Computer Science Technical Report
CSTR-19/2015
June 16, 2015

R. Ştefănescu, A. Sandu

"Efficient approximation of sparse Jacobians for time-implicit reduced order models"

Computational Science Laboratory
Computer Science Department
Virginia Polytechnic Institute and State University
Blacksburg, VA 24060
Phone: (540)-231-2193
Fax: (540)-231-6075
Email: sandu@cs.vt.edu
Web: http://csl.cs.vt.edu

Innovative Computational Solutions

Efficient approximation of sparse Jacobians for time-implicit reduced order models

Răzvan Ştefănescu [*1] and Adrian Sandu [†1]

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

Abstract

This paper introduces a sparse matrix discrete interpolation method to effectively compute matrix approximations in the reduced order modeling framework. The sparse algorithm developed herein relies on the discrete empirical interpolation method and uses only samples of the nonzero entries of the matrix series. The proposed approach can approximate very large matrices, unlike the current matrix discrete empirical interpolation method which is limited by its large computational memory requirements. The empirical interpolation indexes obtained by the sparse algorithm slightly differ from the ones computed by the matrix discrete empirical interpolation method as a consequence of the singular vectors round-off errors introduced by the economy or full singular value decomposition (SVD) algorithms when applied to the full matrix snapshots. When appropriately padded with zeros the economy SVD factorization of the nonzero elements of the snapshots matrix is a valid economy SVD for the full snapshots matrix. Numerical experiments are performed with the 1D Burgers and 2D Shallow Water Equations test problems where the quadratic reduced nonlinearities are computed via tensorial calculus. The sparse matrix approximation strategy is compared against five existing methods for computing reduced Jacobians: a) matrix discrete empirical interpolation method, b) discrete empirical interpolation method, c) tensorial calculus, d) full Jacobian projection onto the reduced basis subspace, and e) directional derivatives of the model along the reduced basis functions. The sparse matrix method outperforms all other algorithms. The use of traditional matrix discrete empirical interpolation method is not possible for very large instances due to its excessive memory requirements.

Keywords: POD; DEIM; implicit reduced-order models; shallow water equations; finite difference;

1 Introduction

Modeling and simulation of multi-scale complex physical phenomena has become an indispensable tool across a wide range of disciplines. This usually translates into large-scale systems of coupled

[*]rstefane@vt.edu

[†]sandu@cs.vt.edu

partial differential equations, ordinary differential equations or differential algebraic equations which often bear an extremely large computational cost and demand excessive storage resources due to the large-scale, nonlinear nature of high-fidelity models. Since many problems arising in practice are stiff an implicit time integrator is applied to advance the solution and to keep the errors in the results bounded. A sequence of high dimensional linear systems are solved iteratively at each time step when a Newton-like method is employed to obtain the solution of the corresponding system of nonlinear algebraic equations. As a result, the computational complexity of individual simulations can become prohibitive even when high-performance computing resources are available.

Not surprisingly, a lot of attention has been paid to reducing the costs of the complex system solutions 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.

Balanced truncation [55, 3, 79, 56] and moment matching [28, 27, 35] have been proving successful in developing reduced order models in the case of linear models. Unfortunately balanced truncation does not extend easily for high-order systems, and several grammians approximations were developed leading to methods such as approximate subspace iteration [5], least squares approximation [40], Krylov subspace methods [43, 36] and balanced Proper Orthogonal Decomposition [91]. Among moment matching methods we mention partial realization [32, 9], Padé approximation [31, 29, 38, 88] and rational approximation [12].

Currently input-independent highly accurate reduced models can be utilized to successfully reproduce the solutions of high-fidelity linear models. However 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 in [8] for nonlinear model order reduction but future investigations are required.

Proper Orthogonal Decomposition (POD) [45, 51, 41, 52] is the most prevalent basis selection method for nonlinear problems. Construction of low relevant manifolds is also the scope of reduced basis method [6, 34, 62, 71, 22, 49] and dynamic mode decomposition [70, 75, 85, 10]. Data analysis using POD and method of snapshots [76, 77, 78] is conducted to extract basis functions, from experimental data or detailed simulations of high-dimensional systems, for subsequent use in Galerkin projections that yield low dimensional dynamical models. Unfortunately the POD Galerkin approach has a major efficiency bottleneck since 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) [6] and its discrete variant DEIM [16, 18, 17] and best points interpolation method [59]. Recently the interpolation selection procedure in DEIM is formulated using a QR factorization with column pivoting [23]. Missing point estimation [4] and Gauss-Newton with approximated tensors [13, 15] methods are relying upon the gappy POD technique [25] and were developed for the same reason.

In the case of implicit POD Galerkin reduced order models solved via Newton based methods, during the on-line stage the Jacobian of the nonlinear term has also a computational cost that depends on the full-order dimension. More precisely, at each iteration, the full Jacobian is evaluated using the reduced order solution and then projected onto the POD manifold to obtain the reduced Jacobian required by Newton solver. One can slightly decrease the computational load of the POD Galerkin method by approximating the reduced Jacobians using the directional derivatives of Newton residuals in the directions of POD basis functions [89]. For polynomial nonlinearities of order p, tensorial calculus transfers several calculations from on-line to off-line stage and proposes

reduced Jacobians computations with a complexity of order of $\mathcal{O}(k^{p+1})$, where k is the dimension of reduced manifold. Such method was applied to obtain implicit reduced order Shallow Water Equations models not only for forward simulation purposes [81] but also for deriving a reduced order optimization framework [82].

Recently the use of interpolation methods relying on greedy algorithms became attractive for calculus of reduced order nonlinear terms derivatives. Chaturantabut [17] proposed a sampling strategy centered on the trajectory of the nonlinear terms in order to approximate the reduced Jacobians. An extension for nonlinear problems that do not have componentwise dependence on the state has been introduced in [93]. More accurate methods directly sample entries of the discrete Jacobians in addition to the nonlinear function. For example, based on EIM, Tonn (2011) [86] developed Multi-Component Empirical Interpolation Method for deriving affine approximations for continuous vector valued functions and Wirtz et al. [92] introduced matrix DEIM (MDEIM) approach to approximate the Jacobian of a nonlinear function to obtain aposteriori error estimates of DEIM reduced nonlinear dynamical system. In the context of the finite element method, an unassembled variant of DEIM was developed [2], [21], [84] and it can be used for approximation of sparse Jacobian matrices arising from element-wise assembly.

This paper introduces the sparse matrix discrete empirical interpolation method (SMDEIM) to construct fast and accurate approximation for sparse parametric matrices such as time dependent Jacobians. SMDEIM is a sparse variant of MDEIM approximation method and relies on the greedy algorithm introduced in [16] for computing approximations of the nonlinear functions. The proposed sparse algorithm utilizes samples of the nonzero entries of the matrix series and the output discrete interpolation indexes slightly differ from the ones obtained using full matrix snapshots.The differences are a consequence of the singular vectors round-off errors introduced by the economy or full SVD algorithms during the factorization of the full matrix snapshots. We proved that the economy SVD of the SMDEIM snapshots matrix when appropriately padded with zeros is a valid thin SVD for the MDEIM snapshots matrix. Now in contrast with MDEIM method, we apply the DEIM algorithm directly to the dense singular vectors and not their extended variant padded with zeros. Since DEIM algorithm selects only interpolation indexes corresponding to non empty rows, the output is similar but less computational demanding. The computational complexity of SMDEIM depends on the number of nonzero elements of the parametric matrices which is typically $\mathcal{O}(n)$, in contrast to MDEIM where the snapshots contain n^2 elements. As an application we integrate the SMDEIM approach with the reduced order modeling framework to deliver fast time implicit surrogate models.

The corresponding reduced order model is compared against the ROM versions obtained via MDEIM and other four type of approaches already existing in the literature for 1D Burgers and 2D Swallow Water Equations models. All the surrogate models employ tensorial calculus discussed in [81] to approximate the reduced nonlinearities so the proposed ROMs differ only in the way they compute the reduced derivatives. The on-line stages of MDEIM and sparse version reduced order models have the same computational complexities. For the off-line stage, the sparse version masively decreases the computational cost required by MDEIM reduced order model thus making it practically attractive. For a small number of DEIM indexes, the MDEIM reduced Jacobians and their sparse variant are as accurate as ones obtained by Galerkin projections and tensorial calculus.

The paper is organized as follows. Section 2 describes the greedy algorithm introduced in [16, 17] for nonlinear functions approximations and extensions for their Jacobians computations. Section 3 introduces the novel SMDEIM methodology. Section 4 presents the reduced order modeling framework focusing on proper orthogonal decomposition method while integrating both MDEIM and SMDEIM approaches for reduced Jacobian computations purposes. Other strategies for com-

puting reduced order derivatives are also described. Section 5 discusses the discrete 1D Burgers and 2D Swallow Water Equations models as well as the results of numerical experiments of the aforementioned reduced order models. Conclusions are drawn in Section 6.

2 Greedy algorithms for Jacobian approximations

An important question in many applications is the following: given a matrix A find an approximation that satisfies certain properties. For example, one may be interested in finding a reliable approximation of A by a matrix of lower rank. The singular value decomposition (SVD) is known to provide the best such approximation for any given fixed rank. In our case we are particular concerned for specific approximation matrices whose structures can be exploit in the framework of reduced order modeling for efficient on-line computation of reduced Jacobian matrices. Unfortunately a SVD matrix approximation only is not able to provide the computational complexity reduction expected by an implicit reduced order model for reduced derivatives calculations. However in combination with a greedy technique the desired matrix structure is obtained and bellow we describe two methodologies already existing in the literature. The new sparse matrix DEIM approach developed in this work seeks to overcome the deficiencies of the currently available approximations.

2.1 Discrete Empirical Interpolation Method for approximation of nonlinear functions

DEIM is a discrete variation of the Empirical Interpolation method proposed by Barrault et al. [6] which provides an efficient way to approximate nonlinear vector valued functions. The application was suggested and analyzed by Chaturantabut and Sorensen in [16, 17, 18].

Let $\mathbf{F} : D \to \mathbb{R}^n,\ D \subset \mathbb{R}^n$ be a nonlinear function. If $V = \{\mathbf{v}_l\}_{l=1}^m,\ \mathbf{v}_l \in \mathbb{R}^n$, is a linearly independent set, for $m \leq n$, then for $\tau \in D$, the DEIM approximation of order m for $\mathbf{F}(\tau)$ in the space spanned by $\{\mathbf{v}_l\}_{l=1}^m$ is given by

$$\mathbf{F}(\tau) \approx V\mathbf{c}(\tau),\ V \in \mathbb{R}^{n\times m},\ \mathbf{c}(\tau) \in \mathbb{R}^m. \tag{1}$$

The basis V can be constructed effectively by applying the POD method on the nonlinear snapshots $\mathbf{F}(\tau^{t_i}),\ \tau^{t_i} \in D$ (τ may be a function defined from $[0,T] \to D$, and τ^{t_i} is the value of τ evaluated at t_i), $i = 1,..,n_s,\ n_s > 0$. Next, interpolation is used to determine the coefficient vector $\mathbf{c}(\tau)$ by selecting m rows $\rho_1,..,\rho_m,\ \rho_i \in \mathbb{N}^*$, of the overdetermined linear system (1) to form a $m-$by$-m$ linear system

$$P^T V\,\mathbf{c}(\tau) = P^T\,\mathbf{F}(\tau),$$

where $P = [\mathbf{e}_{\rho_1},..,\mathbf{e}_{\rho_m}] \in \mathbb{R}^{n\times m},\ \mathbf{e}_{\rho_i} = [0,..0,\underbrace{1}_{\rho_i},0,..,0]^T \in \mathbb{R}^n$. The DEIM approximation of $\mathbf{F}(\tau) \in \mathbb{R}^n$ becomes

$$\mathbf{F}(\tau) \approx V\,(P^TV)^{-1}\,P^T\,\mathbf{F}(\tau). \tag{2}$$

Now the only unknowns that need to be specified are the indexes $\rho_1, \rho_2, ..., \rho_m$ or the matrix P whose dimensions are $n \times m$. These are determined by Algorithm 1.

The DEIM procedure inductively constructs a set of indexes from the input POD basis $\{\mathbf{v}_l\}_{l=1}^m \subset \mathbb{R}^n$. Initially the algorithm searches for the largest value of the first POD basis $|\mathbf{v}_1|$ and the corresponding index represents the first DEIM interpolation index $\rho_1 \in \{1, 2, .., n\}$. The remaining interpolation indexes $\rho_l,\ l = 2, 3.., m$ are selected so that each of them corresponds to the entry of

Algorithm 1 Computation of DEIM Interpolation Indexes

INPUT: $\{\mathbf{v}_l\}_{l=1}^m \subset \mathbb{R}^n$ (linearly independent):
OUTPUT: $\vec{\rho} = [\rho_1, .., \rho_m] \in \mathbb{N}^m$

1: $[|\psi| \;\; \rho_1] = \max |\mathbf{v}_1|, \psi \in \mathbb{R}$ and ρ_1 is the component position of the largest absolute value of v_1, with the smallest index taken in case of a tie.
2: $V = [\mathbf{v}_1] \in \mathbb{R}^n,\; P = [\mathbf{e}_{\rho_1}] \in \mathbb{R}^n,\; \vec{\rho} = [\rho_1] \in \mathbb{N}$.
3: **for** $\ell = 2, .., m$ **do**
4: Solve $(P^T V)\mathbf{c} = P^T \mathbf{v}_\ell$ for $\mathbf{c} \in \mathbb{R}^{\ell-1};\; V, P \in \mathbb{R}^{n\times(\ell-1)}$.
5: $\mathbf{r} = \mathbf{v}_\ell - V\mathbf{c},\; \mathbf{r} \in \mathbb{R}^n$.
6: $[|\psi| \;\; \rho_\ell] = \max\{|\mathbf{r}|\}$.
7: $V \leftarrow [U \;\; \mathbf{u}_\ell],\; P \leftarrow [P \;\; \mathbf{e}_{\rho_\ell}],\; \vec{\rho} \leftarrow \begin{bmatrix} \vec{\rho} \\ \rho_\ell \end{bmatrix}$.
8: **end for**

the largest magnitude of $|\mathbf{r}|$. The vector $\mathbf{r}$ can be viewed as the residual or the error between the input basis $\mathbf{v}_l,\; l = 2, 3.., m$ and its approximation $V\mathbf{c}$ from interpolating the basis $\{\mathbf{v}_1, \mathbf{v}_2, .., \mathbf{v}_{l-1}\}$ at the indexes $\rho_1, \rho_2, .., \rho_{l-1}$. The linear independence of the input basis $\{\mathbf{v}_l\}_{l=1}^m$ guarantees that, in each iteration, $\mathbf{r}$ is a nonzero vector and the output indexes $\{\rho_i\}_{i=1}^m$ are not repeating [17].

An error bound for the DEIM approximation is provided in Chaturantabut and Sorensen [16, 18]. An example of DEIM approximation of a highly nonlinear function defined on a discrete 1D spatial domain can be found in [17], underlying the DEIM efficiency.

Based on the greedy algorithm detailed above we will describe three approaches for approximating the Jacobian of $\mathbf{F}$ denoted by $\mathbf{J_F}(\tau) \in \mathbb{R}^{n\times n}$. The first two techniques DEIM and matrix DEIM were introduced in [17] and [92] while the sparse matrix DEIM algorithm is introduced here for the first time. While DEIM utilizes function samples, the matrix DEIM and sparse matrix DEIM are directly sampling entries of the discrete operator, i.e. the Jacobian of $\mathbf{F}$.

2.2 Discrete Empirical Interpolation Method for approximating Jacobians of nonlinear functions

This method was suggested in [17] for computing Jacobians of nonlinear functions in the framework of reduced order modeling. It proposes a sampling strategy centered on the trajectory of the nonlinear functions and makes use of DEIM approximation formula (2). Extensions for nonlinear problems that do not have componentwise dependence on the state have also investigated in [17, 93]. The DEIM Jacobian approximation is given by

$$\mathbf{J_F}(\tau) \approx V\left(P^T V\right)^{-1} P^T \mathbf{J_F}(\tau), \quad \tau = \{\tau^1, .., \tau^{n_s}\}, \quad \mathbf{J_F}(\tau^i) \in \mathbb{R}^{n\times n}, i = 1, .., n_s, \tag{3}$$

where V is constructed by extracting the left singular vectors of nonlinear snapshots matrix $\mathbf{F}(\tau^i), i = 1, .., n_s,\; n_s > 0$, while matrix P is the output of Algorithm 1.

