

Model Reduction for Fast Simulation and Analysis of Dynamic Systems

Mehdi Ghommem

May 29, 2017

Table of Contents

Table of Contents	1
1 Introduction	3
2 Mode decomposition methods	9
2.1 Problem Formulation	10
2.2 Proper Orthogonal Decomposition	13
2.2.1 Mathematical background and algorithm	13
2.2.2 Numerical example: flow over cylinder	15
2.3 Dynamic Mode Decomposition	20
2.3.1 Mathematical background and algorithms	20
2.3.2 Numerical example: flow over heterogeneous porous media . .	27
2.4 Discrete Empirical Interpolation Method	29
2.4.1 Mathematical background	29
2.4.2 Numerical examples of the DEIM applications	33
Bibliography	42

Chapter 1

Introduction

The performance of multidisciplinary engineered systems (e.g., aerospace vehicles, ships, submarines, offshore structures, high-rise buildings, and wind turbines) could be significantly enhanced through understanding their governing dynamics and possibly exploiting specific phenomena. Equations governing the performance of these systems comprise nonlinear partial differential equations (PDEs). The sources of the nonlinearities in these equations may be material or constitutive, geometric, inertial, body forces, fluid-structure interactions, or friction. Recent advances in computing power in terms of speed and storage capacity (e.g. parallel computing) have enabled the use of direct numerical simulations whereby the aforementioned equations are discretized and integrated with robust numerical algorithms. For instance, higher-order finite-difference schemes on curvilinear and deforming meshes, finite volume, and finite element methods have been successfully implemented and used to solve such equations. While these high-fidelity approaches are powerful in terms of capturing the main physical features, they fail to discern the role of the underlying phenomena that are usually interrelated in a complex manner. This often makes it difficult to characterize the relevant causal mechanisms. Besides, the extensive computational resources and time associated with the use of high fidelity models usually limit the capability to simulate a large number of configurations for design purposes. These shortcomings lead to the need for the development of simplified and reduced-order

models that are capable of embodying relevant physical phenomena and of yielding relevant response characteristics. Such models would place lesser demands on the computational resources and time and (i) enable the reproduction of results of high-fidelity simulations within certain accuracy, (ii) aid in the understanding of the underlying physics and dynamics that govern the system's response, (iii) quantify system response uncertainties to variations in its parameters, and (iv) perform a rapid and reasonably accurate exploration of a large design space.

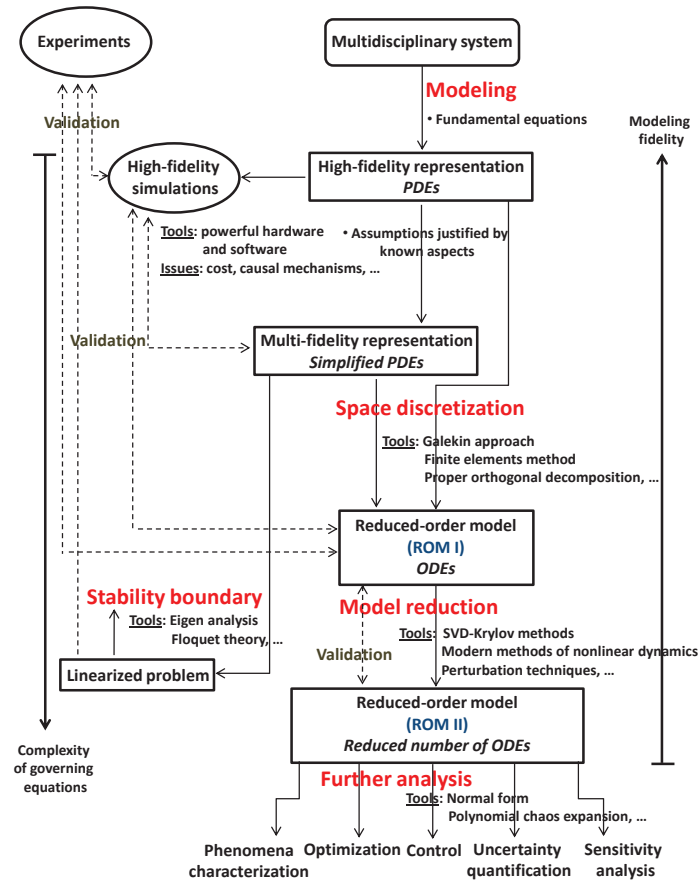


Figure 1.1: Model reduction framework of multidisciplinary systems.

The model reduction framework of multidisciplinary systems, is based on the

schematic presented in Figure 1.1. The response and performance of these systems are usually predicted by mathematical models that are derived from fundamental laws such as conservation of mass, energy, and momentum. The need to capture properly the physics and dynamics of these systems may lead to models of high complexity. Solving such models usually requires the use of powerful hardware and software capabilities. Other alternative to simulate the behavior of these systems is to conduct a full-scale experiment. The high computational cost and time associated with high-fidelity simulations and full-scale experiments present the main motivation for performing reduced-order modeling and analysis.

Depending on the nature of the system and its application, the complexity of governing equations can be significantly reduced by introducing simplifications justified by known physical characteristics. For instance, the prediction of the flutter speed and natural frequencies of a wing under an incoming freestream and the study of their variations as caused by uncertainties of different system parameters could be determined from linear analysis. In aircraft design, aerodynamic loads can be well approximated using simplified governing equations, such as the Euler equations. Furthermore, if the flow is assumed to be irrotational, the velocity field can be extracted from a potential function satisfying the Laplace equation. Such mathematical models and simplifications can be efficiently used to determine system's behavior or flow quantities. Yet, further simplifications might be needed to enable the implementation of a control strategy to reach a desired performance specification, determine optimized configurations or perform sensitivity analysis. To this end, the PDEs governing the behavior of a given dynamical system are usually discretized in the space variables to yield a system of ordinary differential equations (ODEs). The Galerkin projection constitutes a widely used solution methodology for modal discretization. For instance, for problems involving structures such as beams and plates etc, one could use the mode shapes to approximate the system response and discretize the PDEs. For systems involving fluid flows, common approaches to derive reduced-order models are proper orthogonal decomposition (POD) and dynamic mode decomposition

(DMD). The implementation of both approaches is based on representing flow quantities, such as velocity and pressure fields, by a collection of flow values of the fluid system (commonly referred as snapshots). These snapshots are originally obtained through numerical simulations or experiments. Then, a low-dimensional set of basis functions is obtained by processing snapshot data. These functions present dominant structures of the flow and that are represented by a low-dimensional dynamical system that is typically obtained by Galerkin projection. For instance, a reduced-order model for the velocity field can be obtained by projecting the Navier-Stokes equations onto the space formed by the velocity POD modes. This projection yields a set of ODEs that can be directly integrated to reconstruct the velocity field. The obtained model can be easily connected with a controller to achieve a specific behavior.

To allow for more simplified representations, one could follow two different approaches. These can be categorized into physics-based and phenomenological methods. The first model reduction category is based on governing equations of a given system. For instance, perturbation techniques, such as method of multiple scales and center manifold theorem, can be used to derive the normal form of nonlinear systems. This form constitutes an effective tool to capture the main physical behaviors of such systems near the Hopf bifurcation (flutter onset). In addition to yielding a systematic means to gain a better understanding of the physics, the normal form can also serve as the basis for additional analysis. In fact, it enables the immediate characterization of the type of dynamic instability and the identification of the effects of the system's parameters on its response. Besides, it allows for an implementation of linear and nonlinear control to delay instability and/or avoid catastrophic responses associated with the subcritical instability.

The second model reduction category is based on first analyzing and observing a particular phenomenon from experimental or numerical simulation of the dynamical behavior of a nonlinear system and then describing it with analytical functions or existing models such as self-excited oscillators. This approach requires the use of system identification and analysis tools to develop knowledge about the underlying

physics and dynamics.

In this book, we discuss different concepts and methods for characterizing and approximating the behavior of complex nonlinear dynamical systems and covers their numerical implementation. Researchers and graduate students are intended to learn and benefit from the practical use of system identification, analysis tools, and model reduction methods through motivating examples of large nonlinear multidisciplinary systems.

Chapter 2

Mode decomposition methods

Two common mode decomposition methods have been widely used for global model reduction of nonlinear dynamical systems, namely dynamic mode decomposition (DMD) and proper orthogonal decomposition (POD). Both of them are based on processing information from a sequence of snapshots (or instantaneous solutions) to identify a low-dimensional set of basis functions. These functions are then used to derive a low-dimensional dynamical system that is typically obtained by Galerkin projection [1, 2, 3, 4, 5]. Proper orthogonal decomposition constitutes a powerful mode decomposition technique for extracting the most energetic structures from a linear or nonlinear dynamical process [6, 7, 8, 9, 10, 1, 2, 3, 4, 11, 12]. Dynamic mode decomposition has been recently proposed by Schmid [13]. In comparison with POD, this technique is intended to accurately extract the coherent and dynamically relevant structures rather than selecting the dominant modes that capture most of the flow's energy. DMD enables the computation, from simulation and empirical data, of the eigenvalues and eigenvectors of a linear model that best represents the underlying dynamics, even if those dynamics are produced by a nonlinear process. One important feature of this method is its ability to extract dynamic information from flow fields without depending on the availability of a model, but rather is based on a sequence of snapshots. As such, this technique has been successfully applied to the analysis of experimental [14, 15, 16, 17, 18, 19] and numerical [20, 21, 13, 22, 23] flow field data and has shown a great capability to capture the relevant associated dynamics and

help in the characterization of relevant physical mechanisms.

When dealing with reduced-order models of nonlinear systems obtained by projecting the governing equations onto a subspace spanned by the POD and DMD modes, evaluating the projected nonlinear term can be costly since it depends on the full dimension n of the original system. As such, the reduction in the computational cost, when employing mode decomposition methods, can be limited by the full evaluation of the projected nonlinear function. To remedy to this computational issue, Chaturantabut and Sorensen [24] proposed the discrete empirical interpolation method (DEIM) in the context of nonlinear model reduction. DEIM is based on selecting a small set of points and form an approximation of the full nonlinear function by interpolation through the selected points. This method requires the function evaluation at only few selected points $m \ll n$ and enable then significant reduction in the computational cost associated with constructing projection-based reduced-order models.

2.1 Problem Formulation

We present projection-based techniques to reduce the dimension of general large-scale ordinary differential equations (ODEs) systems and parametrized algebraic equations that arise from space discretization of nonlinear partial differential equations (PDEs). We show the applicability of these techniques through generic examples of dynamical examples.

Space discretization, such as finite element, finite volume, and finite difference methods, of a nonlinear PDE with well-defined boundary and initial conditions results in a system of nonlinear ODEs of the form

$$\begin{aligned} \frac{d}{dt}\mathbf{y}(t; \mu) &= \mathbf{A}\mathbf{y}(t; \mu) + \mathbf{F}(\mathbf{y}(t; \mu)) + \mathbf{G}(\mu) \\ \mathbf{y}(0) &= \mathbf{y}_0 \end{aligned} \tag{2.1}$$

where $t \in [0, T]$ refers to time, $\mathbf{y}(t) = [\mathbf{y}_1(t), \dots, \mathbf{y}_n(t)] \in \mathbb{R}^n$, $A \in \mathbb{R}^{n \times n}$ is a discrete approximation of the linear spatial differential operator, $\mathbf{F} = [F(\mathbf{y}_1(t)), \dots, F(\mathbf{y}_n(t))] \in \mathbb{R}^n$ is a nonlinear function, and $\mathbf{G} = [\mathbf{G}_1, \dots, \mathbf{G}_n] \in \mathbb{R}^n$ is a forcing or a source term. The parameter $\mu \in \mathfrak{D} \subset \mathbb{R}^d, d = 1, 2, \dots$ denotes the systems properties (e.g., geometry, material properties, etc) or any input parameter.

