

Comparison of POD reduced order strategies for the nonlinear 2D shallow water equations

Răzvan Ștefănescu^{1,*}, Adrian Sandu¹ and Ionel M. Navon²

¹*Computational Science Laboratory, Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA, 24060, USA*

²*Department of Scientific Computing, The Florida State University, Tallahassee, FL 32306, USA*

SUMMARY

This paper introduces tensorial calculus techniques in the framework of POD to reduce the computational complexity of the reduced nonlinear terms. The resulting method, named tensorial POD, can be applied to polynomial nonlinearities of any degree p . Such nonlinear terms have an online complexity of $\mathcal{O}(k^{p+1})$, where k is the dimension of POD basis and therefore is independent of full space dimension. However, it is efficient only for quadratic nonlinear terms because for higher nonlinearities, POD model proves to be less time consuming once the POD basis dimension k is increased. Numerical experiments are carried out with a two-dimensional SWE test problem to compare the performance of tensorial POD, POD, and POD/discrete empirical interpolation method (DEIM). Numerical results show that tensorial POD decreases by 76× the computational cost of the online stage of POD model for configurations using more than 300,000 model variables. The tensorial POD SWE model was only 2 to 8× slower than the POD/DEIM SWE model but the implementation effort is considerably increased. Tensorial calculus was again employed to construct a new algorithm allowing POD/DEIM SWE model to compute its offline stage faster than POD and tensorial POD approaches. Copyright © 2014 John Wiley & Sons, Ltd.

Received 9 February 2014; Revised 8 July 2014; Accepted 19 July 2014

KEY WORDS: tensorial proper orthogonal decomposition (POD); discrete empirical interpolation method (DEIM); reduced order models (ROMs); shallow water equations (SWE); finite difference methods; alternating direction implicit methods (ADI)

1. INTRODUCTION

Modeling and simulation of multi-scale complex physical phenomena lead to large-scale systems of coupled partial differential equations, ordinary differential equations, and differential algebraic equations. The high dimensionality of these models poses important mathematical and computational challenges. A computationally feasible approach to simulate, control, and optimize such systems is to simplify the models by retaining only those state variables that are consistent with a particular phenomena of interest.

Reduced order modeling refers to the development of low-dimensional models that represent important characteristics of a high-dimensional or infinite-dimensional dynamical system. The reduced order methods can be casted into three broad categories: singular values decomposition (SVD)-based methods, Krylov-based methods, and iterative methods combining aspects of both the SVD and Krylov methods (see, e.g., [1]).

For linear models, methods such as balanced truncation [2–5] and moment matching [6–8] have been proving successful in developing reduced order models (ROMs). However, balanced

*Correspondence to: Răzvan Ștefănescu, Computational Science Laboratory, Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA 24060, USA.

†E-mail: rstefane@vt.edu

truncation does not extend easily for high-order systems, and several grammians approximations were proposed leading to methods such as approximate subspace iteration [9], least squares approximation [10], Krylov subspace methods [11, 12], and balanced POD [13]. Among moment matching methods, we mention partial realization [14, 15], Padé approximation [16–19], and rational approximation [20].

Although for linear models we are able to produce input-independent highly accurate reduced models, in the case of general nonlinear systems, the transfer function approach is not yet applicable and input-specified semi-empirical methods are usually employed. Recently, some encouraging research results using generalized transfer functions and generalized moment matching have been obtained by [21] for nonlinear model order reduction but future investigations are required.

Proper orthogonal decomposition and its variants are also known as Karhunen-Loève expansions [22, 23], principal component analysis [24], and empirical orthogonal functions [25] among others. It is the most prevalent basis selection method for nonlinear problems and, among other requirements, relies on the fact that the desired simulation is well simulated in the input collection. Data analysis using POD is conducted to extract basis functions from experimental data or detailed simulations of high-dimensional systems (method of snapshots introduced by [26–28]) for subsequent use in Galerkin projections that yield low-dimensional dynamical models. Unfortunately, the POD Galerkin approach displays a major disadvantage because its nonlinear reduced terms still have to be evaluated on the original state space making the simulation of the reduced order system too expensive. There exist several ways to avoid this problem such as the empirical interpolation method (EIM) [29] and its discrete variant discrete empirical interpolation method (DEIM) [30–32] and the best points interpolation method [33]. Missing point estimation [34] and Gauss–Newton with approximated tensors [35, 36] methods are relying upon the gappy POD technique [37] and were developed for the same reason. Reduced basis methods have been recently developed and utilize on greedy algorithms to efficiently compute numerical solutions for parametrized applications [29, 38–41].

Dynamic mode decomposition is a relatively recent development in the field of modal decomposition [42–44], which attempts to represent the data sequence by orthogonalizing it in time while the POD approach employs a decomposition based on orthogonality in space [43]. Trajectory piecewise linear method proposed in [45] follows a different strategy where the nonlinear system is represented by a piecewise linear system which can then be efficiently approached by the standard linear reduction method.

Parameter model reduction has emerged recently as an important research direction and [46] highlights the major contribution to the field.

This paper combines POD and tensor calculus techniques to reduce the online computational complexity of the reduced nonlinear terms for an SWEs model. Tensor-based calculus was already applied by [47] and [48] to represent quadratic nonlinearities of reduced order POD models.

We show that the tensorial POD (TPOD) approach can be applied to polynomial nonlinearities of any degree p and its representation has a complexity of $\mathcal{O}(k^{p+1})$, where k is the dimension of POD subspace. This complexity is independent of the full space dimension.

For k between 10 and 50 and $p = 2$, the number of floating-point operations required to calculate the tensorial POD quadratic terms is 10 to 40× lower than in the case of POD model and 10 to 20× higher than for the POD/DEIM. However, CPU time for solving the TPOD SWE model (online stage) is only 2 to 8 times slower than POD/DEIM SWE model for 10^3 – 10^5 grid points, $k \leq 50$, and the number of DEIM interpolation points $m \leq 180$. For example, for an integration interval of 3 h, 10^5 mesh points, $k = 50$, and $m = 70$, tensorial POD, and POD/DEIM are 76× and 450× faster than POD, but the implementation effort of POD/DEIM is considerably increased. Many useful models are characterized by quadratic nonlinearities in both fluid dynamics and geophysical fluid flows including SWE model. In the case of cubic or higher polynomial nonlinearities, the advantage of tensorial POD is lost and its nonlinear computational complexity is similar or larger than the computational complexity of the POD approach. This proves that for models depending only on quadratic nonlinearities, the tensorial POD represents a solid alternative to POD/DEIM, where the implementation effort is considerably larger. We also propose a fast algorithm to pre-compute the reduced order coefficients for polynomial nonlinearities of order p , which allows the POD/DEIM

SWE model to compute its offline stage faster than POD and tensorial POD approaches despite additional SVD calculations and reduced coefficients computations.

The paper is organized as follows. Section 2 reviews the reduced order modeling methodologies used in this work: POD, tensorial POD, and POD/DEIM. Section 3 analyzes the computational complexity of the reduced order polynomial nonlinearities for all three methods and introduces a new DEIM-based algorithm to efficiently compute the coefficients needed for reduced Jacobians. Section 4 discusses the SWEs model and its full implementation, and Section 5 describes the construction of reduced models. Results of extensive illustrative numerical experiments are discussed in Section 6, whereas conclusions are drawn in Section 7.

2. REDUCED ORDER MODELING

For highly efficient flows simulations, reduced order modeling is a powerful tool for representing the dynamics of large-scale dynamical systems using only a smaller number of variables and reduced order basis functions. Three approaches will be considered in this study: POD, tensorial POD (TPOD), and POD/DEIM (POD/DEIM). They are discussed in the following section.

The tensorial POD approach proposed herein is different from the method of [49], which makes use of tensor decompositions for generating POD bases.

2.1. Proper orthogonal decomposition

Proper orthogonal decompositions has been used successfully in numerous applications such as compressible flow [50], computational fluid dynamics [13, 51, 52], aerodynamics [53]. It can be thought of as a Galerkin approximation in the spatial variable built from functions corresponding to the solution of the physical system at specified time instances. Noack *et al.* [54] proposed a system reduction strategy for Galerkin models of fluid flows leading to dynamic models of lower order based on a partition in slow, dominant, and fast modes. San and Iliescu [55] investigate several closure models for POD reduced order modeling of fluid flows and benchmarked against the fine resolution numerical simulation.

In what follows, we will only work with discrete inner products (Euclidian dot product) though continuous products may be employed too. Generally, an atmospheric or oceanic model is usually governed by the following semi-discrete dynamical system

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{F}(\mathbf{x}, t), \quad \mathbf{x}(0) = \mathbf{x}_0 \in \mathbb{R}^n. \quad (1)$$

From the temporal–spatial flow $\mathbf{x}(t) \in \mathbb{R}^n$, we select an ensemble of N_t time instances $\mathbf{x}_1, \dots, \mathbf{x}_{N_t} \in \mathbb{R}^n$, n being the total number of discrete model variables per time step and $N_t \in \mathbb{N}$, $N_t > 0$. Let us define the centering trajectory, shift mode, or mean field correction [56] $\bar{\mathbf{x}} = \frac{1}{N_t} \sum_{i=1}^{N_t} \mathbf{x}_i$. The method of POD consists in choosing a complete orthonormal basis $U = \{\mathbf{u}_i\}$, $i = 1, \dots, k$; $k > 0$; $\mathbf{u}_i \in \mathbb{R}^n$; $U \in \mathbb{R}^{n \times k}$ such that the mean square error between $\mathbf{x}(t)$ and POD expansion $\mathbf{x}^{POD}(t) = \bar{\mathbf{x}} + U\tilde{\mathbf{x}}(t)$, $\tilde{\mathbf{x}}(t) \in \mathbb{R}^k$ is minimized on average. The POD dimension $k \ll n$ is appropriately chosen to capture the dynamics of the flow as described by Algorithm 1.

Another way to compute the POD basis is to make use of SVD, which is less affected by numerical errors than the eigenvalue decomposition. Moreover, the SVD-based POD basis construction is more computationally efficient because it decomposes the snapshots matrix whose condition number is the square root of the correlation matrix K used in Algorithm 1.

To obtain the reduced model of (1), we first employ a numerical scheme to solve the full model for a set of snapshots and follow the aforementioned procedure, then use a Galerkin projection of the full model equations onto the space \mathcal{X}^k spanned by the POD basis elements

$$\frac{d\tilde{\mathbf{x}}(t)}{dt} = U^T \mathbf{F}(\bar{\mathbf{x}} + U\tilde{\mathbf{x}}(t), t), \quad \tilde{\mathbf{x}}(0) = U^T (\mathbf{x}(0) - \bar{\mathbf{x}}). \quad (2)$$

Algorithm 1 POD basis construction

- 1: Calculate the mean $\bar{\mathbf{x}} = \frac{1}{N_t} \sum_{i=1}^{N_t} \mathbf{x}_i$.
 - 2: Set up the correlation matrix $K = [k_{ij}]_{i,j=1,\dots,n}$, where $k_{ij} = \langle \mathbf{x}_i - \bar{\mathbf{x}}, \mathbf{x}_j - \bar{\mathbf{x}} \rangle$, and $\langle \cdot, \cdot \rangle$ being the Euclidian dot product.
 - 3: Compute the eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots \lambda_n \geq 0$ and the corresponding orthogonal eigenvectors $\mathbf{v}^1, \mathbf{v}^2, \dots, \mathbf{v}^n \in \mathbb{R}^n$ of K .
 - 4: Set $\mathbf{u}_i = \langle \mathbf{v}^i, \mathbf{x}_i - \bar{\mathbf{x}} \rangle$, $i = 1, \dots, n$. Then, $\mathbf{u}_i \in \mathbb{R}^n$, $i = 1, \dots, n$ are normalized to obtain an orthonormal basis.
 - 5: Define $I(m) = \frac{\sum_{i=1}^m \lambda_i}{\sum_{i=1}^n \lambda_i}$ and choose k such that $k = \min\{I(m) : I(m) \geq \gamma\}$, where $0 \leq \gamma \leq 1$ is the percentage of total information captured by the reduced space $\text{span}\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k\}$. Usually, γ is taken 0.99.
-

The efficiency of the POD-Galerkin techniques is limited to linear or bilinear terms, because the projected nonlinear term at every discrete time step still depends on the number of variables of the full model

$$\tilde{N}(\tilde{\mathbf{x}}) = \underbrace{U^T}_{k \times n} \underbrace{\mathbf{F}(\tilde{\mathbf{x}} + U\tilde{\mathbf{x}}(t))}_{n \times 1}.$$

To be precise, consider a steady polynomial nonlinearity x^p . A POD expansion involving mean $\bar{\mathbf{x}}$ will unnecessarily complicate the description of tensorial POD representation of a p^{th} -order polynomial nonlinearity. Moreover, the terms depending on $\bar{\mathbf{x}}$ are just a particular case of the term depending only on $U\tilde{\mathbf{x}}$ because vector componentwise multiplication is distributive over vector addition. Consequently, the expansion $\tilde{\mathbf{x}} \approx U\tilde{\mathbf{x}}$ will not decrease the generality of the reduced nonlinear term. In the finite difference (FD) case, the POD Galerkin projection is described as follows.

$$\tilde{N}(\tilde{\mathbf{x}}) = \underbrace{U^T}_{k \times n} \underbrace{(U\tilde{\mathbf{x}})^p}_{n \times 1}, \tag{3}$$

where vector powers are taken componentwise.

