Journal of Computational Physics 324 (2016) 194–209

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Nonlinear model-order reduction for compressible flow solvers using the Discrete Empirical Interpolation Method

Miguel Fosas de Pando [a,*], Peter J. Schmid [b], Denis Sipp [c]

[a] *Dpto. Ing. Mecánica y Diseño Industrial, Escuela Superior de Ingeniería, Universidad de Cádiz, Av. de la Universidad de Cádiz 10, 11519 Puerto Real, Spain*
[b] *Dept. of Mathematics, Imperial College London, London SW7 2AZ, United Kingdom*
[c] *ONERA-DAFE, 8 rue des Vertugadins, 92190 Meudon, France*

ARTICLE INFO

Article history:
Received 8 September 2015
Received in revised form 2 May 2016
Accepted 3 August 2016
Available online 9 August 2016

Keywords:
Reduced-order models
Discrete empirical interpolation method
Proper orthogonal decomposition
Compressible flows
Aeroacoustics

ABSTRACT

Nonlinear model reduction for large-scale flows is an essential component in many fluid applications such as flow control, optimization, parameter space exploration and statistical analysis. In this article, we generalize the POD–DEIM method, introduced by Chaturantabut & Sorensen [1], to address nonlocal nonlinearities in the equations without loss of performance or efficiency. The nonlinear terms are represented by nested DEIM-approximations using multiple expansion bases based on the Proper Orthogonal Decomposition. These extensions are imperative, for example, for applications of the POD–DEIM method to large-scale compressible flows. The efficient implementation of the presented model-reduction technique follows our earlier work [2] on linearized and adjoint analyses and takes advantage of the modular structure of our compressible flow solver. The efficacy of the nonlinear model-reduction technique is demonstrated to the flow around an airfoil and its acoustic footprint. We could obtain an accurate and robust low-dimensional model that captures the main features of the full flow.

1. Introduction

Over the past decades, advances in high-performance computing have enabled numerical simulations of increasingly complex fluid flows. At present, typical supercomputers are capable of handling large-scale flow simulations with numbers of degrees of freedom that range from millions to billions of flow variables. While these simulations present invaluable physical insight into fluid processes and unprecedented opportunities for the analysis and design of fluid devices, the large-scale nature of the flow makes optimization, design and control prohibitively expensive and often beyond the reach of standard algorithms.

For many applications in computational fluid dynamics, simulations often represent multi-query systems where multiple evaluations of the governing equations are necessary to predict the future evolution of the flow fields, to iterate on an optimal control strategy, to scan a region of parameter space, or to gather statistical information about an output quantity in a Monte-Carlo fashion, among many other inquiries. In these applications, the full evaluation of the high-dimensional system would be far too slow and inefficient; for example, in model-predictive control applications, the optimal control strategy has to be determined and has to arrive in real time for the actuator to act.

[*] Corresponding author.
E-mail address: miguel.fosas@uca.es (M. Fosas de Pando).

http://dx.doi.org/10.1016/j.jcp.2016.08.004
0021-9991/

A common strategy to reduce the evaluation cost in the above studies consists in replacing the full dynamical system by a reduced-order model, i.e., a lower-dimensional dynamical system that captures the dynamics of the full system [3,4]. Consequently, the flow response, or other output quantities, can be computed at a fraction of the computational cost [5–9]. In the past, a great deal of research efforts have been devoted at devising reduced-order modeling strategies for fluid systems that exhibit linear dynamics, such as amplifier-type flows. For this case, well-known techniques exist that preserve the principal dynamic features or a specific input–output behavior. In contrast, for the case of fluid systems that are characterized by nonlinear behavior, such as oscillator-type flows settled in self-sustained, saturated limit cycles, the procedural steps for constructing reduced-order models are far less trivial and require further investigation.

A customary technique for the design of reduced-order models consists of projecting the original (full) dynamical system onto a basis generated by the Proper Orthogonal Decomposition (POD) of carefully selected flow field sequences. Even though this approach is successful in reducing the size of the dynamical system and achieves significant savings in the evaluation of linear terms, the cost of evaluating the nonlinear terms is equivalent to that of the full system.

Recently, Chaturantabut and Sorensen (2010) [1], addressed this issue and proposed a novel model-order reduction strategy for nonlinear dynamical systems, named POD–DEIM, that aims at further reducing the evaluation costs of the nonlinear terms in POD-based reduced-order models. This technique considers two different POD-based vector bases: the first basis provides a low-rank approximation of the state vector dynamics, while the second POD-basis furnishes a low-rank approximation of the dynamics governed by the nonlinear terms in the dynamical system. A link between the two bases is established by an interpolation algorithm, the Discrete Empirical Interpolation Method (DEIM), and the evaluation of the nonlinear terms in the reduced-order model is reduced to a small number of interpolation points. These latter interpolation points are judiciously selected using a greedy algorithm that subsequently minimizes the representation error of the nonlinear terms. In principle, the cost of approximating the nonlinear terms, even if evaluated only at a few grid points, may require calculations involving the original (high-dimensional) flow field. However, by introducing certain assumptions regarding the structure of the nonlinear terms, the authors show that the computational costs can be further reduced. The resulting dynamical system is finally projected onto the first POD-basis, yielding a fully-reduced dynamical system. The reader interested in error estimates for POD–DEIM reduced-order models is referred to [10,11].

The POD–DEIM technique has been demonstrated on a number of ordinary and partial differential equations [1,12], including applications in fluid dynamics where it has shown great potential for nonlinear reduced-order modeling. However, the computational savings strongly depend on the structure of the nonlinear terms in the governing equations and in the choice of the spatio-temporal discretization schemes. More specifically, it is sought that the evaluation of the nonlinear terms at the interpolation points depends only on a limited number of neighboring grid points. This limitation restricts the application of the POD–DEIM algorithm to flow solvers that have spatial schemes with narrow stencils, i.e., low-order schemes.

Unfortunately, common high-order numerical flow solvers feature spatial discretization schemes that have wide stencils, such as, e.g., pseudo-spectral and spectral methods, and the computational savings are thus compromised. For this reason, an effective implementation of POD–DEIM for compressible flow solvers is not straightforward. Similar shortcomings are encountered in the context of the evaluation of the Jacobian operator and its adjoint from numerical flow solvers: if the stencil width is small, the resulting operators are sparse and can be stored explicitly. In contrast, if the stencil width is large (as in high-order numerical solvers), the resulting operators are dense and explicit storage typically exceeds available memory resources.

In [2] an efficient procedure for evaluating matrix–vector products involving the Jacobian and its adjoint has been presented. In this study, the authors show that even though the Jacobian is dense, an efficient procedure for matrix–vector evaluations can be derived by isolating the nonlinearities in component-wise functions. In the present work, we use the ideas put forward in [2] to derive an efficient procedure for the construction of reduced-order models using POD–DEIM in compressible flow solvers that overcomes the limitations mentioned above.

The paper is organized as follows. In section 2 we give a brief overview of the POD–DEIM technique as introduced in [1] and demonstrate its main features on the nonlinear Ginzburg–Landau equation. Then, in section 3, we propose an extension to the DEIM algorithm that efficiently takes into account more complicated forms of nonlinearities, as encountered in the inviscid terms of the compressible Euler equations. The application to the viscous, compressible Navier–Stokes case is considered in section 4. These extensions are demonstrated in section 5 when we apply model reduction to the compressible flow around an airfoil. Conclusions and a summary of our results are given in section 6.

2. Model-order reduction by the POD–DEIM method

2.1. Model-order reduction by projection methods

As a starting point we take a coupled system of nonlinear ordinary differential equations of the form

$$\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = \mathbf{F}(\mathbf{v}), \tag{1}$$

that typically arises from the spatial discretization of evolution equations in continuous space. We denote by $\mathbf{v}$ the composite vector containing the field evolution with m entries and by $\mathbf{F}(\mathbf{v})$ the spatially-discretized right-hand-side of the governing equations.

In the context of reduced-order modeling, it is appropriate to consider constant, linear and nonlinear terms in $\mathbf{F}(\mathbf{v})$ separately. For this reason, we may equivalently write the right-hand side $\mathbf{F}(\mathbf{v})$ as $\mathbf{f}+\mathbf{A}\mathbf{v}+\mathbf{F}'(\mathbf{v})$, where $\mathbf{f}$ is a constant column vector with m rows, $\mathbf{A}$ is a $m \times m$ matrix capturing the linear dynamics, and $\mathbf{F}'(\mathbf{v})$ is a nonlinear function of $\mathbf{v}$ that by construction can be written as

$$\mathbf{F}'(\mathbf{v}) = \mathbf{F}(\mathbf{v}) - \mathbf{A}\mathbf{v} - \mathbf{f}. \tag{2}$$

Hence,

$$\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = \mathbf{f} + \mathbf{A}\mathbf{v} + \mathbf{F}'(\mathbf{v}). \tag{3}$$

Projection techniques for model-order reduction typically seek to approximate the state vector $\mathbf{v}$ by an element in a low-rank vector subspace determined by a $m \times k$ matrix $\mathbf{V}$ whose column vectors form a basis for the subspace. In what follows, the basis $\mathbf{V}$ is obtained from the k leading modes, with $k \ll m$, of the proper orthogonal decomposition (POD) of the temporal evolution of $\mathbf{v}$, and we denote by $\{\mathbf{v}\}_{i=1}^{n}$ the set of snapshots from where the vector basis $\mathbf{V}$ is obtained. The state vector $\mathbf{v}$ is then approximated in this basis by $\mathbf{V}\tilde{\mathbf{v}}$, where $\tilde{\mathbf{v}}$ is a column vector with k entries, and the governing equations are projected onto the basis $\mathbf{V}$ using the inner product given by $\langle \mathbf{v}, \mathbf{u} \rangle = \mathbf{v}^{\dagger}\mathbf{u}$. Since the vector columns of $\mathbf{V}$ are mutually orthonormal, the relation $\mathbf{V}^{\dagger}\mathbf{V} = \mathbf{I}$ holds. Note that the developments presented hereinafter can be easily extended to different choices of inner products. Hence,

$$\frac{\mathrm{d}\tilde{\mathbf{v}}}{\mathrm{d}t} = \tilde{\mathbf{f}} + \tilde{\mathbf{A}}\tilde{\mathbf{v}} + \mathbf{V}^{\dagger}\mathbf{F}'(\mathbf{V}\tilde{\mathbf{v}}), \tag{4}$$

where we have introduced $\tilde{\mathbf{f}} = \mathbf{V}^{\dagger}\mathbf{f}$, and $\tilde{\mathbf{A}} = \mathbf{V}^{\dagger}\mathbf{A}\mathbf{V}$. We also note that $\mathbf{V}^{\dagger}\mathbf{F}'(\mathbf{V}\tilde{\mathbf{v}}) = \mathbf{V}^{\dagger}\left(\mathbf{F}(\mathbf{V}\tilde{\mathbf{v}}) - \mathbf{A}\mathbf{V}\tilde{\mathbf{v}} - \mathbf{f}\right)$.

Even though the dimension of the dynamical system has been reduced to k, and the cost of evaluating the linear terms on the right-hand side, i.e., $\tilde{\mathbf{A}}\tilde{\mathbf{v}}$, has been reduced to $\mathcal{O}(k^2)$ operations, the cost of evaluating the nonlinear terms $\mathbf{V}^{\dagger}\mathbf{F}'(\mathbf{V}\tilde{\mathbf{v}})$ has not been reduced yet.

2.2. Discrete Empirical Interpolation Method

The aim of the Discrete Empirical Interpolation Method [1] (DEIM) is to reduce the operation count in the evaluation of nonlinearities — the evaluation of the nonlinear terms in equation (4) still requires $\mathcal{O}(m)$ operations.

To this end, the reduced-order model derived in equation (4) is supplemented with an additional basis, denoted by $\mathbf{U}$, for the specific low-order representation of the nonlinear terms. This basis is generated from the leading l modes of the proper orthogonal decomposition of the sequence of nonlinear snapshots given by $\left\{\mathbf{F}'(\mathbf{v}(t_i))\right\}_{i=1}^{n}$. Hence

$$\mathbf{F}'(\mathbf{v}) \approx \mathbf{F}'(\mathbf{V}\tilde{\mathbf{v}}) \approx \mathbf{U}\tilde{\mathbf{c}}. \tag{5}$$

The above system of equations is overdetermined, and a suitable procedure to obtain the coefficients $\tilde{\mathbf{c}}$ must be introduced. As suggested in [1], one option is to enforce the above relation at l components of $\mathbf{F}_i(\mathbf{V}\tilde{\mathbf{v}})$. In what follows, we will denote this set of components by a vector $\mathbf{p}_i$, where $i \in \{1, \ldots, l\}$, and introduce a matrix $\mathbf{P} \in \mathbb{R}^{m \times l}$ whose columns are the unit vectors $\mathbf{e}_{\mathbf{p}_i}$ in $\mathbb{R}^m$ such that, when $\mathbf{P}^{\dagger}$ is multiplied to a column vector, it selects the $\mathbf{p}_i$ elements of this vector. The vector $\mathbf{p}$ and the matrix $\mathbf{P}$ are thus referred to as interpolation points and row-selector matrix, respectively. We shall later summarize the procedure that selects the interpolation points and the row-selector matrix.

