Non-linearly stable reduced-order models for incompressible flow with energy-conserving finite volume methods

B. Sanderse

Centrum Wiskunde & Informatica, Amsterdam, the Netherlands

A B S T R A C T

A novel reduced-order model (ROM) formulation for incompressible flows is presented with the key property that it exhibits non-linearly stability, independent of the mesh (of the full order model), the time step, the viscosity, and the number of modes. The two essential elements to non-linear stability are: (1) *first discretise* the full order model, and *then project* the discretised equations, and (2) use spatial and temporal discretisation schemes for the full order model that are *globally energy-conserving* (in the limit of vanishing viscosity). For this purpose, as full order model a staggered-grid finite volume method in conjunction with an implicit Runge-Kutta method is employed. In addition, a constrained singular value decomposition is employed which enforces global momentum conservation. The resulting 'velocity-only' ROM is thus globally conserving mass, momentum and kinetic energy. For non-homogeneous boundary conditions, a (one-time) Poisson equation is solved that accounts for the boundary contribution. The stability of the proposed ROM is demonstrated in several test cases. Furthermore, it is shown that explicit Runge-Kutta methods can be used as a practical alternative to implicit time integration at a slight loss in energy conservation.

© 2020 Elsevier Inc. All rights reserved.

1. Introduction

The simulation of turbulent fluid flows is an ongoing challenge in the scientific community. The computational cost of Direct Numerical Simulation (DNS) or Large Eddy Simulation (LES) of turbulent flows quickly becomes imperative when one is interested in *control*, *design*, *optimization* and *uncertainty quantification* [9,47]. For these purposes, a reduction in complexity of the full model is required to arrive at a computationally tractable model, a so-called reduced-order model (ROM). Several techniques exist to construct a ROM, such as balanced truncation, Krylov subspace methods, and POD-Galerkin methods [5,49]. In this work we focus on one of the most popular techniques, the POD-Galerkin method, in which the governing equations of the full model are projected onto a lower-dimensional space via a Galerkin step, with the projection basis determined from a proper orthogonal decomposition (POD) of snapshots of the full order model (FOM).

Projection-based models have been shown to work for a large class of problems, such as diffusion-dominated linear time-invariant (LTI) systems, in which the input-output relation of the full model can be represented by a lower-dimensional model, due to rapid decay of the singular values of the Hankel matrix [9]. However, in turbulent flow, which is a nonlinear, convection-dominated problem, the construction of accurate and stable ROMs is still an open challenge. There are several (related) reasons why current ROMs have issues with accuracy and stability in case of turbulent flows: the Kolmogorov

N-width decays too slowly; the non-linear dynamical system is very sensitive to perturbations; the modes with low energy (small scales, dissipation) which are neglected in the POD procedure are relevant for the dynamics of the large scales; the reduced model can have different stability characteristics [6,13,18,33,37,48].

A number of approaches have been proposed to tackle the issues of accuracy and stability; we summarize the list in [18]: including dissipation via a closure model (see also [15]); modifying the POD basis by including functions that resolve a range of scales; using a minimum residual formulation [14]; using an inner product different from L_2, e.g. based on H_1. In this work we are interested in tackling the issues of accuracy and stability through the promising approach of structure-preserving model reduction, in which reduced-order models are developed in such a way that invariants and/or symmetries of the full model are kept [2,3,14,27,35]. An example is a ROM that inherits the symplectic form of a Hamiltonian system, leading to a ROM that is applicable to stable long-time integration [35].

In the incompressible Navier-Stokes equations, conservation (or dissipation) of kinetic energy is typically used in order to proof non-linear stability, as the kinetic energy is a norm of the solution. Although the incompressible Navier-Stokes equations do not form a Hamiltonian system, several symmetries are present in the equations which are tightly related to conservation of kinetic energy, such as the skew-symmetry of the convective operator, and the symmetry of the diffusive operator. Several adaptations to the classic POD-Galerkin method were developed to take into account symmetry or invariance properties of the Navier-Stokes equations. For example, Balajewicz et al. [7] added a power balance equation for the resolved turbulent kinetic energy when solving for the POD basis functions and coefficients. Mohebujjaman et al. [31] employed a combined projection and data-driven approach in a finite-element context and obtain correction terms by solving a constrained minimization problem, using as Ansatz the negative definiteness of the diffusion operator and the energy-conserving property of the convection term. Mohebujjaman et al. also [32] investigated conservation of mass and energy of the ROM in the context of a finite element framework and discuss the treatment of non-homogeneous boundary conditions via a Stokes extension in order to mimic the continuous energy balance. Carlberg et al. [14] considered conservative model reduction in a finite-volume context by solving a constrained optimization problem at each time step. Rowley et al. [38] considered the choice of an appropriate inner product and corresponding energy norm for compressible flow. Kalashnikova et al. [26] considered energy stability in terms of a continuous formulation. However, to the author's knowledge, a reduced order model for the incompressible Navier-Stokes equations (based on a finite-volume full-order method) that combines global mass, momentum and kinetic energy conservation (in the inviscid limit) has not been developed to date. This will be the subject of this paper.

To achieve energy conservation, we require the discretised full-order model to respect several symmetries of the Navier-Stokes equations: skew-symmetry of the convective term, symmetry of the diffusive term, and the compatibility relation between the divergence (of the velocity) and the gradient (of the pressure). With the latter property satisfied, the contribution of the pressure in the kinetic energy equation disappears (in closed or periodic domains) due to the fact that the velocity-field is divergence free [25]. Similarly, when a divergence-free basis for the velocity-field is taken in a Galerkin method, the pressure gradient term disappears from the momentum equations. The resulting model is 'velocity-only', which has the advantage that it is faster to evaluate in the online phase (compared to a velocity-pressure formulation), and moreover can avoid the inf-sup stability issues associated to a mixed formulation [19], so that stabilization techniques, such as supremizer modes or a pressure Poisson equation [8,11,39,48] are not needed. An important issue when deriving such a 'velocity-only' formulation is, next to the discrete compatibility between divergence and gradient operators, the requirement of a discretely divergence-free velocity field. The importance of this property for the accuracy of the ROM was also confirmed in [32]. Another issue is that the velocity-only formulation requires a method to recover the pressure from the velocity field [4,28] and requires adaptation if the pressure is non-negligible on outflow boundaries [34]. In this work, these issues are effectively handled by choosing as discrete FOM a finite-volume discretisation on a staggered grid. This discretisation is energy-conserving in space [23,53] and time [41], possesses the convective skew-symmetry, the diffusive symmetry, the compatibility relation between divergence and gradient, and has divergence-free velocity fields. Furthermore, it does not require boundary conditions for the pressure on parts of the boundary where the velocity is prescribed, allowing straightforward recovery of the pressure on the ROM level.

Based on this energy-conserving full-order model, we propose a new stable energy-conserving reduced-order model, which possesses non-linear stability, independent of the viscosity, mesh size, time step or number of modes. A constrained SVD method is employed with a weighted inner product, *in such a way that the reduced model is mass-, momentum- and energy-conserving*. The resulting ROM is 'velocity-only' and therefore does not suffer from inf-sup stability issues. As we will show, the pressure can be computed, if desired, as a postprocessing step by solving a Poisson equation at the reduced-order level. An important concept in our work is that we first discretise the equations, and then perform the projection step. Although this approach is common in the dynamical systems community [9], and has also been used by e.g. [14,48], we are explicitly mentioning this choice because it allows us to prove stability on the level of the ROM and because it simplifies the boundary condition treatment. In particular, once the discrete full-order model and its boundary conditions are specified, no additional boundary conditions will be needed on the ROM level. Non-homogeneous boundary conditions are handled in such a way that the ROM stays velocity-only, and such that no additional boundary conditions need to be imposed when recovering the pressure via a pressure Poisson equation.

A graphical summary of the approach is shown in Fig. 1: first spatial discretization, then projection, and then temporal discretization. As said, the order of the first two steps is especially important, because spatial discretization and projection do generally not commute (amongst others due to the presence of boundary conditions). On the other hand, the last two
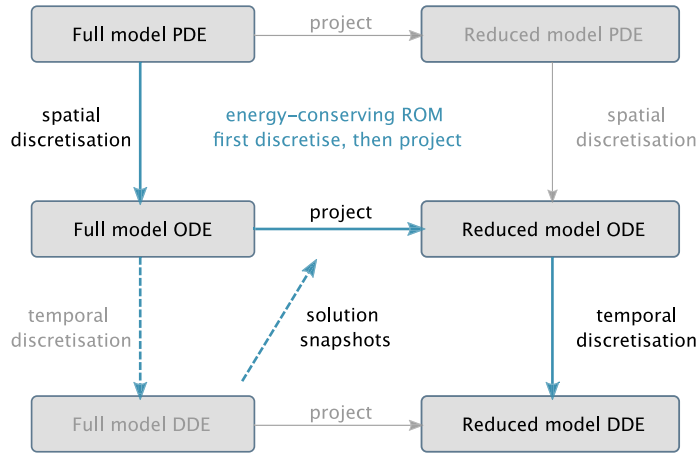
Fig. 1. Our approach to energy-stable reduced order models follows the blue arrows: first spatial discretisation, then projection (ODE = ordinary differential equation, DDE = discretised differential equation). (For interpretation of the colours in the figure(s), the reader is referred to the web version of this article.)

steps generally commute: for many time integration methods the same ROM would be obtained when first discretizing in time and then projecting the fully discrete equations [12]. However, this latter sequence does not give us the flexibility to take a different time integration method for the ROM than for the FOM, e.g. changing from explicit to implicit time integration. This flexibility will be used in the test cases to make our approach computationally efficient.

We limit ourselves in the analysis in several important aspects. First, we will consider the so-called *solution reproduction problem*, which is the first step before solving the full *parametric problem* [18]. Second, we will focus mainly on the *non-linear stability* of the discrete ROM, and refrain from turbulent flows and closure models. Given a non-linearly stable and non-dissipative ROM, we have a framework in which we can in future work assess the accuracy of closure models, e.g. [15,31,54]. Our definition of stability should therefore be interpreted in the classical sense: a certain norm of the solution stays bounded in time. Note that this is different from the 'stabilizing' methods that have been proposed in the ROM community, which are required to counteract numerical oscillations [55]. This latter type of (in)stability is not the focus of this paper.

The novelty of this paper is threefold. First, we derive an energy-conserving reduced-order model, which possesses *nonlinear stability independent of the mesh and time step of the FOM, and independent of the time step and number of modes of the reduced-order model*. Second, we *enforce global momentum conservation via a constrained singular value decomposition*. Third, we propose a procedure to handle non-homogeneous boundary conditions which keeps the ROM 'velocity-only' and the pressure recovery stage straightforward.

This paper is organized as follows. First, in section 2 we discuss symmetry and energy-conservation properties of the full-order model (the incompressible Navier-Stokes equations) at the continuous, semi-discrete and fully discrete level. In section 3 we construct the new POD-Galerkin method, which conserves mass, momentum and energy globally. In section 4 non-homogeneous boundary conditions are discussed. In section 5 we discuss implementation issues, including precomputing the reduced operators, the pressure recovery, and alternative (explicit) time integration strategies. In section 6 the theoretical results are demonstrated for three cases: a shear layer roll-up, a lid-driven cavity flow, and an actuator disk with unsteady wake.

2. Energy-conserving discretisation of the incompressible Navier-Stokes equations

In order to develop the ROM, the energy (in)equalities of the FOM on the continuous and discrete level are needed. These are derived in this section.

2.1. Continuous energy estimate

The incompressible Navier-Stokes equations describe conservation of mass and momentum:

$$\nabla \cdot \boldsymbol{u} = 0, \tag{1}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) = -\nabla p + \nu \nabla \cdot (\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T) + \boldsymbol{f}, \tag{2}$$

where $\boldsymbol{u}(\boldsymbol{x}, t)$ is the velocity field, $p(\boldsymbol{x}, t)$ is the modified pressure (pressure scaled by density), $\boldsymbol{x} \in \Omega \subset \mathbb{R}^d$ ($d = 2$ or 3), t denotes time, $\nu \geq 0$ the kinematic viscosity, and $\boldsymbol{f}(\boldsymbol{x}, t)$ represents body forces. The equations are supplemented with an initial condition

$$\boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{u}_0(\boldsymbol{x}), \tag{3}$$

and boundary conditions, e.g. no-slip conditions

$$\boldsymbol{u} = \boldsymbol{0} \quad \text{on} \quad \partial\Omega. \tag{4}$$

In this section and the next (section 3), the focus will be on periodic and no-slip conditions. The extension to more generic boundary conditions (inflow, outflow) will be addressed in section 4. We introduce the convection and diffusion operators $C(\boldsymbol{u}, \boldsymbol{u}) := \nabla \cdot (\boldsymbol{u} \otimes \boldsymbol{u})$ and $D\boldsymbol{u} := \nabla \cdot (\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T)$. Other forms of the convective operator are detailed in Appendix A.1.

To derive the kinetic energy equation, an inner product is needed. We choose the $L_2(\Omega)$ inner product and induced norm [10]:

$$(\boldsymbol{u}, \boldsymbol{v}) := \int_\Omega \boldsymbol{u} \cdot \boldsymbol{v} \, d\Omega, \qquad \|\boldsymbol{u}\| := (\boldsymbol{u}, \boldsymbol{u})^{1/2}. \tag{5}$$

The kinetic energy is then defined as $K := \frac{1}{2}\|\boldsymbol{u}\|^2$. An equation for the evolution of K is derived by differentiating K in time and substituting the momentum equation:

$$2\frac{dK}{dt} = \frac{d(\boldsymbol{u}, \boldsymbol{u})}{dt} = -(C(\boldsymbol{u}, \boldsymbol{u}), \boldsymbol{u}) - (\boldsymbol{u}, C(\boldsymbol{u}, \boldsymbol{u})) - (\nabla p, \boldsymbol{u}) - (\boldsymbol{u}, \nabla p) + (D\boldsymbol{u}, \boldsymbol{u}) + (\boldsymbol{u}, D\boldsymbol{u}) + (\boldsymbol{u}, \boldsymbol{f}) + (\boldsymbol{f}, \boldsymbol{u}). \tag{6}$$

The equation simplifies due to three symmetry properties. These symmetry properties will be crucial in developing an energy-stable ROM. First, due to the skew-symmetry of $C(\boldsymbol{u}, \boldsymbol{u})$, we have $(C(\boldsymbol{u}, \boldsymbol{u}), \boldsymbol{u}) = 0$ for periodic or no-slip boundary conditions (see also Appendix A.1). Second, the pressure gradient contribution disappears because $(\nabla p, \boldsymbol{u}) = (p, \nabla \cdot \boldsymbol{u}) = 0$. Third, due to the symmetry of the diffusive operator we can write $(D\boldsymbol{u}, \boldsymbol{u}) = -(\nabla \boldsymbol{u}, \nabla \boldsymbol{u})$. The kinetic energy balance then reduces to

$$\frac{dK}{dt} = -\nu \|\nabla \boldsymbol{u}\|^2, \tag{7}$$

in the absence of body forces and boundary conditions. Consequently, in viscous flow the kinetic energy of the flow can only decrease in time through dissipation, and in inviscid flow it is conserved.

2.2. Spatial discretisation and semi-discrete energy equation

In order to construct a non-linearly stable ROM, we first require that the spatial discretisation mimics the energy-conserving properties of the continuous equations just derived. To this end, we consider a finite volume discretisation on a staggered cartesian grid, also known as the Marker-and-Cell (MAC) method [23,44,53] – see Fig. 2. This specific choice is an important one, as will be explained soon. For simplicity, we restrict ourselves to a second-order method in two dimensions and partition the domain in $N_p = N_x \times N_y$ finite volumes. We introduce the (time-dependent) solution vectors $u_h(t) \in \mathbb{R}^{N_p}$, $v_h(t) \in \mathbb{R}^{N_p}$ and $p_h(t) \in \mathbb{R}^{N_p}$, which consist of the (time-dependent) unknowns $u_{i+1/2,j}(t)$, $v_{i,j+1/2}(t)$, and $p_{i,j}(t)$, respectively (for $i = 1 \dots N_x$, $j = 1 \dots N_y$). The explicit time-dependence will be suppressed when no confusion can arise. The horizontal and vertical velocity components are gathered in the vector $V_h = \begin{pmatrix} u_h \\ v_h \end{pmatrix} \in \mathbb{R}^{N_V}$, with $N_V = 2N_p$.