To mitigate this inefficiency, we will employ two approaches: (1) tensorial POD and (2) DEIM. Tensorial POD for nonlinearities treatment was traditionally used in POD Galerkin by the fluid mechanics community [28, 47, 48], and a matrix formulation named pre-computing technique was introduced in [30] for calculation of quadratic nonlinearities. Reduced order treatment of cubic nonlinearities using tensorial POD has been considered in the structural mechanics community [57]. Here, we present the more general case where the POD Galerkin p^{th} -order polynomial nonlinearities are computed via tensorial-based calculus. Although tensorial POD is able to calculate the reduced polynomial nonlinearities independent of n , POD/DEIM method can handle efficiently all types of nonlinearities.

2.2. Tensorial POD

Tensorial POD technique employs the simple structure of the polynomial nonlinearities to remove the dependence on the dimension of the original discretized system by manipulating the order of computing. It can be successfully used in a POD framework for FD, finite element (FE), and finite volume (FV) discretization methods and all other types of discretization methods that engage in spectral expansions. Tensorial POD separates the full spatial variables from reduced variables allowing fast nonlinear terms computations in the online stage. For time-dependent nonlinearities, this implies separation of spatial variables from reduced time variables. Thus, the reduced nonlinear term evaluation requires a tensorial Frobenius dot product computation between rank p tensors, where p is the order of polynomial nonlinearity. The projected spatial variables are stored into tensors and calculated offline. These are also used for reduced Jacobian computation in the online stage.

The tensorial POD representation of (3) is given by vector

$$\mathcal{M} = [\mathcal{M}^i]_{i=1,2,\dots,k}; \quad \mathcal{M}^i = \left\langle \mathbf{M}^i, \tilde{\mathbf{X}} \right\rangle_{\text{Frobenius}} \in \mathbb{R}, \quad i = 1, 2, \dots, k; \quad \mathcal{M} \in \mathbb{R}^k, \quad (4)$$

where p -order tensors $\tilde{\mathbf{X}}$ and \mathbf{M}^i , $i = 1, 2, \dots, k$ are defined as

$$\begin{aligned} \tilde{\mathbf{X}} &= \left[\tilde{\mathbf{X}}_{i_1 i_2 \dots i_p} \right]_{i_1, i_2, \dots, i_p=1, \dots, k} \in \mathbb{R}^{\overbrace{k \times \dots \times k}^{p \text{ times}}}; \quad \tilde{\mathbf{X}}_{i_1 i_2 \dots i_p} = \tilde{\mathbf{x}}_{i_1} \tilde{\mathbf{x}}_{i_2} \dots \tilde{\mathbf{x}}_{i_p} \in \mathbb{R}; \\ \mathbf{M}^i &= \left[\mathbf{M}^i_{i_1 i_2 \dots i_p} \right]_{i_1, i_2, \dots, i_p=1, 2, \dots, k}, \quad i = 1, 2, \dots, k; \quad \mathbf{M}^i \in \mathbb{R}^{\overbrace{k \times \dots \times k}^{p \text{ times}}}; \\ \mathbf{M}^i_{i_1 i_2 \dots i_p} &= \sum_{l=1}^n U_{li} U_{li_1} U_{li_2} \dots U_{li_p} \in \mathbb{R}, \end{aligned} \quad (5)$$

and $\tilde{\mathbf{x}}_{i_j}$, U_{li} , U_{li_j} are just entries of POD reduced order solution $\tilde{\mathbf{x}}$ and POD trial functions basis U . The tensorial Frobenius dot product is defined as

$$\begin{aligned} \langle \cdot, \cdot \rangle_{\text{Frobenius}} &: \mathbb{R}^{\overbrace{k \times \dots \times k}^{p \text{ times}}} \times \mathbb{R}^{\overbrace{k \times \dots \times k}^{p \text{ times}}} \rightarrow \mathbb{R}, \\ \langle \mathbf{A}, \mathbf{B} \rangle_{\text{Frobenius}} &= \mathbf{A} : \mathbf{B} = \sum_{i_1, i_2, \dots, i_p=1}^k A_{i_1 i_2 \dots i_p} B_{i_1 i_2 \dots i_p} \in \mathbb{R}. \end{aligned}$$

We note that \mathbf{M}^i , $i = 1, 2, \dots, k$ are p^{th} -order tensors computed in the offline computational stage and their dimensions do not depend on the full space dimension. For FE and FV, the tensorial POD representations are the same except for the type of products used in the computation of $\mathbf{M}^i_{i_1 i_2 \dots i_p}$ in (5), which now are continuous and replace the sum of products used in the FD case.

Reduced nonlinearities depending on space derivatives are treated similarly as in Equations (3–5) because POD expansion of x_x (space derivative of x) is $U_x \tilde{\mathbf{x}}$, where $U_x \in \mathbb{R}^{n \times k}$ contains the space derivatives of POD basis functions \mathbf{u}_i , $i = 1, 2, \dots, k$, $\mathbf{u}_i \in \mathbb{R}^n$.

2.3. POD and discrete empirical interpolation method

The EIM and its discrete version (DEIM) were developed to approximate the nonlinear term allowing an effectively affine offline–online computational decomposition. Both interpolation methods provide an efficient way to approximate nonlinear functions. They were successfully used in POD framework for FD, FE, and FV discretization methods. A description of EIM in connection with the reduced basis framework and a posteriori error bounds can be found in [58, 59].

The DEIM implementation is based on a POD approach combined with a greedy algorithm, whereas the EIM implementation relies on a greedy algorithm [29].

For $m \ll n$, the FD POD/DEIM nonlinear term approximation is

$$\tilde{N}(\tilde{\mathbf{x}}) \approx \underbrace{U^T V (P^T V)^{-1}}_{\text{pre-computed } k \times m} \underbrace{\mathbf{F}(P^T (\tilde{\mathbf{x}} + U \tilde{\mathbf{x}}))}_{m \times 1},$$

where $V \in \mathbb{R}^{n \times m}$ gathers the first m POD basis modes of nonlinear function \mathbf{F} , whereas $P \in \mathbb{R}^{n \times m}$ is the DEIM interpolation selection matrix. The core of DEIM procedure consists in a greedy algorithm, which inductively constructs P from the linearly independent set V [32, s.3].

The POD/DEIM approximation of (3) is

$$\tilde{N}(\tilde{\mathbf{x}}) \approx \underbrace{U^T V (P^T V)^{-1}}_{\text{pre-computed } k \times m} \underbrace{(P^T U \tilde{\mathbf{x}})^P}_{m \times 1}, \quad (6)$$

where vector powers are taken componentwise and $P^T V \in \mathbb{R}^{m \times m}$, $P^T U \in \mathbb{R}^{m \times k}$. $P^T U$ is also recommended for pre-computation in the offline stage. In order to select an appropriate number of DEIM interpolation points, we seek guidance in the spectrum analysis of the nonlinear snapshots correlation matrix. We are particularly interested in the eigenvalues rate of descent. Lemma (3.2) in [32, s.3.2] provides an error bound for the DEIM approximation $\tilde{N}(\tilde{\mathbf{x}})$ described by Equation 6, which can be reasonably estimated by the largest POD eigenvalue of the snapshots correlation matrix not taken into account by POD basis V .

Discrete empirical interpolation method has developed in several research directions, for example, rigorous state space error bounds [31], a posteriori error estimation [60], 1D FitzHugh–Nagumo model [32], 1D simulating neurons model [61], 1D nonlinear thermal model [62], 1D Burgers equation [30, 63], two-dimensional (2D) nonlinear miscible viscous fingering in porous medium [64], oil reservoirs models [65], and 2D SWE model [66].

We emphasize that only few POD/DEIM studies with FE and FV methods were performed, for example, for electrical networks [67] and for a 2D ignition and detonation problem [68]. Flow simulations past a cylinder using a hybrid-reduced approach combining the quadratic expansion method and DEIM are available in [69].

3. COMPUTATIONAL COMPLEXITY OF THE REDUCED P^{TH} -ORDER NONLINEAR REPRESENTATIONS. ROMS OFFLINE STAGE DISCUSSION

We will focus on FD-reduced order p^{th} -order polynomial nonlinearities formulas introduced in Equations (3, 4, and 6). We begin with an important observation. For POD ROMs construction, the usual approach is to store each of the state variables separately and to project every model equations to a different POD basis corresponding to a state variable whose time derivative is present. This is also the procedure we employed for this study. In this context, n does not denote the total number of state variables but only the number of state variables of the same type, which most of the time is equal with the number of mesh points. Consequently, from now on, we refer to n as the number of spatial points.

Proper orthogonal decomposition representation is computed with a complexity of $\mathcal{O}(p \times k \times n + (p - 1) \times n + k \times n)$ and the POD/DEIM term requires $\mathcal{O}(p \times k \times m + (p - 1) \times m + k \times m)$ basic operations in the online stage. Tensorial POD nonlinear term has a complexity of $\mathcal{O}(k^{p+1})$. Although POD computational complexity still depends on the full space dimension, the other two tensorial POD and POD/DEIM are independent of n . Table I describes the number of operations required to compute the projected p^{th} -order polynomial nonlinearity for each of the three ROMs approaches and various values of n, k, m, p .

Clearly, POD/DEIM provides the fastest nonlinear terms computations in the online stage. For quadratic nonlinearities, that is, $p = 2$ and $n = 10^5$, POD/DEIM outperforms POD and POD tensorial by $10^3 \times$ and $25 \times$, respectively. But these performances are not necessarily translated into the

Table I. Number of floating-point operations in the online stage for different numbers of spatial points n , POD modes k , DEIM points m , and polynomial orders p .

| n | k | m | p | POD | POD/DEIM | Tensorial POD |
|--------|-----|-----|-----|------------|----------|---------------|
| 10^3 | 10 | 10 | 2 | 31,000 | 310 | 2990 |
| 10^3 | 10 | 10 | 3 | 42,000 | 420 | 29,990 |
| 10^3 | 10 | 10 | 4 | 53,000 | 530 | 299,990 |
| 10^4 | 30 | 50 | 2 | 910,000 | 4550 | 80,970 |
| 10^4 | 30 | 50 | 3 | 1,220,000 | 6100 | 2,429,970 |
| 10^5 | 50 | 100 | 2 | 15,100,000 | 15,100 | 374,950 |
| 10^5 | 50 | 100 | 3 | 20,200,000 | 20,200 | 18,749,950 |
| 10^5 | 50 | 100 | 4 | 25,300,000 | 25,300 | 937,499,950 |

POD, proper orthogonal decomposition; DEIM, discrete empirical interpolation method.

same CPU time rates for solving the ROMs solutions because other more time consuming calculations may be needed (reduced Jacobians computations and their LU decompositions). It was already proven by [66] that for an SWE model, DEIM decreases the computational complexity of the POD model by 60× for full space dimensions $n \geq 60,000$ and leads to a CPU time reduction proportional to n . CPU times and error magnitudes comparisons will be discussed in Section 6.

For cubic nonlinearities ($p = 3$), the computational complexities are almost similar for both tensorial POD and POD, whereas for higher nonlinearities (e.g., $p = 4$), tensorial POD cost becomes prohibitive.

In the context of reduced optimization, the offline stage computational complexity weights heavily in the final CPU time costs because several POD bases updates and DEIM interpolation points recalculations are needed during the minimization process. Because the proposed schemes are implicit in time, we need to compute the reduced Jacobians as a part of a Newton-type solver. For the current study, we choose to calculate the derivatives exactly for all three ROMs. Consequently, some reduced coefficients such as tensors \mathbf{M}^i defined in (5) must be calculated for all three reduced approaches including POD/DEIM in the offline stage.

A simple evaluation suggests that POD/DEIM offline stage will be slower than the corresponding tensorial POD and POD stages because more SVD computations are required in addition to particular POD/DEIM coefficients and DEIM index points calculations (see Table II). A more careful examination reveals that we can exploit the structure of POD/DEIM nonlinear term (6) like in the tensorial POD approach (4,5) and provide a fast calculation for \mathbf{M}^i .

Thus, let us denote the pre-computed term and $P^T U$ in (6) by $E = U^T V (P^T V)^{-1} \in \mathbb{R}^{k \times m}$ and $U^m = P^T U \in \mathbb{R}^{m \times k}$, where m is the number of DEIM points.

The p tensor \mathbf{M}^i can be computed as follows:

$$\mathbf{M}_{i_1 i_2 \dots i_p}^i = \sum_{l=1}^m E_{il} U_{li_1}^m U_{li_2}^m \dots U_{li_p}^m \in \mathbb{R}, \quad , i, i_1, i_2, \dots, i_p = 1, 2, \dots, k. \quad (7)$$

Clearly, this estimation is less computationally expensive than (5) because the summation stops at $m \ll n$.

Now, we are making an important observation. Formula (7) makes use of DEIM interpolation points of the nonlinear term $\tilde{N}(\tilde{\mathbf{x}})$ to produce fast computation of the tensors $\mathbf{M}^i, i = 1, 2, \dots, k$. The computational results for the SWE model showed that these surrogate tensors are not approximating well the ones defined by the tensorial POD formula (5). Because all the proposed ROMs require $\mathbf{M}^i, i = 1, 2, \dots, k$ to compute their reduced derivatives (see Table II, line 6), it means that the corresponding reduced Jacobians are not necessarily the same. However, this does not heavily impact the solution of the POD/DEIM SWE model as we will see in Section 6.