Similarly, space discretization of a steady nonlinear PDEs yields a system of algebraic equations of the form

$$\mathbf{R}(\mathbf{y}(\mu)) = \mathbf{A}\mathbf{y}(\mu) + \mathbf{F}(\mathbf{y}(\mu)) + \mathbf{G}(\mu) = 0 \quad (2.2)$$

where $\mathbf{y}(\mu) = [\mathbf{y}_1(\mu), \dots, \mathbf{y}_n(\mu)] \in \mathbb{R}^n$ and \mathbf{A} , \mathbf{F} , and \mathbf{G} are defined as in Equation (2.1). The analysis of Equation (2.2) is usually followed by the computation of the Jacobian as required for characterizing the effect of varying the parameter μ on the stability of the system. The Jacobian is given by

$$\mathbf{J}(\mathbf{y}(\mu)) = \frac{\partial \mathbf{R}(\mathbf{y}(\mu))}{\partial \mathbf{y}} = \mathbf{A} + \mathbf{J}_{\mathbf{F}}(\mathbf{y}(\mu)) \quad (2.3)$$

where

$$\mathbf{J}_{\mathbf{F}}(\mathbf{y}(\mu)) = \text{diag}\left\{\frac{\partial F(\mathbf{y}_1(t))}{\partial \mu}, \dots, \frac{\partial F(\mathbf{y}_n(t))}{\partial \mu}\right\} \quad (2.4)$$

We employ projection-based techniques for constructing reduced-order models of dimension $k \ll n$ that approximates the original system governed by Equation (2.1) or (2.2). To this end, we assume a subspace spanned by a reduced basis $\Phi = \{\phi\}_{i=1}^k \in \mathbb{R}^{n \times k}$ and approximate the solution vector as

$$\mathbf{y}(t) \approx \Phi \alpha(t) \quad (2.5)$$

where $\alpha \in \mathbb{R}^k$. Replacing $\mathbf{y}(t)$ in Equation (2.1) and projecting the result onto the reduced set of basis functions Φ , we obtain the reduced-order model of the form

$$\begin{aligned} \frac{d}{dt}\alpha(t) &= \tilde{M} \tilde{A} \alpha(t) + \tilde{M} \tilde{\mathbf{F}}(\Phi \alpha(t)) + \tilde{M} \tilde{\mathbf{G}} \\ \alpha(0) &= (\Phi^T \Phi)^{-1} \Phi^T \mathbf{y}_0 \end{aligned} \quad (2.6)$$

where

$$\tilde{M} = (\Phi^T \Phi)^{-1} \in \mathbb{R}^{k \times k}, \quad \tilde{A} = \Phi^T A \Phi \in \mathbb{R}^{k \times k}, \quad \tilde{F} = \Phi^T F \in \mathbb{R}^k, \quad \tilde{G} = \Phi^T G \in \mathbb{R}^k. \quad (2.7)$$

As for numerical integration of Equation (2.6), one might apply explicit and implicit schemes (e.g., CrankNicolson method, Euler method, RungeKutta method, etc) to advance in time.

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

$$\tilde{M} \tilde{A} \alpha(\mu) + \tilde{M} \tilde{F}(\Phi \alpha(\mu)) + \tilde{M} \tilde{G} = 0 \quad (2.8)$$

Newton-Raphson method can be used to solve the above set of nonlinear algebraic equations. The corresponding reduced Jacobian is given by

$$\tilde{J}(\alpha(\mu)) = \tilde{A} + \Phi^T \mathbf{J}_F(\Phi \alpha(\mu)) \Phi \quad (2.9)$$

Once the reduced-order model is derived, as detailed above, it may be used to approximate the original system for different initial conditions, system's parameters, and input forcing assuming that the reduced basis functions $\Phi = \{\phi\}_{i=1}^k$ contain enough information about the response characteristics under varying operating conditions. Clearly, these functions or modes should be carefully selected to enable a robust implementation of the reduced-order model that satisfies the followings:

- Dynamically-relevant structures are captured in the modes ϕ_i 's.
- The error $\|y - \Phi \alpha\|$ is small.
- System properties, such as stability, are preserved.
- The procedure is computationally efficient.

The mode decomposition methods for computing the set of reduced basis functions $\Phi = \{\phi\}_{i=1}^k$ are discussed and illustrated with numerical examples in the following sections.

2.2 Proper Orthogonal Decomposition

2.2.1 Mathematical background and algorithm

The proper orthogonal decomposition (POD) is a mode decomposition method that enables the obtention of a compact representation of the data. This method allows for feature extraction by revealing relevant flow structures and model reduction by projecting high-dimensional data into a lower-dimensional space as shown above. The POD method involves the following steps.

We consider a set of snapshots $\mathbb{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_{n_s}\} \in \mathbb{R}^{n \times n_s}$. *Snapshots* refer to equispaced in time instantaneous solutions; that is, $\mathbf{y}_i = \mathbf{y}(t_i) = \mathbf{y}(i \times \Delta t)$ where Δt is the time step. A POD basis of dimension $k \leq \min(n, n_s)$ is a set of orthonormal vectors $\{\phi\}_{i=1}^k$ of which a linear combination best approximates the set \mathbb{Y} . The basis set $\{\phi\}_{i=1}^k$ is obtained by solving the following minimization problem

$$\min_{\Phi} \sum_{i=1}^{n_s} \left\| \mathbf{y}_i - \sum_{j=1}^k (\mathbf{y}_i^T \phi_j) \phi_j \right\|_2^2 \quad (2.10)$$

where the L_2 norm of an n -dimensional vector a is defined as

$$\|a\|_2 = \left(\sum_{i=1}^n a_i^2 \right)^{\frac{1}{2}}.$$

The modes ϕ_j 's are orthonormal; that is,

$$\phi_i^T \phi_j = \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \quad (2.11)$$

and are ranked by their importance in the minimization of the L_2 error given by Equation (2.10). For many physical applications, the norm is chosen so that minimizing the error corresponds physically to optimally capturing the kinetic energy (e.g., fluid flows problems). The solution of the minimization problem is obtained from the set of the left singular vectors of the snapshot matrix \mathbb{Y} . Let the singular value decomposition (SVD) of \mathbb{Y} given by

$$\mathbb{Y} = \mathbf{V} \Sigma \mathbf{W} \quad (2.12)$$

where $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_r] \in \mathbb{R}^{n \times r}$ and $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_r] \in \mathbb{R}^{n_s \times r}$ are orthogonal and $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_r)$ is the set of singular values, with $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$ and $r = \min(n, n_s)$. The POD modes (i.e., optimal solution of Equation (2.10)) are the vectors $\{\mathbf{v}_i\}_{i=1}^k$. The error between the snapshots and their approximation based on the POD modes is quantified in terms of the remaining singular values as

$$\sum_{i=1}^{n_s} \left\| \mathbf{y}_i - \sum_{j=1}^k (\mathbf{y}_i^T \phi_j) \phi_j \right\|_2^2 = \sum_{i=k+1}^r \sigma_i^2 \quad (2.13)$$

As such, the square of the singular values represents a measure of the energy content of each POD mode and thus provides guidance for the number of modes that should be considered in order to capture the relevant physics of the system.

SVD-based approach may have a limited application, especially when dealing with a large data set. For instance, direct numerical simulations (DNS) of fluid flows for which fine meshes (high resolution) imply large system's dimension and high computational cost. Alternatively, one could use the method of snapshots [25] which allows for a significant reduction of the large data sets. In this method, an eigen analysis of the correlation matrix $C = \mathbb{Y}^T \mathbb{Y}$ is performed and the POD modes are obtained as follows:

$$C = \mathbb{Y}^T \mathbb{Y} \in \mathbb{R}^{n_s \times n_s} : C u_i = \sigma_i^2 u_i \quad \text{and} \quad \phi_i = \frac{1}{\sigma_i} \mathbb{Y} u_i \quad (2.14)$$

where σ_i are referred to singular values (as introduced above) and u_i are the eigenvectors of the correlation matrix C . This formulation is well-suited for large data because the eigenvalue problem does not depend on the dimension of \mathbb{Y} but rather depends on the number of snapshots n_s which is assumed to be much smaller than the system's dimension n . The implementation of the POD technique, as described above, is detailed in **Algorithm 1**.

The selection of POD modes is based on an energy ranking (singular values) of the coherent structures. However, the energy may not in all circumstances be the appropriate measure to rank the importance of the flow structures and detect the most dynamically-relevant modes [26]. As such, other alternatives, like dynamic mode decomposition, can be considered to identify these modes.

Algorithm 1 *Proper orthogonal decomposition*

Input: a sequence of n_s snapshots $\mathbb{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_{n_s}\} \in \mathbb{R}^{n \times n_s}$
sampled equispaced in time with Δt

Output: POD modes ϕ_j and associated singular values σ_j
with $j = 1, \dots, r$

1. Collect and store the snapshots in a snapshot matrix

$$\mathbb{Y} = \begin{pmatrix} | & | & \cdots & | \\ \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_{n_s} \\ | & | & \cdots & | \end{pmatrix} \in \mathbb{R}^{n \times n_s}$$

2. Compute the correlation matrix $C = \mathbb{Y}^T \mathbb{Y} \in \mathbb{R}^{n_s \times n_s}$
3. Compute the eigenvalues and eigenvectors of the correlation matrix C
 $Cu_i = \sigma_i^2 u_i$ or $CU = U\Sigma^2$ where $U^T U = I_n$
4. Sort the eigenvalues σ_i^2 and corresponding eigenvectors u_i
in descending order

5. Select the number of modes k so that the error

$$\|E\| = \sum_{i=k+1}^r \sigma_i^2 < \epsilon \text{ where } \epsilon \ll 1$$

6. Compute the matrices U_r and Σ_r by keeping the first r
rows of U and the first r rows and columns of Σ^2 , respectively

$$U = \begin{pmatrix} U_r & \cdots \end{pmatrix} \text{ and } \Sigma^2 = \begin{pmatrix} \Sigma_r^2 & \cdots \\ \vdots & \ddots \end{pmatrix}$$

7. Compute the matrix $M = U_r \Sigma_r^{-1}$
 8. Compute the matrix $\Phi = \mathbb{Y}M$ whose columns are the POD modes
-

2.2.2 Numerical example: flow over cylinder

We present a numerical example to illustrate the effectiveness of POD-projection approach to derive reduced-order models. We perform numerical simulation of the incompressible flow past a two-dimensional circular cylinder at $Re = 180$ governed by the Navier-Stokes equations.

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p - \frac{1}{Re} \nabla^2 \mathbf{u} = 0$$

Despite its simple geometry, analyzing flow past cylinder can be helpful in understanding developing control strategies for complicated flows encountered in real engineered

applications.

POD modes

Sets of instantaneous solutions (or snapshots) of the velocity field obtained from the numerical simulations are generated and stored in snapshot matrix. Then, the POD modes ϕ_i and their corresponding singular values σ_i are obtained by applying **Algorithm 1**, as described above, to the snapshot matrix. In Figure 2.5(a), we plot the singular values of the first 20 modes obtained from a steady mean flow condition. All singular values are normalized with the corresponding value of the first mode (i.e. σ_i/σ_1). We observe that the two modes of each pair have values of the same order. These values decrease from one pair to the next in approximately a geometric progression.

To gain insight into the contribution of each mode to the total energy of the system, we define the cumulative energy as $E = \sum_{i=1}^M \sigma_i$ and the cumulative contribution of the first j modes as $c_j = (\sum_{i=1}^j \sigma_i)/E$. We remark that most of the energy is contained in the first few modes (see Figure 2.5(b)). More specifically, the first ten modes contain more than 99.9% of the total flow energy.