We integrate the divergence-free constraint (1) over a finite volume centred around the unknown $p_{i,j}$, which yields

$$\bar{u}_{i+1/2,j} - \bar{u}_{i-1/2,j} + \bar{v}_{i,j+1/2} - \bar{v}_{i,j-1/2} = 0. \tag{8}$$

The notation $\bar{(.)}$ indicates integration over a face of the finite volume, approximated e.g. by $\bar{u}_{i+1/2,j} = u_{i+1/2,j}\Delta y$. In matrix-vector notation, the above equation can be written for all pressure volumes as

$$M_h V_h = 0, \qquad M_h \in \mathbb{R}^{N_p \times N_V}. \tag{9}$$

Next we integrate the horizontal component of the momentum equation over a finite volume centred around the unknown $u_{i+1/2,j}$. The convective term in divergence form is discretised by (mesh-independent) interpolation of the neighbouring fluxes. The divergence form ensures that momentum is conserved. When ensuring that the velocity field is discretely divergence free, the divergence form can be rewritten into the following equivalent skew-symmetric form (see Appendix A.2):

$$[C_h^u(V_h, u_h)]_{i+1/2,j} := \frac{1}{2}u_{i+3/2,j}\frac{1}{2}\left(\bar{u}_{i+1/2,j} + \bar{u}_{i+3/2,j}\right) - \frac{1}{2}u_{i-1/2,j}\frac{1}{2}\left(\bar{u}_{i-1/2,j} + \bar{u}_{i+1/2,j}\right)$$
$$+ \frac{1}{2}u_{i+1/2,j+1}\frac{1}{2}\left(\bar{v}_{i,j+1/2} + \bar{v}_{i+1,j+1/2}\right) - \frac{1}{2}u_{i+1/2,j-1}\frac{1}{2}\left(\bar{v}_{i,j-1/2} + \bar{v}_{i+1,j-1/2}\right). \tag{10}$$
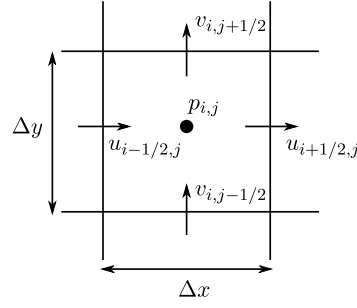
Fig. 2. Staggered grid with positioning of unknowns around a pressure volume.

The skew-symmetric form ensures that kinetic energy is conserved. As a consequence, the discretised convection operator is conserving both momentum and energy. A similar scheme can be derived for the vertical component. The full convection operator then reads

$$C_h(V_h, V_h) = \begin{pmatrix} C_h^u(V_h, u_h) \\ C_h^v(V_h, v_h) \end{pmatrix} =: \tilde{C}_h(V_h)V_h. \tag{11}$$

The notation $\tilde{C}_h(V_h)V_h$ is useful for making a distinction between the convect*ing* quantity (the quantity between brackets) and the convect*ed* quantity. The definition of $\tilde{C}_h$ is possible due to the fact that the nonlinearity of the convective term is only quadratic. The skew-symmetry property can be expressed in terms of $\tilde{C}_h$ as

$$\tilde{C}_h(V_h) = -\tilde{C}_h(V_h)^T. \tag{12}$$

The pressure gradient term in (2) is approximated by $G_h p_h$, where

$$G_h = -M_h^T. \tag{13}$$

This relation between the divergence and gradient operator is a key aspect of our approach. Firstly, it allows us to formulate a velocity-only ROM, as will be detailed in section 3.4. Secondly, the relation can be seen as a compatibility condition that ensures inf-sup stability [20,45], similar to the inf-sup condition typically encountered in finite element methods. Thirdly, it ensures that (like in the continuous case) kinetic energy is conserved by the pressure gradient term, since $V_h^T G_h p_h = -p_h^T M_h V_h = 0$. Relation (13) is naturally satisfied by our choice of a finite volume method on a staggered grid, but can also be achieved with different discretisation methods, including collocated grids [50] and finite element methods [21].

It is important to note that on inflow, no-slip or periodic boundaries, *no boundary conditions are required for the pressure* – they are implied by the boundary conditions for the velocity [44,52]. In short, this is because on a staggered grid the pressure gradient is readily available at the interior velocity points, and on the boundaries the velocity is specified. For details we refer to Appendix B.1.

The diffusive operator is discretised by second order central approximations, and can be represented by

$$D_h V_h, \tag{14}$$

where D_h is a symmetric negative definite matrix, which can be written as $D_h = -Q_h^T Q_h$.

The semi-discrete system then reads

$$M_h V_h = 0, \tag{15}$$

$$\Omega_h \frac{dV_h}{dt} = F_h(V_h, p_h), \tag{16}$$

where $F_h(V_h, p_h)$ contains the convective, diffusive, pressure gradient, and body force contributions:

$$F_h(V_h, p_h) = F_h^{CD}(V_h) - G_h p_h, \tag{17}$$

$$F_h^{CD}(V_h) = -\tilde{C}_h(V_h)V_h + \nu D_h V_h + f_h. \tag{18}$$

f_h is the discrete representation of the body-force (possibly time-dependent, but to ease notation the explicit time-dependence will be left out in f_h, F_h and F_h^{CD}). Ω_h is a (time-independent) diagonal matrix with the finite volume sizes on its diagonal, which is symmetric positive definite. The pressure Poisson equation (PPE) follows by differentiating (15) and substituting (16):

$$L_h p_h = M_h \Omega_h^{-1} F_h^{CD}(V_h), \qquad L_h = M_h \Omega_h^{-1} G_h. \tag{19}$$

Like the momentum equations, the pressure Poisson equation (PPE) does not require the specification of pressure boundary conditions: the PPE is derived from the *discrete* mass and momentum equations instead of the continuous counterparts, so that boundary conditions on the velocity suffice. This paradigm of working on the discrete level will also be applied when deriving the ROM. Furthermore, equation (19) will be the basis for the pressure recovery method in section 5.3.

To arrive at a semi-discrete energy equation a discrete inner product is needed, i.e. a discrete version of (5). The natural choice in a finite volume context is (for $V_h, W_h \in \mathbb{R}^{N_V}$):

$$(V_h, W_h)_{\Omega_h} := V_h^T \Omega_h W_h, \qquad \|V_h\|_{\Omega_h}^2 := (V_h, V_h)_{\Omega_h}, \tag{20}$$

and the discrete energy is defined as $K_h := \frac{1}{2}\|V_h\|_{\Omega_h}^2$. We will also need the unweighted norm $\|V_h\|^2 := V_h^T V_h$. In absence of boundary contributions, the time evolution of K_h is given by

$$2\frac{\mathrm{d}K_h}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t}\left(V_h^T \Omega_h V_h\right)$$
$$= -V_h^T(\tilde{C}_h(V_h)^T + \tilde{C}_h(V_h))V_h - 2p_h^T M_h V_h - 2\nu\|Q_h V_h\|^2. \tag{21}$$

Due to the skew-symmetry property of $\tilde{C}_h$ (equation (12)) and the divergence-freeness of V_h (15), one obtains

$$\frac{\mathrm{d}K_h}{\mathrm{d}t} = -\nu\|Q_h V_h\|^2, \tag{22}$$

which is the semi-discrete counterpart of equation (7).

2.3. Time discretisation and fully discrete energy equation

We continue with the temporal discretisation of equations (15)-(16) with an implicit *s*-stage Runge-Kutta method [41]. The stage values follow from

$$M_h V_h^{n,i} = 0, \tag{23}$$

$$\Omega_h \frac{V_h^{n,i} - V_h^n}{\Delta t} = \sum_{j=1}^{s} a_{ij}(F_h^{CD}(V_h^{n,j}) - G_h p_h^{n,j}), \tag{24}$$

and the solution at the next time step is a combination of the stage values:

$$M_h V_h^{n+1} = 0, \tag{25}$$

$$\Omega_h \frac{V_h^{n+1} - V_h^n}{\Delta t} = \sum_{i=1}^{s} b_i(F_h^{CD}(V_h^{n,i}) - G_h p_h^{n,i}). \tag{26}$$

Here, V_h^n and p_h^n are approximations to $V_h(t^n)$ and $p_h(t^n)$ respectively, which will be collected into a snapshot matrix to be used in the ROM construction (detailed in the next section).

The coefficients a and b of the Runge-Kutta method are chosen such that the temporal discretisation keeps the energy conservation property in the inviscid limit. An example of Runge-Kutta methods that satisfy this property is the family of Gauss methods [41]. We will employ the lowest order Gauss method, obtained for $s = 1$, being the second order implicit midpoint method (corresponding to the following Butcher tableau: $a_{11} = \frac{1}{2}$, $b_1 = 1$). For the Gauss methods, the fully discrete energy equation can be written as (in absence of body forces):

$$\frac{K_h^{n+1} - K_h^n}{\Delta t} = -\nu \sum_{i=1}^{s} \|Q_h V_h^{n,i}\|^2, \tag{27}$$

where K_h^n is an approximation to $K_h(t^n)$. In summary, the careful choice of spatial and temporal discretisation methods has yielded energy equations (22) and (27) that closely mimic the continuous energy estimate (7). The fully discrete energy equation shows that, in the absence of boundary contributions, the energy of the solution can only decrease due to viscous dissipation, independent of the mesh, the time step, or the viscosity.

We note that these implicit methods require the solution of a non-linear system of equations with saddle-point structure, which can be computationally expensive. In section 5.2 we will propose a more practical alternative, by using explicit time integration methods at the FOM level, without compromising the energy-conservation property of the ROM.

3. Energy-conserving POD-Galerkin method

3.1. Introduction

We will follow the ODE-based projection approach [9] in which the POD-Galerkin method is applied to the semi-discrete energy-conserving formulation, i.e. we project the FOM given by equations (15) - (16). This is not the only possibility; one can instead first project the continuous equations (1) and (2) and then discretize the resulting system, common in finite element approaches [34], or project the fully discrete equations [14]. First discretising in space, and then performing the projection of the semi-discrete equations (instead of first projecting and then discretising) has several advantages. Firstly, the treatment of velocity boundary conditions is straightforward, and only has to be done once (when spatially discretising the FOM). Secondly, no pressure boundary conditions are needed, as they are not present in the discrete FOM. Furthermore, in combination with the staggered grid approach, the pressure disappears from the ROM, so that a velocity-only formulation results, and consequently the inf-sup condition is avoided on the level of the ROM.

We make the Ansatz that the velocity field $V_h(t) \in \mathbb{R}^{N_V}$ can be approximated by

$$V_h(t) \approx V_r(t) := \Phi a(t), \tag{28}$$

where $\Phi \in \mathbb{R}^{N_V \times M}$, $a(t) \in \mathbb{R}^M$, and $M \ll N_V$. The subscript r denotes quantities associated to the ROM. Equation (28) is substituted into the FOM and then the equations are projected by left-multiplying with Φ^T. In the POD approach Φ is obtained by performing a singular-value decomposition (SVD) of a snapshot matrix X (this will be detailed below). X contains K snapshots of the velocity field V_h, i.e.

$$X = [V_h^1 \ldots V_h^n \ldots V_h^K], \tag{29}$$

where the snapshots are obtained from the solution of the fully discretised FOM, equations (25)-(26). Each snapshot velocity field is divergence free, so that $M_h X_j = 0$ for each column X_j of X. Φ is subject to the orthonormality condition $\Phi^T \Phi = I$.

3.2. Construction of basis via weighted orthonormality condition

In this work, instead of $\Phi^T \Phi = I$, we use a weighted orthonormality condition, namely

$$\Phi^T \Omega_h \Phi = I. \tag{30}$$

This is consistent with equation (20) and with the form of the ROM momentum equation and the ROM kinetic energy equation, as we will demonstrate next. After substituting (28) in (16) and projecting with Φ^T, one obtains the reduced momentum equation

$$\Phi^T \Omega_h \frac{d\Phi a(t)}{dt} = \Phi^T F_h(\Phi a(t), p_h). \tag{31}$$

The treatment of the pressure in the right-hand side will be discussed in section 3.4. In other words, it is natural to require $\Phi^T \Omega_h \Phi = I_M$ (the identity matrix of dimension $M \times M$), as the ROM then simplifies to

$$\frac{da(t)}{dt} = \Phi^T F_h(\Phi a(t), p_h). \tag{32}$$

In practice, the right-hand side $\Phi^T F_h(\Phi a(t), p_h)$ is rewritten as $F_r(a(t))$, where the ROM operator F_r is precomputed offline. This will be discussed in section 5.1. With condition (30), the energy of the FOM is approximated by

$$K_h(t) \approx K_r(t) = \frac{1}{2}(\Phi a(t))^T \Omega_h \Phi a(t) = \frac{1}{2} a(t)^T \Phi^T \Omega_h \Phi a(t) = \frac{1}{2} a(t)^T a(t) = \frac{1}{2} \|a\|^2. \tag{33}$$

The choice (30) thus simplifies the expression for both momentum and energy considerably.

We now specify the construction of Φ. Given that the energy norm is chosen to be based on (33), the basis Φ should be computed from the following minimization problem [1,36]:

$$\Phi = \arg\min_{\Phi} \|(I - \Phi\Phi^T \Omega_h)X\|_F^2 \qquad \text{subject to} \qquad \Phi^T \Omega_h \Phi = I_M, \tag{34}$$

instead of the classical 'unweighted' minimization problem:

$$\hat{\Phi} = \arg\min_{\Phi} \|(I - \Phi\Phi^T)X\|_F^2 \qquad \text{subject to} \qquad \hat{\Phi}^T \hat{\Phi} = I_M. \tag{35}$$

The solution of the weighted problem can be expressed in terms of the solution of the unweighted problem as follows. Let

$$\Phi = \Omega_h^{-1/2} \hat{\Phi}, \tag{36}$$

where $\hat{\Phi}$ follows from the SVD of the scaled snapshot matrix $\hat{X} = \Omega_h^{1/2} X$:

$$\hat{X} = \hat{\Phi} \Sigma \Psi^*. \tag{37}$$

Since Ω_h is a diagonal matrix, its matrix square root is trivial to compute. The dimensions of the matrices in the SVD are

$$\hat{\Phi}, \Phi \in \mathbb{R}^{N_V \times N_V}, \qquad \Sigma \in \mathbb{R}^{N_V \times K}, \qquad \Psi \in \mathbb{R}^{K \times K}. \tag{38}$$

The columns of $\hat{\Phi}$, denoted by $\hat{\Phi}_j$, are the eigenvectors of the correlation matrix $\hat{X}\hat{X}^T$, i.e.

$$\hat{X}\hat{X}^T \hat{\Phi}_j = \lambda_j \hat{\Phi}_j, \tag{39}$$

where the eigenvalues λ are related to the singular values σ (diagonal entries of Σ) by $\sqrt{\lambda_i(\hat{X}\hat{X}^T)} = \sigma_i(\hat{X})$.

The basis for the ROM is obtained by taking (truncating) the first M columns of Φ. M is typically prescribed by analysing the decay of the singular values σ.

In summary, the sequence to obtain Φ is: gather snapshots of the velocity field in X; compute $\hat{X}$; compute the SVD of $\hat{X}$ to get $\hat{\Phi}$; compute Φ; truncate Φ. Note that for the (common) case that $N_V \gg K$, one can alternatively use the method of snapshots [46] in which an eigenvalue problem for $\hat{X}^T \hat{X}$ is solved, instead of computing the SVD of $\hat{X}$.

3.3. Mass conservation of the ROM

It is well-known that the mass conservation equation is identically satisfied by the ROM approximation, if the boundary conditions are no-slip or periodic [33]. In a finite volume context, this is shown as follows. The divergence-free condition (15) becomes

$$M_h \Phi a(t) = 0. \tag{40}$$

Rewriting equation (39) yields

$$XX^T \Omega_h \Phi_j = \lambda_j \Phi_j. \tag{41}$$

Left-multiplying this equation with M_h and using that the snapshots collected in X are divergence-free ($M_h X_j = 0$) yields

$$\lambda_j M_h \Phi_j = 0. \tag{42}$$

In other words, the ROM velocity field $V_r = \Phi a$ satisfies the divergence-free condition (40), independent of the value of the coefficients $a(t)$, as long as $\lambda_j \neq 0$. Note that for non-homogeneous boundary conditions this is not the case. We will present a boundary condition treatment in section 4.