2.3 Matrix Discrete Empirical Interpolation Method

This method applies the greedy technique described in subsection 2.1 in a different manner than DEIM approach discussed in the previous subsection. Thus instead of using snapshots of the nonlinear function $\mathbf{F}$, Jacobian snapshots written as vectors feed the Algorithm 1 providing a direct approximation of $\mathbf{J_F}(\tau)$. It was introduced by Wirtz et al. [92] to develop an efficient a-posteriori

error estimation for POD-DEIM reduced nonlinear dynamical systems. In particular the matrix DEIM approach was employed for an efficient off-line/on-line approximation of logarithmic Lipschitz constants of linear functions. A similar idea named "Multi-Component EIM" has been formulated in Tonn [86] to derive affine approximations for continuous vector valued functions.

First, we define the transformation $A \to T[A]$ which maps the entries of a matrix $A \in \mathbb{R}^{n\times n}$ column-wise into the vector $T[A] \in \mathbb{R}^{n^2\times 1}$. Next, we compute the economy or thin SVD [87] of the matrix of snapshots $\left[T[\mathbf{J_F}(\tau^i)]\right]_{i=1,..,n_s} \in \mathbb{R}^{n^2\times n_s},\ \tau^i \in D,\ i = 1,..,n_s,\ n_s > 0$ and the left singular vectors $V_J \in \mathbb{R}^{n^2\times n_s}$ are given by

$$\left[\, T[\mathbf{J_F}(\tau^1)], \ldots, T[\mathbf{J_F}(\tau^{n_s})] \,\right] = V_J\ \Sigma_J\ W_J^T, \tag{4}$$

where $\Sigma_J \in \mathbb{R}^{n_s\times n_s}$ is a diagonal matrix containing the singular values of $\left[T[\mathbf{J_F}(\tau^i)]\right]_{i=1,..,n_s}$ and $W_J \in \mathbb{R}^{n_s\times n_s}$ gathers the right singular vectors of the matrix of snapshots.

Let $m \le n_s$ be the number of DEIM index points and $P_J \in \mathbb{R}^{n^2\times n_s}$ represents the index points transformation matrix obtained by application of DEIM Algorithm 1 to the left singular vector matrix V_J. Then the m^{th} order matrix DEIM approximation of $\mathbf{J_F}(\tau),\ \tau = \{\tau^1, .., \tau^{n_s}\}$ is

$$\mathbf{J_F}(\tau) \approx T^{-1}\left[\, V_m\ \left(P_m^T V_m\right)^{-1}\ P_m^T\, T[\mathbf{J_F}(\tau)] \,\right], \tag{5}$$

where $V_m = V_J(:, 1:m)$, $P_m = P_J(:, 1:m)$.

However for large values of n the SVD factorization calculation demands increased computational resources and the memory required to store the left singular vectors V_J increases substantially thus limiting the algorithm application. This is the case even if a sparse SVD is employed since the output singular vectors are generally not sparse.

3 Sparse Matrix Discrete Empirical Interpolation Method

The most general method to compute the SVD factorization uses two phases. Initially the MDEIM snapshots matrix $\left[T[\mathbf{J_F}(\tau^i)]\right]_{i=1,..,n_s} \in \mathbb{R}^{n^2\times n_s}$ is brought into a bidiagonal form and then the bidiagonal matrix is diagonalized [30, 87]. The first stage usually employs Golub-Kahan or Lawson-Hanson-Chan bidiagonalizations while in the second stage a variant of the QR or divide-and-conquer algorithms are applied to generate the diagonal form. The cumulated computational cost of the SVD applied to the MDEIM snapshots matrix in case $n_s \ll n^2$ is $O(n^2 \cdot n_s^2)$ and singular vectors of n^2 size need to be stored in the memory.

Since usually the MDEIM Jacobian snapshots matrix is sparse we can apply a fill reducing ordering [19, 20] before the QR factorization to minimize the number of non-zeros in R or use a profile reduction ordering of R [39]. Other sparse singular value decomposition approaches relying on blocked algorithms have been proposed in [64]. The Lanczos subspace iteration based algorithms implemented in SVDPACK [Berry 1992] and PROPACK [Larsen 1998] are probably the most succesfully aproaches for finding the sparse SVD. While significantly decreasing the factorization computational cost, the sparse SVD methodologies still require to store singular vectors of size n^2. Moreover in case a thin SVD is applied round-off errors usually spoil the sparsity structure of the singular vectors.

To mitigate this drawback we propose a sparse version of matrix DEIM algorithm which relies on the fact that typically the Jacobians of large-scale time dependent problems have few nonzero entries and preserve their structure in time. We will build the snapshots matrix containing only the nonzero elements of the Jacobian matrices thus significantly decreasing the computation cost of the thin SVD

factorization and memory requirement for saving the corresponding singular vectors. Owning to the Jacobian structure we will prove that the thin SVD of the SMDEIM snapshots when appropriately padded with zeros is a valid thin SVD for the MDEIM snapshots $\left[T[\mathbf{J_F}(\tau^i)]\right]_{i=1,..,n_s} \in \mathbb{R}^{n^2\times n_s}$. The algorithm can be also adapted to accommodate Jacobian matrices with structures that vary in time.

For the moment we assume that Jacobian snapshots $\mathbf{J_F}(\tau^{t_i})$ have the same sparsity pattern for all $\tau^{t_i} \in D$, $i = 1, .., n_s$ as it is the case for the majority of time dependent problems. Moreover we consider that only r entries of $\mathbf{J_F}(\tau^{t_i}), \quad i = 1, .., n_s$, out of n^2 are different from zero. This suggests that MDEIM snapshots matrix $\left[T[\mathbf{J_F}(\tau^i)]\right]_{i=1,..,n_s}$ has a rank less than or equal to $\min(r, n_s)$ and contains $n^2 - r$ rows with zero elements only. Usually the number of nonzero entries r is larger than the number of snapshots n_s. Now we can appropriately select the r nonzero rows of MDEIM snapshots matrix by using a truncated identity matrix $\bar{P} \in \mathbb{R}^{r\times n^2}$

$$\bar{P}\left[T[\mathbf{J_F}(\tau^i)]\right]_{i=1,..,n_s} \in \mathbb{R}^{r\times n_s}. \tag{6}$$

Consequently $\bar{P} = \bar{p}_{jl}, j = 1, .., r; l = 1, .., n^2$ represents a linear transformation $\bar{P} : \mathbb{R}^{n^2} \to \mathbb{R}^r$ that contains only one element other than zero per each line, i.e. $\bar{p}_{jo_j} = 1, \ j = 1, ..r$, where $o_j \in \{1, 2, .., n^2\}$ are the r locations corresponding to the nonzero elements of $T[\mathbf{J_F}(\tau^i)] \in \mathbb{R}^{n^2}$. Here and in the subsequent lemma we abuse of notations of $\bar{P}$ and $\bar{P}^T$ to denote both the linear transformations and their matrices representations, respectively.

Lemma 3.1. *The transpose of matrix $\bar{P}$, $\bar{P}^T : \mathbb{R}^r \to \mathbb{R}^{n^2}$ represents the inverse transformation of $\bar{P}$.*

Proof. First let us denote by $G = \bar{P}^T\bar{P}$, $G = g_{jl}, \ j, l = 1, .., n^2 \in \mathbb{R}^{n^2\times n^2}$,

$$g_{jl} = \sum_{k=1}^{r} \bar{p}_{kj}\bar{p}_{kl}, \quad j, l = 1, .., n^2.$$

Since $\bar{p}_{jo_j} = 1, \ j = 1, ..r$, then the only components of G other then zero are $g_{o_jo_j} = 1, \ j = 1, .., r$. Now the elements of $T[\mathbf{J_F}(\tau^i)]$ different then zero are located at $o_j, \ j = 1, .., r$ positions and thus

$$\bar{P}^T P T[\mathbf{J_F}(\tau^i)] = T[\mathbf{J_F}(\tau^i)], \forall i = 1, .., n_s.$$

It is easy to show that $\bar{P} \cdot \bar{P}^T \in \mathbb{R}^{r\times r}$ is the identify matrix which completes the proof. □

Next we compute the thin singular value decomposition of the dense snapshots matrix (6) denoted by SMDEIM snapshots matrix and the left singular vectors $V_{J_{nz}} \in \mathbb{R}^{r\times n_s}$ are given by

$$\bar{P}\left[T[\mathbf{J_F}(\tau^i)]\right]_{i=1,..,n_s} = V_{J_{nz}} \ \Sigma_{J_{nz}} \ W_{J_{nz}}^T, \tag{7}$$

where $\Sigma_{J_{nz}} \in \mathbb{R}^{n_s\times n_s}$ is the singular values diagonal matrix and $W_{J_{nz}} \in \mathbb{R}^{n_s\times n_s}$ collects the right singular vectors of the dense matrix.

Lemma 3.2. *$\bar{P}^T V_{J_{nz}} \ \Sigma_{J_{nz}} \ W_{J_{nz}}^T$ is a thin SVD representation of the MDEIM snaspshots matrix $\left[T[\mathbf{J_F}(\tau^i)]\right]_{i=1,..,n_s}$.*

Proof. If we multiply the left hand side of the equation (7) by $\bar{P}^T$ and make use of the Lemma 3.1 we immediatly obtain

$$\left[T[\mathbf{J_F}(\tau^i)]\right]_{i=1,..,n_s} = \bar{P}^T V_{J_{nz}} \ \Sigma_{J_{nz}} \ W_{J_{nz}}^T. \tag{8}$$

Now $\Sigma_{J_{nz}}$ is a diagonal matrix with positive real entries and $W_{J_{nz}}$ describes an orthogonal matrix since both of the matrices are obtained from the singular value decomposition in (7). According to [87] we only need to demostrate that $\bar{P}^T V_{J_{nz}}$ has orthonormal columns in order to complete the proof. This is obvious since for all $j, l = 1, .., n_s$

$$\langle v_j, v_l \rangle_2 = \langle \bar{v}_j, \bar{v}_l \rangle_2 = \begin{cases} 1, & j = l \\ 0, & j \neq l, \end{cases}$$

where $\langle \cdot, \cdot \rangle_2$ is the Euclidian product and v_j and $\bar{v}_j, j = 1, .., n_s$ are the columns of matrices $\bar{P}^T V_{J_{nz}}$ and $V_{J_{nz}}$, respectively. □

Now instead of applying the DEIM algorithm 1 to the left singular vectors $\bar{P}^T V_{J_{nz}}$ of $\big[T[\mathbf{J_F}(\tau^i)]\big]_{i=1,..,n_s}$ as in the matrix DEIM approach we propose to use the left singular vectors $V_{J_{nz}}$ of $\bar{P}\big[T[\mathbf{J_F}(\tau^i)]\big]_{i=1,..,n_s}$ as input for the DEIM algorithm 1. The output DEIM indexes match perfectly in both situations, however in the latest case the computational load is decreased leading to the DEIM index points matrix $P_{J_{nz}} \in \mathbb{R}^{r \times n_s}$. By leting $m \leq n_s$ be the number of DEIM index points, we obtain the following m^{th} DEIM approximation

$$\bar{P}\big[T[\mathbf{J_F}(\tau^i)]\big] \approx V_{m_{nz}} \Big(P_{m_{nz}}^T V_{m_{nz}} \Big)^{-1} P_{m_{nz}}^T \bar{P}\big[T\big[\mathbf{J_F}(\tau^i)\big]\big], \forall i = 1, .., n_s,$$

where $V_{m_{nz}} = V_{J_{nz}}(:, 1:m),\ P_{m_{nz}} = P_{J_{nz}}(:, 1:m)$.

Now multiplying the left hand side of the above equation with $\bar{P}^T$ and using Lemma 3.1 and inverse transformation T^{-1} we obtain the m^{th} order sparse matrix DEIM approximation of $\mathbf{J_F}(\tau),\ \forall \tau = \{\tau^1, .., \tau^{n_s}\}$

$$\mathbf{J_F}(\tau) \approx T^{-1} \left[\bar{P}^T \left[V_{m_{nz}} \Big(P_{m_{nz}}^T V_{m_{nz}} \Big)^{-1} P_{m_{nz}}^T \bar{P} \Big[T\big[\mathbf{J_F}(\tau)\big] \Big] \right] \right], \tag{9}$$

The following result based on [17, Lemma 3.2] provides an error bound for the SMDEIM approximation.

Lemma 3.3. *Let $\mathbf{F} \in \mathbb{R}^n$ be a sparse column-wise vector representation of a matrix with $r > 0$ nonzero entries located at $o_j,\ j = 1, 2, .., r,\ o_j \in \{1, 2, .., n\}$. Let $\bar{P} \in \mathbb{R}^{r \times n}$ be a truncated identity matrix with the nonzero elements $\bar{p}_{j,o_j} = 1,\ j = 1, .., r$ such that $\bar{P}\,\mathbf{F} \in \mathbb{R}^r$ comprises only the non-zero elements of $\mathbf{F}$. Let $V_{m_{nz}} = \{\bar{v}_l\}_{l=1}^m \in \mathbb{R}^{r \times m},\ m > 0$ be a collection of orthonormal vectors and*

$$\hat{\mathbf{F}} = \bar{P}^T V_{m_{nz}} \left(P_{m_{nz}}^T V_{m_{nz}} \right)^{-1} P_{m_{nz}}^T \bar{P}\,\mathbf{F}, \tag{10}$$

be the sparse matrix DEIM approximation of order $m \ll n$ for $\mathbf{F}$ with $P_{m_{nz}} = [\mathbf{e}_{\rho_1}, .., \mathbf{e}_{\rho_m}] \in \mathbb{R}^{r \times m}$ being the output of DEIM Algorithm 1 having as input the basis $V_{m_{nz}}$. Then the following result holds

$$\|\mathbf{F} - \hat{\mathbf{F}}\|_2 \leq \| \Big(P_{m_{nz}}^T V_{m_{nz}} \Big)^{-1} \|_2 \, \| \Big(I - V_{m_{nz}} V_{m_{nz}}^T \Big) \bar{P}\,\mathbf{F} \|_2, \tag{11}$$

where $\| \cdot \|_2$ is the appropriate vector or matrix $2-$ norm.

Proof. Since $P_{m_{nz}} \in \mathbb{R}^{r \times m}$ is the output of the DEIM algorithm 1 with the input basis $V_{m_{nz}}$, then DEIM approximation of $\bar{P}\,\mathbf{F}$ in the space spanned by $\{\bar{v}_l\}_{l=1}^m$ is:

$$\widehat{\bar{P}\,\mathbf{F}} = V_{m_{nz}} \left(P_{m_{nz}}^T V_{m_{nz}} \right)^{-1} P_{m_{nz}}^T \bar{P}\,\mathbf{F}. \tag{12}$$

According to [17, Lemma 3.2] we have the following bound

$$\|\bar{P}\,\mathbf{F} - \widehat{\bar{P}\,\mathbf{F}}\|_2 \leq \|\Big(P_{m_{nz}}^T V_{m_{nz}}\Big)^{-1}\|_2\,\|\Big(I - V_{m_{nz}} V_{m_{nz}}^T\Big)\bar{P}\,\mathbf{F}\|_2 \tag{13}$$

From lemma 3.1 we obtain that

$$\mathbf{F} = \bar{P}^T\bar{P}\,\mathbf{F}, \tag{14}$$

and by applying equations (10),(12) we get

$$\|\mathbf{F} - \hat{\mathbf{F}}\|_2 = \|\bar{P}^T\bar{P}\,\mathbf{F} - \bar{P}^T\widehat{\bar{P}\,\mathbf{F}}\|_2 \leq \|\bar{P}^T\|_2\,\|\bar{P}\,\mathbf{F} - \widehat{\bar{P}\,\mathbf{F}}\|_2. \tag{15}$$

From the definition of matrix $\bar{P}$ we have $\|\bar{P}^T\|_2 = 1$ which completes the proof. □

The above approximation can be applied to any algebraic structure that can be reduced to a vector. The danger of directly applying the full or thin SVD to factorize the MDEIM snapshots matrix consists in generating dense singular vectors due to the round-off errors which subsequently may lead to DEIM indexes pointing to zero element rows of the Jacobian. The newly proposed version of MDEIM avoids interpolating the Jacobian zeros even for large number of DEIM indexes providing a fast and efficient approximation of the Jacobian matrix with constant sparse structure.