Table II. ROMs offline stage procedures—POD coefficients $\mathbf{M}_1^i, \mathbf{M}_2^i$ are required for reduced Jacobian calculation. Only tensorial POD uses them also for right-hand side terms computations during the quasi-Newton iterations required by Gustafsson’s nonlinear ADI FD scheme.

| POD | Tensorial POD | POD/DEIM |
|--|---|--|
| Generate snapshots SVD for u, v, ϕ | Generate snapshots SVD for u, v, ϕ | Generate snapshots SVD for u, v, ϕ |
| — | — | SVD for all nonlinear terms |
| — | — | DEIM index points for all nonlinear terms |
| Calc. POD coefficients \mathbf{M}_j^i (11) (Reduced Jac. calc.) | Calc. POD coefficients \mathbf{M}_j^i (11), (Reduced Jac. and right-hand side terms calc.) | Calc. POD coefficients \mathbf{M}_j^i (7), (Reduced Jac. calc.) |
| — | — | Calc. all POD/DEIM coef. such as E_{11} |

ROM, reduced order model; ADI, alternating direction fully implicit; FD, finite difference; POD, proper orthogonal decomposition; DEIM, discrete empirical interpolation method; SVD, singular values decomposition.

4. THE SHALLOW WATER EQUATIONS

In meteorological and oceanographic problems, one is often not interested in small time steps because the discretization error in time is small compared with the discretization error in space. SWE can be used to model Rossby and Kelvin waves in the atmosphere, rivers, lakes, and oceans as well as gravity waves in a smaller domain. The alternating direction fully implicit (ADI) scheme [70] considered in this paper is first-order in both time and space and it is stable for large Courant–Friedrichs–Levy (CFL) condition numbers (we tested the stability of the scheme for a CFL condition number equal up to 8.9301). It was also proved that the method is unconditionally stable for the linearized version of the SWE model. Other research work on this topic includes efforts of [71] and [72].

We are solving the SWE model using the β -plane approximation on a rectangular domain [70].

$$\frac{\partial w}{\partial t} = A(w) \frac{\partial w}{\partial x} + B(w) \frac{\partial w}{\partial y} + C(y)w, \quad (x, y) \in [0, L] \times [0, D], \quad t \in (0, t_f], \quad (8)$$

where $w = (u, v, \phi)^T$ is a vector function and u, v are the velocity components in the x -direction and y -direction, respectively. Geopotential is computed using $\phi = 2\sqrt{gh}$, h being the depth of the fluid and g the acceleration due to gravity.

The matrices A, B , and C are

$$A = - \begin{pmatrix} u & 0 & \phi/2 \\ 0 & u & 0 \\ \phi/2 & 0 & u \end{pmatrix}, \quad B = - \begin{pmatrix} v & 0 & 0 \\ 0 & v & \phi/2 \\ 0 & \phi/2 & v \end{pmatrix}, \quad C = \begin{pmatrix} 0 & f & 0 \\ -f & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

where f is the Coriolis term

$$f = \hat{f} + \beta(y - D/2), \quad \beta = \frac{\partial f}{\partial y}, \quad \forall y,$$

with \hat{f} and β constants.

We assume periodic solutions in the x -direction for all three state variables, whereas in the y -direction,

$$v(x, 0, t) = v(x, D, t) = 0, \quad x \in [0, L], \quad t \in (0, t_f]$$

and Neumann boundary condition is considered for u and ϕ .

Initially, $w(x, y, 0) = \psi(x, y)$, $\psi : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$, $(x, y) \in [0, L] \times [0, D]$. Now, we introduce a mesh of $n = N_x \cdot N_y$ equidistant points on $[0, L] \times [0, D]$, with $\Delta x = L/(N_x - 1)$, $\Delta y = D/(N_y - 1)$. We also discretize the time interval $[0, t_f]$ using N_t equally distributed points and $\Delta t = t_f/(N_t - 1)$. Next, we define vectors of unknown variables of dimension n containing approximate solutions such as

$$\mathbf{w}(t_N) \approx [w(x_i, y_j, t_N)]_{i=1,2,\dots,N_x, j=1,2,\dots,N_y} \in \mathbb{R}^n, \quad N = 1, 2, \dots, N_t.$$

The semi-discrete equations of SWE (8) are as follows:

$$\begin{aligned} \mathbf{u}' &= -F_{11}(\mathbf{u}, \phi) - F_{12}(\mathbf{u}, \mathbf{v}) + \mathbf{F} \odot \mathbf{v}, \\ \mathbf{v}' &= -F_{21}(\mathbf{u}, \mathbf{v}) - F_{22}(\mathbf{v}, \phi) - \mathbf{F} \odot \mathbf{u}, \\ \phi' &= -F_{31}(\mathbf{u}, \phi) - F_{32}(\mathbf{v}, \phi), \end{aligned}$$

where \odot is the MATLAB componentwise multiplication operator, $\mathbf{u}', \mathbf{v}', \phi'$ denote semi-discrete time derivatives, $\mathbf{F} = \underbrace{[\mathbf{f}, \mathbf{f}, \dots, \mathbf{f}]}_{N_x}$ stores Coriolis components $\mathbf{f} = [f(y_j)]_{j=1,2,\dots,N_y}$, whereas the

nonlinear terms F_{i1} and F_{i2} , $i = 1, 2, 3$, involving derivatives in x and y directions, respectively, are defined as follows:

$$\begin{aligned} F_{i1}, F_{i2} : \mathbb{R}^n \times \mathbb{R}^n &\rightarrow \mathbb{R}^n, i = 1, 2, 3, F_{11}(\mathbf{u}, \boldsymbol{\phi}) = \mathbf{u} \odot A_x \mathbf{u} + \frac{1}{2} \boldsymbol{\phi} \odot A_x \boldsymbol{\phi}, \\ F_{12}(\mathbf{u}, \mathbf{v}) &= \mathbf{v} \odot A_y \mathbf{u}, F_{21}(\mathbf{u}, \mathbf{v}) = \mathbf{u} \odot A_x \mathbf{v}, F_{22}(\mathbf{v}, \boldsymbol{\phi}) = \mathbf{v} \odot A_y \mathbf{v} \\ &+ \frac{1}{2} \boldsymbol{\phi} \odot A_y \boldsymbol{\phi}, \\ F_{31}(\mathbf{u}, \boldsymbol{\phi}) &= \frac{1}{2} \boldsymbol{\phi} \odot A_x \mathbf{u} + \mathbf{u} \odot A_x \boldsymbol{\phi}, F_{32}(\mathbf{v}, \boldsymbol{\phi}) = \frac{1}{2} \boldsymbol{\phi} \odot A_y \mathbf{v} + \mathbf{v} \odot A_y \boldsymbol{\phi}. \end{aligned}$$

Here, $A_x, A_y \in \mathbb{R}^{n \times n}$ are constant coefficient matrices for discrete first-order and second-order differential operators, which take into account the boundary conditions.

The numerical scheme was implemented in Fortran and uses a sparse matrix environment. For operations with sparse matrices, we employed that SPARSEKIT library [73] and the sparse linear systems obtained during the quasi-Newton iterations were solved using MGMRES library [74–76]. Here we did not decouple the model equations like in Stefanescu and Navon [66], where the Jacobian is either block cyclic tridiagonal or block tridiagonal. We followed this approach because we plan to implement a 4D Var data assimilation system based on ADI SWE, and the adjoints of the decoupled systems cannot be solved with the same implicit scheme applied for solving the forward model.

5. REDUCED ORDER SHALLOW WATER EQUATION MODELS

Here, we will not describe the entire POD SWE, tensorial POD SWE, and POD/DEIM SWE discrete models but only introduce the projected nonlinear term F_{11} for all three ROMs. ADI discrete equations were projected onto reduced POD subspaces and a detailed description of the reduced equations for POD and POD/DEIM is available in [66].

Depending on the type of reduced approaches, the Galerkin projected nonlinear term \tilde{F}_{11} assumes the following form

POD

$$\tilde{F}_{11} = U^T F_{11} = \underbrace{U^T}_{k \times n} \left(\underbrace{(U \tilde{\mathbf{u}} \odot (U_x \tilde{\mathbf{u}}))}_{n \times 1} \right) + \frac{1}{2} \underbrace{U^T}_{k \times n} \left(\underbrace{(\Phi \tilde{\boldsymbol{\phi}} \odot (\Phi_x \tilde{\boldsymbol{\phi}}))}_{n \times 1} \right), \quad (9)$$

where U and Φ contains the POD bases corresponding to state variables u and ϕ while the POD basis derivatives are included in $U_x = A_x U \in \mathbb{R}^{n \times k}$ and $\Phi_x = A_x \Phi \in \mathbb{R}^{n \times k}$.

Tensorial POD

$$\tilde{F}_{11} = U^T F_{11} \in \mathbb{R}^k; [\tilde{F}_{11}]_i = \langle \mathbf{M}_1^i, \tilde{\mathbf{U}} \rangle_{\text{Frobenius}} + \langle \mathbf{M}_2^i, \tilde{\boldsymbol{\Phi}} \rangle_{\text{Frobenius}}, i = 1, 2, \dots, k; \quad (10)$$

$$\tilde{\mathbf{U}} = [\tilde{\mathbf{U}}_{i,j}]_{i,j=1,\dots,k} \in \mathbb{R}^{k \times k}; \tilde{\mathbf{U}}_{i,j} = \tilde{\mathbf{u}}_i \tilde{\mathbf{u}}_j \in \mathbb{R}, \tilde{\boldsymbol{\Phi}} = [\tilde{\boldsymbol{\Phi}}_{i,j}]_{i,j=1,\dots,k} \in \mathbb{R}^{k \times k}; \tilde{\boldsymbol{\Phi}}_{i,j} = \tilde{\boldsymbol{\phi}}_i \tilde{\boldsymbol{\phi}}_j \in \mathbb{R},$$

where $\tilde{\mathbf{u}} \in \mathbb{R}^k$ and $\tilde{\boldsymbol{\phi}} \in \mathbb{R}^k$ are reduced state variables.

$$\begin{aligned} \mathbf{M}_1^i &= [\mathbf{M}_{1i_1 i_2}^i]_{i_1, i_2=1,\dots,k} \in \mathbb{R}^{k \times k}; \mathbf{M}_{1i_1 i_2}^i = \sum_{l=1}^n U_{li} U_{li_1} U_{xli_2} \in \mathbb{R}, \\ \mathbf{M}_2^i &= [\mathbf{M}_{2i_1 i_2}^i]_{i_1, i_2=1,\dots,k} \in \mathbb{R}^{k \times k}; \mathbf{M}_{2i_1 i_2}^i = \sum_{l=1}^n U_{li} \Phi_{li_1} \Phi_{xli_2} \in \mathbb{R}, \end{aligned} \quad (11)$$

and U_x and Φ_x were defined earlier.

POD/DEIM

$$\tilde{F}_{11} \approx \underbrace{U^T V_{F_{11}} (P_{F_{11}}^T V_{F_{11}})^{-1}}_{\text{pre-computed } k \times m} \left(\underbrace{(P_{F_{11}}^T U \tilde{\mathbf{u}})}_{m \times 1} \odot \underbrace{(P_{F_{11}}^T U_x \tilde{\mathbf{u}})}_{m \times 1} + \underbrace{(P_{F_{11}}^T \Phi \tilde{\phi})}_{m \times 1} \odot \underbrace{(P_{F_{11}}^T \Phi_x \tilde{\phi})}_{m \times 1} \right), \quad (12)$$

where $V_{F_{11}} \in \mathbb{R}^{n \times m}$ collects the first m POD basis modes of nonlinear function F_{11} , whereas $P_{F_{11}} \in \mathbb{R}^{n \times m}$ is the DEIM interpolation selection matrix. Let us denote the pre-computed term by $E_{11} = U^T V_{F_{11}} (P_{F_{11}}^T V_{F_{11}})^{-1}$.

Tensors as \mathbf{M}_1^i and \mathbf{M}_2^i (11) must also be computed in the case of POD and POD/DEIM because the analytic form of reduced Jacobian was employed. This approach reduces the CPU time of POD model because usually, the reduced Jacobians are obtained by projecting the full Jacobian at every time step. A generalization of DEIM to approximate operators has not been yet developed but has the ability to decrease more the computational complexity of POD/DEIM approach. Some related work includes [77], who developed multi-component EIM for deriving affine approximations for continuous vector valued functions. Wirtz *et al.* [60] introduced the matrix DEIM approach to approximate the Jacobian of a nonlinear function. Chaturantabut [30] proposed a sampling strategy centered on the trajectory of the nonlinear functions in order to approximate the reduced Jacobian. An extension for nonlinear problems that do not have componentwise dependence on the state has been introduced in [78].

Table II contains the procedure list required by all three algorithms in the offline stage. \mathbf{M}_1^i , \mathbf{M}_2^i , and E_{11} are POD and POD/DEIM coefficients related to nonlinear term F_{11} , similar coefficients being required for computation of other reduced nonlinear terms.

6. NUMERICAL RESULTS

For all tests, we derived the initial conditions from the initial height condition no. 1 of [79], that is,

$$h(x, y, 0) = H_0 + H_1 + \tanh\left(9 \frac{D/2 - y}{2D}\right) + H_2 \text{sech}^2\left(9 \frac{D/2 - y}{2D}\right) \sin\left(\frac{2\pi x}{L}\right).$$

The initial velocity fields are derived from the initial height field using the geostrophic relationship

$$u = \left(\frac{-g}{f}\right) \frac{\partial h}{\partial y}, \quad v = \left(\frac{g}{f}\right) \frac{\partial h}{\partial x}.$$

We use the following constants $L = 6000km$, $D = 4400km$, $\hat{f} = 10^{-4}s^{-1}$, $\beta = 1.5 \cdot 10^{-11}s^{-1}m^{-1}$, $g = 10ms^{-2}$, $H_0 = 2000m$, $H_1 = 220m$, $H_2 = 133m$. Figure 1 depicts the initial geopotential isolines and the geostrophic wind field.