3.4. Momentum conservation of the ROM

The ROM momentum equation was given by equation (32):

$$\begin{aligned}
\frac{da(t)}{dt} &= \Phi^T F_h(\Phi a(t), p_h), \\
&= \Phi^T(-\tilde{C}_h(\Phi a(t))\Phi a(t) + \nu D_h \Phi a(t) + f_h - G_h p_h), \\
&= \Phi^T(-\tilde{C}_h(\Phi a(t))\Phi a(t) + \nu D_h \Phi a(t) + f_h), \\
&= C_r(a(t)) + \nu D_r a(t) + f_r, \\
&= F_r(a(t)),
\end{aligned} \tag{43}$$

where the convective and diffusive ROM operators C_r and D_r, and the body force vector f_r are precomputed in an offline setting (see section 5.1).

The pressure gradient term disappears because the pressure gradient is linked to the divergence operator according to equation (13):

$$\Phi^T G_h = (G_h^T \Phi)^T = -(M_h \Phi)^T = 0. \tag{44}$$

We stress that the pressure term is not 'neglected' (in the sense of being an approximation); it is really identically equal to zero. This is accomplished via two crucial properties: (i) the FOM velocity snapshots (and thus the basis Φ) are divergence-free, and (ii) the divergence operator and pressure gradient operator are compatible according to equation (13). In other words, the ROM velocity field is by construction divergence-free, and the pressure (a Lagrange multiplier) is not needed anymore to enforce this property. The disappearance of the pressure term leads to a 'velocity-only' ROM. Consequently,

potential issues of inf-sup stability on the reduced order level are avoided. In section 3.5 it will furthermore be shown that this velocity-only ROM possesses non-linear stability.

This 'velocity-only' formulation is also possible in the case of non-homogeneous boundary conditions, including the important case of open flows. This will be discussed in section 4. If the pressure field is desired, it can be recovered via a post-processing step, as will be detailed in section 5.3.

Momentum is, unlike mass, not a locally conserved quantity. However, momentum *is* globally conserved in case of periodic boundary conditions: integration of the incompressible Navier-Stokes equations over a domain Ω with periodic boundary conditions yields

$$\frac{\mathrm{d}\boldsymbol{P}(t)}{\mathrm{d}t} = 0, \qquad \text{where} \qquad \boldsymbol{P}(t) = \int\limits_{\Omega} \boldsymbol{u}\,\mathrm{d}\Omega, \tag{45}$$

so that momentum $\boldsymbol{P}$ is exactly conserved in time. We will require the ROM to satisfy this property. Define the FOM global momentum of each velocity component as

$$P_h^u(t) = e_u^T \Omega_h V_h(t), \tag{46}$$

$$P_h^v(t) = e_v^T \Omega_h V_h(t), \tag{47}$$

where $e_u, e_v \in \mathbb{R}^{N_V}$. e_u contains a 1 for indices associated with the u-velocity component, and e_v contains a 1 for indices associated with the v-velocity component, such that $e = e_u + e_v = [1, 1, \ldots, 1]^T$. Evolution of the u-component of the FOM global momentum is given by

$$\frac{\mathrm{d}P_h^u}{\mathrm{d}t} = e_u^T \Omega_h \frac{\mathrm{d}V_h(t)}{\mathrm{d}t} = e_u^T F_h(V_h, p_h) = 0, \tag{48}$$

with a similar expression for P_h^v. This expression evaluates to zero because of the telescoping property of finite volume methods in combination with a periodic domain.

Evolution of global momentum predicted by the ROM reads

$$\frac{\mathrm{d}P_h^u}{\mathrm{d}t} \approx \frac{\mathrm{d}P_r^u}{\mathrm{d}t} = e_u^T \Omega_h \Phi \frac{\mathrm{d}a(t)}{\mathrm{d}t} = e_u^T \Omega_h \Phi \Phi^T F_h(\Phi a(t), p_h). \tag{49}$$

In order to obtain global conservation of momentum of the ROM, we enforce the basis vectors Φ to satisfy

$$e_u^T \Omega_h \Phi \Phi^T = e_u^T, \tag{50}$$

with a similar expression for e_v. In other words, the projection of the vectors e_u and e_v by $\Omega_h \Phi \Phi^T$ should be exact. When performing the SVD without truncation this property can be easily achieved by adding these vectors to the snapshot matrix, since a property of the SVD is that the projection of vectors in the snapshot matrix is exact. However, upon truncating the decomposition to arrive at a reduced dimension, this property is generally lost. We therefore propose to enforce property (50) by using an approach that enforces linear constraints in constructing the POD basis, as detailed in [56,57]. In short, the idea is to change the minimization problem that underlies the SVD (equation (34)) by a constrained minimization problem, which can be efficiently solved by using a QR-decomposition of the constraint matrix and a projected snapshot set. In the current setting, the constraint matrix is particularly simple, and we can avoid computing the QR-decomposition with the following construction. Furthermore, our analysis is for the weighted norm, instead of the unweighted case [57]. First, we collect the vectors that should be exactly projected by the truncated SVD in the matrix E:

$$E = [e_u\, e_v], \tag{51}$$

scaled to have norm equal to 1. E represents the constraint matrix, which is called $\boldsymbol{G}$ in the notation of [57]. Next, we perform an update of the snapshot matrix X:

$$\tilde{X} = X - EE^T \Omega_h X, \tag{52}$$

and we determine its SVD

$$\tilde{X} = \tilde{\Phi}\tilde{\Sigma}\tilde{\Psi}^*. \tag{53}$$

Subsequently, we add E to $\tilde{\Phi}$, and then we truncate:

$$\Phi = [E\, \tilde{\Phi}]_M, \tag{54}$$

where the subscript M indicates that the first M columns are used. The resulting Φ satisfies equation (50) (and a similar equation for e_v). The proof is given in Appendix C. Note that, when M is given, enforcing global momentum comes at the price of losing two or three of the modes present in $\tilde{\Phi}$ (depending on the spatial dimension of the problem).

The initial condition for a is given by

$$a(0) = \Phi^T \Omega_h V_h(0). \tag{55}$$

Consequently, the initial momentum (u-component) is given by

$$P_r^u(0) = e_u^T \Omega_h \Phi a(0) = e_u^T \Omega_h \Phi \Phi^T \Omega_h V_h(0), \tag{56}$$

whereas the initial momentum of the FOM is

$$P_h^u(0) = e_u^T \Omega_h V_h(0). \tag{57}$$

The error between the two is

$$P_r^u(0) - P_h^u(0) = -e_u^T (I - \Omega_h \Phi \Phi^T) \Omega_h V_h(0) = 0, \tag{58}$$

when the constrained SVD is employed: the total momentum of the ROM is constant in time and equals the total momentum of the FOM, for the case of periodic boundary conditions. The same equation holds for the v-component. Upon time discretisation, no additional measures need to be taken to ensure that momentum stays conserved, since it is a linear constraint which is conserved by any consistent time integration method. The fully discrete momentum error is thus:

$$\epsilon_P := P_r^{u,n} - P_h^{u,n} = \underbrace{P_r^{u,n} - P_r^u(0)}_{\text{time integration error ROM}} + \underbrace{P_r^u(0) - P_h^u(0)}_{\text{projection error, (58)}} + \underbrace{P_h^u(0) - P_h^{u,n}}_{\text{time integration error FOM}} = 0. \tag{59}$$

We note that if the constrained SVD is *not* employed, the third term remains 0, but the first and second term both become nonzero. The first term scales with the right-hand side of (49) (and the order of accuracy of the ROM time integration). The second term scales with $e_u^T (I - \Omega_h \Phi \Phi^T) \Omega_h V_h(0)$.

Note that this global momentum-conserving construction is different from the conservative model reduction method from Carlberg et al. [14]. We consider global momentum conservation for the case of periodic boundary conditions, by adapting the construction of the SVD, whereas in [14] a constrained optimization problem is considered, which minimizes the residual of the full-order model over subdomains. In our view, conservation means that the integral of a certain quantity (mass, momentum, energy) stays invariant in time, which only holds for particular boundary conditions; in [14] the term conservation is used to indicate the difference between the rate of change of a conserved quantity and the contribution of surface integrals and source terms. Furthermore, our method differs in the fact that we are not only considering primary conserved quantities, but are considering kinetic energy conservation (a so-called secondary or 'derived' quantity) which has the advantage that it provides a non-linear stability bound to the solution. This is detailed in the next section.

3.5. Energy conservation of the ROM

One of the key questions in this paper is whether the kinetic energy of the ROM can be bounded in a similar way as the energy of the FOM (equations (7), (22), (27)). To this end, we differentiate the expression for the energy of the ROM, as given by equation (33), and simplify by using equations (40) and (43) (in absence of body forces):

$$\begin{aligned} 2\frac{dK_r(t)}{dt} &= \frac{da^T}{dt}a + a^T\frac{da}{dt}, \\ &= -(\Phi^T \tilde{C}_h(\Phi a)\Phi a)^T a - a^T(\Phi^T \tilde{C}_h(\Phi a)\Phi a) + \nu(\Phi^T D\Phi a)^T a + a^T \nu \Phi^T D_h \Phi a, \\ &= -a^T \Phi^T(\tilde{C}_h(\Phi a)^T + \tilde{C}_h(\Phi a))\Phi a - 2\nu\|Q_h \Phi a\|^2, \\ &= -2\nu\|Q_h \Phi a\|^2, \\ &= -2\nu\|Q_r a\|^2. \end{aligned} \tag{60}$$

The crucial steps in the derivation are the fact that the basis is divergence-free ($M_h\Phi = 0$) and that the following properties of the FOM spatial discretisation operators hold: $G_h^T = -M_h$; $\tilde{C}_h(\Phi a)$ is skew-symmetric; D_h is symmetric negative definite. In terms of the offline ROM operators (detailed in section 5.1) one can equivalently state that each element $C_{r,i}$ of the convective operator C_r is skew-symmetric, and that the diffusive operator D_r is symmetric negative definite. This last observation allows us to write D_r as $-Q_r^T Q_r$, with $Q_r = Q_h \Phi$.

In summary, the energy evolution of the ROM in absence of boundary contributions and body forces is given by

$$\boxed{\frac{dK_r(t)}{dt} = -\nu\|Q_r a\|^2.} \tag{61}$$

Consequently, *the ROM is non-linearly stable, independent of the number of POD modes used.*

In the inviscid limit, the energy of the ROM stays constant in time:

$$K_r(t) = K_r(0),$$ (62)

where

$$K_r(0) = \frac{1}{2}a(0)^T a(0) = \frac{1}{2}(\Phi^T \Omega_h V_h(0))^T \Phi^T \Omega_h V_h(0) = \frac{1}{2}V_h(0)^T \Omega_h \Phi \Phi^T \Omega_h V_h(0).$$ (63)

The FOM kinetic energy is given by

$$K_h(0) = \frac{1}{2}V_h(0)^T \Omega_h V_h(0),$$ (64)

and the error in the ROM (due to truncation) is therefore given by

$$K_r(0) - K_h(0) = -\frac{1}{2}V_h(0)^T(I - \Omega_h \Phi \Phi^T)\Omega_h V_h(0).$$ (65)

Similar to the addition of the global momentum constraint, it is possible to add $V_h(0)$ to the truncated SVD so that it is projected exactly by $\Omega_h \Phi \Phi^T$. This guarantees that in the inviscid case the kinetic energy of the ROM remains equal to that of the FOM. However, for viscous simulations, the ROM will not reproduce the kinetic energy evolution exactly (compare equation (61) to (22)), and so we do not enforce this property.

The last step in obtaining the ROM is to specify a time discretisation for equation (32) such that a fully discrete equivalent of (33) is obtained. The key is, not surprisingly, to use the energy-conserving Runge-Kutta time discretisation methods introduced in section 2.2. For example, the implicit midpoint method applied to (32) reads

$$\frac{a^{n+1} - a^n}{\Delta t} = F_r(a^{n+1/2}),$$ (66)

where $a^{n+1/2} = \frac{1}{2}(a^n + a^{n+1})$. This non-linear equation is solved efficiently by using Newton's method; this requires the Jacobian of $F_r(a)$, which is given in section 5.1. The corresponding energy evolution is

$$\boxed{\frac{K_r^{n+1} - K_r^n}{\Delta t} = -\nu \|Q_r a^{n+1/2}\|^2,}$$ (67)

which is strictly decreasing in time when the viscosity ν is nonzero, and hence *the fully discrete ROM solution is stable*.

The energy equality (67) is derived for homogeneous (no-slip, periodic) boundary conditions and in the absence of body forces. For more generic boundary conditions, such as inflow conditions, the energy of the flow is not strictly decreasing in time. In this case, the ROM estimate should still be such that it mimics the energy estimate of the FOM (see e.g. [44] for the boundary contributions to the energy equation). Further details on general boundary conditions are given in section 4.

The energy-conserving property expressed by equations (61) and (67) is *independent* of whether the snapshot matrix has been generated using an energy-conserving discretisation method. The only condition on the snapshot matrix is that the snapshots are divergence-free. Of course, when considering the error of the fully discrete ROM with respect to the fully discrete FOM, the FOM energy error *is* present:

$$\epsilon_K := K_r^n - K_h^n = \underbrace{K_r^n - K_r(0)}_{\text{time integration error ROM}} + \underbrace{K_r(0) - K_h(0)}_{\text{projection error}} + \underbrace{K_h(0) - K_h^n}_{\text{time integration error FOM}}.$$ (68)

The first term of this expansion is zero when employing the energy-conserving ROM satisfying (67) (for $\nu = 0$). The second term is generally nonzero and depends on the number of modes used and the decay of the singular values. The last term is zero when an energy-conserving time integration method is used on the FOM level.

4. Non-homogeneous boundary conditions

In this section we extend the results of the previous section to the more generic case of (stationary) non-homogeneous boundary conditions and forcing terms, which is known to be non-trivial (see for example [30,34,55]). One of the main issues is the pressure term, whose contribution does not vanish [34]; in the kinetic energy equation, this gives the term

$$(\boldsymbol{u}, \nabla p) = \int_{\partial \Omega} p \, \boldsymbol{u} \cdot \boldsymbol{n} \, \mathrm{d}S,$$ (69)

and in the Galerkin projection a similar term appears. In the spirit of [7,17,22,25], we will split the velocity field into two components: a time-dependent field $\boldsymbol{u}_{\text{hom}}(\boldsymbol{x}, t)$ that satisfies homogeneous boundary conditions, and a stationary field $\boldsymbol{u}_{\text{bc}}(\boldsymbol{x})$ that satisfies non-homogeneous boundary conditions,

$$\boldsymbol{u}(\boldsymbol{x}, t) = \boldsymbol{u}_{\text{hom}}(\boldsymbol{x}, t) + \boldsymbol{u}_{\text{bc}}(\boldsymbol{x}). \tag{70}$$

The non-homogeneous term is also known as a lifting function [19,24]. Both terms satisfy $\nabla \cdot \boldsymbol{u}_{\text{hom}}(\boldsymbol{x}, t) = 0$ and $\nabla \cdot \boldsymbol{u}_{\text{bc}}(\boldsymbol{x}) = 0$. In the case of Dirichlet boundary conditions we have $\boldsymbol{u}_{\text{bc}}(\boldsymbol{x}) = \boldsymbol{u}_{\partial \Omega}$ and $\boldsymbol{u}_{\text{hom}}(\boldsymbol{x}) = \boldsymbol{0}$ on $\partial \Omega$. The stationary field is found from $\boldsymbol{u}_{\text{bc}} = \nabla q$, where $q(\boldsymbol{x})$ follows from solving the Poisson equation

$$\nabla^2 q(\boldsymbol{x}) = 0 \qquad \text{with} \qquad \nabla q = \boldsymbol{u}_{\partial \Omega} \quad \text{on} \quad \partial \Omega. \tag{71}$$

Subsequently, with this construction in the kinetic energy equation for $\boldsymbol{u}_{\text{hom}}$ the pressure term contribution still disappears:

$$(\boldsymbol{u}_{\text{hom}}, \nabla p) = \int_{\partial \Omega} p \, \boldsymbol{u}_{\text{hom}} \cdot \boldsymbol{n} \, dS = 0. \tag{72}$$

With this insight, a ROM formulation incorporating non-homogeneous boundary conditions is constructed, but starting from the semi-discrete instead of the continuous equations. The semi-discrete equations (15)-(16) with non-homogeneous boundary conditions read:

$$M_h V_h(t) = y_M, \tag{73}$$

$$\Omega_h \frac{dV_h(t)}{dt} = F_h^{CD}(V_h(t)) - (G_h p_h(t) + y_G), \tag{74}$$

with

$$F_h^{CD}(V_h(t)) = -C_h(V_h(t), V_h(t)) + \nu(D_h V_h(t) + y_D) + f_h. \tag{75}$$

The boundary terms such as $y_M \in \mathbb{R}^{N_p}$, $y_G, y_D \in \mathbb{R}^{N_V}$ are described for example in [44] and summarized in Appendix B.1. This formulation holds for both inflow, outflow, symmetry, periodic, and no-slip conditions, and also encompasses the case of body forces via the term f_h. In order to retain a velocity-only ROM formulation, the two properties mentioned in section 3.4 need to be satisfied: the basis Φ should be divergence-free, and the pressure gradient and divergence operator should be compatible.