Formula (9) can be adapted to approximate a Jacobian matrix with time variable sparse structure too. One possibility would be to identify distinctive paterns of the Jacobians and form separate snapshots matrices that lead to different transformations $\bar{P}$ and local in time singular vectors $V_{m_{nz}}$ and DEIM indexes $P_{m_{nz}}$. For large number of snapshots n_s, unsupervised learning tools such as biclustering methods [54] can be employed to identify the distinctive structures in data matrices. The other approach consists in selecting r as the largest number of nonzero locations available at one time step over the entire time interval and the corresponding sparsity structure will define a new transformation $\bar{P}$.

Next the proposed Jacobian approximation will be tested in the framework of reduced order modeling. Traditionally the reduced Galerkin nonlinearities and their derivatives computations are considered time consuming since they still depend on the dimension of the full space. The sparse matrix DEIM technique proposes an efficient off-line stage in comparison with the traditional MDEIM method while maintaining the same computational complexity in the on-line stage.

4 Reduced order modeling

Our plan is to integrate the proposed Jacobian matrix approximations in the reduced order modeling framework and generate faster off-line/on-line implicit reduced order models for large spatial configurations. We will consider the Proper Orthogonal Decomposition technique combined with tensorial calculus [81] as the main strategy for deriving the surrogates models and their reduced nonlinearities. For reduced Jacobian computations we propose six different techniques including MDEIM and its sparse variant, DEIM, tensorial, direct projection and directional derivative methods.

4.1 Proper Orthogonal Decomposition

Proper Orthogonal Decompositions has been used successfully in numerous applications such as compressible flow [68], computational fluid dynamics [48, 69, 91], and aerodynamics [11]. 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. [61] 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 [74] investigate several closure models for POD reduced order modeling of fluids 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 discrete dynamical system written in the residual form

$$\mathbf{r}^i(\mathbf{x}_{t_i}) = \mathbf{x}_{t_i} - \mathbf{x}_{t_{i-1}} - \Delta t\, \mathbf{F}(\mathbf{x}_{t_i}) = 0, \quad \text{for } i = 1, .., N_t, \quad N_t \in \mathbb{N}, \tag{16}$$

where $\mathbf{x}_{t_i} \in \mathbb{R}^n$ is the state and $\mathbf{r}^i : \mathbb{R}^n \to \mathbb{R}^n$ denotes the residual operator at time step t_i. Usually a Newton based approach is employed to solve (16). Once the discrete solution is obtained, we define the centering trajectory, shift mode, or mean field correction [60] $\bar{\mathbf{x}} = \frac{1}{N_t}\sum_{i=1}^{N_t} \mathbf{x}_{t_i}$. The method of POD consists in choosing a complete orthonormal basis $U = \{\mathbf{u}_i\},\ i = 1, .., k;\ k > 0;\ u_i \in \mathbb{R}^n;\ U \in \mathbb{R}^{n\times k}$ such that the mean square error between $\mathbf{x}_{t_i}$ and POD expansion

$$\mathbf{x}_{t_i}^{POD} = \bar{\mathbf{x}} + U\tilde{\mathbf{x}}_{t_i},\ \tilde{\mathbf{x}}_{t_i} \in \mathbb{R}^k \tag{17}$$

is minimized on average. The POD dimension $k \ll n$ is appropriately chosen to capture the dynamics of the flow as described by Algorithm 2.

Algorithm 2 POD basis construction

1: Calculate the mean $\bar{\mathbf{x}} = \frac{1}{N_t}\sum_{i=1}^{N_t} \mathbf{x}_{t_i}$.
2: Set up the correlation matrix $K = [k_{ij}]_{i,j=1,..,N_t}$ where $k_{ij} = \langle \mathbf{x}_{t_i} - \bar{\mathbf{x}}, \mathbf{x}_{t_j} - \bar{\mathbf{x}} \rangle_2$.
3: Compute the eigenvalues $\lambda_1 \geq \lambda_2 \geq ...\lambda_{N_t} \geq 0$ and the corresponding orthogonal eigenvectors $\mathbf{v}^1, \mathbf{v}^2, .., \mathbf{v}^{N_t} \in \mathbb{R}^{N_t}$ of K.
4: Set $\mathbf{u}_i = \sum_{j=1}^{N_t} \mathbf{v}_j^i (\mathbf{x}_{t_i} - \bar{\mathbf{x}})$, $i = 1, .., N_t$. Then, $\mathbf{u}_i \in \mathbb{R}^n,\ i = 1, .., N_t$ are normalized to obtain an orthonormal basis.
5: Define $I(m) = \left(\sum_{i=1}^{m} \lambda_i\right) / \left(\sum_{i=1}^{N_t} \lambda_i\right)$ and choose k such that $k = \min\{I(m) : I(m) \geq \gamma\}$ where $0 \leq \gamma \leq 1$ is the percentage of total informations captured by the reduced space $\text{span}\{\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_k\}$. Usually γ is taken 0.99.

Singular value decomposition is another choice for POD basis construction and is less affected by numerical errors than the eigenvalue decomposition. Moreover, the SVD-based POD basis construction is more computational efficient since it decomposes the snapshots matrix whose condition number is the square root of the correlation matrix K used in Algorithm 2. The snapshots matrix should also contain the difference quotients of the state variables in order to achieve optimal pointwise in time rates of convergence with respect to the number of POD basis functions [47, 42].

The Galerkin projection of the full model equations onto the space spanned by the POD basis elements leads to the reduced order model

$$\tilde{\mathbf{r}}^i(\tilde{\mathbf{x}}_{t_i}) = \tilde{\mathbf{x}}_{t_i} - \tilde{\mathbf{x}}_{t_{i-1}} - \Delta t\, \tilde{\mathbf{F}}(\tilde{\mathbf{x}}_{t_i}),\ i = 1, .., N_t, \tag{18}$$

where $\tilde{\mathbf{x}}_{t_i} \in \mathbb{R}^k$ and $\tilde{\mathbf{F}} : \mathbb{R}^k \to \mathbb{R}^k, \tilde{\mathbf{F}}(\tilde{\mathbf{x}}_{t_i}) = U^T\mathbf{F}(\bar{\mathbf{x}} + U\tilde{\mathbf{x}}_{t_i})$ are the reduced state and nonlinear function, respectively and $\tilde{\mathbf{r}}^i : \mathbb{R}^k \to \mathbb{R}^k$ denotes the reduced residual operator.

The majority of the current reduced discrete schemes available in the literature are usually explicit or at most semi-implicit in time thus avoiding computing reduced Jacobians. Consequently

most of the attempts to increase the efficiency of reduced order models were focused on providing efficient off-line/on-line decoupled approximations for the nonlinear terms only. In this research we shift the attention toward generating efficient off-line/on-line approximations of the reduced Jacobians.

4.2 Reduced Jacobian computations

In this subsection we integrate the Jacobian approximations MDEIM and its sparse version discussed in Sections 2.3 and 3 into reduced order modeling framework to enable fast and accurate estimations of the reduced Jacobians of the nonlinear terms. Here is the first time when these strategies are employed for construction of implicit reduced order schemes. Wirtz et al. [92] introduced MDEIM to develop an efficient off-line/on-line approximation of logarithmic Lipschitz constants of linear functions and delivered a-posteriori error estimates of DEIM reduced nonlinear dynamical system. Along with the dense and sparse matrix DEIM approximation methods we describe the current available techniques used to compute the reduced Jacobians. We will begin with the exact formulations and then continue with the approximation techniques including the novel MDEIM expressions.

Direct projection method The simplest approach for calculating the reduced Jacobian $\mathbf{J}_{\tilde{\mathbf{F}}}(\tilde{\mathbf{x}}_{t_i})$ is to follow the analytical route. The derivatives of the function $\tilde{\mathbf{F}}(\tilde{\mathbf{x}}_{t_i})$ with respect to the $\tilde{\mathbf{x}}_{t_i}$ are computed using the chain rule and we get

$$\mathbf{J}_{\tilde{\mathbf{F}}}(\tilde{\mathbf{x}}_{t_i}) = U^T \mathbf{J}_{\mathbf{F}}(\bar{\mathbf{x}} + U\tilde{\mathbf{x}}_{t_i})\, U, \qquad \mathbf{J}_{\tilde{\mathbf{F}}}(\tilde{\mathbf{x}}_{t_i}) \in \mathbb{R}^{k\times k}, \tag{19}$$

where

$$\mathbf{J}_{\tilde{\mathbf{F}}}(\tilde{\mathbf{x}}_{t_i}) = \frac{\partial \tilde{\mathbf{F}}}{\partial \tilde{\mathbf{x}}_{t_i}}(\tilde{\mathbf{x}}_{t_i}), \qquad \mathbf{J}_{\mathbf{F}}(\bar{\mathbf{x}} + U\tilde{\mathbf{x}}_{t_i}) = \frac{\partial \mathbf{F}}{\partial \mathbf{x}_{t_i}}(\bar{\mathbf{x}} + U\tilde{\mathbf{x}}_{t_i})$$

There is no off-line cost for this strategy since all the computations are performed on-line. At every time step the full Jacobian $\mathbf{J}_{\mathbf{F}}(\bar{\mathbf{x}} + U\tilde{\mathbf{x}}_{t_i})$ is evaluated using the reduced solution and then projected to the reduced space. Suppose the complexity for evaluating the r nonzero elements of the full Jacobian is $\mathcal{O}(\alpha(r))$, where α is some function of r, then the on-line computational complexity of $\mathbf{J}_{\tilde{\mathbf{F}}}(\tilde{\mathbf{x}}_{t_i})$ is of order of $\mathcal{O}(\alpha(r)) + nk + rk + nk^2)$ in case the sparse structure of the Jacobian is expoited. Unfortunately this approach is extremely costly and for large number of mesh points, it leads to slower reduced order models in comparison with the high fidelity versions.

Tensorial method For $\mathbf{F}$ containing only polynomial nonlinearities, tensorial calculus can be applied to compute $\mathbf{J}_{\tilde{\mathbf{F}}}(\tilde{\mathbf{x}}_{t_i})$ and most of the required calculations can be translated to the off-line stage, making the on-line phase independent of n. To emphasize the tensorial procedure we assume that $\mathbf{F}$ presents only a quadratic nonlinearity, thus, at time t_i, $\mathbf{F} = \mathbf{x}_{t_i} \odot \mathbf{x}_{t_i}$, $\mathbf{x}_{t_i} \in \mathbb{R}^n$ and $\odot$ is the componentwise operator. The Jacobian of $\mathbf{F}$ calculated at $\mathbf{x}_{t_i}$ is a diagonal matrix

$$\mathbf{J}_{\mathbf{F}}(\mathbf{x}_{t_i}) = \operatorname{diag}\left\{2\mathbf{x}^1_{t_i}, 2\mathbf{x}^2_{t_i}, ..., 2\mathbf{x}^n_{t_i}\right\} \in \mathbb{R}^{n\times n},$$

where $\mathbf{x}^k_{t_i}$ represents the k component of the vector $\mathbf{x}_{t_i}$. Then according to (19) we get

$$\mathbf{J}_{\tilde{\mathbf{F}}}(\tilde{\mathbf{x}}_{t_i}) = 2\, U^T \operatorname{diag}\left\{\bar{\mathbf{x}} + U(1,:)\tilde{\mathbf{x}}_{t_i}, \bar{\mathbf{x}} + U(2,:)\tilde{\mathbf{x}}_{t_i}, ..., \bar{\mathbf{x}} + U(n,:)\tilde{\mathbf{x}}_{t_i}\right\} U,$$

and subsequently

$$\mathbf{J}_{\tilde{\mathbf{F}}}(\tilde{\mathbf{x}}_{t_i}) = T_1 + T_2^i, \tag{20}$$

where $T_1 = 2U^T\big(\underbrace{[\bar{\mathbf{x}} \quad \bar{\mathbf{x}} \quad \cdot\cdot \quad \bar{\mathbf{x}}]}_{k \text{ times}} \odot U\big) \in \mathbb{R}^{k\times k}$ and $T_2^i \in \mathbb{R}^{k\times k}$, $T_2^i(j,l) = \sum_{p=1}^{k} \tilde{\mathbf{x}}_{t_i}^p \cdot \big(g_{lp}^j + g_{pl}^j\big),\ j,l = 1,..,k$. Tensor $G \in \mathbb{R}^{k\times k\times k}$ is defined by

$$g_{lp}^j = U(s,j)\cdot U(s,l)\cdot U(s,p),\ j,l,p = 1,..,k \tag{21}$$

and $\tilde{\mathbf{x}}_{t_i}^p$ is the p component of $\tilde{\mathbf{x}}_{t_i}$. Now T_1 and G are computed off-line and the computational complexity is of order $\mathcal{O}(k^3 n)$. For the on-line stage T_2^i is required and its computational complexity is $\mathcal{O}(k^3)$. In the case of a polynomial nonlinearity of order p the computational complexity for calculating the reduced Jacobian using tensorial calculus in the on-line stage is $\mathcal{O}(k^{p+1})$ while the off-line components require $\mathcal{O}(k^{p+1})$ flops. We already applied this strategy to generate implicit reduced SWE models in [80, 81].

Directional derivatives method One can decrease the computational load of the direct projection method by approximating $\mathbf{J}_{\mathbf{F}}(\bar{\mathbf{x}} + U\tilde{\mathbf{x}}_{t_i})$ using the directional derivatives of $\mathbf{F}$ in the directions of POD basis functions $\mathbf{u_j} = \mathbf{U}(:,\mathbf{j})$, for $j = 1, 2.., k$

$$\frac{\partial \mathbf{F}}{\partial \mathbf{x}_{t_i}}(\bar{\mathbf{x}} + U\tilde{\mathbf{x}}_{t_i})\mathbf{u_j} = \begin{pmatrix} \nabla \mathbf{F}_1(\bar{\mathbf{x}} + U\tilde{\mathbf{x}}_{t_i}) \\ \nabla \mathbf{F}_2(\bar{\mathbf{x}} + U\tilde{\mathbf{x}}_{t_i}) \\ . \\ \nabla \mathbf{F}_n(\bar{\mathbf{x}} + U\tilde{\mathbf{x}}_{t_i}) \end{pmatrix} \mathbf{u_j} = \begin{pmatrix} \nabla_{\mathbf{u_j}} \mathbf{F}_1(\bar{\mathbf{x}} + U\tilde{\mathbf{x}}_{t_i}) \\ \nabla_{\mathbf{u_j}} \mathbf{F}_2(\bar{\mathbf{x}} + U\tilde{\mathbf{x}}_{t_i}) \\ . \\ \nabla_{\mathbf{u_j}} \mathbf{F}_n(\bar{\mathbf{x}} + U\tilde{\mathbf{x}}_{t_i}) \end{pmatrix}. \tag{22}$$

The vector valued function is written using its scalar components $\mathbf{F} = (\mathbf{F}_1, \mathbf{F}_2, \cdot\cdot, \mathbf{F}_n)^T$, $\mathbf{F}_l : \mathbb{R}^n \to \mathbb{R},\ l = 1,..n$ and their gradients $\nabla \mathbf{F}_l$ belong to $\mathbb{R}^{1\times n}$. By $\nabla_{\mathbf{u_j}} \mathbf{F}_l$ we denote the directional derivative of $\mathbf{F}_l$ in the direction of $\mathbf{u_j}$.

$$\nabla_{\mathbf{u_j}} \mathbf{F}_l \approx \frac{\mathbf{F}_l(\bar{\mathbf{x}} + U\tilde{\mathbf{x}}_{t_i} + h\mathbf{u_j}) - \mathbf{F_l}(\bar{\mathbf{x}} + \mathbf{U}\tilde{\mathbf{x}}_{\mathbf{t_i}})}{h}. \tag{23}$$

If the cost for evaluating the n scalar components of F is $\mathcal{O}(\alpha(n))$, the computational complexity of this method includes effort in the on-line stage only and is of order of $\mathcal{O}(k\alpha(n) + nk^2)$. The accuracy level depends on the values of h. Vermeulen and Heemink [89] linearized a high-order nonlinear model and their reduced model was obtained using $h = 0.01$. This strategy is a non-intrusive approach allowing for the reduced Jacobian computation by making use of only the high-fidelity function.