Multiplying equation (5) from the left by $\mathbf{P}^{\dagger}$, the coefficient vector $\tilde{\mathbf{c}}$ is obtained by inverting the matrix $(\mathbf{P}^{\dagger}\mathbf{U})$, which is now a square $l \times l$ matrix and it can be shown to be non-singular (see below). Hence,

$$\tilde{\mathbf{c}} = \left(\mathbf{P}^{\dagger}\mathbf{U}\right)^{-1} \mathbf{P}^{\dagger}\mathbf{F}'(\mathbf{V}\tilde{\mathbf{v}}), \tag{6}$$

and the reduced-order model reads now

$$\frac{\mathrm{d}\tilde{\mathbf{v}}}{\mathrm{d}t} = \mathbf{V}^{\dagger}\mathbf{A}\mathbf{V}\tilde{\mathbf{v}} + (\mathbf{V}^{\dagger}\mathbf{U})\left(\mathbf{P}^{\dagger}\mathbf{U}\right)^{-1}\mathbf{P}^{\dagger}\mathbf{F}(\mathbf{V}\tilde{\mathbf{v}}). \tag{7}$$

The operation count has not been reduced yet: the evaluation of $\mathbf{P}^{\dagger}\mathbf{F}(\mathbf{V}\tilde{\mathbf{v}})$—representing the contribution of the nonlinearities at the interpolation points—still requires $\mathcal{O}(m)$ operations. Nevertheless, by introducing the assumption that $\mathbf{F}'(\mathbf{v})$ is a component-wise function, i.e., the i-th component $\mathbf{F}'_i(\mathbf{v})$ only depends on the i-th component of the state vector $\mathbf{v}_i$, we can write $\mathbf{P}^{\dagger}\mathbf{F}'(\mathbf{V}\tilde{\mathbf{v}}) = \mathbf{F}'(\mathbf{P}^{\dagger}\mathbf{V}\tilde{\mathbf{v}})$, reducing thus the cost of evaluation. (In section 3 this restriction will be relaxed.) Finally, the nonlinear reduced-order model reads

$$\frac{\mathrm{d}\tilde{\mathbf{v}}}{\mathrm{d}t} = \mathbf{V}^{\dagger}\mathbf{A}\mathbf{V}\tilde{\mathbf{v}} + (\mathbf{V}^{\dagger}\mathbf{U})\left(\mathbf{P}^{\dagger}\mathbf{U}\right)^{-1}\mathbf{F}'(\mathbf{P}^{\dagger}\mathbf{V}\tilde{\mathbf{v}}), \tag{8}$$

or, using a more compact notation,

$$\frac{\mathrm{d}\tilde{\mathbf{v}}}{\mathrm{d}t} = \tilde{\mathbf{f}} + \tilde{\mathbf{A}}\tilde{\mathbf{v}} + \tilde{\mathbf{U}}\mathbf{F}'(\hat{\mathbf{V}}\tilde{\mathbf{v}}), \tag{9}$$

where

$$\tilde{\mathbf{U}} = (\mathbf{V}^{\dagger}\mathbf{U})\left(\mathbf{P}^{\dagger}\mathbf{U}\right)^{-1}, \tag{10}$$

$$\hat{\mathbf{V}} = \mathbf{P}^{\dagger}\mathbf{V}, \tag{11}$$

and as before, $\tilde{\mathbf{f}} = \mathbf{V}^{\dagger}\mathbf{f}$ and $\tilde{\mathbf{A}} = \mathbf{V}^{\dagger}\mathbf{A}\mathbf{V}$. At this point, it only remains to provide an algorithm to select the indices of the l interpolation points $\mathbf{p}$ and, consequently, the row-selector matrix $\mathbf{P}$. To this end, the procedure given in [1] and reproduced in Algorithm 1 is followed: given a basis $\mathbf{U}$, a greedy algorithm successively chooses the maximum of the absolute value of the current residual approximation to the nonlinear basis. In particular, it is shown [1, Lemma 3.2] that the row-selector matrix $\mathbf{P}$ obtained following this procedure always lead to matrices $(\mathbf{P}^{\dagger}\mathbf{U})$, equation (10), that are non-singular. The interested reader is referred to [1] for further details.

Algorithm 1 DEIM algorithm (adapted from [1]).

```
function DEIM($\mathbf{U}$)
Require: rank($\mathbf{U}$) $= l$
    $\mathbf{p}, \mathbf{P} \leftarrow \mathbf{0}_l, \mathbf{0}_{m\times l}$
    $\mathbf{p}_1 \leftarrow \arg\max |\mathbf{U}(:,1)|$
    $\mathbf{P}(\mathbf{p}_1, 1) \leftarrow 1$
    for $j \leftarrow 2, l$ do
        $\tilde{\mathbf{c}} \leftarrow \left(\mathbf{P}(:, :j-1)^{\dagger}\mathbf{U}(:, :j-1)\right)^{-1}\left(\mathbf{P}(:, :j-1)^{\dagger}\mathbf{U}(:, j)\right)$
        $\hat{\mathbf{r}} \leftarrow \mathbf{U}(:, j) - \mathbf{U}(:, :j-1)\tilde{\mathbf{c}}$
        $\mathbf{p}_j \leftarrow \arg\max |\hat{\mathbf{r}}|$
        $\mathbf{P}(\mathbf{p}_j, j) \leftarrow 1$
    end for
end function
```

2.3. Demonstration on the nonlinear Ginzburg–Landau equation

To demonstrate the design of reduced-order models using the above POD–DEIM algorithm, we consider the nonlinear Ginzburg–Landau equation and construct an associated reduced-order model. A range of alternative techniques for reducing this equation have been reported in Ilak et al. (2010) [13]. The nonlinear Ginzburg–Landau equation reads

$$\frac{\partial\phi}{\partial t} + \nu\frac{\partial\phi}{\partial x} = \gamma\frac{\partial^2\phi}{\partial x^2} + \mu\phi - a\left|\phi\right|^2\phi, \tag{12}$$

defined for $(t, x) \in [0, t_f] \times [x_0, x_f]$, and supplemented by the boundary conditions

$$\phi(x_0, t) = 0 \qquad \text{and} \qquad \frac{\partial\phi}{\partial x}(x_f, t) = 0. \tag{13}$$

The parameters ν, γ, μ and a are taken as

$$\nu = U + 2\mathrm{i}c_u, \quad \gamma = 1 + \mathrm{i}c_d, \qquad \mu = (\mu_0 - c_u^2) + \mu_2\frac{x^2}{2}, \tag{14}$$

and, following the study of Ilak et al. (2010) [13], we select the governing parameters in the above equation as $U = 2$, $c_u = 0.2$, $\mu_0 = 0.41$, $\mu_2 = 0.01$, $c_d = -1$, and $a = 0.1$.

The computational domain covers the x-axis from $x_0 = -50$ to $x_f = 50$, discretized in 512 equispaced mesh points. Equation (12) is then reformulated into the form given in equation (1) by setting $\mathbf{f} = 0$, $\mathbf{A}\phi = -\nu\partial_x\phi + \gamma\partial_{xx}\phi + \mu\phi$ and $\mathbf{F}'(\phi) = -a|\phi|^2\phi$, and it is propagated in time. Note that with these definitions, $\mathbf{F}'(\phi)$ is a component-wise function.

The nonlinear Ginzburg–Landau equation with the parameters selected above exhibits a saturated limit cycle, as visualized in Fig. 1 by the amplitude ϕ. From a given initial condition, the equation has been evolved over $T = 200$ time units, and snapshots have been gathered every $\Delta = 0.8$ time units, resulting in a sequence of $n = 250$ snapshots. Two POD bases are then computed using the snapshot method: a first POD basis $\mathbf{V}$ from the snapshot sequence $\{\mathbf{v}(t_i)\}_{i=1}^{n}$, and a second POD basis $\mathbf{U}$ from the nonlinear snapshots $\left\{\mathbf{F}'(\mathbf{v}(t_i))\right\}_{i=1}^{n}$. In Fig. 2(a), we display, for each basis, the energy percentage that is neglected by considering the first k modes, i.e.

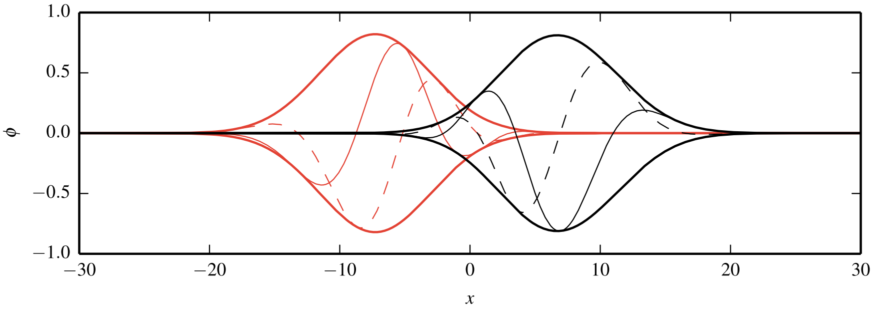

Fig. 1. Limit cycle in the nonlinear complex Ginzburg–Landau equation showing the real (thin line) and imaginary parts (dashed line) and magnitude (thick line) of the initial condition (red curves) and the long-time behavior (black curves). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

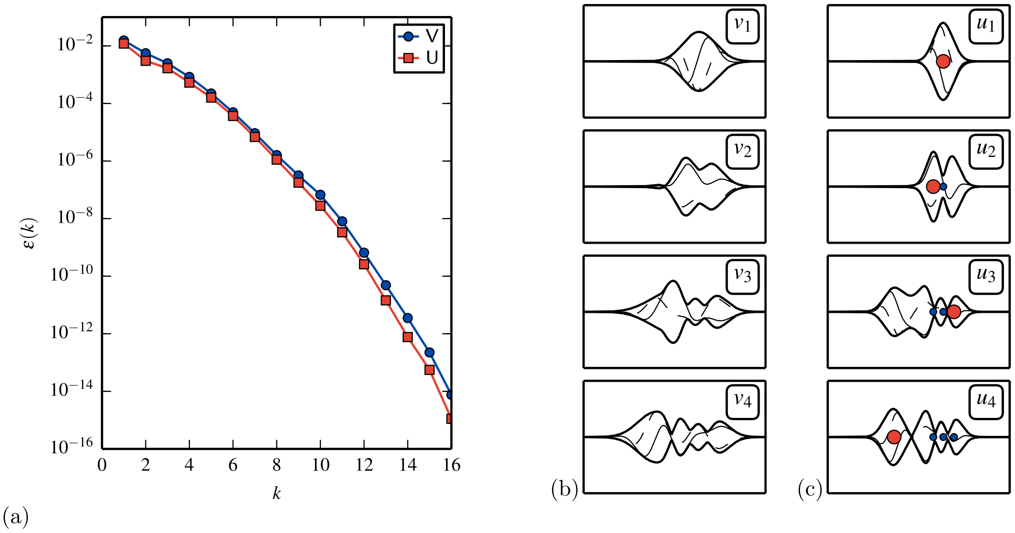

Fig. 2. (a) Energy percentage of the modes neglected for bases **V** and **U** (see text for details), (b) leading modes (first four) of the snapshot basis **V** and (c) leading modes (first four) of the nonlinear snapshot basis **U** together with the corresponding interpolation points. In (c), the red dots indicate the incremental choice of the interpolation points using Algorithm 1, while blue dots indicate the previously chosen interpolation points. The thick, thin and dashed lines represent, respectively, the magnitude, the real part and the imaginary part of each vector basis. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

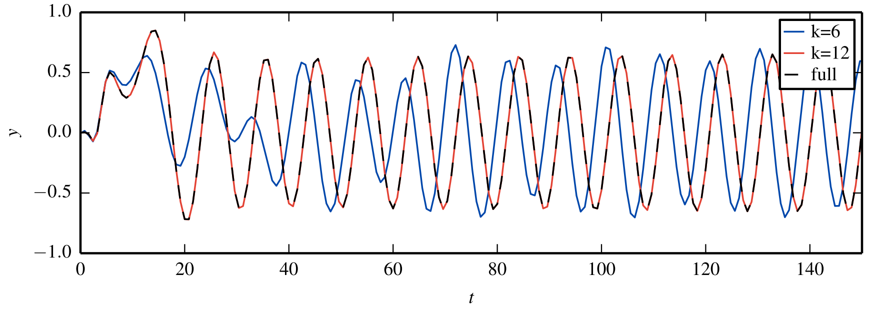

Fig. 3. Comparison between the original system and the reduced-order model for the nonlinear Ginzburg–Landau equations showing the real part of the amplitude at $x \approx 8.5$.

$$\varepsilon(k) = 1 - \frac{\sum_{i=k+1}^{n} \sigma_i^2}{\sum_{i=1}^{n} \sigma_i^2}, \tag{15}$$

where σ_i denotes the i-th singular value associated with each vector basis. We show in Figs. 2(b)–(c) the spatial structures of the bases **V** and **U**, respectively. In the case of the nonlinear basis **U**, we also indicate the location of the interpolation points obtained by the DEIM-algorithm (Algorithm 1). In Fig. 3, we compare the temporal evolution of the original system and reduced-order model for $k = 6$ and $k = 12$ modes; we display the amplitude ϕ at $x \approx 8.5$. It is worth mentioning that, in this case, the same number of modes has been considered for the bases $\mathbf{V}$ and $\mathbf{U}$. While differences are certainly noticeable for $k = 6$, an excellent match between the full and the reduced system is achieved for $k = 12$, in terms of capturing the transient behavior as well as the amplitude of the oscillations, frequency and phase in the long-time regime. It can easily be verified (not shown here) that, as the number of modes is increased, the relative error between the reduced and original system decreases, until it reaches about 10^{-6}, where it ultimately saturates.