The first property (divergence-free basis) is achieved as follows. We approximate the FOM velocity field with a ROM with homogeneous boundary conditions and a term that incorporates the boundary conditions [22]:

$$V_h(t) \approx V_r(t) + V_{bc} = \Phi a(t) + V_{bc}. \tag{76}$$

This is a generalization of the ROM in section 3, which corresponds to the case $V_{bc} = 0$. V_{bc} is chosen such that

$$M_h(V_r(t) + V_{bc}) = y_M, \tag{77}$$

which reduces to $M_h V_{bc} = y_M$, since $V_r(t)$ satisfies homogeneous boundary conditions. The solution for V_{bc} is the discrete version of equation (71), namely

$$V_{bc} = \Omega_h^{-1} G_h \zeta_h, \qquad \text{where} \qquad L_h \zeta_h = y_M. \tag{78}$$

Thus, the solution of one Poisson equation at the FOM level is needed to find the boundary terms V_{bc}, which is best performed during the offline FOM stage, when the Poisson equation is already being solved many times. This V_{bc} is subtracted from the velocity snapshots V_h in order to arrive at the snapshot matrix of V_r:

$$X = [V_h^1 - V_{bc} \ldots V_h^n - V_{bc} \ldots V_h^K - V_{bc}], \tag{79}$$

which satisfies $M_h X_j = 0$ for each column j of X, and the projection matrix Φ thus satisfies $M_h \Phi_j = 0$.

The second property (compatibility of divergence and gradient operators) is naturally achieved on a staggered grid when the boundary conditions are periodic or no-slip (the extension to inflow conditions and symmetry conditions is straightforward [44]). The case of outflow conditions is more involved. We prescribe the normal component of the stress tensor [11]:

$$(-p\boldsymbol{I} + \nu \nabla \boldsymbol{u}) \cdot \boldsymbol{n} = \begin{pmatrix} -p_\infty \\ 0 \end{pmatrix}, \tag{80}$$

where $p_\infty(\boldsymbol{x}, t)$ can be space- and time-dependent (it will be taken 0 in our open-flow test case). The finite volumes adjacent to the outflow boundary feature this term. Consequently, the expression for the pressure gradient is

$$G_h p_h(t) + y_G, \tag{81}$$

where $y_G \in \mathbb{R}^{N_V}$ contains the value of p_∞ in the entries corresponding to an outflow boundary volume. The divergence-gradient relation $G_h = -M_h^T$ (equation (13)) is then still satisfied (for details, see [44]). In the ROM, the pressure contribution thus becomes

$$\Phi^T (G_h p_h(t) + y_G) = -p_h^T(t) M_h \Phi + \Phi^T y_G = \Phi^T y_G, \tag{82}$$

which can be easily precomputed. Consequently, the ROM formulation remains 'velocity-only', just like in the homogeneous case:

$$\Phi^T \Omega_h \frac{\mathrm{d}}{\mathrm{d}t} (\Phi a(t) + V_{bc}) = \frac{\mathrm{d}a(t)}{\mathrm{d}t} = F_r(a(t)), \tag{83}$$

where precomputing F_r is now slightly more involved due to the presence of V_{bc}:

$$F_r(a(t)) = \Phi^T \left(-C_h(\Phi a(t) + V_{bc}, \Phi a(t) + V_{bc}) + \nu (D_h(\Phi a(t) + V_{bc}) + y_D) - y_G + f_h \right). \tag{84}$$

This construction is detailed in Appendix B.2.

Note that for the case of time-dependent or parameter-dependent boundary conditions appearing in y_M, the method needs to be extended by incorporating pressure snapshots and a projection of the divergence-free constraint. Although this is the subject of future work, in section 5.3 we will give details on the projection of the divergence-free constraint in order to formulate a pressure Poisson equation for the recovery of the pressure.

5. Implementation aspects and pressure recovery

5.1. Offline decomposition

The semi-discrete and fully discrete ROM momentum equations feature the right-hand side (see equations (32) and (66))

$$F_r(a) := \Phi^T F_h(\Phi a) = \Phi^T (-\tilde{C}_h(\Phi a)\Phi a + \nu D_h \Phi a + f_h). \tag{85}$$

To be efficient as a ROM, this term should be evaluated such that it does not depend on operations on the FOM-level. This is performed by precomputing the convective and diffusive operators as follows. For ease of notation, we discuss the case of homogeneous boundary conditions. The non-homogeneous case is discussed in Appendix B.

The projection of the (linear) diffusive operator is straightforward and can be written as

$$\Phi^T D_h(\Phi a) = D_r a, \tag{86}$$

where $D_r = \Phi^T D_h \Phi \in \mathbb{R}^{M \times M}$. The projection of the nonlinear convective operator is more involved, but given that the nonlinearity is only quadratic, it can be written as

$$\Phi^T \tilde{C}_h(\Phi a)(\Phi a) = C_r(a \otimes a) = \begin{bmatrix} C_{r,1} & \dots & C_{r,M} \end{bmatrix} (a \otimes a), \tag{87}$$

where $C_r \in \mathbb{R}^{M \times M^2}$ is a "matricized" third order tensor, consisting of M components $C_{r,i} \in \mathbb{R}^{M \times M}$:

$$C_{r,i} = \Phi^T \tilde{C}_h(\Phi_i)\Phi. \tag{88}$$

In the current work, similar to [48], the number of modes that we require for an accurate ROM is sufficiently low to warrant that storing and pre-computing this third order tensor is not prohibitive. If many modes would be required, if higher-order or non-polynomial nonlinearities are present in the equations (e.g. present in a source term), or if the discretisation of the quadratic term is nonlinear (e.g. a flux-limiting scheme), precomputing as high-order tensors is not possible or not efficient. In such cases, one can resort for example to the discrete empirical interpolation method (DEIM) [16], or a lifting procedure [29]. Designing a DEIM procedure such that the skew-symmetry of the convective operator is retained will be subject of future work.

The entire semi-discrete ROM formulation for both homogeneous and non-homogeneous boundary conditions can be summarized as

$$\frac{\mathrm{d}a}{\mathrm{d}t} = F_r(a) = F_{r,2}(a \otimes a) + F_{r,1}a + F_{r,0}, \tag{89}$$

where $F_{r,0} \in \mathbb{R}^M$ contains all terms that are independent of the solution (boundary conditions, source terms), $F_{r,1} \in \mathbb{R}^{M \times M}$ contains linear terms (the diffusive operator and the interaction of the convective operator with the inhomogeneous boundary conditions), and $F_{r,2} \in \mathbb{R}^{M \times M^2}$ contains the quadratic terms of the convective operator. The Jacobian of F_r with respect to a, as needed for implicit time integration (equation (66)), is given by

$$J_r = \frac{\partial F_r}{\partial a} = F_{r,2}(I_M \otimes a + a \otimes I_M) + F_{r,1}, \tag{90}$$

where I_M is the $M \times M$ identity matrix.

5.2. Alternative time integration methods and linear stability

From an implementation point of view, the implicit time integration methods proposed (on both the FOM and ROM level) might seem daunting at first sight. However, one should note that the Jacobian (90) is rather simple to form (because the nonlinearity is only quadratic). Furthermore, because the pressure is absent in the ROM, one does *not* need to solve a coupled system of saddlepoint type, as is the case on the FOM level.

In any case, one can think of test cases in which explicit time integration (at FOM and/or ROM level) is more efficient than implicit time integration. In that case, we propose the following strategy. First, solve the FOM with a time integration method of choice (e.g. explicit, IMEX, etc.) – this does *not* affect energy conservation of the ROM as long as the FOM spatial discretization is energy-conserving. Second, use linear stability theory to estimate the eigenvalues of the ROM operator and use this to determine an efficient time integrator and time step for the ROM. Since each $C_{r,i}$ is a skew-symmetric matrix, it has purely imaginary eigenvalues; similarly, since D_r is a symmetric negative definite matrix, its eigenvalues are negative and real. Gershgorin's theorem can then be used to find a bound on the eigenvalues of the combined operator from which the time step can be determined [51]. Although in this second step the non-linear stability property is strictly speaking lost, linear stability of the ROM in combination with adaptive high-order time stepping can result in a negligible energy error, which could form an efficient alternative to implicit time integration. Of course, this depends on the application under consideration. In the test cases in section 6, several simulations with explicit Runge-Kutta methods (for both FOM and ROM) will be performed to show the potential of this alternative. The explicit Runge-Kutta method applied to the FOM reads [42]:

$$\Omega_h \hat{V}_h^{n,i} = \Omega_h V_h^n + \Delta t \sum_{j=1}^{i} \tilde{a}_{ij}(F_h^{CD}(V_h^{n,j}) - G_h p_h^{n,j}), \tag{91}$$

$$L_h p_h^{n,i} = M_h \hat{V}_h^{n,i} - y_M, \tag{92}$$

$$\Omega_h V_h^{n,i} = \Omega_h \hat{V}_h^{n,i} - G_h p_h^{n,i}. \tag{93}$$

Here $\tilde{a}$ denotes the shifted Butcher tableau, which includes the b coefficients. For details, we refer to [42]. The application of an explicit Runge-Kutta method to the ROM is straightforward, as the pressure is absent.

5.3. Recovering the pressure

The pressure is not part of the ROM formulation (43), because the reduced velocity field is by construction divergence-free. In principle one could recover the pressure once the velocity field is known (via $V_h \approx V_r = \Phi a$) by solving a Poisson equation at the FOM level, but then the computational costs scale with the FOM dimension. Instead, we aim for pressure recovery at the ROM level. Roughly speaking, two approaches exist [28]: (i) solve a pressure Poisson equation (PPE) on the ROM level, or (ii) use the momentum equation recovery formulation with supremizer stabilization. In this work we follow the PPE strategy, as it naturally fits in the proposed framework.

The ROM - PPE can be derived in two ways. Firstly, one can combine the divergence-free constraint and momentum equation on the ROM level: project the divergence equation based on a suitable pressure space, take its time derivative, and substitute the projected momentum equation. However, given that the velocity basis is divergence-free, this leads to a singular pressure equation, for the same reason that the pressure disappeared in equation (43). The second approach, which we follow, is to combine the divergence-free constraint and momentum equation on the FOM level, to obtain the corresponding PPE on the FOM level (equation (19)), and then project it to the ROM level [4].

In order to obtain a pressure estimate on the ROM level, we collect full-order snapshots of the pressure field in order to approximate the pressure in terms of

$$p_h(t) \approx p_r(t) = \Pi q(t), \tag{94}$$

where $\Pi \in \mathbb{R}^{N_p \times M_p}$ is constructed from a truncated SVD of snapshots of p_h (similar to the construction of Φ), such that $\Pi^T \Omega_p \Pi = I_{M_p}$, where M_p is the number of pressure modes and Ω_p is a diagonal matrix containing the size of the pressure volumes. The velocity and pressure snapshots and modes are thus considered in a decoupled fashion [11]. It is possible to include the temporal mean of the snapshots of the pressure in equation (94), like in [4], but we did not observe significant differences in the results in our test cases. Substituting the approximation for p_h into equation (19) and projecting with Π^T gives the PPE-ROM:

$$L_r q(t) = \Pi^T M_h \Omega_h^{-1} F_h^{CD}(\Phi a(t)), \tag{95}$$

where $L_r \in \mathbb{R}^{M_p \times M_p}$ is the reduced Poisson operator, given by

$$L_r = \Pi^T L_h \Pi = \Pi^T M_h \Omega_h^{-1} G_h \Pi = -(G_h \Pi)^T \Omega_h^{-1}(G_h \Pi), \tag{96}$$

which is symmetric negative definite, like L_h, although it is not sparse. Like for the full PPE equation, explained in section 2.2, no pressure boundary conditions are needed in (95), as they were built into the divergence-free constraint. We

stress again that this has been accomplished by deriving the PPE based on the spatially discrete mass and momentum equations. Compared to the derivation of the ROM-PPE based on projection at the continuous level, see e.g. [28] (where boundary conditions need to be prescribed for the PPE), this is a strong advantage of our approach.

In order to evaluate equation (95) using only ROM operators, a second offline decomposition is required, similar to what is proposed in [11]. This decomposition is constructed exactly the same as the one described in section 5.1, with the main difference that instead of pre-multiplying with Φ^T, we pre-multiply with $\Pi^T M_h \Omega_h^{-1} \in \mathbb{R}^{M_p \times N_V}$.

6. Results

In this section we show the results of three test cases. In the first test case, we demonstrate the stability and momentum and energy conservation properties of the ROM through an inviscid simulation of a shear-layer roll-up. In the second test case we consider the simulation of a lid-driven cavity, a common test case used in the ROM community, for which several stabilization techniques have been tested. We will show that, independent of the number of modes, no stabilization method is needed in our approach, and that explicit time integration methods can be efficiently employed. In the third test case, we demonstrate the treatment of non-homogeneous boundary conditions (including outflow conditions), and unsteady forcing terms by simulating the unsteady wake behind an actuator disk. We will show significant speed ups and accurate pressure recovery.

In all test cases, the number of modes M_p for the pressure basis is taken equal to the number of velocity modes M. Note that the pressure is only computed as a post-processing step at the end of each time step; it is not needed in the ROM formulation to advance the velocity coefficients. The natural error norm to measure the accuracy of the ROM with respect to the FOM is the weighted L_2-norm, see (20). This is the vector norm that corresponds to the Frobenius norm being minimized in equation (34). This norm is scaled with the norm of a characteristic velocity V_{ref}. The error in the velocity at each time step is thus computed as

$$\epsilon_V^n = \frac{\|V_r^n - V_h^n\|_{\Omega_h}}{\|V_{\text{ref}}\|_{\Omega_h}}. \tag{97}$$

We note that the error between ROM and FOM is not only due to projection when different time integration methods are used on the ROM and FOM level. In that case, the error can be decomposed as:

$$V_r^n - V_h^n = \underbrace{V_r^n - V_r(t^n)}_{\text{time integration error ROM}} + \underbrace{V_r(t^n) - V_h(t^n)}_{\text{projection error}} + \underbrace{V_h(t^n) - V_h^n}_{\text{time integration error FOM}}, \tag{98}$$

which illustrates that the projection is performed at the semi-discrete level (see Fig. 1). Note that in the second and third test case in this paper, these errors are absent, as the ROM and FOM use the same time integration method, so that we effectively project the fully discrete equations. In the first test case we study the error in momentum and energy, equations (59) and (67).

Similar to (97), the ROM pressure error is defined as

$$\epsilon_p^n = \frac{\|p_r^n - p_h^n\|_{\Omega_p}}{\|p_{\text{ref}}\|_{\Omega_p}}. \tag{99}$$

Since the pressure in the incompressible Navier-Stokes equations is determined up to a constant, both the ROM and FOM pressure fields are shifted to have the same spatial mean. We will also report the basis projection error in the second and third test case:

$$\epsilon_{V,\text{best}}^n = \|V_{\text{best}}^n - V_h^n\|_{\Omega_h}, \tag{100}$$

where V_{best} is obtained by projecting the FOM solution onto the ROM basis, i.e. $V_{\text{best}}^n = \Phi \Phi^T \Omega_h V_h^n$.