DEIM method Chaturantabut [16] noticed that the Jacobian of a vector valued function can be approximated using the POD/DEIM approximation of the function itself (3), i.e.

$$\mathbf{J}_{\tilde{\mathbf{F}}}(\tilde{\mathbf{x}}_{t_i}) \approx \underbrace{U^T V (P^T V)^{-1}}_{\text{precomputed:}k\times m} \underbrace{P^T \mathbf{J}_{\mathbf{F}}(\bar{\mathbf{x}} + U\tilde{\mathbf{x}}_{t_i})}_{m\times n} \underbrace{U}_{n\times k}. \tag{24}$$

Here m is the number of DEIM points and their locations given by P^T are obtained by applying the DEIM algorithm 1 with nonlinear term basis V as input. Typically the Jacobians of large-scale problems are sparse, and then the approximation (24) will be very efficient. Assuming an average of μ nonzero Jacobian elements per each row, the reduced derivatives calculations during the on-line stage require a computational complexity of order $\mathcal{O}(\mu k^2 m + mk^2 + \alpha(\mu m))$, where $\mathcal{O}(\alpha(\mu m))$

stands for the cost of evaluating the μm full Jacobian entries. More details on the sparse procedure are available in [17].

The off-line cost of computing DEIM reduced Jacobian arises from the singular value decomposition of the nonlinear term snapshots $\left(\mathcal{O}(n \cdot n_s^2)\right)$, DEIM algorithm for selecting the interpolation points $\left(\mathcal{O}(m^2 \cdot n + m^3)\right)$ [23] and matrix operations in (24) $\left(\mathcal{O}(m^3 + n \cdot m^2 + k^2 \cdot m)\right)$. No additional effort is needed in this stage in the case the DEIM method is employed to approximate the reduced nonlinear term too.

Matrix DEIM method By using MDEIM Jacobian approximation (5) inside of (19) we obtain the reduced Jacobian approximation

$$\mathbf{J}_{\tilde{\mathbf{F}}}(\tilde{\mathbf{x}}_{t_i}) \approx U^T \, T^{-1} \left[V_m \, \left(P_m^T V_m \right)^{-1} \, P_m^T \, T[\mathbf{J}_{\mathbf{F}}(\bar{\mathbf{x}} + U\tilde{\mathbf{x}}_{t_i})] \right] \, U. \tag{25}$$

From here, one can easily prove that

$$\mathbf{J}_{\tilde{\mathbf{F}}}(\tilde{\mathbf{x}}_{t_i}) \approx \tilde{T}^{-1} \left[\underbrace{C}_{k^2 \times n^2} \cdot \underbrace{V_m \, \left(P_m^T V_m \right)^{-1}}_{n^2 \times m} \cdot \underbrace{P_m^T \, T[\mathbf{J}_{\mathbf{F}}(\bar{\mathbf{x}} + U\tilde{\mathbf{x}}_{t_i})]}_{m \times 1} \right], \tag{26}$$

where transformation $B \to \tilde{T}[B]$ maps the entries of a matrix $B \in \mathbb{R}^{k \times k}$ column-wise into a vector of the size $\mathbb{R}^{k^2}$ and $\tilde{T}^{-1}$ is its inverse. Matrix C is defined bellow

$$C(i,:) = T[\mathbf{u}_j \mathbf{u}_l^T]^T, \quad i = 1, .., k^2, \tag{27}$$

with each i corresponding to a pair of indexes $(j, l),\ j, l = 1, .., k$. The transformation T is defined in Section 2.3.

Now the complexity of the off-line stage of MDEIM is dominated by the computation of the matrix C and its product with $V_m \, \left(P_m^T V_m \right)^{-1}$ which requires $\mathcal{O}(n^4 \cdot k^2 + n^2 \cdot m\dot{k}^2 + n^2 \cdot m^2 + m^3)$. Other costs arise from SVD calculation of the nonlinear term snapshots $\left(\mathcal{O}(n^2 \cdot n_s^2)\right)$ and DEIM algorithm for selecting the interpolation indexes $\left(\mathcal{O}(m^2 \cdot n^2 + m^3)\right)$. The on-line computational complexity of $\mathbf{J}_{\tilde{\mathbf{F}}}(\tilde{\mathbf{x}}_{t_i})$ is of $\mathcal{O}(k^2 \cdot m)$ plus the cost of evaluating m entries of $\mathbf{J}_{\mathbf{F}}(\bar{\mathbf{x}} + U\tilde{\mathbf{x}}_{t_i})$ that needs $\mathcal{O}(\alpha(m))$ flops.

Sparse Matrix DEIM method The sparse version of the MDEIM method was derived to alleviate the memory requirement of storing jacobian snapshots $\mathbf{J}_{\mathbf{F}}(\mathbf{x}_{t_i}),\ i = 1, .., n_s$ of $\mathbb{R}^{n^2}$ size. By applying SMDEIM approximation (9) inside equation (19), the computational complexity for calculating the reduced Jacobians off-line components will depend only on the number of nonzero entries of the high-fidelity Jacobian, dimension of POD basis and number of DEIM indexes,

$$\mathbf{J}_{\tilde{\mathbf{F}}}(\tilde{\mathbf{x}}_{t_i}) \approx U^T T^{-1} \left[\bar{P}^T \left[V_{m_{nz}} \left(P_{m_{nz}}^T V_{m_{nz}} \right)^{-1} P_{m_{nz}}^T \, \bar{P} \left[T \left[\mathbf{J}_{\mathbf{F}}(\bar{\mathbf{x}} + U\tilde{\mathbf{x}}_{t_i}) \right] \right] \right] \right] U. \tag{28}$$

Next, it follows that

$$\mathbf{J}_{\tilde{\mathbf{F}}}(\tilde{\mathbf{x}}_{t_i}) \approx \tilde{T}^{-1} \left[\underbrace{\tilde{C}}_{k^2 \times r} \underbrace{V_{m_{nz}} \left(P_{m_{nz}}^T V_{m_{nz}} \right)^{-1}}_{r \times m} \underbrace{P_{m_{nz}}^T \, \bar{P} \left[T \left[\mathbf{J}_{\mathbf{F}}(\bar{\mathbf{x}} + U\tilde{\mathbf{x}}_{t_i}) \right] \right]}_{m \times 1} \right], \tag{29}$$

where the rows of matrix $\tilde{C}$ are computed with the following formula

$$\tilde{C}(i,:) = \mathbf{u}_j(\mathbf{coef1}) \odot \mathbf{u}_l(\mathbf{coef2}), \quad i = 1,..,k^2. \tag{30}$$

Each i corresponds to a pair of indexes $(j,l),\ j,l = 1,..,k$ and vectors $\mathbf{coef1},\ \mathbf{coef2} \in \mathbb{R}^r$ store the $\mathbf{J_F}$ matrix column and row indexes where nonzero entries are found.

Now the cost for assemblying the matrix $\tilde{C}$ and its product with $V_{m_{nz}}\left(P_{m_{nz}}^T V_{m_{nz}}\right)^{-1}$ is of order of $\mathcal{O}(r \cdot k^2 + r \cdot m \cdot k^2 + r \cdot m^2 + m^3)$. In addition the off-line stage cost includes the computation of singular value decomposition of the dense nonlinear snapshots $\mathcal{O}(r \cdot n_s^2)$ and DEIM indexes via Algorithm1 – $\mathcal{O}(m^2 \cdot r + m^3)$. The on-line cost of $\mathbf{J}_{\tilde{\mathbf{F}}}(\tilde{\mathbf{x}}_{t_i})$ is the same as in the case of MDEIM approximation and counts $\mathcal{O}(k^2 \cdot m + \alpha(m))$ flops. Tables 1,2 resume the findings of this section.

		MDEIM	SMDEIM	DEIM
Off-line	SVD	$\mathcal{O}(n^2 \cdot n_s^2)$	$\mathcal{O}(r \cdot n_s^2)$	$\mathcal{O}(n \cdot n_s^2)$
	DEIM indexes	$\mathcal{O}(m^2 \cdot n^2 + m^3)$	$\mathcal{O}(m^2 \cdot r + m^3)$	$\mathcal{O}(m^2 \cdot n + m^3)$
	other	$\mathcal{O}(n^4 \cdot k^2 + n^2 \cdot m \cdot k^2 + n^2 \cdot m^2 + m^3)$	$\mathcal{O}(r \cdot k^2 + r \cdot m \cdot k^2 + r \cdot m^2 + m^3)$	$\mathcal{O}(m^3 + n \cdot m^2 + k^2 \cdot m)$
On-line		$\mathcal{O}(k^2 \cdot m + \alpha(m))$	$\mathcal{O}(k^2 \cdot m + \alpha(m))$	$\mathcal{O}(\mu k^2 m + m k^2 + \alpha(\mu m))$

Table 1: Computational complexities of the reduced Jacobians. n, n_s, r, μ, m and k denote the numbers of independent variables, snapshots of the Jacobian, nonzero entries of a Jacobian snapshot, average nonzeros entries of a Jacobian snapshot per row, DEIM indexes and size of the POD basis. By $\alpha(p)$ we mean the cost of evaluating p entries of the high-fidelity Jacobian linearized at $\bar{\mathbf{x}} + U\tilde{\mathbf{x}}_{t_i}$.

	tensorial	Direct proj.	Directional deriv.
Off-line	$\mathcal{O}(k^{p+1} \cdot n)$	–	–
On-line	$\mathcal{O}(k^{p+1})$	$\mathcal{O}(\alpha(r)) + nk + rk + nk^2)$	$\mathcal{O}(k\alpha(n) + nk^2)$

Table 2: Computational complexities of the reduced Jacobians. The results for tensorial method correspond to a p^{th} polinomial nonlinearity. By $\alpha(p)$ we denote the cost for evaluating p entries of the high-fidelity Jacobian at $\bar{\mathbf{x}} + U\tilde{\mathbf{x}}_{t_i}$ or function $\mathbf{F}$ for direct projection method or directional derivative approach, respectively.

The most expensive on-line stage is proposed by the direct projection technique which is an exact method. By transferring some of the calculations to the off-line stage, the other exact approach, tensorial method becomes competive against the DEIM based techniques but only for quadratic nonlinearities [81]. As an approximation method, one should expect that directional derivative approximation would be faster than the exact methods. This is not the case since its complexity depends on the number of space points n. This technique is preferred for situation when the partial derivatives of the function are difficult to compute analitically and only function evaluations are needed. The choice of h in (23) must be careful considered. Among all the proposed techniques, MDEIM and SMDEIM own the fastest on-line stage. DEIM method is much faster in the off-line stage but it is a price paid at the expense of the Jacobian accuracy. DEIM method for the Jacobian approximation guarantees accurate entrees only along the rows indicated by the DEIM points. This is not the case for the MDEIM and SMDEIM formulations which preserve the Jacobian accuracy

globally as DEIM method does for the function approximation. The newly introduced sparse version of the MDEIM technique now poses the properties required for large-scale simulations with an off-line cost depending only on the number of Jacobian nonzero elements, POD basis dimension and number of DEIM points.

5 Numerical Experiments

We consider two nonlinear test problems, the 1D Burgers and the 2D Shallow Water Equations, and first compare the accuracy of various greedy based Jacobian approximations described in Sections 2 and 3. Next, we analyze the performance of the novel reduced order models obtained by integrating MDEIM and SMDEIM into POD/ROM framework against the available techniques already existing in the literature and discussed in Section 4.2. Both the high-fidelity and the corresponding reduced order models make use of the same time discretization schemes thus avoiding additional errors inside the surrogate models solutions. In all of the experiments the nonlinear terms are computed using tensorial calculus.

5.1 One-dimensional Burgers' equation

5.1.1 Numerical scheme

Burgers' equation is a fundamental partial differential equation from fluid mechanics. It occurs in various areas of applied mathematics. For a given velocity u and viscosity coefficient μ, the model considered here has the following form

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \mu\frac{\partial^2 u}{\partial x^2}, \quad x \in [0, L],\ t \in (0, t_f]. \tag{31}$$

We assume Dirichlet homogeneous boundary conditions $u(0,t) = u(L,t) = 0,\ t \in (0, t_f]$ and as initial conditions we use a seventh degree polynomial depicted in Figure 1

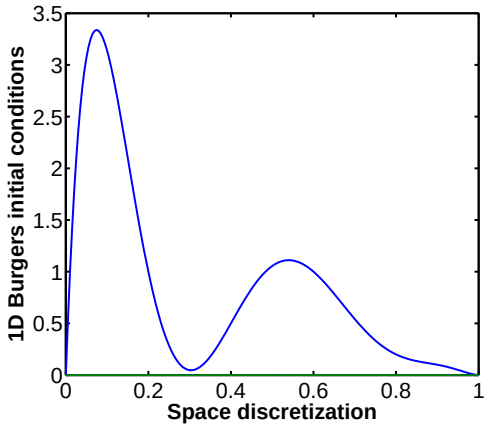


Figure 1: Seventh order polynomial used as initial conditions for 1D Burgers model.

Let us introduce a mesh of n equidistant space points on $[0, L]$, with $\Delta x = L/(n-1)$. For the time interval $[0, t_\mathrm{f}]$ we employ N_t equally distributed points with $\Delta t = t_\mathrm{f}/(N_t - 1)$. By defining the vector of unknown variables of dimension $n-2$ (we eliminate the known boundaries) with

$\boldsymbol{u}(t_N) \approx [u(x_i, t_N)]_{i=1,2,..,n-2} \in \mathbb{R}^{n-2},\ N = 1, 2, ..N_t$, the semi-discrete version of 1D Burgers model (31) is:

$$\mathbf{u}' = -\mathbf{u} \odot A_x \boldsymbol{u} + \mu A_{xx} \boldsymbol{u}, \tag{32}$$

where $\mathbf{u}'$ denotes the semi-discrete time derivative of $\mathbf{u}$. $A_x,\ A_{xx} \in \mathbb{R}^{(n-2)\times(n-2)}$ are the central difference first-order and second-order space derivatives operators which include also the boundary conditions.

The viscosity parameter is set to $\mu = 0.01$, the final time $t_{\mathrm{f}} = 2$ and $L = 1$. The backward Euler method is employed for time discretization and it is implemented in Matlab. The nonlinear algebraic systems are solved using Newton-Raphson method and the maximum number of Newton iterations allowed per each time step is set to 50. The solution is considered accurate enough when the euclidian norm of the residual is less than 10^{-10}.

5.1.2 Greedy based Jacobian approximation techniques

Here we discuss different aspects characterizing the newly introduced SMDEIM method and compare its properties including spectrum of snapshots matrix, locations of DEIM indexes and approximation accuracy against the ones proposed by MDEIM and DEIM methods using the high-fidelity framework.

In order to generate the greedy based Jacobian approximations we use 401 time snapshots, i.e. $N_t = 401$. Thus we have 401 model Jacobian (including the linear and non-linear terms derivatives) snapshots rearranged in vector format and 401 snapshots of the advection term of the 1D-Burgers model (32). For the SMDEIM method, the matrix of snapshots belongs to $\mathbb{R}^{[3(n-4)+4]\times 401}$ and counts only the nonzero entries of the MDEIM snapshots matrix of dimensions $\mathbb{R}^{(n-2)^2\times 401}$. The nonlinear term snapshots matrix contains $(n-2)\times 401$ dense elements. The numerical experiments for this subsection are performed using a mesh of $n = 201$ space points.

Figure 2 illustrates the singular values of the SMDEIM and MDEIM snapshots matrices which are very similar. As expected, the computational time for obtaining the SVD decompositions of the SMDEIM matrix is 4 times smaller than in the case of MDEIM matrix factorization.