Most of the depicted results are obtained in the case when the domain is discretized using a mesh of $376 \times 276 = 103,776$ points, with $\Delta x = \Delta y = 16$ km. We select two integration time windows of 24 and 3 h and we use 91 time steps ($N_t = 91$) with $\Delta t = 960$ s and $\Delta t = 120$ s.

Alternating direction fully implicit FD SWE scheme proposed by Gustafsson [70] is first employed in order to obtain the numerical solution of the SWE model. The implicit scheme allows us to integrate in time at the CFL condition of $\sqrt{gh}(\Delta t/\Delta x) < 8.9301$.

The nonlinear algebraic systems of ADI FD SWE scheme is solved using quasi-Newton method, and the LU decomposition is performed every six time steps.

We derive the ROMs by employing a Galerkin projection. The POD basis functions are constructed using 180 snapshots (91 time steps provide 180 state variables snapshots by taking into account the intermediary solutions of the ADI scheme too) obtained from the numerical solution of the full-order ADI FD SWE model at equally spaced time steps for each time interval $[0, 24h]$ and

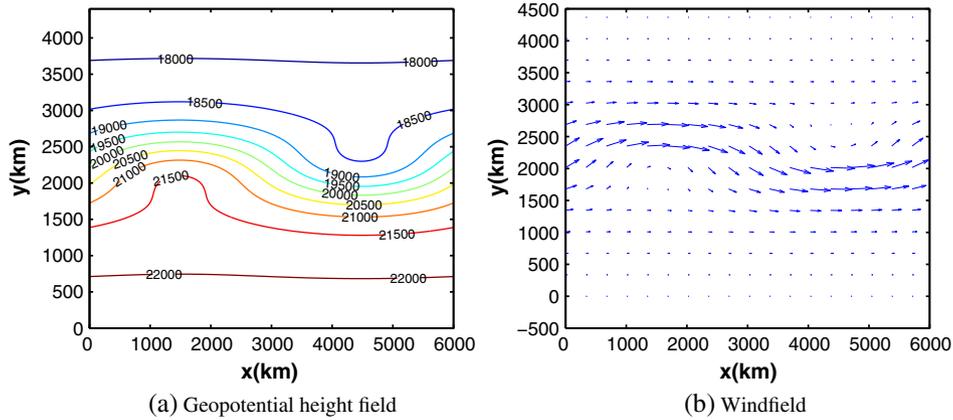


Figure 1. Initial condition: geopotential height field for the Grammeltvedt initial condition and windfield (the velocity unit is 1 km/s) calculated from the geopotential field using the geostrophic approximation.

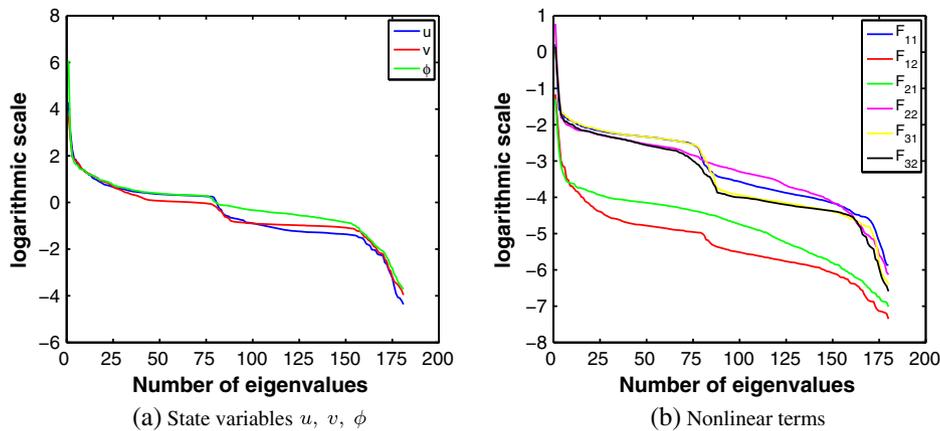


Figure 2. The decay around the singular values of the snapshots solutions for u , v , and ϕ and nonlinear terms for $\Delta t = 960$ s and integration time window of 24 h.

$[0, 3h]$. Figures 2 and 3 show the decay around the eigenvalues of the snapshot solutions for u , v , ϕ and the nonlinear snapshots F_{11} , F_{12} , F_{21} , F_{22} , and F_{31} , F_{32} . These spectrums underline that 70 and 180 DEIM points are sufficient for generating accurate nonlinear DEIM approximations for both the SWE terms and spatial configurations with errors of order of 10^{-6} . We notice that the singular values decay much faster when the model is integrated for 3 h. Consequently, this translates in a more accurate solution representations for all three ROM methods using the same number of POD modes. For both time configurations and all tests in this study, the dimensions of the POD bases for each variable are taken to be 50, capturing more than 99% of the system energy. The largest neglected eigenvalues corresponding to state variables u , v , and ϕ are 2.23, 1.16, and 2.39 for $t_f = 24$ h and 0.0016, 0.0063, and 0.0178 for $t_f = 3$ h, respectively.

Next, we apply DEIM algorithm and calculate the interpolation points to improve the efficiency of the POD Galerkin terms and to achieve a complexity reduction of the nonlinear terms with a complexity proportional to the number of reduced variables, as in the case of tensorial POD. Figures 4 and 5 illustrate the distribution of the first 100 spatial points selected by the DEIM algorithm together with the isolines of the nonlinear terms statistics. Each of these statistics contains the maximum values of the corresponding nonlinear term over time in every space location. Maximum is preferred instead of time averaging because a better correlation between location of DEIM points and physical structures was observed in the former case.

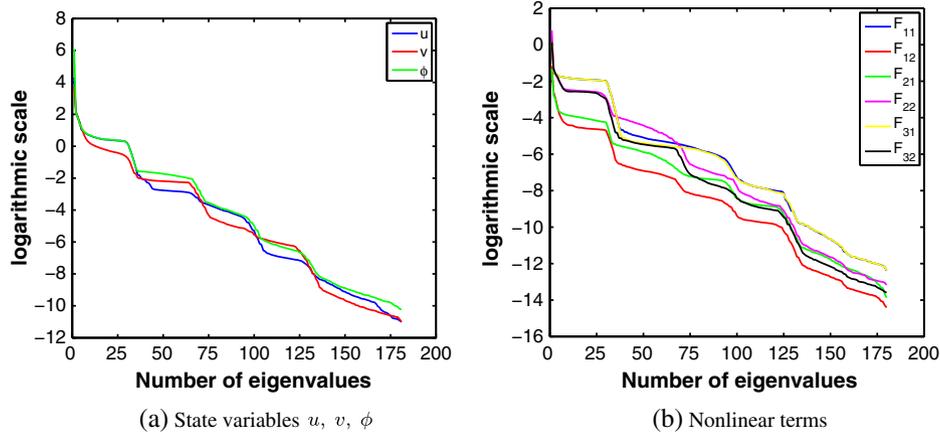


Figure 3. The decay around the singular values of the snapshots solutions for u , v , ϕ and nonlinear terms for $\Delta t = 120$ s and a time integration window of 3 h.

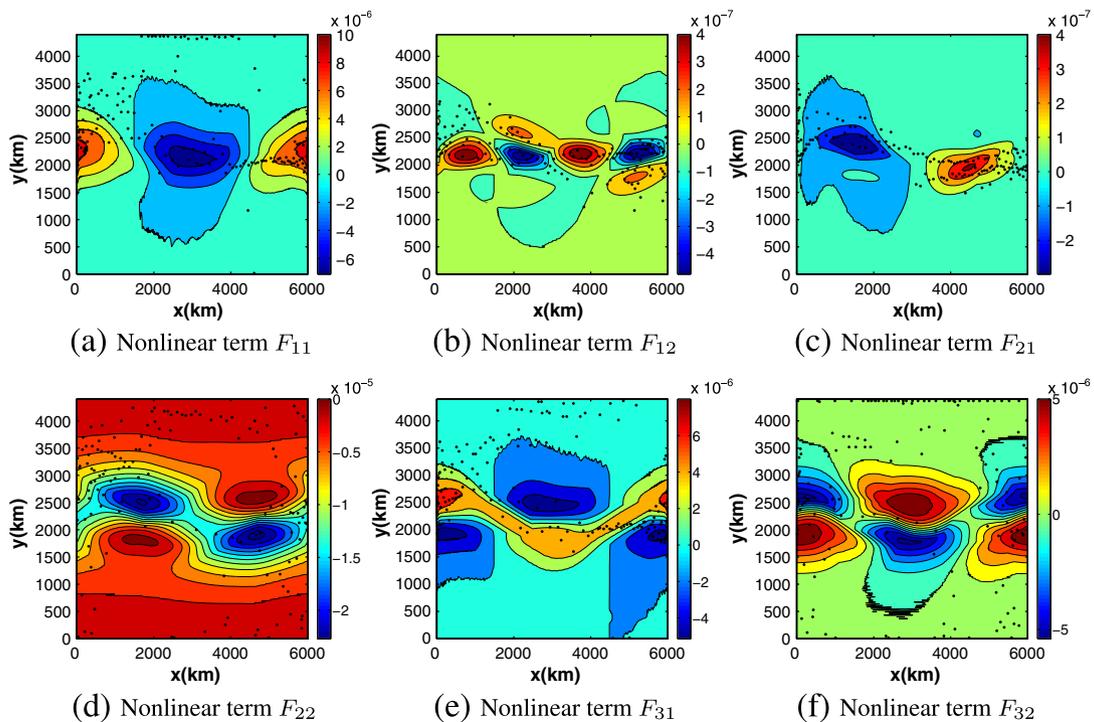


Figure 4. The first 100 DEIM interpolation points corresponding to all nonlinear terms in the SWE model for time integration window for 24 h. The background consists in isolines of the maximum values of the nonlinear terms over time.

Another way to compute the interpolation points would be to apply the greedy algorithms simultaneously as in [80]. This approach will limit the number of selected DEIM points leading to faster online surrogates.

However, in most cases, the spatial positions of the interpolation points do not follow the nonlinear statistic structures. This is more visible in Figure 5 for $t_f = 3$ h. This proves that DEIM algorithm does not particularly take into account the physical structures of the nonlinear terms but searches (in a greedy manner) to minimize the error (residual) between each column of the input basis (POD basis of the nonlinear term snapshots) and its proposed low-rank approximations [66, p.16].

This can be clearly seen in Figure 6, where the isolines of both the absolute values of DEIM residuals over time and the 9^{th} residual absolute values are depicted. It is the case of the nonlinear

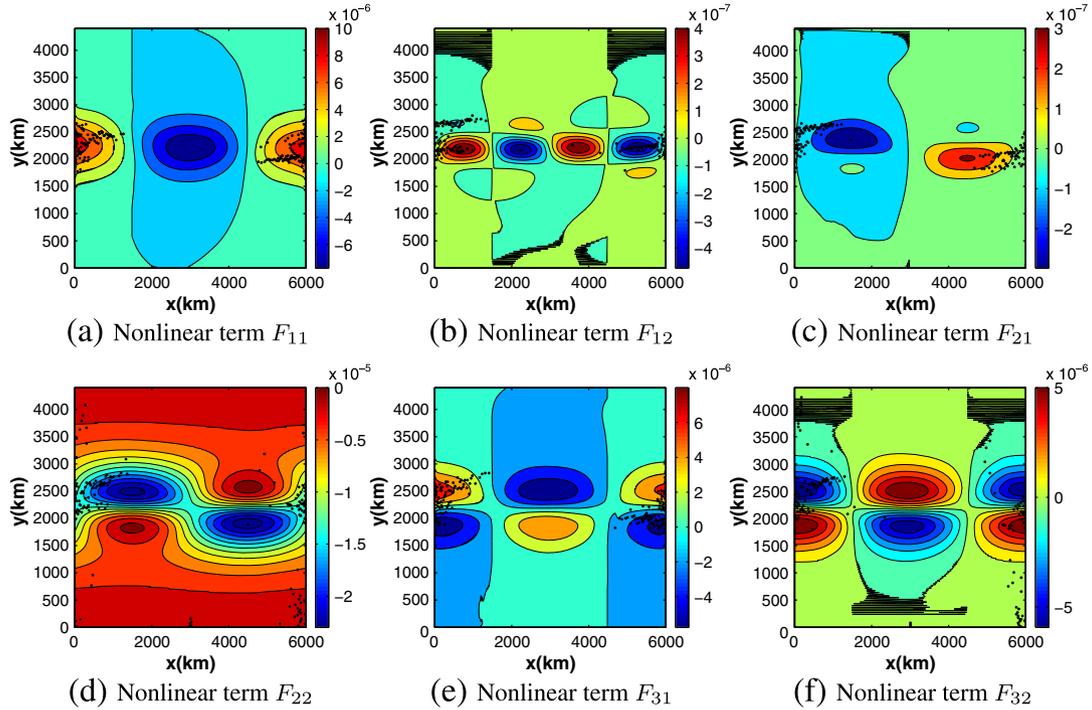


Figure 5. A total of 100 DEIM interpolation points corresponding to all nonlinear terms in the SWE model for time integration window of 3 h. The background consists in isolines of the maximum values of the nonlinear terms over time. Most of the points are concentrated in the region with larger errors depicted in Figure 8.