An important aspect of the ROM is the computational speed-up compared to the FOM. We will split the ROM time in three components: (1) (offline) computation of the basis (via SVD); (2) (offline) precomputing the ROM operators, (3) (online) time-stepping of the ROM. The time needed to compute the error of the ROM with respect to the FOM will be excluded from the online timings, as it can be computed as a postprocessing step (as long as the time-dependent ROM coefficients are stored while marching in time). The basis construction via SVD could be accelerated by using the method of snapshots, but for the relatively small (two-dimensional) problems considered here, the computational gain is not significant.

6.1. Shear-layer roll-up

We simulate the roll-up of a shear-layer, similar to [44]. The simulation domain is $[0, 2\pi] \times [0, 2\pi]$, with periodic boundary conditions and the following initial condition:
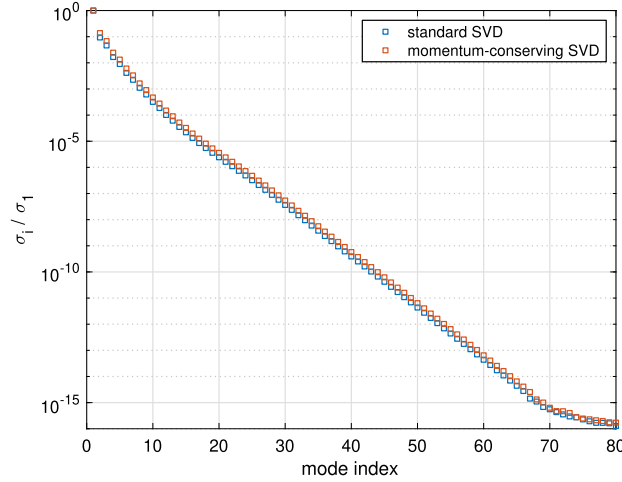
Fig. 3. Singular values for inviscid shear-layer roll-up.

$$u_0(x, y) = 1 + \begin{cases} \tanh(\frac{y-\pi/2}{\delta}), & y \le \pi, \\ \tanh(\frac{3\pi/2-y}{\delta}), & y > \pi, \end{cases} \qquad v_0(x, y) = \epsilon \sin(x), \tag{101}$$

where $\delta = \pi/15$ and $\epsilon = 1/20$. Compared to [44], a constant has been added to $u_0(x, y)$, in order to ensure that the global momentum of the u- and v- components differ. In the inviscid case, the energy of the flow should be exactly conserved. The FOM discretisation consists of 200×200 finite volumes, giving a total of $N_V + N_p = 1.2 \cdot 10^5$ unknowns. Time integration of the FOM is performed with explicit RK4 [42] with a time step of $\Delta t = 0.01$ from $t = 0$ to $t = 4$, resulting in $K = 401$ snapshots (as mentioned in section 5.2, the FOM snapshots need not be energy-conserving). The singular values of the velocity snapshot matrix are shown in Fig. 3. The ROM basis consists of the first M left singular vectors of the snapshot matrix, where we take $M = 2, 4, 8, 16$. The rapid decay in the singular values indicates that the problem is suited for dimension reduction. The effect of using the proposed momentum-conserving SVD of section 3.4 instead of the standard SVD is a small shift in the singular values.

Time integration of the ROM is performed with the implicit midpoint method, with the same Δt and end time as used for the FOM. The energy error between fully discrete ROM and FOM can be decomposed according to equation (68):

$$\epsilon_K := K_r^n - K_h^n = \underbrace{K_r^n - K_r(0)}_{\text{time integration error ROM}} + \underbrace{K_r(0) - K_h(0)}_{\text{projection error}} + \underbrace{K_h(0) - K_h^n}_{\text{time integration error FOM}}. \tag{102}$$

We are interested in the first two terms, whose sum is shown in Fig. 4a (the last error term is nonzero, as the FOM time integration is explicit RK4). The first term remains at machine-precision zero, as is further detailed in Fig. 5. The ROM is therefore conserving kinetic energy: $K_r^n = K_r(0)$, independent of whether the standard SVD or the momentum-conserving SVD is used. The error displayed in the figure thus corresponds to the second term: $K_r(0) - K_h(0)$, which is due to projecting the initial FOM velocity field onto the truncated snapshot basis, and decreases with increasing M according to the singular value decay. We observe that, especially for small M, the momentum conserving approach is less accurate in terms of the energy error. This is because two modes have been sacrificed in order to achieve momentum conservation. For the case $M = 2$ this means that momentum is enforced, but that the FOM snapshots are not taken into account in the basis Φ.

In Fig. 4b the momentum error of the u-component is plotted as a function of time. For the standard SVD, the error in global momentum increases as a function of time, and decreases when more modes are taken. With the momentum-conserving SVD, the error in momentum stays at machine precision, independent of the number of modes. As noted in section 3.5, addition of the initial velocity field as constraint in the SVD can also force the kinetic energy error to zero, independent of M. This is not considered here, as it would not generalize to the case of viscous flows.

Fig. 5 further details the energy error of the ROM when excluding the projection error, i.e. we plot the first term of equation (102): $K_r^n - K_r(0)$. For implicit midpoint this error remains at machine precision, as expected. When changing the ROM time integration to an explicit (fourth-order Runge-Kutta) method, an energy error is introduced which grows steadily in time. However, given that this error is insignificant compared to the projection error shown in Fig. 4a, RK4 can be a practical alternative given that it is faster to execute on a per-time step basis than the implicit midpoint method (provided that the time-step restriction on the explicit method is not too severe). This is shown in Fig. 6, which compares the computational cost of the ROM to that of the FOM, for both implicit and explicit time integration of the ROM. For $M \le 8$ we have a computational speed-up factor of at least 50, if we include the offline phase in the speed-up computation. When not considering the offline phase, the speed-up factor is much larger: around 400 for implicit midpoint, and almost 1000 for explicit RK4. We note that $M \le 8$ is realistic from the viewpoint of accuracy: for $M = 8$ we have a relative energy error
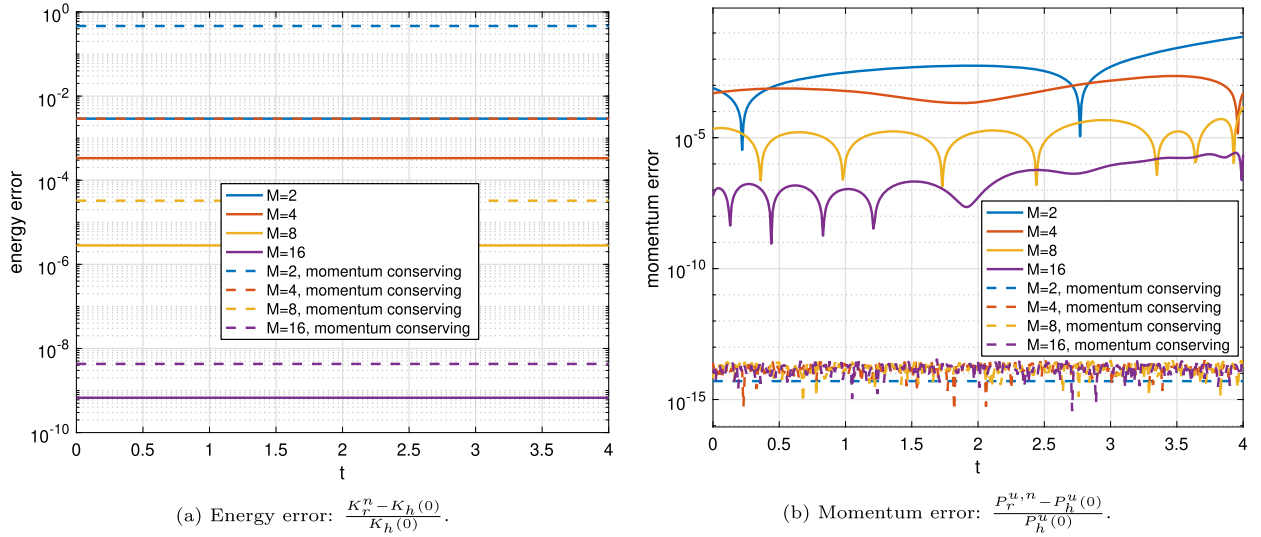
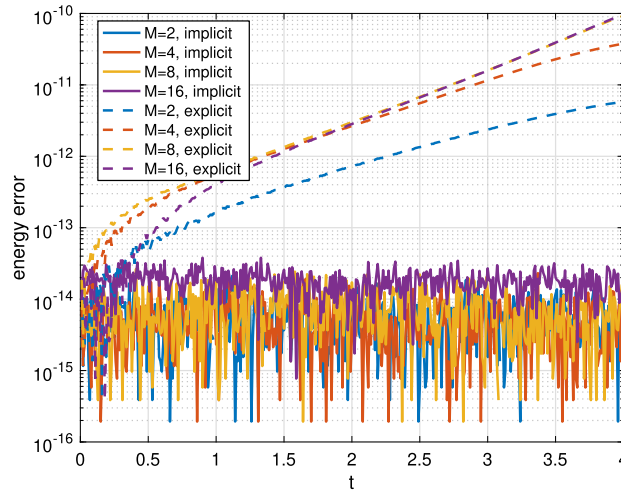**Fig. 4.** Energy and momentum conservation of ROM with respect to FOM initial condition.

Fig. 5. Energy error of the ROM with respect to ROM initial condition, $\frac{K_r^n - K_r(0)}{K_r(0)}$, for implicit midpoint and explicit RK4.

which is less than 10^{-5} (Fig. 4a). For $M > 10$, the computational costs of the ROM are dominated by the offline phase (precomputing the operators, especially the convective operator); the SVD and the online phase are negligible compared to the precomputing phase. Given the small energy error with explicit RK4, and the fact that it is significantly faster to execute, the next test cases will be performed with explicit RK4 for both the FOM and ROM.

6.2. Lid-driven cavity

We perform as second test case a common test used to assess ROMs for the incompressible Navier-Stokes equations (see e.g. [18,48]): a lid-driven cavity flow at Re = 1, 000. The velocity of the lid enters as boundary contribution in y_D, but does not appear in y_M in the divergence-free constraint, since y_M only contains velocity components normal to the boundary. Consequently, in this test case the procedures described in section 4 are not required ($V_{bc} = 0$). No measures need to be taken to ensure stability of the ROM, as it is stable by design.

The full-order model is run on a grid with 100×100 volumes, resulting in $N_V + N_p = 3 \cdot 10^4$ unknowns, at a time step of $\Delta t = 0.01$, until a final time $T = 10$. This results in $K = 1, 001$ snapshots. Note that, as discussed in sections 5.2 and 6.1, both FOM and ROM are now integrated with an explicit RK4 method, in order to show that in practice explicit methods can be an excellent alternative to the implicit methods proposed in section 3.5. In contrast to e.g. [48], stable and accurate results are obtained without requiring a stabilization method.
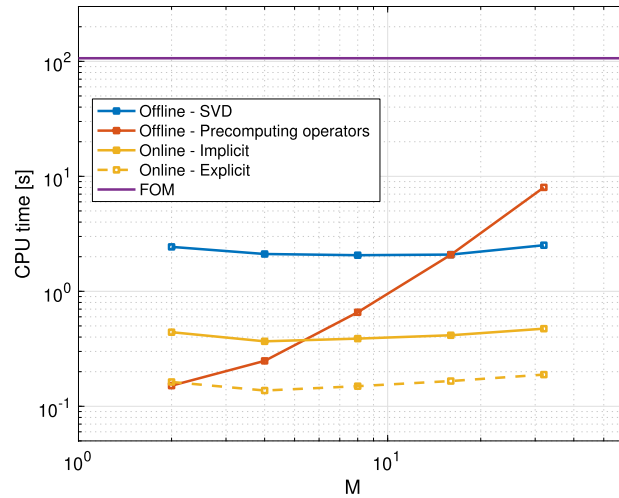
Fig. 6. Computational time as a function of number of modes in ROM for inviscid shear-layer roll-up.

(a) Velocity: ϵ_V^n.   (b) Pressure: ϵ_p^n.
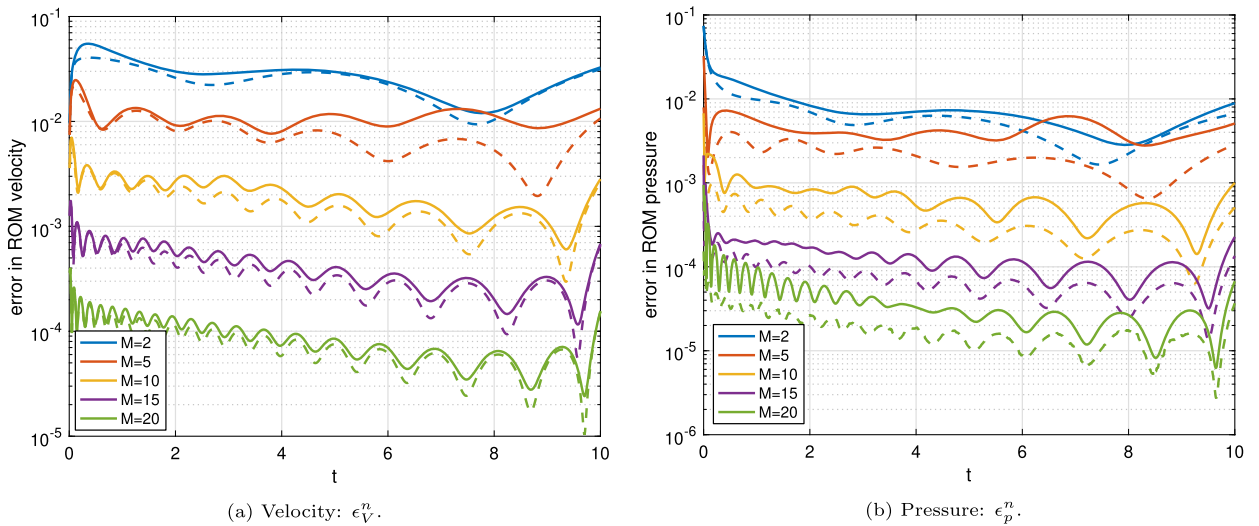
Fig. 7. Error in ROM with respect to the FOM as a function of time for different number of modes for the lid-driven cavity flow. Dashed lines: basis projection error (projecting snapshots onto truncated basis).

The error with respect to the FOM is given by equation (97) where $V_{\mathrm{ref}} = 1$ (the lid velocity). We study the effect of increasing the number of modes on the accuracy of the velocity field, while using the full snapshot set as basis for the SVD; see Fig. 7a. We clearly see how the accuracy increases when increasing the number of modes. Furthermore, the error in the ROM is very close to the 'best possible' error, being the basis projection error, equation (100). The behaviour for the accuracy of the pressure, using the pressure recovery method described in section 5.3, is very similar, as displayed in Fig. 7b. The error is computed from (99) with $p_{\mathrm{ref}} = 1$, which is chosen according to the typical pressure differential between the upper-right and upper-left corner of the cavity.

Note that the kinetic energy is not conserved in this test case, but rather increases as a function of time, as kinetic energy is added to the flow through the moving lid, which is initially larger than the energy dissipation in the interior of the cavity. When the flow reaches a steady state, the two effects will balance each other.

