|Y_j - \mathbf{V}\widehat{Y}_j\|^2 \;\leq\; \bar{\mathbf{C}}\left(\bar{\mathcal{E}}_{\mathbf{y}} + \bar{\mathcal{E}}_{\mathbf{f}}\right), \tag{3.35}$$

where $\bar{\mathbf{C}} = \max\{1 + \bar{c}_\mu\alpha^2 T, \bar{c}_\mu\beta^2 T\}$, $\bar{c}_\mu := \bar{a}_\mu e^{2\gamma\bar{b}_\mu}$. Note that for $\zeta := (1 - \triangle t\mu)^{-1}$ and $\mu = \mu(\mathbf{A})$, if $\mu < 0$, then $0 < \zeta < 1$ and

$$\bar{b}_\mu \leq \triangle t\zeta \left(\sum_{\ell=0}^{\infty} \zeta^\ell\right) = \triangle t\zeta \left(\frac{1}{1-\zeta}\right) = \triangle t \frac{1/(1-\triangle t\mu)}{1 - 1/(1-\triangle t\mu)} = \frac{1}{|\mu|},$$

and similarly, then $0 < \zeta^2 < 1$ and

$$\bar{a}_\mu \leq 2\triangle t \frac{\zeta^2}{1-\zeta^2} = 2\triangle t \frac{1/(1-\triangle t\mu)^2}{1 - 1/(1-\triangle t\mu)^2} = \frac{1}{|\mu| + (\triangle t|\mu|^2)/2}.$$

That is $\bar{c}_\mu \leq \left(\frac{1}{|\mu|+(\triangle t|\mu|^2)/2}\right) e^{2\gamma/|\mu|}$ which is uniformly bounded for a fixed $\triangle t$. In this case, $\bar{c}_\mu$ converges to c_μ in the continuous setting as $\triangle t \to 0$. The following summarizes the error bounds just derived in § 3.2.1 and § 3.2.2.

Theorem 3.2.1 *Let $\mathbf{y}(t)$ be the solution of the original full-order system (3.1) and $\widehat{\mathbf{y}}(t)$ be the solution of the POD-DEIM reduced system (3.3), for $t \in [0,T]$. Let $\mu = \mu(\mathbf{A})$ be the logarithmic norm defined in (3.13) and assume that $\mathbf{F}(t,\mathbf{y})$ in (3.1) is Lipschitz continuous in the second argument, with Lipschitz constant L_f as in (3.2). Let Y_j and $\widehat{Y}_j$ be the solutions of the discretized systems (3.4) and (3.5) from implicit Euler method at $t_j = j\triangle t \in [0,T]$, $\triangle t = T/n_t$ for $j = 0, \ldots n_t$. Assume that $\mu\triangle t < 1$. Then*

$$\int_0^T \|\mathbf{y}(t) - \mathbf{V}\widehat{\mathbf{y}}(t)\|^2 dt \;\leq\; \mathbf{C}\left(\mathcal{E}_{\mathbf{y}} + \mathcal{E}_{\mathbf{f}}\right), \tag{3.36}$$

$$\sum_{j=0}^{n_t} \|Y_j - \mathbf{V}\widehat{Y}_j\|^2 \;\leq\; \bar{\mathbf{C}}\left(\bar{\mathcal{E}}_{\mathbf{y}} + \bar{\mathcal{E}}_{\mathbf{f}}\right), \tag{3.37}$$

where $\mathbf{C} := \max\{1 + c_\mu\alpha^2 T, c_\mu\beta^2 T\}$, $\bar{\mathbf{C}} := \max\{1 + \bar{c}_\mu\alpha^2 T, \bar{c}_\mu\beta^2 T\}$,

$$\alpha := \|\mathbf{V}^T\mathbf{A}\| + \|\mathbf{V}^T\mathbb{P}\|L_f, \quad \beta := \|\mathbf{V}^T(\mathbf{I}-\mathbb{P})\|, \quad \gamma := \|\mathbf{V}^T\mathbb{P}\|L_f, \tag{3.38}$$

$$c_\mu := a_\mu e^{2\gamma b_\mu}, \text{ with } \begin{cases} a_\mu = \frac{1}{\mu}(e^{2\mu T}-1), & b_\mu = \frac{1}{\mu}(e^{\mu T}-1) \quad , \quad \mu \neq 0 \\ a_\mu = 2T, & b_\mu = T \qquad\qquad , \quad \mu = 0 \end{cases} \tag{3.39}$$

$$\bar{c}_\mu := \bar{a}_\mu e^{2\gamma\bar{b}_\mu}, \text{ with } \bar{a}_\mu = 2\triangle t\zeta^2\left(\frac{1-\zeta^{2n_t}}{1-\zeta^2}\right), \; \bar{b}_\mu = \triangle t\zeta\left(\frac{1-\zeta^{n_t}}{1-\zeta}\right), \tag{3.40}$$

$\zeta = (1-\triangle t\mu)^{-1}$ and $\mathcal{E}_{\mathbf{y}}, \mathcal{E}_{\mathbf{f}}, \bar{\mathcal{E}}_{\mathbf{y}}, \bar{\mathcal{E}}_{\mathbf{f}}$ are the minimum $\mathcal{L}^2$-norm errors as defined in (3.6) and (3.7).

Remark 3.2.2 *Using the notation and assumptions from Theorem 3.2.1:*

(i) If $\mu < 0$, then $a_\mu, b_\mu < \frac{1}{|\mu|}$ and $\bar{a}_\mu < \frac{1}{|\mu|+(\triangle t|\mu|^2)/2}, \bar{b}_\mu < \frac{1}{|\mu|}$. That is, c_μ and $\bar{c}_\mu$ in (3.39) can be bounded by a constant independent of T or n_t (for fixed $\triangle t$):

$$c_\mu \;<\; \frac{e^{2\gamma/|\mu|}}{|\mu|}, \quad \bar{c}_\mu < \frac{e^{2\gamma/|\mu|}}{|\mu| + (\triangle t|\mu|^2)/2}. \tag{3.41}$$

(ii) When the POD-DEIM reduced system (3.3) is constructed from the POD basis matrices $\mathbf{V} \in \mathbb{R}^{n\times k}$, *and* $\mathbf{U} \in \mathbb{R}^{n\times m}$ *of solution snapshots and nonlinear snapshots, respectively, which satisfy (3.8), then, from (3.9) and (3.10),* $\mathcal{E}_{\mathbf{y}} = \sum_{\ell=k+1}^{r} \lambda_\ell^\infty, \mathcal{E}_{\mathbf{f}} = \sum_{\ell=m+1}^{r_s} s_\ell^\infty$. *In this case, if also* $\mu = \mu(A) < 0$, *then from (i) the error bound can be simplified as*

$$\int_0^T \|\mathbf{y}(t) - \mathbf{V}\widehat{\mathbf{y}}(t)\|^2 dt \;\leq\; \mathbf{C}_o \left(\sum_{\ell=k+1}^{r} \lambda_\ell^\infty + \sum_{\ell=m+1}^{r_s} s_\ell^\infty \right), \tag{3.42}$$

where $\mathbf{C}_o := \max\{1 + c_o\alpha^2 T, c_o\beta^2 T\}$, $c_o = \frac{e^{2\gamma/|\mu|}}{|\mu|}$ *with* α,β, γ *from (3.38).*

(iii) Similarly, when the discretized POD-DEIM reduced system (3.5) is constructed from the POD basis matrices $\mathbf{V} \in \mathbb{R}^{n\times k}$, *and* $\mathbf{U} \in \mathbb{R}^{n\times m}$ *of snapshot matrices* $\mathbb{Y} = [Y_1, \ldots, Y_{n_t}]$ *and* $\mathbb{F} = [\mathbf{F}(t_1, Y_1), \ldots, \mathbf{F}(t_{n_t}, Y_{n_t})] \in \mathbb{R}^{n\times n_t}$, *then using (3.11) and (3.12) gives* $\bar{\mathcal{E}}_{\mathbf{y}} = \sum_{\ell=k+1}^{\bar{r}} \lambda_\ell, \bar{\mathcal{E}}_{\mathbf{f}} = \sum_{\ell=m+1}^{\bar{r}_s} s_\ell$. *In this case, if also* $\mu = \mu(\mathbf{A}) < 0$, *then from (i),*

$$\sum_{j=0}^{n_t} \|Y_j - \mathbf{V}\widehat{Y}_j\|^2 \;\leq\; \bar{\mathbf{C}}_o \left(\sum_{\ell=k+1}^{\bar{r}} \lambda_\ell + \sum_{\ell=m+1}^{\bar{r}_s} s_\ell \right), \tag{3.43}$$

where $\bar{\mathbf{C}}_o := \max\{1+\bar{c}_o\alpha^2 T, \bar{c}_o\beta^2 T\}$. $\bar{c}_o = \frac{e^{2\gamma/|\mu|}}{|\mu|+(\triangle t|\mu|^2)/2}$, *with* α,β, γ *from (3.38).*

When (ii) or (iii) of Remark 3.2.2 holds true, $\mathcal{E}_{\mathbf{y}}$ and $\mathcal{E}_{\mathbf{f}}$ in (3.6) or $\bar{\mathcal{E}}_{\mathbf{y}}$ and $\bar{\mathcal{E}}_{\mathbf{f}}$ in (3.7) are minimized as noted earlier. For a special case, when (i) and (iii) in Theorem 3.2.1 are both true, the pointwise error in the discrete setting is uniformly bounded at each time step $j = 1, \ldots, n_t$:

$$\|Y_j - \mathbf{V}\widehat{Y}_j\|^2 \;\leq\; \bar{c} \left(\sum_{\ell=k+1}^{\bar{r}} \lambda_\ell + \sum_{\ell=m+1}^{\bar{r}_s} s_\ell \right), \tag{3.44}$$

where $\bar{c} := 2\max\{1 + \bar{c}_\mu \alpha^2, \bar{c}_\mu \beta^2\}$, $\bar{c}_o = \frac{e^{2\gamma/|\mu|}}{|\mu|+(\Delta t|\mu|^2)/2}$, with α,β, γ defined as in (3.38).

The error analysis in this section has illustrated the basic idea concerning how the parabolicity assumption together with the combination of the POD-DEIM approach will lead to a bound on the state approximation error. However, it depends upon the ability to separate out a constant matrix $\mathbf{A}$ on the right hand side of the ODE system. The key tool in this analysis has been the logarithmic norm. The next section will utilize a generalization to obtain an error estimate that does not require the constant matrix $\mathbf{A}$.

3.3 Analysis based on generalized logarithmic norm

A logarithmic norm was used in the previous section to analyze the state approximation error of the POD-DEIM system. That approach required the presence of a constant matrix $\mathbf{A}$. More generally, as is done in [82], one can apply a logarithmic norm argument to a local linearization about the trajectory. The analysis in this section will employ a generalization of the logarithmic norm that avoids the need for a linearization or for the presence of a constant $\mathbf{A}$. The generalization of logarithmic norm to unbounded nonlinear operators was introduced through *logarithmic Lipschitz constants* in [81] to avoid working with linearizations and logarithmic norms that are only applicable to linear operators. Here, this tool will be used to develop a *conceptual* framework suitable for analyzing POD-DEIM reduced systems of nonlinear

ODEs. Consider nonlinear ODEs of the form:

$$\dot{\mathbf{y}}(t) \;=\; \mathbf{F}(t, \mathbf{y}(t)), \quad \mathbf{y}(0) = \mathbf{y}_0, \tag{3.45}$$

where $\mathbf{F} : [0, T] \times \mathcal{Y} \to \mathbb{R}^n$, $\mathcal{Y} \subseteq \mathbb{R}^n$ with the POD-DEIM reduced system of the form:

$$\dot{\widehat{\mathbf{y}}}(t) \;=\; \widehat{\mathbf{F}}(t, \widehat{\mathbf{y}}(t)), \quad \widehat{\mathbf{y}}(0) = \mathbf{V}^T \mathbf{y}_0, \tag{3.46}$$

where $\widehat{\mathbf{F}} : [0, T] \times \widehat{\mathcal{Y}} \to \mathbb{R}^k$, $\widehat{\mathcal{Y}} \subseteq \mathbb{R}^k$, $\widehat{\mathbf{F}}(t, \widehat{\mathbf{y}}) = \mathbf{V}^T \mathbb{P} \mathbf{F}(t, \mathbf{V}\widehat{\mathbf{y}})$ for $\widehat{\mathbf{y}} \in \widehat{\mathcal{Y}}$, $t \in [0, T]$. Note that the POD reduced system can be obtained by replacing $\mathbb{P}$ with the n-by-n identity matrix. Hence, the error bounds derived in this section also apply to the POD reduced system. This section will use the Euclidian inner product $\langle \cdot, \cdot \rangle : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, for some positive integer d, i.e. $\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{u}^T \mathbf{v}$ for $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$, and its induced norm $\|\mathbf{u}\| = \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle}$, $\mathbf{u} \in \mathbb{R}^d$. As in [82], for a map $\mathbf{F} : [0, T] \times \mathcal{Y} \to \mathbb{R}^d$, $\mathcal{Y} \subseteq \mathbb{R}^d$, the least upper bound (lub) *logarithmic Lipschitz constants* with respect to the inner product $\langle \cdot, \cdot \rangle$ can be defined, uniformly for all $t \in [0, T]$, as:

$$M[\mathbf{F}] := \sup_{\mathbf{u} \neq \mathbf{v}} \frac{\langle \mathbf{u} - \mathbf{v}, \mathbf{F}(t, \mathbf{u}) - \mathbf{F}(t, \mathbf{v}) \rangle}{\|\mathbf{u} - \mathbf{v}\|^2}. \tag{3.47}$$

The convergence of the solution as well as the stability of the corresponding POD-DEIM reduced system can be analyzed by using these logarithmic Lipschitz constants. The map $\mathbf{F}$ is called *uniformly negative monotone* if $M[\mathbf{F}] < 0$, in which case it will be shown that the error bound of the reduced-order solution is uniformly bounded on $t \in [0, T]$.

The asymptotic error analysis will be considered first in § 3.3.1 for the continuous setting, where the overall accuracy of the reduced system is only contributed from

applying the POD-DEIM technique without other effects, such as the choice of time integration method. Then, a framework for error analysis in the discrete setting for the implicit Euler time integration scheme will be presented in § 3.3.2. Note that Lipschitz continuity of $\mathbf{F}$ is the only main assumption used in this section. The resulting error bounds in the 2-norm, which are summarized in Theorem 3.3.1, reflect the approximation property of POD based scheme through the decay of singular values, as in §3.2. The differences of the results here from the ones in §3.2 will be discussed at the end of this section.

3.3.1 Error bounds in continuous ODE setting

Consider the error of the solution from the POD-DEIM reduced system of the form

$$\mathbf{e}(t) = \mathbf{y}(t) - \mathbf{y}_r(t), \quad \mathbf{y}_r(t) := \mathbf{V}\widehat{\mathbf{y}}(t),$$

where $\mathbf{V} \in \mathbb{R}^{n\times k}$ is the POD basis matrix with $\mathbf{y}$ and $\widehat{\mathbf{y}}$ satisfying (3.45) and (3.46), respectively. Again, put

$$\mathbf{e}(t) = \rho(t) + \theta(t),$$

where $\rho(t) := \mathbf{y}(t) - \mathbf{V}\mathbf{V}^T\mathbf{y}(t)$, $\theta(t) := \mathbf{V}\mathbf{V}^T\mathbf{y}(t) - \mathbf{V}\widehat{\mathbf{y}}(t)$, and note that $\widehat{\mathbf{y}}(0) = \mathbf{V}^T\mathbf{y}_0$ implies $\theta(0) = 0$. Note also that $\rho(t)^T\theta(t) = 0$ implies that $\|e(t)\|^2 = \|\rho(t)\|^2 + \|\theta(t)\|^2$. Define $\widehat{\theta}(t) := \mathbf{V}^T\theta(t) = \mathbf{V}^T\mathbf{y}(t) - \widehat{\mathbf{y}}(t)$. As before, $\theta(t) = \mathbf{V}\widehat{\theta}(t)$ and hence $\|\theta(t)\| =$

$\|\widehat{\theta}(t)\|$. Now, consider

$$\dot{\widehat{\theta}}(t) = \mathbf{V}^T\dot{\mathbf{y}}(t) - \dot{\widehat{\mathbf{y}}}(t) = \mathbf{V}^T\mathbf{F}(t,\mathbf{y}(t)) - \widehat{\mathbf{F}}(t,\widehat{\mathbf{y}}(t)) \tag{3.48}$$

$$= \widehat{\mathbf{F}}(t,\mathbf{V}^T\mathbf{y}(t)) - \widehat{\mathbf{F}}(t,\widehat{\mathbf{y}}(t)) + \widehat{\mathbf{r}}(t), \tag{3.49}$$

where

$$\widehat{\mathbf{r}}(t) := \mathbf{V}^T\mathbf{F}(t,\mathbf{y}(t)) - \widehat{\mathbf{F}}(t,\mathbf{V}^T\mathbf{y}(t)). \tag{3.50}$$

Next, since $\|\widehat{\theta}(t)\|^2 = \widehat{\theta}(t)^T\widehat{\theta}(t)$,

$$\begin{aligned}
\frac{d}{dt}\|\widehat{\theta}(t)\| &= \frac{\left\langle \widehat{\theta}(t), \dot{\widehat{\theta}}(t) \right\rangle}{\|\widehat{\theta}(t)\|} \\
&= \frac{\left\langle \widehat{\theta}(t), \widehat{\mathbf{F}}(t,\mathbf{V}^T\mathbf{y}(t)) - \widehat{\mathbf{F}}(t,\widehat{\mathbf{y}}(t)) + \widehat{\mathbf{r}}(t) \right\rangle}{\|\widehat{\theta}(t)\|} \\
&= \frac{\left\langle \widehat{\theta}(t), \widehat{\mathbf{F}}(t,\mathbf{V}^T\mathbf{y}(t)) - \widehat{\mathbf{F}}(t,\widehat{\mathbf{y}}(t)) \right\rangle}{\|\widehat{\theta}(t)\|} + \frac{\left\langle \widehat{\theta}(t), \widehat{\mathbf{r}}(t) \right\rangle}{\|\widehat{\theta}(t)\|} \\
&\leq M[\widehat{\mathbf{F}}]\|\widehat{\theta}(t)\| + \|\widehat{\mathbf{r}}(t)\|.
\end{aligned}$$

Notice that $\|\widehat{\mathbf{r}}(t)\|$ is independent of $\|\widehat{\theta}(t)\|$ and hence Gronwall's inequality is not required here. Since $\|\theta(t)\| = \|\widehat{\theta}(t)\|$ and $\|\theta(0)\| = 0$, then

$$\|\theta(t)\| \leq e^{M[\widehat{\mathbf{F}}]t}\|\theta(0)\| + \int_0^t e^{M[\widehat{\mathbf{F}}](t-\tau)}\|\widehat{\mathbf{r}}(\tau)\|d\tau = \int_0^t e^{M[\widehat{\mathbf{F}}](t-\tau)}\|\widehat{\mathbf{r}}(\tau)\|d\tau. \tag{3.51}$$

Now, the expression for $\widehat{\mathbf{r}}(t)$ can be rewritten as the sum of differences, which can be estimated in terms of the neglected singular values as follows. From Lemma 2.2.3,

for $\mathbf{w}(t) = \mathbf{F}(t, \mathbf{y}(t)) - \mathbf{U}\mathbf{U}^T\mathbf{F}(t, \mathbf{y}(t))$,

$$\begin{aligned}\widehat{\mathbf{r}}(t) &= \mathbf{V}^T\mathbf{F}(t, \mathbf{y}(t)) - \widehat{\mathbf{F}}(t, \mathbf{V}^T\mathbf{y}(t)) = \mathbf{V}^T[\mathbf{F}(t, \mathbf{y}(t)) - \mathbb{P}\mathbf{F}(t, \mathbf{V}\mathbf{V}^T\mathbf{y}(t))] \\ &= \mathbf{V}^T[\mathbf{F}(t, \mathbf{y}(t)) - \mathbb{P}\mathbf{F}(t, \mathbf{y}(t)) + \mathbb{P}\mathbf{F}(t, \mathbf{y}(t)) - \mathbb{P}\mathbf{F}(t, \mathbf{V}\mathbf{V}^T\mathbf{y}(t))] \\ &= \mathbf{V}^T(\mathbf{I} - \mathbb{P})\mathbf{w}(t) + \mathbf{V}^T\mathbb{P}(\mathbf{F}(t, \mathbf{y}(t)) - \mathbf{F}(t, \mathbf{V}\mathbf{V}^T\mathbf{y}(t))).\end{aligned}$$

The Lipschitz continuity of $\mathbf{F}$ implies $\|\mathbf{F}(t, \mathbf{y}(t)) - \mathbf{F}(t, \mathbf{V}\mathbf{V}^T\mathbf{y}(t))\| \leq L_f\|\mathbf{y}(t) - \mathbf{V}\mathbf{V}^T\mathbf{y}(t)\| = L_f\|\rho(t)\|$, so that

$$\|\widehat{\mathbf{r}}(t)\| \leq \alpha\|\rho(t)\| + \beta\|\mathbf{w}(t)\|, \tag{3.52}$$

where $\alpha := \|\mathbf{V}^T\mathbb{P}\|L_f$, $\beta := \|\mathbf{V}^T(\mathbf{I} - \mathbb{P})\|$. Thus, by applying the Cauchy-Schwarz inequality and triangle inequality to (3.51) and (3.52),

$$\|\theta(t)\|^2 \leq a_M(T)\Big(\alpha^2\mathcal{E}_{\mathbf{y}} + \beta^2\mathcal{E}_{\mathbf{f}}\Big),$$

for all $t \in [0, T]$, where $a_M(t) := 2\int_0^t e^{2M[\widehat{\mathbf{F}}](t-\tau)}d\tau = \begin{cases} \frac{1}{M[\widehat{\mathbf{F}}]}(e^{2M[\widehat{\mathbf{F}}]t} - 1), & M[\widehat{\mathbf{F}}] \neq 0 \\ 2t, & M[\widehat{\mathbf{F}}] = 0 \end{cases}$

and $\mathcal{E}_{\mathbf{y}} = \int_0^T \|\rho(t)\|^2 dt$, $\mathcal{E}_{\mathbf{f}} = \int_0^T \|\mathbf{w}(t)\|^2 dt$, as defined in (3.6). Finally,

$$\int_0^T \|e(t)\|^2 dt = \int_0^T \|\rho(t)\|^2 dt + \int_0^T \|\theta(t)\|^2 dt \leq \mathcal{C}\Big(\mathcal{E}_{\mathbf{y}} + \mathcal{E}_{\mathbf{f}}\Big),$$

where $\mathcal{C} = \max\{1 + a_M(T)\alpha^2 T, a_M(T)\beta^2 T\}$. When $M[\widehat{\mathbf{F}}] < 0$, $a_M(T) \leq \frac{1}{|M[\widehat{\mathbf{F}}]|}$, which is independent of T.