The streamwise and crossflow velocity modes in the steady flow are plotted in Figures 2.2 and 2.3. These modes constitute the dominant spatial structures of the flow field. Similarly to the singular values, the POD modes show a pattern of pair similarities. Looking at the first six POD modes of the streamwise and crossflow velocity components in Figures 2.2 and 2.3, respectively, it is observed that the POD modes $\phi_i^u (i = 1, 2, 5, 6)$ of the streamwise component are antisymmetric with respect to the x -axis; that is,

$$\phi_i^u(x, -y) = -\phi_i^u(x, y), \quad (2.15a)$$

while the $\phi_i^u (i = 3, 4)$ are symmetric; that is,

$$\phi_i^u(x, -y) = \phi_i^u(x, y). \quad (2.15b)$$

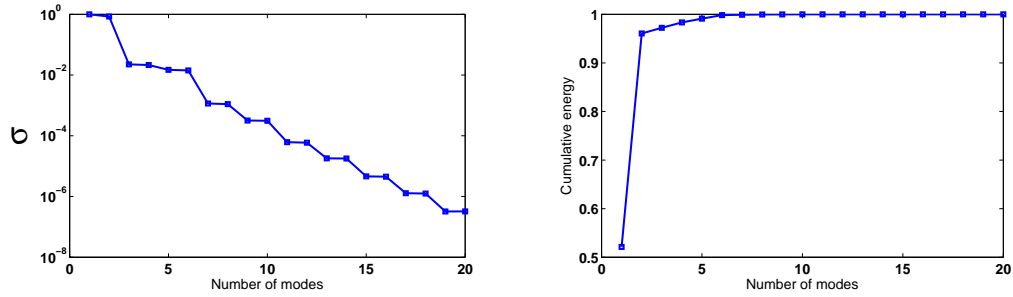
On the other hand, the POD modes ϕ_i^v of the crossflow component are symmetric for

$i = 1, 2, 5, 6$ with respect to the x -axis; that is,

$$\phi_i^v(x, -y) = \phi_i^v(x, y), \quad (2.16a)$$

and antisymmetric for $i = 3, 4$; that is,

$$\phi_i^v(x, -y) = -\phi_i^v(x, y). \quad (2.16b)$$



(a) Normalized singular values of velocity POD modes (b) Normalized cumulative energy of velocity POD modes

Figure 2.1: Normalized singular values and cumulative energy content of velocity POD modes. *Reprinted with permission of PCFD.*

Reduced-order model

As shown in Section 2.1, the first step towards deriving a reduced-order model of the velocity field is to expand the fluctuations (obtained by subtracting the mean flow) in a Galerkin representation in terms of the POD modes (Φ_i); i.e. we let

$$\mathbf{y}(t) \approx \bar{\mathbf{y}} + \Phi \alpha(t) \quad (2.17)$$

where \mathbf{y} denotes the streamwise and crossflow velocities (discrete values) at the n grid points, $\bar{\mathbf{y}}$ is the mean velocities. Then, we use the approximation given by Equation (2.17) and project the Navier-Stokes equations onto the space formed by the POD modes ϕ_i 's. Similarly to Equation (2.6), the obtained reduced system is of the form

$$\frac{d}{dt}\alpha_l(t) = \mathcal{A}_l + \sum_{i=1}^k \mathcal{B}_{li}\alpha_i(t) + \sum_{i=1}^k \sum_{j=1}^k \mathcal{C}_{lij}\alpha_i(t)\alpha_j(t), \quad \text{for } l = 1, \dots, k \quad (2.18)$$

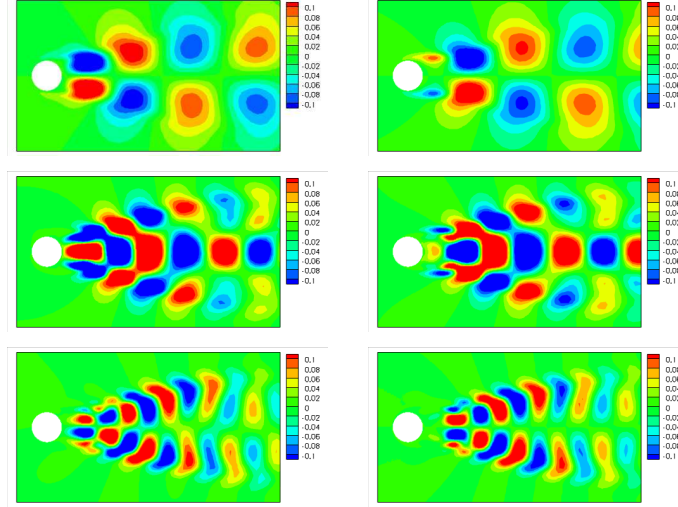


Figure 2.2: Contours of the streamwise velocity POD modes (upper frame: ϕ_i^u , $i = 1, 2$, center frame: ϕ_i^u , $i = 3, 4$, and lower frame: ϕ_i^u , $i = 5, 6$). *Reprinted with permission of PCFD.*

In Equation (2.18), \mathcal{A} is an $k \times 1$ vector resulting from the average flow field, \mathcal{B} is the linear part of the dynamical system, which originates from the interaction of the average field with the eigenfunctions and the linear dissipative operator, and \mathcal{C} is a tensor representing the quadratic nonlinearity of the Navier-Stokes equations ([27, 1]). It should be noted here that using Green's theorem and the divergence-free property and for the case of $p = 0$ on the outer flow boundary Ω_{so} , the pressure term is not part of Equation (2.18) ([28]). The POD eigenfunctions are identically zero on the inflow boundary because the average flow is subtracted from the total flow. However, in case of Neumann boundary conditions on Ω_{so} , the contribution of the pressure term is not exactly zero for the cylinder wake. At $10D$ downstream from the cylinder (D is the cylinder diameter), i.e. the outflow boundary, the pressure effects of the wake are minimal. As such, their representative terms can be assumed to vanish in the reduced-order model ([27]).

To assess the validity of the reduced-order model, given by Equation (2.18), we consider the first 10 POD modes and compare in figure 2.4 the streamwise velocity

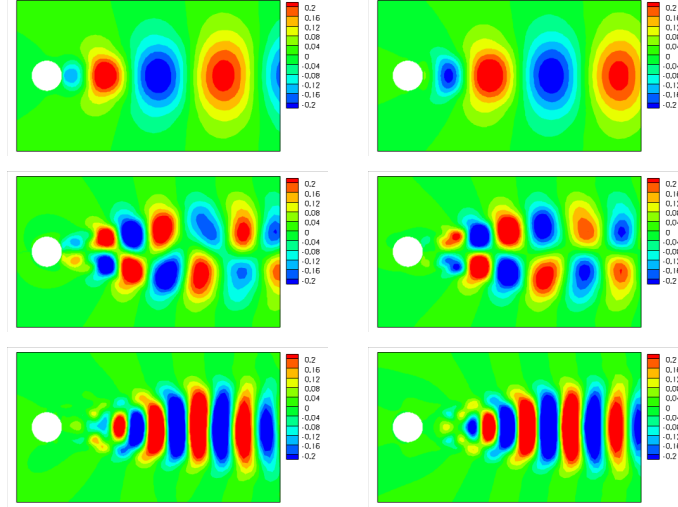


Figure 2.3: Contours of the crossflow velocity POD modes (upper frame: ϕ_i^v , $i = 1, 2$, center frame: ϕ_i^v , $i = 3, 4$, and lower frame: ϕ_i^v , $i = 5, 6$). *Reprinted with permission of PCFD.*

at a probe located near to the cylinder obtained from the POD-based reduced-order model with the DNS data in time domain. It can be clearly seen that the velocity fields constructed using the POD-based approach and the CFD simulations are in agreement. This shows the suitability of the use of POD technique for model reduction of nonlinear fluid flows. The obtained reduced-order model could serve as the basis to apply additional analyses such implementing control strategies [29, 30] and perform sensitivity analysis [11, 31, 12].

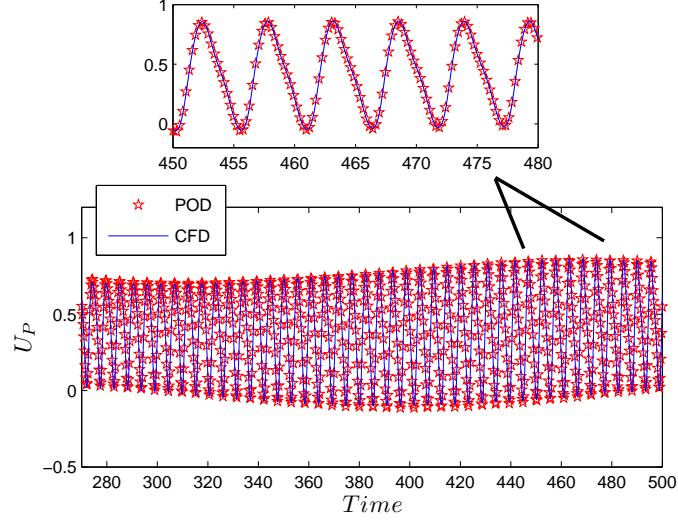


Figure 2.4: Streamwise velocity at a probe located near to the cylinder obtained from the POD-based reduced-order model with the DNS data in time domain.

2.3 Dynamic Mode Decomposition

2.3.1 Mathematical background and algorithms

Dynamic mode decomposition (DMD) has been widely applied on experimental [14, 15, 16, 17, 18, 19] and numerical [13, 21, 20, 21, 13, 22, 23, 26] flow field data to identify dominant coherent structures and help in understanding the underlying physics. Similarly as POD, these structures can be used to project a large-scale problem onto a low-dimensional subspace to obtain a dynamical system with much fewer degrees of freedom. Following, we present the basic principles and mathematical background of DMD.

The DMD method is based on postprocessing a sequence of snapshots to extract the dynamic information. Let a snapshot sequence, separated by a constant time step Δt , collected in a matrix \mathbb{Y} , as defined in Section 2.2. For the sake of subsequent

analysis, we write the snapshot matrix as

$$\mathbb{Y}_1^{n_s} = \{\mathbf{y}_1, \dots, \mathbf{y}_{n_s}\} \in \mathbb{R}^{n \times n_s} \quad (2.19)$$

where the subscript and superscript are introduced to denote the first and last elements of the snapshot sequence, respectively. The DMD method uses the Arnoldi approach to relate the solution field \mathbf{y}_i to the subsequent solution field \mathbf{y}_{i+1} through a linear mapping A ; that is,

$$\mathbf{y}_{i+1} = A\mathbf{y}_i. \quad (2.20)$$

This assumption leads to a representation of the solution field as a Krylov sequence

$$\mathbb{Y}_1^{n_s} = \{\mathbf{y}_1, A\mathbf{y}_1, A^2\mathbf{y}_1, \dots, A^{n_s-1}\mathbf{y}_1\} \quad (2.21)$$

The objective is to determine the main characteristics of the dynamical process represented by the linear mapping A (even if the solution field involves nonlinear aspects). This is performed by computing (or approximating) the eigenvectors and eigenvalues of the matrix A . For instance, if the matrix A represents the Jacobian matrix, this analysis would produce stability information (the complex eigenvalues correspond to growth rates and frequencies). For a very large system, these computations may be numerically intractable. Furthermore, for instance in experimental fluid simulations, the exact form of the matrix A is not given. As such, an efficient and fast numerical approach that approximates well the relevant dynamics would be useful. The DMD method enables the approximation of the eigenvalues and eigenvectors of the matrix A and proceeds as follows.