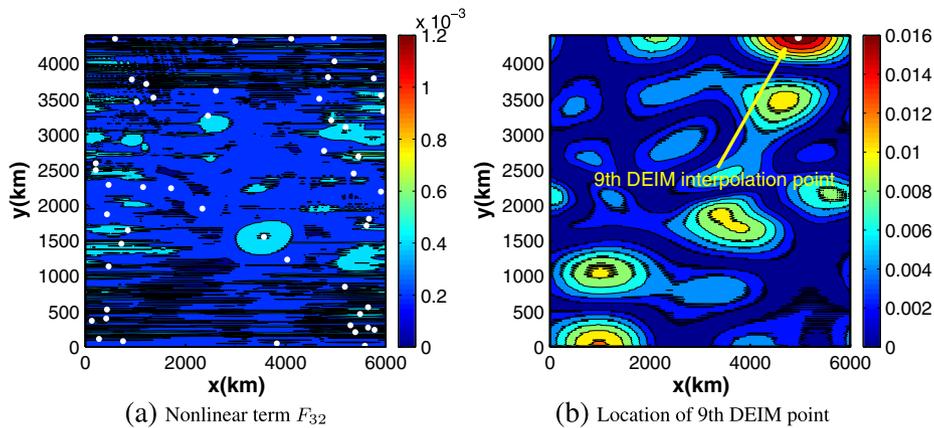


Figure 6. The first 40 DEIM interpolation points corresponding to F_{32} in the SWE model for time integration window of 24 h (panel a). The background consists in isolines of the averaged values of the absolute values of the DEIM residuals over time (panel a) and the 9th residual absolute values (panel b).

term F_{32} and its snapshots that were obtained after a 24-h time integration of the SWE model. The white dots in Figure 6(a) are the first 40 DEIM points, whereas in Figure 6(b), the arrow points out to the 9th DEIM interpolation point located outside the nonlinear statistics areas presented in Figure 4(f).

Figures 7 and 8 depict the grid point absolute error of the POD, tensorial POD, and POD/DEIM solutions with respect to the full solutions. For POD/DEIM ROM, we use 180 interpolation points. The magnitude of the errors is similar for each of the method proposed in this study. Moreover,

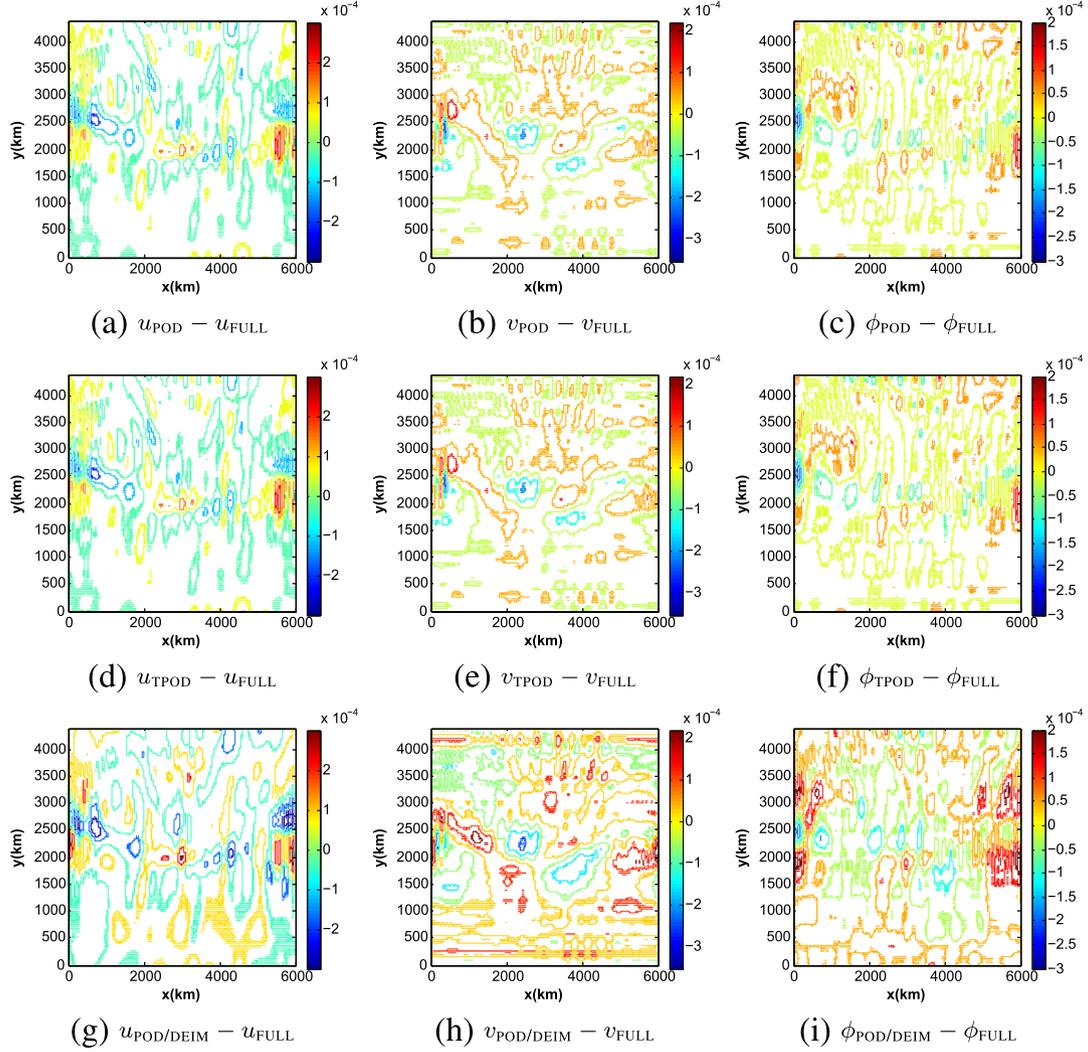


Figure 7. Absolute errors between POD, tensorial POD, and POD/DEIM solutions and the full trajectories at $t = 24$ h ($\Delta t = 960$ s). The number of DEIM points was taken 180.

we observed that error isolines distribution in Figure 8 is well correlated with the location of interpolation points illustrated in Figure 5 underlying the empirical characteristics of DEIM.

In addition, we propose two metrics to quantify the accuracy levels of POD, tensorial POD, and POD/DEIM approaches. First, we use the following norm

$$\frac{1}{N_t} \sum_{i=1}^{N_t} \frac{\|w^{\text{FULL}}(:, t_i) - w^{\text{ROM}}(:, t_i)\|_2}{\|w^{\text{FULL}}(:, t_i)\|_2}.$$

$i = 1, 2, \dots, t_f$ and calculate the relative errors for all three variables of SWE model $w = (u, v, \phi)$. The results are presented in Table III. We perform numerical experiments using two configurations of the numbers of DEIM points depending on the SWE time integration window. For $t_f = 24$ h, we used $m = 80$ and 180 while for $t_f = 3$ h, 70 and 180 DEIM points were considered. For the 24-h tests, we notice that more than 70 number of DEIM points are needed for convergence of quasi-Newton method for POD/DEIM SWE scheme explaining the choice of two DEIM points configurations.

Root mean square error is also employed to compare the ROMs. Tables IV and V show the RMSE for final times together with the CPU times of the online stage of ROMs.

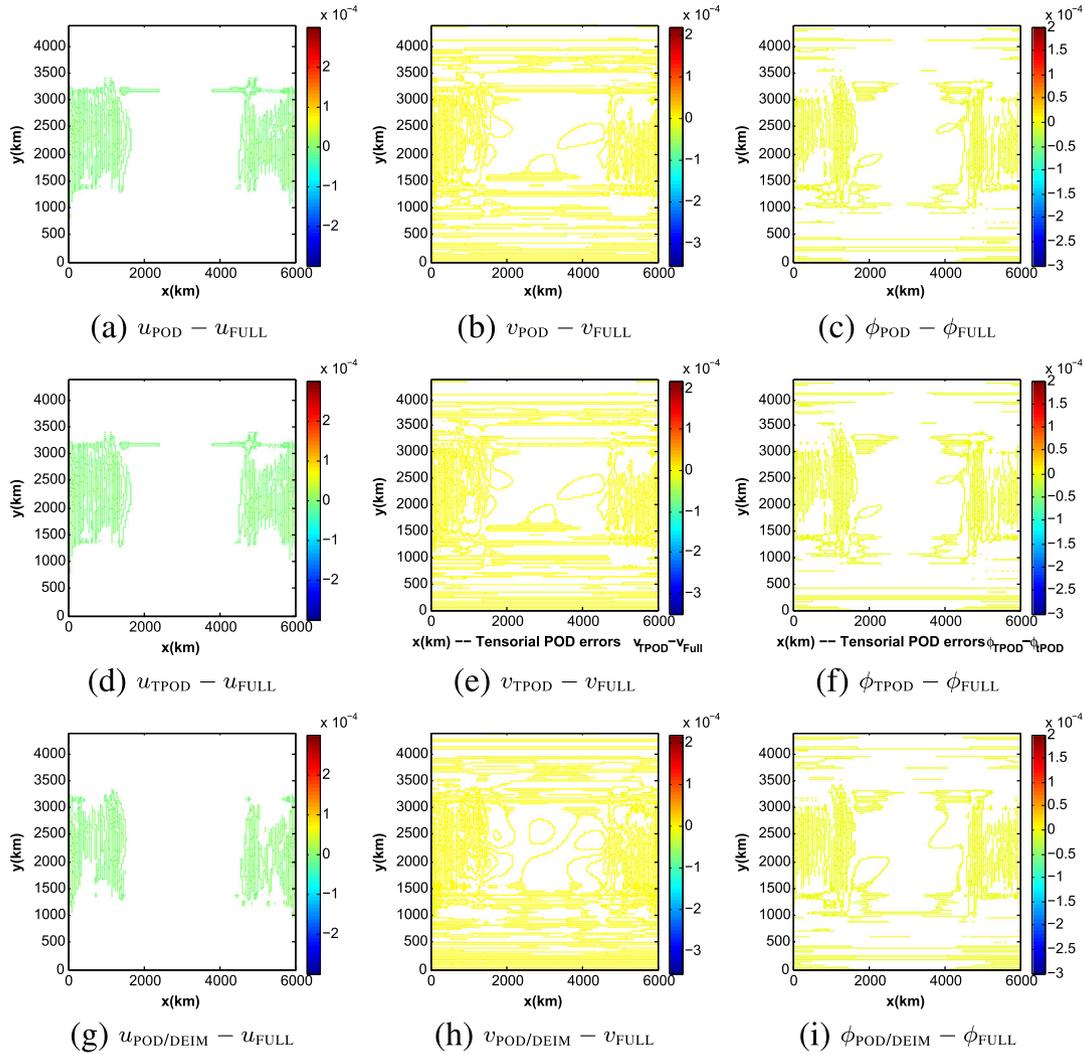


Figure 8. Absolute errors between POD, tensorial POD, and POD/DEIM solutions and the full trajectories at $t = 3$ h ($\Delta t = 120$ s). The number of DEIM points was taken 180.

Table III. Relative errors for each of the model variables for $t_f = 24$ h (left) and $t_f = 3$ h (right). The POD bases dimensions were taken 50. The results for various numbers of DEIM points are shown.

| | POD | Tensorial POD | POD/DEIM $m = 180$ | POD/DEIM $m = 80$ |
|--------|----------|---------------|--------------------|-------------------|
| u | 1.276e-3 | 1.276e-3 | 1.622e-3 | 3.96e-3 |
| v | 3.426e-3 | 3.426e-3 | 4.639e-3 | 1.198e-2 |
| ϕ | 2.110e-5 | 2.110e-5 | 2.489e-5 | 5.77e-5 |

| | POD | Tensorial POD | POD/DEIM $m = 180$ | POD/DEIM $m = 70$ |
|--------|----------|---------------|--------------------|-------------------|
| u | 7.711e-6 | 7.711e-6 | 7.965e-6 | 9.301e-6 |
| v | 1.665e-5 | 1.666e-5 | 1.73e-5 | 1.975e-5 |
| ϕ | 1.389e-7 | 1.389e-7 | 1.426e-7 | 1.483e-7 |

POD, proper orthogonal decomposition; DEIM, discrete empirical interpolation method.

Table IV. CPU time gains and the root mean square errors for each of the model variables at $t_f = 24$ h. Number of POD modes was $k = 50$ and two tests with different numbers of DEIM points $m = 70, 180$ were simulated.

| | Full ADI SWE | POD | Tensorial POD | POD/DEIM $m = 180$ | POD/DEIM $m = 80$ |
|----------|--------------|-----------|---------------|--------------------|-------------------|
| CPU time | 1813.992s | 191.785 | 2.491 | 1.046 | 0.663 |
| u | — | 9.095e-3 | 9.095e-3 | 1.555e-2 | 5.574e-2 |
| v | — | 8.812e-3 | 8.812e-3 | 1.348e-2 | 4.907e-2 |
| ϕ | — | 6.987e-3e | 6.987e-3 | 1.13e-2 | 5.288e-2 |

POD, proper orthogonal decomposition; DEIM, discrete empirical interpolation method; ADI, alternating direction fully implicit; SWE, shallow water equation.

Table V. CPU time gains and the root mean square errors for each of the model variables at $t_f = 3$ h for a 3-h time integration window. Number of POD modes was $k = 50$ and two tests with different numbers of DEIM points $m = 70, 180$ were simulated.

| | Full ADI SWE | POD | Tensorial POD | POD/DEIM $m = 180$ | POD/DEIM $m = 70$ |
|----------|--------------|-----------|---------------|--------------------|-------------------|
| CPU time | 950.0314s | 161.907 | 2.125 | 0.642 | 0.359 |
| u | — | 5.358e-5 | 5.358e-5 | 5.646e-5 | 7.453e-5 |
| v | — | 2.728e-5 | 2.728e-5 | 3.418e-5 | 4.233e-5 |
| ϕ | — | 8.505e-5e | 8.505e-5 | 8.762e-5 | 9.212e-5 |

POD, proper orthogonal decomposition; DEIM, discrete empirical interpolation method; ADI, alternating direction fully implicit; SWE, shallow water equation.