3.3.2 Error bounds in discretized ODE setting

Using our analysis of the full trajectory as a guide, by analogy to (3.45) and (3.46), this section shall analyze the discrete systems obtained from backward Euler time in-

tegration corresponding to the full-order system and the POD-DEIM reduced system in the form: for $Y_0 = \mathbf{y}_0$ and $\widehat{Y}_0 = \mathbf{V}^T\mathbf{y}_0$,

$$\frac{Y_j - Y_{j-1}}{\triangle t} = \mathbf{F}(t_j, Y_j), \quad \frac{\widehat{Y}_j - \widehat{Y}_{j-1}}{\triangle t} = \widehat{\mathbf{F}}(t_j, \widehat{Y}_j), \tag{3.53}$$

$\triangle t = T/n_t$, where n_t is the number of time steps, Y_j and $\widehat{Y}_j$ are approximations of $\mathbf{y}(t_j)$ and $\widehat{\mathbf{y}}(t_j)$ respectively, at $t_j = j\triangle t$, $j = 0, \ldots, n_t$. Assume that $\triangle t$ (or n_t) is chosen so that $\triangle t M[F] < 1$. Consider the error:

$$E_j = Y_j - \mathbf{V}\widehat{Y}_j,$$

where Y_j is the solution of full-order system, and $\widehat{Y}_j$ is the solution of the POD-DEIM reduced system in (3.53), for $j = 1, \ldots, n_t$. Write

$$E_j = \rho_j + \theta_j,$$

where $\rho_j := Y_j - \mathbf{V}\mathbf{V}^T Y_j$, $\theta_j := \mathbf{V}\mathbf{V}^T Y_j - \mathbf{V}\widehat{Y}_j$. Define $\widehat{\theta}_j := \mathbf{V}^T\theta_j = \mathbf{V}^T Y_j - \widehat{Y}_j$. As before, $\theta_j = \mathbf{V}\widehat{\theta}_j$, $\|\theta_j\| = \|\widehat{\theta}_j\|$ and $\rho_j^T\theta_j = 0$. From (3.53), consider

$$\begin{aligned} \frac{\widehat{\theta}_j - \widehat{\theta}_{j-1}}{\triangle t} &= \mathbf{V}^T\left(\frac{Y_j - Y_{j-1}}{\triangle t}\right) + \frac{\widehat{Y}_j - \widehat{Y}_{j-1}}{\triangle t} = \mathbf{V}^T\mathbf{F}(t_j, Y_j) + \widehat{\mathbf{F}}(t_j, \widehat{Y}_j) \\ &= \widehat{\mathbf{F}}(t_j, \mathbf{V}^T Y_j) - \widehat{\mathbf{F}}(t_j, \widehat{Y}_j) + \widehat{R}_j, \end{aligned}$$

where

$$\widehat{R}_j = \mathbf{V}^T\mathbf{F}(t_j, Y_j) - \widehat{\mathbf{F}}(t_j, \mathbf{V}^T Y_j). \tag{3.54}$$

Then,

$$\begin{aligned}
\frac{\|\widehat{\theta}_j\| - \|\widehat{\theta}_{j-1}\|}{\triangle t} &\leq \frac{1}{\triangle t}\left(\frac{\langle\widehat{\theta}_j, \widehat{\theta}_j\rangle}{\|\widehat{\theta}_j\|} - \frac{\langle\widehat{\theta}_j, \widehat{\theta}_{j-1}\rangle}{\|\widehat{\theta}_j\|}\right) \\
&= \frac{1}{\|\widehat{\theta}_j\|}\left\langle\widehat{\theta}_j, \frac{\widehat{\theta}_j - \widehat{\theta}_{j-1}}{\triangle t}\right\rangle \\
&= \frac{1}{\|\widehat{\theta}_j\|}\left\langle\widehat{\theta}_j, \widehat{\mathbf{F}}(t_j, \mathbf{V}^T Y_j) - \widehat{\mathbf{F}}(t_j, \widehat{Y}_j) + \widehat{R}_j\right\rangle \\
&= \frac{1}{\|\widehat{\theta}_j\|}\left\langle\theta_j, \widehat{\mathbf{F}}(t_j, \mathbf{V}^T Y_j) - \widehat{\mathbf{F}}(t_j, \widehat{Y}_j)\right\rangle + \frac{1}{\|\widehat{\theta}_j\|}\left\langle\widehat{\theta}_j, \widehat{R}_j\right\rangle \\
&\leq M[\widehat{\mathbf{F}}]\|\widehat{\theta}_j\| + \|\widehat{R}_j\|,
\end{aligned}$$

where the first inequality follows from $\langle\widehat{\theta}_j, \widehat{\theta}_{j-1}\rangle \leq \|\widehat{\theta}_j\|\|\widehat{\theta}_{j-1}\|$; the last equality used $\langle\widehat{\theta}_j, \widehat{\mathbf{F}}(\mathbf{V}^T Y_j) - \widehat{\mathbf{F}}(\widehat{Y}_j)\rangle \leq M[\widehat{\mathbf{F}}]\|\widehat{\theta}_j\|^2$ from (3.47); and the last inequality follows from $\langle\widehat{\theta}_j, \widehat{R}_j\rangle \leq \|\widehat{\theta}_j\|\|\widehat{R}_j\|$. That is, by using $\|\widehat{\theta}_j\| = \|\theta_j\|$, for $\zeta := \frac{1}{1-\triangle t M[\widehat{\mathbf{F}}]}$,

$$\|\theta_j\| \leq \zeta\left(\|\theta_{j-1}\| + \triangle t\|\widehat{R}_j\|\right) \leq \zeta^j\|\theta_0\| + \triangle t\sum_{\ell=1}^{j}\zeta^\ell\|\widehat{R}_{j-\ell+1}\|. \tag{3.55}$$

As in the continuous case, $\|\widehat{R}_\ell\|$ will be written as a sum of differences that can be estimated using the neglected singular values. First, consider

$$\begin{aligned}
\widehat{R}_\ell &= \mathbf{V}^T\mathbf{F}(t_\ell, Y_\ell) - \widehat{\mathbf{F}}(t_\ell, \mathbf{V}\mathbf{V}^T Y_\ell) = \mathbf{V}^T[\mathbf{F}(t_\ell, Y_\ell) - \mathbb{P}\mathbf{F}(t_\ell, \mathbf{V}\mathbf{V}^T Y_\ell)] \\
&= \mathbf{V}^T[\mathbf{F}(t_\ell, Y_\ell) - \mathbb{P}\mathbf{F}(t_\ell, Y_\ell) + \mathbb{P}\mathbf{F}(t_\ell, Y_\ell) - \mathbb{P}\mathbf{F}(t_\ell, \mathbf{V}\mathbf{V}^T Y_\ell)] \\
&= \mathbf{V}^T(\mathbf{I} - \mathbb{P})\mathbf{w}_\ell + \mathbf{V}^T\mathbb{P}(\mathbf{F}(t_\ell, Y_\ell) - \mathbf{F}(t_\ell, \mathbf{V}\mathbf{V}^T Y_\ell)),
\end{aligned}$$

where $\mathbf{w}_\ell = (\mathbf{I} - \mathbf{U}\mathbf{U}^T)\mathbf{F}(t_\ell, Y_\ell)$ from Lemma 2.2.3. The Lipschitz continuity of $\mathbf{F}$ implies $\|\mathbf{F}(t_\ell, Y_\ell) - \mathbf{F}(t_\ell, \mathbf{V}\mathbf{V}^T Y_\ell)\| \leq L_f\|Y_\ell - \mathbf{V}\mathbf{V}^T Y_\ell\| = L_f\|\rho_\ell\|$, and thus

$$\|\widehat{R}_\ell\| \leq \alpha\|\rho_\ell\| + \beta\|\mathbf{w}_\ell\|, \tag{3.56}$$

where $\alpha := \|\mathbf{V}^T\mathbb{P}\|L_f$, $\beta := \|\mathbf{V}^T(\mathbf{I}-\mathbb{P})\|$. From (3.55), since $\theta_0 = 0$, then by applying again the Cauchy-Schwarz inequality and triangle inequality, for $j = 0, \ldots, n_t$,

$$\|\theta_j\|^2 \;\leq\; (\triangle t)^2 \left(\sum_{\ell=1}^{j} \zeta^{2\ell}\right)\left(\sum_{\ell=1}^{j} \|\widehat{R}_\ell\|^2\right) \leq (\triangle t)^2\, \bar{a}_M \left(\alpha^2 \bar{\mathcal{E}}_{\mathbf{y}} + \beta^2 \bar{\mathcal{E}}_{\mathbf{f}}\right),$$

where $\bar{a}_M := 2\sum_{\ell=1}^{n_t} \zeta^{2\ell}$ and $\bar{\mathcal{E}}_{\mathbf{y}} = \sum_{\ell=1}^{j} \|\rho_\ell\|^2$, $\bar{\mathcal{E}}_{\mathbf{f}} = \sum_{\ell=1}^{j} \|\mathbf{w}_\ell\|^2$, defined earlier in (3.7). Finally, using $\sum_{\ell=0}^{n_t} \|E_\ell\|^2 = \sum_{\ell=0}^{n_t} \|\rho_\ell\|^2 + \sum_{\ell=0}^{n_t} \|\theta_\ell\|^2$ gives

$$\sum_{\ell=0}^{n_t} \|E_\ell\|^2 \;\leq\; \bar{C}\left(\bar{\mathcal{E}}_{\mathbf{y}} + \bar{\mathcal{E}}_{\mathbf{f}}\right), \tag{3.57}$$

where $\bar{C} = \max\{1+\bar{a}_M \triangle t \alpha^2 T, \bar{a}_M \triangle t \beta^2 T\}$ and for $T = n_t \triangle t$. When $M[\widehat{\mathbf{F}}] < 0$, for all $j = 1, 2, \ldots, n_t$, $q_j = \sum_{\ell=1}^{j} \zeta^{2\ell} \leq \sum_{\ell=1}^{\infty} \zeta^{2\ell} = \sum_{\ell=0}^{\infty} \zeta^{2\ell} - 1 = \frac{1}{1-\zeta^2} - 1 = \frac{1}{(1-\triangle t M[\widehat{\mathbf{F}}])^2 - 1}$. Therefore the norm of the total error $\|E_j\|$ is uniformly bounded on $[0, T]$ as shown below:

$$\|E_\ell\|^2 \;=\; \|\rho_\ell\|^2 + \|\theta_\ell\|^2 \;\leq \bar{c}\left(\sum_{\ell=k+1}^{r} \lambda_\ell + \sum_{\ell=m+1}^{r_s} s_\ell\right), \tag{3.58}$$

where $\bar{c} = \max\{1 + \bar{q}\alpha^2, \bar{q}\beta^2\}$, $\bar{q} = \frac{1}{|M[\widehat{\mathbf{F}}]| + \triangle t M[\widehat{\mathbf{F}}]^2/2}$.

The following theorem summarizes the results of error bounds for POD-DEIM solutions which are derived in this section through the application of logarithmic Lipschitz constant $M[\cdot]$.

Theorem 3.3.1 *Let $\mathbf{y}(t)$ be the solution of the original full-order system (3.45) and $\widehat{\mathbf{y}}(t)$ be the solution of the POD-DEIM reduced system (3.46), for $t \in [0, T]$. Let Y_j and $\widehat{Y}_j$ be the solutions of the discretized systems of (3.45) and (3.46), respectively,*

obtained from implicit Euler time integration at $t_j = j\triangle t \in [0,T]$, $\triangle t = T/n_t$ *for* $j = 0, \ldots n_t$. *Let* $M[\widehat{\mathbf{F}}]$ *be the logarithmic Lipschitz constant of* $\widehat{\mathbf{F}}$ *defined as in (3.47) and assume that* $\mathbf{F}(t,\mathbf{y})$ *in (3.45) is Lipschitz continuous with Lipschitz constant* L_f *as in (3.2). Assume also that* $\triangle t$ *(or* n_t*) is chosen so that* $\triangle t M[F] < 1$. *Then*

$$\int_0^T \|\mathbf{y}(t) - \mathbf{V}\widehat{\mathbf{y}}(t)\|^2 dt \;\leq\; \mathcal{C}\left(\mathcal{E}_{\mathbf{y}} + \mathcal{E}_{\mathbf{f}}\right), \tag{3.59}$$

$$\sum_{j=0}^{n_t} \|Y_j - \mathbf{V}\widehat{Y}_j\|^2 \;\leq\; \bar{\mathcal{C}}\left(\bar{\mathcal{E}}_{\mathbf{y}} + \bar{\mathcal{E}}_{\mathbf{f}}\right), \tag{3.60}$$

where $\mathcal{C} := \max\{1 + c_M\alpha^2 T, c_M\beta^2 T\}$ *and* $\bar{\mathcal{C}} := \max\{1 + \bar{c}_M\alpha^2 T, \bar{c}_M\beta^2 T\}$,

$$\alpha \;:=\; \|\mathbf{V}^T\mathbb{P}\|L_f, \quad \beta := \|\mathbf{V}^T(\mathbf{I} - \mathbb{P})\|, \quad \zeta := \frac{1}{1 - \triangle t M[\widehat{\mathbf{F}}]}, \tag{3.61}$$

$$c_M \;:=\; \begin{cases} \frac{e^{2M[\widehat{\mathbf{F}}]T} - 1}{M[\widehat{\mathbf{F}}]}, & M[\widehat{\mathbf{F}}] \neq 0 \\ 2T, & M[\widehat{\mathbf{F}}] = 0 \end{cases}, \bar{c}_M := \triangle t\zeta^2\left(\frac{1 - \zeta^{2n_t}}{1 - \zeta^2}\right), \tag{3.62}$$

and $\mathcal{E}_{\mathbf{y}}, \mathcal{E}_{\mathbf{f}}, \bar{\mathcal{E}}_{\mathbf{y}}, \bar{\mathcal{E}}_{\mathbf{f}}$ *are defined as in (3.6) and (3.7).*

Remark 3.3.2 *Using the notation and assumptions from Theorem 3.3.1:*

(i) If $M[\widehat{\mathbf{F}}] < 0$, *then* c_M *and* $\bar{c}_M$ *in (3.62) are bounded by*

$$c_M \;<\; \frac{1}{|M[\widehat{\mathbf{F}}]|}, \;\; \textit{and} \;\; \bar{c}_M < \frac{1}{|M[\widehat{\mathbf{F}}]| + \triangle t M[\widehat{\mathbf{F}}]^2/2}. \tag{3.63}$$

(ii) When the POD basis matrices $\mathbf{V} \in \mathbb{R}^{n\times k}$ *and* $\mathbf{U} \in \mathbb{R}^{n\times m}$ *used in (3.46), respectively, satisfy (3.9) and (3.10), then* $\mathcal{E}_{\mathbf{y}} = \sum_{\ell=k+1}^{r} \lambda_\ell^\infty, \mathcal{E}_{\mathbf{f}} = \sum_{\ell=m+1}^{r_s} s_\ell^\infty$. *In this case, if also* $M[\widehat{\mathbf{F}}] < 0$, *then from (i),*

$$\int_0^T \|\mathbf{y}(t) - \mathbf{V}\widehat{\mathbf{y}}(t)\|^2 dt \;\leq\; \mathcal{C}_o\left(\sum_{\ell=k+1}^{r} \lambda_\ell^\infty + \sum_{\ell=m+1}^{r_s} s_\ell^\infty\right), \tag{3.64}$$

where $\mathcal{C}_o := \max\{1 + \alpha^2 T/|M[\widehat{\mathbf{F}}]|, \beta^2 T/|M[\widehat{\mathbf{F}}]|\}$.

(iii) Analogously, when $\mathbf{V} \in \mathbb{R}^{n\times k}$ *and* $\mathbf{U} \in \mathbb{R}^{n\times m}$ *used in (3.46) are the POD basis matrices of* $\mathbb{Y} = [Y_1, \ldots, Y_{n_t}]$ *and* $\mathbb{F} = [\mathbf{F}(t_1, Y_1), \ldots, \mathbf{F}(t_{n_t}, Y_{n_t})] \in \mathbb{R}^{n\times n_t}$*, then using (3.11) and (3.12) gives* $\bar{\mathcal{E}}_{\mathbf{y}} = \sum_{\ell=k+1}^{\bar{r}} \lambda_\ell, \bar{\mathcal{E}}_{\mathbf{f}} = \sum_{\ell=m+1}^{\bar{r}_s} s_\ell$*. In this case, if, also* $M[\widehat{\mathbf{F}}] < 0$*, then from (i),*

$$\sum_{j=0}^{n_t} \|Y_j - \mathbf{V}\widehat{Y}_j\|^2 \;\leq\; \bar{C}_o \left(\sum_{\ell=k+1}^{\bar{r}} \lambda_\ell + \sum_{\ell=m+1}^{\bar{r}_s} s_\ell \right), \tag{3.65}$$

where $\bar{C}_o := \max\{1 + \bar{q}\alpha^2 T, \bar{q}\beta^2 T\}$, $\bar{q} = \frac{1}{|M[\widehat{\mathbf{F}}]| + \triangle t M[\widehat{\mathbf{F}}]^2/2}$.

The bounds for pointwise errors can be obtained similarly and are given below.

Remark 3.3.3 *Using the notation and assumptions from Theorem 3.3.1:*

When (i) and (iii) of Remark 3.3.2 hold true, the norm of the pointwise error in the discrete setting is uniformly bounded at each time step:

$$\|Y_\ell - \mathbf{V}\widehat{Y}_\ell\|^2 \;\leq\; \bar{c} \left(\sum_{\ell=k+1}^{\bar{r}} \lambda_\ell + \sum_{\ell=m+1}^{\bar{r}_s} s_\ell \right), \quad \textit{for all } \ell = 1, \ldots n_t, \tag{3.66}$$

where $\bar{c} = \max\{1 + \bar{q}\alpha^2, \bar{q}\beta^2\}$, $\bar{q} = \frac{1}{|M[\widehat{\mathbf{F}}]| + \triangle t M[\widehat{\mathbf{F}}]^2/2}$.

Notice that, for $M[\mathbf{F}] < 0$, the error bound (3.60) in the discretized setting converges to the bound (3.59) in the continuous setting. In particular, as $\triangle t \to 0$, it was shown in [50] that $\bar{\mathcal{E}}_{\mathbf{y}}$ and $\bar{\mathcal{E}}_{\mathbf{f}}$ converge to $\mathcal{E}_{\mathbf{y}}$ and $\mathcal{E}_{\mathbf{f}}$, respectively; and from (3.63), we have that the bound for $\bar{c}_M$ converges to the bound for c_M.

Notice also that there are two main differences for the error bounds in the continuous setting from (3.36) of Theorem 3.2.1 and from (3.59) of Theorem 3.3.1: one in the quantities $\mu(\cdot)$ and $M[\cdot]$; and the other in the terms c_μ and c_M. Note that

$\mu(\cdot)$ and $M[\cdot]$ are the same when they are applied to linear operators, and hence there is no need to introduce the notion of logarithmic Lipschitz constant for linear systems. With nonlinearities, however, applying the logarithmic Lipschitz constant $M[\cdot]$ will allow us to avoid using Gronwall's inequality, as required in the standard approach for deriving error bounds, which often gives pessimistic bounds with exponential growth, e.g. the term c_μ in (3.41) has the exponential part, $e^{2\gamma b_\mu}$, arising from applying Gronwall's inequality in (3.23), while c_M in (3.63) does not.

The derivations of the error bounds presented in this chapter provide weighting coefficients of the least-squares errors $\mathcal{E}_{\mathbf{y}}$, $\mathcal{E}_{\mathbf{f}}$ (or $\bar{\mathcal{E}}_{\mathbf{y}}$, $\bar{\mathcal{E}}_{\mathbf{f}}$ in discrete cases) for the solution snapshots and the nonlinear snapshots, which further imply the contributions of the error from POD and DEIM in the overall approximation. These bounds clearly explain the *stagnation* of the errors as observed in the numerical results shown in Chapters 4 and 5 (see e.g. Fig. 4.4 and Fig. 4.9). Moreover, for some simple problems, these bounds can be used for determining a *suitable* dimension (k, m) for the POD-DEIM approximation. Appendix B illustrates an application of the error estimates given in this chapter.

3.4 Conclusion

This chapter derived the error bounds of the state approximations from the POD-DEIM reduced systems for the ODEs with Lipschitz continuous nonlinearities. The analysis was considered in the continuous setting where the availability of the solu-

tions was assumed on the entire time interval and the overall accuracy of the reduced system was only contributed from applying the POD-DEIM technique. A framework for error analysis was given in the discrete setting for the implicit Euler time integration scheme, which can be extended to other numerical methods. The proposed error bounds in both continuous and discrete settings were derived through a standard approach using logarithmic norms, as well as through an application of generalized logarithmic norms [81]. The conditions under which the reduction error is uniformly bounded were also discussed. The resulting error bounds in the 2-norm reflect the approximation property of the POD based scheme through the decay of the corresponding singular values.

The next chapter will demonstrate the applications of the POD-DEIM model reduction technique through some numerical examples.

Chapter 4

Model Problems/Numerical Examples

This chapter illustrates how to apply the Proper Orthogonal Decomposition (POD) with the Discrete Empirical Interpolation Method (DEIM) introduced in Chapter 2 to nonlinear systems from finite difference (FD) discretizations of two problems. The first is a nonlinear 1-D PDE arising in neuron modeling. The second is a nonlinear 2-D steady state problem whose solution is obtained by solving its FD discretized system by using Newton's method. In both experiments, computation time was reduced roughly by a factor of 100. A more complex numerical result will be considered in the next chapter through the application of two-phase miscible flow in porous media.