Assuming that for sufficiently long sequence, the vector \mathbf{y}_{n_s} can be represented by a linear combination of the previous solution fields; that is,

$$\mathbf{y}_{n_s} = \sum_{i=1}^{n_s-1} a_i \mathbf{y}_i + \mathbf{r} \quad (2.22)$$

or

$$\mathbf{y}_{n_s} = \mathbb{Y}_1^{n_s-1} \mathbf{a} + \mathbf{r} \quad (2.23)$$

where $\mathbf{a} = \{a_1, a_2, \dots, a_{n_s-1}\}^T$ and \mathbf{r} is the residual vector. Combining Equations (2.21) and (2.23) and rearranging the result, we obtain

$$A \mathbb{Y}_1^{n_s-1} = \mathbb{Y}_2^{n_s} = \mathbb{Y}_1^{n_s-1} \mathbf{S} + \mathbf{r} \mathbf{e}_{n_s-1}^T \quad (2.24)$$

where $\mathbf{e}_{n_s-1}^T = \begin{pmatrix} 0 & \dots & 0 & 1 \end{pmatrix}$ is the $(n_s - 1)$ unit vector and the matrix \mathbf{S} is a companion matrix defined as:

$$\mathbf{S} = \begin{pmatrix} 0 & & & a_1 \\ 1 & 0 & & a_2 \\ & \ddots & \ddots & \vdots \\ & & 1 & 0 & a_{n_s-2} \\ & & & 1 & a_{n_s-1} \end{pmatrix} \quad (2.25)$$

The unknown matrix \mathbf{S} is determined by minimizing the residual \mathbf{r} which is obtained by expressing the n_s^{th} snapshot \mathbf{y}_{n_s} by a linear combination of $\{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \dots, \mathbf{y}_{n_s-1}\}$ in a least-squares sense. The minimization problem to determine \mathbf{S} is expressed then as:

$$\mathbf{S} = \min_{\mathbf{S}} \|\mathbb{Y}_2^{n_s} - \mathbb{Y}_1^{n_s-1} \mathbf{S}\| \quad (2.26)$$

To avoid cumbersome notation, we use the same variable for the minimizer as the variable that is being minimized. The solution can be determined either using a QR-decomposition of the snapshot matrix $\mathbb{Y}_1^{n_s-1}$; that is [21],

$$\mathbf{S} = \mathbf{R}^{-1} \mathbf{Q}^T \mathbb{Y}_2^{n_s} \quad (2.27)$$

where the matrices \mathbf{Q} and \mathbf{R} are obtained from QR decomposition of the snapshot matrix $\mathbb{Y}_1^{n_s-1}$ or using the Moore-Penrose pseudoinverse of the matrix $\mathbb{Y}_1^{n_s-1}$ to obtain a solution [26]

$$\mathbf{S} = \left((\mathbb{Y}_1^{n_s-1})^T \mathbb{Y}_1^{n_s-1} \right)^{-1} (\mathbb{Y}_1^{n_s-1})^T \mathbb{Y}_2^{n_s} \quad (2.28)$$

Once \mathbf{S} is computed, the next step is to evaluate its eigenvalues and eigenvectors collected in the diagonal matrix \mathbf{D} , the matrix \mathbf{X} , respectively.

Finally, the DMD spectrum λ_j is obtained by transforming the eigenvalues of \mathbf{S} from the time-stepper format to the format more commonly used in stability theory (accomplished through a logarithmic mapping), and the dynamic modes ϕ_j are computed by weighing the snapshot based by the eigenvectors of \mathbf{S} ; that is,

$$\lambda_j = \log(\mathbf{D}_{jj})/\Delta t \quad (2.29)$$

$$\text{and } \phi_j = \mathbb{Y}_1^{n_s-1} \mathbf{X}_j \quad (2.30)$$

where \mathbf{X}_j is the j^{th} column of the matrix \mathbf{X} .

The above algorithm to compute the dynamic modes based on the companion matrix \mathbf{S} may be ill-conditioned in practical situation. As such, Schmid [13] proposed a more robust implementation. The algorithm includes the following steps. First, we let $\mathbb{Y}_1^{n_s-1} = \mathbf{U} \Sigma \mathbf{W}$ be the singular value decomposition of the data sequence $\mathbb{Y}_1^{n_s-1}$. Substituting the SVD representation $\mathbf{U} \Sigma \mathbf{W}$ into Equation (2.24) and multiplying the result by from the left \mathbf{U}^T and by $\mathbf{W}^T \Sigma^{-1}$ from the right, we obtain the following matrix

$$\mathbf{U}^T \mathbf{A} \mathbf{U} = \mathbf{U}^T \mathbb{Y}_2^{n_s} \mathbf{W}^T \Sigma^{-1} \equiv \tilde{\mathbf{S}} \quad (2.31)$$

For configuration sets with $\dim(\mathbf{y}_i) = n \gg n_s$, the method of snapshots, as described in Section 2.2, is usually used to avoid the computational burden associated with the singular value decomposition of large matrices. The matrix \mathbf{U} contains the POD modes of the sequence of snapshots $\mathbb{Y}_1^{n_s-1}$, and then, one may conclude that the matrix $\tilde{\mathbf{S}}$ is obtained from the projection of the linear operator A , which is used to approximate the underlying dynamical process, onto a the POD basis. A key advantage of the above implementation is the ability to take into account the rank-deficiency of $\mathbb{Y}_1^{n_s-1}$ by considering a limited basis \mathbf{U} given only by the non-zero singular values of Σ (or by singular values above a threshold that can be determined based the amount of cumulative energy that needs to be captured). The dynamic modes are computed from the matrix $\tilde{\mathbf{S}}$ as follows:

$$\phi_j = \mathbf{U} \mathbf{v}_j, \quad (2.32)$$

where \mathbf{v}_j is the j^{th} eigenvector of $\tilde{\mathbf{S}}$, i.e., $\tilde{\mathbf{S}} \mathbf{v}_j = \mu_j \mathbf{v}_j$, and \mathbf{U} is the matrix collecting the right singular vectors of the snapshot sequence $\mathbb{Y}_1^{n_s-1}$. In **Algorithm 2** and **3**, we provide the detailed steps of both approaches to compute the DMD modes.

In a recent paper, Chen et al. [26] proposed an optimized version of the DMD algorithm where they employ a global optimization technique to minimize the residual error at all snapshots instead of the error at only the last snapshot. They tested the approach over a variety of fluid problems and showed its superiority in capturing the relevant frequencies and reproducing flowfields with small projection errors.

Algorithm 2 *Dynamic mode decomposition - approach 1*

Input: a sequence of n_s snapshots $\mathbb{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_{n_s}\} \in \mathbb{R}^{n \times n_s}$
sampled equispaced in time with Δt

Output: dynamic modes ϕ_j and associated dynamic mode spectrum λ_j
with $j = 1, \dots, r$

1. Collect and store the snapshots in two consecutive snapshot sequences

$$\mathbb{Y}_1^{n_s-1} = \begin{pmatrix} | & | & & | \\ \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_{n_s-1} \\ | & | & & | \end{pmatrix} \quad \text{and} \quad \mathbb{Y}_2^{n_s} = \begin{pmatrix} | & | & & | \\ \mathbf{y}_2 & \mathbf{y}_2 & \cdots & \mathbf{y}_{n_s} \\ | & | & & | \end{pmatrix}$$

2. Compute the QR decomposition of the matrix $\mathbb{Y}_1^{n_s-1} = Q R$

3. Compute the matrix \mathbf{S} as $\mathbf{S} = R^{-1} Q^T \mathbb{Y}_2^{n_s}$
or compute the matrix \mathbf{S} using the Moore-Penrose pseudoinverse of the matrix $\mathbb{Y}_1^{n_s-1}$ to obtain a solution as

$$\mathbf{S} = \left((\mathbb{Y}_1^{n_s-1})^T \mathbb{Y}_1^{n_s-1} \right)^{-1} (\mathbb{Y}_1^{n_s-1})^T \mathbb{Y}_2^{n_s}$$

4. Compute the eigenvalues and eigenvectors of the matrix \mathbf{S}
 $\mathbf{S} \mathbf{X}_i = \mathbf{D}_{ii} \mathbf{X}_i$ or $\mathbf{S} \mathbf{X} = \mathbf{X} \mathbf{D}$ where \mathbf{D} is a diagonal matrix
 5. Compute the dynamic mode spectrum λ as $\lambda_i = \log(\mathbf{D}_{ii})/\Delta t$
 6. Compute the dynamic modes as $\phi_i = \mathbb{Y}_1^{n_s-1} \mathbf{X}_i$
-
-

To illustrate the capability of the DMD-based approach in approximating the eigenvalues of a linear operator defined by the matrix A , we consider a test case in which we pick a random matrix $A \in \mathbb{R}^{n \times n}$ and a random vector $\mathbf{y}_1 \in \mathbb{R}^n$ and construct

Algorithm 3 *Dynamic mode decomposition - approach 2*

Input: a sequence of n_s snapshots $\mathbb{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_{n_s}\} \in \mathbb{R}^{n \times n_s}$
sampled equispaced in time with Δt

Output: dynamic modes ϕ_j and associated dynamic mode spectrum λ_j
with $j = 1, \dots, r$

1. Collect and store the snapshots in two consecutive snapshot sequences

$$\mathbb{Y}_1^{n_s-1} = \begin{pmatrix} | & | & & | \\ \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_{n_s-1} \\ | & | & & | \end{pmatrix} \quad \text{and} \quad \mathbb{Y}_2^{n_s} = \begin{pmatrix} | & | & & | \\ \mathbf{y}_2 & \mathbf{y}_2 & \cdots & \mathbf{y}_{n_s} \\ | & | & & | \end{pmatrix}$$

2. Perform SVD of the snapshot matrix $\mathbb{Y}_1^{n_s-1} = \mathbf{U} \Sigma \mathbf{W}$
or compute the correlation matrix $\mathbf{C} = (\mathbb{Y}_1^{n_s-1})^T \mathbb{Y}_1^{n_s-1}$
and proceed as described in Algorithm 1
 3. Evaluate the matrix $\tilde{\mathbf{S}}$ as $\tilde{\mathbf{S}} = \mathbf{U}^T \mathbb{Y}_2^{n_s} \mathbf{W}^T \Sigma^{-1}$
 4. Compute the eigenvalues and eigenvectors of the matrix $\tilde{\mathbf{S}}$
 $\tilde{\mathbf{S}} \mathbf{X}_i = \mathbf{D}_{ii} \mathbf{X}_i$ or $\tilde{\mathbf{S}} \mathbf{X} = \mathbf{X} \mathbf{D}$ where \mathbf{D} is a diagonal matrix
 5. Compute the dynamic mode spectrum λ as $\lambda_i = \log(\mathbf{D}_{ii})/\Delta t$
 6. Compute the dynamic modes as $\phi_i = \mathbf{U} \mathbf{X}_i$
-
-

a set of solutions as:

$$\mathbb{Y}_1^{n_s} = \{\mathbf{y}_1, A\mathbf{y}_1, A^2\mathbf{y}_1, \dots, A^{n_s-1}\mathbf{y}_1\} \quad (2.33)$$

We set n_s equal to 10 and n equal to 10 and 20 and show in Figure 2.5 the exact and approximate eigenvalues λ 's of the matrix A . The approximate eigenvalues are computed following the DMD-based approach using both **Algorithm 2** and **Algorithm 3**. For $n = 10$, the DMD method reproduces exactly all the eigenvalues while for a larger system ($n = 20$) only the first n_s are predicted by the DMD method. Clearly, DMD does a good job in computing the eigenvalues of the matrix A based on a set of snapshots without a prior knowledge of the exact form of A . Further, we the robustness the second approach (**Algorithm 3**) in computing properly the dynamic