Thus, for 103,776 spatial points, tensorial POD method reduces the computational complexity of the nonlinear terms in comparison with the POD ADI SWE model and overall decreases the computational time with a factor of $77\times$ for 24 h time integration and $76\times$ for a 3-h time window integration. POD/DEIM outperforms POD being $450\times$ and $250\times$ time faster for 70 and 180 DEIM interpolation points and a time integration window of 3 h. For $t_f = 24$ h, POD/DEIM SWE model is $183\times$ and $289\times$ time faster than POD SWE model for $m = 80$ and 180. In terms of CPU time, tensor POD SWE model is only $3.75\times$ and $2.38\times$ slower than POD/DEIM SWE model for configurations of 80 and 180 DEIM points and $t_f = 24$ h, whereas for $t_f = 3$ h, the new tensorial POD scheme is $5.9\times$ and $3.3\times$ less efficient than POD/DEIM SWE model for $m = 70$ and $m = 180$. This suggests that operations such as Jacobian computations and its LU decomposition required by both reduced order approaches weight more in the overall CPU time cost because the quadratic nonlinear complexity of the tensorial POD requires 374,950 floating-point operations and POD/DEIM only 27,180 ($m = 180$, see Section 3). Given that the implementation effort is much reduced, in the cases of models depending only on quadratic nonlinearities, the tensorial POD poses the appropriate characteristics of a reduced order method and represent a solid alternative to the POD/DEIM approach.

For cubical nonlinearities and larger, tensorial POD loses, its ability to deliver fast calculations (see Table I), thus the POD/DEIM should be employed.

In our case, the Jacobians are calculated analytically and their computations depend only on the reduced space dimension k . However, more gain can be obtained if DEIM would be applied to approximate the reduced Jacobians but this is the subject of future research.

The computational savings and accuracy levels obtained by the ROMs studied in this paper depend on the number of POD modes and number of DEIM points. These numbers may be large in practice in order to capture well the full model dynamics. For example, in the case of a time window integration of 24 h, if someone would ask to increase the ROMs solutions accuracy with only one order of magnitude, the POD basis dimension must be at least larger than 100 that will drastically compromise the time performances of ROMs methods. Elegant solutions to this problem were proposed by Rapún and Vega [81] and Peherstorfer *et al.* [82], where local POD and local DEIM versions were proposed. The idea of a local approach for nonlinear model reduction with local POD and local GNAT was first proposed by Amsallem *et al.* [83]. Machine learning techniques such as

K -means [84–86] can be used for both time and space partitioning. A recent study investigating cluster-based reduced order modeling was proposed by [87].

Figure 9 depicts the efficiency of tensorial POD and POD/DEIM SWE schemes as a function of spatial discretization points in the case of $t_f = 3$ h. We compare the results obtained for eight different mesh configurations $n = 31 \times 23, 61 \times 45, 101 \times 71, 121 \times 89, 151 \times 111, 241 \times$

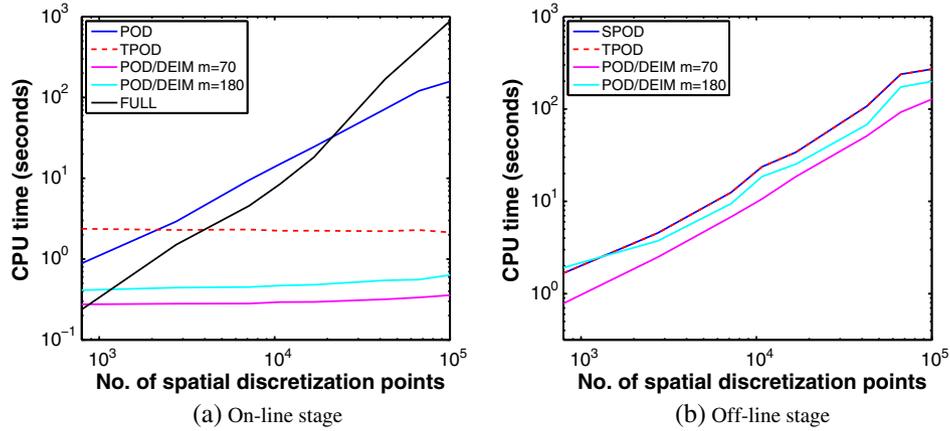


Figure 9. CPU time versus the number of spatial discretization points for $t_f = 3$ h; number of POD modes = 50; two different numbers of DEIM points 70 and 180 have been employed.

Table VI. Relative errors for each of the model variables for $t_f = 24$ h (left) and $t_f = 3$ h (right) using initial conditions different than the ones used to train POD basis. The POD bases dimensions were taken 50. The results for 180 numbers of DEIM points are shown.

| | POD | Tensorial POD | POD/DEIM $m = 180$ |
|--------|----------|---------------|--------------------|
| u | 9.383e-3 | 9.383e-3 | 2.068e-2 |
| v | 3.443e-2 | 3.443e-2 | 6.542e-2 |
| ϕ | 7.627e-5 | 7.627e-5 | 2.148e-4 |

| | POD | Tensorial POD | POD/DEIM $m = 180$ |
|--------|----------|---------------|--------------------|
| u | 9.468e-3 | 9.468e-3 | 9.494e-3 |
| v | 2.413e-2 | 2.413e-2 | 2.59e-2 |
| ϕ | 1.116e-5 | 1.116e-5 | 7.708e-5 |

POD, proper orthogonal decomposition; DEIM, discrete empirical interpolation method.

Table VII. CPU time gains and the root mean square errors for each of the model variables at $t_f = 24$ h. Number of POD modes was $k = 50$ and two tests with different number of DEIM points $m = 180, 70$ were simulated.

| | Full ADI SWE | POD | Tensorial POD | POD/DEIM $m = 180$ |
|----------|--------------|---------|---------------|--------------------|
| CPU time | 1813.992s | 224.447 | 2.431 | 1.024 |
| u | — | 4.73e-2 | 4.73e-2 | 2.08e-1 |
| v | — | 5.69e-2 | 5.69e-2 | 1.17e-1 |
| ϕ | — | 3.93e-2 | 3.93e-2 | 1.15e-1 |

POD, proper orthogonal decomposition; DEIM, discrete empirical interpolation method; ADI, alternating direction fully implicit; SWE, shallow water equation.

177, 301×221 , 376×276 . CPU time performances of the offline and online stages of the ROMs SWE schemes are compared because reduced order optimization algorithms include both phases.

For the online stage, once the number of spatial discretization points exceeds then 151×111 tensorial POD scheme is $10\times$ faster than the POD scheme. The performances of POD/DEIM depend on the number of DEIM points and the numerical results display a $10\times$ time reduction of the CPU

Table VIII. CPU time gains and the root mean square errors for each of the model variables at $t_f = 3$ h for a 3 h time integration window. Number of POD modes was $k = 50$ and one test with 180 number of DEIM points is presented.

| | Full ADI SWE | POD | Tensorial POD | POD/DEIM $m = 180$ |
|----------|--------------|-----------|---------------|--------------------|
| CPU time | 950.0314s | 174.52 | 1.8559 | 0.6399 |
| u | — | $7.23e-2$ | $7.23e-2$ | $7.15e-2$ |
| v | — | $2.48e-2$ | $2.48e-2$ | $3.037e-2$ |
| ϕ | — | $2.88e-2$ | $2.88e-2$ | $2.89e-2$ |

POD, proper orthogonal decomposition; DEIM, discrete empirical interpolation method; ADI, alternating direction fully implicit; SWE, shallow water equation.

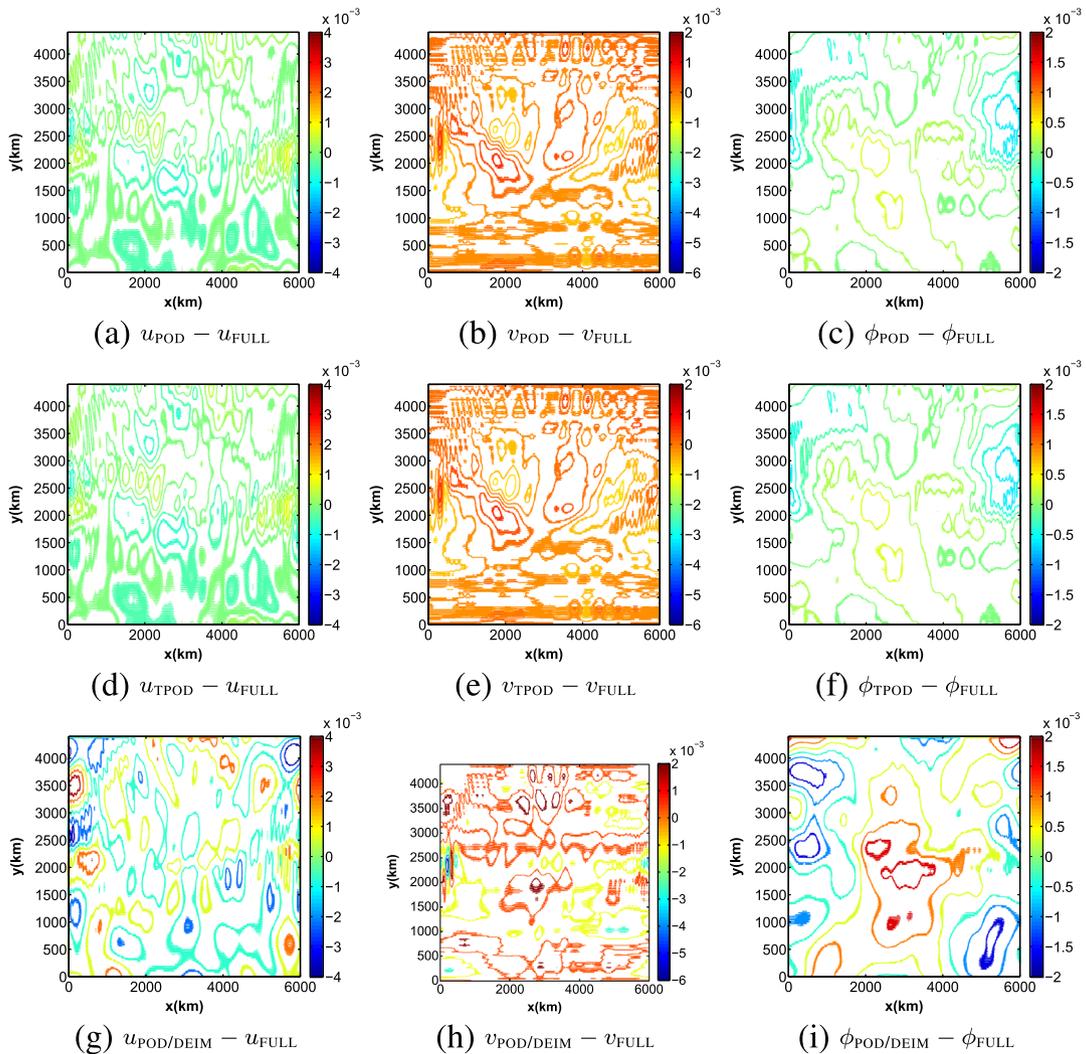


Figure 10. Absolute errors between POD, tensorial POD, and POD/DEIM solutions and the full trajectories at $t = 24$ h ($\Delta t = 960$ s) using initial conditions different than the ones used for ROM training. The number of DEIM points was taken 180.

costs in comparison with the POD model outcome when $n \geq 61 \times 45$ and $n \geq 101 \times 71$ for $m = 70$ and $m = 180$, respectively.

The new algorithm introduced in Section 3 relying on DEIM interpolation points delivers fast tensorial calculations required for computing the reduced Jacobian in the online stage and thus allowing POD/DEIM SWE scheme to have the fastest offline stage (Figure 9(b)). This gives a good advantage of ROM optimization based on discrete empirical interpolation method supposing that quality approximations of nonlinear terms and reduced Jacobians are delivered because during optimization, input data are different than the ones used to generate the DEIM interpolation points. DEIM was first employed by Baumann [88] to solve a reduced 4D Var data assimilation problem, and good results were obtained for a 1D Burgers model. Extensions to 2D models are still not available in the literature.