3. Application to the Euler equations

Advancing to more complex governing equations, we discuss in this section the application of the Discrete Empirical Interpolation Method to the compressible Euler equations. Even though only the two-dimensional case is considered, the extension to the three-dimensional counterpart is straightforward.

We consider the Euler equations for an ideal gas written in terms of pressure p, entropy s and velocity $\mathbf{u}$ in nondimensional form. The reference values for the non-dimensional flow variables are given by a reference state, i.e. pressure p_∞, velocity U_∞ and density ρ_∞; on the other hand, the entropy s, spatial coordinates $\mathbf{x}$ and time t have been non-dimensionalized by the gas constant r, a characteristic length L and characteristic time L/U_∞, respectively. These equations read

$$\frac{\partial p}{\partial t} + \mathbf{u} \cdot \nabla p = -\gamma p \nabla \cdot \mathbf{u}, \tag{16a}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\gamma \mathrm{M}^2} \frac{1}{\rho} \nabla p, \tag{16b}$$

$$\frac{\partial s}{\partial t} + \mathbf{u} \cdot \nabla s = 0, \tag{16c}$$

where M stands for the Mach number, ρ for the density, and γ is the heat capacity ratio. This system of equations is augmented by the non-dimensional equation for the entropy $s = (\gamma - 1) \log(p/\rho^\gamma)$. Once equations (16) are discretized in space and suitable boundary conditions are specified, we arrive at a system of ordinary differential equations of the form

$$\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = \mathbf{F}(\mathbf{v}), \tag{17}$$

where $\mathbf{v}$ is the composite column vector containing the state variables at every grid point. More explicitly, we write

$$\mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_i \\ \vdots \\ \mathbf{v}_p \end{bmatrix}, \tag{18}$$

where p is the number of grid points, and $\mathbf{v}_i$ refers to the vector containing the state variables at the i-th grid point, i.e., $\mathbf{v}_i = \begin{bmatrix} \mathrm{p} & \mathrm{s} & \mathbf{u}^T \end{bmatrix}_i^T$, $i \in \{1, \ldots, p\}$. The number of state variables per grid point is $q = 2 + d$ where $d = 2$ in two-dimensions and $d = 3$ in three dimensions. Consequently, the length of the vector $\mathbf{v}$ is $m = pq$. In the following, our aim is to rewrite equation (17) in a convenient form for the application of the Discrete Empirical Interpolation Method.

3.1. Split between linear and nonlinear terms

In order to apply the reduced-order modeling approach given in section 2, equation (17) must be recast to explicitly separate linear and nonlinear terms. We start by decomposing the state vector $\mathbf{v}$ into a reference state $\bar{\mathbf{v}}$, known beforehand, plus a disturbance $\mathbf{v}'$. In the remaining of the paper, we will consider reference states $\bar{\mathbf{v}}$ that are independent of time. Depending on the application, the field $\bar{\mathbf{v}}$ can be a time-averaged flow field or a steady solution of the governing equations. The Euler equations in semi-discretized form, equation (17), then read

$$\frac{\mathrm{d}\mathbf{v}'}{\mathrm{d}t} = \mathbf{F}(\bar{\mathbf{v}} + \mathbf{v}'). \tag{19}$$

The nonlinear function $\mathbf{F}(\bar{\mathbf{v}} + \mathbf{v}')$ is now rewritten to extract explicitly its first two Taylor-expansion terms about $\bar{\mathbf{v}}$. We have

$$\mathbf{F}(\bar{\mathbf{v}} + \mathbf{v}') = \mathbf{F}(\bar{\mathbf{v}}) + \mathbf{A}(\bar{\mathbf{v}})\mathbf{v}' + \ldots, \tag{20}$$

where $\mathbf{A}(\bar{\mathbf{v}})$ is the Jacobian matrix of the function $\mathbf{F}(\mathbf{v})$ evaluated at the reference state $\bar{\mathbf{v}}$, i.e.,

$$\mathbf{A}(\bar{\mathbf{v}}) = \left. \frac{\partial \mathbf{F}}{\partial \mathbf{v}} \right|_{\bar{\mathbf{v}}}. \tag{21}$$

Higher-order terms are recast into a nonlinear function $\mathbf{F}'(\bar{\mathbf{v}}; \mathbf{v}')$ as defined in equation (2).

To further develop this idea, we illustrate the split between linear and nonlinear terms by the contribution of the advection terms $-\mathbf{u}\cdot\nabla\mathbf{u}$, which appear isolated in equation (16), to the right-hand side nonlinear function $\mathbf{F}(\mathbf{v})$, constant term $\mathbf{F}(\bar{\mathbf{v}})$, linear operator $\mathbf{A}(\bar{\mathbf{v}})$ and nonlinear terms $\mathbf{F}'(\bar{\mathbf{v}};\bar{\mathbf{v}})$. We obtain

$$\mathbf{F}(\bar{\mathbf{v}}+\mathbf{v}') = -(\bar{\mathbf{u}}+\mathbf{u}')\cdot\nabla(\bar{\mathbf{u}}+\mathbf{u}'), \tag{22}$$

$$\mathbf{A}(\bar{\mathbf{v}})\mathbf{v}' = -(\bar{\mathbf{u}}\cdot\nabla\mathbf{u}'+\mathbf{u}'\cdot\nabla\bar{\mathbf{u}}), \tag{23}$$

$$\mathbf{F}(\bar{\mathbf{v}}) = -\bar{\mathbf{u}}\cdot\nabla\bar{\mathbf{u}}, \tag{24}$$

and

$$\begin{aligned}\mathbf{F}'(\bar{\mathbf{v}};\mathbf{v}') &= -(\bar{\mathbf{u}}+\mathbf{u}')\cdot\nabla(\bar{\mathbf{u}}+\mathbf{u}')+\bar{\mathbf{u}}\cdot\nabla\bar{\mathbf{u}}+\bar{\mathbf{u}}\cdot\nabla\mathbf{u}'+\mathbf{u}'\cdot\nabla\bar{\mathbf{u}}\\ &= -\mathbf{u}'\cdot\nabla\mathbf{u}'.\end{aligned} \tag{25}$$

In the next section, the contribution of the advection terms will be used to illustrate a modular approach to the DEIM approximation of the nonlinear terms.

At this point, the governing equations are recast into the form given in equation (3). By approximating the disturbances into the previously computed POD basis $\mathbf{V}$ and projecting the governing equations onto this basis we arrive at an equation of the form given in equation (4). In what follows, we focus on the low-rank representation of $\mathbf{V}^\dagger\mathbf{F}'(\bar{\mathbf{v}};\mathbf{V}\mathbf{v}')$.

3.2. Approximation of non-component-wise nonlinear terms

The approximation for the nonlinear term given by the Discrete Empirical Interpolation Method reads

$$\begin{aligned}\mathbf{V}^\dagger\mathbf{F}'(\bar{\mathbf{v}};\mathbf{V}\tilde{\mathbf{v}}) &\approx \tilde{\mathbf{U}}\mathbf{P}^\dagger\mathbf{F}'(\bar{\mathbf{v}};\mathbf{V}\tilde{\mathbf{v}})\\ &= \tilde{\mathbf{U}}\left[\mathbf{P}^\dagger\mathbf{F}(\bar{\mathbf{v}}+\mathbf{V}\tilde{\mathbf{v}})-\hat{\mathbf{A}}\tilde{\mathbf{v}}-\hat{\mathbf{f}}\right],\end{aligned} \tag{26}$$

where $\hat{\mathbf{A}}=\mathbf{P}^\dagger\mathbf{A}(\bar{\mathbf{v}})\mathbf{V}$, $\hat{\mathbf{f}}=\mathbf{P}^\dagger\mathbf{F}(\bar{\mathbf{v}})$, and, as before, $\tilde{\mathbf{U}}=(\mathbf{V}^\dagger\mathbf{U})(\mathbf{P}^\dagger\mathbf{U})^{-1}$. Note that since $\mathbf{v}_i'$ is a vector field with q components, the row-selector matrix selects the corresponding state variable at the given grid point. It is important to point out at this stage that the operation count in evaluating $\mathbf{F}'(\bar{\mathbf{v}};\mathbf{v}')$ cannot be reduced because, in general, $\mathbf{F}(\mathbf{v})$ is not a point-wise function, and, as a consequence, we are not justified in applying the row-selector matrix $\mathbf{P}^\dagger$ to its argument. It is equally important to stress that the assumption of a component-wise nonlinearity has been instrumental in reducing the evaluation of the nonlinear terms from every grid point to the DEIM-interpolation points — a crucial step in arriving at a nonlinear, reduced-order model. For applications to the Euler and Navier–Stokes equations this assumption no longer holds: the type of nonlinear terms contained in the discretized equations is, in general, not component-wise because of the spatial derivatives, and the procedure outlined above does not directly apply.

In [1], the authors provide an extension to the DEIM technique that permits the consideration of general non-component-wise nonlinearities. Following this procedure, the authors then introduce sparse matrices to take into account the dependence on the different grid points. However, such an approach relies on the sparsity of the spatial differentiation operators. In most high-order finite-difference approximations, though, the corresponding differentiation matrices either have a large bandwidth or are dense, and this latter approach becomes impractical as the matrices typically exceed the available storage capacity.

Similar drawbacks are also encountered in the evaluation of linear, direct and adjoint operators in numerical solvers that feature high-order spatial discretizations [2], where it is shown that these limitations can be overcome by recasting nonlinear terms as component-wise functions. As before, we illustrate this procedure on the contribution of the advection terms $-\mathbf{u}\cdot\nabla\mathbf{u}$ to $\mathbf{F}(\mathbf{v})$:

$$f_u(u,v) = -(u\partial_x u+v\partial_y u), \tag{27a}$$

$$f_v(u,v) = -(u\partial_x v+v\partial_y v). \tag{27b}$$

Upon spatial discretization, the advection terms evaluated at the i-th grid point read

$$\mathbf{F}_i(\mathbf{u}) = -\begin{bmatrix}\mathbf{D}_{x,ij}\mathbf{u}_j & \mathbf{D}_{y,ij}\mathbf{u}_j\end{bmatrix}\mathbf{u}_i, \tag{28}$$

where $\mathbf{u}_i=\begin{bmatrix}u & v\end{bmatrix}_i^T$, and the matrices $\mathbf{D}_x$ and $\mathbf{D}_y$ represent the discrete spatial differentiation operators of a given vector field along the x- and y-direction, respectively. The question that has to be addressed then is how to reduce the operation count in the evaluation of the DEIM approximation of equation (28). In order to restate equation (28) as a component-wise function, such that the operation count can be reduced, we introduce auxiliary variables $\mathbf{r}$ and $\mathbf{s}$ defined as $\mathbf{r}=\mathbf{D}_x\mathbf{u}$ and $\mathbf{s}=\mathbf{D}_y\mathbf{u}$, and then define the point-wise function $\hat{\mathbf{F}}_i(\mathbf{u},\mathbf{r},\mathbf{s})=\begin{bmatrix}\mathbf{r}_i & \mathbf{s}_i\end{bmatrix}\mathbf{u}_i$ in terms of these auxiliary variables. Algorithmically, we compute the nonlinear terms in the following two steps

$$\mathbf{r},\mathbf{s} \leftarrow \mathbf{D}_x\mathbf{u},\mathbf{D}_y\mathbf{u}, \tag{29a}$$

$$\hat{\mathbf{F}}_i(\mathbf{u},\mathbf{r},\mathbf{s}) = \begin{bmatrix}\mathbf{r}_i & \mathbf{s}_i\end{bmatrix}\mathbf{u}_i. \tag{29b}$$

Even though $\mathbf{F}_i(\mathbf{u})$ is equivalent to $\hat{\mathbf{F}}_i(\mathbf{u}, \mathbf{D}_x\mathbf{u}, \mathbf{D}_y\mathbf{u})$, the latter form is preferable because $\hat{\mathbf{F}}(\mathbf{v})$ is a point-wise function, and we are justified in applying the row-selector matrix $\mathbf{P}^\dagger$ to its arguments, i.e., $\mathbf{P}^\dagger\mathbf{F}(\mathbf{u}) = \hat{\mathbf{F}}(\mathbf{P}^\dagger\mathbf{u}, \mathbf{P}^\dagger\mathbf{D}_x\mathbf{u}, \mathbf{P}^\dagger\mathbf{D}_y\mathbf{u})$. In applying the row-selector matrix to the arguments of $\hat{\mathbf{F}}$, the row-selector matrix has to be modified so that it also selects the remaining state variables at a given grid point (other than the variable selected by the DEIM algorithm). When the row-selector matrix $\mathbf{P}$ is applied to the arguments, its dimension is $pq \times lq$. To keep the notation compact for the remainder of this article, we will simply use $\mathbf{P}$ for the row-selector matrix, and its appropriate dimension will be clearly deduced from the context.