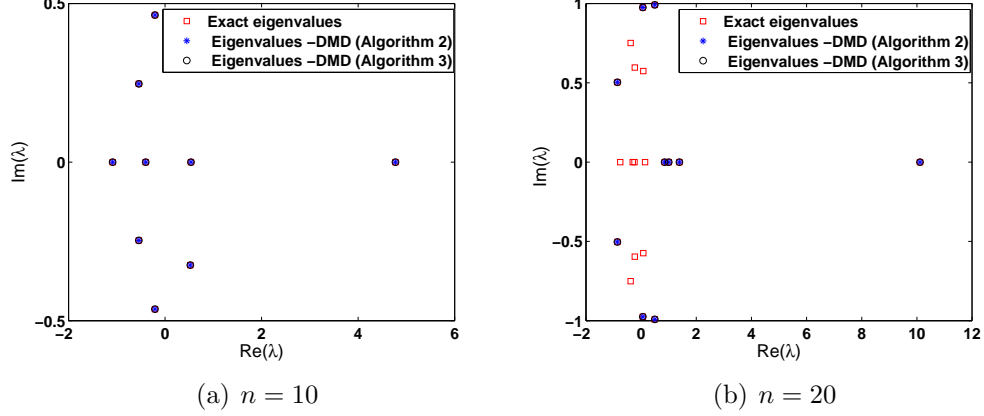


Figure 2.5: Exact and approximate eigenvalues (based on DMD formulation) of the matrix A .

modes and their associated spectrum, we consider the matrix as

$$A = B^T \begin{pmatrix} \frac{1}{n} & 0 & \cdots & 0 \\ 0 & \frac{2}{n} & 0 & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix} B \quad \text{where } B = \text{rand}(n, n) \quad (2.34)$$

We keep n_s equal to 10 and set n equal to 40. We plot in Figure 2.6 the exact and approximate eigenvalues λ 's of the matrix A . Again, the approximate eigenvalues are computed following the DMD-based approach using **Algorithm 2** and **Algorithm 3**. Clearly, the eigenvalues predicted from the DMD-based approach when using **Algorithm 2** deviate from the exact ones. On the other hand, **Algorithm 3** show robustness in computing accurately the first n_s eigenvalues of the matrix A . Thus, we use the second approach in the following numerical results.

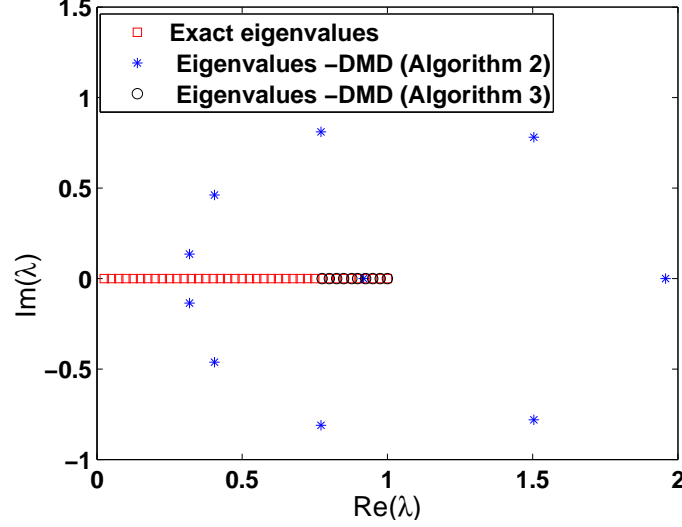


Figure 2.6: Exact and approximate eigenvalues (based on DMD formulation) of the matrix A .

2.3.2 Numerical example: flow over heterogeneous porous media

To illustrate the usefulness and effectiveness of DMD to construct reduced-order models of large dynamical systems, we consider a time-dependent, single-phase porous media flow governed by the following parabolic partial differential equation:

$$\frac{\partial p}{\partial t} - \nabla \cdot (\kappa(x) \nabla p) = f(x) \quad \text{in } \Omega, \quad (2.35)$$

where p denotes the pressure, Ω is a bounded domain, f is a forcing term, and $\kappa(x)$ is a positive-definite scalar function. The coefficient κ represents the ratio of the permeability over the fluid viscosity and is considered to be a highly-heterogeneous field with high contrast (i.e., there are large variations in the permeability). We consider a permeability field shown in Figure 2.7. It presents low-conductivity channels within an otherwise homogeneous domain. Homogeneous Dirichlet boundary conditions are assumed.

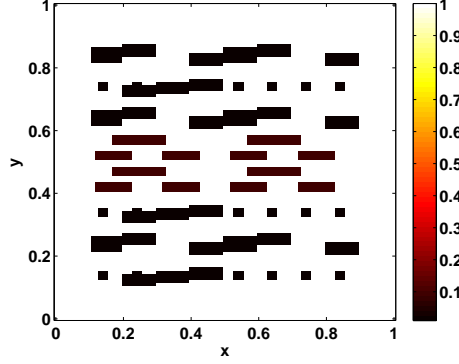


Figure 2.7: Permeability field κ ; low conductivity channels.

The finite element discretization of Equation (2.35) yields a system of ordinary differential equations given by

$$\mathbf{M}\dot{\mathbf{P}} + \mathbf{A}\mathbf{P} = \mathbf{F} \quad (2.36)$$

where \mathbf{P} and \mathbf{F} are vectors collecting the solution values and forcing at the nodes. Here, $\mathbf{A} = (a_{ij})$, $a_{ij} = \int_{\Omega} \kappa \nabla \phi_i \cdot \nabla \phi_j$, $\mathbf{M} = (m_{ij})$, $m_{ij} = \int_{\Omega} \phi_i \phi_j$, where ϕ_i is piecewise linear basis functions defined on a triangulation of Ω and ∇ denotes spatial gradient. We employ the implicit backward Euler scheme for time marching and record the first 50 (i.e., $n_s = 50$) instantaneous solutions and collect them in a snapshot matrix

$$\mathbb{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_{n_s}\} \in \mathbb{R}^{n \times n_s}$$

We postprocess the snapshot matrix to compute the DMD modes $\Phi = \{\phi\}_{i=1}^k$ and use these modes to approximate the pressure field as $\mathbf{P} \approx \Phi \alpha$. In Figure 2.8, we plot the six DMD modes considered in this study. Then, we proceed with the Galerkin projection to obtain a reduced-order model in the form of Equation (2.6). The system has been reduced then three orders of magnitude (from $n = 10201$ to $k = 6$). Figure 2.9 depicts the steady-state pressure field obtained from the fully-resolved solution and that obtained from DMD-based reduced-order model. We observe good agreement between the two sets of data. The relative error between the reference and

approximate steady-state solutions defined as

$$\|E\| = \frac{\|p(x, t_\infty) - \Phi(x)\alpha(t_\infty)\|}{\|p(x, t_\infty)\|} \times 100\%$$

is equal to 0.035 %. These results show the capability of DMD modes to capture the main flow characteristics. We note that the DMD modes are computed from only the first few snapshots and then the reduced-order model can be used to forecast the flow field with good accuracy for long time periods without going over the numerical integration of the full system. This can be explained by the ability of DMD modes to accurately extract the information relevant to long-time dynamics; that is, the slowly-decaying eigenmodes that correspond to largest eigenvalues.

2.4 Discrete Empirical Interpolation Method

The nonlinear term $\mathbf{F}(\Phi \alpha(t))$ in the reduced-order model, given by Equation (2.6), has a computational complexity that depends on the dimension of the full system n . It requires about $2 \times n \times k$ flops for matrix multiplications and also involves a full evaluation of the nonlinear function \mathbf{F} at the n -dimensional vector $\Phi \alpha(t)$. As such, solving the reduced system might still require extensive computational resources and time. To remedy this computational complexity, one might consider the use of the discrete empirical interpolation method (DEIM).

DEIM is based on the approximation of the nonlinear terms by means of an interpolatory projection of few $m \ll n$ selected snapshots of the nonlinear terms. In this case, significant computational savings can be obtained in a forward run of nonlinear models.

2.4.1 Mathematical background

In this section, we review the DEIM as presented in [24]. Let $\mathbf{f}(\tau) \in \mathbb{R}^n$ denotes a nonlinear function where τ refers to time t or any control parameter μ . We assume an

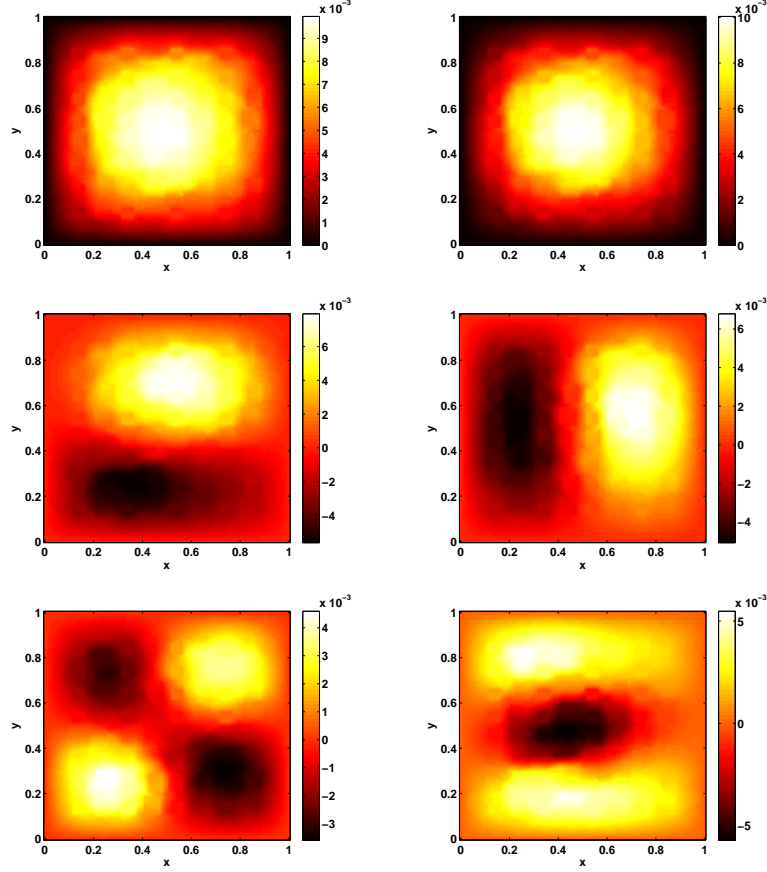


Figure 2.8: DMD modes of the pressure field.

approximation of the nonlinear function \mathbf{f} obtained by projecting it into a subspace spanned by the basis functions $\mathbf{U}_f = (\mathbf{u}_{f_1}, \dots, \mathbf{u}_{f_m}) \in \mathbb{R}^{n \times m}$ as

$$\mathbf{f}(\tau) \approx \mathbf{U}_f \mathbf{c}(\tau) \quad (2.37)$$

To compute the coefficient vector \mathbf{c} , we consider a matrix

$$\mathbf{P} = [\mathbf{e}_{\varphi_1}, \dots, \mathbf{e}_{\varphi_m}] \in \mathbb{R}^{n \times m}$$

where $\mathbf{e}_{\varphi_i} = [0, \dots, 0, 1, 0, \dots, 0]^T \in \mathbb{R}^n$ is the φ_i^{th} column of the identity matrix $\mathbf{I}_n \in \mathbb{R}^{n \times n}$ for $i = 1, \dots, m$. Multiplying Equation (2.37) by \mathbf{P}^T and assuming that