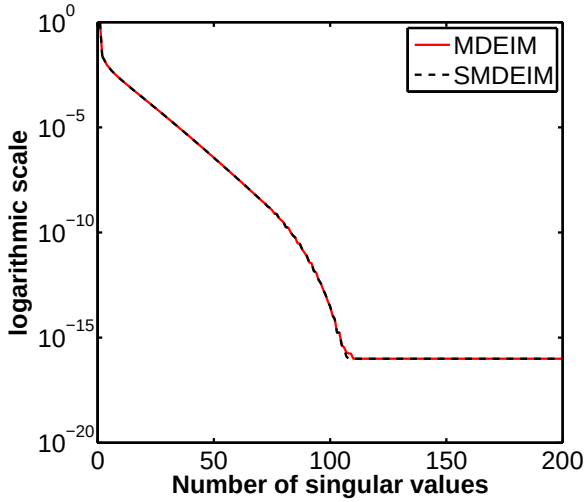
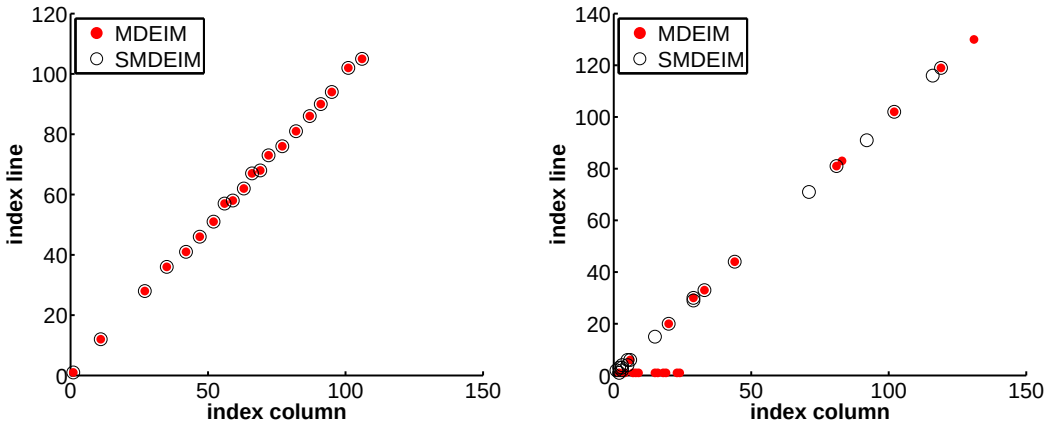


Figure 2: Singular values of MDEIM and SMDEIM snapshots matrices

Moreover, each singular vector of the MDEIM snapshot matrix has 989 nonzero entries while a Jacobian snapshot contains only 595 nonzero elements. This is a very well know behaviour, since

the singular vectors of a sparse matrix are usually denser. However, in our case the structure of the MDEIM snapshots is regular having entire null rows owing to the Jacobian pattern and the additional nonzero singular vectors artifacts arise from the round-off errors introduced by the matrix factorization. This can be noticed in Figure 3 where the interpolation indexes generated by algorithm 1 using MDEIM and SMDEIM singular vectors are depicted. Figure 3(a) shows a perfect match of the first 20 DEIM indexes that correspond to singular values ranging from 467.43 to 0.107. Figure 3(b) presents the DEIM indexes for the $100^{\text{th}} - 120^{\text{th}}$ singular values with ranges between $1.41e^{-11}$ to $4.65e^{-14}$. At this low magnitude DEIM indexes mismatches can be noticed. We remark the multiple indexes of the MDEIM singular vectors outside the diagonal band that point to zero entries of the Jacobian snapshots. The first mismatch occurs at the 84^{th} singular value where the level of energ y is $2.66e^{-8}$. However for most of the applications there is no need to select so many DEIM indexes including those corresponding to such small singular vectors, perhaps except simulating turbulence. For finding 20 pair of DEIM indexes, algorithm 1 was 24 times faster when using the SMDEIM singular vectors.



(a) 1D-Burgers - $1^{\text{st}} - 20^{\text{th}}$ interpolation indexes

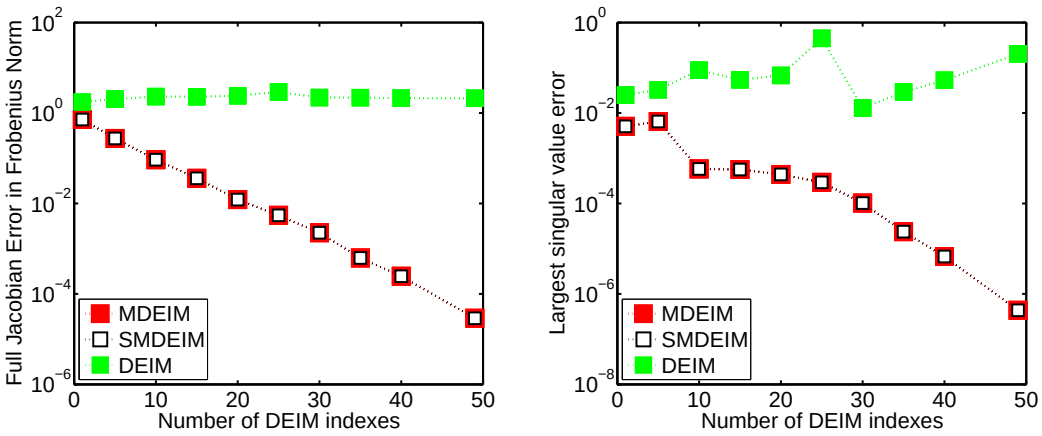
(b) 1D-Burgers - $100^{\text{th}} - 120^{\text{th}}$ interpolation indexes

Figure 3: Localization of DEIM indexes using MDEIM and SMDEIM singular vectors

Finally we compare the accuracy levels of DEIM (3), MDEIM (5) and SMDEIM (9) approximations of the 1D Burgers model Jacobian at initial time. The DEIM based Jacobian approximation (3) requires adding the derivatives of the linear terms while for the matrix DEIM approximations the linear and nonlinear partial derivatives are both included into the snapshots. Figure 4 depicts the error of the Jacobian approximations using the Frobenius norm (left panel) and the absolute value of the discrepancies in the largest singular value of the matrices approximations and its true representation (right panel). The MDEIM and SMDEIM Jacobian approximations accuracy is improved with the increase of the DEIM indexes. This is not the case for the DEIM approximations which preserves the accuracy only for the rows $P^T\mathbf{J_F}(\tau)$ selected by the interpolation matrix in (3). The SMDEIM Jacobian approximation quality is similar as the MDEIM proposal but it is more advantageous since it is obtained at much lower computational cost.

5.1.3 Performance of implicit reduced order models

In general, the implicit discrete problems obtained from discretization of nonlinear partial differential equations and ordinary differential equations are solved by employing some sort of a Newton



(a) 1D-Burgers - 201 space points indexes (b) 1D-Burgers - 201 space points

Figure 4: Full Jacobians errors at initial time - Frobenius norm - Largest SVD.

based technique. It requires some residuals computations and their space derivatives evaluations. This is also the case for reduced implicit discrete problems. While for reduced residual calculations we will apply the tensorial POD approach, for reduced Jacobians computations we will make use of six different techniques described in subsection 4.2 including the newly introduced SMDEIM method. For simplicity we decide to employ a reduced order expansion (17) that does not account for the mean.

We will compare the computational off-line/on-line costs as well as the accuracy of the proposed methods. As measures we propose the Frobenius norm of the errors between the reduced Jacobian and the solutions of the reduced order models. Details about the number of Newton iterations are presented for each method.

We derive the reduced order 1D Burgers model by employing a Galerkin projection. The constructed POD basis functions are the singular vectors of the state variable and nonlinear term snapshots matrices obtained from the numerical solution of the full - order implicit Euler 1D Burgers discrete model. Figure 5 shows the decay around the singular values of the solution u and the advection term $u\frac{\partial u}{\partial x}$ for 401 snapshots equally distributed in the interval $[0, 2]$ and 201 number of space points. This configuration is used for the majority of the experiments in this subsection.

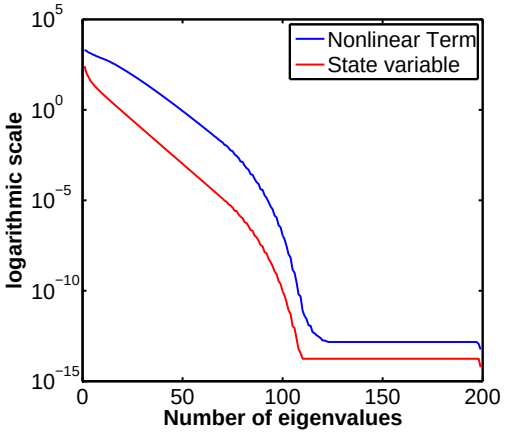
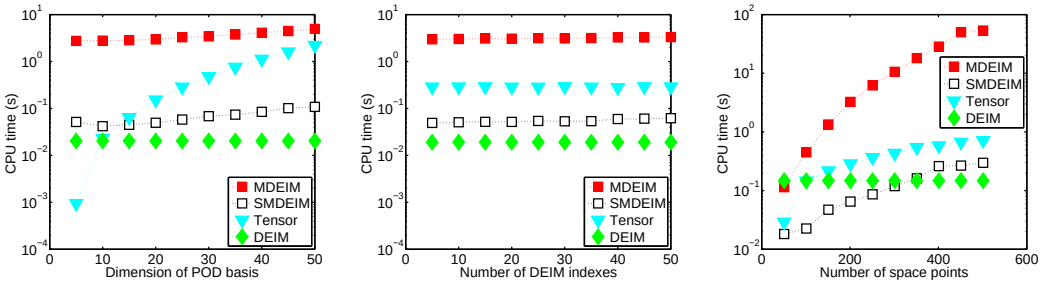


Figure 5: Spectrum properties of state variable and nonlinear term snapshots matrices.

Off-line computational performances We begin our comparison study by focusing on the off-line CPU costs of the reduced Jacobian methods. Only DEIM based techniques and tensorial method have off-line stages thus only their performances will be discussed here. Various POD basis dimensions k, number of interpolation indexes m, space points n and time steps N_t are considered.

First the SVD factorizations of the MDEIM, SMDEIM ad DEIM snapshots matrices are derived. Next the DEIM algorithm 1 computes the interpolation indexes for each set of singular vectors. Then matrices $U^TV(P^TV)^{-1}$, $C \cdot V_m \left(P_m^T V_m\right)^{-1}$ and $\tilde{C} \cdot V_{m_{nz}} \left(P_{m_{nz}}^T V_{m_{nz}}\right)^{-1}$ in (3), (5) and (9) are assembled along with tensor G (21) and the total computational costs of the above off-line operations are counted separately for each method and depicted in Figure 6. In panel (a), we set the number of selected interpolation indexes to 30 and notice that for a reduced basis with 50 modes the SMDEIM is 45.7 times faster than MDEIM. The DEIM method has the smallest off-line computational cost while tensorial method CPU time tends to increase exponentially with the growth of POD basis dimension. For the numerical experiment depicted in Figure 6(b) we choose 25 POD basis functions. The off-line CPU times of the greedy based techniques are slightly increased with the growth of the number of the interpolation indexes. For $m = 50$, we remark that SMDEIM is 54.2 times more rapid than MDEIM. Once we increase the number of mesh points we observe in Figure 6(c) that the sparse version becomes much faster. For 25 POD basis functions and 30 interpolation indexes the sparse MDEIM is approximately 200 times faster than the MDEIM method for 501 space points and 1001 time steps.

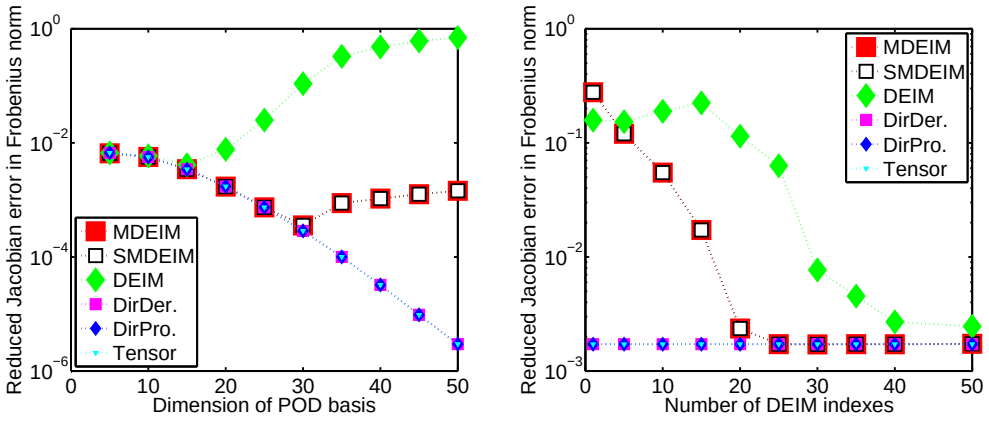


(a) $m = 30$; (b) $k = 25$; (c) $m = 30$, $k = 25$;

Figure 6: Off-line computational time performances of MDEIM and SMDEIM

Reduced Jacobian errors The accuracy of the reduced Jacobian approximations is measured in the reduced space against the standard $U^T\mathbf{J}_{\mathbf{r}^i}(\mathbf{x}_{t_i})U$, where the residual full Jacobian is computed from the high fidelity model solutions. Figure 7 describes the reduced derivatives errors at the initial time step calculated using Frobenius norm for various number of POD basis functions and DEIM indexes.

For the experiment designed in Figure 7(a), the number of interpolation indexes is set to 30 while the POD basis dimension is increased. We notice that the DEIM based approximations quality is highly dependent on the size of the reduced manifold. However SMDEIM and MDEIM are less affected since their proposed Jacobians are 2 orders of magnitude more accurate than DEIM approximation. Figure 7(b) illustrates the impact of an increased number of interpolation indexes onto the quality of Jacobian approximations while the dimension of the POD basis is maintained steady with $k = 20$. SMDEIM and MDEIM requires 25 interpolation indexes to reach the same precision as the tensorial, directional derivatives and direct projection methods while DEIM needs more than 50 points for the same accuracy level.



(a) $m = 30$ (b) $k = 20$

Figure 7: Reduced Jacobians errors

POD basis dimension	5	10	15	20	25	30	35	40	45	50
MDEIM/SMDEIM	3.98	4.11	4.37	4.50	4.61	4.66	4.71	4.75	4.80	4.82
DEIM	4.00	4.11	4.45	4.95	5.73	7.03	10.05	10.89	11.39	11.59
DirDer/DirPro/Tensor	3.85	4.09	4.37	4.50	4.61	4.66	4.70	4.78	4.81	4.83
Full	4.83	4.83	4.83	4.83	4.83	4.83	4.83	4.83	4.83	4.83

Table 3: The mean variation of Newton-Raphson iterates per time step along the change in the POD basis dimension

On-line computational performances Here we compare the on-line characteristics of the proposed implicit reduced order models. Two features are of interest, i.e. CPU time and solution accuracy. Figure 8(a) shows the amount of integration time required by the reduced order models to obtain their solutions with respect to the dimension of POD basis U. We set the number of DEIM interpolation indexes to 30. SMDEIM and MDEIM perform similarly thus confirming the teoretical on-line computational complexities derived in subsection 4.2. For 50 POD basis modes the MDEIM and SMDEIM are 2×, 3.75×, 4×, 7.43× 16× and 16.2× faster than directional derivative, direct projection, DEIM, high-fidelity and tensorial models, respectively. The efficiency loss in the case of tensorial method is in accordance with the results in table 2. The DEIM reduced order model performance is affected by the quality of the Jacobian approximation as we can notice in Figure 7(a) which doubles the averaged number of Newton-Raphson iterations per each time step (see table 3). While directional derivative and direct projection reduced order models use Matlab implementations based on vector operations, the other models don't due to their core algorithms nature. It is well known that Matlab is tuned to enhance efficient vector operations calculations thus the former models are advantaged. Similar Frobenius norms measuring the discrepancies between the projected reduced and high-fidelity solutions are obtained by all reduced order models (not shown due to data redundancy), however DEIM requires more Newton iterations as seen in table 3.