In Fig. 8 the computational time required by the ROM is compared to the FOM. Like in the previous test case, the speedup of the ROM compared to the FOM is significant: the online phase is about 100 times faster than the FOM. However, for a large number of modes ($M \geq 40$), precomputing the operators becomes significant and is dominating the ROM cost. In practice, such a large number of modes is not needed, since for $M = 15$ the error in velocity and pressure is already smaller than 10^{-3} for almost the entire simulation. Fig. 8 also reveals that the computational costs associated with pressure recovery is significant (dashed lines): it adds roughly a factor of 2 to the offline stage, because it requires an additional SVD and an additional set of precomputed operators.
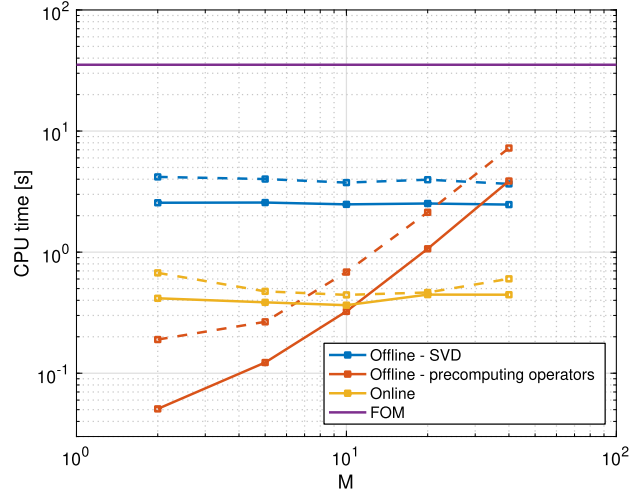
Fig. 8. Computational time as a function of number of modes in ROM. Dashed lines: including pressure recovery.
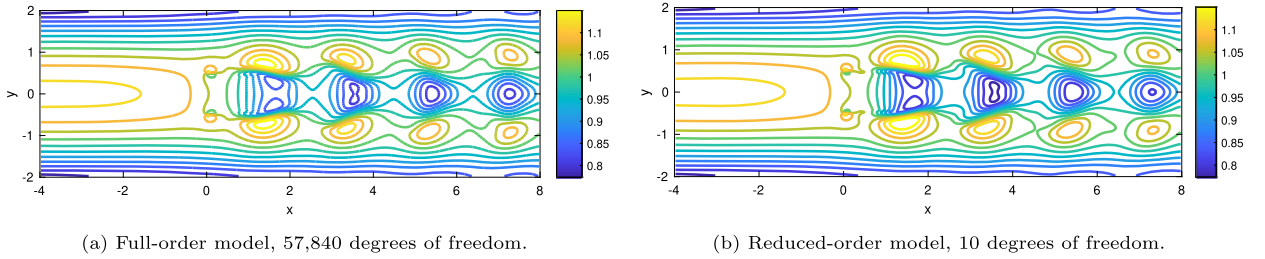

(a) Full-order model, 57,840 degrees of freedom.　　　　　(b) Reduced-order model, 10 degrees of freedom.

Fig. 9. Contours of velocity magnitude for actuator disk test case, at $t = 20$.

6.3. Actuator in non-uniform inflow

In this test case we consider an actuator disk in a non-uniform flow field. The actuator disk is added as an unsteady sink term in the momentum equations and is typically used to model the flow through wind turbines [43]. This test case features non-homogeneous boundary conditions and we therefore require the method proposed in section 4.

The test case set up is as follows. We consider a simulation domain $[-4, 8] \times [-2, 2]$ with the following inflow conditions at $x = -4$:

$$u(x = -4, y) = \frac{3}{4} - \frac{3}{32}(y - 2)(y + 2). \tag{103}$$

This is a parabolic velocity profile with a mean equal to 1. At the domain boundaries $x = 8$, $y = -2$, and $y = 2$ we employ outflow conditions at which the total stress is prescribed, according to equation (80). For example, at $x = 8$ we have, taking $p_\infty = 0$:

$$-p + \nu \frac{\partial u}{\partial x} = 0, \qquad \frac{\partial v}{\partial x} = 0. \tag{104}$$

The initial condition is the parabolic velocity profile and the Reynolds number is 500. The actuator is an infinitely thin disk located at $x = 0$, $-0.5 \leq y \leq 0.5$, with a thrust coefficient of $C_T = \frac{1}{2}$, inducing a discontinuity in the pressure (for more details, see [40]). The additional term in the FOM u-momentum equation is

$$f_h^u(t) = -C_T \Delta y(1 + \sin(\pi t)), \tag{105}$$

for the finite volumes intersected by the actuator. The first part of this expression is time-independent and can be precomputed. The second part $(1 + \sin(\pi t))$ is simple to evaluate in online mode and is multiplied with the precomputed forcing term.

We first simulate the FOM with 240×80 finite volumes from $t = 0$ to $t = 20$ with $\Delta t = 0.025$ and a classic RK4 scheme, giving $K = 801$ snapshots. The resulting velocity field at $t = 20$ is shown in Fig. 9a. Based on the FOM, the velocity field due to nonhomogeneous boundary conditions, V_{bc}, is computed from equation (78). Note that V_{bc} is a vector field defined throughout the domain, and not only on the boundary. The components of V_{bc} are shown in Fig. 10. The V_{bc} field
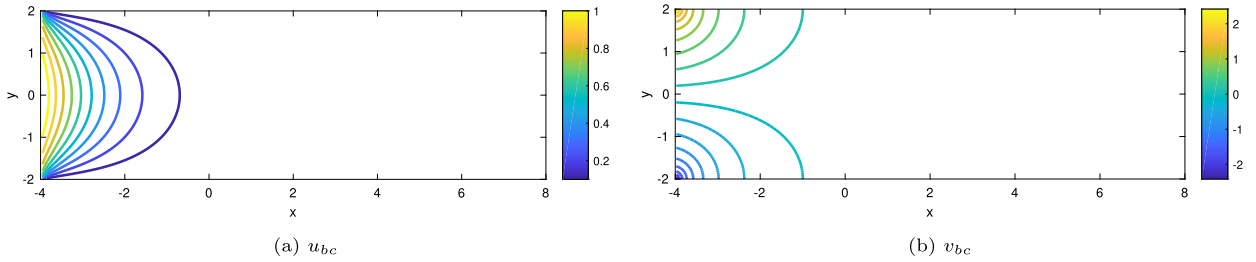
(a) u_{bc}

(b) v_{bc}

Fig. 10. Components of boundary condition function V_{bc} for actuator test case.

(a) Full-order model, 57,840 degrees of freedom.

(b) Reduced-order model, 10 degrees of freedom.
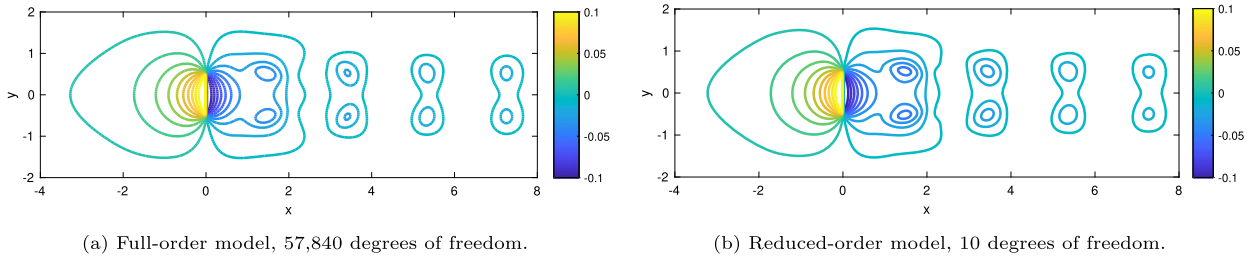
Fig. 11. Contours of pressure for actuator disk test case, at $t = 20$.
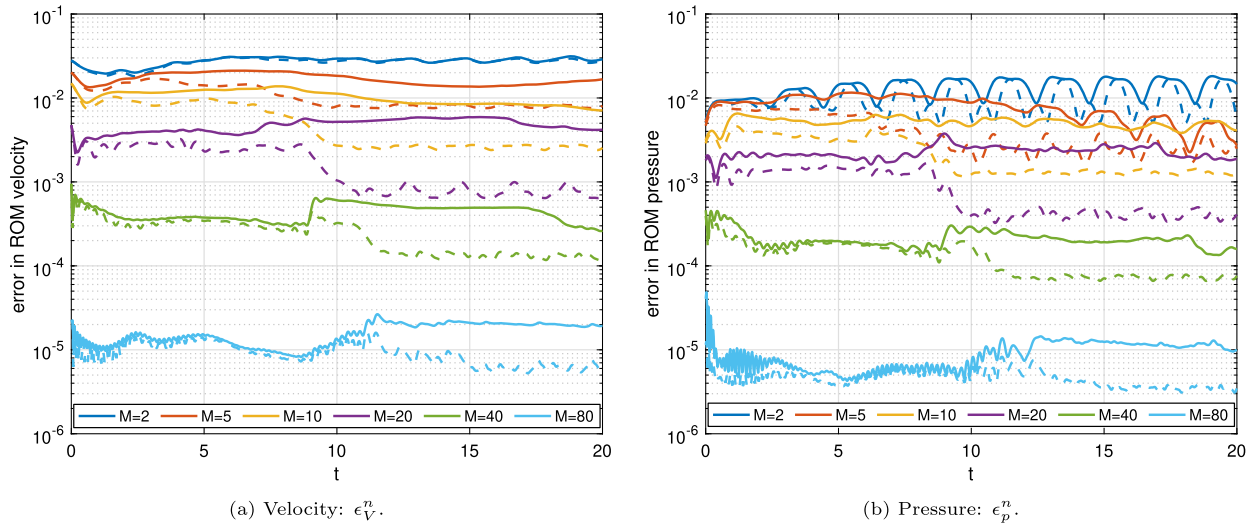

(a) Velocity: ϵ_V^n.

(b) Pressure: ϵ_p^n.

Fig. 12. Errors in ROM with respect to FOM for actuator test case. Dashed lines: basis projection error (projecting snapshots onto truncated basis).

is subtracted from the snapshot matrix. We then simulate the ROM with $M = 10$ modes, and the same time integration method and time step. Fig. 9b shows the ROM velocity field, which is almost identical to the one obtained by the FOM. Given the velocity field, the pressure can be recovered using the technique from section 5.3. This leads to an accurate pressure field, as shown in Fig. 11. Similar to the closed flow (lid-driven cavity) case, also for this open flow case we do not need to specify any additional pressure boundary conditions at the ROM level. The boundary conditions were specified on the discrete FOM level, and inherited by the ROM via the projection of the boundary vectors.

A quantitative comparison of the velocity error and the pressure error for different values of M is given in Fig. 12. The reference velocity used in (97) is the average inflow velocity, $V_{\mathrm{ref}} = 1$, whereas the reference pressure used in (99) is the pressure jump over the actuator, $p_{\mathrm{ref}} = \frac{1}{2} C_T = \frac{1}{4}$. For all values of M, stable results are obtained without requiring stabilization techniques, while clear convergence is achieved upon increasing M. For example, for $M = 10$, the scaled L_2 error in the velocity field is around 10^{-2}. The error in mass conservation of the ROM velocity field, $V_r = \Phi a + V_{bc}$, is shown in Fig. 13. As theoretically derived, our method keeps the velocity field of the ROM divergence-free, also in the case of open flows.

The computational speed-up of the ROM compared to the FOM is displayed in Fig. 14. Similar to the other test cases, the M^3 scaling of the precomputing phase is visible. For $M = 10$, the case discussed above, the speed-up factor is around 20
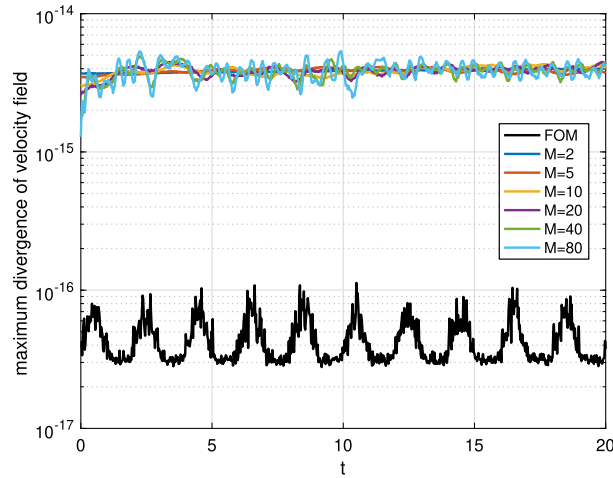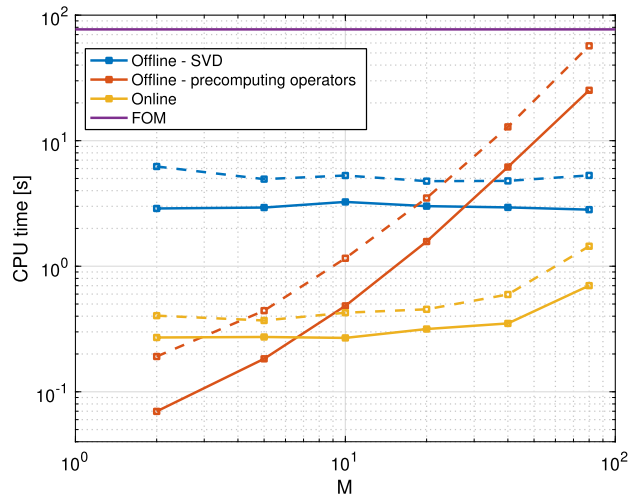
Fig. 13. Divergence error.

Fig. 14. Computational time as a function of number of modes in ROM. Dashed lines: including pressure recovery.

when including the precomputing phase in the ROM timing, and more than 100 when not considering the precomputing phase. Like for the lid-driven cavity test case, including pressure recovery roughly doubles the required computational time.

7. Conclusions

In this paper we have proposed a novel approach to arrive at an unconditionally stable reduced-order model (ROM) for the incompressible Navier-Stokes equations. The approach hinges on the following four ingredients. First, we have expressed non-linear stability through kinetic energy conservation. Second, we have used a spatially energy-conserving discretisation method as the full order model (FOM), which has a skew-symmetric convection operator, a divergence-free velocity field, and satisfies a discrete compatibility relation between the divergence and gradient operators. Third, by performing the projection of the full order model after spatial discretisation an unconditionally stable semi-discrete ROM is obtained. Last, we have used an energy-conserving time integration method. The resulting ROM is velocity-only and non-linearly stable. The stability of the method has been numerically confirmed for the roll-up of an inviscid shear-layer, for which exact energy conservation was obtained, independent of the number of modes.

In addition, we have used a constrained SVD approach in order to guarantee momentum conservation on periodic domains. Enforcing momentum conservation comes at the cost of losing a few modes (2 or 3) in the projection matrix, which can be accounted for by taking a few extra modes at a slight increase in computational effort. The constrained SVD approach can be extended to include other linear constraints apart from global momentum.

Furthermore, we have derived a boundary condition treatment for non-homogeneous boundary conditions. The adage of first discretising, then projecting simplifies the boundary condition treatment considerably, as the boundary conditions are

built into the discretisation operators and boundary vectors. The ROM follows from projecting these discrete FOM operators and boundary vectors. To satisfy the divergence-free constraint, the ROM velocity field is written in terms of a field with homogeneous boundary conditions and a non-homogeneous term, where the latter is obtained by solving once a Poisson equation at the FOM level. In future work, we plan to extend the approach to the case of unsteady and/or parametric boundary conditions.

Several test cases, both closed flow and open flow, show the stability, accuracy, and efficiency of the proposed method. Exact energy conservation is shown for an inviscid shear-layer roll-up on a periodic domain, by using implicit time integration. In practice, high-order explicit Runge-Kutta methods have been shown to form a practical alternative, being computationally efficient and suffering only from a small energy error. The computational speed-up of the ROM compared to the FOM is typically one to two orders of magnitude, depending on the test case and on the number of modes used (and whether the offline phase is included in the computation of the speed-up factor). The pressure is not part of the ROM formulation, but can be recovered once the velocity field is known. This roughly doubles the computational costs, both offline and online (if performed at every time step). With an increasing number of modes, the precomputing phase (in particular assembling the reduced convective operator) becomes the dominant factor in the ROM execution. In our test cases this is not a concern, as the number of modes is typically sufficient before the precomputing phase becomes a dominant factor. Nevertheless, one could reduce the complexity of the convective operator (a third order tensor) by using hyper-reduction techniques such as discrete empirical interpolation. In future work we will assess whether hyper-reduction can be performed in such a way that the skew-symmetry of the convective operator is retained. Further efficiency gains can be obtained by making the time step adaptive, e.g. by estimating the eigenvalues of the ROM operators.