4.1 The FitzHugh-Nagumo (F-N) System

The FitzHugh-Nagumo system is used in neuron modeling. It is a simplified version of the Hodgkin-Huxley model, which describes in a detailed manner activation and deactivation dynamics of a spiking neuron [76, 23]. This system [23] is given by (4.1)–(4.4). For $x \in [0, L], t \geq 0$,

$$\varepsilon v_t(x,t) = \varepsilon^2 v_{xx}(x,t) + f(v(x,t)) - w(x,t) + c, \tag{4.1}$$

$$w_t(x,t) = bv(x,t) - \gamma w(x,t) + c, \tag{4.2}$$

with nonlinear function $f(v) = v(v-0.1)(1-v)$. The initial and boundary conditions are:

$$v(x,0) = 0, \quad w(x,0) = 0, \quad x \in [0, L], \tag{4.3}$$

$$v_x(0,t) = -i_0(t), \quad v_x(L,t) = 0, \quad t \geq 0, \tag{4.4}$$

where the parameters are given by $L = 1$, $\varepsilon = 0.015$, $b = 0.5$, $\gamma = 2$, $c = 0.05$. The stimulus is $i_0(t) = 50000t^3 \exp(-15t)$. The variables v and w are voltage and recovery of voltage, respectively. Note that this is not a scalar equation and requires a slight generalization of the problem setting discussed earlier in Chapter 2. However, the FD discretization does indeed yield a system of ODEs of the same form as (2.1), as shown next.

4.1.1 Full Order Model of FD Discretized System

For illustration purposes, the central FD discretization in the spatial variable with forward Euler time integration scheme is used in this section to construct a discretized system of the PDE in (4.1) and (4.2). Consider first the discretization of the spatial domain $x_i = i\triangle x$ for $i = 0, 1, \ldots, n+1$ with $x_0 = 0$ and $x_{n+1} = L$ and the discretization of the time domain $t_j = j\triangle t$ for $j = 0, 1, \ldots$, where $\triangle x$ is the spatial stepsize and $\triangle t$ is the time stepsize. Let v_i^j and w_i^j denote the solution of the discretized system at the mesh point (x_i, t_j) of $v(x_i, t_j)$ and $w(x_i, t_j)$, respectively. For $i = 0, \ldots, n+1$, and $j = 0, 1, \ldots,$

$$\varepsilon\left(\frac{v_i^{j+1} - v_i^j}{\triangle t}\right) = \varepsilon^2\left(\frac{v_{i-1}^j - 2v_i^j + v_{i+1}^j}{(\triangle x)^2}\right) + f(v_i^j) - w_i^j + c \tag{4.5}$$

$$\frac{w_i^{j+1} - w_i^j}{\triangle t} = bv_i^j - \gamma w_i^j + c, \tag{4.6}$$

with initial conditions: $v_i^0 = 0$ and $w_i^0 = 0$ for all $i = 1, \ldots, n+1$, and the boundary conditions: $\frac{v_1^j - v_0^j}{\triangle x} = -i_0(t_j) \Rightarrow v_0^j = v_1^j + \triangle x\, i_0(t_j)$, and $\frac{v_{n+1}^j - v_n^j}{\triangle x} = 0 \Rightarrow v_{n+1}^j = v_n^j$ for $j = 0, 1, \ldots$. That is,

$$\frac{v_0^j - 2v_1^j + v_2^j}{\triangle x^2} = \frac{(v_1^j + \triangle x\, i_0(t_j)) - 2v_1^j + v_2^j}{(\triangle x)^2} = \frac{-v_1^j + v_2^j}{(\triangle x)^2} + \frac{i_0(t_j)}{\triangle x}, \tag{4.7}$$

$$\frac{v_{n-1}^j - 2v_n^j + v_{n+1}^j}{(\triangle x)^2} = \frac{v_{n-1}^j - 2v_n^j + v_n^j}{(\triangle x)^2} = \frac{v_{n-1}^j - v_n^j}{(\triangle x)^2}. \tag{4.8}$$

Let $\mathbf{v}^j = [v_1^j, \ldots, v_n^j]^T \in \mathbb{R}^n$, $\mathbf{w}^j = [w_1^j, \ldots, w_n^j]^T \in \mathbb{R}^n$, and $\mathbf{y}^j = \begin{bmatrix} \mathbf{v}^j \\ \mathbf{w}^j \end{bmatrix} \in \mathbb{R}^{2n}$.

Then, the full-order FD system is of the form: for $j = 0, 1, 2, \ldots,$

$$\mathbf{E}\frac{1}{\triangle t}(\mathbf{y}^{j+1} - \mathbf{y}^j) = \mathbf{A}\mathbf{y}^j + \mathbf{g}(t_j) + \mathbf{F}(\mathbf{y}^j) \quad \text{and} \quad \mathbf{y}^0 = 0, \tag{4.9}$$

$$\mathbf{E} = \begin{bmatrix} \varepsilon\mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_n \end{bmatrix} \in \mathbb{R}^{2n\times 2n}, \quad \mathbf{A} = \begin{bmatrix} -\frac{\varepsilon^2}{\triangle x^2}\mathbf{K} & -\mathbf{I}_n, \\ b\mathbf{I}_n & -\gamma\mathbf{I}_n \end{bmatrix} \in \mathbb{R}^{2n\times 2n}$$

$$\mathbf{K} = \begin{pmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 1 \end{pmatrix} \in \mathbb{R}^{n\times n}, \qquad \mathbf{I}_n \in \mathbb{R}^{n\times n} = \text{identity matrix},$$

$$\mathbf{g}(t) = \frac{\varepsilon^2}{\triangle x}\begin{bmatrix} \mathbf{g}_0(t) \\ \mathbf{0} \end{bmatrix} + \mathbf{c} \in \mathbb{R}^{2n}, \quad \text{with } \mathbf{g}_0(t) = \begin{bmatrix} i_0(t) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^n, \mathbf{c} = c\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^{2n},$$

$$\mathbf{F}(\mathbf{y}^j) = \begin{bmatrix} f(\mathbf{v}^j) \\ \mathbf{0} \end{bmatrix} \in \mathbb{R}^{2n}, \text{ with } f(\mathbf{v}^j) = \begin{bmatrix} f(v_1^j) \\ \vdots \\ f(v_n^j) \end{bmatrix} \in \mathbb{R}^n, \quad j = 0, 1, 2, \ldots.$$

4.1.2 A POD-Galerkin Reduced Order Model

The POD basis used for constructing a reduced-order system can be computed from a given set of *snapshots*. In this setting, a *snapshot* is defined as the numerical solution of (4.1)-(4.4) at a particular time t. Consider a set of n_s snapshots at times $t_1, \ldots, t_{n_s}$. Let $\mathbf{v}^\ell$ and $\mathbf{w}^\ell$ be the ℓ^{th} snapshots from (4.9) at time t_ℓ. Define snapshot matrices:

$$\mathbb{V} = \begin{bmatrix} | & & | \\ \mathbf{v}^1 & \ldots & \mathbf{v}^{n_s} \\ | & & | \end{bmatrix} \in \mathbb{R}^{n \times n_s}, \quad \mathbb{W} = \begin{bmatrix} | & & | \\ \mathbf{w}^1 & \ldots & \mathbf{w}^{n_s} \\ | & & | \end{bmatrix} \in \mathbb{R}^{n \times n_s}. \tag{4.10}$$

Let $r = \min\{\text{rank}(\mathbb{V}), \text{rank}(\mathbb{W})\}$. The POD basis matrix of dimension $k \leq r$, denoted by $\mathbf{U}^v \in \mathbb{R}^{n \times k}$, for the snapshots $\{\mathbf{v}^\ell\}_{\ell=1}^{n_s}$ is formed by k left singular vectors of $\mathbb{V}$ corresponding to the first k largest singular values of $\mathbb{V}$ (and similarly for the POD basis matrix, denoted by $\mathbf{U}^w \in \mathbb{R}^{n \times k}$, for the snapshots $\{\mathbf{w}_\ell\}_{\ell=1}^{n_s}$). Define

$$\mathbf{U} = \begin{bmatrix} \mathbf{U}^v & \mathbf{0} \\ \mathbf{0} & \mathbf{U}^w \end{bmatrix} \in \mathbb{R}^{2n \times 2k}. \tag{4.11}$$

The reduced-order system of the discretized FD is obtained by projecting the system and the solution onto the range of $\mathbf{U}$. By replacing $\mathbf{y}^j$ in the full-order system with $\mathbf{U}\widehat{\mathbf{y}}^j$, $\widehat{\mathbf{y}}^j \in \mathbb{R}^{2k}$, and applying the Galerkin projection, the reduced system is of the form

$$\underbrace{\mathbf{U}^T\mathbf{E}\mathbf{U}}_{\widehat{\mathbf{E}}} \frac{1}{\triangle t}(\widehat{\mathbf{y}}^{j+1} - \widehat{\mathbf{y}}^j) = \underbrace{\mathbf{U}^T\mathbf{A}\mathbf{U}}_{\widehat{\mathbf{A}}}\widehat{\mathbf{y}}^j + \underbrace{\mathbf{U}^T\mathbf{g}(t_j)}_{\widehat{\mathbf{g}}(t_j)} + \underbrace{\mathbf{U}^T\mathbf{F}(\mathbf{U}\widehat{\mathbf{y}}^j)}_{\widehat{\mathbf{F}}(\widehat{\mathbf{y}}^j)} \quad ; \quad \widehat{\mathbf{y}}^0 = 0. \tag{4.12}$$

The resulting POD reduced system is given by

$$\widehat{\mathbf{E}}\frac{1}{\triangle t}(\widehat{\mathbf{y}}^{j+1} - \widehat{\mathbf{y}}^j) = \widehat{\mathbf{A}}\widehat{\mathbf{y}}^j + \widehat{\mathbf{g}}(t_j) + \widehat{\mathbf{F}}(\widehat{\mathbf{y}}^j) \quad \text{and} \quad \widehat{\mathbf{y}}^0 = 0, \tag{4.13}$$

where $\widehat{\mathbf{E}} = \mathbf{U}^T\mathbf{E}\mathbf{U} = \begin{bmatrix} \varepsilon\mathbf{I}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_k \end{bmatrix} \in \mathbb{R}^{2k\times 2k}$, $\mathbf{I}_k \in \mathbb{R}^{k\times k}$ is the identity matrix; $\widehat{\mathbf{A}} = \mathbf{U}^T\mathbf{A}\mathbf{U}$; $\widehat{\mathbf{g}}(t_j) = \mathbf{U}^T\mathbf{g}(t_j)$; and $\widehat{\mathbf{F}}(\widehat{\mathbf{y}}^j) = \mathbf{U}^T\mathbf{F}(\mathbf{U}\widehat{\mathbf{y}}^j)$.

Notice that, although the equation in (4.13) is expressed in the expansion of the reduced (POD) basis, the complexity in computing the nonlinear term still depends on the dimension n of the full FD system. In particular, the nonlinear term is $\widehat{\mathbf{F}}(\widehat{\mathbf{y}}^j) = \begin{bmatrix} \widehat{\mathbf{F}}^v(\widehat{\mathbf{v}}^j) \\ \mathbf{0} \end{bmatrix}$, where $\widehat{\mathbf{F}}^v(\widehat{\mathbf{v}}^j)$ is of the form

$$\widehat{\mathbf{F}}^v(\widehat{\mathbf{v}}^j) = \underbrace{(\mathbf{U}^v)^T}_{k\times n}\underbrace{f(\mathbf{U}^v\widehat{\mathbf{v}}^j)}_{n\times 1} \in \mathbb{R}^k. \tag{4.14}$$

As discussed in Chapter 2, the problem here is that $f(\mathbf{U}^v\widehat{\mathbf{v}}^j)$ cannot be precomputed, since it depends on the unknown vector $\widehat{\mathbf{v}}^j$. DEIM will be applied to (4.14), as shown next.

4.1.3 Reduced-Order Model from POD-DEIM Method

The (on-line) dependence on the dimension of the full FD discretized system in (4.13) can be removed by using DEIM as described in Section 2.2 in Chapter 2. The POD basis of the nonlinear snapshots will be used as an input basis for the DEIM algorithm (see Algorithm 1). The POD basis of the nonlinear snapshots is constructed from the solutions of the full FD system as follows. Let $\{\mathbf{v}^1, \ldots, \mathbf{v}^{n_s}\}$ be a set of solutions from the full FD system (4.9) and recall that the nonlinear function $f(\mathbf{v}^\ell)$ is evaluated at

$\mathbf{v}^\ell$ componentwise, for $\ell = 1, \ldots, n_s$. Define

$$\mathbb{F} = \begin{bmatrix} | & & | \\ f(\mathbf{v}^1) & \cdots & f(\mathbf{v}^{n_s}) \\ | & & | \end{bmatrix} \in \mathbb{R}^{n \times n_s}. \tag{4.15}$$

The POD basis matrix of dimension $m \leq \text{rank}(\mathbb{F})$, denoted by $\mathbf{U}^f \in \mathbb{R}^{n \times m}$, for the snapshots $\{f(\mathbf{v}^\ell)\}_{\ell=1}^{n_s}$, is the matrix consisting of left singular vectors of $\mathbb{F}$ corresponding to the first m largest singular values. With input basis vectors from $\mathbf{U}^f$, Algorithm 1 in Chapter 2 for DEIM is then used to generate interpolation indices $\hat{\wp} = [\wp_1, \ldots, \wp_m]^T$ for constructing matrix $\mathbf{P}$ defined in (2.13). The DEIM approximation is then

$$f(\mathbf{U}^v \widehat{\mathbf{v}}) \simeq \mathbf{U}^f (\mathbf{P}^T \mathbf{U}^f)^{-1} \mathbf{P}^T f(\mathbf{U}^v \widehat{\mathbf{v}}) = \mathbf{U}^f (\mathbf{P}^T \mathbf{U}^f)^{-1} f(\underbrace{\mathbf{P}^T \mathbf{U}^v}_{\mathbf{D}} \widehat{\mathbf{v}}), \tag{4.16}$$

where the last equality follows from the fact that f is a componentwise evaluation function. Note that $\mathbf{D} := \mathbf{P}^T \mathbf{U}^v \in \mathbb{R}^{m \times k}$ can be precomputed by selecting the rows $\wp_1, \ldots, \wp_m$ of $\mathbf{U}^v$. Hence, the nonlinear term (4.14) is approximated by

$$\widehat{\mathbf{F}}^v(\widehat{\mathbf{v}}) \simeq \underbrace{(\mathbf{U}^v)^T \mathbf{U}^f (\mathbf{P}^T \mathbf{U}^f)^{-1}}_{\mathbf{C}:\ k \times m} \underbrace{f(\mathbf{D}\widehat{\mathbf{v}})}_{m \times 1} = \mathbf{C}\, f(\mathbf{D}\widehat{\mathbf{v}}), \tag{4.17}$$

where $\mathbf{C} := (\mathbf{U}^v)^T \mathbf{U}^f (\mathbf{P}^T \mathbf{U}^f)^{-1} \in \mathbb{R}^{k \times m}$ can be precomputed so that there is no dependence on dimension of original FD system. Finally, from (4.13), the approximate DEIM reduced system is given by

$$\widehat{\mathbf{E}} \frac{1}{\triangle t} (\widehat{\mathbf{y}}^{j+1} - \widehat{\mathbf{y}}^j) = \widehat{\mathbf{A}} \widehat{\mathbf{y}}^j + \widehat{\mathbf{g}}(t_j) + \begin{bmatrix} \mathbf{C}\, f(\mathbf{D}\widehat{\mathbf{v}}^j) \\ \mathbf{0} \end{bmatrix} \quad \text{and} \quad \widehat{\mathbf{y}}^0 = 0, \tag{4.18}$$

where $\widehat{\mathbf{E}}$, $\widehat{\mathbf{A}}$, $\widehat{\mathbf{g}}(t)$ are defined as in (4.13); $\mathbf{C}$, $\mathbf{D}$ are defined as in (4.17); and $f(\mathbf{D}\widehat{\mathbf{v}}^j) \in \mathbb{R}^m$ is evaluated componentwise at m entries of $\mathbf{D}\widehat{\mathbf{v}}^j \in \mathbb{R}^m$.

4.1.4 Numerical Results

The dimension of the full-order FD system is 1024. The POD basis vectors are constructed from 100 snapshot solutions obtained from the solutions of the full-order FD system at equally-spaced time steps in the interval $[0, 8]$.

Figure 4.2 shows the fast decay around the first 40 singular values of the snapshot solutions for v, w, and the nonlinear snapshots $f(v)$. The plots of the numerical solutions for v and w are presented in Figure 4.1. This system has a limit cycle for each spatial variable x. The solutions v and w are therefore illustrated through plots of a phase-space diagram in Figure 4.3 for the solutions of the full-order system and the POD-DEIM reduced system using both POD and DEIM of dimension 5. From the figure, this reduced-order system captures the limit cycle of the original full-order system very well. The average relative errors of the solutions of the reduced systems and the average CPU time (scaled with the CPU time from sparse full-order system) for each time step from different dimensions of POD and DEIM are presented in Figure 4.4.

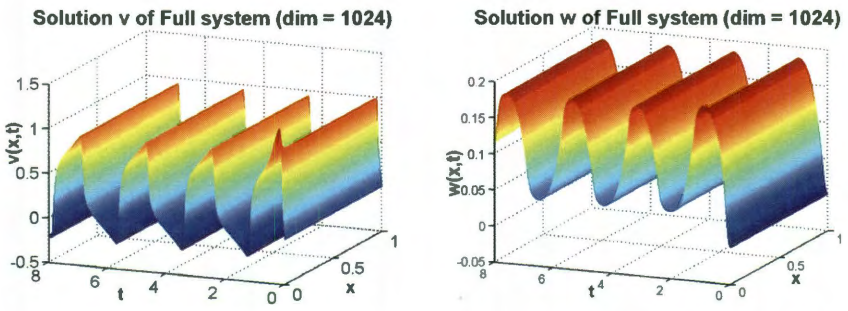


Figure 4.1: Numerical solutions v and w from the original FD system (dim 1024) of F-N system (4.1)–(4.4).

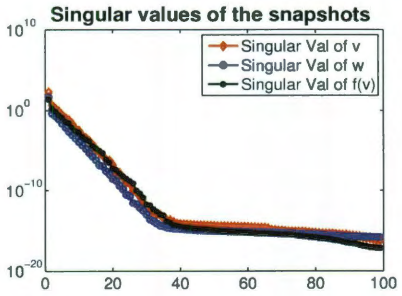


Figure 4.2: The singular values of the 100 snapshot solutions for v, w, and $f(v)$ from the full-order FD discretization of the F-N system.

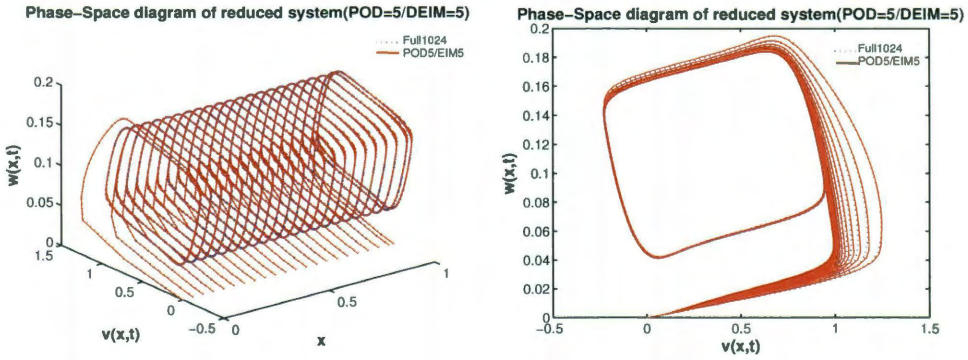


Figure 4.3: Left: Phase-space diagram of v and w at different spatial points x from the FD system (dim 1024) and the *POD-DEIM* reduced systems (dim 5). Right: Corresponding projection of the solutions at different values of x onto the v-w plane.

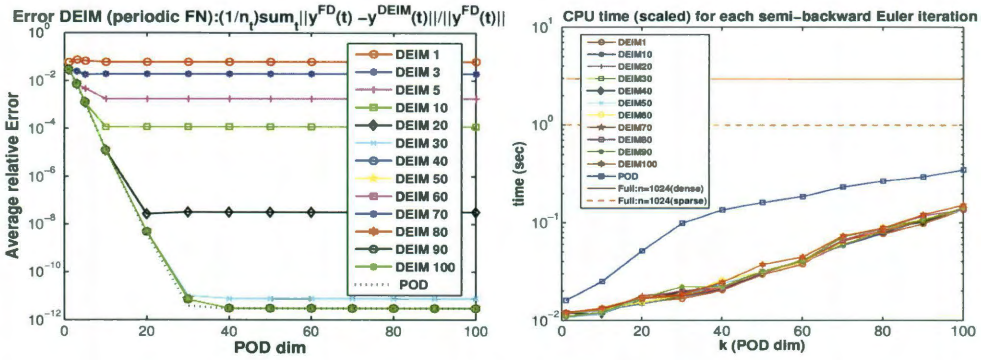


Figure 4.4: Left: Average relative errors from the POD-DEIM reduced system (solid lines) and from POD reduced systems (dashed line) for the F-N system. Once the dimension of DEIM reaches 40, the approximation errors from the POD-DEIM and POD reduced systems are indistinguishable. Right: Average online CPU time (scaled with the CPU time of the full-sparse system) in each time step of semi-implicit Euler method.

4.2 A Nonlinear 2-D Steady State Problem

This section illustrates an application of the POD-DEIM method to a nonlinear parametrized PDE in a 2-D spatial domain (from [38]):

$$-\nabla^2 u(x,y) + s(u(x,y);\mu) = 100\sin(2\pi x)\sin(2\pi y), \tag{4.19}$$

$$s(u;\mu) = \frac{\mu_1}{\mu_2}(e^{\mu_2 u} - 1), \tag{4.20}$$

where the spatial variables $(x,y) \in \Omega = (0,1)^2$ and the parameters are $\mu = (\mu_1, \mu_2) \in \mathscr{D} = [0.01, 10]^2 \subset \mathbb{R}^2$, with a homogeneous Dirichlet boundary condition.

4.2.1 Model Reduction of the FD Discretized System

Central finite differences will be used to construct a spatial discretization of the steady state equations, then Newton's method will be applied to solve for the solution at each given pair of parameter $\mu = (\mu_1, \mu_2)$.