For the next experiment we measure the effect of an increased number of space points and the results are depicted in Figure 8(b). The dimension of POD basis is set to 25 and 30 DEIM interpolation indexes are selected. While only directional derivative and direct projection have on-line computational complexities depending on the dimension of the full space, all of the proposed

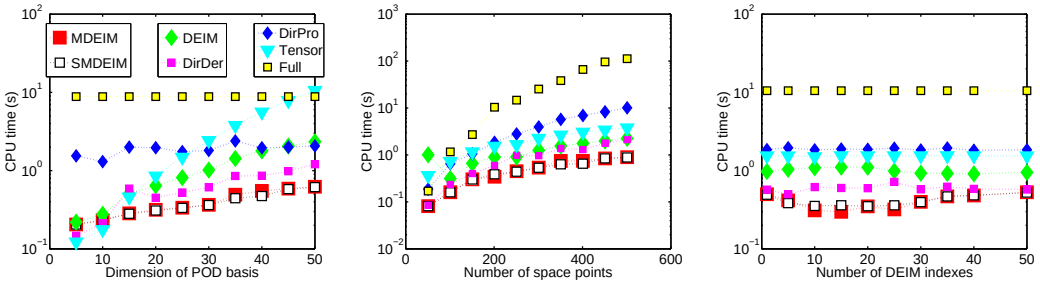
Number of space points	51	101	151	201	251	301	351	401	451	501
MDEIM/SMDEIM	6.14	5.38	4.95	4.61	4.44	4.33	4.23	4.12	3.97	3.92
DEIM	6.22	6.35	5.82	5.73	5.74	5.41	5.84	5.54	5.09	5.62
DirDer/DirPro/Tensor	6.14	5.38	4.95	4.61	4.43	4.33	4.23	4.12	3.95	3.92
Full	4.14	4.14	4.14	4.14	4.14	4.14	4.14	4.14	4.14	4.14

Table 4: The mean variation of Newton-Raphson iterates per time step along the change in the number of space points

Number of DEIM indexes	1	5	10	15	20	25	30	35	40	50
MDEIM/SMDEIM	8.58	6.88	5.64	4.98	4.62	4.61	4.61	4.61	4.61	4.61
DEIM	11.38	11.41	11.48	11.12	10.29	7.62	5.73	4.95	4.65	4.61
DirDer/DirPro/Tensor	4.61	4.61	4.61	4.61	4.61	4.61	4.61	4.61	4.61	4.61
Full	4.83	4.83	4.83	4.83	4.83	4.83	4.83	4.83	4.83	4.83

Table 5: The mean variation of Newton-Raphson iterates per time step along the change in the number of DEIM indexes

reduced order models suffer as the number of meshed points becomes larger. As the space dimension increases the number of time discretization points is raised too. For example, for 501 mesh points we use 1001 time steps and MDEIM and SMDEIM are 2.4×, 2.55×, 4.2×, 11× and 102× times faster than the directional derivative, DEIM, tensorial, direct projection, and full models. The quality of the Jacobian approximations is reflected in the number of Newton-Raphson iterations, and again DEIM requires more loops (see table 4) to achieve the same level of accuracy as the other reduced order models.



(a) $m = 30$ (b) $m = 30,\ k = 25$ (c) $k = 25$

Figure 8: On-line computational time performances of reduced order models

By increasing the number of interpolation indexes to 50 the CPU times for MDEIM and SMDEIM reduced order models are slightly increased as seen in Figure 8(c). The number of Newton iterations for matrix DEIM techniques becomes similar with those of exact Jacobian techniques for m larger than 20 indexes, while in the case of DEIM more than 40 DEIM indexes are required (5). This is also noticed in Figure 8(c) where the DEIM reduced order model CPU time is decreasing even if the number of interpolation indexes is raised.

5.2 Two-dimensional Shallow Water Equations

5.2.1 Numerical Scheme

In meteorological and oceanographic problems, one is often not interested in small time steps because the discretization error in time is small compared to the discretization error in space. The alternating direction fully implicit (ADI) scheme [37] considered in this paper is first order in both time and space and it is stable for large CFL condition numbers. It was also proved that the method is unconditionally stable for the linearized version of the SWE model. Other research work on this topic include efforts of Fairweather and Navon [26] and Navon and Villiers [57].

We are solving the SWE model using the β-plane approximation on a rectangular domain [37]

$$\frac{\partial w}{\partial t} = A(w)\frac{\partial w}{\partial x} + B(w)\frac{\partial w}{\partial y} + C(y)w, \quad (x,y) \in [0,L] \times [0,D], \quad t \in (0, t_{\mathrm{f}}], \tag{33}$$

where $w = (u, v, \phi)^T$ is a vector function and u, v are the velocity components in the x and y directions, 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

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

and f is the Coriolis term

$$f = \hat{f} + \beta(y - D/2), \ \beta = \frac{\partial f}{\partial y}, \quad y \in [0, D],$$

with $\hat{f}$ and β constants.

We assume periodic solutions in the x direction for all three state variables while in the y direction

$$v(x, 0, t) = v(x, D, t) = 0, \ x \in [0, L], \ t \in (0, t_{\mathrm{f}}]$$

and Neumann boundary condition are considered for u and ϕ.

We derive the initial conditions from the initial height condition No. 1 of Grammeltvedt [33] i.e.

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

and the initial velocity fields are calculated from the initial height field using the geostrophic relationship.

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 $[0, t_{\mathrm{f}}]$ using N_t equally distributed points and $\Delta t = t_{\mathrm{f}}/(N_t - 1)$. Next we define vectors of unknown variables containing approximate solutions such as

$$\boldsymbol{w}(t_N) \approx [w(x_i, y_j, t_N)]_{i=1,2,..,N_x-2,\ j=1,2,..,N_y-2} \in \mathbb{R}^{3(N_x-2)\times(N_y-2)}, \ N = 1, 2, ..N_t, \text{ (no boundaries included)}$$

The semi-discrete equations of SWE (33) contain six nonlinear functions F_{11}, F_{12}, F_{21}, F_{22}, F_{31}, F_{32} as described in [81, Section 4]. The ADI scheme splits the finite difference equations into two, taking

implicitly the x derivatives terms first while the y derivatives components are treated implicitly in the second stage. The discrete model was implemented in Fortran and uses a sparse matrix environment . For operations with sparse matrices we employed SPARSEKIT library [72] and the sparse linear systems obtained during the quasi-Newton iterations were solved using MGMRES library [7, 46, 73]. The LU decomposition is performed at every time step. All numerical experiments use the following constants: $L = 6000\,km$, $D = 4400\,km$, $t_{\mathrm{f}} = 3\,hours$, $\hat{f} = 10^{-4}\,sec^{-1}$, $\beta = 1.5 \cdot 10^{-11}\,sec^{-1}m^{-1}$, $g = 10\,m\,sec^{-2}$, $H_0 = 2000\,m$, $H_1 = 220\,m$, $H_2 = 133\,m$. The nonlinear algebraic systems are solved using a Newton-Raphson method and the allowed number of Newton iterations per each time step is set to 50. The solution is considered accurate enough when the euclidian norm of the residual is less than $1e^{-10}$.

5.2.2 Greedy based Jacobian approximation techniques

Initially we discuss the spectrum characteristics of the SMDEIM and MDEIM snapshots matrices of the high-fidelity 2D SWE model and then we compare the accuracy of their output Jacobian approximations against the one proposed by DEIM where the building blocks are the nonlinear functions approximations. The numerical experiments in this subsection are obtained for a space mesh of 21×15 points, with $\Delta x = 300$km and $\Delta y \approx 315$km. The integration time windows is set to 6h and we use 91 time steps ($N_t = 91$) with $\Delta t = 240$s.

The Jacobian approximations are constructed using 180 snapshots (ADI scheme has an intermediary step, thus the 91 time steps provide 180 state variables, Jacobians and nonlinear terms snapshots) obtained from the numerical solution of the full - order ADI finite difference SWE model. For the MDEIM method the matrices of snapshots belong to $\mathbb{R}^{\left(9(N_x-2)^2(N_y-2)^2\right)\times 180}$ for both directions. In the case of SMDEIM technique we extract the nonzero elements of the Jacobians and form matrices of snapshots of the sizes $32(N_y-2)+16(N_x-4)(N_y-2)\times 180$ (x direction) and $22(N_x-2)+16(N_y-4)(N_x-2)\times 180$ (y direction). The DEIM based Jacobians of the ADI SWE finite difference model are constructed using the Jacobians of the 6 nonlinear functions and each of the corresponding matrices of snapshots has $(N_x-2)(N_y-2)\times 180$ dense elements.

Figure 9 illustrates the singular values of the MDEIM and SMDEIM snapshots matrices in the x direction. The spectra are similar however the SMDEIM method is approximately $144\times$ times faster. Moreover, the number of nonzero entries of the MDEIM singular vectors is 4228, where 276 of these values are artificially created by the matrix factorization as a consequence of the induced round-off errors.

Figure 10 depicts the location of the DEIM indexes obtained by the algorithm 1 using MDEIM and SMDEIM singular vectors of the x-derivative implicit 2D SWE model. The first 20 indices shown in panel (a) correspond to singular values between 366 and $1e-3$ and match perfectly. For extremely small singular values ranging from $1e-12$ to $1e-13$ the output DEIM indexes differ significantly as noticed in panel (b). The MDEIM singular vectors propose also indexes outside of the Jacobian nonzeros bands not seen in the case of SMDEIM approach. However, for most of the applications, selecting such large number of DEIM indexes does not necessary enhance the quality of the approximation obtained with a smaller number of indices. The computational cost of finding the first 20 DEIM indexes using SMDEIM singular vectors is approximately $220\times$ times faster than in the case of employing MDEIM singular vectors.

Next we assemble the approximations (5), (9) and (3) of the x-derivative implicit 2D SWE model Jacobian at the initial time step. The Frobenius norm of the errors between the matrix approximations and its true representation is shown in Figure 11 (a). MDEIM and SMDEIM outputs present similar accuracy levels and the mismatches Frobenius norms are inverse proportionally with

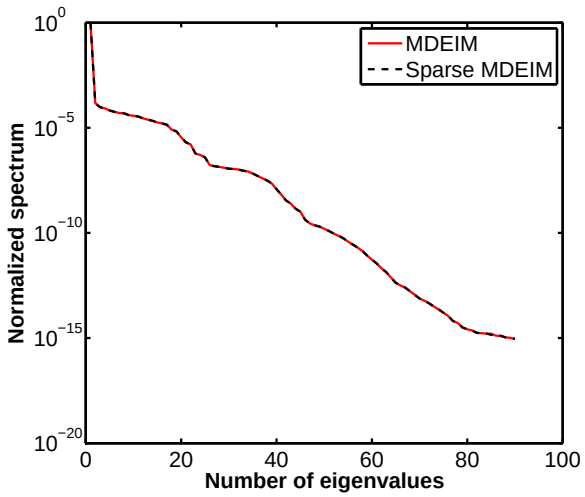
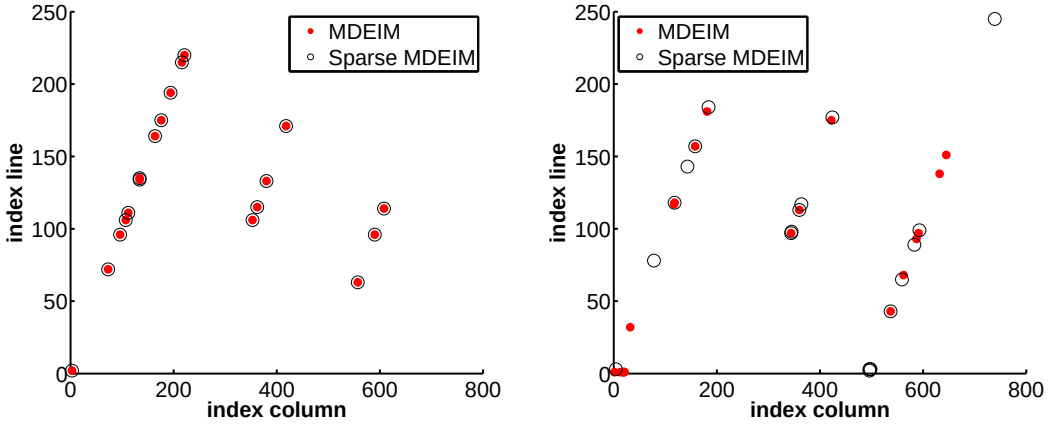


Figure 9: Spectrum properties of MDEIM and SMDEIM snapshots matrices.



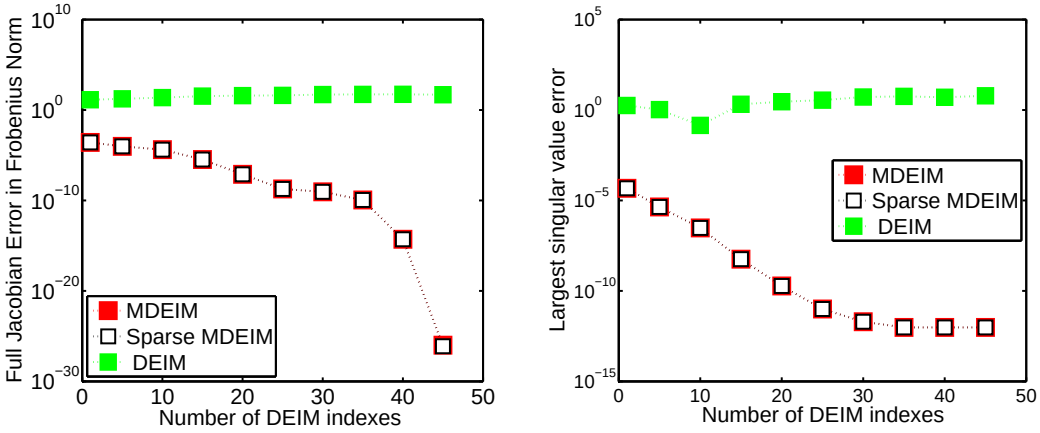
(a) 2D-SWE - $1^{\text{th}} - 20^{\text{th}}$ interpolation indexes (b) 2D-SWE - $80^{\text{th}} - 100^{\text{th}}$ interpolation indexes

Figure 10: Localization of DEIM indexes using MDEIM and SMDEIM singular vectors

the number of DEIM indexes. DEIM approximation is accurate only for the Jacobian rows selected by the interpolation indexes explaining the constant green trajectory even if the number of indexes is increased. The discrepancies between the largest singular value of the exact Jacobian and the greedy based approximations present similar pattern as in the case of the Frobenius norms. This confirms that matrix DEIM approximations are more accurate than the DEIM proposed Jacobian.

5.2.3 Performance of implicit reduced order models

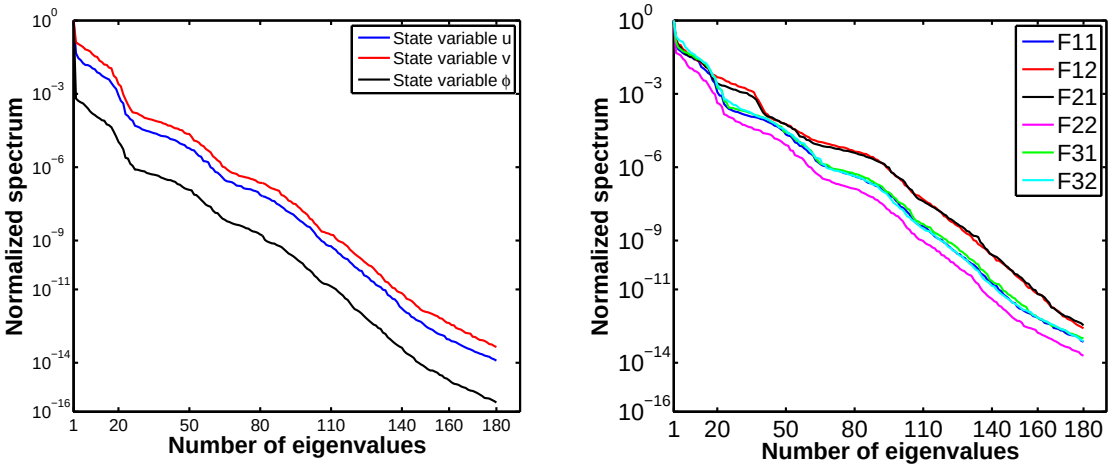
The proposed Jacobian approximations are embedded into POD reduced order framework using a Galerkin projection and the resulting reduced order models are compared in terms of computational cost and solution accuracy. Most of the discussed results are obtained for a space resolution of 61×45 points, with $\Delta x = \Delta y = 100km$. The models are integrated $6h$ in time and the number of time steps is set to $N_t = 91$. The POD bases functions of the state variables, Jacobians matrices and nonlinear terms are constructed using 180 snapshots obtained from the numerical solution of the



(a) 2D-SWE - 21×15 space points (b) 2D-SWE - 21×15 space points

Figure 11: Full Jacobian errors at initial time - Frobenius norm - Largest SVD.

full-order ADI finite difference SWE model. Figures 12(a–b) show the decay around the eigenvalues of the snapshot solutions for $u,\ v,\ \phi$ and the nonlinear snapshots $F_{11},\ F_{12},\ F_{21},\ F_{22},\ F_{31},\ F_{32}$. The state variables spectra decrease faster than those of the nonlinear terms.