As mentioned, this paper has focused mainly on the stability of ROMs, and less on the issue of accuracy in turbulent flows. Our view is that further studies on the accuracy of ROMs (e.g. through closure modelling techniques) will benefit significantly from the framework proposed in this paper, as it is stable and non-dissipative by design. Energy conservation is obtained in part because the skew-symmetry property of the convective term is unchanged upon projection. However, this means that the POD-Galerkin method in its basic form cannot be suitable as a reduced model for turbulent flows, as there is no additional energy dissipation coming from the projected convective terms. This is in line with the common understanding that additional measures need to be taken to make POD-Galerkin methods applicable to turbulent flows [18].

CRediT authorship contribution statement

B. Sanderse: Conceptualization, Methodology, Software, Writing - original draft, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Appendix A. Alternative forms of the convective operator

A.1. Continuous

The convective operator in divergence form can be written such that the role of the advecting velocity $\boldsymbol{c} = \boldsymbol{u}$ becomes more clear:

$$C_{\mathrm{div}}(\boldsymbol{c}, \boldsymbol{u}) = \nabla \cdot (\boldsymbol{c} \otimes \boldsymbol{u}). \tag{A.1}$$

This distinction allows us to write the divergence form in terms of the advective form C_{adv} as follows:

$$C_{\mathrm{div}}(\boldsymbol{c}, \boldsymbol{u}) = C_{\mathrm{adv}}(\boldsymbol{c}, \boldsymbol{u}) + \boldsymbol{u}\,(\nabla \cdot \boldsymbol{c})\,, \tag{A.2}$$

where

$$C_{\mathrm{adv}}(\boldsymbol{c}, \boldsymbol{u}) = (\boldsymbol{c} \cdot \nabla)\boldsymbol{u}. \tag{A.3}$$

Another commonly used form is the so-called skew-symmetric form,

$$C_{\mathrm{skew}}(\boldsymbol{c}, \boldsymbol{u}) := \frac{1}{2}C_{\mathrm{div}}(\boldsymbol{c}, \boldsymbol{u}) + \frac{1}{2}C_{\mathrm{adv}}(\boldsymbol{c}, \boldsymbol{u}) = \frac{1}{2}\nabla \cdot (\boldsymbol{c} \otimes \boldsymbol{u}) + \frac{1}{2}(\boldsymbol{c} \cdot \nabla)\boldsymbol{u}. \tag{A.4}$$

In case that the advective velocity field is divergence-free ($\nabla \cdot \boldsymbol{c} = 0$) and the velocity field is sufficiently smooth, the concepts of divergence, advective and skew-symmetric form are equivalent:

$$C(\boldsymbol{c}, \boldsymbol{u}) = C_{\mathrm{div}}(\boldsymbol{c}, \boldsymbol{u}) = C_{\mathrm{adv}}(\boldsymbol{c}, \boldsymbol{u}) = C_{\mathrm{skew}}(\boldsymbol{c}, \boldsymbol{u}). \tag{A.5}$$

The notion of skew-symmetry is related to the following property (independent of the divergence-freeness of $\boldsymbol{c}$):

$$
\begin{aligned}
C_{\mathrm{skew}}(\boldsymbol{c}, \boldsymbol{u}) \cdot \boldsymbol{v} &= \frac{1}{2}(\nabla \cdot (\boldsymbol{c} \otimes \boldsymbol{u})) \cdot \boldsymbol{v} + \frac{1}{2}((\boldsymbol{c} \cdot \nabla)\boldsymbol{u}) \cdot \boldsymbol{v}, \\
&= \frac{1}{2}\left[((\boldsymbol{c} \cdot \nabla)\boldsymbol{u}) \cdot \boldsymbol{v} + (\boldsymbol{u} \cdot \boldsymbol{v})\nabla \cdot \boldsymbol{c}\right] + \frac{1}{2}\left[\nabla \cdot ((\boldsymbol{u} \cdot \boldsymbol{v})\boldsymbol{c}) - (\boldsymbol{u} \cdot \boldsymbol{v})\nabla \cdot \boldsymbol{c} - ((\boldsymbol{c} \cdot \nabla)\boldsymbol{v}) \cdot \boldsymbol{u}\right], \\
&= \frac{1}{2}((\boldsymbol{c} \cdot \nabla)\boldsymbol{u}) \cdot \boldsymbol{v} - \frac{1}{2}((\boldsymbol{c} \cdot \nabla)\boldsymbol{v}) \cdot \boldsymbol{u} + \frac{1}{2}\nabla \cdot ((\boldsymbol{u} \cdot \boldsymbol{v})\boldsymbol{c}).
\end{aligned}
\tag{A.6}
$$

Upon integration over the entire domain, the contribution of the last term cancels in case of periodic or no-slip boundary conditions, and we obtain

$$(C_{\mathrm{skew}}(\boldsymbol{c}, \boldsymbol{u}), \boldsymbol{v}) = \frac{1}{2}((\boldsymbol{c} \cdot \nabla)\boldsymbol{u}, \boldsymbol{v}) - \frac{1}{2}((\boldsymbol{c} \cdot \nabla)\boldsymbol{v}, \boldsymbol{u}). \tag{A.7}$$

The convective operator in skew-symmetric form is skew-symmetric 'a priori' (i.e. without the assumption that $\nabla \cdot \boldsymbol{c} = 0$),

$$(C_{\mathrm{skew}}(\boldsymbol{c}, \boldsymbol{u}), \boldsymbol{v}) = -(\boldsymbol{u}, C_{\mathrm{skew}}(\boldsymbol{c}, \boldsymbol{v})). \tag{A.8}$$

The convective operators in advective or divergence form are skew-symmetric provided that $\nabla \cdot \boldsymbol{c} = 0$. In that case we have, for example,

$$(C_{\mathrm{div}}(\boldsymbol{c}, \boldsymbol{u}), \boldsymbol{v}) = -(\boldsymbol{u}, C_{\mathrm{div}}(\boldsymbol{c}, \boldsymbol{v})). \tag{A.9}$$

A.2. Discrete

In two dimensions, the integral of $C_{\mathrm{div}}^u = \frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y}$ over a finite volume surrounding $u_{i+1/2,j}$ is approximated by

$$C_h^u(V_h, u_h)_{i+1/2,j} := \bar{u}_{i+1,j}u_{i+1,j} - \bar{u}_{i,j}u_{i,j} + \bar{v}_{i+1/2,j+1/2}u_{i+1/2,j+1/2} - \bar{v}_{i+1/2,j-1/2}u_{i+1/2,j-1/2}. \tag{A.10}$$

When interpolating the velocities by mesh-independent weighting of the neighbouring velocities (e.g. $u_{i+1,j} = \frac{1}{2}(u_{i+1/2,j} + u_{i+3/2,j})$), leaving the interpolation of the fluxes $\bar{(\cdot)}$ still unspecified, this convective term can be expressed in terms of a matrix-vector product as

$$C_h^u(V_h, u_h) = \tilde{C}_h^u(V_h)u_h, \tag{A.11}$$

where (focusing on the u-velocities)

$$
\tilde{C}_h^u(V_h) = \frac{1}{2}
\begin{pmatrix}
\ddots & \ddots & & \ddots & & \\
& -\bar{u}_{i,j} & \bar{u}_{i+1,j} - \bar{u}_{i,j} & & \bar{u}_{i+1,j} & \\
& & -\bar{u}_{i+1,j} & & \bar{u}_{i+2,j} - \bar{u}_{i+1,j} & \bar{u}_{i+2,j} \\
& & & \ddots & & \ddots & \ddots
\end{pmatrix}.
\tag{A.12}
$$

This formulation is possible because the nonlinearity in the convective terms is only quadratic. We note that, apart from the diagonal elements, the matrix is skew-symmetric, independent of the interpolation method for the fluxes $\bar{u}$, $\bar{v}$. Subsequently, the fluxes are computed via mesh-independent weighting of the neighbouring velocities (e.g. $\bar{u}_{i+1,j} = \frac{1}{2}(\bar{u}_{i+1/2,j} + \bar{u}_{i+3/2,j})$). The convective terms can then be rewritten as

$$
\begin{aligned}
&\frac{1}{2}u_{i+1/2,j}\left[\frac{1}{2}(\bar{u}_{i+1/2,j} + \bar{u}_{i+3/2,j}) - \frac{1}{2}(\bar{u}_{i-1/2,j} + \bar{u}_{i+1/2,j}) + \frac{1}{2}(\bar{v}_{i,j+1/2} + \bar{v}_{i+1,j+1/2}) - \frac{1}{2}(\bar{v}_{i,j-1/2} + \bar{v}_{i+1,j-1/2})\right] \\
&\qquad + \frac{1}{2}u_{i+3/2,j}\frac{1}{2}\left(\bar{u}_{i+1/2,j} + \bar{u}_{i+3/2,j}\right) - \frac{1}{2}u_{i-1/2,j}\frac{1}{2}\left(\bar{u}_{i-1/2,j} + \bar{u}_{i+1/2,j}\right) \\
&\qquad\qquad + \frac{1}{2}u_{i+1/2,j+1}\frac{1}{2}\left(\bar{v}_{i,j+1/2} + \bar{v}_{i+1,j+1/2}\right) - \frac{1}{2}u_{i+1/2,j-1}\frac{1}{2}\left(\bar{v}_{i,j-1/2} + \bar{v}_{i+1,j-1/2}\right), \quad \text{(A.13)}
\end{aligned}
$$

where the term between brackets [.] is zero as long as the continuity equation is satisfied.

Appendix B. Assembling the ROM operators with boundary conditions

B.1. Summary of boundary conditions of FOM

The semi-discrete equations with non-homogeneous boundary conditions read (equations (73)-(74)):

$$M_h V_h(t) = y_M, \tag{B.1}$$

$$\Omega_h \frac{dV_h(t)}{dt} = F_h^{CD}(V_h(t)) - (G_h p_h(t) + y_G), \tag{B.2}$$

with

$$F_h^{CD}(V_h(t)) = -C_h(V_h(t), V_h(t)) + \nu(D_h V_h(t) + y_D) + f_h. \tag{B.3}$$

The vector V_h does not feature unknowns on the boundaries, but only unknowns associated to interior finite volumes (except in the case of outflow conditions). An example of the positioning of unknowns near a horizontal inflow boundary is given in Fig. B.15.

The divergence-free condition evaluated in a volume adjacent to a boundary then reads:

$$u_{3/2,j}\Delta y - u_b \Delta y + v_{1,j+1/2}\Delta x - v_{1,j-1/2}\Delta x = 0, \tag{B.4}$$

where in terms of equation (B.1), y_M contains $u_b \Delta y$, and the matrix M_h the remaining terms. The momentum equation does not need to be solved for u_b, as it is known, and is thus solved for the interior points only. For all interior points, the gradient of the pressure is readily available (in this case $(p_{2,j} - p_{1,j})/\Delta x$. Consequently, no pressure boundary conditions need to be specified. It follows directly that for the Poisson equation also no pressure boundary conditions are required: differentiate (B.4) in time, and substitute the momentum equations for $\frac{du_{3/2,j}}{dt}$, $\frac{dv_{1,j+1/2}}{dt}$ and $\frac{dv_{1,j-1/2}}{dt}$. This gives the discrete Poisson equation without requiring the specification of pressure boundary conditions.

The diffusive discretization including boundary conditions reads

$$D_h V_h + y_D. \tag{B.5}$$

On a uniform cartesian grid, $D_h \in \mathbb{R}^{N_V \times N_V}$ contains the standard $\begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$ stencil (for each velocity component and each coordinate direction), and $y_D \in \mathbb{R}^{N_V}$ is a vector containing mostly zeros, except for the finite volumes adjacent to a boundary, which contain the value given by the boundary condition.

The convective discretization including boundary conditions is given by

$$C_h(V_h, V_h) = K_h((I_h V_h + y_I) \circ (A_h V_h + y_A)), \tag{B.6}$$

where $K_h \in \mathbb{R}^{N_V \times N_F}$ is a differencing matrix from faces to volume midpoints (with -1 and 1 as entries), $I_h \in \mathbb{R}^{N_F \times N_V}$ interpolates the (convecting) fluxes to the finite volume faces, $A_h \in \mathbb{R}^{N_F \times N_V}$ in a similar fashion averages the (convected) velocities to the faces, and $\circ$ is the elementwise (Hadamard) product. $y_I \in \mathbb{R}^{N_F}$ and $y_A \in \mathbb{R}^{N_F}$ have nonzero entries for the finite volumes adjacent to boundaries. The simplified form $\tilde{C}(V_h)V_h$ from Appendix A.2, in which no boundary conditions were present, is obtained as a special case when $y_I = y_A = 0$:

$$\tilde{C}_h(V_h) = K_h \operatorname{diag}(I_h V_h) A_h, \tag{B.7}$$

where diag is the operator that transforms a vector to a diagonal matrix with the entries of the vector on its diagonal.

B.2. Precomputing the ROM including boundary conditions

In this section we explain the projection of the FOM including boundary conditions into a formulation amenable for offline decomposition. We also mention how the ROM operators can be obtained in a rather non-intrusive way from an existing FOM code. There are two sources of boundary conditions when projecting the FOM: (i) the boundary conditions already present in the FOM, explained in the previous section in terms of the vectors y_D, y_I, y_A, and y_G, acting *only* on finite volumes adjacent to a boundary, and (ii) the boundary field V_{bc} constructed in section 4.

With both sources present, the ROM formulation of the diffusive term is

$$\Phi^T(D_h(\Phi a + V_{bc}) + y_D) = (\Phi^T D_h \Phi)a + \Phi^T(D_h V_{bc} + y_D) =: D_r a + y_{D_r}. \tag{B.8}$$

In our FOM code implementation, the diffusive terms (equation (B.5)) are obtained by calling a subroutine `Diffu-sion(Vh)`. Given such a subroutine, the ROM terms are simple to obtain, as is best explained using a piece of pseudocode:
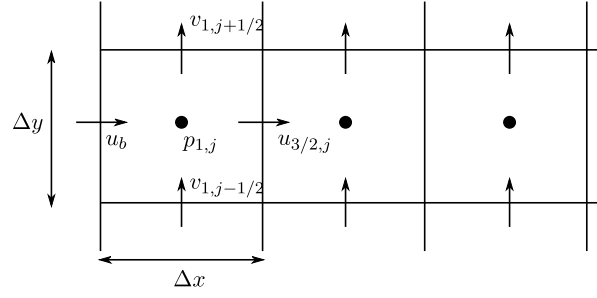
Fig. B.15. Staggered grid and unknowns near inflow boundary. u_b is given and not part of the vector of unknowns.

```
yDr = Phi'*Diffusion(Vbc)
for i=1:M
   Dr(:,i) = Phi'*(Diffusion(Phi(:,i)) - Diffusion(0))
end
```

A similar approach is used to obtain the ROM formulation of the convective terms. These terms read

$$\Phi^T K_h((I_h(\Phi a + V_{bc}) + y_I) \circ (A_h(\Phi a + V_{bc}) + y_A)), \tag{B.9}$$

which can be written as

$$C_{r,2}(a \otimes a) + C_{r,1}a + y_{C_r}. \tag{B.10}$$

The fact that the ROM convective terms feature not only a purely quadratic term, but also the linear term $C_{r,1}a$, is due to the boundary conditions. This was also observed in e.g. [4,7]. Similar to the diffusive terms, our FOM code implementation of the convective terms (equation (B.7)) is given by Convection(Vh,Vh), where the first argument corresponds to the convecting velocity and the second to the convected velocity, as explained in A.1. The terms in (B.10) are obtained from:

```
yCr = Phi'*Convection(Vbc,Vbc)
for i=1:M
   Cr1(:,i) = Phi'*(Convection(Vbc,Phi(:,i)) + Convection(Phi(:,i),Vbc)
                    - Convection(Vbc,0) - Convection(0,Vbc))
   for j=1:M
      k = (i-1)*M + j
      Cr2(:,k) = Phi'*(Convection(Phi(:,j),Phi(:,i)) - Convection(Phi(:,j),0)
                       - Convection(0,Phi(:,i)) + Convection(0,0))
   end
end
```