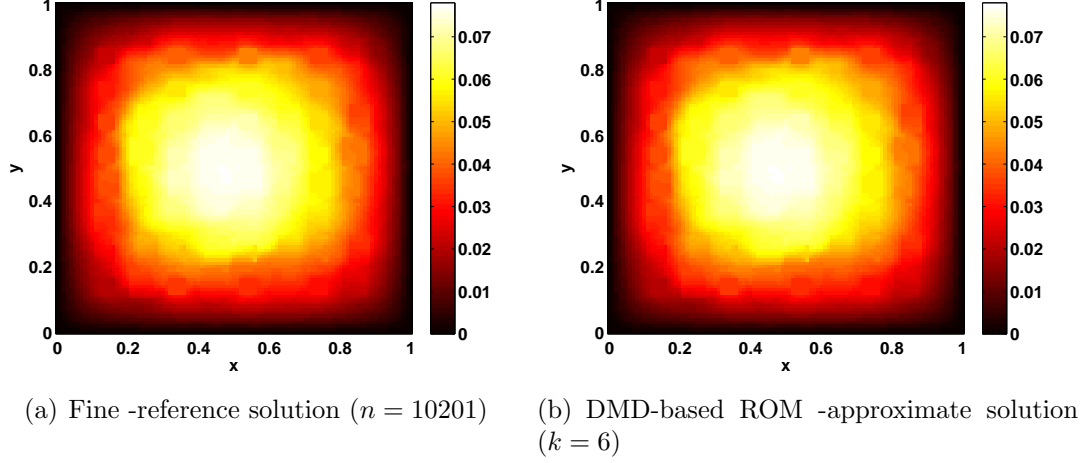


Figure 2.9: Steady-state pressure fields. Comparison between reference solution of the fine scale problem with those obtained from the DMD-based approach.

the matrix $\mathbf{P}^T \mathbf{U}_f$ is nonsingular, we obtain

$$\mathbf{f}(\tau) \approx \mathbf{U}_f \mathbf{c}(\tau) = \mathbf{U}_f (\mathbf{P}^T \mathbf{U}_f)^{-1} \mathbf{P}^T \mathbf{f}(\tau) \quad (2.38)$$

To summarize, the approximation of the nonlinear function $\mathbf{f}(\tau)$, as given by Equation (2.38), requires the followings:

- the computation of the projection basis $\mathbf{U}_f = (\mathbf{u}_{f_1}, \dots, \mathbf{u}_{f_m})$
- the identification of the indices $\{\varphi_1, \dots, \varphi_m\}$

To determine the projection basis $\mathbf{U}_f = (\mathbf{u}_{f_1}, \dots, \mathbf{u}_{f_m})$, we assemble the function evaluations in an $n \times n_s$ matrix $(\mathbf{f}(\tau_1), \dots, \mathbf{f}(\tau_{n_s}))$ and employ the POD technique (i.e., apply **Algorithm 1**) to compute the POD modes. These modes are used as the projection basis in the approximation given by Equation (2.37). As for the interpolation indices $\{\varphi_1, \dots, \varphi_m\}$, they are selected using the DEIM algorithm [24] as given below.

For the application of DEIM to nonlinear reduced-order models given by Equation

Algorithm 4 *Discrete empirical interpolation method [24]*

Input: a projection basis $\mathbf{U}_f = (\mathbf{u}_{f_1}, \dots, \mathbf{u}_{f_m}) \in \mathbb{R}^{n \times m}$
obtained by applying POD on a sequence of n_s function evaluations

Output: the interpolation indices $\vec{\wp} = (\wp_1, \dots, \wp_m)^T$

1. Set $[\rho], \wp_1] = \max\{[\mathbf{u}_{f_1}]\}$
2. Set $\mathbf{U}_f = (\mathbf{u}_{f_1})$, $\mathbf{P} = (\mathbf{e}_{\rho_1})$, and $\vec{\wp} = (\wp_1, \dots, \wp_m)$
3. **for** $\ell = 2$ **to** m **do**
 - Solve $(\mathbf{P}^T \mathbf{U}_f) \mathbf{c} = \mathbf{P}^T \mathbf{u}_{f_\ell}$
 - Compute $\mathbf{r} = \mathbf{u}_{f_\ell} - \mathbf{U}_f \mathbf{c}$
 - Compute $[\rho], \wp_\ell] = \max\{[\mathbf{r}]\}$
 - Set $\mathbf{U}_f = (\mathbf{U}_f \mathbf{u}_{f_\ell})$, $\mathbf{P} = (\mathbf{P} \mathbf{e}_{\rho_\ell})$, and $\vec{\wp} = \begin{pmatrix} \vec{\wp} \\ \wp_\ell \end{pmatrix}$

end for

$\mathbf{e}_{\wp_i} = [0, \dots, 0, 1, 0, \dots, 0]^T$ is the \wp_i^{th} column of the identity matrix \mathbf{I}_n
 $[\rho], \wp_\ell] = \max\{[\mathbf{r}]\}$ implies $[\rho] = [\mathbf{r}_{\wp_\ell}] = \max_{i=1, \dots, n} \{[\mathbf{r}_i]\}$

(2.6), the approximation of the nonlinear function is written as

$$\begin{aligned} \mathbf{F}(\Phi \alpha(t)) &\approx \mathbf{U}_f (\mathbf{P}^T \mathbf{U}_f)^{-1} \mathbf{P}^T \mathbf{F}(\Phi \alpha(t)) \\ &= \mathbf{U}_f (\mathbf{P}^T \mathbf{U}_f)^{-1} \mathbf{F}(\mathbf{P}^T \Phi \alpha(t)) \end{aligned} \quad (2.39)$$

and then

$$\tilde{\mathbf{F}}(\Phi \alpha(t)) \approx \Phi^T \mathbf{U}_f (\mathbf{P}^T \mathbf{U}_f)^{-1} \mathbf{F}(\mathbf{P}^T \Phi \alpha(t)) \quad (2.40)$$

In Equation (2.40), the term $\Phi^T \mathbf{U}_f (\mathbf{P}^T \mathbf{U}_f)^{-1} \in \mathbb{R}^{k \times m}$ is computed once and stored and the term $(\mathbf{P}^T \Phi) \alpha(t) \in \mathbb{R}^{m \times 1}$ is obtained by extracting the rows \wp_1, \dots, \wp_m (identified from **Algorithm 4**) of the basis functions or modes collected in Φ and then multiplying the result by α . Here comes the computational saving since such operation requires only the evaluation of the nonlinear function at m selected points rather than performing a full evaluation over the whole system domain.

2.4.2 Numerical examples of the DEIM applications

To demonstrate the applicability of DEIM in approximating nonlinear functions, we consider two parametrized functions as studied in [24] and a nonlinear PDE governing the microwave heating of a square slab.

A one-dimensional nonlinear function

We consider a nonlinear parametrized function $F : [-1, 1] \times [1, \pi] \mapsto \mathbb{R}$ given by

$$F(x; \mu) = (1 - x) \cos(3\pi\mu(x + 1))e^{-(1+x)\mu}$$

where $x \in [-1, 1]$ and $\mu \in [1, \pi]$. We let the vector $\mathbf{x} = (x_1, \dots, x_n)^T \in \mathbb{R}^n$ formed of n equally-spaced points in $[-1, 1]$ and set $n = 100$. For the sake of the subsequent analysis, we introduce the following vector $\mathbf{f}(\mu)$ that collects the function evaluations at the points x_i ; that is,

$$\mathbf{f}(\mu) = (F(x_1; \mu), \dots, F(x_n; \mu))^T \in \mathbb{R}^n$$

To proceed with DEIM, we first construct the snapshot matrix obtained by evaluating the function \mathbf{f} at equally-spaced values $\{\mu^\ell\}_{\ell=1}^{n_s}$ of μ in $[1, \pi]$; that is,

$$\begin{pmatrix} | & | & & | \\ \mathbf{f}(\mu^1) & \mathbf{f}(\mu^2) & \dots & \mathbf{f}(\mu^{n_s}) \\ | & | & & | \end{pmatrix} \in \mathbb{R}^{n \times n_s},$$

and then, we compute the first ten POD modes collected in the matrix \mathbf{U}_f . Figure 2.10 shows the singular values and the corresponding first three modes. The number of snapshots of the nonlinear function is $n_s = 50$. As mentioned in the previous section, the selection of POD modes is based on an energy ranking of the coherent structures. As such, we can clearly observe from Figure 2.10(a) that the first few modes contain most of the energy associated with the snapshot matrix. Next, we select the interpolation indices $\vec{\wp} = (\wp_1, \dots, \wp_m)^T$ using **Algorithm 4** which enable the determination of the DEIM points only at which the nonlinear function is

evaluated. The approximate nonlinear function is given then by

$$\tilde{\mathbf{f}}(\mu) = \mathbf{U}_f(\mathbf{P}^T \mathbf{U}_f)^{-1} [F(x_{\varphi_1}; \mu), \dots, F(x_{\varphi_m}; \mu)]^T \in \mathbb{R}^n \quad (2.41)$$

where $\mathbf{P} = [\mathbf{e}_{\varphi_1}, \dots, \mathbf{e}_{\varphi_m}]$ and $\mathbf{e}_{\varphi_i} = [0, \dots, 0, 1, 0, \dots, 0]^T \in \mathbb{R}^n$ is the φ_i^{th} column of the identity matrix $\mathbf{I}_n \in \mathbb{R}^{n \times n}$ for $i = 1, \dots, m$.

In Figure 2.11, we compare the approximate function obtained from DEIM with $m = 10$ dimensions against the original system of dimension $n = 100$ at different values of μ . The good agreement observed between the two sets of data shows the capability of DEIM to approximate the nonlinear function using few selected points.

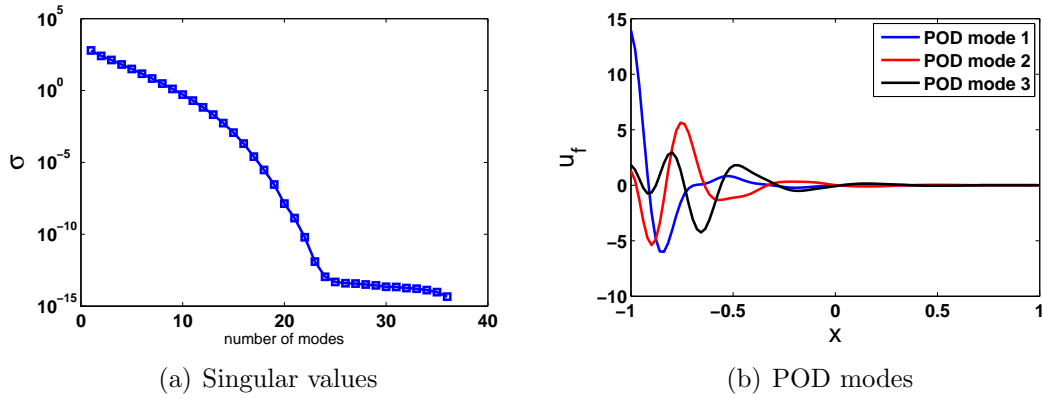


Figure 2.10: Singular values and the corresponding first three POD modes.

We define the relative error between the nonlinear function and its approximations as $\|E\| = \|F - \tilde{F}\|/\|\tilde{F}\|$ where \tilde{F} is obtained from DEIM approximation and POD L_2 projection; that is,

$$\tilde{F} = \Phi(\Phi^T \Phi)^{-1} \Phi^T F$$

where Φ is the matrix collecting the POD modes. In Figure 2.12, we show the variations of the relative error with the number of POD modes or DEIM dimension. As expected, the error decreases as the number of POD modes/DEIM grid points is increased. Furthermore, POD-based approach yields slightly smaller errors since POD modes form an optimal subspace. However, the latter approach requires function

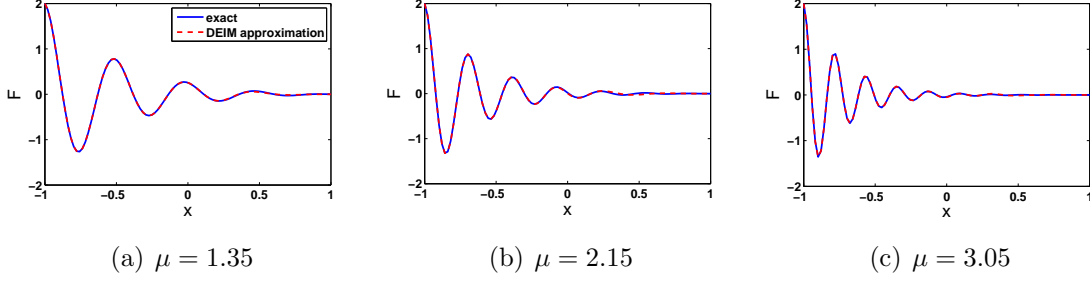


Figure 2.11: Comparison of the DEIM approximation of dimension $m = 10$ with the original nonlinear function of dimension $n = 100$ for different values of μ .

evaluations at all grid points ($n = 100$) while DEIM-based approach requires only the evaluation of the nonlinear function at the interpolation points ($m = 10$).