A similar procedure allows for the restatement of the right-hand side of the Euler equations in terms of a component-wise nonlinear function. For the sake of argument, we first consider the function $\mathbf{F}(\mathbf{v})$ for the two-dimensional case:

$$f_p(p, s, u, v) = -(u\partial_x p + v\partial_y p) - \gamma p(\partial_x u + \partial_y v), \tag{30a}$$

$$f_u(p, s, u, v) = -(u\partial_x u + v\partial_y u) - \frac{1}{\gamma \mathrm{M}^2}\frac{\partial_x p}{\rho(p, s)}, \tag{30b}$$

$$f_v(p, s, u, v) = -(u\partial_x v + v\partial_y v) - \frac{1}{\gamma \mathrm{M}^2}\frac{\partial_y p}{\rho(p, s)}, \tag{30c}$$

$$f_s(p, s, u, v) = -(u\partial_x s + v\partial_y s). \tag{30d}$$

Second, we introduce auxiliary variables for the state vector, i.e., $\mathbf{r} = \mathbf{D}_x\mathbf{v}$ and $\mathbf{s} = \mathbf{D}_y\mathbf{v}$. Third, we recast $\mathbf{F}(\mathbf{v})$ as a component-wise function $\hat{\mathbf{F}}(\mathbf{v}, \mathbf{r}, \mathbf{s})$ by decoupling the evaluation of the spatial derivatives and the evaluation of nonlinearities, i.e., $\mathbf{F}(\mathbf{v}) = \hat{\mathbf{F}}(\mathbf{v}, \mathbf{D}_x\mathbf{v}, \mathbf{D}_y\mathbf{v})$, or using a more compact notation, $\mathbf{F}(\mathbf{v}) = \hat{\mathbf{F}}(\mathbf{D}_\xi\mathbf{v})$.

At this point, we can perform the DEIM approximation of the nonlinear terms in the Euler equations. Symbolically, we write

$$\begin{aligned} \mathbf{P}^\dagger\mathbf{F}(\bar{\mathbf{v}} + \mathbf{V}\tilde{\mathbf{v}}) &= \mathbf{P}^\dagger\hat{\mathbf{F}}(\mathbf{D}_\xi(\bar{\mathbf{v}} + \mathbf{V}\tilde{\mathbf{v}})), \\ &= \hat{\mathbf{F}}(\mathbf{P}^\dagger\mathbf{D}_\xi(\bar{\mathbf{v}} + \mathbf{V}\tilde{\mathbf{v}})). \end{aligned} \tag{31}$$

Introducing the vectors $\hat{\bar{\mathbf{v}}}_\xi = \mathbf{P}^\dagger\mathbf{D}_\xi\bar{\mathbf{v}}$ that contain the value of the reference field variables and their spatial derivatives at the interpolation points, and $\hat{\mathbf{V}}_\xi = \mathbf{P}^\dagger\mathbf{D}_\xi\mathbf{V}$ which is the matrix whose column vectors contain the value of the spatial derivatives of the basis $\mathbf{V}$ at the interpolation points, the above relation reads

$$\mathbf{P}^\dagger\mathbf{F}(\bar{\mathbf{v}} + \mathbf{V}\tilde{\mathbf{v}}) = \hat{\mathbf{F}}(\hat{\bar{\mathbf{v}}}_\xi + \hat{\mathbf{V}}_\xi\tilde{\mathbf{v}}). \tag{32}$$

Note that, ξ takes on the values $\xi \in \{0, x, y\}$ and $\mathbf{D}_0 = \mathbf{I}$. At this point the operation count has been reduced. On the one hand, the evaluation of the left-hand-side expression requires first to form the vector $\bar{\mathbf{v}} + \mathbf{V}\tilde{\mathbf{v}}$ and second to evaluate $\mathbf{F}(\cdot)$; the operation count is $O(\alpha(m) + mk)$, where $\alpha(m)$ is the evaluation cost of $\mathbf{F}(\cdot)$. On the other hand, the evaluation of the right-hand-side requires first to form $\hat{\bar{\mathbf{v}}}_\xi + \hat{\mathbf{V}}_\xi\tilde{\mathbf{v}}$ and then to evaluate $\hat{\mathbf{F}}(\cdot)$; in this case, the operation count is reduced to $O(\alpha(l) + lk)$, with $k, l \ll m$, and similarly, $\alpha(k)$ is the evaluation cost of $\hat{\mathbf{F}}(\cdot)$.

3.3. Practical implementation of the model reduction

In a final step, the nonlinear reduced-order model is obtained by inserting equations (32) and (26) into equation (4) which yields

$$\frac{\mathrm{d}\tilde{\mathbf{v}}}{\mathrm{d}t} = \tilde{\mathbf{f}} + \tilde{\mathbf{A}}\tilde{\mathbf{v}} + \tilde{\mathbf{U}}\left[\hat{\mathbf{F}}(\hat{\bar{\mathbf{v}}}_\xi + \hat{\mathbf{V}}_\xi\tilde{\mathbf{v}}) - \hat{\mathbf{A}}\tilde{\mathbf{v}} - \hat{\mathbf{f}}\right]. \tag{33}$$

The computational cost of the right-hand-side of the above reduced-order model is $O(\alpha(l) + k^2 + (3 + d)kl)$, with $k, l \ll m$, $d = 2$ for two dimensions and $d = 3$ for three dimensions.

Normally, the various nonlinear functions introduced above and their Jacobians are not readily available in a nonlinear numerical solver; consequently, the implementation of the reduced-order model given in equation (33) seems tedious and unwieldy at first sight. In this section we address the practical implementation, reusing functionalities already present in common flow solvers with minimal code development efforts.

First, the POD bases $\mathbf{V}$ and $\mathbf{U}$ need to be constructed. To this end, we start with a given reference field $\bar{\mathbf{v}}$ and an ensemble of snapshots for the flow field $\{\mathbf{v}(t_i)\}_{i=1}^n$. The set of snapshots for the perturbation $\mathbf{v}'$ — which we will use for the basis $\mathbf{V}$ — can be easily constructed noting that

$$\left\{\mathbf{v}'(t_i)\right\}_{i=1}^n = \{\mathbf{v}(t_i) - \bar{\mathbf{v}}\}_{i=1}^n. \tag{34}$$

On the other hand, the construction of a set of snapshots for the nonlinear function $\left\{\mathbf{F}'(\bar{\mathbf{v}}; \mathbf{v}'(t_i))\right\}_{i=1}^n$ is more involved: in typical flow solvers, the nonlinear function $\mathbf{F}'(\bar{\mathbf{v}}; \mathbf{v}')$ is commonly not implemented. Instead, we opt for the evaluation based on the definition given in equation (2). Therefore, the set of snapshots from which the nonlinear basis $\mathbf{U}$ arises reads

$$\left\{\mathbf{F}'(\bar{\mathbf{v}}; \mathbf{v}'(t_i))\right\}_{i=1}^n = \left\{\mathbf{F}(\bar{\mathbf{v}} + \mathbf{v}'(t_i)) - \mathbf{A}(\bar{\mathbf{v}})\mathbf{v}'(t_i) - \mathbf{F}(\bar{\mathbf{v}})\right\}_{i=1}^n. \tag{35}$$

Table 1
Nonlinear reduced-order model for the Euler equations: summary table. n is the number of snapshots, $m = pq$, is the length of the state vector, where p is the number of grid points, $q = 2 + d$ is the number of flow variables per grid point, $d = 2$ for two-dimensional flows and $d = 3$ for three-dimensional flows. The dimension of the basis $\mathbf{V}$ is k and the dimension of the basis $\mathbf{U}$ is l.

Vector/matrix	Number	Size	Description
$\bar{\mathbf{v}}$	1	$pq \times 1$	Reference field
$\{\mathbf{v}'(t_i)\}_{i=1}^{n}$	n	$pq \times 1$	Set of snapshots for the disturbances $\mathbf{v}' = \mathbf{v} - \bar{\mathbf{v}}$
$\mathbf{V}$	1	$pq \times k$	POD-based vector basis for the disturbances $\mathbf{v}'$
$\tilde{\mathbf{f}} = \mathbf{V}^{\dagger}\mathbf{F}(\bar{\mathbf{v}})$	1	$k \times 1$	Projection of the constant term $\mathbf{F}(\bar{\mathbf{v}})$ onto basis $\mathbf{V}$
$\tilde{\mathbf{A}} = \mathbf{V}^{\dagger}\mathbf{A}(\bar{\mathbf{v}})\mathbf{V}$	1	$k \times k$	Reduced-order linear dynamics
$\{\mathbf{F}'(\bar{\mathbf{v}}; \mathbf{v}'(t_i))\}_{i=1}^{n}$	n	$pq \times 1$	Set of snapshots for the nonlinear term $\mathbf{F}'(\bar{\mathbf{v}}; \mathbf{v}')$
$\mathbf{U}$	1	$pq \times l$	POD-based vector basis for the nonlinear term $\mathbf{F}'(\bar{\mathbf{v}}; \mathbf{v}')$
$\mathbf{P}$	1	$pq \times l$	Row-selector matrix
$\tilde{\mathbf{U}} = (\mathbf{V}^{\dagger}\mathbf{U})(\mathbf{P}^{\dagger}\mathbf{U})^{-1}$	1	$k \times l$	Interpolation matrix
$\hat{\bar{\mathbf{v}}}_{\xi} = \mathbf{P}^{\dagger}\mathbf{D}_{\xi}\bar{\mathbf{v}}$	$d + 1$	$lq \times 1$	Spatial derivatives vector $\bar{\mathbf{v}}$ at the interpolation points
$\hat{\mathbf{V}}_{\xi} = \mathbf{P}^{\dagger}\mathbf{D}_{\xi}\mathbf{V}$	$d + 1$	$lq \times k$	Spatial derivatives of the basis $\mathbf{V}$ at the interpolation points
$\hat{\mathbf{A}} = \mathbf{P}^{\dagger}\mathbf{A}(\bar{\mathbf{v}})\mathbf{V}$	1	$l \times k$	Linear terms at the interpolation points
$\hat{\mathbf{f}} = \mathbf{P}^{\dagger}\mathbf{F}(\bar{\mathbf{v}})$	1	$l \times 1$	Constant term $\mathbf{F}(\bar{\mathbf{v}})$ at the interpolation points

Algorithm 2 Nonlinear model-order reduction of the Euler equations using POD–DEIM.

```
Require: Reference field $\bar{\mathbf{v}}$, snapshots $\{\mathbf{v}(t_i)\}_{i=1}^{n}$, number of POD modes $k$ and $l$.
Generate snapshot sequence for the perturbations $\{\mathbf{v}'(t_i)\}_{i=1}^{n}$.
Construct $\mathbf{V}$ from the leading $k$ modes of the POD decomposition of $\{\mathbf{v}'(t_i)\}_{i=1}^{n}$.
Construct matrix $\tilde{\mathbf{A}}$ and vector $\tilde{\mathbf{f}}$.
Generate snapshot sequence for the nonlinearities $\{\mathbf{F}'(\bar{\mathbf{v}}; \mathbf{v}'(t_i))\}_{i=1}^{n}$ (equation (35)).
Construct $\mathbf{U}$ from the leading $l$ modes of the POD decomposition of $\{\mathbf{F}'(\bar{\mathbf{v}}; \mathbf{v}'(t_i))\}_{i=1}^{n}$.
Obtain interpolation point grid indices $\mathbf{p}_j$ and row selector matrix $\mathbf{P}$ (Algorithm 1).
Construct matrices $\tilde{\mathbf{U}}$, $\hat{\mathbf{V}}_{\xi}$, $\hat{\mathbf{A}}$, and vectors $\hat{\bar{\mathbf{v}}}_{\xi}$, and $\hat{\mathbf{f}}$.
```

In the above, it has been assumed that the matrix–vector product $\mathbf{A}(\bar{\mathbf{v}})\mathbf{v}'$ is available; however, depending on the specifics of the numerical code, this might not be the case. Nevertheless, $\mathbf{A}(\bar{\mathbf{v}})\mathbf{v}'$ can still be evaluated by quasi-linearization using a first-order approximation

$$\mathbf{A}(\bar{\mathbf{v}})\mathbf{v}' \approx \frac{\mathbf{F}(\bar{\mathbf{v}} + \epsilon\mathbf{v}') - \mathbf{F}(\bar{\mathbf{v}})}{\epsilon}. \tag{36}$$

The cost of evaluating $\mathbf{A}(\bar{\mathbf{v}})\mathbf{v}'$ is typically $O(m)$, where m is the number of flow variables. The reader is referred to [14] for practical guidelines for choosing the small parameter ϵ. Once a suitable inner product is given, the bases $\mathbf{V}$ and $\mathbf{U}$ are then constructed from the first k and l leading modes, respectively, of the POD decomposition of $\{\mathbf{v}'(t_i)\}_{i=1}^{n}$ and $\{\mathbf{F}'(\bar{\mathbf{v}}; \mathbf{v}'(t_i))\}_{i=1}^{n}$.