Let $0 = x_0 < x_1 < \cdots < x_{n_x} < x_{n_x+1} = 1$ and $0 = y_0 < y_1 < \cdots < y_{n_y} < y_{n_y+1} = 1$ be equally spaced points on the x-axis and y-axis for generating the grid points on the domain Ω, and let $n := n_x n_y$ be the dimension of the discretized full-order system. Let u_{ij} denote an approximation of the solution $u(x_i, y_j)$ for $i = 1, \ldots, n_x$, $j = 1, \ldots, n_y$ and let $\triangle x = 1/(n_x + 1)$, $\triangle y = 1/(n_y + 1)$, so that the standard central finite difference approximation gives

$$\nabla^2 u \approx \frac{u_{i-1,j} - 2u_{ij} + u_{i+1,j}}{(\triangle x)^2} + \frac{u_{i,j-1} - 2u_{ij} + u_{i,j+1}}{(\triangle y)^2}.$$

Define $\mathbf{u} = [\underbrace{u_{11}, u_{21}, \ldots, u_{n_x 1}}_{y=y_1}, \underbrace{u_{12}, u_{22}, \ldots, u_{n_x 2}}_{y=y_2}, \ldots, \underbrace{u_{1n_y}, u_{2n_y}, \ldots, u_{n_x n_y}}_{y=y_{n_y}}]^T \in \mathbb{R}^n$ to be the unknown vector. By using homogeneous Dirichlet boundary conditions, the discretized system can be written in the form

$$\mathbf{b} + \mathbf{A}\mathbf{u} + \mathbf{F}(\mathbf{u}; \mu) = 0, \tag{4.21}$$

where $\mathbf{F}(\mathbf{u}; \mu) = s(\mathbf{u}; \mu)$ with s evaluated componentwise at the entries of $\mathbf{u}$ and $\mathbf{b} = 100 \sin(2\pi X) \sin(2\pi Y) \in \mathbb{R}^n$, $X = [\underbrace{x_1, x_2, \ldots, x_{n_x}}, \ldots, \underbrace{x_1, x_2, \ldots, x_{n_x}}]^T \in \mathbb{R}^n$, $Y = [\underbrace{y_1, y_1, \ldots, y_1}_{n_x}, \ldots, \underbrace{y_{n_y}, y_{n_y}, \ldots, y_{n_y}}_{n_x}]^T \in \mathbb{R}^n$, with $\mathbf{b}$ evaluated componentwise at

the vectors X and Y, and

$$\mathbf{A} = -\begin{pmatrix} \mathbf{E} & \mathbf{B} & & & & \\ \mathbf{B} & \mathbf{E} & \mathbf{B} & & & \\ & \ddots & \ddots & \ddots & & \\ & & & \mathbf{B} & \mathbf{E} & \mathbf{B} \\ & & & & \mathbf{B} & \mathbf{E} \end{pmatrix} \in \mathbb{R}^{n\times n},$$

$$\text{with} \quad \mathbf{E} = \begin{pmatrix} \alpha & \beta & & & & \\ \beta & \alpha & \beta & & & \\ & \ddots & \ddots & \ddots & & \\ & & \beta & \alpha & \beta & \\ & & & \beta & \alpha \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} \gamma & & & \\ & \ddots & & \\ & & \gamma & \\ & & & \gamma \end{pmatrix} \in \mathbb{R}^{n_x\times n_x},$$

$$\text{for} \quad \alpha = -\frac{2}{(\triangle x)^2} - \frac{2}{(\triangle y)^2}, \beta = \frac{1}{(\triangle x)^2}, \gamma = \frac{1}{(\triangle y)^2}.$$

Notice that the system (4.21) is in a similar form as the steady state parametrized system given in (2.2) of Chapter 2, and hence the construction of POD and POD-DEIM reduced systems discussed earlier can be applied to this problem and will not be repeated the details here. The full-order system, the POD reduced system of dimension k, and the POD-DEIM reduced system of dimension (k, m), $k, m \ll n$, can be written as:

$$\begin{aligned} &\text{Full:} && \mathbf{G}(\mathbf{u}) := \mathbf{b} + \mathbf{A}\mathbf{u} + \mathbf{F}(\mathbf{u};\mu) = 0; \\ &\text{POD:} && \widetilde{\mathbf{G}}(\widehat{\mathbf{u}}) := \widehat{\mathbf{b}} + \widehat{\mathbf{A}}\widehat{\mathbf{u}} + \mathbf{V}^T\mathbf{F}(\mathbf{V}\widehat{\mathbf{u}};\mu) = 0; \quad \widehat{\mathbf{A}} = \mathbf{V}^T\mathbf{A}\mathbf{V}, \widehat{\mathbf{b}} = \mathbf{V}^T\mathbf{b}, \\ &\text{POD-DEIM:} && \widehat{\mathbf{G}}(\widehat{\mathbf{u}}) := \widehat{\mathbf{b}} + \widehat{\mathbf{A}}\widehat{\mathbf{u}} + \mathbf{B}\mathbf{F}(\mathbf{V}_{\wp}\widehat{\mathbf{u}};\mu) = 0; \quad \mathbf{B} = \mathbf{V}^T\mathbf{U}(\mathbf{P}^T\mathbf{U})^{-1}, \mathbf{V}_{\wp} = \mathbf{P}^T\mathbf{V}, \end{aligned} \tag{4.22}$$

where $\mathbf{V} \in \mathbb{R}^{n\times k}$ and $\mathbf{U} \in \mathbb{R}^{n\times m}$ are the POD basis matrices for the solution snapshots $\{\mathbf{u}(\mu^j)\}_{j=1}^{n_s}$ and nonlinear snapshots $\{\mathbf{F}(\mathbf{u}(\mu^j);\mu^j)\}_{j=1}^{n_s}$, respectively, with n_s sampled parameters $\{\mu^j = (\mu_1^j, \mu_2^j)\}_{j=1}^{n_s}$. The matrices $\widehat{\mathbf{A}} \in \mathbb{R}^{k\times k}$, $\widehat{\mathbf{b}} \in \mathbb{R}^k$, $\mathbf{B} \in \mathbb{R}^{k\times m}$, $\mathbf{V}_{\wp} \in \mathbb{R}^{m\times k}$ can be pre-computed, stored, and re-used. To solve the full-order system $\mathbf{G}(\mathbf{u}) = 0$ for $\mathbf{u}$ and the reduced systems $\widetilde{\mathbf{G}}(\widehat{\mathbf{u}}) = 0$, $\widehat{\mathbf{G}}(\widehat{\mathbf{u}}) = 0$ for $\widehat{\mathbf{u}}$, Newton's method will be used, and the iteration updates are given by

$$\begin{aligned}
&\text{Full:} && \mathbf{u} \leftarrow \mathbf{u} - \mathbf{J}(\mathbf{u})^{-1}\mathbf{G}(\mathbf{u}), \quad \mathbf{J}(\mathbf{u}) := \mathbf{A} + \operatorname{diag}\{\mathbf{F}'(\mathbf{u};\mu)\} \\
&\text{POD:} && \widehat{\mathbf{u}} \leftarrow \widehat{\mathbf{u}} - \widetilde{\mathbf{J}}(\widehat{\mathbf{u}})^{-1}\widetilde{\mathbf{G}}(\widehat{\mathbf{u}}), \quad \widetilde{\mathbf{J}}(\widehat{\mathbf{u}}) := \widehat{\mathbf{A}} + \mathbf{V}^T\operatorname{diag}\{\mathbf{F}'(\mathbf{V}\widehat{\mathbf{u}};\mu)\}\mathbf{V} \\
&\text{POD-DEIM:} && \widehat{\mathbf{u}} \leftarrow \widehat{\mathbf{u}} - \widehat{\mathbf{J}}(\widehat{\mathbf{u}})^{-1}\widehat{\mathbf{G}}(\widehat{\mathbf{u}}), \quad \widehat{\mathbf{J}}(\widehat{\mathbf{u}}) := \widehat{\mathbf{A}} + \mathbf{B}\ \operatorname{diag}\{\mathbf{F}'(\mathbf{V}_{\wp}\widehat{\mathbf{u}};\mu)\}\mathbf{V}_{\wp},
\end{aligned} \tag{4.23}$$

where $\mathbf{J} \in \mathbb{R}^{n\times n}$, $\widetilde{\mathbf{J}} \in \mathbb{R}^{k\times k}$, and $\widehat{\mathbf{J}} \in \mathbb{R}^{k\times k}$ denote the Jacobian matrices for the corresponding systems. The computational cost of performing these updates in the Newton iterations is given in Appendix A. The numerical results will be illustrated next.

4.2.2 Numerical Results

Newton iterations in (4.23) are applied to solve the full-order system (4.21), as well as the reduced systems constructed from the POD-Galerkin and POD-DEIM approaches. The spatial grid points (x_i, y_j) are equally spaced in Ω for $i, j = 1, \ldots, 50$. The full dimension is then $n = 2500$. Figures 4.5 and 4.6 show the singular values and the first 6 corresponding POD bases of the uniformly selected 144 sampled snapshot solutions for (4.19) and of the uniformly selected 144 nonlinear snapshots for

(4.20). Figure 4.7 shows the distribution of the first 30 points in Ω selected from the DEIM algorithm. Figure 4.8 shows that the POD-DEIM reduced system (with POD and DEIM having dimension 6) can accurately reproduce the solution of the full-order system of dimension 2500 with error of $O(10^{-3})$. The average errors and the average CPU time (scaled with the CPU time from sparse full-order system) for each Newton iteration of the reduced systems with different dimensions of POD and DEIM are presented in Figure 4.9. The average CPU times for higher dimensions are shown earlier in §2.2.6. These errors are averaged over a set of 225 parameters μ that were not used to obtain the sample snapshots. This suggests that the DEIM-POD reduced-order system can give a good approximation to the original system with any value of parameter $\mu \in \mathscr{D}$.

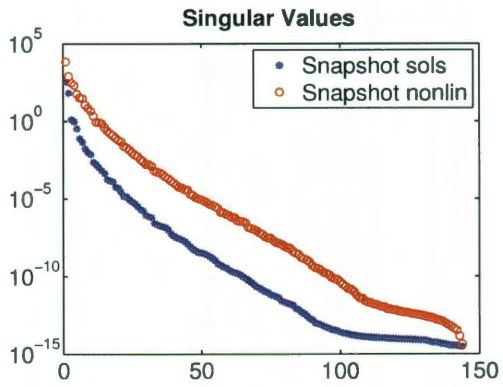


Figure 4.5: Singular values of the snapshot solutions u from (4.19) and the nonlinear snapshots $s(u;\mu)$ from (4.20).

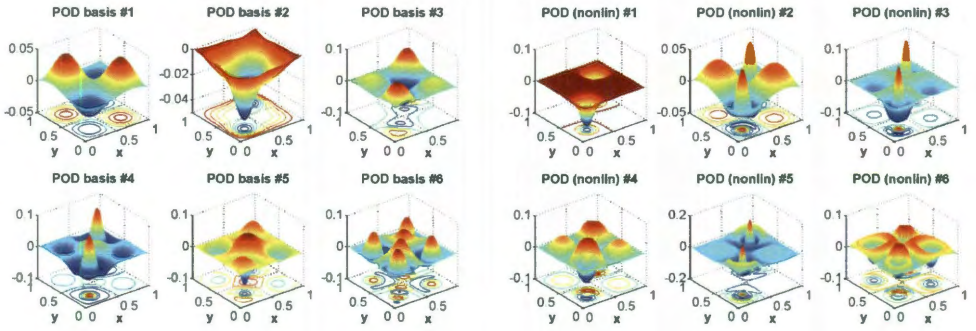


Figure 4.6: The first 6 dominant POD basis vectors of the snapshot solutions u from (4.19) and of the nonlinear snapshots $s(u;\mu)$ from (4.20).

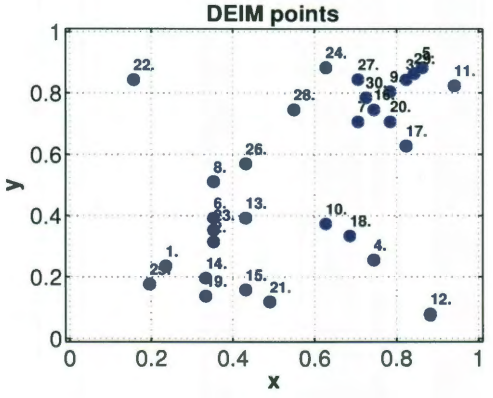


Figure 4.7: First 30 points selected by DEIM

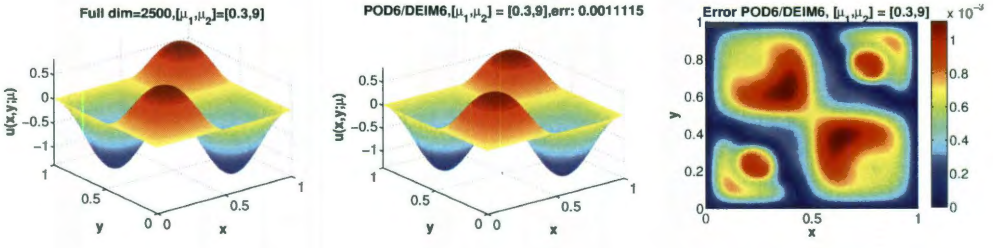


Figure 4.8: Numerical solution from the full-order system (dim= 2500) with the solution from POD-DEIM reduced system (POD dim = 6, DEIM dim = 6) for $\mu = (\mu_1, \mu_2) = (0.3, 9)$. The last plot shows the corresponding errors at the grid points.

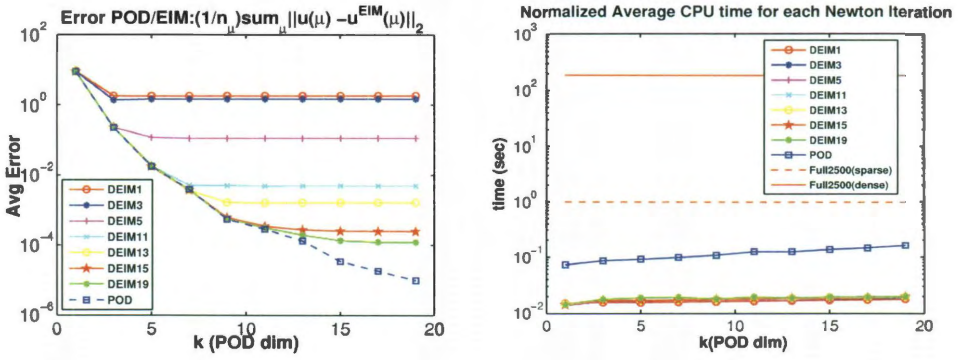


Figure 4.9: Average error from POD-DEIM reduced systems and average CPU time (scaled) in each Newton iteration for solving the steady state 2-D problem.

Chapter 5

Application of the POD-DEIM approach to Nonlinear Miscible Viscous Fingering in Porous Media

This chapter extends the application of POD-DEIM model reduction technique from the last chapter to a more complex simulation of nonlinear miscible viscous fingering in a 2-D porous medium, which is commonly used to describe many important physical phenomena, such as oil recovery process, chromatographic separation, filtration, and pollutant dispersion. This chapter demonstrates that this POD-DEIM approach can provide a vast reduction in complexity arising from nonlinearities, as compared to that of the POD-Galerkin approach. As a result, simulation times can be decreased by as much as three orders of magnitude. Specifically, as shown later in this chapter,

the dynamics of viscous fingering in the full-order system of dimension 15000 can be captured accurately by the POD-DEIM reduced system of dimension 40, with the computational time reduced by factor of $\mathcal{O}(1000)$. Hence, the procedure presented here provides a promising model reduction framework for subsequent research on more extensive nonlinear flow in porous media.

5.1 Introduction

Numerical simulations of nonlinear miscible viscous fingering have been carried out using various discretization schemes such as finite difference, finite volume, finite element, discontinuous Galerkin and Pseudo-Fourier spectral methods [86, 87, 34, 47, 45, 59, 84, 75]. The dimension of the discretized system is determined by the number of grid points in the flow domain. Usually finer grids and smaller time steps are required to capture the fine structure of the viscous fingering to obtain numerical solutions with higher accuracy. This results in a significant increase in the computational time and data storage requirements. Model reduction techniques can be used to overcome this difficulty.

As noted in the previous chapter, POD can be efficiently used to construct a problem specific set of basis functions with global support that capture the dominant characteristics of the system of interest. Fine scale details at grid points are encoded in this global basis. In the context of fluid flow in porous media, POD with Galerkin projection has been used as a model reduction procedure in many previous investiga-

tions such as [90, 92, 91, 57] for groundwater flow, [42, 56, 29, 17, 16] for immiscible two-phase (oil-water) reservoir simulation, and [36, 79, 80, 35] for miscible flow for the enhanced oil recovery (EOR) process. In the case of flows described by linear governing equations, e.g [90], the POD-Galerkin technique substantially reduces the computational complexity and simulation time. However, the standard POD alone may not give this vast reduction in the case of nonlinear flow models, as observed in [17, 16] for oil-water reservoir simulation.

The efficiency in solving the POD reduced system is limited to the linear and bilinear terms, as discussed earlier in previous chapters. In subsurface flow applications, this limitation was observed in previous works such as [17, 16]. In [17], the missing point estimation(MPE) [8] was used with a greedy algorithm [9] and a sequential QR decomposition (SQRD) approach to improve the choice of selected rows in POD vectors, and a clustering technique was applied to optimize snapshots for POD. A speedup of 10 was achieved when compared to a specialized solver and up to 700 when compared with a generic solver for the full order system. However, the numerical results in [17] indicate that to obtain reasonably good accuracy, the number of selected rows from MPE still had to be relatively large compared to the dimension of the POD basis (e.g., for the original system of dimension 60000, to obtain average relative error $\mathcal{O}(10^{-2})$, it is required to use 34 POD basis vectors with 19441 selected rows from MPE). In subsequent work [16] based on linearization of the governing equations, the trajectory piecewise-linear (TPWL) approach was applied together with POD and a

significant speedup with factor of 200-1000 was achieved. Here, DEIM will be used for approximating nonlinear terms to improve the POD procedure in the application of nonlinear miscible flow in porous media.

The formulation of the governing equations describing the nonlinear miscible viscous fingering in a 2-D porous medium, presented here in § 5.2, as well as the FD discretization scheme are taken from [84]. The matrix form of the full-order system and its corresponding reduced-order systems, both from POD and POD with DEIM are given in § 5.3 and § 5.4. Section 5.4 also discusses a practical method for computing a POD basis from a sampled set from a high-dimensional subspace. The numerical results are presented in § 5.5. To illustrate a potential usefulness of dimension reduction for parametrized systems, the POD-DEIM approach is also used to construct a single reduced-order model that can provide an accurate representation of the original full-order system over the entire specified range of parameter values. The POD-DEIM approach is also applied to a closely related problem of miscible flow with viscous fingering induced by a chemical reaction, and is shown to be equally effective on this problem. Finally, the conclusions and possible extensions for this application are discussed in § 5.6.

5.2 Governing Equations

A viscous fingering (VF) instability occurs when a less viscous fluid moves through a porous medium occupied with another more viscous fluid, which leads to the de-

velopment of finger-shaped intrusions flowing between the two fluids. An extensive number of studies have been done, both experimentally and numerically, to observe, investigate, and predict the flow displacement behavior as well as the fingering mechanisms, such as spreading, shielding, tip splitting, and coalescence (see, e.g. [86, 87, 34, 47, 45, 59, 84, 75] for more details). The equations of motion given in [84] are used here to describe the viscous fingering in horizontal flow of an incompressible fluid through a 2-D homogeneous porous medium of length L_x (horizontal) and width L_y (vertical), with a constant permeability K. The fluid is assumed to be injected horizontally from the left boundary with a uniform velocity U. Assume that the porous medium is already occupied by another fluid with higher viscosity than the injected fluid and that the two fluids are miscible. This flow evolution can be described by a system of nonlinear coupled equations derived from Darcy's law with the conservation laws of mass, momentum, and energy as shown below:

$$\nabla \cdot \mathbf{u} = 0 \tag{5.1}$$

$$\nabla P = -\frac{\mu}{K}\mathbf{u} \tag{5.2}$$

$$\frac{\partial c}{\partial t} + \mathbf{u} \cdot \nabla c = D\nabla^2 c + f(c), \tag{5.3}$$

$$\rho c_p \left[\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T\right] = D_T \nabla^2 T + (-\triangle H) f(c), \tag{5.4}$$

where $f(c)$ denotes the rate of autocatalytic reaction defined by $f(c) = -c(k_a + k_r c)(c - c_1)$ with constant parameters k_a, k_r, c_1, ρ, c_p; $\mathbf{u} = [u, w]^T \in \mathbb{R}^2$ is the velocity with components in x and y coordinates; P is the pressure; c is the concentration of the injected fluid; T is the temperature; μ is the viscosity depending on c and

T given by $\mu = \mu_0 e^{[-R_c c + R_T T]}$, with constant μ_0 and constant log-mobility ratios R_c and R_T; D, D_T and $\triangle H$ denote diffusion coefficients and enthalpy, which are assumed to be constant. This section will follow a common procedure for solving the system of equations (5.1)-(5.4) by first nondimensionalizing the system and then converting it into the form of streamfunction and vorticity, which is finally solved numerically by a discretization scheme (see e.g. [87, 84]). Define a streamfunction $\psi(x, y)$ so that $u = \frac{\partial \psi}{\partial y}, w = -\frac{\partial \psi}{\partial x}$ and define the vorticity $\omega(x, y)$ as $\omega = (\nabla \times \mathbf{u}) \cdot \mathbf{k} = \frac{\partial w}{\partial x} - \frac{\partial u}{\partial y}$ where $\mathbf{k} = [0, 0, 1]^T$. The equations (5.1)-(5.4) then can be transformed to nondimensionalized equations with respect to a moving reference frame in terms of streamfunction ψ and vorticity ω as:

$$\nabla^2 \psi = -\omega \tag{5.5}$$

$$\omega = -R_c \left(\psi_x c_x + \psi_y c_y + c_y\right) + R_T \left(\psi_x T_x + \psi_y T_y + T_y\right) \tag{5.6}$$

$$\frac{\partial c}{\partial t} + \psi_y c_x - \psi_x c_y = \nabla^2 c + \mathrm{Da} f(c), \tag{5.7}$$

$$\frac{\partial T}{\partial t} + \psi_y T_x - \psi_x T_y = \mathrm{Le} \nabla^2 T + \mathrm{sgn}(\phi) \mathrm{Da} f(c), \tag{5.8}$$

where R_c and R_T are constants (log-mobility ratios) determining the effects of concentration and temperature to the viscosity; Da (Damköhler number) and Le (Lewis number) are constant dimensionless parameters; $\mathrm{sgn}(\phi) = 1$ for exothermic reactions and $\mathrm{sgn}(\phi) = -1$ for endothermic reactions; $\psi_x = \frac{\partial \psi}{\partial x}, \psi_y = \frac{\partial \psi}{\partial y}$, $c_x = \frac{\partial c}{\partial x}, c_y = \frac{\partial c}{\partial y}$, $T_x = \frac{\partial T}{\partial x}, T_y = \frac{\partial T}{\partial y}$. The unknowns of these transformed equations (5.5)-(5.8) are $c(x, y, t)$, $T(x, y, t)$, $\psi(x, y, t)$, $\omega(x, y, t)$, for $(x, y) \in \Omega$ with dimensionless domain $\Omega = [0, \alpha \mathrm{Pe}] \times [0, \mathrm{Pe}] \subset \mathbb{R}^2$ and constant aspect ratio $\alpha := L_x / L_y$; and for time

$t \in [0, t_f]$ with (dimensionless) final simulation time t_f. Note that the dimensionless parameter Péclet number Pe, defined as Pe =: UL_x/D, determines the ratio of the rate of convective transport to the rate of diffusive transport; it also represents the length of the dimensionless flow domain.