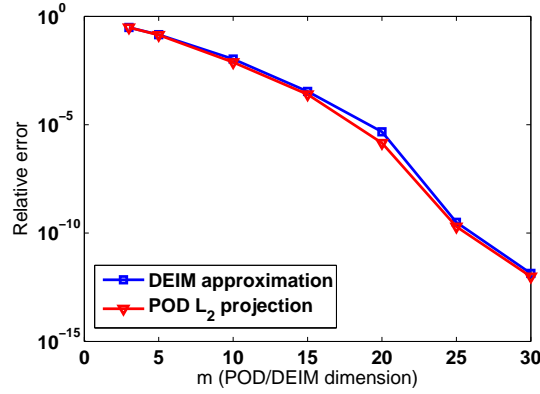


Figure 2.12: Relative error between the approximation and the original nonlinear function: POD L_2 projection vs. DEIM.

A two-dimensional nonlinear function

We consider a two-dimensional parametrized function $F : [0.1, 0.9]^2 \times [1, \pi]^2 \mapsto \mathbb{R}$ given by

$$F(x; y; \mu_1; \mu_2) = \frac{1}{\sqrt{(x - \mu_1)^2 + (y - \mu_2)^2 + 0.1^2}}$$

where $(x, y) \in [0.1, 0.9]^2$ and $\mu = (\mu_1, \mu_2) \in [1, \pi]^2$. We let the vectors $\mathbf{x} = (x_1, \dots, x_{n_x})^T \in \mathbb{R}^{n_x}$ and $\mathbf{y} = (y_1, \dots, y_{n_y})^T \in \mathbb{R}^{n_y}$ formed of equally-spaced points in $[0.1, 0.9]$ with $n_x = n_y = 20$. We introduce the following vector $\mathbf{f}(\mu)$ that collects the function evaluations at the grid points (x_i, y_j) for $i = 1, \dots, n_x$ and $j = 1, \dots, n_y$; that is,

$$\mathbf{f}(\mu) = (F(x_1; y_1; \mu), F(x_2; y_1; \mu), \dots, F(x_{n_x}; y_1; \mu), \dots, F(x_{n_x}; y_{n_y}; \mu))^T \in \mathbb{R}^n$$

where $n = n_x \times n_y = 400$. Next, we construct the snapshot matrix obtained by evaluating the function \mathbf{f} at uniformly selected control parameters $\{\mu^\ell\}_{\ell=1}^{n_s}$ of μ in $[1, \pi]^2$; that is,

$$\begin{pmatrix} | & | & & | \\ \mathbf{f}(\mu^1) & \mathbf{f}(\mu^2) & \dots & \mathbf{f}(\mu^{n_s}) \\ | & | & & | \end{pmatrix} \in \mathbb{R}^{n \times n_s},$$

and then, we compute its POD modes. Figures 2.13(a) and 2.14 show the singular values and the corresponding first six POD modes, respectively.

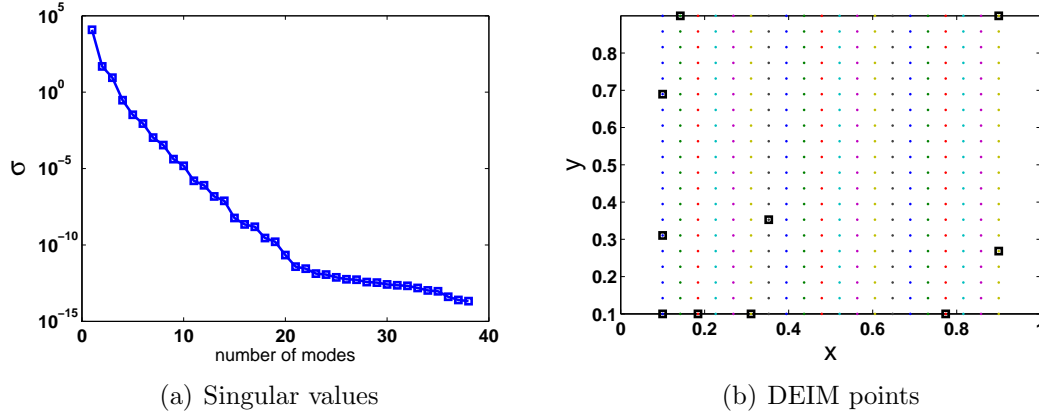


Figure 2.13: (a) Singular values and (b) first ten selected DEIM points for the nonlinear function represented by black rectangles.

We determine the interpolation indices using **Algorithm 4** which enable the determination of the DEIM points only at which the nonlinear function is evaluated.

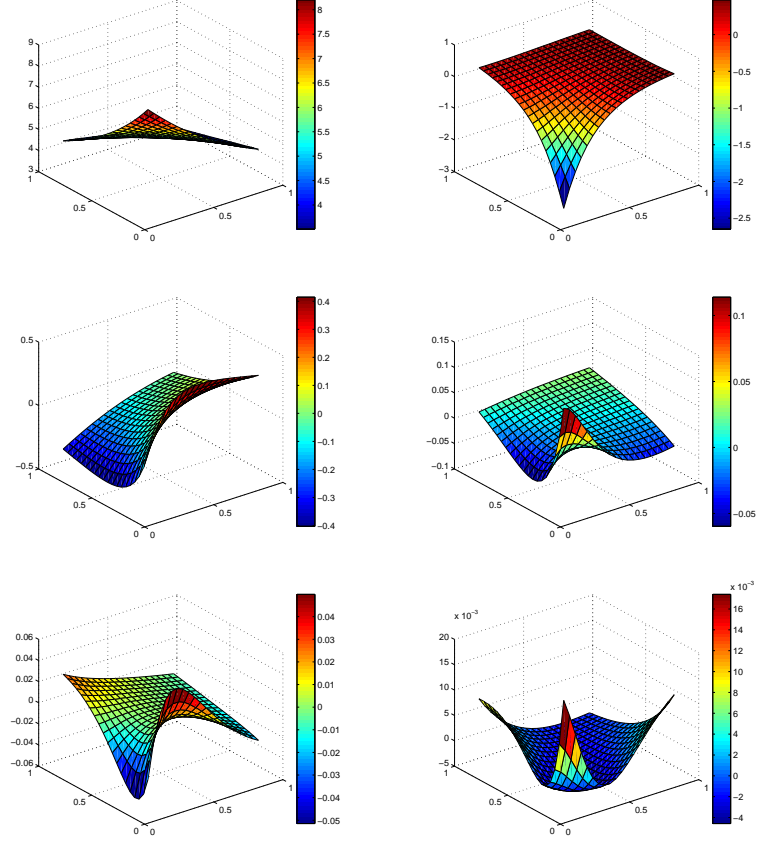


Figure 2.14: POD modes of the snapshots of the nonlinear function.

The first ten selected DEIM points are shown in Figure 2.13(b). The approximate function is computed as given in Equation (2.41). In Figure 2.15, we compare the approximate functions obtained from DEIM and POD L_2 projection with $m = 10$ dimensions against the original function of dimension $n = 400$ for $\mu = (\mu_1, \mu_2) = (-0.05, -0.05)$. As expected, projecting the nonlinear function onto the set of POD modes (optimal subspace) yields a good approximation of the nonlinear function. The good agreement observed between the original function and its approximation based on DEIM shows again the capability of this technique to reproduce the original nonlinear function using few selected points.

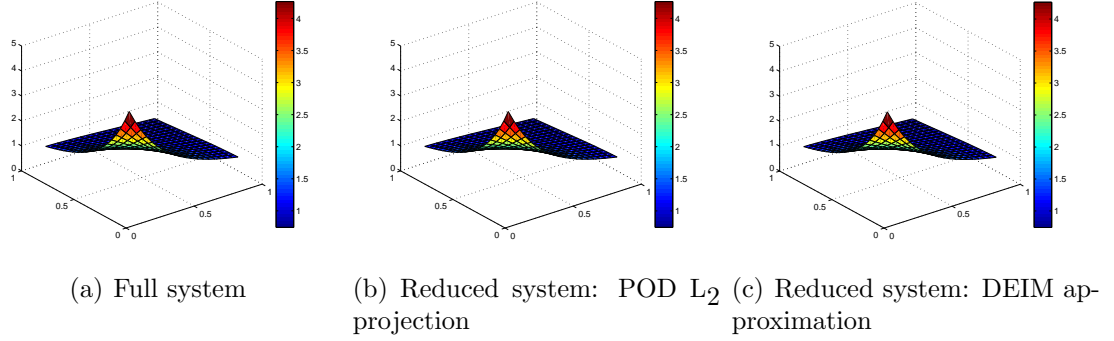


Figure 2.15: Comparison of the DEIM and POD approximations of dimension $m = 10$ with the original nonlinear function of dimension $n = 400$ for $\mu = (\mu_1, \mu_2) = (-0.05, -0.05)$.

The relative error, as defined above, is plotted against the number of POD modes or DEIM dimension in Figure 2.16. Again, we observe that the error decreases as the number of POD modes/DEIM grid points is increased. We note that for the POD-based approach, the nonlinear function has been evaluated at all grid points ($n = 400$) while DEIM-based approach requires only the evaluation of the nonlinear function at the interpolation points ($m = 10$).

A nonlinear unsteady problem: microwave heating of a square slab

As an example of a nonlinear unsteady problem, we consider the heating of a square slab using microwave energy. This heating is governed by the following forced heat equation

$$\begin{aligned}
 \nabla^2 u &= \frac{\partial u}{\partial t} - \beta u^\nu \quad \text{in } \Omega \\
 u &= 1 \quad \text{on } \partial\Omega \\
 u &= 1 \quad \text{at } t = 0
 \end{aligned} \tag{2.42}$$

where u denotes the temperature, $\Omega = [0, 1] \times [0, 1]$, the parameters introduced to model the microwave radiation through the material are given by $\beta = 2.8$ and $\nu = 3$.

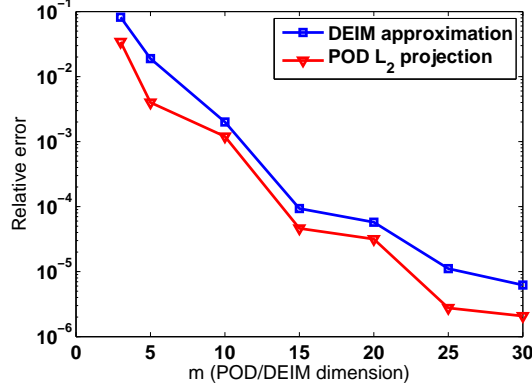


Figure 2.16: Relative error between the approximation and the original nonlinear function: POD L_2 projection vs. DEIM.