Next, we test the SWE ROMs for a set of initial conditions that differs from the ones employed for POD bases construction. For SWE experiments in the context of parametric variation, we refer to [66]. The experimental configuration consists in 376×276 space points, 91 time steps, and two integration time windows with $t_f = 3$ h and $t_f = 24$ h. The model parameters defined at the beginning of the section are kept the same. The [79] initial conditions are used to train the state

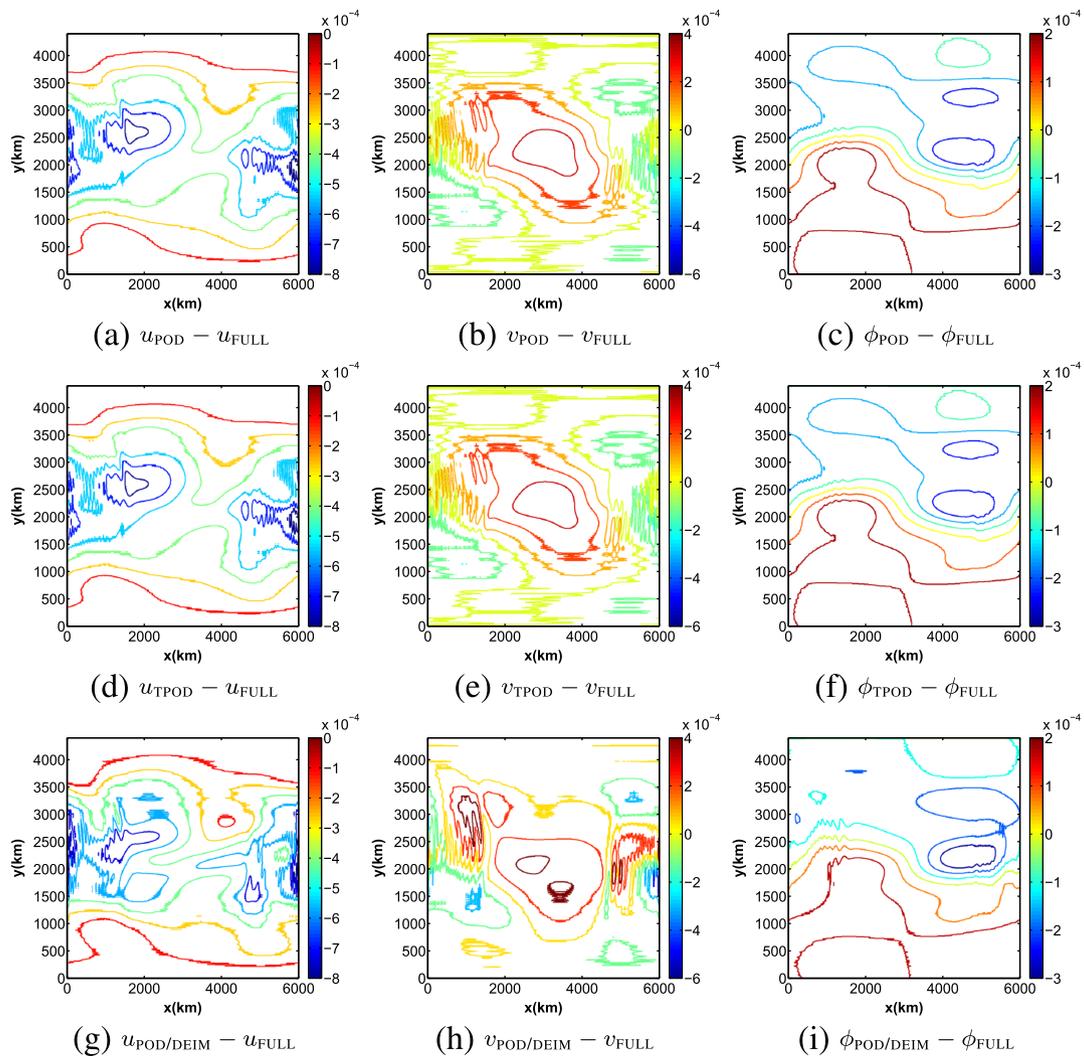


Figure 11. Absolute errors between POD, tensorial POD, and POD/DEIM solutions and the full trajectories at $t = 3$ h ($\Delta t = 120$ s) using initial conditions different than the ones used for ROM training. The number of DEIM points was taken 180.

variables and nonlinear terms-reduced bases. The DEIM points, tensors \mathbf{M}^i , and all POD/DEIM coefficients are also computed (for a comprehensive view of the offline calculations see table II). Online reduced order simulations are then performed using perturbed initial conditions. Constant noise in the form of state variables percentages is added as follows: 5% for zonal and meridional winds and 2% for geopotential field. Table VI describes the relative errors of the proposed ROMs solutions using 50 POD basis functions and 180 DEIM points. As we expected, the noise impacted more the 3-h simulations leading to an accuracy loss of 2–3 orders of magnitude.

Tables VII and VIII show the RMSE for final times together with the CPU times of the online stage of ROMs. More Newton iterations are required for convergence, and in terms of CPU time, it affects more the ROMs employing the largest vectors. It is the case of POD model, which is now $92\times$ and $94\times$ slower than tensorial POD model and $219\times$ and $273\times$ less efficient than POD/DEIM model for $t_f = 24$ h and $t_f = 3$ h, respectively.

Figures 10 and 11 present the grid point absolute errors of the ROMs solutions versus the full model outputs. For 24-h tests, we notice that POD/DEIM model requires more than 180 DEIM points to obtain the same level of accuracy as POD and TPOD solutions. For 3 h simulations, the reduced order solutions have a much closer error isolines patterns.

It is obvious that the accuracy of the online predictions depends on the relevance of the online problem training. To tackle this matter, several approaches have been proposed in the literature. A priori adaptive methods were promoted in an attempt to reduce the ROM sizes while preserving or even increasing accuracy leading to specific ROMs with respect to the input space [89–91], state space [82], and time domain [92]. When ROM solution is detected to be less accurate, a posteriori error estimation tools can enhance the ROM quality by either reverting to the high fidelity model [93] or refine the reduced basis vectors without incurring large-scale operations [94].

7. CONCLUSIONS

It is well known that in POD Galerkin, the cost of evaluating nonlinear terms during the online stage depends on the full space dimension, and this constitutes a major efficiency bottleneck. The present manuscript applies tensorial calculus techniques, which allow fast computations of POD reduced quadratic nonlinearities. We show that tensorial POD can be applied to all type of polynomial nonlinearities, and the resulting nonlinear terms have a complexity of $\mathcal{O}(k^{p+1})$ operations, where k is the dimension of POD subspace and p is the polynomial degree. Consequently, this approach eliminates the dependency on the full space dimension, while yielding the same reduced solution accuracy as POD. Despite being independent of the number of mesh points, tensorial POD is efficient only for quadratic nonlinear terms because for higher nonlinearities, POD proves to be less time consuming once the POD basis dimension k is increased.

The efficiency of tensorial POD is compared against that of POD and of POD/DEIM. We theoretically analyze the number of floating-point operations required as a function of polynomial degree p , the number of DOFs of the high fidelity model n , of POD modes k , and of DEIM interpolation points m . For quadratic nonlinearities and k between 10 and 50 modes, the tensorial POD needs 10 to $40\times$ fewer operations than the POD approach and 10 to $20\times$ more operations than the POD/DEIM with $m = 100$. But these performances are not translated into the same CPU time rates for solving the ROMs solutions because other more time-consuming calculations are needed.

Numerical experiments are carried out using a 2D ADI SWE FD model. ROMs were developed using each of the three ROM methods and Galerkin projection. The spectral analysis of snapshots matrices reveals that local versions of ROMs lead to more accurate results. Consequently, we focus on 3-h time integration windows. The tensorial POD SWE model becomes considerably faster than POD model when the dimension of the full model increases. For example, for 100,000 spatial points, the tensorial POD SWE model yields the same solutions accuracy as POD but is $76\times$ faster. Numerical experiments of POD/DEIM SWE scheme revealed a considerable reduction of the computational complexity. For a number of 70 DEIM points, POD/DEIM SWE model is $450\times$ faster than POD, but only $6\times$ faster than tensorial POD.

For implicit schemes depending only on quadratic nonlinearities, the tensorial POD represents a solid alternative to POD/DEIM, where the implementation effort is considerably larger. This is due to the simpler nonlinearities treatment utilized by tensorial POD surrogate models while POD/DEIM reduced models require additional nonlinearities snapshots computations and their corresponding POD basis calculations. However, for cubic and higher-order nonlinearities, tensorial POD loses its ability to deliver fast calculations and the POD/DEIM approach should be employed.

We also propose a new DEIM-based algorithm that allows fast computations of the tensors needed by reduced Jacobians calculations in the online stage. The resulting offline POD/DEIM stage is the fastest among the ones considered here, even if additional SVD decompositions and low-rank terms are computed. This is an important advantage in optimization problems based on POD/DEIM surrogates where the reduced order bases need to be updated multiple times.

Ongoing work by the authors focuses on reduced order constrained optimization. The current research represents an important step toward developing tensorial POD and POD/DEIM 4D variational data assimilation systems, which are not available in the literature for complex models.

APPENDIX

Here, we describe the applications of POD, tensorial POD, and POD/DEIM for treatment of the polynomial quadratic nonlinearity $N(\mathbf{x}) = x^2$ in the framework of reduced order modeling. We assume a Galerkin projection for constructing the surrogate models with $U \in \mathbb{R}^{n \times k}$ denoting the POD basis. The dimension of the full space is n , whereas k represents the number of POD basis functions. We assume a POD expansion of the state $\mathbf{x} \approx U\tilde{\mathbf{x}}$ and obtain the reduced order quadratic term $\tilde{N}(\tilde{\mathbf{x}}) \approx N(\mathbf{x})$ for the proposed approaches.

POD

$$\tilde{N}(\tilde{\mathbf{x}}) = \underbrace{U^T}_{k \times n} \underbrace{(U\tilde{\mathbf{x}})^2}_{n \times 1}, \quad \tilde{N}(\tilde{\mathbf{x}}) \in \mathbb{R}^k, \quad (\text{A.1})$$

where vector powers are taken componentwise.

Tensorial POD

$$\tilde{N}(\tilde{\mathbf{x}}) = [\tilde{N}_i]_{i=1,\dots,k} \in \mathbb{R}^k; \quad \tilde{N}_i = \sum_{j=1}^k \sum_{l=1}^k T_{i,j,l} \tilde{x}_j \tilde{x}_l, \quad (\text{A.2})$$

where the rank-three tensor T is defined as

$$T = (T_{i,j,l})_{i,j,l=1,\dots,k} \in \mathbb{R}^{k \times k \times k}, \quad T_{i,j,l} = \sum_{r=1}^n U_{r,i} U_{r,j} U_{r,l}.$$

POD/DEIM

$$\tilde{N}(\tilde{\mathbf{x}}) \approx \underbrace{U^T V (P^T V)^{-1}}_{k \times m} \underbrace{(P^T U \tilde{\mathbf{x}})^2}_{m \times 1}, \quad (\text{A.3})$$

where m is the number of interpolation points, $V \in \mathbb{R}^{n \times m}$ contains the first m POD basis modes of the nonlinear term, whereas $P \in \mathbb{R}^{n \times m}$ is the DEIM interpolation selection matrix computed by the greedy algorithm described in [32, s.3.1].

ACKNOWLEDGEMENTS

The work of Dr. Răzvan Stefanescu and Prof. Adrian Sandu was supported by the NSF CCF-1218454, AFOSR FA9550-12-1-0293-DEF, AFOSR 12-2640-06 and by the Computational Science Laboratory at

Virginia Tech. Prof. I.M. Navon acknowledges the support of NSF grant ATM-0931198. Răzvan Ștefănescu would like to thank Dr. Bernd R. Noack for his valuable suggestions on the current research topic that partially inspired the present manuscript. The contribution of two anonymous reviewers to improving and clarifying the presentation of our manuscript is acknowledged.

REFERENCES

1. Antoulas AC. *Approximation of Large-Scale Dynamical Systems*, Advances in Design and Control. SIAM: Philadelphia, 2005.
2. Moore BC. Principal component analysis in linear systems: controllability, observability, and model reduction. *IEEE Transactions on Automatic Control* 1981; **26**(1):17–32.
3. Antoulas AC. Approximation of large-scale dynamical systems. *Society for Industrial and Applied Mathematics* 2009; **6**:376–377.
4. Sorensen DC, Antoulas AC. The Sylvester equation and approximate balanced reduction. *Linear Algebra and its Applications* 2002; **351-352**(0):671–700.
5. Mullis CT, Roberts RA. Synthesis of minimum roundoff noise fixed point digital filters. *IEEE Transactions on Circuits and Systems* 1976; **CAS-23**:551–562.
6. Freund RW. Model reduction methods based on Krylov subspaces. *Acta Numerica* 2003; **12**:267–319.
7. Feldmann P, Freund RW. Efficient linear circuit analysis by Pade approximation via the Lanczos process. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 1995; **14**:639–649.
8. Grimme EJ. Krylov projection methods for model reduction. *Ph.D. Thesis*, University of Illinois at Urbana-Champaign, 1997.
9. Baker M, Mingori D, Goggins P. Approximate subspace iteration for constructing internally balanced reduced-order models of unsteady aerodynamic systems. *AIAA Meeting Papers on Disc* 1996:1070–1085. A9626908, AIAA Paper 96-1441.
10. Hodel AS. Least squares approximate solution of the Lyapunov equation. *Proceedings of the 30th IEEE Conference on Decision and Control, IEEE Publications*, Piscataway, NJ, 1991; 1619–1624.
11. Jaimoukha I, Kasenally E. Krylov subspace methods for solving large Lyapunov equations. *SIAM Journal of Numerical Analysis* 1994; **31**(1):227–251.
12. Gudmundsson T, Laub A. Approximate solution of large sparse Lyapunov equations. *IEEE Transactions on Automatic Control* 1994; **39**(5):1110–1114.
13. Willcox K, Peraire J. Balanced model reduction via the proper orthogonal decomposition. *AIAA Journal* 2002; **40**(11):2323–2330.
14. Gragg WB, Lindquist A. On the partial realization problem. *Linear Algebra and Its Applications, Special Issue on Linear Systems and Control* 1983; **50**:277–319.
15. Benner P, Sokolov VI. Partial realization of descriptor systems. *Systems & Control Letters* 2006; **55**(11):929–938.
16. Gragg WB. The Padé table and its relation to certain algorithms of numerical analysis. *SIAM Review* 1972; **14**:1–62.
17. Gallivan K, Grimme E, Dooren P Van. Padé approximation of large-scale dynamic systems with Lanczos methods. *Proceedings of the 33rd IEEE Conference on Decision and Control*, vol. 1, Lake Buena Vista, FL, December 1994; 443–448.
18. Gutknecht MH. The Lanczos process and Padé approximation. In *Proceedings of the Cornelius Lanczos International Centenary Conference*, Brown JD, et al. (eds). SIAM: Philadelphia, 1994; 61–75.
19. Van Dooren P. *The Lanczos algorithm and Padé approximations*, Short course, benelux meeting on systems and control, 1995.
20. Bultheel A, De Moor B. Rational approximation in linear systems and control. *Journal of Computational and Applied Mathematics* 2000; **121**:355–378.
21. Benner P, Breiten T. Two-sided moment matching methods for nonlinear model reduction. *Technical Report MPIMD/12-12*, Max Planck Institute Magdeburg Preprint: Magdeburg, Germany, 2012.
22. Karhunen K. *Zur spektraltheorie stochastischer prozesse*, vol. 37. Annales Academiae Scientiarum Fennicae, 1946.
23. Loève MM. *Probability Theory*. Van Nostrand: Princeton, NJ, 1955.
24. Hotelling H. Analysis of a complex of statistical variables with principal components. *Journal of Educational Psychology* 1933; **24**:417–441.
25. Lorenz EN. Empirical orthogonal functions and statistical weather prediction. *Technical Report*, Massachusetts Institute of Technology, Dept. of Meteorology: Cambridge, MA, 1956.
26. Sirovich L. Turbulence and the dynamics of coherent structures. I. Coherent structures. *Quarterly of Applied Mathematics* 1987; **45**(3):561–571.
27. Sirovich L. Turbulence and the dynamics of coherent structures. II. Symmetries and transformations. *Quarterly of Applied Mathematics* 1987; **45**(3):573–582.
28. Sirovich L. Turbulence and the dynamics of coherent structures. III. Dynamics and scaling. *Quarterly of Applied Mathematics* 1987; **45**(3):583–590.
29. Barrault M, Maday Y, Nguyen NC, Patera AT. An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathématique* 2004; **339**(9):667–672.
30. Chaturantabut S. Dimension Reduction for Unsteady Nonlinear Partial Differential Equations via Empirical Interpolation Methods. *Technical Report TR09-38, CAAM*, Rice University: Houston, TX, 2008.