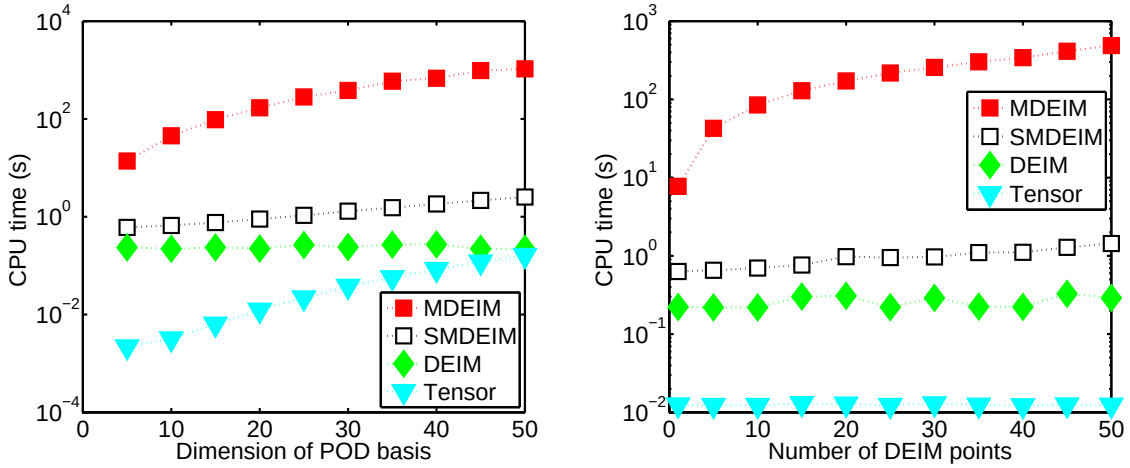
(a) Spectrum rate of decay (b) Eigenvalues rate of decay

Figure 12: Spectrum properties of snapshots matrices

Off-line computational performances During the off-line stages operations such as the singular value decomposition of the Jacobian snapshots matrices, calculation of interpolation indexes and computations of matrices $U^T V (P^T V)^{-1}$, $C \cdot V_m \left(P_m^T V_m\right)^{-1}$ and $\tilde{C} \cdot V_{m_{nz}} \left(P_{m_{nz}}^T V_{m_{nz}}\right)^{-1}$ in (3), (5) and (9) and tensor G (21) are required only by 4 of the discussed reduced order models described in subsection 4.2. These are the greedy based and tensorial surrogate models. Figure 13 describes the total off-line computational time required for the on-line Jacobians evaluations as a function of the POD state dimension and number of interpolation indexes. In Figure 13(a) we set the number of selected interpolation indexes to 20 and vary the dimensions of POD bases of state variables $\mathbf{u}$, $\mathbf{v}$ and $\boldsymbol{\phi}$ between 5 and 50. Tensorial approach has the smallest off-line cost however it becomes

slower with the increase of the POD basis size. Among the greedy based techniques, SMDEIM computational effort is 400× times smaller than in the case of MDEIM method and only 9× times larger than the DEIM CPU time. It is worth mentioning the compromise proposed by SMDEIM method which manages to preserve the same level of accuracy as the MDEIM method at reasonable costs.

Similar results are obtained if the number of DEIM interpolation indexes are varied as seen in Figure 13(b). The number of POD bases functions is choose 20 and the SMDEIM approach shows its efficiency being 1600 times faster than the MDEIM method.



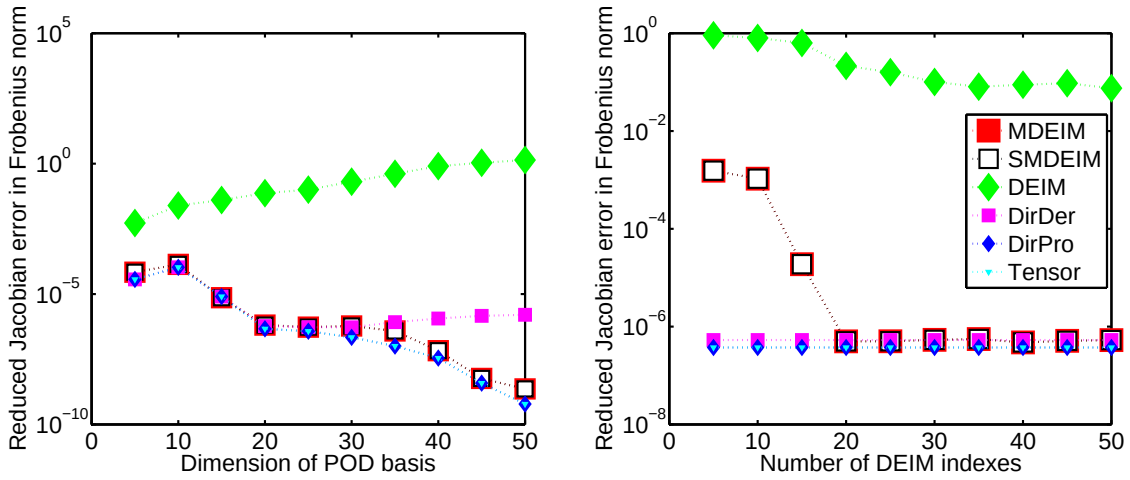
(a) $m = 20$; (b) $k = 20$;

Figure 13: Off-line computational time performances of DEIM based and tensorial ROMs

Reduced Jacobian errors The reduced residual Jacobian errors obtained using DEIM, MDEIM, SMDEIM, directional derivative, direct projection and tensorial methods are illustrated in Figure 14, where Frobenius norm is employed. Initially we measure the reduced Jacobian errors with respect to the number of POD bases functions and the results are presented in Figure 14(a). The number of DEIM indexes is set to 30. SMDEIM reduced derivatives present similar levels of accuracy as the ones computed using the MDEIM method for 50 POD basis functions. The direct projection and tensorial approaches present the most accurate reduced Jacobian. The directional derivative reduced derivatives are calculated for $h = 0.01$ in (23) and the precision is more than 3 order of magnitude lower than in the case of SMDEIM, direct projection and tensorial methods. As expected the DEIM approximation is less accurate explained by the level of precision of its full Jacobian approximation depicted in Figure 11.

Figure 14(b) describes the impact of the number of interpolation indexes onto the quality of reduced Jacobians. The dimension of POD bases is set to 25 and once the number of DEIM indexes is larger than 20 the MDEIM and SMDEIM reduced Jacobians are almost as accurate as the direct projection, directional derivative and tensorial Jacobians. In Figure 14(b) by increasing the number of DEIM points to 50 the quality of DEIM reduced Jacobian approximation is enhanced by one order of magnitude.

On-line computational performances At this space resolution, MDEIM already requires storing snapshots vectors of size $57e+6$ and the cost of assemblying the Jacobian approximation fir 50



(a) $m = 30$; (b) $k = 25$;

Figure 14: Reduced Jacobians errors

POD basis dimension	5	10	15	20	25	30	35	40	45	50
SMDEIM	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00
DEIM	4.00	5.00	5.00	5.00	5.00	6.00	7.00	8.00	9.00	9.00
DirDer/DirPro/Tensor	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00
Full	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00

Table 6: The mean variation of Newton-Raphson iterates per time step along the change in the POD basis dimension

DEIM points is of order of hundred of seconds (see Figure 13(a)). For 101×71 space points the computational complexity for building the current version of MDEIM reduced order model becomes prohibitive. Since the planned experiments for this subsection include testing the performances of reduced order models as functions of space dimension it is computationally infeasible to run the MDEIM model.

Here we analyze the on-line computational CPU times obtained by the studied reduced order models with respect to the number of POD bases functions, space points and DEIM indexes (see Figures 15). Since both direct projection and directional derivative reduced Jacobian computations depend on the full space dimension the corresponding reduced order models do not gain much efficiency sometimes being slower even than the high-fidelity model. Among them, the directional derivative approach is faster. Initially we fix the number of DEIM points to 30 and start increasing the size of POD basis. For smaller values of k, tensorial approach leads to the fastest surrogate model as noticed in Figure 15(a). Tensorial POD has a theoretical computational complexity depending on k and for POD size of 50 we notice that SMDEIM is $1.25\times$ times faster than the tensorial calculus based reduced order model thus confirming the theoretical results. The DEIM based surrogate model is $2.75\times$ slower than the SMDEIM model explained by their corresponding computational complexities in table 1 and the lack of Jacobian accuracy noticed in Figure 14. The latter forces the corresponding reduced order model to increase the number of Newton iterations (see table 6) in order to achieve a similar level of solution accuracy as the other surrogate models.

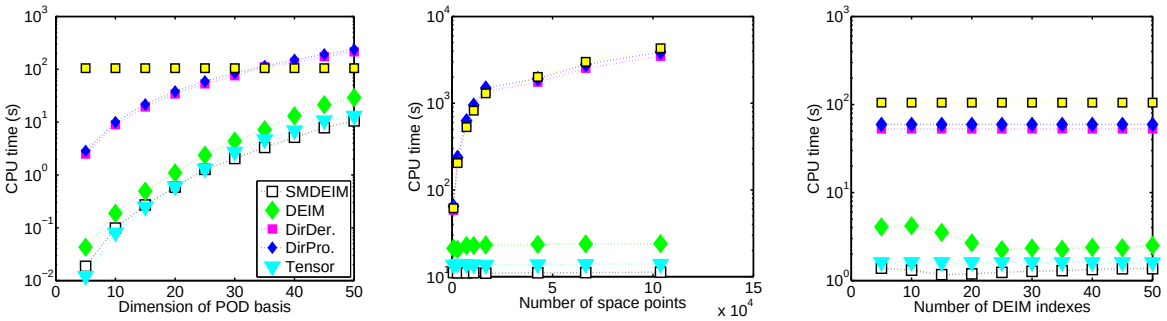
Next we set the number of DEIM indexes to 30 and the dimension of POD bases to 50. Figure

No of space points	713	2745	7171	10769	16761	42657	66521	103776
SMDEIM	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00
DEIM	6.00	6.00	7.00	7.00	7.00	8.00	8.00	8.00
DirDer/DirPro/Tensor	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00
Full	5.00	4.00	5.00	6.00	6.00	5.00	5.00	6.00

Table 7: The mean variation of Newton-Raphson iterates per time step along the change in the space dimension

15(b) describes the computational costs of the discussed reduced order models as a function of number of space points. For more than 10^5 mesh points we notice that reduced order SMDEIM model is 1.24, 2.12, 309, 344 and 400 times faster than tensorial, DEIM, directional derivative, direct projection and high-fidelity models. Clearly there is no advantage of using the direct projection and directional derivative approaches from computational complexity point of view. However they are useful due to their non-intrusive nature making them easy to implement even for very complex models.

The number of averaged Newton iterations per time step for DEIM reduced order model increases with the space dimensions while for SMDEIM, tensorial, directional derivative and direct projection it remains constant. The SMDEIM and tensorial reduced order models efficiency is also a consequence of the reduced number of Newton iterations in addition to the reduced number of degrees of freedom as shown in table 7.



(a) $m = 30$; (b) $m = 30,\ k = 50$; (c) $, k = 25$;

Figure 15: On-line computational time performances of 2D-SWE reduced order models

Figure 15(c) shows the on-line CPU times of the reduced order models for various numbers of DEIM indexes. We notice that the computational cost of DEIM reduced order model is decreasing and this behaviour is explained by the reduction in the average Newton iterations number per time steps (see table 8). This is a consequence of the Jacobians accuracy gain once the number of DEIM points is increased.

For all the experiments, the proposed reduced order solutions present similar accuracy levels obtained using various numbers of Newton iterations as seen in tables across this subsection. For different models, where higher nonlinearities are presented, we anticipate a different scenario where the reduced models errors are more sensitive to the quality of reduced order Jacobian. These scenarios advocate the use of a SMDEIM reduced order model since it will be much faster than tensorial model and more accurate than DEIM based reduced order model.

No of DEIM indexes	5	10	15	20	25	30	35	40	45	50
SMDEIM	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00
DEIM	13.00	12.00	10.00	6.00	6.00	5.00	5.00	5.00	5.00	5.00
DirDer/DirPro/Tensor	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00
Full	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00

Table 8: The mean variation of Newton-Raphson iterates per time step along the change in the number of DEIM indexes

The computational savings and accuracy levels obtained by the SMDEIM reduced order model depend on the number of POD modes and number of DEIM interpolation indexes. These numbers may be large in practice in order to capture well the full model dynamics. Local POD and local DEIM versions were proposed by Rapún and Vega [66] and Peherstorfer et al. [63] to alleviate this problem. A similar strategy can be applied for Jacobian approximation thus improving the performances of SMDEIM models. The idea of a local approach for nonlinear model reduction with local POD and local GNAT was first proposed by Amsallem et al. [1]. Machine learning techniques such as K-means [50, 53, 83] can be used for both time and space partitioning. A recent study investigating cluster-based reduced order modeling was proposed by Kaiser et al. [44]. First order and second order adjoint methodologies can be employed to compute useful error indicators [65] in the context of building statistically modeling errors [67, 24] to enhance the SMDEIM reduced order model solutions.

6 Conclusions

In the POD Galerkin approach to reduced order modeling the cost of evaluating nonlinear terms and their derivatives during the on-line stage scales with the full space dimension, and this constitutes a major efficiency bottleneck. This work introduces the sparse matrix discrete empirical interpolation method to compute accurate approximations of parametric matrices with constant sparsity structure. The approach is employed in a reduced order modeling framework to efficiently obtain accurate reduced order Jacobians. The sparse algorithm utilizes samples of of only the nonzero entries of the matrix series. The economy SVD factorization of the nonzero elements of the snapshots matrix, when appropriately padded with zeros, is equivalent to a valid economy SVD for the full snapshots matrix. The sparse SVD is applied to much smaller vectors and is therefore considerably more efficient.

In contrast with the traditional matrix DEIM method, the sparse version can be applied to much larger problems since its off-line computational complexity depends on the number of nonzero Jacobian elements, dimension of state POD basis, and the number of DEIM indexes. The on-line cost for computing reduced derivatives using SMDEIM is similar with the cost required by MDEIM.

An important application of SMDEIM is the construction of reduced order implicit time integration schemes since many problems arising in practice are stiff and so are the corresponding reduced order models.

Several strategies for Jacobian computations are implemented and the performance of the corresponding reduced order models is analyzed. These strategies are based on discrete empirical interpolation method applied to function, tensorial calculus, the full Jacobian projection onto the reduced basis subspace, and directional derivatives. Numerical experiments are carried out using the one dimensional Burgers equation and two dimensional SWE models. The construction of re-

duced order models is based on proper orthogonal decompositions and Galerkin projection, and the reduced nonlinearities are constructed using tensorial calculus. In the majority of experiments SMDEIM provides the fastest on-line reduced order model. For 10^5 mesh points the reduced order SMDEIM SWE model is 1.24, 2.12, 309, 344, and 400 faster than the tensorial, DEIM, direct projection, directional derivative, and high- fidelity models, respectively. For this space configuration the memory burden of the off-line stage of the traditional MDEIM reduced order model exceeded our computational resources - it was not possible to store n^2 dimensional vectors as required, with n representing the number of discrete variables. While being slower, the direct projection and directional derivative methods propose non-intrusive reduced order models. The numerical results showed that DEIM Jacobians approximations have a lower accuracy when using samples from the underlying functions instead of the Jacobian matrices. Therefore a larger number of Newton iterations are equired by the DEIM reduced order model to produce solutions as accurate as those of the other discussed surrogate models.

Future research will focus on decreasing the temporal complexity of the DEIM implicit scheme by exploiting the knowledge of the model's temporal behavior as proposed in Carlberg et al. [14]. Forecasting the unknown variables of the reduced-order system of nonlinear equations at future time steps provides an initial guess for the Newton-like solvers and can significantly decrease the number of linear systems solved at each step.

For the current test problems the reduced order models solutions are not very sensitive to the quality of the reduced order Jacobian approximations. However, for higher nonlinearities we anticipate a different behavior where the model errors are affected more by the quality of reduced order derivatives. For these cases, the SMDEIM reduced model is expected to be much faster than the tensorial model and more accurate than the DEIM surrogate model.

As future application we propose the use of SMDEIM reduced Jacobian approximation for strongly coupled fluid-structure problems (see Vierendeels et al. [90]) to approximate the Jacobian of the fluid and/or structural problem during the coupling iterations. On-going work by the authors focuses on reduced order constrained optimization. The current research represents an important step toward implementing the SMDEIM method for solving a reduced order optimal control problem such as the one discussed by Negri et al. [58].