The nonlinearities in (5.5)-(5.8) can be defined as:

$$N(\psi, v) := \psi_x v_x + \psi_y v_y, \quad F(\psi, v) := \psi_y v_x - \psi_x v_y, \quad f(c) := -c(c-1)(c+d). \quad (5.9)$$

In (5.5)-(5.8), periodic boundary conditions are imposed along top-bottom boundaries for c, T, ψ and Dirichlet boundary conditions are imposed along left-right boundaries for c, T, ψ. No boundary conditions are required for the vorticity ω, since it is defined by an algebraic expression. The initial conditions are:

$$c(x, y, 0) = T(x, y, 0) = \begin{cases} 1, & x \leq \hat{x} \\ 0, & x > \hat{x} \end{cases}, \quad (5.10)$$

for all $y \in [0, \text{Pe}]$, where $\hat{x}$ is the interface location (in this chapter, $\hat{x} = \alpha \text{Pe}/2$) and $\psi(x, y, 0) = 0$ for all $(x, y) \in \Omega$.

5.3 Finite Difference (FD) Discretized System

Central finite differences are used to construct a spatial discretization of equations (5.5)-(5.8) to obtain a system of nonlinear ODEs (5.11)-(5.14). Then the forward time integration with a predictor-corrector scheme introduced in [84] is applied to (5.11)-(5.14) to obtain FD solution at each time step.

Let $0 = x_0 < x_1 < \cdots < x_{n_x} < x_{n_x+1} = \alpha\mathrm{Pe}$ and $0 = y_0 < y_1 < \cdots < y_{n_y} < y_{n_y+1} = \mathrm{Pe}$ be equally spaced points on x-axis and y-axis for generating the grid points on the dimensionless domain $\Omega = [0, \alpha\mathrm{Pe}] \times [0, \mathrm{Pe}]$ with $dx = \alpha\mathrm{Pe}/(n_x + 1)$ and $dy = \mathrm{Pe}/(n_y + 1)$. Define vectors of unknown variables of dimension $n := n_y n_x$ as $\mathbf{c}(t), \mathbf{T}(t), \psi(t), \omega(t) \in \mathbb{R}^n$, containing approximate solutions for $c(x_i, y_j, t)$, $T(x_i, y_j, t)$, $\psi(x_i, y_j, t)$, and $\omega(x_i, y_j, t)$ at grid points (x_i, y_j) for $i = 1, \ldots, n_x$ and $j = 1, \ldots, n_y$. The corresponding spatial finite difference discretized system of (5.5)-(5.8) then becomes a system of nonlinear ODEs coupled with algebraic equations, which can be written in matrix form as follows. For $t \in [0, t_f]$,

$$\frac{d\mathbf{c}(t)}{dt} = -\mathbf{F}(\psi(t), \mathbf{c}(t)) + [\mathbf{A}\mathbf{c}(t) + \mathbf{b}] + \mathrm{Da}\mathbf{f}(\mathbf{c}(t)) \tag{5.11}$$

$$\frac{d\mathbf{T}(t)}{dt} = -\mathbf{F}(\psi(t), \mathbf{T}(t)) + \mathrm{Le}[\mathbf{A}\mathbf{T}(t) + \mathbf{b}] + \mathrm{sgn}(\phi)\mathrm{Da}\mathbf{f}(\mathbf{c}(t)) \tag{5.12}$$

$$\omega(t) = -R_c\left[\mathbf{N}(\psi(t), \mathbf{c}(t)) + \mathbf{A}_y\mathbf{c}(t)\right] + R_T\left[\mathbf{N}(\psi(t), \mathbf{T}(t)) + \mathbf{A}_y\mathbf{T}(t)\right] \tag{5.13}$$

$$\mathbf{A}\psi(t) = -\omega(t), \tag{5.14}$$

where the nonlinear functions $\mathbf{F}$, $\mathbf{N} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ and $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ are defined as

$$\mathbf{F}(\psi, \mathbf{c}) = (\mathbf{A}_y\psi).*(\mathbf{A}_x\mathbf{c} + \mathbf{b}_x) - (\mathbf{A}_x\psi).*(\mathbf{A}_y\mathbf{c}), \tag{5.15}$$

$$\mathbf{N}(\psi, \mathbf{c}) = (\mathbf{A}_x\psi).*(\mathbf{A}_x\mathbf{c} + \mathbf{b}_x) + (\mathbf{A}_y\psi).*(\mathbf{A}_y\mathbf{c}), \tag{5.16}$$

$$\mathbf{f}(\mathbf{c}) = -\mathbf{c}.*(\mathbf{c} - 1).*(\mathbf{c} + d), \tag{5.17}$$

with '.*' denoting componentwise multiplication as used in MATLAB; $\mathbf{A}_x, \mathbf{A}_y, \mathbf{A} \in \mathbb{R}^{n\times n}$ are (sparse) constant coefficient matrices for discrete first-order and second-order differential operators; $\mathbf{b}, \mathbf{b}_x \in \mathbb{R}^n$ are constant vectors reflecting the boundary

conditions. In general, the discretized system for this nonlinear VF has to be very large to capture the fine details of fingers flowing through the domain, especially for high Péclet number. This, therefore, causes substantial increases in computational time and memory storage, which may further make it impossible to perform the simulation in a reasonable computational time. The next section will apply the model reduction techniques from Chapter 2 to overcome this difficulty.

5.4 Reduced-Order System

As described in Chapter 2, the Proper Orthogonal Decomposition (POD) and Discrete Empirical Interpolation Method (DEIM) are applied to construct a reduced-order system of the full-order system (5.11)-(5.14) described in the previous section. Sections 5.4.1 and 5.4.2 give the details of constructing this reduced-order system.

5.4.1 POD reduced system

In this setting, *snapshots* are the numerically sampled solutions at particular time steps or at particular parameter values. POD gives an optimal set of basis vectors that minimize the mean square error from approximating these snapshots and can be obtained from the singular value decomposition (SVD).

The POD basis here is constructed for each variable separately since they are governed by distinct physics. Let $\widehat{\mathbf{C}} = [\mathbf{c}^1, \ldots, \mathbf{c}^{n_s}] \in \mathbb{R}^{n \times n_s}$ be the snapshot matrix for concentration with $\mathbf{c}^j$ denoting the solution of the FD discretized system at time

t_j. The POD basis of dimension k for the snapshots $\{\mathbf{c}^j\}_{j=1}^{n_s}$ is the set of left singular vectors of $\widehat{\mathbf{C}}$ corresponding to the k largest singular values, i.e. columns of $\mathbf{V} = \widehat{\mathbf{V}}(:, 1:k) \in \mathbb{R}^{n\times k}$ for $k < r_c := \text{rank}(\widehat{\mathbf{C}})$, where $\widehat{\mathbf{C}} = \widehat{\mathbf{V}}\mathbf{\Sigma}\mathbf{Z}^T$ is the SVD of $\widehat{\mathbf{C}}$ with $\mathbf{\Sigma} = \text{diag}(\sigma_1, \ldots, \sigma_{r_c}) \in \mathbb{R}^{r_c\times r_c}$; $\sigma_1 \geq \cdots \geq \sigma_{r_c} > 0$ and $\widehat{\mathbf{V}} \in \mathbb{R}^{n\times r_c}$, $\mathbf{Z} \in \mathbb{R}^{n_s\times r_c}$ having orthonormal columns. Similarly, let $\mathbf{Q}$, $\mathbf{U}$, $\mathbf{W} \in \mathbb{R}^{n\times k}$ be POD basis matrices of dimension k for the snapshots $\{\mathbf{T}^j\}_{j=1}^{n_s}$, $\{\omega^j\}_{j=1}^{n_s}$, and $\{\psi^j\}_{j=1}^{n_s}$.

Then the POD reduced-order system is constructed by applying the Galerkin projection method to equations (5.11)-(5.14) by first replacing $\mathbf{c}$, $\mathbf{T}$, ω, ψ with their approximations $\mathbf{V}\tilde{\mathbf{c}}$, $\mathbf{Q}\tilde{\mathbf{T}}$, $\mathbf{U}\tilde{\omega}$, $\mathbf{W}\tilde{\psi}$, respectively, for reduced variables $\tilde{\mathbf{c}}$, $\tilde{\mathbf{T}}$, $\tilde{\omega}$, $\tilde{\psi} \in \mathbb{R}^k$, and then premultiplying equation (5.11) by $\mathbf{V}^T$, equation (5.12) by $\mathbf{Q}^T$, and equations (5.13) and (5.14) by $\mathbf{U}^T$. The resulting POD reduced system is

$$\frac{d\tilde{\mathbf{c}}(t)}{dt} = -\mathbf{V}^T\tilde{\mathbf{F}}_1(\tilde{\psi}(t), \tilde{\mathbf{c}}(t)) + [\underbrace{\mathbf{V}^T\mathbf{A}\mathbf{V}}_{=:\tilde{\mathbf{A}}_1}\tilde{\mathbf{c}}(t) + \underbrace{\mathbf{V}^T\mathbf{b}}_{=:\tilde{\mathbf{b}}_1}] + \text{Da}\mathbf{V}^T\mathbf{f}(\mathbf{V}\tilde{\mathbf{c}}(t)) \tag{5.18}$$

$$\frac{d\tilde{\mathbf{T}}(t)}{dt} = -\mathbf{Q}^T\tilde{\mathbf{F}}_2(\tilde{\psi}(t), \tilde{\mathbf{T}}(t)) + \text{Le}[\underbrace{\mathbf{Q}^T\mathbf{A}\mathbf{Q}}_{=:\tilde{\mathbf{A}}_2}\tilde{\mathbf{T}}(t) + \underbrace{\mathbf{Q}^T\mathbf{b}}_{=:\tilde{\mathbf{b}}_2}] + \text{sgn}(\phi)\text{Da}\mathbf{Q}^T\mathbf{f}(\mathbf{V}\tilde{\mathbf{c}}(t)) \tag{5.19}$$

$$\tilde{\omega}(t) = -R_c\left[\mathbf{U}^T\tilde{\mathbf{N}}_1(\tilde{\psi}(t), \tilde{\mathbf{c}}(t)) + \underbrace{\mathbf{U}^T\mathbf{A}_y\mathbf{V}}_{=:\tilde{\mathbf{A}}_3}\tilde{\mathbf{c}}(t)\right] + R_T\left[\mathbf{U}^T\tilde{\mathbf{N}}_2(\tilde{\psi}(t), \tilde{\mathbf{T}}(t)) + \underbrace{\mathbf{U}^T\mathbf{A}_y\mathbf{Q}}_{=:\tilde{\mathbf{A}}_4}\tilde{\mathbf{T}}(t)\right] \tag{5.20}$$

$$\underbrace{\mathbf{U}^T\mathbf{A}\mathbf{W}}_{=:\tilde{\mathbf{A}}_5}\tilde{\psi}(t) = -\tilde{\omega}(t), \tag{5.21}$$

where $\tilde{\mathbf{F}}_1$, $\tilde{\mathbf{F}}_2$, $\tilde{\mathbf{N}}_1$, $\tilde{\mathbf{N}}_2$: $\mathbb{R}^k \times \mathbb{R}^k \to \mathbb{R}^n$,

$$\tilde{\mathbf{F}}_1(\tilde{\psi}, \tilde{\mathbf{c}}) = (\underbrace{\mathbf{A}_y\mathbf{W}}\,\tilde{\psi}). * (\underbrace{\mathbf{A}_x\mathbf{V}}\,\tilde{\mathbf{c}} + \mathbf{b}_x) - (\underbrace{\mathbf{A}_x\mathbf{W}}\,\tilde{\psi}). * (\underbrace{\mathbf{A}_y\mathbf{V}}\,\tilde{\mathbf{c}}), \tag{5.22}$$

$$\tilde{\mathbf{F}}_2(\tilde{\psi}, \tilde{\mathbf{T}}) = (\underbrace{\mathbf{A}_y\mathbf{W}}\,\tilde{\psi}). * (\underbrace{\mathbf{A}_x\mathbf{Q}}\,\tilde{\mathbf{T}} + \mathbf{b}_x) - (\underbrace{\mathbf{A}_x\mathbf{W}}\,\tilde{\psi}). * (\underbrace{\mathbf{A}_y\mathbf{Q}}\,\tilde{\mathbf{T}}), \tag{5.23}$$

$$\tilde{\mathbf{N}}_1(\tilde{\psi}, \tilde{\mathbf{c}}) = (\underbrace{\mathbf{A}_x\mathbf{W}}\,\tilde{\psi}). * (\underbrace{\mathbf{A}_x\mathbf{V}}\,\tilde{\mathbf{c}} + \mathbf{b}_x) + (\underbrace{\mathbf{A}_y\mathbf{W}}\,\tilde{\psi}). * (\underbrace{\mathbf{A}_y\mathbf{V}}\,\tilde{\mathbf{c}}), \tag{5.24}$$

$$\tilde{\mathbf{N}}_2(\tilde{\psi}, \tilde{\mathbf{T}}) = (\underbrace{\mathbf{A}_x\mathbf{W}}\,\tilde{\psi}). * (\underbrace{\mathbf{A}_x\mathbf{Q}}\,\tilde{\mathbf{T}} + \mathbf{b}_x) + (\underbrace{\mathbf{A}_y\mathbf{W}}\,\tilde{\psi}). * (\underbrace{\mathbf{A}_y\mathbf{Q}}\,\tilde{\mathbf{T}}), \tag{5.25}$$

$$\mathbf{f}(\mathbf{V}\tilde{\mathbf{c}}) = -\mathbf{V}\tilde{\mathbf{c}}. * (\mathbf{V}\tilde{\mathbf{c}} - 1). * (\mathbf{V}\tilde{\mathbf{c}} + d). \tag{5.26}$$

The coefficient matrices $\tilde{\mathbf{A}}_1, \ldots, \tilde{\mathbf{A}}_5 \in \mathbb{R}^{k\times k}$ and vectors $\tilde{\mathbf{b}}_1, \tilde{\mathbf{b}}_2 \in \mathbb{R}^k$ defined in (5.18)-(5.21) for the linear terms of the POD reduced system as well as the coefficient matrices in the nonlinear functions from (5.22)-(5.26) (i.e. $\mathbf{A}_y\mathbf{W}$, $\mathbf{A}_x\mathbf{V}$, $\mathbf{A}_x\mathbf{W}$, $\mathbf{A}_y\mathbf{V}$, $\mathbf{A}_x\mathbf{Q}$, $\mathbf{A}_y\mathbf{Q} \in \mathbb{R}^{n\times k}$ grouped by the curly braces) can be precomputed, retained, and re-used in all time steps. However, performing the componentwise multiplications in (5.22)-(5.26) and computing the projected nonlinear terms in (5.18)-(5.21):

$$\mathbf{V}^T\tilde{\mathbf{F}}_1(\tilde{\psi}, \tilde{\mathbf{c}}), \quad \mathbf{V}^T\mathbf{f}(\mathbf{V}\tilde{\mathbf{c}}), \quad \mathbf{Q}^T\tilde{\mathbf{F}}_2(\tilde{\psi}, \tilde{\mathbf{T}}), \quad \mathbf{Q}^T\mathbf{f}(\mathbf{V}\tilde{\mathbf{c}}), \quad \mathbf{U}^T\tilde{\mathbf{N}}_1(\tilde{\psi}, \tilde{\mathbf{c}}), \quad \mathbf{U}^T\tilde{\mathbf{N}}_2(\tilde{\psi}, \tilde{\mathbf{T}}) \tag{5.27}$$

still have computational complexities depending on the dimension n of the original system (from both evaluating the nonlinear functions and performing matrix multiplications for projecting on POD bases). The Discrete Empirical Interpolation Method (DEIM) is used to remove this dependency as shown in the next section.

Memory requirements for the POD reduced system

Besides the complexity of the POD-Galerkin technique as discussed in Chapter 2, the memory storage requirement can also be an issue for the POD reduced system. To obtain the approximate solution from the POD reduced system, one must store POD reduced solutions of order $\mathcal{O}(kn_t)$ and POD basis matrices of order $\mathcal{O}(nk)$. This can be much smaller than the required memory space to store $\mathcal{O}(nn_t)$ of the full-order solutions when $k \ll n_t$ and $k \ll n$. However, coefficient matrices in the POD reduced system are generally *dense* and they may require memory space more than those in the full-order system due to the nonlinear terms. As discussed above, the coefficient matrices that must be retained while solving the POD reduced system are of order $\mathcal{O}(k^2)$ for projected linear terms $\tilde{\mathbf{A}}_1, \ldots, \tilde{\mathbf{A}}_5$ with projected constant vectors $\tilde{\mathbf{b}}_1, \tilde{\mathbf{b}}_2$; and $\mathcal{O}(nk)$ for the nonlinear terms (5.22)-(5.25). These $\mathcal{O}(nk)$ coefficient matrices are indeed needed to avoid inefficient computation of the prolongation of the reduced variables back to the original dimension in (5.22)-(5.25) at every time step. The problem is that memory space of order $\mathcal{O}(nk)$ can clearly exceed the $\mathcal{O}(n)$ memory requirement for the *sparse* coefficient matrices of the full-order system. The DEIM approximation allows further precomputation so that this required memory space for coefficient matrices can be reduced, as shown next.

5.4.2 POD-DEIM reduced system

The projected nonlinear function in (5.27) can be approximated by DEIM in a form that enables precomputation so that the computational cost is decreased and independent of original dimension n. Evaluating the approximate nonlinear term from DEIM does not require a prolongation of the reduced state variables back to the original high dimensional state approximation, as is required to evaluate the nonlinearity in the original POD approximation, e.g., for $\mathbf{f}$ in (5.26). Only a few entries of the original nonlinear term corresponding to the specially selected interpolation indices from DEIM must be evaluated at each time step. The DEIM approximation is given formally in Definition 2.2.1 and the procedure for selecting DEIM indices is given in Algorithm 1 from Chapter 2.

DEIM approximation is next applied to each of the nonlinear functions $\tilde{\mathbf{F}}_1$, $\tilde{\mathbf{F}}_2$, $\tilde{\mathbf{N}}_1$, $\tilde{\mathbf{N}}_2$, and $\mathbf{f}$ defined in (5.22)-(5.26). Only DEIM approximation of $\tilde{\mathbf{F}}_1$ shall be presented here in detail. Other nonlinear functions can be treated similarly. Let $\mathbf{U}^{F_1} \in \mathbb{R}^{n\times m}$, $m \leq n$, be the POD basis matrix of rank m for snapshots from the nonlinear function $\mathbf{F}_1$ in (5.15), which can be obtained at the same time as the solution snapshots. Then $\mathbf{U}^{F_1}$ is used to select a set of m DEIM indices, denoted by $\vec{\wp}^{F_1} = [\wp_1^{F_1}, \ldots, \wp_m^{F_1}]^T$. From Definition 2.2.1, the DEIM approximation is then of the form $\tilde{\mathbf{F}}_1 \approx \mathbf{U}^{F_1}(\mathbf{P}_{F_1}^T\mathbf{U}^{F_1})^{-1}\tilde{\mathbf{F}}_1^m$ and the projected nonlinear term $\mathbf{V}^T\tilde{\mathbf{F}}_1(\tilde{\psi}, \tilde{\mathbf{c}})$ in (5.27)

of the POD reduced system then can be approximated as

$$\mathbf{V}^T\tilde{\mathbf{F}}_1(\tilde{\psi},\tilde{\mathbf{c}}) \approx \underbrace{\mathbf{V}^T\mathbf{U}^{F_1}(\mathbf{P}_{F_1}^T\mathbf{U}^{F_1})^{-1}}_{\mathbf{E}_1}\tilde{\mathbf{F}}_1^m(\tilde{\psi},\tilde{\mathbf{c}}), \tag{5.28}$$

where $\tilde{\mathbf{F}}_1^m(\tilde{\psi},\tilde{\mathbf{c}}) = \mathbf{P}_{F_1}^T\tilde{\mathbf{F}}_1(\tilde{\psi},\tilde{\mathbf{c}})$. By using the fact that $\tilde{\mathbf{F}}_1$ in (5.22) is a pointwise function, $\tilde{\mathbf{F}}_1^m : \mathbb{R}^k \times \mathbb{R}^k \to \mathbb{R}^m$ can be defined as

$$\tilde{\mathbf{F}}_1^m(\tilde{\psi},\tilde{\mathbf{c}}) := (\underbrace{\mathbf{P}_{F_1}^T\mathbf{A}_y\mathbf{W}}\tilde{\psi}).*(\underbrace{\mathbf{P}_{F_1}^T\mathbf{A}_x\mathbf{V}}\tilde{\mathbf{c}}+\underbrace{\mathbf{P}_{F_1}^T\mathbf{b}_x}) - (\underbrace{\mathbf{P}_{F_1}^T\mathbf{A}_x\mathbf{W}}\tilde{\psi}).*(\underbrace{\mathbf{P}_{F_1}^T\mathbf{A}_y\mathbf{V}}\tilde{\mathbf{c}}). \tag{5.29}$$

Each of the m-by-k coefficient matrices and the m-vector grouped by the curly brackets in the above equation, as well as $\mathbf{E}_1 := \mathbf{V}^T\mathbf{U}^{F_1}(\mathbf{P}_{F_1}^T\mathbf{U}^{F_1})^{-1} \in \mathbb{R}^{k\times m}$ from (5.28), can be precomputed and re-used at all time steps, so that the computational complexity of the approximate nonlinear term (5.28) is independent of the full-order dimension n. Finally, the POD-DEIM reduced system is of the form:

$$\frac{d\tilde{\mathbf{c}}(t)}{dt} = -\mathbf{E}_1\tilde{\mathbf{F}}_1^m(\tilde{\psi}(t),\tilde{\mathbf{c}}(t)) + [\tilde{\mathbf{A}}_1\tilde{\mathbf{c}}(t)+\tilde{\mathbf{b}}_1] + \mathrm{Da}\mathbf{E}_2\mathbf{f}(\mathbf{P}_f^T\mathbf{V}\tilde{\mathbf{c}}(t)) \tag{5.30}$$

$$\frac{d\tilde{\mathbf{T}}(t)}{dt} = -\mathbf{E}_3\tilde{\mathbf{F}}_2^m(\tilde{\psi}(t),\tilde{\mathbf{T}}(t)) + \mathrm{Le}[\tilde{\mathbf{A}}_2\tilde{\mathbf{T}}(t)+\tilde{\mathbf{b}}_2] + \mathrm{sgn}(\phi)\mathrm{Da}\mathbf{E}_4\mathbf{f}(\mathbf{P}_f^T\mathbf{V}\tilde{\mathbf{c}}(t)) \tag{5.31}$$

$$\tilde{\omega}(t) = -R_c\left[\mathbf{E}_5\tilde{\mathbf{N}}_1^m(\tilde{\psi}(t),\tilde{\mathbf{c}}(t)) + \tilde{\mathbf{A}}_3\tilde{\mathbf{c}}(t)\right] + R_T\left[\mathbf{E}_6\tilde{\mathbf{N}}_2^m(\tilde{\psi}(t),\tilde{\mathbf{T}}(t)) + \tilde{\mathbf{A}}_4\tilde{\mathbf{T}}(t)\right] \tag{5.32}$$

$$\tilde{\mathbf{A}}_5\tilde{\psi}(t) = -\tilde{\omega}(t), \tag{5.33}$$

where $\tilde{\mathbf{F}}_2^m$, $\tilde{\mathbf{N}}_1^m$, $\tilde{\mathbf{N}}_2^m$, can be defined analogously to $\tilde{\mathbf{F}}_1^m$, and $\mathbf{E}_2,\ldots,\mathbf{E}_6 \in \mathbb{R}^{k\times m}$ can be obtained in a similar manner from other nonlinear functions as for $\mathbf{E}_1$. The equations (5.30) and (5.31) used the fact that $\mathbf{f}$ is also a componentwise function, i.e., $\mathbf{f}(\mathbf{c}_j) = [\mathbf{f}(\mathbf{c})]_j$, which implies $\mathbf{P}_f^T\mathbf{f}(\mathbf{V}\tilde{\mathbf{c}}(t)) = \mathbf{f}(\mathbf{P}_f^T\mathbf{V}\tilde{\mathbf{c}}(t))$ where $\mathbf{P}_f$ is defined

analogously to $\mathbf{P}_{F_1}$. Note that pre-multiplying $\mathbf{P}_f^T$ to $\mathbf{V}$ is equivalent to selecting rows of $\mathbf{V}$ corresponding to DEIM indices, and hence the matrix multiplication for $\mathbf{P}_f^T\mathbf{V}$ need not be performed explicitly. Hence, it is only required to store an m-vector of DEIM indices for each of the nonlinear functions, instead of the matrix $\mathbf{P}_{F_1}$ or $\mathbf{P}_f$.