Provided that the basis $\mathbf{U}$ has been computed, the DEIM algorithm given in Algorithm 1 is then employed to select the l interpolation points, and thus the row-selector $m \times l$ matrix $\mathbf{P}$. Note that by construction the matrix $(\mathbf{P}^{\dagger}\mathbf{U})$ is non-singular. The matrices $\tilde{\mathbf{A}}$, $\tilde{\mathbf{U}}$, $\hat{\mathbf{V}}_{\xi}$, $\hat{\mathbf{A}}$ and the vectors $\tilde{\mathbf{f}}$, and $\hat{\bar{\mathbf{v}}}_{\xi}$ then follow directly from their definitions. A summary of the procedure described above and the different matrices that have been introduced is presented in Table 1 and Algorithm 2, respectively.

4. Reduced-order modeling for the Navier–Stokes equations using POD–DEIM

In a further step in complexity, we extend the reduced-order modeling procedure described in section 3 to the compressible Navier–Stokes equations. The application of the Discrete Empirical Interpolation Method to these equations introduces an additional complexity: the highest-order derivative is increased by one order, and if turbulence models or temperature-dependent viscosity or heat conductivity is introduced, additional nonlinear functions are introduced. In what follows, we focus on our formulation of the compressible Navier–Stokes equations in terms of pressure, entropy and velocity variables; nonetheless, the procedure below can be readily adapted to different formulations.

The compressible Navier–Stokes for an ideal gas, appropriately non-dimensionalized, read

$$\frac{1}{p}\left(\frac{\partial p}{\partial t} + \mathbf{u}\cdot\nabla p\right) = -\gamma\nabla\cdot\mathbf{u} + (\gamma - 1)\left(\frac{\partial s}{\partial t} + \mathbf{u}\cdot\nabla s\right), \tag{37a}$$

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = -\frac{1}{\gamma \mathrm{M}^2}\nabla p + \frac{1}{\mathrm{Re}}\nabla\cdot\boldsymbol{\tau}, \tag{37b}$$

$$p\left(\frac{\partial s}{\partial t} + \mathbf{u}\cdot\nabla s\right) = \frac{\gamma \mathrm{M}^2}{\mathrm{Re}}\boldsymbol{\tau} : \nabla\mathbf{u} - \frac{\gamma}{(\gamma - 1)\,\mathrm{Pr}\mathrm{Re}}\nabla\cdot\mathbf{q}, \tag{37c}$$

with the constitutive relations for the heat flux $\mathbf{q}$ and the viscous stress tensor $\boldsymbol{\tau}$ given by

$$\mathbf{q} = -\kappa \nabla T, \tag{38a}$$

$$\boldsymbol{\tau} = \mu \left(\nabla \mathbf{u} + \nabla \mathbf{u}^t\right) + \left(\mu_v - \frac{2}{3}\mu\right)(\nabla \cdot \mathbf{u})\,\mathbf{I}, \tag{38b}$$

together with the non-dimensional state equation $p = \rho T$, and the non-dimensional equation for the entropy $(\gamma - 1)s = \log(p/\rho^\gamma)$. The reference value for the temperature is $T_\infty = p_\infty/(r\rho_\infty)$; the reference values for the remaining flow variables, spatial coordinates and time are given in section 3. These equations can be easily recast into the form in equation (1). In the above, Re denotes the Reynolds number, Pr is the Prandtl number, κ represents the heat conductivity, μ is the dynamic viscosity and μ_v stands for the volumetric viscosity coefficient written in non-dimensional form. Upon spatial discretization, we introduce the column vectors $\mathbf{v}_i$ and $\mathbf{w}_i$ at every grid point given in the two-dimensional case by

$$\mathbf{v}_i = \begin{bmatrix} p & s & u & v \end{bmatrix}_i^T, \tag{39a}$$

$$\mathbf{w}_i = \begin{bmatrix} \tau_{xx} & \tau_{yy} & \tau_{xy} & q_x & q_y \end{bmatrix}_i^T, \tag{39b}$$

and define the state vectors $\mathbf{v}$ and $\mathbf{w}$ defined as the composite vectors containing the values of $\mathbf{v}_i$ and $\mathbf{w}_i$, respectively, at every grid point. Recalling that the number of grid points is p, the length of the column vector $\mathbf{v}$ is $m_v = pq_v$, with $q_v = 4$ (in the three-dimensional case $q_v = 5$), and the length of the column vector $\mathbf{w}$ is $m_w = pq_w$ with $q_w = 5$ (in the three-dimensional case $q_w = 9$). With these considerations, the governing equations can be recast into the following form

$$\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = \mathbf{F}(\mathbf{v}, \mathbf{w}), \tag{40a}$$

$$\mathbf{w} = \mathbf{G}(\mathbf{v}). \tag{40b}$$

It should be emphasized that the function $\mathbf{G}(\mathbf{v})$ is nonlinear: for instance, although the heat flux $\mathbf{q} = -\kappa \nabla T(p, s)$ is linear in the temperature variable T, it is nonlinear in the pressure p and entropy s variables, which renders the heat flux in our implementation a nonlinear function. Similarly, if a temperature-dependent viscosity $\mu(T)$ is specified, such as given by Sutherland's law, the viscous stress tensor also becomes a nonlinear function of the state variables.

4.1. Approximation of the inviscid terms

Analogously to the developments presented in section 3, we start by considering the linear and nonlinear dynamics separately by introducing a reference field $\bar{\mathbf{v}}$ plus disturbances $\mathbf{v}'$. For convenience, such a decomposition is also introduced for the intermediate variable: we define $\bar{\mathbf{w}} = \mathbf{G}(\bar{\mathbf{v}})$ and set $\mathbf{w} = \bar{\mathbf{w}} + \mathbf{w}'$. Hence, equation (40a) reads

$$\frac{\mathrm{d}\mathbf{v}'}{\mathrm{d}t} = \mathbf{F}(\bar{\mathbf{v}}) + \mathbf{A}(\bar{\mathbf{v}}, \bar{\mathbf{w}})\mathbf{v}' + \mathbf{F}'(\bar{\mathbf{v}}, \bar{\mathbf{w}}; \mathbf{v}', \mathbf{w}'), \tag{41a}$$

$$\mathbf{w}' = \mathbf{G}'(\bar{\mathbf{v}}; \mathbf{v}'), \tag{41b}$$

where

$$\mathbf{A}(\bar{\mathbf{v}}, \bar{\mathbf{w}}) = \left.\frac{\partial \mathbf{F}}{\partial \mathbf{v}}\right|_{\bar{\mathbf{v}},\bar{\mathbf{w}}} + \left.\frac{\partial \mathbf{F}}{\partial \mathbf{w}}\right|_{\bar{\mathbf{v}},\bar{\mathbf{w}}} \left.\frac{\partial \mathbf{G}}{\partial \mathbf{v}}\right|_{\bar{\mathbf{v}}}, \tag{42}$$

$$\mathbf{F}'(\bar{\mathbf{v}}, \bar{\mathbf{w}}; \mathbf{v}', \mathbf{w}') = \mathbf{F}(\bar{\mathbf{v}} + \mathbf{v}', \bar{\mathbf{w}} + \mathbf{w}') - \mathbf{A}(\bar{\mathbf{v}}, \bar{\mathbf{w}})\mathbf{v}' - \mathbf{F}(\bar{\mathbf{v}}, \bar{\mathbf{w}}), \tag{43}$$

$$\mathbf{G}'(\bar{\mathbf{v}}; \mathbf{v}') = \mathbf{G}(\bar{\mathbf{v}} + \mathbf{v}') - \mathbf{G}(\bar{\mathbf{v}}). \tag{44}$$

Given an orthogonal basis $\mathbf{V}$, an equation for the temporal evolution of the approximation $\tilde{\mathbf{v}}$ is obtained by setting $\mathbf{v}' \approx \mathbf{V}\tilde{\mathbf{v}}$ in equation (41a), and left-multiplying by $\mathbf{V}^\dagger$. We obtain

$$\frac{\mathrm{d}\tilde{\mathbf{v}}}{\mathrm{d}t} = \tilde{\mathbf{f}} + \tilde{\mathbf{A}}\tilde{\mathbf{v}} + \mathbf{V}^\dagger \mathbf{F}'(\bar{\mathbf{v}}, \bar{\mathbf{w}}; \mathbf{V}\tilde{\mathbf{v}}, \mathbf{w}'), \tag{45}$$

where, as before, $\tilde{\mathbf{f}} = \mathbf{V}^\dagger \mathbf{F}(\bar{\mathbf{v}}, \bar{\mathbf{w}})$ and $\tilde{\mathbf{A}} = \mathbf{V}^\dagger \mathbf{A}(\bar{\mathbf{v}}, \bar{\mathbf{w}})\mathbf{V}$. Approximating the nonlinear terms in the above equation using basis $\mathbf{U}$ and DEIM, the nonlinear terms read

$$\begin{aligned}\mathbf{V}^\dagger \mathbf{F}'(\bar{\mathbf{v}}, \bar{\mathbf{w}}; \mathbf{V}\tilde{\mathbf{v}}, \mathbf{w}') &\approx (\mathbf{V}^\dagger \mathbf{U})(\mathbf{P}^\dagger \mathbf{U})^{-1} \mathbf{P}^\dagger \mathbf{F}'(\bar{\mathbf{v}}, \bar{\mathbf{w}}; \mathbf{V}\tilde{\mathbf{v}}, \mathbf{w}') \\ &= (\mathbf{V}^\dagger \mathbf{U})(\mathbf{P}^\dagger \mathbf{U})^{-1} \left[\mathbf{P}^\dagger \mathbf{F}(\bar{\mathbf{v}} + \mathbf{V}\tilde{\mathbf{v}}, \bar{\mathbf{w}} + \mathbf{w}') - \mathbf{P}^\dagger \mathbf{A}(\bar{\mathbf{v}}, \bar{\mathbf{w}})\mathbf{V}\tilde{\mathbf{v}} - \mathbf{P}^\dagger \mathbf{F}(\bar{\mathbf{v}}, \bar{\mathbf{w}})\right],\end{aligned} \tag{46}$$

and, defining $\tilde{\mathbf{U}} = (\mathbf{V}^\dagger \mathbf{U})(\mathbf{P}^\dagger \mathbf{U})^{-1}$, $\hat{\mathbf{A}} = \mathbf{P}^\dagger \mathbf{A}(\bar{\mathbf{v}}, \bar{\mathbf{w}})\mathbf{V}$ and $\hat{\mathbf{f}} = \mathbf{P}^\dagger \mathbf{F}(\bar{\mathbf{v}}, \bar{\mathbf{w}})$, we write

$$\mathbf{V}^\dagger \mathbf{F}'(\bar{\mathbf{v}}, \bar{\mathbf{w}}; \mathbf{V}\tilde{\mathbf{v}}, \mathbf{w}') \approx \tilde{\mathbf{U}} \left[\mathbf{P}^\dagger \mathbf{F}(\bar{\mathbf{v}} + \mathbf{V}\tilde{\mathbf{v}}, \bar{\mathbf{w}} + \mathbf{w}') - \hat{\mathbf{A}}\tilde{\mathbf{v}} - \hat{\mathbf{f}}\right]. \tag{47}$$