Using the above configuration, the problem governed by Equation (2.42) is solved on a fine mesh grid over a time interval of $T = [0 \ 1.9]$ with a time step $\Delta t = 5 \times 10^{-2}$. This time interval is observed to be long enough so that the steady-state solution is achieved. The dimension of the full-order system is $n = 6561$. Next, we proceed with model reduction based on projection-based method, as detailed in Section 2.1, where we combine POD and DEIM techniques. The result is a reduced-order model in the form of Equation (2.6) where the nonlinear term is approximated via interpolation of a set of few selected points as given by Equation (2.40). The interpolation points are shown in Figure 2.17. We note that the POD modes are computed from the first 30 snapshots obtained by solving the full-order system at equally-spaced time steps in the interval of $[0 \ 0.3]$ in which the temperature does not reach steady state yet. In Figure 2.18, we plot the steady-state temperature obtained from the full-order system and the reduced-order model based on 10 POD modes and 30 DEIM interpolation points used for the approximation of the nonlinear term. Clearly, the reduced system predicts accurately the temperature while reducing significantly the dimension of the problem.

To gain an insight on the trade-offs between the gain in the computational cost

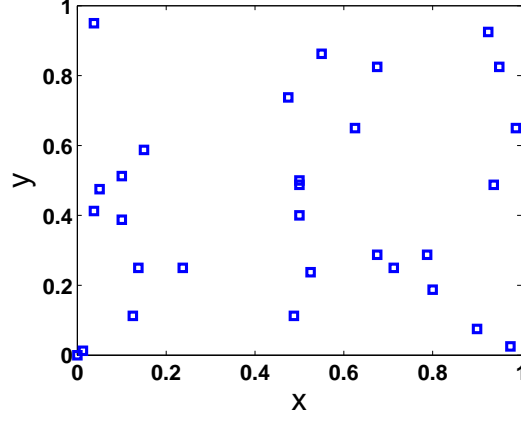


Figure 2.17: The first 30 selected interpolation points by the DEIM algorithm.

and solution accuracy, we show in Figure 2.19, the variations of the relative error and the execution time in seconds for different numbers of POD modes and DEIM interpolation points. The numerical simulations are performed on a laptop featuring Intel® Xeon® CPU X5650 2.67 GHZ processor. We note the total time taken to run the full-order system is 1 mn 50 s. Clearly, the coupled POD-DEIM model reduction approach yields a significant reduction in the computational cost while reproducing the solutions of the full-order system with a good accuracy. Furthermore, increasing the number of modes when constructing the reduced-order model yields smaller errors, but this is at the price of an increase in the computational cost.

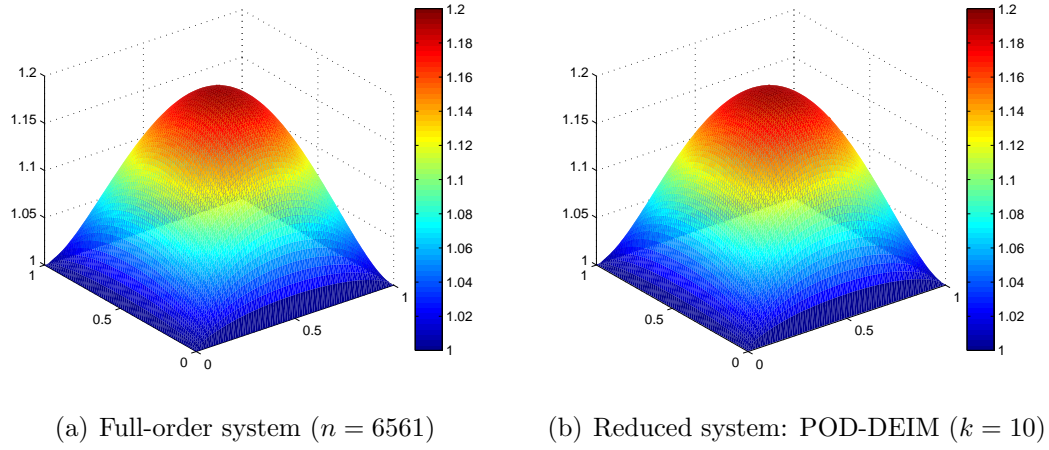


Figure 2.18: Comparison between the fully-resolved steady-state temperature with that obtained from the combined POD-DEIM model reduction approach. 10 POD modes and 30 DEIM interpolation points are considered.

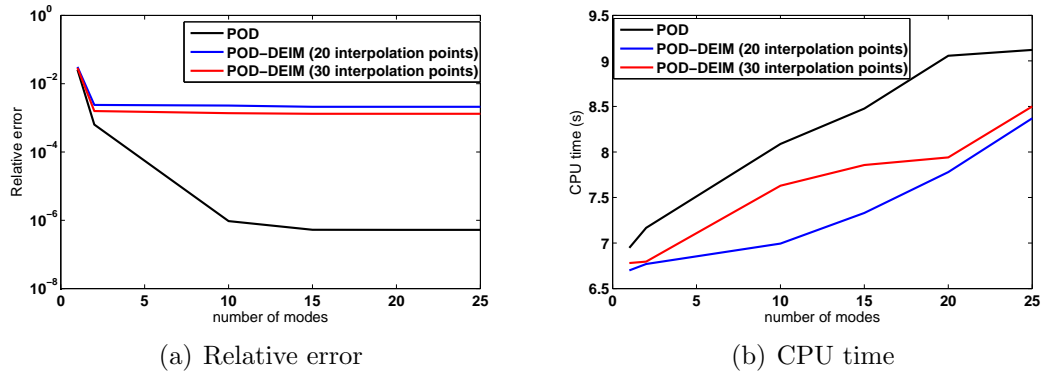


Figure 2.19: Variations of the (a) relative error and (b) computational time for the POD- and coupled POD-DEIM based reduced-order models. The total time taken to run the full-order system is 1 mn 50 s.

Bibliography

- [1] I. Akhtar, A. H. Nayfeh, C. J. Ribbens, On the stability and extension of reduced-order Galerkin models in incompressible flows: A numerical study of vortex shedding, *Theoretical and Computational Fluid Dynamics* 23 (3) (2009) 213–237.
- [2] Z. Wang, I. Akhtar, J. Borggaard, T. Iliescu, Two-level discretizations of nonlinear closure models for proper orthogonal decomposition, *Journal of Computational Physics* 230 (1) (2011) 126–146.
- [3] Z. Wang, I. Akhtar, J. Borggaard, T. Iliescu, Proper Orthogonal Decomposition Closure Models for Turbulent Flows: A Numerical Comparison, *Computer Methods in Applied Mechanics and Engineering* 237–240 (2012) 10–26.
- [4] I. Akhtar, Z. Wang, J. Borggaard, T. Iliescu, A new closure strategy for proper orthogonal decomposition reduced-order models, *Journal of Computational and Nonlinear Dynamics* 7 (3) (2012) 034503.
- [5] M. Ghommem, I. Akhtar, M. R. Hajj, A low-dimensional tool for predicting force decomposition coefficients and varying inflow conditions, *Progress in Computational Fluid Dynamics*, in press.
- [6] H. P. Bakewell, J. L. Lumley, Viscous sublayer and adjacent wall region in turbulent pipe flow, *The Physics of Fluids* 10 (9) (1967) 1880–1889.

- [7] L. Sirovich, Turbulence and the dynamics of coherent structures, *Quarterly of Applied Mathematics* 45 (1987) 561–590.
- [8] A. E. Deane, I. G. Kevrekidis, G. E. Karniadakis, S. A. Orsag, Low-dimensional models for complex geometry flows: Application to grooved channels and circular cylinder, *Physics of Fluids A* 3 (10) (1991) 2337–2354.
- [9] G. Berkooz, P. Holmes, J. L. Lumley, The proper orthogonal decomposition in the analysis of turbulent flows, *Annual Review of Fluid Mechanics* 53 (1993) 321–575.
- [10] P. Holmes, J. L. Lumley, G. Berkooz, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, Cambridge University Press, Cambridge, UK, 1996.
- [11] A. Hay, J. Borggaard, I. Akhtar, D. Pelletier, Reduced-order models for parameter dependent geometries based on shape sensitivity analysis, *Journal of Computational Physics* 229 (2010) 1327–1352.
- [12] A. Hay, I. Akhtar, J. T. Borggaard, On the use of sensitivity analysis in model reduction to predict flows for varying inflow conditions, *International Journal for Numerical Methods in Fluids* 68 (1) (2012) 122–134.
- [13] P. J. Schmid, Dynamic mode decomposition of numerical and experimental data, *Journal of Fluid Mechanics* 656 (2010) 5–28.
- [14] D. Duke, J. Soria, D. Honnery, An error analysis of the dynamic mode decomposition, *Experiments in Fluids* 52 (2012) 529–542.
- [15] P. J. Schmid, Application of the dynamic mode decomposition to experimental data, *Experiments in Fluids* 50 (2011) 1123–1130.
- [16] C. Pan, D. Yu, J. Wang, Dynamical mode decomposition of gurney flap wake flow, *Theoretical and Applied Mechanics Letters* 1 (2011) 012002.

- [17] P. J. Schmid, Dynamic mode decomposition of experimental data, in: 8th International Symposium on Particle Image Velocimetry - PIV09, 2009.
- [18] P. J. Schmid, K. E. Meyer, O. Pust, Dynamic mode decomposition and proper orthogonal decomposition of flow in a lid-driven cylindrical cavity, in: 8th International Symposium on Particle Image Velocimetry - PIV09, 2009.
- [19] F. Lusseyran, F. Gueniat, J. Basley, C. L. Douay, L. R. Pastur, T. M. Faure, P. Schmid, Flow coherent structures and frequency signature: application of the dynamic modes decomposition to open cavity flow, *Journal of Physics: Conference Series* 318 (2011) 042036.
- [20] T. W. Muld, G. Efraimsson, D. S. Henningson, Flow structures around high-speed train extracted using proper orthogonal decomposition and dynamic mode decomposition, *Computers & Structures* 57 (2012) 87–97.
- [21] P. J. Schmid, L. Li, M. P. Juniper, O. Pust, Applications of the dynamic mode decomposition, *Theoretical and Computational Fluid Dynamics* 25 (2011) 249–259.
- [22] A. Seena, H. J. Sung, Dynamic mode decomposition of turbulent cavity flows for self-sustained oscillations, *International Journal of Heat and Fluid Flow* 32 (2011) 1098–1110.
- [23] Y. Mizuno, D. Duke, C. Atkinson, J. Soria, Investigation of wall-bounded turbulent flow using dynamic mode decomposition, *Journal of Physics: Conference Series* 318 (2011) 042040.
- [24] S. Chaturantabut, D. C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, *SIAM Journal on Scientific Computing* 32 (2010) 2737–2764.
- [25] L. Sirovich, M. Kirby, Low-dimensional procedure for the characterization of human faces, *Journal of Optical Society of America A* 4 (3) (1987) 529–524.

- [26] K. K. Chen, J. H. Tu, C. W. Rowley, Variants of dynamic mode decomposition: Boundary conditions, koopman, and fourier analysis, *Journal of Nonlinear Science* DOI: 10.1007/s00332-012-9130-9.
- [27] B. R. Noack, P. Papas, P. A. Monkewitz, The need for a pressure-term representation in empirical galerkin models of incompressible shear flows, *Journal of Fluid Mechanics* 523 (2005) 339–365.
- [28] X. Ma, G. Karniadakis, A low-dimensional model for simulating three-dimensional cylinder flow, *Journal of Fluid Mechanics* 458 (2002) 181–190.
- [29] I. Akhtar, Parallel simulations, reduced-order modeling, and feedback control of vortex shedding using fluidic actuators, Ph.D. thesis, Virginia Tech, Blacksburg, VA (2008).
- [30] I. Akhtar, A. H. Nayfeh, Model based control of vortex shedding using fluidic actuators, *Journal of Computational and Nonlinear Dynamics* 5 (4) (2010) 041015.
- [31] I. Akhtar, J. Borggaard, A. Hay, Shape sensitivity analysis in flow models using a finite-difference approach, *Mathematical Problems in Engineering* 2010, Article ID 209780 (2010) 1–22.