Memory storage requirement for the POD-DEIM reduced system

As in the case of the POD reduced system, to recover the approximate solution from the POD-DEIM reduced system, it is required to store reduced solutions of order $\mathcal{O}(kn_t)$ and POD basis matrices of order $\mathcal{O}(nk)$. The precomputed coefficient matrices that one must retain are of order $\mathcal{O}(k^2)$ for the projected linear terms $\tilde{\mathbf{A}}_1, \ldots, \tilde{\mathbf{A}}_5 \in \mathbb{R}^{k\times k}$, with the projected constant vectors $\tilde{\mathbf{b}}_1, \tilde{\mathbf{b}}_2 \in \mathbb{R}^k$; $\mathcal{O}(m)$ for the DEIM indices; and $\mathcal{O}(mk)$ for the nonlinear terms, $\mathbf{E}_1, \ldots, \mathbf{E}_6 \in \mathbb{R}^{k\times m}$ and the m-by-k matrices inside the nonlinear functions such as the ones for $\tilde{\mathbf{F}}_1^m$ in (5.29). This memory requirement is clearly less than the one for the POD reduced system and is independent of the original dimension n. These precomputed coefficient matrices allow a substantial reduction in computational complexity, which now depends on only the dimensions k of POD and m of DEIM (but not n). DEIM therefore improves the efficiency of the POD approximation and achieves a complexity reduction of the nonlinear term with a complexity proportional to the number of reduced variables. This efficiency reflects in the speedup of simulation time presented in § 5.5.

Remark on the computation of a POD basis

To compute a POD basis for a snapshot matrix in $\mathbb{R}^{n\times n_s}$, when the spatial dimension n of the discretization is much larger than the number of snapshots n_s, it may not be efficient to use the SVD directly. In particular, let $\mathbf{Y}$ be the n-by-n_s matrix of snapshots with $n \gg n_s$. In this case, the POD basis is commonly obtained from the eigenvalue decomposition of the smaller matrix $\mathbf{Y}^T\mathbf{Y} \in \mathbb{R}^{n_s\times n_s}$. However, the round-off error from matrix multiplication for constructing $\mathbf{Y}^T\mathbf{Y}$ can affect the resulting POD basis. Alternatively, as suggested in [3], an efficient procedure for computing the SVD of $\mathbf{Y}$ is to first perform the QR factorization of $\mathbf{Y}$, and then compute the SVD of the (smaller) n_s-by-n_s matrix $\mathbf{R}$ where $\mathbf{Y} = \mathbf{QR}$ is the QR decomposition of $\mathbf{Y}$ with $\mathbf{Q} \in \mathbb{R}^{n\times n_s}$ denoting a matrix with orthonormal columns and $\mathbf{R} \in \mathbb{R}^{n_s\times n_s}$ denoting an upper triangular matrix. Let $\mathbf{R} = \mathbf{U}\Sigma\mathbf{V}^T$ be the SVD of $\mathbf{R}$. Then the SVD of $\mathbf{Y}$ is finally given by $\mathbf{Y} = (\mathbf{QU})\Sigma\mathbf{V}^T$ and the POD basis can be obtained from the columns of $\mathbf{QU}$. To preserve the numerical stability for the case $n \gg n_s$, QR factorization of $\mathbf{Y}$ can be computed by a Gram-Schmidt process with reorthogonalization algorithm [26]. This approach also makes it possible to update the POD basis when additional snapshots are included.

5.5 Numerical Results

This section presents three numerical experiments. The first one considers the POD-DEIM reduced system for a set of fixed parameters. The second one considers the

reduced system that can be used for various values of the Péclet number in a certain range. The last one considers miscible flow with viscous fingering induced by a simple chemical reaction. For all these cases, in addition to the initial condition for c given in § 5.2, random noise between 0 and 1 is added at each grid point on the interface to trigger the instability in reasonable computing time as done in many investigations such as [87, 34, 84]. The accuracy in all numerical cases is measured by the (2-norm) average relative error, E_c, defined as

$$E_c := \frac{1}{n_t} \sum_{j=1}^{n_t} \frac{\|\mathbf{c}_j - \mathbf{c}_j^r\|_2}{\|\mathbf{c}_j\|_2},$$

where $\mathbf{c}_j \in \mathbb{R}^n$ denotes the solution for concentration of the full-order system at time t_j; $\mathbf{c}_j^r := \mathbf{V}\tilde{\mathbf{c}}_j \in \mathbb{R}^n$ with $\tilde{\mathbf{c}}_j \in \mathbb{R}^k$ being the solution from a reduced system (POD or POD-DEIM) at time t_j; and POD basis matrix $\mathbf{V} \in \mathbb{R}^{n\times k}$ for $\mathbf{c}$.

5.5.1 Fixed Parameters

The system (5.5)-(5.8) is solved numerically using a finite difference scheme from [84]. This section considers the isothermal case (constant temperature: $R_T = 0$). The parameters used here are $R_c = 3$; $R_T = 0$; $a = 2$; Pe $= 250$; $Le = 1$; $Da = 0.01$; $d = 0.1$. The number of spatial grid points is 150 on the x-axis and 100 on the y-axis. The dimension of the full-order system is then 15000.

The singular values of 250 solution snapshots and nonlinear snapshots are shown in Figure 5.1. In Figure 5.2, the solutions for concentration from the POD-DEIM reduced system (5.30)-(5.33), with POD and DEIM of dimension 40, are shown with

the corresponding ones from the full-order system and also the corresponding absolute errors at the grid points. This figures shows that POD-DEIM reduces more than 300 times in dimension and reduces the computational time by factor of $\mathcal{O}(10^3)$ with $\mathcal{O}(10^{-3})$ error as shown in Table 5.1.

From the error plot in Figure 5.3, each POD-DEIM error curve (solid line) initially decreases as the dimension of the POD basis increases, then the error stagnates once a certain dimension of POD basis is reached. The stagnation may result when the DEIM approximation error exceeds the POD approximation error, and in this case DEIM accuracy does not improve further even by increasing the dimension of the POD basis. On the other hand, for a fixed dimension of POD basis, the errors from POD-DEIM reduced systems decrease as the dimension of DEIM increases, but they do not get lower than the POD errors. That is, once the DEIM error is essentially equal to the POD error, no further reduction of DEIM error is possible through increasing the dimension of the DEIM approximation. The error plots also indicate an *optimal* choice of DEIM dimension for a given POD dimension (and vice versa), which is the *'corner'* of each curve. However, these error curves are not known in advance and hence cannot be used to determine the reduced dimension in practice. The plot of the CPU time in Figure 5.3 used in computing the POD reduced system clearly reflects the dependency on the dimension of the original full-order system. Figure 5.3 and Table 5.1 show a significant improvement in computational time of the POD-DEIM reduced system from both the POD reduced system and the full-order system.

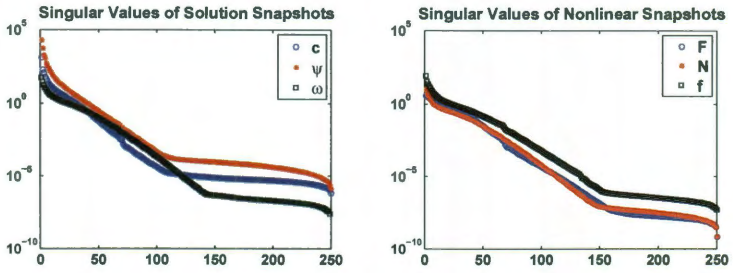


Figure 5.1: Singular values of the solution snapshots and the nonlinear snapshots.

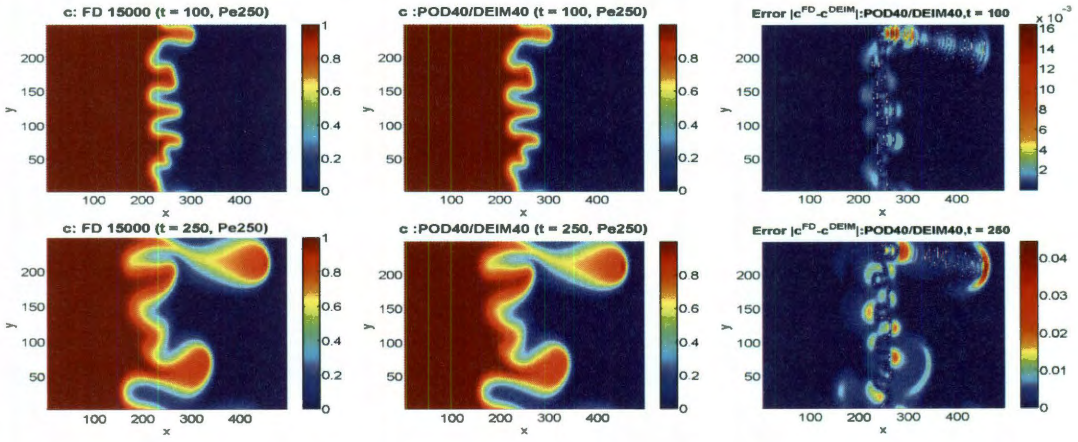
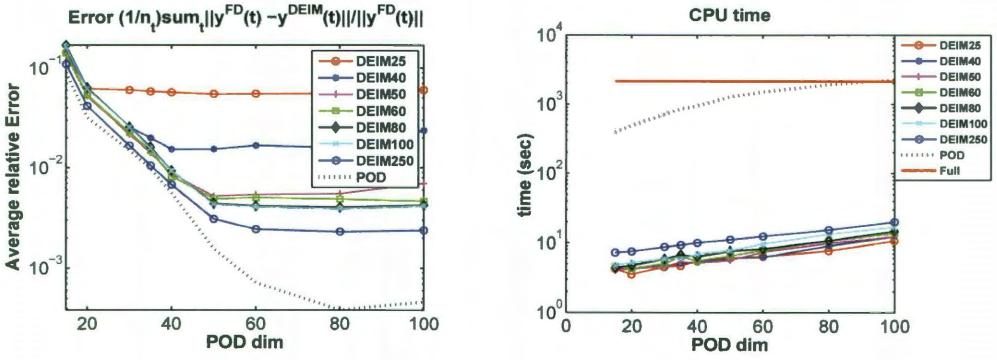


Figure 5.2: Concentration plots of the injected fluid (from the left half) at time $t = 100$ and $t = 250$ from the full-order system of dimension 15000 and from the POD-DEIM reduced system with both POD and DEIM having dimension 40 (fixed parameters).

Dimension	Avg Rel Error of c	CPU time (sec)	~Ratio CPU time
Full 15000 (FD)	-	2.138×10^3	1
POD20	5.597×10^{-3}	1.206×10^2	1/18
POD20/DEIM20	2.041×10^{-2}	9.225×10^{-1}	1/2318
POD40	4.066×10^{-4}	2.442×10^2	1/9
POD40/DEIM40	2.045×10^{-3}	1.275	1/1677

Table 5.1: Average relative error (2-norm) of the solution for the concentration c and CPU time of the full-order system, POD reduced system, and POD-DEIM reduced system with Pe = 250 (fixed parameters) with the ratios of the CPU time normalized by the time of full-order system.



(a) (b)

Figure 5.3: (a) Average relative errors of $\mathbf{y} = [c; \psi; \omega]$: defined as $\mathbf{E} := \frac{1}{n_t} \sum_{j=1}^{n_t} \frac{\|\mathbf{y}_j - \mathbf{y}_j^r\|_2}{\|\mathbf{y}_j\|_2}$, from the POD-DEIM reduced system compared with the ones from the POD reduced system. (b) CPU time of the full system, POD reduced system, and POD-DEIM reduced system.

5.5.2 Varying Péclet number: Pe $\in [110, 120]$

Consider the same numerical setup as for the previous case in Section 5.5.1, except that this numerical experiment is now interested in the parameter Pe in the interval $[110, 120]$. The POD basis used for approximating the solution space is constructed from 398 snapshots taken from two full-order FD systems corresponding to Pe $=$ 110 and 120 (199 snapshots are uniformly selected in time $t \in [0, 200]$ from each system). The resulting POD-DEIM reduced system can be used to approximate systems with arbitrarily parameter Pe in the interval $[110, 120]$. To demonstrate the effectiveness of this reduced system, consider the solutions of the VF system with parameter Pe $= 115$ which was not used in constructing the POD bases of this POD-DEIM reduced system as shown in Figure 5.4 for concentration from the POD-DEIM reduced system with POD of dimension 30 and DEIM of dimension 50, as well as the corresponding absolute error at the grid points when compared with the full-order system of dimension 15000. The corresponding average relative error is $\mathcal{O}(10^{-3})$ for this 300 times reduction in dimension. An envisioned use of this reduction is to conduct many different simulations with various settings of the Péclet number. To illustrate the potential to drastically reduce simulation time without loss of accuracy, consider this miscible flow system with different Péclet numbers ranging across the entire interval $[110, 120]$. Specifically, 11 simulations will be conducted corresponding to Pe $= 110, 111, \ldots, 119, 120$. As expected, the POD-DEIM approach significantly reduced the total simulation time from 2.33 hours for the full system to roughly 13

seconds with accuracy $\mathcal{O}(10^{-3})$ as shown in Table 5.3. The POD reduced model hardly reduced the computation time by comparison, e.g., from Table 5.3, the POD system of dimension 30 reduces computational time only by a factor of 5, while the POD-DEIM system (POD=30, DEIM=50) reduces it roughly by factor of 700.

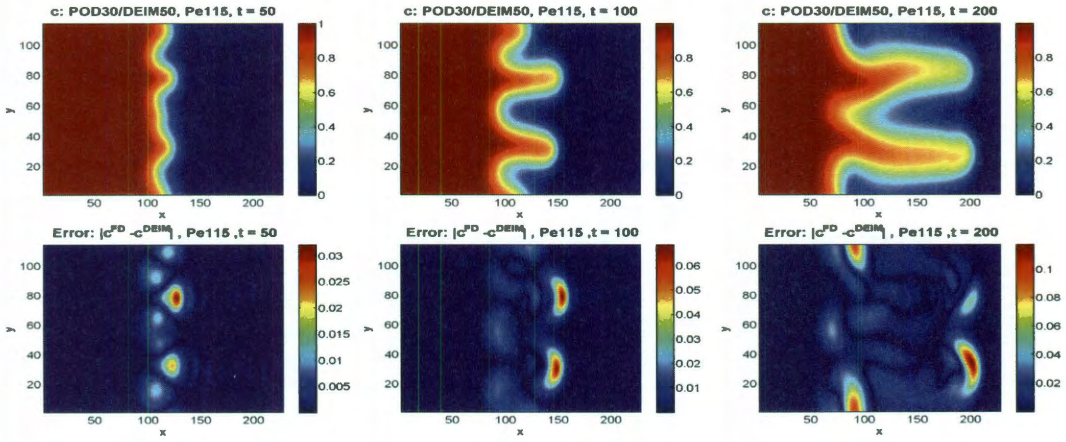


Figure 5.4: Concentration plots of the injected fluid at time $t = 50, 100, 200$ from the POD-DEIM reduced system with POD and DEIM having dimensions 30 and 50, with the corresponding absolute error at the grid points when compared with the full-order system of dimension 15000 (Péclet number $\mathrm{Pe} = 115$).

5.5.3 Miscible Viscous Fingering Induced by Chemical Reaction

This section considers a system from [34] that describes miscible flow with viscous fingering induced by a simple chemical reaction $A + B \rightarrow C$, which occurs at the interface of the reactants A and B, producing a product C. The system of governing equations is in a similar form to the one presented in Section 5.2 and given by the

Dimension	Avg Rel Error of c	CPU time (sec)	Ratio CPU
Full 15000 (FD)	-	7.384×10^2	1
POD30	5.907×10^{-3}	1.338×10^2	1/6
POD30/DEIM30	3.133×10^{-2}	0.843	1/876
POD30/DEIM50	7.395×10^{-3}	0.909	1/812
POD50	5.910×10^{-3}	2.434×10^2	1/3
POD50/DEIM50	8.579×10^{-3}	1.150	1/642

Table 5.2: Average relative error (2-norm) of the concentration c and CPU time (sec) for solving the full-order system, POD reduced system, and POD-DEIM reduced system with Péclet number $\mathrm{Pe} = 115$, which is arbitrary chosen from the interval $[110, 120]$, with the ratios of the CPU time normalized by the time of the full-order system.

Dimension	Avg Rel. Error	Avg CPU time	CPU time 11 runs	Ratio CPU
Full 15000 (FD)	-	7.384×10^2	8.402×10^3 ($\sim$ 2.3 hrs)	1
POD30	3.958×10^{-3}	1.351×10^2	1.486×10^3	1/6
POD30/DEIM30	3.164×10^{-2}	0.858	9.440	1/890
POD30/DEIM50	6.016×10^{-3}	0.924	10.169	1/826
POD50	3.773×10^{-3}	2.452×10^2	2.697×10^3	1/3
POD50/DEIM50	5.550×10^{-3}	1.154	12.692	1/662

Table 5.3: Average relative error (2-norm) of c and CPU time (sec) for solving 11 runs: $\mathrm{Pe} = 110, 111, \ldots, 120$, with the ratios of the CPU time normalized by the time of the full-order system.

convection-diffusion-reaction equations as shown in [34]. Let a, b, c be the concentrations of the two reactants A and B and of the product C; and D_A, D_B, D_C be constant diffusion coefficients of A, B, C, with viscosity $\mu(c) := \mu_0 e^{R(c/c_0)}$, where R is the log-mobility ratio. When $R > 0$, a more viscous product C is produced at the interface and the less viscous reactant pushes the more viscous product as shown in Figure 5.5. The numerical technique presented in Section 5.2 is used for this experiment. The dimensionless parameters (additional to the previous cases) are the ratios of the diffusion coefficients of A and B: $\delta_A = D_A/D_C$, $\delta_B = D_B/D_C$.

The numerical results presented here use parameters: $R = 3$, Pe $= 250$, $Le = 1$, Da $= 1$, $d = 0.1$, $\delta_A = 1$, $\delta_B = 5$, with aspect ratio $\alpha = 3$. Periodic boundary conditions are used in both x and y coordinates. Initially, the reactant B is sandwiched between the reactant A. Figure 5.5 illustrates the concentrations of A, B and C in a 2-D homogeneous porous medium at time $t = 500$. Similar to previous numerical cases, it shows that the POD-DEIM reduced model with POD and DEIM of dimension 30 and 40 can accurately capture the VF dynamics of the full-order system having dimension 15000 with substantially less CPU time, i.e., $\mathcal{O}(1000)$ reduction, as shown in Table 5.4. Note that this system is more complex than the previous cases due to the number of variables, as well as the nonlinear reaction terms. This type of nonlinear system is influenced by various parameters (e.g., Pe, δ_A, δ_B, Da) and the parametric study therefore becomes an important tool and a common method for analyzing the dynamics of this system as done in [34]. Hence, the POD-DEIM is a

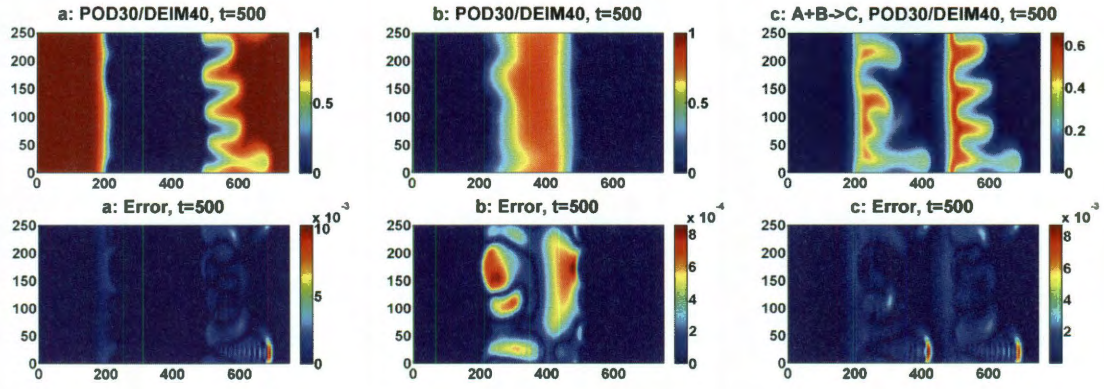


Figure 5.5: Concentration plots in the flow domain of reactants A, B and the product C from the reaction $A + B \rightarrow C$ at time $t = 500$ from the POD-DEIM reduced system with POD and DEIM having dimensions 30 and 40, with the corresponding absolute errors at the grid points when compared to the full-order system of dimension 15000 (fixed parameters).

promising technique for improving the efficiency of the simulation for this parametric study.