The contribution of pressure boundary conditions (only needed in the case of outflow boundaries) in the ROM simply reads $y_{G_r} = \Phi^T y_G$, and the contribution of body forces similarly reads $f_r = \Phi^T f_h$. The entire ROM formulation then reads

$$\frac{da}{dt} = F_r(a) = F_{r,2}(a \otimes a) + F_{r,1}a + F_{r,0}, \tag{B.11}$$

where

$$F_{r,2} = -C_{r,2}, \tag{B.12}$$

$$F_{r,1} = -C_{r,1} + \nu D_r, \tag{B.13}$$

$$F_{r,0} = -y_{C_r} + \nu y_{D_r} - y_{G_r} + f_r. \tag{B.14}$$

Appendix C. Proof of momentum conservation in weighted norm

In this section we prove that the constrained SVD construction in section 3.4 is conserving momentum for the weighted norm (30) and keeps Φ divergence-free. The steps in the construction can be summarized as follows:

Form adapted snapshot matrix:	$\tilde{X} = X - EE^T\Omega_h X,$	(C.1)
Transform to include weighted norm:	$\hat{X} = \Omega_h^{1/2}\tilde{X},$	(C.2)

Perform SVD of $\hat{X}$: $$\hat{X} = \hat{\Phi}\Sigma\Psi^*, \tag{C.3}$$

Transform back to include weighted norm: $$\tilde{\Phi} = \Omega_h^{-1/2}\hat{\Phi}, \tag{C.4}$$

Add E and truncate: $$\Phi = [E \ \tilde{\Phi}]_M. \tag{C.5}$$

Note that the matrix E is scaled such that

$$E^T\Omega_h E = I. \tag{C.6}$$

The proof that Φ satisfies $\Phi\Phi^T\Omega_h E = E$ is a substitution exercise:

$$
\begin{aligned}
\Phi\Phi^T\Omega_h E &= [E \ \tilde{\Phi}]_M[E \ \tilde{\Phi}]_M^T\Omega_h E \\
&= EE^T\Omega_h E + \tilde{\Phi}_{M-1}\tilde{\Phi}_{M-1}^T\Omega_h E \\
&= E + \Omega_h^{-1/2}\hat{\Phi}\hat{\Phi}^T\Omega_h^{1/2}E \qquad \text{(omitting the truncation subscript).}
\end{aligned}
\tag{C.7}
$$

We proceed to show that the second term equals zero. For this we use equation (39): the expression for the modes $\hat{\Phi}$ in terms of the snapshots $\hat{X}$:

$$\lambda_j\hat{\Phi}_j = \hat{X}\hat{X}^T\hat{\Phi}_j \quad \rightarrow \quad \lambda_j\hat{\Phi}_j^T\Omega_h^{1/2}E = \hat{\Phi}_j^T\hat{X}\hat{X}^T\Omega_h^{1/2}E. \tag{C.8}$$

Substituting the expression for the adapted snapshot matrix yields:

$$
\begin{aligned}
\hat{\Phi}_j^T\Omega_h^{1/2}E &= \frac{1}{\lambda_j}\hat{\Phi}_j\hat{X}\hat{X}^T\Omega_h^{1/2}E, \\
&= \frac{1}{\lambda_j}\hat{\Phi}_j\Omega_h^{1/2}\tilde{X}\tilde{X}^T\Omega_h E, \\
&= \frac{1}{\lambda_j}\hat{\Phi}_j\Omega_h^{1/2}(X - EE^T\Omega_h X)(X - EE^T\Omega_h X)^T\Omega_h E.
\end{aligned}
\tag{C.9}
$$

The terms including the snapshot matrix X can be written as

$$
\begin{aligned}
(X - EE^T\Omega_h X)(X - EE^T\Omega_h X)^T\Omega_h E =& XX^T\Omega_h E - EE^T\Omega_h XX^T\Omega_h E \\
& - XX^T\Omega_h EE^T\Omega_h E + EE^T\Omega_h XX^T\Omega_h EE^T\Omega_h E, \\
=& XX^T\Omega_h E - EE^T\Omega_h XX^T\Omega_h E \\
& - XX^T\Omega_h E + EE^T\Omega_h XX^T\Omega_h E, \\
=& 0.
\end{aligned}
\tag{C.10}
$$

Consequently, as long as $\lambda_j \neq 0$, $\hat{\Phi}_j^T\Omega_h^{1/2}E = 0$, and thus $\Phi\Phi^T\Omega_h E = E$.

It remains to prove that the momentum-conserving construction keeps the basis Φ divergence-free, in other words, whether

$$M_h\Phi = M_h[E \ \tilde{\Phi}]_M \tag{C.11}$$

equals zero. Since $M_h E = 0$, we need to only consider $M_h\tilde{\Phi}_j$ for each column j of $\tilde{\Phi}$.

$$
\begin{aligned}
M_h\tilde{\Phi}_j &= M_h\Omega_h^{-1/2}\hat{\Phi}, \\
&= \frac{1}{\lambda_j}M_h\Omega_h^{-1/2}\hat{X}\hat{X}^T\hat{\Phi}_j, \\
&= \frac{1}{\lambda_j}M_h\tilde{X}\tilde{X}^T\Omega_h^{1/2}\hat{\Phi}_j, \\
&= \frac{1}{\lambda_j}M_h(X - EE^T\Omega_h X)(X - EE^T\Omega_h X)^T\Omega_h^{1/2}\hat{\Phi}_j, \\
&= 0,
\end{aligned}
\tag{C.12}
$$

where the last equality follows from $M_h E = 0$ and $M_h X = 0$.

References

[1] B.M. Afkham, A. Bhatt, B. Haasdonk, J.S. Hesthaven, Symplectic model-reduction with a weighted inner product, ArXiv e-prints: arXiv:1803.07799, 2018.

[2] B.M. Afkham, J.S. Hesthaven, Structure preserving model reduction of parametric Hamiltonian systems, SIAM J. Sci. Comput. 39 (6) (2017) A2616–A2644.

[3] B.M. Afkham, J.S. Hesthaven, Structure-preserving model-reduction of dissipative Hamiltonian systems, J. Sci. Comput. 81 (1) (2019) 3–21.

[4] I. Akhtar, A.H. Nayfeh, C.J. Ribbens, On the stability and extension of reduced-order Galerkin models in incompressible flows: a numerical study of vortex shedding, Theor. Comput. Fluid Dyn. 23 (3) (2009) 213–237.

[5] A.C. Antoulas, Approximation of Large-Scale Dynamical Systems, Society for Industrial and Applied Mathematics, 2005.

[6] N. Aubry, W.-Y. Lian, E.S. Titi, Preserving symmetries in the proper orthogonal decomposition, SIAM J. Sci. Comput. 14 (2) (1993) 483–505.

[7] M.J. Balajewicz, E.H. Dowell, B.R. Noack, Low-dimensional modelling of high-Reynolds-number shear flows incorporating constraints from the Navier–Stokes equation, J. Fluid Mech. 729 (2013) 285–308.

[8] F. Ballarin, A. Manzoni, A. Quarteroni, G. Rozza, Supremizer stabilization of POD-Galerkin approximation of parametrized steady incompressible Navier-Stokes equations, Int. J. Numer. Methods Eng. 102 (5) (2015) 1136–1161.

[9] P. Benner, S. Gugercin, K. Willcox, A survey of projection-based model reduction methods for parametric dynamical systems, SIAM Rev. 57 (4) (2015) 483–531.

[10] L. Berselli, T. Iliescu, W.J. Layton, Mathematics of Large Eddy Simulation of Turbulent Flows, Springer-Verlag, 2006.

[11] A. Caiazzo, T. Iliescu, V. John, S. Schyschlowa, A numerical investigation of velocity-pressure reduced order models for incompressible flows, J. Comput. Phys. 259 (2014) 598–616.

[12] K. Carlberg, M. Barone, H. Antil, Galerkin v. least-squares Petrov–Galerkin projection in nonlinear model reduction, J. Comput. Phys. 330 (2017) 693–734.

[13] K. Carlberg, C. Bou-Mosleh, C. Farhat, Efficient non-linear model reduction via a least-squares Petrov-Galerkin projection and compressive tensor approximations, Int. J. Numer. Methods Eng. 86 (2) (2011) 155–181.

[14] K. Carlberg, Y. Choi, S. Sargsyan, Conservative model reduction for finite-volume models, J. Comput. Phys. 371 (2018) 280–314.

[15] W. Cazemier, R. Verstappen, A. Veldman, Proper orthogonal decomposition and low-dimensional models for driven cavity flows, Phys. Fluids 10 (7) (1998) 1685–1699.

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

[17] M. Couplet, C. Basdevant, P. Sagaut, Calibrated reduced-order POD-Galerkin system for fluid flow modelling, J. Comput. Phys. 207 (1) (2005) 192–220.

[18] L. Fick, Y. Maday, A.T. Patera, T. Taddei, A stabilized POD model for turbulent flows over a range of Reynolds numbers: optimal parameter sampling and constrained projection, J. Comput. Phys. 371 (2018) 214–243.

[19] E. Fonn, H. van Brummelen, T. Kvamsdal, A. Rasheed, Fast divergence-conforming reduced basis methods for steady Navier–Stokes flow, Comput. Methods Appl. Mech. Eng. 346 (2019) 486–512.

[20] T. Gallouët, R. Herbin, J.C. Latché, K. Mallem, Convergence of the marker-and-cell scheme for the incompressible Navier–Stokes equations on non-uniform grids, Found. Comput. Math. 18 (1) (2018) 249–289.

[21] P.M. Gresho, R.L. Sani, Incompressible Flow & the Finite Element Method - Advection-Diffusion & Isothermal Laminar Flow, John Wiley & Sons, Ltd, 1998.

[22] M.D. Gunzburger, J.S. Peterson, J.N. Shadid, Reduced-order modeling of time-dependent PDEs with multiple parameters in the boundary data, Comput. Methods Appl. Mech. Eng. 196 (4–6) (2007) 1030–1047.

[23] F.H. Harlow, J.E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Phys. Fluids 8 (12) (1965) 2182–2189.

[24] S. Hijazi, G. Stabile, A. Mola, G. Rozza, Data-driven POD-Galerkin reduced order model for turbulent flows, J. Comput. Phys. 416 (2019) 109513.

[25] P. Holmes, J.L. Lumley, G. Berkooz, C.W. Rowley, Galerkin projection, in: Turbulence, Coherent Structures, Dynamical Systems and Symmetry, 2012, pp. 106–129.

[26] I. Kalashnikova, M.F. Barone, S. Arunajatesan, B.G. van Bloemen Waanders, Construction of energy-stable projection-based reduced order models, Appl. Math. Comput. 249 (2014) 569–596.

[27] B. Karasözen, M. Uzunca, Energy preserving model order reduction of the nonlinear Schrödinger equation, Adv. Comput. Math. 44 (6) (2018) 1769–1796.

[28] K. Kean, M. Schneier, Error analysis of supremizer pressure recovery for POD based reduced order models of the time-dependent Navier-Stokes equations, ArXiv e-print: arXiv:1909.06022, sep 2019.

[29] B. Kramer, K.E. Willcox, Nonlinear model order reduction via lifting transformations and proper orthogonal decomposition, AIAA J. 57 (6) (2019) 2297–2307.

[30] 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 International Publishing, 2014, pp. 235–273.

[31] M. Mohebujjaman, L.G. Rebholz, T. Iliescu, Physically constrained data-driven correction for reduced-order modeling of fluid flows, Int. J. Numer. Methods Fluids 89 (3) (2019) 103–122.

[32] M. Mohebujjaman, L.G. Rebholz, X. Xie, T. Iliescu, Energy balance and mass conservation in reduced order models of fluid flows, J. Comput. Phys. 346 (2017) 262–277.

[33] B.R. Noack, From snapshots to modal expansions-bridging low residuals and pure frequencies, J. Fluid Mech. 802 (2016) 1–4.

[34] B.R. Noack, P. Papas, P.A. Monkewitz, The need for a pressure-term representation in empirical Galerkin models of incompressible shear flows, J. Fluid Mech. 523 (2005) 339–365.

[35] L. Peng, K. Mohseni, Symplectic model reduction of Hamiltonian systems, SIAM J. Sci. Comput. 38 (1) (2016) A1–A27.

[36] A. Quarteroni, A. Manzoni, F. Negri, Reduced Basis Methods for Partial Differential Equations, Springer International Publishing, 2016.

[37] D. Rempfer, On low-dimensional Galerkin models for fluid flow, Theor. Comput. Fluid Dyn. 14 (2) (2000) 75–88.

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

[39] S. Rubino, Numerical analysis of a projection-based stabilized POD-ROM for incompressible flows, ArXiv e-print: arXiv:1907.09213, 2019.

[40] B. Sanderse, ECNS: Energy-Conserving Navier-Stokes Solver - Verification of steady laminar flows, ECN-E11042. Technical report, Energy Research Centre of the Netherlands, 2011.

[41] B. Sanderse, Energy-conserving Runge-Kutta methods for the incompressible Navier-Stokes equations, J. Comput. Phys. 233 (1) (2013) 100–131.

[42] B. Sanderse, B. Koren, Accuracy analysis of explicit Runge-Kutta methods applied to the incompressible Navier-Stokes equations, J. Comput. Phys. 231 (8) (2012) 3041–3063.

[43] B. Sanderse, S. Pijl, B. Koren, Review of computational fluid dynamics for wind turbine wake aerodynamics, Wind Energy 14 (7) (2011) 799–819.

[44] B. Sanderse, R. Verstappen, B. Koren, Boundary treatment for fourth-order staggered mesh discretizations of the incompressible Navier-Stokes equations, J. Comput. Phys. 257 (2014) 1472–1505.

[45] D. Shin, J.C. Strikwerda, Inf-sup conditions for finite-difference approximations of the Stokes equations, J. Aust. Math. Soc. Ser. B, Appl. Math 39 (1) (1997) 121–134.
[46] L. Sirovich, Turbulence and the dynamics of coherent structures. I. Coherent structures, Q. Appl. Math. 45 (3) (1987) 561–571.
[47] R. Smith, Uncertainty Quantification, Theory, Implementation, and Applications, SIAM, 2014.
[48] G. Stabile, G. Rozza, Finite volume POD-Galerkin stabilised reduced order methods for the parametrised incompressible Navier-Stokes equations, Comput. Fluids 173 (2018) 273–284.
[49] K. Taira, S.L. Brunton, S.T. Dawson, C.W. Rowley, T. Colonius, B.J. McKeon, O.T. Schmidt, S. Gordeyev, V. Theofilis, L.S. Ukeiley, Modal analysis of fluid flows: an overview, AIAA J. 55 (12) (2017) 4013–4041.
[50] F. Trias, O. Lehmkuhl, A. Oliva, C. Pérez-Segarra, R. Verstappen, Symmetry-preserving discretization of Navier–Stokes equations on collocated unstructured grids, J. Comput. Phys. 258 (2014) 246–267.
[51] F.X. Trias, O. Lehmkuhl, A self-adaptive strategy for the time integration of Navier-Stokes equations, Numer. Heat Transf., Part B, Fundam. 60 (2011) 116–134.
[52] A.E.P. Veldman, "Missing" boundary conditions? Discretize first, substitute next, and combine later, SIAM J. Sci. Stat. Comput. 11 (1) (1990) 82–91.
[53] R. Verstappen, A. Veldman, Symmetry-preserving discretization of turbulent flow, J. Comput. Phys. 187 (1) (2003) 343–368.
[54] Z. Wang, I. Akhtar, J. Borggaard, T. Iliescu, Proper orthogonal decomposition closure models for turbulent flows: a numerical comparison, Comput. Methods Appl. Mech. Eng. 237–240 (2012) 10–26.
[55] J. Weller, E. Lombardi, M. Bergmann, A. Iollo, Numerical methods for low-order modeling of fluid flows based on POD, Int. J. Numer. Methods Fluids 63 (2009) 249–268.
[56] M. Xiao, P. Breitkopf, R.F. Coelho, P. Villon, W. Zhang, Proper orthogonal decomposition with high number of linear constraints for aerodynamical shape optimization, Appl. Math. Comput. 247 (2014) 1096–1112.
[57] M. Xiao, P. Breitkopf, R. Filomeno Coelho, C. Knopf-Lenoir, P. Villon, W. Zhang, Constrained proper orthogonal decomposition based on QR-factorization for aerodynamical shape optimization, Appl. Math. Comput. 223 (2013) 254–263.