As before, we introduce auxiliary variables and recast function $\mathbf{F}(\mathbf{v},\mathbf{w})$ into a point-wise function such that

$$\mathbf{F}(\mathbf{v},\mathbf{w}) = \hat{\mathbf{F}}(\mathbf{D}_\xi \mathbf{v}, \mathbf{D}_\xi \mathbf{w}), \tag{48}$$

where $\mathbf{D}_\xi$ is the differentiation operator corresponding to $\mathbf{v}$ and $\mathbf{w}$, respectively. We point out that the dimensions of the differentiation operators $\mathbf{D}_\xi$ and the row-selector matrix $\mathbf{P}^\dagger$ differ when applied to $\mathbf{v}$ or $\mathbf{w}$, since the number of variables per grid point for the vectors $\mathbf{v}$ and $\mathbf{w}$ differ. Nevertheless, the same notation will be used independently of the dimension of the vector field on which they act, and their appropriate dimension will be deduced from the context. Therefore,

$$\begin{aligned}\mathbf{P}^\dagger\mathbf{F}(\bar{\mathbf{v}}+\mathbf{V}\tilde{\mathbf{v}}, \bar{\mathbf{w}}+\mathbf{w}') &= \mathbf{P}^\dagger\hat{\mathbf{F}}(\mathbf{D}_\xi(\bar{\mathbf{v}}+\mathbf{V}\mathbf{v}), \mathbf{D}_\xi(\bar{\mathbf{w}}+\mathbf{w}'))\\ &= \hat{\mathbf{F}}(\mathbf{P}^\dagger\mathbf{D}_\xi(\bar{\mathbf{v}}+\mathbf{V}\tilde{\mathbf{v}}), \mathbf{P}^\dagger\mathbf{D}_\xi(\bar{\mathbf{w}}+\mathbf{w}'))\end{aligned} \tag{49}$$

or, introducing $\hat{\bar{\mathbf{v}}}_\xi = \mathbf{P}^\dagger\mathbf{D}_\xi\bar{\mathbf{v}}$, $\hat{\mathbf{V}}_\xi = \mathbf{P}^\dagger\mathbf{D}_\xi\mathbf{V}$, $\hat{\bar{\mathbf{w}}}_\xi = \mathbf{P}^\dagger\mathbf{D}_\xi\bar{\mathbf{w}}$ and $\hat{\mathbf{w}}_\xi = \mathbf{P}^\dagger\mathbf{D}_\xi\mathbf{w}'$, we arrive at

$$\mathbf{P}^\dagger\mathbf{F}(\bar{\mathbf{v}}+\mathbf{V}\tilde{\mathbf{v}}, \bar{\mathbf{w}}+\mathbf{w}') = \hat{\mathbf{F}}(\hat{\bar{\mathbf{v}}}_\xi + \hat{\mathbf{V}}_\xi\tilde{\mathbf{v}}, \hat{\bar{\mathbf{w}}}_\xi + \hat{\mathbf{w}}_\xi). \tag{50}$$

4.2. Approximation of the viscous terms: additional bases for nonlinearities

We stress that the evaluation of $\hat{\mathbf{w}}_\xi$ in equation (50) still requires a large operation count: the evaluation of $\hat{\mathbf{w}}_\xi$ reads

$$\hat{\mathbf{w}}_\xi = \mathbf{P}^\dagger\mathbf{D}_\xi\mathbf{G}'(\bar{\mathbf{v}};\mathbf{V}\tilde{\mathbf{v}}) \tag{51}$$

$$= \mathbf{P}^\dagger\mathbf{D}_\xi\left(\mathbf{G}(\bar{\mathbf{v}}+\mathbf{V}\tilde{\mathbf{v}}) - \mathbf{G}(\bar{\mathbf{v}})\right). \tag{52}$$

The evaluation of $\mathbf{G}(\bar{\mathbf{v}}+\mathbf{V}\tilde{\mathbf{v}})$ carries a cost of the order of the number of variables in the original system.

In the following, to further reduce the operation count, we propose to approximate these terms by applying again the DEIM algorithm. First, an additional basis $\mathbf{W}$, obtained from the l_w leading modes of the POD decomposition of the snapshots $\left\{\mathbf{G}'(\bar{\mathbf{v}};\mathbf{v}'(t_i))\right\}_{i=1}^n$ is introduced. Second, Algorithm 1 is followed to obtain a set of interpolation points $\mathbf{q}_j$ and, consequently, a row-selector matrix $\mathbf{Q}$ to approximate $\mathbf{G}(\bar{\mathbf{v}}+\mathbf{V}\tilde{\mathbf{v}})$ in this basis. Hence

$$\begin{aligned}\hat{\mathbf{w}}_\xi &= \mathbf{P}^\dagger\mathbf{D}_\xi\left[\mathbf{G}(\bar{\mathbf{v}}+\mathbf{V}\tilde{\mathbf{v}}) - \mathbf{G}(\bar{\mathbf{v}})\right]\\ &\approx (\mathbf{P}^\dagger\mathbf{D}_\xi\mathbf{W})(\mathbf{Q}^\dagger\mathbf{W})^{-1}\left[\mathbf{Q}^\dagger\mathbf{G}(\bar{\mathbf{v}}+\mathbf{V}\tilde{\mathbf{v}}) - \mathbf{Q}^\dagger\mathbf{G}(\bar{\mathbf{v}})\right].\end{aligned} \tag{53}$$

In the following, we introduce $\hat{\mathbf{W}}_\xi = (\mathbf{P}^\dagger\mathbf{D}_\xi\mathbf{W})(\mathbf{Q}^\dagger\mathbf{W})^{-1}$, and $\check{\mathbf{g}} = \mathbf{Q}^\dagger\mathbf{G}(\bar{\mathbf{v}})$. It is noted that matrix $(\mathbf{Q}^\dagger\mathbf{W})$ is invertible by construction. As before, we recast $\mathbf{G}(\mathbf{v})$ as a component-wise function by introducing auxiliary variables for the spatial derivatives, i.e., $\mathbf{G}(\mathbf{v}) = \hat{\mathbf{G}}(\mathbf{D}_\eta\mathbf{v})$. We obtain

$$\hat{\mathbf{w}}_\xi = \hat{\mathbf{W}}_\xi\left[\hat{\mathbf{G}}(\check{\bar{\mathbf{v}}}_\eta + \check{\mathbf{V}}_\eta\tilde{\mathbf{v}})] - \check{\mathbf{g}}\right] \tag{54}$$

where we have used the abbreviations $\check{\bar{\mathbf{v}}}_\eta = \mathbf{Q}^\dagger\mathbf{D}_\eta\bar{\mathbf{v}}$, $\check{\mathbf{V}}_\eta = \mathbf{Q}^\dagger\mathbf{D}_\eta\mathbf{V}$.

4.3. Practical implementation

The reduced-order model for the compressible Navier–Stokes equations is given by

$$\frac{\mathrm{d}\tilde{\mathbf{v}}}{\mathrm{d}t} = \tilde{\mathbf{f}} + \tilde{\mathbf{A}}\tilde{\mathbf{v}} + \tilde{\mathbf{U}}\left[\hat{\mathbf{F}}(\hat{\bar{\mathbf{v}}}_\xi + \hat{\mathbf{V}}_\xi\tilde{\mathbf{v}}, \hat{\bar{\mathbf{w}}}_\xi + \hat{\mathbf{w}}_\xi) - \hat{\mathbf{A}}\tilde{\mathbf{v}} - \hat{\mathbf{f}}\right], \tag{55a}$$

and

$$\hat{\mathbf{w}}_\xi = \hat{\mathbf{W}}_\xi\left[\hat{\mathbf{G}}(\check{\bar{\mathbf{v}}}_\eta + \check{\mathbf{V}}_\eta\tilde{\mathbf{v}})] - \check{\mathbf{g}}\right]. \tag{55b}$$

A summary of the different matrices involved in the reduced-order model as well as their respective size is provided in Table 2; the procedure to generate these matrices is described in Algorithm 3. As in the case of the Euler equations, the vectors $\tilde{\mathbf{f}}$, $\hat{\bar{\mathbf{v}}}_\xi$, $\hat{\bar{\mathbf{w}}}_\xi$, $\hat{\mathbf{f}}$, $\check{\bar{\mathbf{v}}}_\eta$ and $\check{\mathbf{g}}$ and the matrices $\tilde{\mathbf{A}}$, $\tilde{\mathbf{U}}$, $\hat{\mathbf{V}}_\xi$, $\hat{\mathbf{A}}$, $\hat{\mathbf{W}}_\xi$ and $\check{\mathbf{V}}_\eta$ are computed offline. Once these matrices have been computed, the evaluation cost of the right-hand-side of the reduced order model in equation (55) has been reduced to $O(\alpha(l) + \beta(l) + k^2 + l^2 + (4+2d)lk)$ operations, where we have considered for simplicity $l = l_u \approx l_w$. In the above $\alpha(\cdot)$ and $\beta(\cdot)$ is the computational cost of evaluating $\hat{\mathbf{F}}(\cdot)$ and $\hat{\mathbf{G}}(\cdot)$, respectively.

Table 2
Nonlinear reduced-order model for the Navier–Stokes equations: summary table. The number of snapshots is n, the number of grid points is p, $q = 2$ for two-dimensional flows and $q = 3$ for three-dimensional flows. The length of the state vector $\mathbf{v}$ is $m_v = pq_v$, where $q_v = 2 + d$ is the number of flow variables per grid point; the length of the state vector $\mathbf{w}$ is $m_w = pq_w$, where $q_w = 5$ in two-dimensions and $q_w = 9$ in three dimensions. The size of the bases $\mathbf{V}$, $\mathbf{U}$ and $\mathbf{W}$ is, respectively k, l_u and l_w.

Vector/matrix	Number	Size	Description
$\bar{\mathbf{v}}$	1	$pq_v \times 1$	Reference field
$\left\{\mathbf{v}'(t_i)\right\}_{i=1}^{n}$	n	$pq_v \times 1$	Set of snapshots for the disturbances $\mathbf{v}' = \mathbf{v} - \bar{\mathbf{v}}$
$\mathbf{V}$	1	$pq_v \times k$	POD-based vector basis for the disturbances $\mathbf{v}'$
$\tilde{\mathbf{f}} = \mathbf{V}^\dagger \mathbf{F}(\bar{\mathbf{v}}, \bar{\mathbf{w}})$	1	$k \times 1$	Projection of $\mathbf{F}(\bar{\mathbf{v}}, \bar{\mathbf{w}})$ onto basis $\mathbf{V}$
$\tilde{\mathbf{A}} = \mathbf{V}^\dagger \mathbf{A}(\bar{\mathbf{v}}, \bar{\mathbf{w}})\mathbf{V}$	1	$k \times k$	Reduced-order linear dynamics
$\bar{\mathbf{w}} = \mathbf{G}(\bar{\mathbf{v}})$	1	$pq_w \times 1$	Reference field
$\left\{\mathbf{F}'(\bar{\mathbf{v}}, \bar{\mathbf{w}}; \mathbf{v}'(t_i), \mathbf{w}'(t_i))\right\}_{i=1}^{n}$	n	$pq_v \times 1$	Set of snapshots for the nonlinear term $\mathbf{F}'(\bar{\mathbf{v}}, \bar{\mathbf{w}}; \mathbf{v}', \mathbf{w}')$
$\mathbf{U}$	1	$pq_v \times l_u$	POD-based vector basis for the nonlinear term $\mathbf{F}'(\bar{\mathbf{v}}, \bar{\mathbf{w}}; \mathbf{v}', \mathbf{w}')$
$\mathbf{p}$	1	$l_u \times 1$	Interpolation points (I)
$\mathbf{P}$	1	$pq_{v,w} \times l_u$	Row-selector matrix for the interpolation points $\mathbf{p}$
$\tilde{\mathbf{U}} = (\mathbf{V}^\dagger \mathbf{U})(\mathbf{P}^\dagger \mathbf{U})^{-1}$	1	$k \times l_u$	Interpolation matrix (I)
$\hat{\bar{\mathbf{v}}}_\xi = \mathbf{P}^\dagger \mathbf{D}_\xi \bar{\mathbf{v}}$	$d + 1$	$l_u q_v \times 1$	Spatial derivatives of the vector $\bar{\mathbf{v}}$ at the interpolation points $\mathbf{p}$
$\hat{\mathbf{V}}_\xi = \mathbf{P}^\dagger \mathbf{D}_\xi \mathbf{V}$	$d + 1$	$l_u q_v \times k$	Spatial derivatives of the basis $\mathbf{V}$ at the interpolation points $\mathbf{p}$
$\hat{\bar{\mathbf{w}}}_\xi = \mathbf{P}^\dagger \mathbf{D}_\xi \bar{\mathbf{w}}$	$d + 1$	$l_u q_w \times 1$	Spatial derivatives of the vector $\bar{\mathbf{w}}$ at the interpolation points $\mathbf{p}$
$\hat{\mathbf{A}} = \mathbf{P}^\dagger \mathbf{A}(\bar{\mathbf{v}}, \bar{\mathbf{w}})\mathbf{V}$	1	$l_u \times k$	Linear terms at the interpolation points $\mathbf{p}$
$\hat{\mathbf{f}} = \mathbf{P}^\dagger \mathbf{F}(\bar{\mathbf{v}}, \bar{\mathbf{w}})$	1	$l_u \times 1$	Constant term at the interpolation points $\mathbf{p}$
$\left\{\mathbf{G}'(\bar{\mathbf{v}}; \mathbf{v}'(t_i))\right\}_{i=1}^{n}$	n	$pq_w \times 1$	Set of snapshots for the nonlinear term $\mathbf{G}'(\bar{\mathbf{v}}; \mathbf{v}')$
$\mathbf{W}$	1	$pq_w \times l_w$	POD-based vector basis for the nonlinear term $\mathbf{G}'(\bar{\mathbf{v}}; \mathbf{v}')$
$\mathbf{q}$	1	$l_w \times 1$	Interpolation points (II)
$\mathbf{Q}$	1	$pq_{v,w} \times l_w$	Row-selector matrix for the interpolation points $\mathbf{q}$
$\hat{\mathbf{W}}_\xi = (\mathbf{P}^\dagger \mathbf{D}_\xi \mathbf{W})(\mathbf{Q}^\dagger \mathbf{W})^{-1}$	$d + 1$	$l_u q_w \times l_w$	Interpolation matrix (II)
$\check{\bar{\mathbf{v}}}_\eta = \mathbf{Q}^\dagger \mathbf{D}_\eta \bar{\mathbf{v}}$	$d + 1$	$l_u q_w \times 1$	Reference field spatial derivatives at the interpolation points $\mathbf{q}$
$\check{\mathbf{V}}_\eta = \mathbf{Q}^\dagger \mathbf{D}_\eta \mathbf{V}$	$d + 1$	$l_u q_w \times k$	Spatial derivatives of basis $\mathbf{V}$ at the interpolation points $\mathbf{q}$
$\check{\mathbf{g}} = \mathbf{Q}^\dagger \mathbf{G}(\bar{\mathbf{v}})$	1	$l_w \times 1$	Constant term at the interpolation points $\mathbf{q}$