5.6 Conclusions and Remarks

The model reduction technique combining POD with DEIM has been shown to be efficient for capturing the dynamics in the VF simulation with substantial reduction in dimension and computational time. The failure to decrease complexity with the standard POD technique was clearly demonstrated by the comparative computational times shown in, e.g., the plot of CPU time in Figure 5.3. DEIM was shown to be very effective in overcoming the deficiencies of POD with respect to *general* nonlinearities in VF simulation. The preliminary numerical results in the previous section provide

Dimension	Avg Rel Error of concentrations	CPU time (sec)	Ratio CPU
Full 15000 (FD)	-	1.699×10^3	1
POD10	4.561×10^{-3}	1.757×10^2	1/10
POD10/DEIM10	8.255×10^{-3}	1.612	1/1054
POD20	9.131×10^{-4}	3.057×10^2	1/6
POD20/DEIM20	3.267×10^{-3}	1.970	1/862
POD30	4.006×10^{-4}	4.435×10^2	1/4
POD30/DEIM40	8.382×10^{-4}	2.567	1/661
POD40	3.162×10^{-4}	6.325×10^2	1/3
POD40/DEIM40	4.867×10^{-4}	2.791	1/609

Table 5.4: Average relative error (2-norm) of the solution for the concentrations a, b, c of the reactants A, B, and the product C and CPU time of the full-order system, POD reduced system, and POD-DEIM reduced system (fixed parameters) with the ratios of the CPU time normalized by the time of the full-order system.

a promising extension of the POD-DEIM approach to speed up the VF simulations in parametric study.

Note that, in Section 5.5.2, the variation of Péclet number is only considered in a relatively small range. It is possible to consider varying multiple parameters at the same time with a larger range for each of them as done in, e.g., [27, 39]. The framework presented here can still be used with only minor modifications. In general, the quality of the sampled snapshots can affect the efficiency of the POD-DEIM approximation. In this chapter, the snapshots are selected uniformly over the sampled space. It is possible to apply more efficient algorithms for selecting snapshots, such as those proposed in [15, 60, 41]. While this possibility has not been considered here, I hope to investigate this, as well as the other issues discussed above. These issues still remain as challenging research topics and will be left for future work.

Chapter 6

Conclusions and Future Work

This thesis developed a model reduction technique for general large-scale nonlinear ODE systems by combining POD with DEIM, as described in Chapter 2. DEIM was demonstrated to overcome the deficiencies of POD with respect to general nonlinearities. An error bound for the DEIM approximation of a nonlinear vector-valued function was proposed in Lemma 2.2.3, showing the obtained approximation to be nearly optimal. The state space error bounds of the POD-DEIM reduced systems for the ODEs with Lipschitz continuous nonlinearities were derived in Chapter 3. The analysis was particularly relevant to ODE systems arising from spatial discretizations of parabolic PDEs. These error bounds were considered in both continuous and discrete settings, and they were derived through a standard approach using logarithmic norms, as well as through an application of generalized logarithmic norms [81]. The conditions under which the reduction error is uniformly bounded were also discussed.

The resulting error bounds in the $\mathcal{L}^2$-norm reflect the approximation property of the POD based scheme through the decay of the corresponding singular values. These bounds clearly explain the *stagnation* of the errors observed in the numerical results shown in Chapters 4 and 5 (see e.g., Figs 4.4, 4.9, 5.3). Moreover, for some simple problems, these bounds can be used for determining a *suitable* dimension (k, m) for the POD-DEIM approximation, as illustrated in Appendix B.

The numerical results in Chapter 4 illustrate that the POD-DEIM approach not only gives an accurate reduced system that is substantially smaller than the original system with a general nonlinearity, but it also preserves the steady state behavior (e.g., the limit cycle) of the original system. The average errors for the POD-DEIM approach in Figures 4.4 and 4.9 show that the accuracy of the approximation depends on the dimensions of both POD and DEIM. An application of POD-DEIM approach to two-phase miscible flow in 2-D porous media presented in Chapter 5 was demonstrated to be efficient for capturing the complex dynamics of the original system, with substantial reduction in dimension and computational time. The failure to decrease complexity with the standard POD technique was clearly demonstrated by the comparative computational times shown in, e.g., the plot of CPU time in Figure 5.3.

Current and Future Research

- **Adaptive POD basis:** Due to the data-dependent nature of the POD basis, the POD-DEIM approach generally cannot be expected to give good approx-

imations for systems with parameters lying outside the sampling parameter domains from which the POD basis is constructed. One possible way to handle this issue is to develop an adaptive framework that incorporates the scheme for efficiently updating the POD basis to improve the accuracy of the reduced-order systems.

- **Extending error and stability analysis:** The error analysis given in Chapter 3 mainly provides the theoretical insight into the factors contributing to the accuracy of the POD-DEIM technique for a certain class of nonlinear dynamical systems. It therefore still remains to perform sensitivity and stability analysis, as well as to extend this error analysis to a boarder class of nonlinear parametrized problems. It is also important to investigate an alternative error estimate that is useful in practice, in the sense that it can be efficiently computed in addition to accurately predicting the error.

- **Constructing a POD-DEIM reduced system for a nonlinear model using snapshots from linear or linearized models:** The POD basis is generally derived from a set of sampled solution trajectories (snapshots) from the original large-scale nonlinear systems. These snapshots therefore could be very expensive to obtain. To reduce this computational cost, the corresponding simplified linear or linearized models could be used instead to generate these snapshots. This idea is shown to be promising through preliminary results obtained from its application on a model of polymer dynamics.

It is important to emphasize the future investigation of the stability issue for the POD-DEIM approach, as listed above. Recently, this issue has come to the forefront in some practical large-scale problems. I hope that the error analysis given in this thesis, particulary in the generalized setting of logarithmic Lipschitz constant [82], will give a good starting point for this investigation.

In addition to the items given above, other possible future research includes: incorporating the POD-DEIM technique with higher-order numerical scheme; developing simulation software based on the POD-DEIM procedure integrated with existing ODE solvers for different classes of nonlinear dynamical systems; combining DEIM with other projection-based model reduction techniques such as Krylov-based approximation methods; and applying this method to other applications such as optimization and uncertainty analysis. The extensions discussed in this section will allow a boarder impact on model reduction for practical large-scale nonlinear problems.

Bibliography

[1] D. Amsallem, J. Cortial, K. Carlberg, and C. Farhat. A method for interpolating on manifolds structural dynamics reduced-order models. *International Journal for Numerical Methods in Engineering*, 80(9):1241–1258, 2009.

[2] D. Amsallem and C. Farhat. An interpolation method for adapting reduced-order models and application to aeroelasticity. *AIAA Journal*, 46(7):1803–1813, 2008.

[3] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. C. Sorensen. *LAPACK Users' Guide Third Edition.* SIAM, 1999.

[4] A. C. Antoulas, D. C. Sorensen, and S. Gugercin. A survey of model reduction methods for large-scale systems. *Contemporary Mathematics*, 280:193–219, 2001.

[5] P. Astrid. Fast reduced order modeling technique for large scale LTV systems. In *Proceedings of the 2004 American Control Conference*, volume 1, pages 762–767, 30June- 2July 2004.

[6] P. Astrid. *Reduction of process simulation models: a proper orthogonal decomposition approach.* PhD thesis, Department of Electrical Engineering, Eindhoven University of Technology, November 2004.

[7] P. Astrid and S. Weiland. On the construction of pod models from partial observations. In *CDC-ECC 05 44th IEEE Conference on Decision and Control and 2005 European Control Conference*, pages 2272–2277, Dec 2005.

[8] P. Astrid, S. Weiland, K. Willcox, and T. Backx. Missing point estimation in models described by proper orthogonal decomposition. In *CDC 43rd IEEE Conference on Decision and Control*, volume 2, pages 1767–1772, Dec 2004.

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

[10] Z. Bai. Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems. *Applied Numerical Mathematics*, 43(1-2):9–44, 2002.

[11] 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(9):667–672, 2004.

[12] T. Bechtold, M. Striebel, K. Mohaghegh, and E. J. W. ter Maten. Nonlinear Model Order Reduction in Nanoelectronics: Combination of POD and TPWL. *PAMM*, 8(1):10057–10060, 2008.

[13] R. Bellman. The stability of solutions of linear differential equations. *Duke Math. J.*, 10(4):643–647, 1943.

[14] T. Bui-Thanh, M. Damodaran, and K. Willcox. Aerodynamic Data Reconstruction and Inverse Design using Proper Orthogonal Decomposition. *AIAA Journal*, 42(8):1505–1516, August 2004.

[15] T. Bui-Thanh, K. Willcox, and O. Ghattas. Model Reduction for Large-Scale Systems with High-Dimensional Parametric Input Space. *SIAM J. Sci. Comput*, 30(6):3270–3288, 2008.

[16] M. A. Cardoso and L. J. Durlofsky. Linearized reduced-order models for subsurface flow simulation. *Journal of Computational Physics*, 229(3):681–700, 2010.

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

[18] K. Carlberg and C Farhat. A low-cost, goal-oriented compact proper orthogonal decomposition basis for model reduction of static systems. *International Journal for Numerical Methods in Engineering*, 2010.

[19] S. Chaturantabut. Dimension Reduction for Unsteady Nonlinear Partial Differential Equations via Empirical Interpolation Methods. Master's thesis, Rice University, 2008.

[20] Y. Chen. Model Order Reduction for Nonlinear Systems. Master's thesis, Massachusetts Institute of Technology, 1999.

[21] Y. Chen and J. White. A Quadratic Method for Nonlinear Model Order Reduction. In *Technical Proceedings of the 2000 International Conference on Modeling and Simulation of Microsystems*, pages 477–480, 2000.

[22] D. S. Clark. Short proof of a discrete Gronwall inequality. *Discrete Applied Mathematics*, 16(3):279 – 281, 1987.

[23] S. J. Cox and L. Ji. Discerning ionic currents and their kinetics from input impedance data. *Bulletin of Mathematical Biology*, 63:909–932, 2001. 10.1006/bulm.2001.0250.

[24] G. Dahlquist. Stability and error bounds in the numerical integration of ordinary differential equations. *Transactions of the Royal Institute of Technology 130, Stockholm, Sweden*, 1959.

[25] G. Dahlquist. 33 years of numerical instability, part i. *BIT Numerical Mathematics*, 25:188–204, 1985. 10.1007/BF01934997.

[26] J. W. Daniel, W. B. Gragg, L. Kaufman, and G. W. Stewart. Reorthogonalization and Stable Algorithms for Updating the Gram-Schmidt QR Factorization. *Mathematics of Computation*, 30(136):772–795, 1976.

[27] S. Deparis and G. Rozza. Reduced basis method for multi-parameter-dependent steady navier-stokes equations: Applications to natural convection in a cavity. *Journal of Computational Physics*, 228(12):4359 – 4378, 2009.

[28] N. Dong and J. Roychowdhury. Piecewise polynomial nonlinear model reduction. pages 484–489, Los Alamitos, CA, USA, 2003. IEEE Computer Society.

[29] J.F.M. Van Doren, R. Markovinović, and J. D. Jansen. Accelerating iterative solution methods using reduced-order models as solution predictors. *Computational Geosciences*, 10:137–158, 2006.

[30] J. L. Eftang, D. J. Knezevic, and A. T. Patera. An hp certified reduced basis method for parametrized parabolic partial differential equations. *Mathematical and Computer Modelling of Dynamical Systems (to appear)*, 2010.

[31] R. Everson and L. Sirovich. Karhunen-Loeve procedure for gappy data. *Journal of the Optical Society of America A*, 12:1657–1664, August 1995.

[32] F.Ebert. A note on pod model reduction methods for daes. *Preprint 06-364 (Matheon),Inst. f. Mathematik, TU Berlin*, 2006.

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

[34] T. Gérard and A. De Wit. Miscible viscous fingering induced by a simple $A+B \rightarrow C$ chemical reaction. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 79(1), 2009.

[35] R. Gharbi, F. Qasem, and N. Smaoui. Characterizing Miscible Displacements in Heterogeneous Reservoirs Using the KL Decomposition. *Petroleum Science And Technology*, 21(5,6):747–776, 2003.

[36] R Gharbi, N Smaoui, and E.J. Peters. Prediction of unstable fluid displacements in porous media using the karhunen-loève decomposition. *In Situ*, 21(4):331–356, 1997.

[37] J. R. Gilbert, C. Moler, and R. Schreiber. Sparse matrices in matlab: design and implementation. *SIAM J. Matrix Anal. Appl.*, 13(1):333–356, 1992.

[38] M. A. Grepl, Y. Maday, N. C. Nguyen, and A. T. Patera. Efficient Reduced-Basis Treatment of Nonaffine and Nonlinear Partial Differential Equations. *Mathematical Modelling and Numerical Analysis*, 41(3):575–605, 2007.

[39] M. A. Grepl and A. T. Patera. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *M2AN*, 39(1):157–181, 2005.

[40] T. H. Gronwall. Note on the derivatives with respect to a parameter of the solutions of a system of differential equations. *The Annals of Mathematics*, 20(4):292–296, 1919.

[41] B. Haasdonk and M. Ohlberger. Adaptive basis enrichment for the reduced basis method applied to finite volume schemes. *In Proc. 5th International Symposium on Finite Volumes for Complex Applications,June 08–13, Aussois, France*, 2008.

[42] T. Heijn, R. Markovinović, and J.D. Jansen. Generation of low-order reservoir models using system-theoretical concepts. *SPE Journal*, 9:202–218, 2004.

[43] C. Homescu, L. R. Petzold, and R. Serban. Error estimation for reduced-order models of dynamical systems. *SIAM Journal on Numerical Analysis*, 43(4):1693–1714, 2005.

[44] C. Homescu, L. R. Petzold, and R. Serban. Error estimation for reduced-order models of dynamical systems. *SIAM Review*, 49(2):277–299, 2007.

[45] M. N. Islam and J. Azaiez. Fully implicit finite difference pseudo-spectral method for simulating high mobility-ratio miscible displacements. *International Journal for Numerical Methods in Fluids*, 47(2):161–183, 2005.

[46] K. Ito and S. S. Ravindran. A Reduced Order Method for Simulation and Control of Fluid Flows. *Journal of Computational Physics*, 143(2):403–425, 1998.

[47] Jim Douglas , Jr. Finite difference methods for two-phase incompressible flow in porous media. *SIAM Journal on Numerical Analysis*, 20(4):681–696, 1983.

[48] K. Kunisch and S. Volkwein. Control of the Burgers Equation by a Reduced-Order Approach Using Proper Orthogonal Decomposition. *J. Optim. Theory Appl.*, 102(2):345–371, 1999.

[49] K. Kunisch and S. Volkwein. Galerkin Proper Orthogonal Decomposition Methods for Parabolic Problem. *Numerische Mathematik*, 90(1):117–148, November 2001.

[50] K. Kunisch and S. Volkwein. Galerkin Proper Orthogonal Decomposition Methods for a General Equation in Fluid Dynamics. *SIAM J. Numer. Anal.*, 40(2):492–515, 2002.

[51] K. Kunisch and S. Volkwein. Proper Orthogonal Decomposition for Optimality Systems. *Mathematical Modelling and Numerical Analysis*, 42(1):1–23, 2008.

[52] R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK Users' Guide: Solution of Large-scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods.* SIAM, 1998.

[53] C. F. Van Loan. The sensitivity of the matrix exponential. *SIAM Journal on Numerical Analysis*, 14:971–981, 1977.

[54] L. Machiels, Y. Maday, I. B. Oliveira, A. T. Patera, and D. V. Rovas. Output bounds for reduced-basis approximations of symmetric positive definite eigenvalue problems. *Comptes Rendus de l'Acadmie des Sciences - Series I - Mathematics*, 331(2):153 – 158, 2000.

[55] Y. Maday, A. T. Patera, and G. Turinici. A priori convergence theory for reduced-basis approximations of single-parameter elliptic partial differential equations. *J. Sci. Comput.*, 17(1-4):437–446, 2002.

[56] R. Markovinović and J. D. Jansen. Accelerating iterative solution methods using reduced-order models as solution predictors. *International Journal for Numerical Methods in Engineering*, 68(5):525–541, 2006.

[57] J. McPhee and W. W.-G. Yeh. Groundwater management using model reduction via empirical orthogonal functions. *Journal of Water Resources Planning and Management*, 134(2):161–170, 2008.

[58] M. Meyer and H. G. Matthies. Efficient model reduction in non-linear dynamics using the karhunen-love expansion and dual-weighted-residual methods. *Computational Mechanics*, 31:179–191, 2003. 10.1007/s00466-002-0404-1.

[59] M. Mishra, M. Martin, and A. De Wit. Differences in miscible viscous fingering of finite width slices with positive or negative log-mobility ratio. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 78(6), 2008.

[60] N. C. Nguyen. A posteriori error estimation and basis adaptivity for reduced-basis approximation of nonaffine-parametrized linear elliptic partial differential equations. *J. Comput. Phys.*, 227(2):983–1006, 2007.

[61] N. C. Nguyen, A. T. Patera, and J. Peraire. A "Best Points" Interpolation Method for Efficient Approximation of Parametrized Functions. *Int. J. Numer. Meth. Engng*, 73:521–543, 2007.

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

[63] N. C. Nguyen, G. Rozza, and A. T. Patera. Reduced basis approximation and a posteriori error estimation for the time-dependent viscous burgers equation. *Calcolo*, 46(3):157–185, June 2009.

[64] A. K. Noor and J. M. Peters. Reduced Basis Technique for Nonlinear Analysis of Structures. *AIAA J.*, 19:455–462, 1980.

[65] H. G. Park and M. Zak. Model reconstruction using POD method for gray-box fault detection. *Aerospace Conference, 2003. Proceedings. 2003 IEEE*, 7:3087–3093, 8-15, 2003.

[66] P. S. Peterson. The Reduced-Basis Method for Incompressible Viscous Flow calculations. *SIAM J. Scientific and Statistical Computing*, 19:777–786, 1989.

[67] J. R. Phillips. Projection frameworks for model reduction of weakly nonlinear systems. In *DAC '00: Proceedings of the 37th Annual Design Automation Conference*, pages 184–189, New York, NY, USA, 2000. ACM.

[68] S. Prajna. POD Model Reduction with Stability Guarantee. *Decision and Control. 42nd IEEE Conference*, 5:5254–5258, Dec. 2003.

[69] C. Prud'homme, D. V. Rovas, K. Veroy, L. Machiels, Y. Maday, A. T. Patera, and G. Turinici. Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods. *Journal of Fluids Engineering*, 124(1):70–80, 2002.

[70] M. Rathinam and L. R. Petzold. A new look at proper orthogonal decomposition. *SIAM Journal on Numerical Analysis*, 41(5):1893–1925, 2003.

[71] M. Rewienski and J. White. A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices. *Computer-Aided Design, International Conference*, page 252, 2001.

[72] M. Rewienski and J. White. A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions*, 22(2):155–170, Feb 2003.

[73] M. Rewienski and J. White. Model order reduction for nonlinear dynamical systems based on trajectory piecewise-linear approximations. *Linear Algebra and its Applications*, 415(2-3):426–454, 2006. Special Issue on Order Reduction of Large-Scale Systems.

[74] M. J. Rewieński. *A Trajectory Piecewise-Linear Approach to Model Order Reduction of Nonlinear Dynamical Systems.* PhD thesis, Massachusetts Institute of Technology, 2003.

[75] B. Riviere and M.F. Wheeler. Miscible displacement in porous media. *Computational Methods in Water Resources, Developments in Water Science*, pages 907–914, 2002.

[76] A. Rocsoreanu, C.and Georgescu and N. Giurgiteanu. *The FitzHugh-Nagumo Model: Bifurcation and Dynamics.* Springer, 2000.

[77] C. W. Rowley. Model Reduction for Fluids, using Balanced Proper Orthogonal Decomposition. *International Journal of Bifurcation and Chaos (IJBC)*, 15(3):997–1013, 2005.

[78] C. W. Rowley, T. Colonius, and R. M. Murray. Model Reduction for Compressible Flows using POD and Galerkin Projection. *Physica D: Nonlinear Phenomena*, 189(1-2):115– 129, 2004.

[79] N. Smaoui and A. A. Garrouch. A new approach combining Karhunen-Love decomposition and artificial neural network for estimating tight gas sand permeability. *Journal of Petroleum Science and Engineering*, 18(1-2):101 – 112, 1997.

[80] N. Smaoui and R. Gharbi. Modelling Miscible Fluid Displacements in Porous Media Using Karhunen-Loéve Decomposition and Artificial Neural Networks. *Applied Mathematical Modelling*, 24:657–675, 2000.

[81] G. Söderlind. Bounds on nonlinear operators in finite-dimensional Banach spaces. *Numerische Mathematik*, 50:27–44, 1986. 10.1007/BF01389666.

[82] G. Söderlind. The logarithmic norm. history and modern theory. *BIT Numerical Mathematics*, 46:631–652, 2006. 10.1007/s10543-006-0069-9.

[83] T. Ström. On logarithmic norms. *SIAM Journal on Numerical Analysis*, 12(5):741–753, 1975.

[84] S. Swernath and S. Pushpavanam. Viscous fingering in a horizontal flow through a porous medium induced by chemical reactions under isothermal and adiabatic conditions. *The Journal of Chemical Physics*, 127(20), 2007.

[85] D. B. Szyld. The Many Proofs of an Identity on the Norm of Oblique Projections. *Numerical Algorithms*, 42:309–323, 2006.

[86] C. T. Tan and G. M. Homsy. Stability of miscible displacements in porous media: Rectilinear flow. *Physics of Fluids*, 29(11):3549–3556, 1986.

[87] C. T. Tan and G. M. Homsy. Simulation of nonlinear viscous fingering in miscible displacement. *Physics of Fluids*, 31(6):1330–1338, 1988.

[88] S. Utku, J. L. M. Clemente, and M. Salama. Errors in reduction methods. *Computers & Structures*, 21(6):1153–1157, 1985.