31. Chaturantabut S, Sorensen DC. A state space error estimate for POD-DEIM nonlinear model reduction. *SIAM Journal on Numerical Analysis* 2012; **50**(1):46–63.
32. Chaturantabut S, Sorensen DC. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing* 2010; **32**(5):2737–2764.
33. Nguyen NC, Patera AT, Peraire J. A ‘best points’ interpolation method for efficient approximation of parametrized function. *International Journal for Numerical Methods in Engineering* 2008; **73**:521–543.
34. Astrid P, Weiland S, Willcox K, Backx T. Missing point estimation in models described by proper orthogonal decomposition. *IEEE Transactions on Automatic Control* 2008; **53**(10):2237–2251.
35. Carlberg K, Bou-Mosleh C, Farhat C. Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations. *International Journal for Numerical Methods in Engineering* 2011; **86**(2):155–181.
36. Carlberg K, Tuminaro R, Boggsz P. Efficient structure-preserving model reduction for nonlinear mechanical systems with application to structural dynamics, preprint, Sandia National Laboratories, Livermore, CA 94551, USA, 2012.
37. Everson R, Sirovich L. Karhunen–Loeve procedure for gappy data. *Journal of the Optical Society of America A* 1995; **12**:1657–64.
38. Grepl MA, Patera AT. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *ESAIM: Mathematical Modelling and Numerical Analysis* 2005; **39**(01):157–181.
39. Patera AT, Rozza G. Reduced basis approximation and a posteriori error estimation for parametrized partial differential equations, MIT, 2007.
40. Rozza G, Huynh DBP, Patera AT. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Archives of Computational Methods in Engineering* 2008; **15**(3):229–275.
41. Dihlmann M, Haasdonk B. Certified PDE-constrained parameter optimization using reduced basis surrogate models for evolution problems. *Submitted to the Journal of Computational Optimization and Applications* 2013.
42. Rowley CW, Mezic I, Bagheri S, Schlatter, Henningson DS. Spectral analysis of nonlinear flows. *Journal of Fluid Mechanics* 2009; **641**:115–127.
43. Schmid PJ. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics* 2010; **656**:5–28.
44. Tissot G, Cordier L, Benard N, Noack BR. Dynamic mode decomposition of PIV measurements for cylinder wake flow in turbulent regime. *Proceedings of the 8th International Symposium on Turbulent and Shear Flow Phenomena, TSFP-8*, 2013.
45. Rewieński M, White J. A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices. In *Proceedings of the 2001 IEEE/ACM International Conference on Computer-Aided Design, ICCAD '01*. IEEE Press: Piscataway, NJ, USA, 2001; 252–257.
46. Benner P, Gugercin S, Willcox K. A survey of model reduction methods for parametric systems. *Technical Report MPIMD/13-14*, Max Planck Institute Magdeburg Preprint: Magdeburg, Germany, 2013.
47. Kunisch K, Volkwein S. Control of the Burgers equation by a reduced-order approach using proper orthogonal decomposition. *Journal of Optimization Theory and Applications* 1999; **102**(2):345–371.
48. Kunisch K, Volkwein S, Xie L. HJB-POD-based feedback design for the optimal control of evolution problems. *SIAM Journal on Applied Dynamical Systems* 2004; **3**(4):701–722.
49. Belzen F, Weiland S. A tensor decomposition approach to data compression and approximation of ND systems. *Multidimensional Systems and Signal Processing* 2012; **23**(1-2):209–236.
50. Rowley CW, Colonius T, Murray RM. Model reduction for compressible flows using POD and Galerkin projection. *Physica D: Nonlinear Phenomena Nonlinear Phenomena* 2004; **189**(1-2):115–129.
51. Kunisch K, Volkwein S. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM Journal on Numerical Analysis* 2002; **40**(2):492–515.
52. Rowley CW. Model reduction for fluids, using balanced proper orthogonal decomposition. *International Journal of Bifurcation and Chaos (IJBC)* 2005; **15**(3):997–1013.
53. Bui-Thanh T, Damodaran M, Willcox K. Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition. *AIAA Journal* 2004; **42**(8):1505–1516.
54. Noack BR, Schlegel M, Morzynski M, Tadmor G. System reduction strategy for galerkin models of fluid flows. *International Journal for Numerical Methods in Fluids* 2010; **63**(2):231–248.
55. San O, Iliescu T. Proper orthogonal decomposition closure models for fluid flows: Burgers equation. *Technical Report arXiv:1308.3276 [physics.flu-dyn]*, Cornell University: Ithaca, NY, 2013.
56. Noack BR, Afanasiev K, Morzynski M, Tadmor G, Thiele F. A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *Journal of Fluid Mechanics* 2003; **497**:335–363.
57. Barbič J, James DL. Real-time subspace integration for St. Venant-Kirchhoff deformable models. In *ACM SIGGRAPH 2005 Papers, SIGGRAPH '05*. ACM: New York, NY, USA, 2005; 982–990.
58. Maday Y, Nguyen NC, Patera AT, Pau GSH. A general multipurpose interpolation procedure: the magic points. *Communications on Pure and Applied Analysis* 2009; **8**(1):383–404.
59. Grepl MA, Maday Y, Nguyen NC, Patera AT. Efficient reduced-basis treatment nonaffine and nonlinear partial differential equations. *Modélisation Mathématique et Analyse Numérique* 2007; **41**(3):575–605.
60. Wirtz D, Sorensen DC, Haasdonk B. A-posteriori error estimation for DEIM reduced nonlinear dynamical systems, SRC SimTech Preprint Series, 2012.

61. Kellems AR, Chaturantabut S, Sorensen DC, Cox SJ. Morphologically accurate reduced order modeling of spiking neurons. *Journal of Computational Neuroscience* 2010; **28**:477–494.
62. Hochman A, Bond BN, White JK. A stabilized discrete empirical interpolation method for model reduction of electrical thermal and microelectromechanical systems. *Design Automation Conference (DAC), 48th ACM/EDAC/IEEE* 2011:540–545.
63. Aanonsen TO. Empirical interpolation with application to reduced basis approximations. *Ph.D. Thesis*, Department of Mathematical Sciences, Norwegian University of Science and Technology, 2009.
64. Chaturantabut S, Sorensen DC. Application of POD and DEIM on dimension reduction of non-linear miscible viscous fingering in porous media. *Mathematical and Computer Modelling of Dynamical Systems* 2011; **17**(4): 337–353.
65. Suwartadi E. Gradient-based methods for production optimization of oil reservoirs. *Ph.D. Thesis*, Department of Engineering Cybernetics, Norwegian University of Science and Technology, 2012.
66. Stefanescu R, Navon IM. POD/DEIM nonlinear model order reduction of an ADI implicit shallow water equations model. *Journal of Computational Physics* 2013; **237**:95–114.
67. Hinze M, Kunkel M. Discrete empirical interpolation in POD model order reduction of drift-diffusion equations in electrical networks. *Scientific Computing in Electrical Engineering SCEE 2010 Mathematics in Industry* 2012; **16**(5):423–431.
68. Bo NguyenVan. Computational simulation of detonation waves and model reduction for reacting flows. *Ph.D. Thesis*, Singapore-MIT Alliance, National University of Singapore, 2011.
69. Xiao D, Fang F, Buchan AG, Pain CC, Navon IM, Du J, Hu G. Non-linear model reduction for the Navier-Stokes equations using residual DEIM method. *Journal of Computational Physics* 2014; **263**:1–18.
70. Gustafsson B. An alternating direction implicit method for solving the shallow water equations. *Journal of Computational Physics* 1971; **7**:239–254.
71. Fairweather G, Navon IM. A linear ADI method for the shallow water equations. *Journal of Computational Physics* 1980; **37**:1–18.
72. Navon IM, DeVilliers R. GUSTAF: a Quasi-Newton nonlinear ADI FORTRAN IV program for solving the shallow-water equations with augmented Lagrangians. *Computers and Geosciences* 1986; **12**(2):151–173.
73. Saad Y. Sparsekit: a basic tool kit for sparse matrix computations. *Technical Report*, Computer Science Department, University of Minnesota: Minneapolis, MN, 1994.
74. Barrett R, Berry M, Chan TF, Demmel J, Donato J, Dongarra J, Eijkhout V, Pozo R, Romine C, der Vorst H, Van. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods* (2nd edn). SIAM: Philadelphia, PA, 1994.
75. Kelley CT. *Iterative Methods for Linear and Nonlinear Equations*, Frontiers in Applied Mathematics. SIAM: Philadelphia, PA, 1995.
76. Saad Y. *Iterative Methods for Sparse Linear Systems* (2nd edn). Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2003.
77. Tonn T. Reduced-basis method (RBM) for non-affine elliptic parametrized PDEs, (PhD), Ulm University, 2012.
78. Zhou YB. Model reduction for nonlinear dynamical systems with parametric uncertainties, (M.S), Massachusetts Institute of Technology, Dept. of Aeronautics and Astronautics, 2012.
79. Grammelvedt A. A survey of finite difference schemes for the primitive equations for a barotropic fluid. *Monthly Weather Review* 1969; **97**(5):384–404.
80. Carlberg K, Farhat C, Cortial J, Amsallem D. The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows. *Journal of Computational Physics* 2013; **242**:623–647.
81. Rapún ML, Vega JM. Reduced order models based on local POD plus Galerkin projection. *Journal of Computational Physics* 2010; **229**(8):3046–3063.
82. Peherstorfer B, Butnaru D, Willcox K, Bungartz HJ. Localized discrete empirical interpolation method: MIT Aerospace Computational Design Laboratory, 2013.
83. Amsallem D, Zahr MJ, Farhat C. Nonlinear model order reduction based on local reduced-order bases. *International Journal for Numerical Methods in Engineering* 2012; **92**(10):891–916.
84. Lloyd S. Least squares quantization in PCM. *IEEE Transactions on Information Theory* 1957; **28**:129–137.
85. MacQueen J. Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* 1967; **1**:281–297.
86. Steinhaus H. Sur la division des corps matériels en parties. *Bulletin of the Polish Academy of Sciences* 1956; **4**(12):801–804.
87. Kaiser E, Noack BR, Cordier L, Spohn A, Segond M, Abel M, Daviller G, Niven RK. Cluster-based reduced-order modelling of a mixing layer. *Technical Report arXiv:1309.0524 [physics.flu-dyn]*, Cornell University: Ithaca, NY, 2013.
88. Baumann MM. Nonlinear model order reduction using POD/DEIM for optimal control of Burgers equation. *Master's Thesis*, Netherlands, 2013.
89. Amsallem D, Farhat C. An interpolation method for adapting reduced-order models and application to aeroelasticity. *AIAA Journal* 2008; **46**(7):1803–1813.
90. Eftang JL, Patera AT, Rønquist EM. An “hp” certified reduced basis method for parametrized elliptic partial differential equations. *SIAM Journal on Scientific Computing* 2010; **32**(6):3170–3200.

91. Drohmann M, Haasdonk B, Ohlberger M. Adaptive reduced basis methods for nonlinear convection diffusion equations. In *Finite Volumes for Complex Applications VI Problems & perspectives*, vol. 4, June 6–10, 2011, Fořt J, Fürst J, Halama J, Herbin R, Hubert F (eds)., Springer Proceedings in Mathematics. Springer Berlin Heidelberg, Prague, 2011; 369–377.
92. Dihlmann M, Drohmann M, Haasdonk B. Model reduction of parametrized evolution problems using the reduced basis method with adaptive time partitioning. *Technical Report*, Stuttgart Research Centre for Simulation Technology: Stuttgart, Germany, 2011.
93. Arian E, Fahl M, Sachs EW. Trust-region proper orthogonal decomposition for flow control. *ICASE: Technical Report 2000-25*: Hampton, VA, 2000.
94. Carlberg K. Adaptive h -refinement for reduced-order models, Computing Research Repository, abs/1404.0442, 2014.