Algorithm 3 Nonlinear model-order reduction of the compressible Navier–Stokes equations.

```
Require: Reference field $\bar{\mathbf{v}}$, snapshots $\{\mathbf{v}(t_i)\}_{i=1}^{n}$, number of POD modes $k$, $l_v$ and $l_w$.
Generate snapshot sequence for the perturbations $\left\{\mathbf{v}'(t_i)\right\}_{i=1}^{n}$.
Construct $\mathbf{V}$ from the leading $k$ modes of the POD decomposition of $\left\{\mathbf{v}'(t_i)\right\}_{i=1}^{n}$.
Construct matrix $\tilde{\mathbf{A}}$ and vector $\tilde{\mathbf{f}}$.
Compute $\mathbf{G}(\bar{\mathbf{v}})$ and generate snapshot sequence for $\left\{\mathbf{w}'(t_i)\right\}_{i=1}^{n}$.
Generate snapshot sequences for the nonlinearities $\left\{\mathbf{F}'(\bar{\mathbf{v}}, \bar{\mathbf{w}}; \mathbf{v}'(t_i), \mathbf{w}'(t_i))\right\}_{i=1}^{n}$.
Construct $\mathbf{U}$ from the leading $l_v$ modes of the POD decomposition of $\left\{\mathbf{F}'(\bar{\mathbf{v}}, \bar{\mathbf{w}}; \mathbf{v}'(t_i), \mathbf{w}'(t_i))\right\}_{i=1}^{n}$.
Obtain interpolation point grid indices $\mathbf{p}_j$ and row selector matrices $\mathbf{P}$, $\mathbf{P}_v$ and $\mathbf{P}_w$.
Construct matrices $\tilde{\mathbf{U}}$, $\hat{\mathbf{V}}_\xi$, $\hat{\mathbf{A}}$, and vectors $\hat{\bar{\mathbf{v}}}_k$, and $\hat{\mathbf{f}}$.
Construct $\mathbf{W}$ from the leading $l_w$ modes of the POD decomposition of $\left\{\mathbf{G}'(\bar{\mathbf{v}}; \mathbf{v}'(t_i))\right\}_{i=1}^{n}$.
Obtain interpolation point grid indices $\mathbf{q}_j$ and row selector matrix $\mathbf{Q}$.
Construct matrices $\hat{\mathbf{W}}_\xi$, $\check{\mathbf{V}}_\eta$, and vectors $\check{\bar{\mathbf{v}}}_\eta$, and $\check{\mathbf{g}}$.
```

5. Examples of reduced-order models

In this section we demonstrate the model-reduction technique described above on a compressible Navier–Stokes solver. This code solves equations (37) together with equations (38) on multiblock, curvilinear grids using high-order schemes and a domain decomposition technique to perform parallel calculations. The equations are implemented in the so-called pseudo-characteristics formulation [15]. Discretization in space is performed using the Compact Upwind Low-Dissipation (CULD) schemes [16] for the advection terms and centered third-order schemes for the diffusive terms [17]. Integration in time is carried out using a low-storage 4-th-order Runge–Kutta scheme (see [18]). Suitable boundary conditions are implemented using the Navier–Stokes characteristic-based boundary conditions [19,20], and sponge layers [21] are included, where appropriate, to avoid spurious reflections from the boundaries of the computational domain. Further details about the numerical code and the choice of non-dimensional variables can be found in Fosas de Pando et al. (2014) [22].

5.1. Vortex shedding and sound radiation in the wake of an airfoil

We consider the two-dimensional compressible flow around a NACA-0012 airfoil at a zero angle-of-attack. The Reynolds number based on the chord-length of the airfoil is chosen as $\mathrm{Re} = 10000$, the Mach number is taken as $\mathrm{M} = 0.6$, the heat capacity ratio is $\gamma = 1.4$, and the Prandtl number is fixed at $\mathrm{Pr} = 0.71$. To properly resolve the flow in the vicinity of the airfoil as well as the free-stream and wake, a nonuniform numerical grid is required that ultimately yields a dynamical

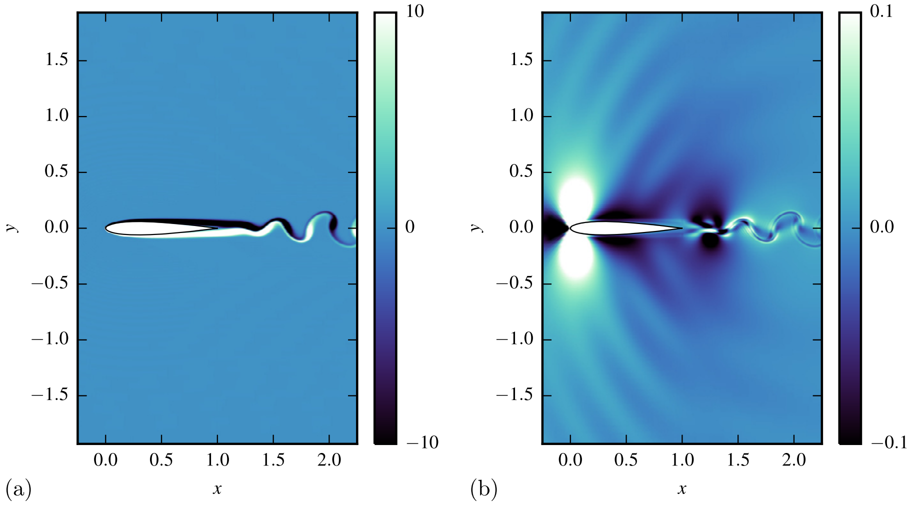

Fig. 4. Instantaneous flow field about a NACA 0012 airfoil at a zero angle-of-attack, visualized (a) by the vorticity field showing periodic vortex shedding into the wake and (b) by the dilatation field showing significant upstream sound directivity. The Reynolds number is $\mathrm{Re} = 10^4$, and the Mach number is $\mathrm{M} = 0.6$.

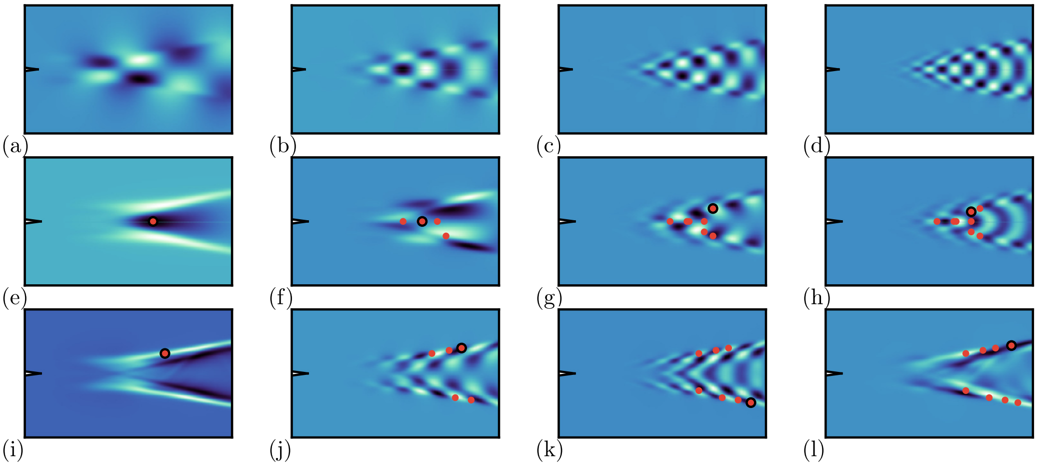

Fig. 5. Selected modes of (a–d) POD basis $\mathbf{V}$ of the snapshot series, (e–h) POD basis $\mathbf{U}$, and (i–l) POD basis $\mathbf{W}$ of the nonlinear snapshots together with the interpolation points selected by the DEIM algorithm. In (a–h) cases the stream-wise component of the velocity vector is represented, whereas in (i–l) the stream-wise component of the heat flux is depicted.

system with $m = 10^6$ degrees of freedom. Integration in time, starting from a suitable initial condition, has been performed until transients attenuate and a finite-amplitude limit-cycle emerges at $t \approx 100$. At this point, the flow fields are sampled and stored every $\Delta = 0.01$ time units; this produces a data sequence from $t_0 = 100$ to $t_f = 105$ with $n = 500$ snapshots.

Fig. 4 shows the main elements of the limit-cycle state: vortices (visualized by the ω_z-component) shed from the vicinity of the trailing edge into the wake (see Fig. 4(a)); the corresponding dilatation field $d = \partial_x u + \partial_y v$ shows acoustic waves radiating from the trailing edge with a preferred upstream directionality (Fig. 4(b)). The mean flow $\bar{\mathbf{v}}$ is taken as the reference field.

We proceed by forming the different bases, i.e., $\mathbf{V}$, $\mathbf{U}$, and $\mathbf{W}$, from the snapshot sequence $\{\mathbf{v}(t_i)\}_{i=1}^n$, together with their respective DEIM-points. To this end, we follow the three steps: (i) the basis $\mathbf{V}$ is generated by computing the Proper Orthogonal Decomposition or singular value decomposition of the snapshot matrix consisting of $\left\{\mathbf{v}'(t_i)\right\}_{i=1}^n$, (ii) the vector bases $\mathbf{U}$ and $\mathbf{W}$ for the nonlinear terms are extracted via Proper Orthogonal Decomposition from the nonlinear-terms snapshots $\left\{\mathbf{F}'(\bar{\mathbf{v}}; \mathbf{v}'(t_i))\right\}_{i=1}^n$, and $\left\{\mathbf{G}'(\bar{\mathbf{v}}; \mathbf{v}'(t_i))\right\}_{i=1}^n$, and (iii) the grid coordinates of the discrete interpolation points are determined, for each of the nonlinear bases, using the above DEIM algorithm.

Figs. 5(a–d) depict the dominant structures of the basis $\mathbf{V}$, visualized by the perturbation stream-wise velocity component u'. All modes are characterized by spatial support in the wake, with specific wavelengths and symmetry patterns. Only modes $k = 1$ and $k = 2$ exhibit acoustic radiation in the far field (not shown). The principal vectors of the bases $\mathbf{U}$ and $\mathbf{W}$ are displayed, respectively, in Figs. 5(e–h) and Figs. 5(i–l), together with their interpolation points.

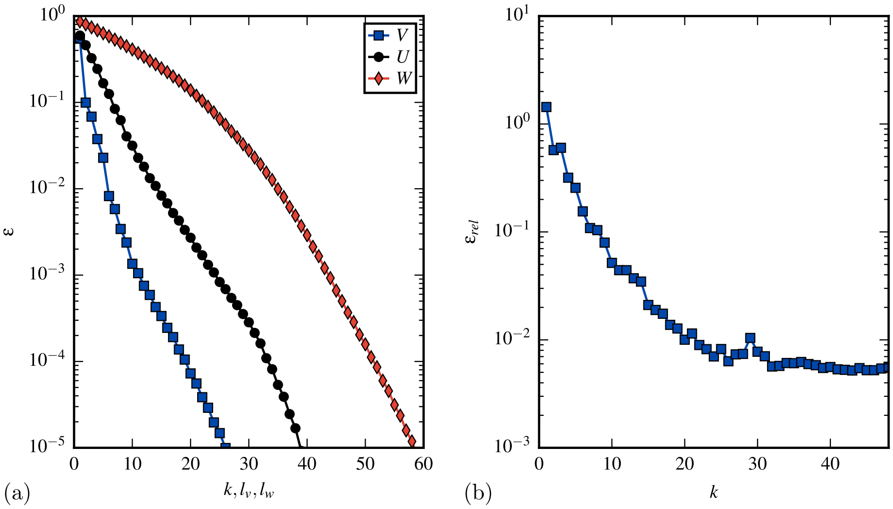

Fig. 6. (a) Residual ε for the different bases **V**, **U** and **W**. (b) Error between the full system and the reduced order models for varying number of vector basis.