Acknowledgments

The work of Dr. Răzvan Stefanescu and Prof. Adrian Sandu was supported by awards NSF CCF–1218454, NSF DMS–1419003, AFOSR FA9550–12–1–0293–DEF, AFOSR 12-2640-06, and by the Computational Science Laboratory at Virginia Tech.

References

References

[1] D. Amsallem, M.J. Zahr, and C. Farhat. Nonlinear model order reduction based on local reduced-order bases. International Journal for Numerical Methods in Engineering, 92(10): 891–916, 2012.

[2] Harbir Antil, Matthias Heinkenschloss, and Danny C Sorensen. Application of the discrete empirical interpolation method to reduced order modeling of nonlinear and parametric systems. In Reduced order methods for modeling and computational reduction, pages 101–136. Springer, 2014.

[3] A.C. Antoulas. Approximation of large-scale dynamical systems. Society for Industrial and Applied Mathematics, 6:376–377, 2009.

[4] P. Astrid, S. Weiland, K. Willcox, and T. Backx. Missing point estimation in models described by Proper Orthogonal Decomposition. IEEE Transactions on Automatic Control, 53(10):2237–2251, 2008.

[5] M. Baker, D. Mingori, and P. Goggins. Approximate Subspace Iteration for Constructing Internally Balanced Reduced-Order Models of Unsteady Aerodynamic Systems. AIAA Meeting Papers on Disc, pages 1070–1085, 1996.

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

[7] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst. Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition. SIAM, Philadelphia, PA, 1994.

[8] P. Benner and T. Breiten. Two-sided moment matching methods for nonlinear model reduction. Technical Report MPIMD/12-12, Max Planck Institute Magdeburg Preprint, June 2012.

[9] P. Benner and V.I. Sokolov. Partial realization of descriptor systems. Systems & Control Letters, 55(11):929 –938, 2006.

[10] D.A. Bistrian and I.M. Navon. Comparison of optimized Dynamic Mode Decomposition vs POD for the shallow water equations model reduction with large-time-step observations . Technical report, Submited to International Journal for Numerical Methods in Fluids, 2014.

[11] T. Bui-Thanh, M. Damodaran, and K. Willcox. Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition. AIAA Journal, pages 1505–1516, 2004.

[12] A. Bultheel and B. De Moor. Rational approximation in linear systems and control. Journal of Computational and Applied Mathematics, 121:355–378, 2000.

[13] K. Carlberg, C. Bou-Mosleh, and C. Farhat. Efficient non-linear model reduction via a least-squares Petrov-ŰGalerkin projection and compressive tensor approximations. International Journal for Numerical Methods in Engineering, 86(2):155–181, 2011.

[14] K. Carlberg, J. Ray, and B. van Bloemen Waanders. Decreasing the temporal complexity for nonlinear, implicit reduced-order models by forecasting. Technical Report arXiv:1209.5455, Submitted to Computational Methods in Applied Mechanics and Engineering, September 2012.

[15] K. Carlberg, R. Tuminaro, and P. Boggsz. Efficient structure-preserving model reduction for nonlinear mechanical systems with application to structural dynamics. preprint, Sandia National Laboratories, Livermore, CA 94551, USA, 2012.

[16] S. Chaturantabut. Dimension Reduction for Unsteady Nonlinear Partial Differential Equations via Empirical Interpolation Methods. Technical Report TR09-38,CAAM, Rice University, 2008.

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

[18] S. Chaturantabut and D.C. Sorensen. A state space error estimate for POD-DEIM nonlinear model reduction. SIAM Journal on Numerical Analysis, 50(1):46–63, 2012.

[19] T. A. Davis, J. R. Gilbert, S. I. Larimore, , and E. G. Ng. A column approximate minimum degree ordering algorithm. ACM Trans. Math. Softw., 30(3):353âĂŞ376, 2004.

[20] T. A. Davis, J. R. Gilbert, S. I. Larimore, , and E. G. Ng. Algorithm 836:colamd, a column approximate minimum degree ordering algorithm. ACM Trans. Math. Softw., 30(3):377âĂŞ380, 2004.

[21] RJ Dedden. Model order reduction using the discrete empirical interpolation method. Master's thesis, TU Delft, Delft University of Technology, 2012.

[22] M. Dihlmann and B. Haasdonk. Certified PDE-constrained parameter optimization using reduced basis surrogate models for evolution problems. Submitted to the Journal of Computational Optimization and Applications, 2013. URL `http://www.agh.ians.uni-stuttgart.de/publications/2013/DH13`.

[23] Z. Drmac and S. Gugercin. A New Selection Operator for the Discrete Empirical Interpolation Method – improved a priori error bound and extensions, 2015. URL `http://arxiv.org/abs/1505.00370`.

[24] M. Drohmann and K. Carlberg. The ROMES method for statistical modeling of reduced-order-model error. Technical Report arXiv:1405.5170, Cornell University, May 2014.

[25] R. Everson and L. Sirovich. KarhunenŰ-Loeve procedure for gappy data. Journal of the Optical Society of America A, 12:1657–64, 1995.

[26] G. Fairweather and I.M. Navon. A linear ADI method for the shallow water equations. Journal of Computational Physics, 37:1–18, 1980.

[27] P. Feldmann and R.W. Freund. Efficient linear circuit analysis by Pade approximation via the Lanczos process. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 14:639–649, 1995.

[28] R.W. Freund. Model reduction methods based on Krylov subspaces. Acta Numerica, 12: 267–319, 2003.

[29] K. Gallivan, E. Grimme, and P. Van Dooren. Padé approximation of large-scale dynamic systems with lanczos methods. In Decision and Control, 1994., Proceedings of the 33rd IEEE Conference on, volume 1, pages 443–448 vol.1, Dec 1994.

[30] G.H. Golub and C.F. van Loan. Matrix Computations. The Johns Hopkins University Press, 3rd ed. edition, 1996.

[31] W.B. Gragg. The Padé table and its relation to certain algorithms of numerical analysis. SIAM Review, 14:1–62, 1972.

[32] W.B. Gragg and A. Lindquist. On the partial realization problem. Linear Algebra and Its Applications, Special Issue on Linear Systems and Control, 50:277 –319, 1983.

[33] A. Grammeltvedt. A survey of finite difference schemes for the primitive equations for a barotropic fluid. Monthly Weather Review, 97(5):384–404, 1969.

[34] M.A. Grepl and A.T. Patera. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. ESAIM: Mathematical Modelling and Numerical Analysis, 39(01):157–181, 2005.

[35] Grimme1997. Krylov projection methods for model reduction. PhD thesis, Univ. Illinois, Urbana-Champaign, 1997.

[36] T. Gudmundsson and A. Laub. Approximate Solution of Large Sparse Lyapunov Equations. IEEE Transactions on Automatic Control, 39(5):1110–1114, 1994.

[37] B. Gustafsson. An alternating direction implicit method for solving the shallow water equations. Journal of Computational Physics, 7:239–254, 1971.

[38] M.H. Gutknecht. The Lanczos process and Padé approximation. Proc. Cornelius Lanczos Intl. Centenary Conference, edited by J.D. Brown et al., SIAM, Philadelphia, pages 61–75, 1994.

[39] W.W. Hager. Minimizing the proïňĄle of a symmetric matrix. SIAM J. Sci., 23(5): 1799âĂŞ1816, 2002.

[40] A.S. Hodel. Least Squares Approximate Solution of the Lyapunov Equation. Proceedings of the 30th IEEE Conference on Decision and Control, IEEE Publications, Piscataway, NJ, 1991.

[41] H. Hotelling. Analysis of a complex of statistical variables with principal components. Journal of Educational Psychology, 24:417–441, 1933.

[42] T. Iliescu and Z. Wang. Are the Snapshot Difference Quotients Needed in the Proper Orthogonal Decomposition? Technical Report arXiv:1303.6012, Cornell University, March 2013.

[43] I. Jaimoukha and E. Kasenally. Krylov Subspace Methods for Solving Large Lyapunov Equations. SIAM Journal of Numerical Analysis, 31(1):227–251, 1994.

[44] E. Kaiser, Bernd R. Noack, L. Cordier, A. Spohn, M. Segond, M. Abel, G. Daviller, and R.K. Niven. Cluster-based reduced-order modelling of a mixing layer. J. Fluid Mech., 754:365–414, 2014.

[45] K. Karhunen. Zur spektraltheorie stochastischer prozesse. Annales Academiae Scientarum Fennicae, 37, 1946.

[46] C. T. Kelley. Iterative Methods for Linear and Nonlinear Equations. Number 16 in Frontiers in Applied Mathematics. SIAM, 1995.

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

[48] K. Kunisch and S. Volkwein. Galerkin Proper Orthogonal Decomposition Methods for a General Equation in Fluid Dynamics. SIAM Journal on Numerical Analysis, 40(2):492–515, 2002.

[49] C. Lieberman, K. Willcox, and O. Ghattas. Parameter and state model reduction for large-scale statistical inverse problems. SIAM Journal on Scientific Computing, 32(5):2523–2542, 2010.

[50] S. Lloyd. Least squares quantization in PCM. IEEE Trans. Inform. Theory, 28:129–137, 1957.

[51] M.M. Loève. Probability Theory. Van Nostrand, Princeton, NJ, 1955.

[52] E.N. Lorenz. Empirical Orthogonal Functions and Statistical Weather Prediction. Technical report, Massachusetts Institute of Technology, Dept. of Meteorology, 1956.

[53] J. MacQueen. Some methods for classification and analysis of multivariate observations. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1:281–297, 1967.

[54] S.C. Madeira and A.L. Oliveira. Biclustering algorithms for biological data analysis: A survey. IEEE/ACM Trans. Comput. Biol. Bioinformatics, 1(1):24–45, 2004.

[55] B.C. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. IEEE Transactions on Automatic Control, 26(1):17–32, 1981.

[56] C.T. Mullis and R.A. Roberts. Synthesis of Minimum Roundoff Noise Fixed Point Digital Filters. IEEE Transactions on Circuits and Systems, CAS-23:551–562, 1976.

[57] I. M. Navon and R. De Villiers. GUSTAF: A Quasi-Newton Nonlinear ADI FORTRAN IV Program for Solving the Shallow-Water Equations with Augmented Lagrangians. Computers and Geosciences, 12(2):151–173, 1986.

[58] F. Negri, G. Rozza, A. Manzoni, and A. Quarteroni. Reduced basis method for parametrized elliptic optimal control problems. SIAM Journal on Scientific Computing, 35(5):A2316–A2340, 2013.

[59] N.C. Nguyen, A.T. Patera, and J. Peraire. A 'best points' interpolation method for efficient approximation of parametrized function. International Journal for Numerical Methods in Engineering, 73:521–543, 2008.

[60] B.R. Noack, K. Afanasiev, M. Morzynski, G. Tadmor, and F. Thiele. A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. Journal of Fluid Mechanics, 497:335–363, 2003. ISSN 0022-1120.

[61] B.R. Noack, M. Schlegel, M. Morzynski, and G. Tadmor. System reduction strategy for galerkin models of fluid flows. International Journal for Numerical Methods in Fluids, 63(2):231–248, 2010.

[62] A.T. Patera and G. Rozza. Reduced basis approximation and a posteriori error estimation for parametrized partial differential equations, 2007.

[63] B. Peherstorfer, D. Butnaru, K. Willcox, and H.J. Bungartz. Localized Discrete Empirical Interpolation Method. MIT Aerospace Computational Design Laboratory Technical Report TR-13-1, 2013.

[64] S Rajamanickam. EfiňĄcient algorithms for sparse singular value decomposition. PhD thesis, University of Florida, 2009.

[65] V. Rao and A. Sandu. A Posteriori Error Estimates for DDDAS Inference Problems. Procedia Computer Science, 29(0):1256 – 1265, 2014. 2014 International Conference on Computational Science.

[66] M.L. Rapún and J.M. Vega. Reduced order models based on local POD plus Galerkin projection. Journal of Computational Physics, 229(8):3046–3063, 2010.

[67] O. Roderick, M. Anitescu, and Y. Peet. Proper orthogonal decompositions in multifidelity uncertainty quantification of complex simulation models. Technical report, Submited to International Journal of Computer Mathematics, 2013.

[68] C. W. Rowley, T. Colonius, , and R. M. Murray. Model reduction for compressible flows using POD and Galerkin projection. Physica D. Nonlinear Phenomena, 189(1–2):115–129, 2004.

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

[70] C.W. Rowley, I. Mezic, S. Bagheri, P.Schlatter, and D.S. Henningson. Spectral analysis of nonlinear flows. Journal of Fluid Mechanics, 641:115–127, 2009.

[71] G. Rozza, D.B.P. Huynh, and A.T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. Archives of Computational Methods in Engineering, 15(3):229–275, 2008.

[72] Y. Saad. Sparsekit: a basic tool kit for sparse matrix computations. Technical Report, Computer Science Department, University of Minnesota, 1994.

[73] Y. Saad. Iterative Methods for Sparse Linear Systems. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2nd edition, 2003.

[74] O. San and T. Iliescu. Proper orthogonal decomposition closure models for fluid flows: Burgers equation. Technical Report arXiv:1308.3276 [physics.flu-dyn], Cornell University, August 2013.

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

[76] L. Sirovich. Turbulence and the dynamics of coherent structures. I. Coherent structures. Quarterly of Applied Mathematics, 45(3):561–571, 1987. ISSN 0033-569X.

[77] L. Sirovich. Turbulence and the dynamics of coherent structures. II. Symmetries and transformations. Quarterly of Applied Mathematics, 45(3):573–582, 1987. ISSN 0033-569X.

[78] L. Sirovich. Turbulence and the dynamics of coherent structures. III. Dynamics and scaling. Quarterly of Applied Mathematics, 45(3):583–590, 1987. ISSN 0033-569X.

[79] D.C. Sorensen and A.C. Antoulas. The Sylvester equation and approximate balanced reduction. Linear Algebra and its Applications, 351-352(0):671–700, 2002.

[80] R. Stefanescu and I.M. Navon. POD/DEIM Nonlinear model order reduction of an ADI implicit shallow water equations model. Journal of Computational Physics, 237:95–114, 2013.

[81] R. Stefanescu, A. Sandu, and I.M. Navon. Comparison of POD Reduced Order Strategies for the Nonlinear 2D Shallow Water Equations. International Journal for Numerical Methods in Fluids, 76(8):497–521, 2014.

[82] R. Stefanescu, A. Sandu, and I.M. Navon. POD/DEIM Reduced-Order Strategies for Efficient Four Dimensional Variational Data Assimilation. Technical Report TR 3, Virginia Polytechnic Institute and State University, March 2014, also submitted to Journal of Computational Physics.

[83] H. Steinhaus. Sur la division des corps matériels en parties. Bulletin of the Polish Academy of Sciences, 4(12):801–804, 1956.

[84] Paolo Tiso and Daniel J Rixen. Discrete empirical interpolation method for finite element structural dynamics. In Topics in Nonlinear Dynamics, Volume 1, pages 203–212. Springer, 2013.

[85] G. Tissot, L. Cordir, N. Benard, and B. Noack. Model reduction using Dynamic Mode Decomposition. Comptes Rendus Mécanique, 342(6-7):410–416, 2014.

[86] T. Tonn. Reduced-Basis Method (RBM) for Non-Affine Elliptic Parametrized PDEs. (PhD), Ulm University, 2012.

[87] L.N. Trefethen and D. Bau III. Numerical Linear Algebra. Society for Industrial and Applied Mathematics, 1st ed. edition, 1997.

[88] P. Van Dooren. The Lanczos algorithm and Padé approximations. In Short Course, Benelux Meeting on Systems and Control, 1995.

[89] P.T.M. Vermeulen and A.W. Heemink. Model-reduced variational data assimilation. Mon. Wea. Rev., 134:2888–2899, 2006.

[90] J. Vierendeels, L. Lanoye, J. Degroote, and P. Verdonck. Implicit coupling of partitioned fluid-structure interaction problems with reduced order models. Comput. Struct., 85(11-14): 970–976, 2007.

[91] K. Willcox and J. Peraire. Balanced model reduction via the Proper Orthogonal Decomposition. AIAA Journal, pages 2323–2330, 2002.

[92] D. Wirtz, D.C. Sorensen, and B. Haasdonk. A-posteriori error estimation for DEIM reduced nonlinear dynamical systems. SRC SimTech Preprint Series, 2012.

[93] Y.B. Zhou. Model reduction for nonlinear dynamical systems with parametric uncertainties. (M.S), Massachusetts Institute of Technology, Dept. of Aeronautics and Astronautics, 2012.