[89] A. Verhoeven. *Redundancy Reduction of IC Models by Multirate Time-Integration and Model Order Reduction.* PhD thesis, Department of Mathematics and Computer Science, Eindhoven University of Technology, 2008.

[90] P. T. M. Vermeulen, A. W. Heemink, and C. B. M. Te Stroet. Reduced models for linear groundwater flow models using empirical orthogonal functions. *Advances in Water Resources*, 27(1):57 – 69, 2004.

[91] P. T. M. Vermeulen, A. W. Heemink, and J. R. Valstar. Inverse modeling of groundwater flow using model reduction. *Water Resour. Res.*, 41:W06003, 2005.

[92] P. T. M. Vermeulen, C. B. M. Te Stroet, and A. W. Heemink. Model inversion of transient nonlinear groundwater flow models using model reduction. *Water Resour. Res.*, 42:W09417, 2006.

[93] K. Veroy, D. V. Rovas, and A. T. Patera. A posteriori error estimation for reduced-basis approximation of parametrized elliptic coercive partial differential equations: 'convex inverse' bound conditioners. *ESAIM: Control, Optimisation and Calculus of Variations*, 8:1007–1028, 2002.

[94] S. Volkwein. Model reduction using proper orthogonal decomposition. Lecture note, April 2008. http://www.uni-graz.at/imawww/volkwein/POD.pdf.

[95] K. Willcox. Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition. *Computers & Fluids*, 35(2):208–226, 2006.

[96] K. Willcox and J. Peraire. Balanced model reduction via the proper orthogonal decomposition. *AIAA Journal*, 40(11):2323–2330, 2002.

Appendix A

Computational Complexity Details

Additional details on computational complexity in Section 2.2.6 of Chapter 2 will be presented here. Tables A.1 and Table A.2 give the computational complexity for each iteration when solving (2.60) by the forward Euler method and (2.61) by Newton's method, respectively. The corresponding plots of these tables are shown in Figures A.1 and A.3. Note that each plot in Figures A.1 to A.4 is scaled so that the value of the Flops or the CPU time for the *sparse* full-order system (sparse coefficient matrix $\mathbf{A}$) is equal to 1. Note also that $\alpha(p)$ denotes the Flops for evaluating the nonlinear function $\mathbf{F}$ at p components and $\alpha^d(p)$, used in Table A.2, denotes the Flops for evaluating derivative of the nonlinear function $\mathbf{F}$ at p components. When $\mathbf{F}$ evaluates at its input vector componentwise, $\alpha(p)$ and $\alpha^d(p)$ are linear in p. In this case, the computational complexities for evaluating one forward Euler time step and performing one Newton iteration of the full-order system, the POD reduced system,

and the POD-DEIM reduced system are shown in the last columns of Table A.1 and Table A.2, respectively.

Although the forward (explicit) Euler method may not be the best approach due to the step limiting stability issue, its cost per iteration is typical of other explicit methods, and hence it is suitable for illustration purposes. An implicit scheme would require solution of a nonlinear system at each time step. The computational complexity for each Newton iteration is shown in Table A.2. In practice, the CPU time may not be directly proportional to these predicted Flops, since there are many other factors that might affect the CPU timings [37] . However, this analysis does reflect the relative computational requirements and may be useful for predicting expected relative computational times and performance enhancements possible with DEIM.

When $\mathbf{A} \in \mathbb{R}^{n\times n}$ represents the discretization of a linear differential operator, it is usually sparse. Then, from Table A.1, the sparsity of $\mathbf{A}$ can be employed, so that the total complexity for each iteration of the full-order system becomes $\mathcal{O}(n)$ instead of $\mathcal{O}(n^2)$. Similarly, from Table A.2, the total complexity becomes $\mathcal{O}(n^2)$ instead of $\mathcal{O}(n^3)$. In this case, the total complexity of the POD reduced system can be higher than the complexity of the full-order system as shown in Figures A.1 and A.3. For example, the results in Figure A.3 for the steady-state problem with dimension of the (sparse) full-order system $n = 2500$, indicate that roughly when $k = 50$ or $nk^2 = n^2$, the computational time of the POD reduced system starts to exceed the computational time of the full-order system. This follows from Table A.2 that the

complexity $\mathcal{O}(k^3 + nk^2)$ for POD reduced system is equivalent to the complexity $\mathcal{O}(n^2)$ for the sparse full-order system when $k^2 \approx n$.

This inefficiency of the POD reduced system indeed occurs in the actual computation as shown in Figures A.2 and A.4. From Figure A.2, for the unsteady nonlinear system, the CPU time of the POD reduced system used for computing each time step exceeds the CPU time for the original system as soon as its dimension reaches 30. The same phenomenon happens for the POD reduced system of the steady-state problem as shown in Figure A.4, which illustrates the (scaled) CPU time of the highly nonlinear 2-D steady state problem introduced in Chapter 4. The corresponding POD-DEIM reduced system with both POD and DEIM having dimension 15 is order $\mathcal{O}(100)$ faster than the original system with $\mathcal{O}(10^{-4})$ accuracy. On the other hand, the POD reduced system of dimension 15 gives only $\mathcal{O}(10)$ reduction in CPU time from the original system, with roughly the same order of accuracy as the POD-DEIM reduced system. These demonstrate the inefficiency of the POD reduced system that has been remedied by the introduction of DEIM.

System	Computation in forward Euler	Complexity (1 time step)	Total Complexity For linear $\alpha(\cdot), \alpha^d(\cdot)$
Full	$\mathbf{y} \leftarrow \mathbf{y} + dt(\mathbf{A}\mathbf{y} + \mathbf{F}(\mathbf{y}))$	$2n^2 + 2n + \alpha(n)$ or $cn + \alpha(n)$ $c \sim$ sparsity of $\mathbf{A}$	▶ Dense $\mathbf{A}$: $\mathcal{O}(n^2)$ ▶ Sparse $\mathbf{A}$: $\mathcal{O}(n)$ ▶ Sparse $\mathbf{A}$ (MATLAB): $\mathcal{O}(n\log(n))$ [37]
POD	$\widehat{\mathbf{y}} \leftarrow \widehat{\mathbf{y}} + dt(\widehat{\mathbf{A}}\widehat{\mathbf{y}} + \mathbf{V}^T\mathbf{F}(\mathbf{V}\widehat{\mathbf{y}}))$	$2k^2 + 2k + \alpha(n) + 4nk$	$\mathcal{O}(k^2 + nk)$
POD-DEIM	$\widehat{\mathbf{y}} \leftarrow \widehat{\mathbf{y}} + dt(\widehat{\mathbf{A}}\widehat{\mathbf{y}} + \mathbf{B}\mathbf{F}(\mathbf{V}_{\vec{\wp}}\widehat{\mathbf{y}}))$ where $\mathbf{B} = \mathbf{V}^T\mathbf{U}\mathbf{U}_{\vec{\wp}}^{-1}$, $\mathbf{U}_{\vec{\wp}} = \mathbf{P}^T\mathbf{U}, \mathbf{V}_{\vec{\wp}} = \mathbf{P}^T\mathbf{V}$	$2k^2 + 2k + \alpha(m) + 4mk$	$\mathcal{O}(k^2 + mk)$

Table A.1: Comparison of the computational complexity for each time step of forward Euler method.

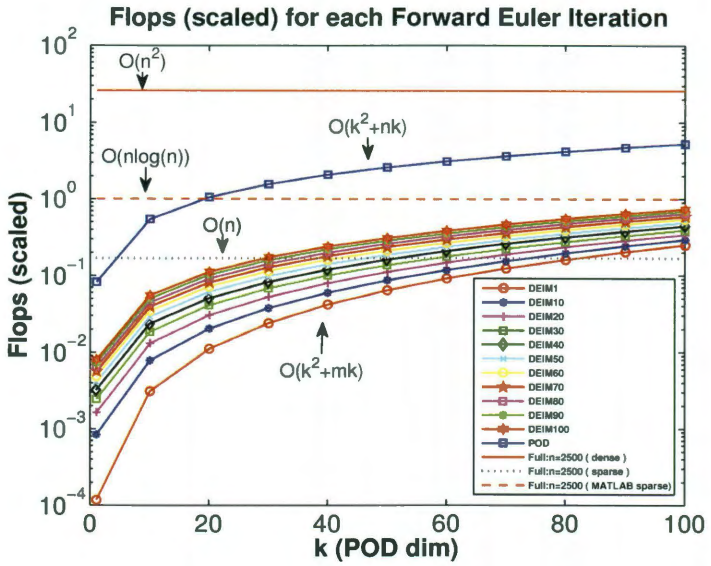


Figure A.1: Approximate Flops (scaled with Flops for the full-sparse system) for each time step of forward Euler.

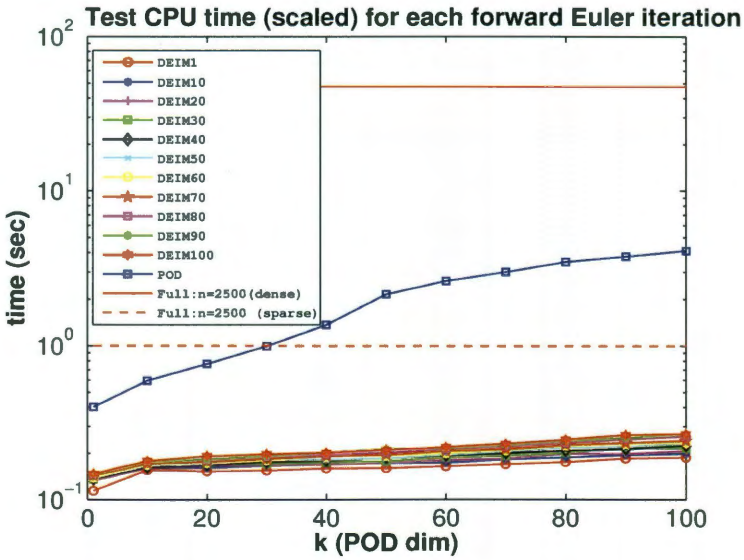


Figure A.2: Average CPU time (scaled with CPU time for the full-sparse system) for each time step of forward Euler.

System	Computation in Newton iteration	Complexity (1 iteration)	Total Complexity linear $\alpha(\cdot), \alpha^d(\cdot)$
Full	$\mathbf{G}(\mathbf{y}) = \mathbf{A}\mathbf{y} + \mathbf{F}(\mathbf{y})$ $\mathbf{J}(\mathbf{y}) = \mathbf{A} + \text{diag}\{\mathbf{F}'(\mathbf{y})\}$ $\mathbf{y} \leftarrow \mathbf{y} - \mathbf{J}(\mathbf{y})^{-1}\mathbf{G}(\mathbf{y})$	$2n^2 + \alpha(n) + n$ or $cn + \alpha(n)$ $n^2 + \alpha^d(n)$ or $n + \alpha^d(n)$ $\mathcal{O}(n^3)$ or $\mathcal{O}(n^2)$ $c \sim$ nonzero per row of $\mathbf{A}$	$\mathcal{O}(n^3)$ Sparse: $\mathcal{O}(n^2)$
POD	$\widehat{\mathbf{G}}(\mathbf{y}) = \widehat{\mathbf{A}}\widehat{\mathbf{y}} + \mathbf{V}^T\mathbf{F}(\mathbf{V}\widehat{\mathbf{y}})$ $\widehat{\mathbf{J}}(\mathbf{y}) = \widehat{\mathbf{A}} + \mathbf{V}^T\text{diag}\{\mathbf{F}'(\mathbf{V}\widehat{\mathbf{y}})\}\mathbf{V}$ $\widehat{\mathbf{y}} \leftarrow \widehat{\mathbf{y}} - \widehat{\mathbf{J}}(\mathbf{y})^{-1}\widehat{\mathbf{G}}(\mathbf{y})$	$2k^2 + \alpha(n) + k + 4nk$ $k^2 + \alpha^d(n) + 4nk + 2nk^2$ $\mathcal{O}(k^3)$	$\mathcal{O}(k^3 + nk^2)$
POD-DEIM	$\widehat{\mathbf{G}}(\mathbf{y}) = \widehat{\mathbf{A}}\widehat{\mathbf{y}} + \mathbf{B}\mathbf{F}(\mathbf{V}_{\vec{\wp}}\widehat{\mathbf{y}})$ $\widehat{\mathbf{J}}(\mathbf{y}) = \widehat{\mathbf{A}} + \mathbf{B}\ \text{diag}\{\mathbf{F}'(\mathbf{V}_{\vec{\wp}}\widehat{\mathbf{y}})\}\mathbf{V}_{\vec{\wp}}$ $\widehat{\mathbf{y}} \leftarrow \widehat{\mathbf{y}} - \widehat{\mathbf{J}}(\mathbf{y})^{-1}\widehat{\mathbf{G}}(\mathbf{y})$ where $\mathbf{B} = \mathbf{V}^T\mathbf{U}\mathbf{U}_{\vec{\wp}}^{-1}$, $\mathbf{U}_{\vec{\wp}} = \mathbf{P}^T\mathbf{U}, \mathbf{V}_{\vec{\wp}} = \mathbf{P}^T\mathbf{V}$	$2k^2 + \alpha(m) + k + 4mk$ $k^2 + \alpha^d(m) + 4mk + 2mk^2$ $\mathcal{O}(k^3)$	$\mathcal{O}(k^3 + mk^2)$

Table A.2: Comparison of the computational complexity for each Newton iteration.

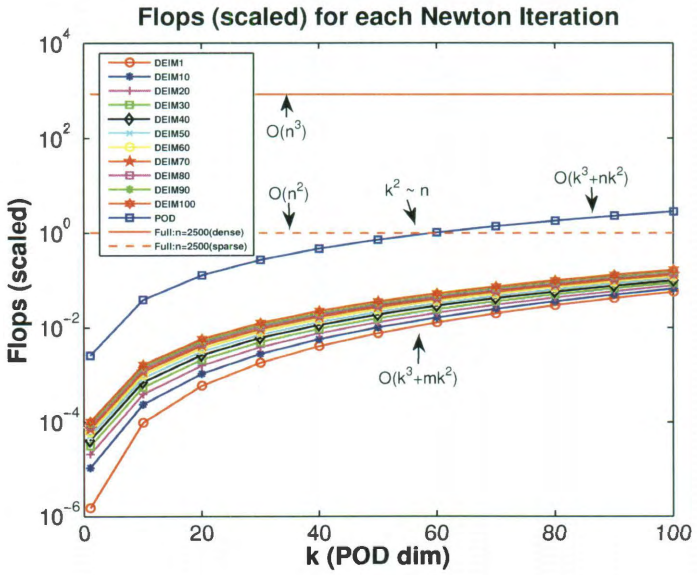


Figure A.3: Approximate Flops (scaled with Flops for the full-sparse system) for each Newton iteration from Table A.2.

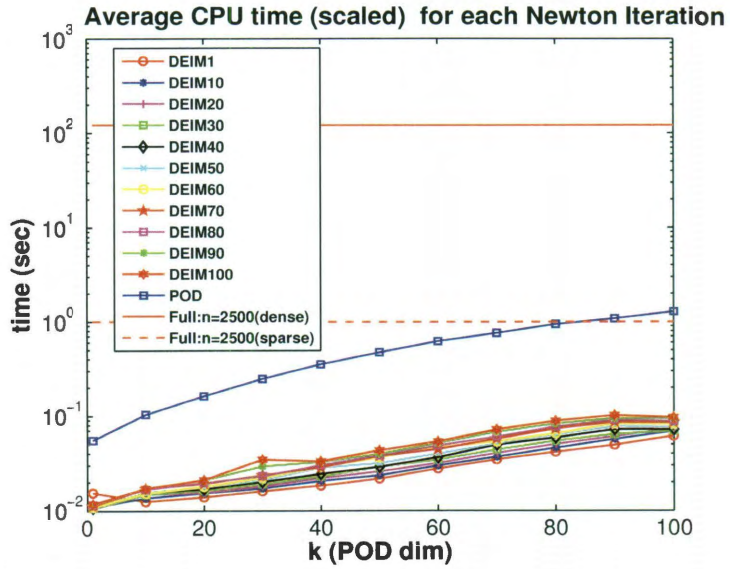


Figure A.4: Average CPU time (scaled with CPU time for the full-sparse system) for each Newton iteration for solving the steady-state 2D problem.

Appendix B

Example: State-space error bounds

The error analysis for the state-space solutions from the POD-DEIM reduced systems given in Chapter 3 mainly provides theoretical insight into the factors that contribute to the accuracy of the POD-DEIM technique. In general, these bounds may not be useful for predicting exact errors (pessimistic bounds). Applications of these bounds will be considered here through a heuristic linearization approximation.

B.1 Example: POD-DEIM Model Reduction for Finite Difference System of Burgers' Equation

Consider again the 1D unsteady Burgers' Equation:

$$\frac{\partial}{\partial t}y(x,t) = \nu\frac{\partial^2}{\partial x^2}y(x,t) - \frac{\partial}{\partial x}\left(\frac{y(x,t)^2}{2}\right) \qquad x \in [0,1], t \geq 0 \tag{B.1}$$

$$y(0,t) = y(1,t) = 0, t \geq 0 \text{ and } y(x,0) = y_0(x), x \in [0,1],$$

where $y(x,t)$ is the unknown function of time t and location $x \in \Omega \equiv [0,1]$; ν is a diffusion coefficient (viscosity parameter); and $y_0(x)$ is an initial condition. The initial condition used here is $y_0(x) = f(x) - f(0)$, where $f(x) = e^{-(15(x-0.5))^2)}$; $\nu = 0.1$; $t \in [0,1]$. Finite difference (FD) approximation for the spatial discretization gives

$$\frac{d}{dt}\mathbf{y}(t) = \nu \mathbf{A}\mathbf{y}(t) + \mathbf{F}(\mathbf{y}), \tag{B.2}$$

where $\mathbf{A} \in \mathbb{R}^{n\times n}$ is the discrete Laplace operator; $\mathbf{F}(\mathbf{y}) = -\mathbf{y}. * \mathbf{A}_x\mathbf{y}$ with first-order discrete differential operator $\mathbf{A}_x \in \mathbb{R}^{n\times n}$ and '.*' denotes pointwise multiplication (note: $-\frac{\partial}{\partial x}\left(\frac{y(x,t)^2}{2}\right) = -y(x,t)\frac{\partial y(x,t)}{\partial x}$). Here the full-order dimension n is 100. Fig. B.1 shows the solution of the full-order system and the singular values of the solution snapshots and nonlinear snapshots. The POD-DEIM reduced system is then constructed as described in Chapter 2. The accuracy of this reduced system is shown next with the approximate state error bounds from Chapter 3.

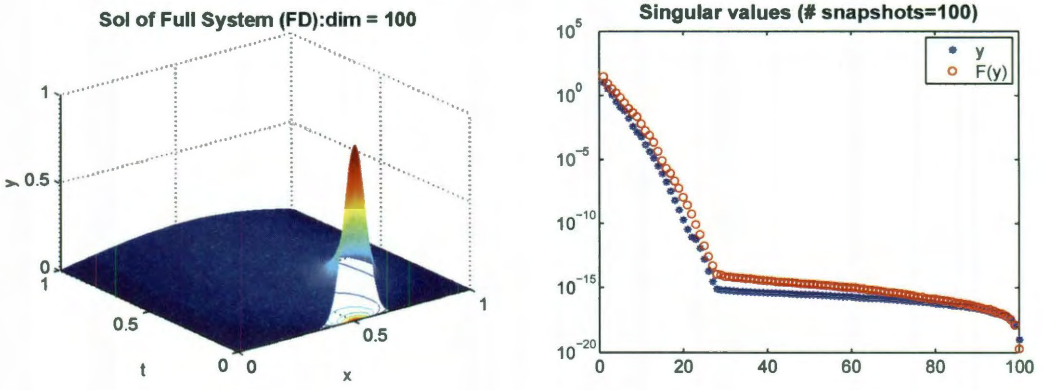


Figure B.1: Solution of Burgers' equation from full-order FD system and the singular values of 100 snapshots

B.2 Numerical Results on Approximate State-Space Error bounds

It is possible to compute realistic error bounds based on the derivation in Chapter 3 by using linearization and estimating the Jacobian (to avoid the exponential term). Fig. B.2 shows some preliminary results of these approximate error bounds for different reduced dimensions k, m for POD and DEIM. This result illustrates that the error bounds provided in this thesis can be used to determine a *suitable*[1] dimension (k, m) for the POD-DEIM reduced system.

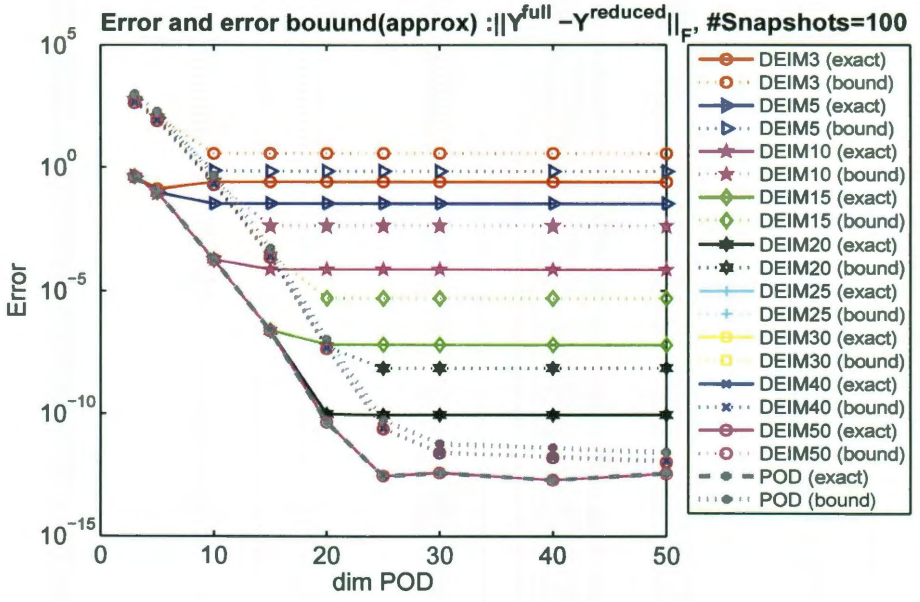


Figure B.2: Exact errors and *approximate* error bounds at 100 time steps for POD and POD-DEIM reduced systems constructed from POD bases of all 100 solution snapshots.

[1]Ideally, for a given level of accuracy, it is desirable to use the *optimal* dimension (k, m) which can be selected from the *"kinks"* or *"knees"* of the error curves.