The residuals $\varepsilon(\cdot)$, as defined in equation (15), are displayed in Fig. 6(a) as a function of the number k of basis vectors that are retained. We have constructed reduced-order models for increasing sizes of the vector bases, following the above procedure. For a given number of basis vectors in **V**, a criterion for the number of nonlinear basis vectors in **U** and **W** has to be introduced. We consider two choices: (i) we select an equal number of vectors, and (ii) we include as many vectors as necessary to match the respective residuals.

For each choice, we integrate the reduced-order model in time over $T = 5$ units and compare the results to the original (non-reduced) system by computing the relative residual ε_{rel} given by

$$\varepsilon_{\text{rel}} = \sqrt{\frac{\sum_{j=1}^{n} \|\mathbf{v}'(t_j) - \mathbf{V}\tilde{\mathbf{v}}(t_j)\|^2}{\sum_{j=1}^{n} \|\mathbf{v}'(t_j)\|^2}}. \tag{56}$$

The result is shown in Fig. 6(b), where a decrease in the residual can be observed as the basis vectors are augmented, until a residual level of approximately $5 \cdot 10^{-3}$ is reached for $k = 32$ beyond which no more improvements could be made. No appreciable difference between the two suggested criteria for choosing the number of vectors in the nonlinear bases could be identified. In Fig. 7(c) the spatial distribution of the relative residual, given by $\varepsilon(\mathbf{x}_i) = \sqrt{\sum_{j=1}^{n} \|\mathbf{v}_i'(t_j) - \mathbf{V}_i\tilde{\mathbf{v}}(t_j)\|^2 / \sum_{j=1}^{n} \|\mathbf{v}_i'(t_j)\|^2}$ is also displayed.

Robustness of the reduced-order model to perturbations about the limit cycle and the correct prediction of the long-time behavior constitute qualities that are important in many applications. To gain more insight into these properties of our reduced-order models and thus of our model-reduction algorithm, we present in Fig. 8 the norm of the state vector for long integration times starting from various initial conditions. Among these initial conditions are starting values on the limit cycle (indicated by the dotted line) and starting values far from the limit cycle or the mean flow (solid line). For severely truncated systems with $k < 4$, the reduced-order models diverge, no matter whether we start at the limit cycle or far from it. Increasing the number of basis vector, for example by choosing $k = 4$, renders the reduced-order models stable, but a discrepancy to the correct limit-cycle amplitude is observed over the long term. For $k > 16$, this discrepancy is substantially reduced, and the limit-cycle amplitude is reproduced remarkably accurately and robustly by the reduced-order model, independent of the starting point of the time integration.

6. Conclusions

An effective algorithm for the design of nonlinear reduced-order models of large-scale compressible flows has been presented. It relies on an earlier study [1] and is based on an POD-based Galerkin projection using multiple bases. These bases are attuned to particular terms of the governing equations, in our case the nonlinear terms, to produce a high-fidelity, more localized and robust representation of the dynamic processes arising from these terms. The various bases are joined by the Discrete Empirical Interpolation Method (DEIM) that impose the nonlinear terms at specific points, which are chosen by a greedy algorithm to sequentially reduce the residual of the nonlinear reconstruction. The original model-reduction algorithm has been generalized to take into account nonlocal nonlinearities (present in simulations of compressible flow). This has been accomplished by introducing auxiliary variables that are treated independently and thus reduce nonlocal nonlinearities to local ones. An efficient implementation of the nonlinear model-reduction technique, taking advantage of

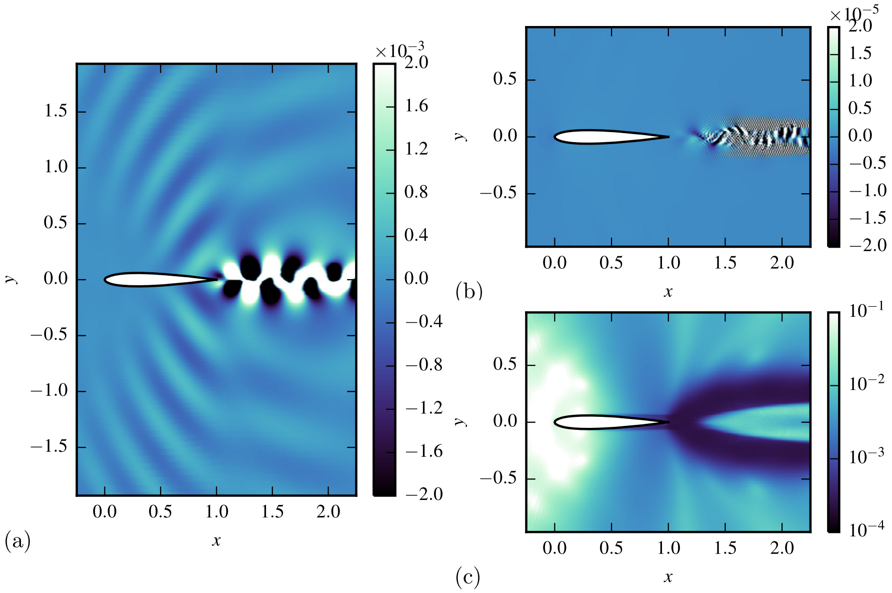

Fig. 7. (a) Pressure field at $t = 105$ computed using the POD–DEIM reduced-order model with $k = 32$, (b) instantaneous pressure-field difference between the reduced-order model and the numerical simulation (note the different color scale), and (c) spatial distribution of the relative residual for the complete simulation. See text for details.

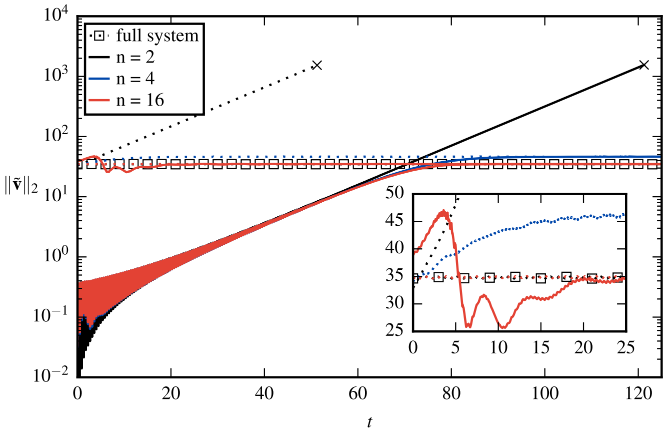

Fig. 8. Norm of the state vector for reduced-order models of orders $k = 2$, $k = 4$ and $k = 16$ for different initial conditions: steady state (dotted line), zero initial condition and perturbations around the limit cycle. The amplitude of the limit-cycle of the full system is also shown.

our previous work [2], has been achieved, that results in a flexible and adaptive framework for the design of nonlinear low-dimensional models of complex flows.

As a validation case, we chose the compressible flow in the wake of a NACA0012 airfoil. This type of flows contains a range of spatial and local scales with localized nonlinearities and is typical and representative of large-scale multi-physics simulations. The reduced-order models contained only a few degrees of freedom (with the full simulation of the order of 10^6 degrees of freedom) but, nonetheless, captured the mean features of the flow with remarkable accuracy. Moreover, an encouraging and important robustness of the reduced-order model has been found, as it recovered the dynamics of the full system even for off-design conditions. Both accuracy and robustness are important and necessary attributes for reduced-order models in control applications. Our future efforts will focus on the design of nonlinear reduced-order models for flow control of complex flows and on nonlinear model-predictive control applications based on our POD–DEIM models.

Acknowledgements

Parts of this work were developed at the CTR 2014 Summer Program. The authors wish to thank the organizers and the participants of this program for the stimulating and supportive environment. Research funding by the *Fondation de Recherche pour l'Aéronautique et l'Espace (FRAE)* through the project *Estimation, Contrôle et Stabilisation d'Ecoulements Aérodynamique (ECOSEA)* is gratefully acknowledged.

References

[1] S. Chaturantabut, D.C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, SIAM J. Sci. Comput. 32 (5) (2010) 2737–2764.

[2] M. Fosas de Pando, D. Sipp, P.J. Schmid, Efficient evaluation of the direct and adjoint linearized dynamics from compressible flow solvers, J. Comput. Phys. 231 (23) (2012) 7739–7755.

[3] A.C. Antoulas, Approximation of Large-Scale Dynamical Systems, Advances in Design and Control, vol. 6, SIAM Publishing, 2004.

[4] A. Quarteroni, G. Rozza, Reduced Order Methods for Modeling and Computational Reduction, MS&A — Modeling, Simulation & Applications, Springer Verlag, 2014.

[5] K. Ito, S.S. Ravindran, A reduced-order method for simulation and control of fluid flows, J. Comput. Phys. 143 (2) (1998) 403–425.

[6] S.S. Ravindran, Control of flow separation over a forward-facing step by model reduction, Comput. Methods Appl. Math. 191 (41) (2002) 4599–4617.

[7] M. Samimy, M. Debiasi, E. Caraballo, A. Serrani, X. Yuan, J. Little, J.H. Myatt, Feedback control of subsonic cavity flows using reduced-order models, J. Fluid Mech. 579 (2007) 315.

[8] A. Barbagallo, D. Sipp, P.J. Schmid, Closed-loop control of an open cavity flow using reduced-order models, J. Fluid Mech. 641 (2009) 1–50.

[9] T. Lassila, A. Manzoni, A. Quarteroni, G. Rozza, Model order reduction in fluid dynamics: challenges and perspectives, in: Reduced Order Methods for Modeling and Computational Reduction, Springer Verlag, 2014, pp. 235–273.

[10] S. Chaturantabut, D.C. Sorensen, A state space error estimate for POD–DEIM nonlinear model reduction, SIAM J. Numer. Anal. 50 (1) (2012) 46–63.

[11] D. Wirtz, D. Sorensen, B. Haasdonk, A posteriori error estimation for DEIM reduced nonlinear dynamical systems, SIAM J. Sci. Comput. 36 (2) (2014) 1–27.

[12] S. Chaturantabut, D.C. Sorensen, Application of POD and DEIM on dimension reduction of non-linear miscible viscous fingering in porous media, Math. Comput. Model. Dyn. Syst. 17 (4) (2011) 337–353.

[13] M. Ilak, S. Bagheri, L. Brandt, C.W. Rowley, D.S. Henningson, Model reduction of the nonlinear complex Ginzburg–Landau equation, SIAM J. Appl. Dyn. Syst. 9 (4) (2010) 1284–1302.

[14] D.A. Knoll, D.E. Keyes, Jacobian-free Newton–Krylov methods: a survey of approaches and applications, J. Comput. Phys. 193 (2) (2004) 357–397.

[15] J. Sesterhenn, A characteristic-type formulation of the Navier–Stokes equations for high order upwind schemes, Comput. Fluids 30 (1) (2000) 37–67.

[16] N.A. Adams, K. Shariff, A high-resolution hybrid compact-ENO scheme for shock-turbulence interaction problems, J. Comput. Phys. 127 (1) (1996) 27–51.

[17] S.K. Lele, Compact finite difference schemes with spectral-like resolution, J. Comput. Phys. 103 (1) (1992) 16–42.

[18] C.A. Kennedy, M.H. Carpenter, R.M. Lewis, Low-storage, explicit Runge–Kutta schemes for the compressible Navier–Stokes equations, Appl. Numer. Math. 35 (3) (2000) 177–219.

[19] T.J. Poinsot, S.K. Lele, Boundary conditions for direct simulations of compressible viscous flows, J. Comput. Phys. 101 (1) (1992) 104–129.

[20] G. Lodato, P. Domingo, L. Vervisch, Three-dimensional boundary conditions for direct and large-eddy simulation of compressible viscous flows, J. Comput. Phys. 227 (10) (2008) 5105–5143.

[21] D.J. Bodony, Analysis of sponge zones for computational fluid mechanics, J. Comput. Phys. 212 (2) (2006) 681–702.

[22] M. Fosas de Pando, P.J. Schmid, D. Sipp, A global analysis of tonal noise in flow around aerofoils, J. Fluid Mech. 754 (2014) 5–38.