

A POD-Galerkin reduced order model for the Navier–Stokes equations in stream function-vorticity formulation

Michele Girfoglio ^a, Annalisa Quaini ^b, Gianluigi Rozza ^{a,*}

^a SISSA, International School for Advanced Studies, Mathematics Area, mathLab, via Bonomea, Trieste 265 34136, Italy

^b Department of Mathematics, University of Houston, Houston TX 77204, USA



ARTICLE INFO

Keywords:

Navier–stokes equations
Stream function-vorticity formulation
Proper orthogonal decomposition
Reduced order model
Galerkin projection

ABSTRACT

We develop a Proper Orthogonal Decomposition (POD)-Galerkin based Reduced Order Model (ROM) for the efficient numerical simulation of the parametric Navier–Stokes equations in the stream function-vorticity formulation. Unlike previous works, we choose different reduced coefficients for the vorticity and stream function fields. In addition, for parametric studies we use a global POD basis space obtained from a database of time dependent full order snapshots related to sample points in the parameter space. We test the performance of our ROM strategy with the well-known vortex merger benchmark and a more complex case study featuring the geometry of the North Atlantic Ocean. Accuracy and efficiency are assessed for both time reconstruction and physical parameterization.

1. Introduction

The formulation of the 2D Navier–Stokes equations in terms of stream function and vorticity represents an attractive alternative to the model in primitive variables for two main reasons: (i) there are only two scalar unknowns and (ii) the divergence free constraint for the velocity is automatically satisfied by the definition of the stream function. Computational studies on this formulation can be found in, e.g., [1–5].

While there exists an abundance of literature on Reduced Order Models (ROMs) for the Navier–Stokes equations formulated in primitive variables starting from different Full Order Methods (FOMs), e.g. Finite Element methods [6–8] or Finite Volume methods [9–11], it is only relatively recently that ROMs have been applied to the stream function-vorticity formulation [12–18].

In this paper, we develop a POD-Galerkin ROM for the stream function-vorticity formulation. The main building blocks of our approach are:

- the collection of a database of simulations using a computationally efficient finite volume method;
- the extraction of the most energetic modes representing the system dynamics through Proper Orthogonal Decomposition (POD);
- a Galerkin projection on the space spanned by these most energetic modes for the computation of stream function and vorticity reduced coefficients.

Two are the main novelties of our approach. First, unlike previous works [12,13,15–18] we consider different coefficients for the approximation of the vorticity and stream function fields. This choice leads to two important consequences: (i) the stream function basis functions do not depend on the particular vorticity basis functions, but are instead computed directly from the stream function high-fidelity solutions during the offline phase; (ii) the reduced spaces for the stream function and vorticity can have different dimensions. Both points increase the accuracy of the ROM approximation. The second novelty pertains the parametric study with respect to two crucial model parameters (Reynolds number and strength of the forcing term) for which we use a global POD basis space computed by time dependent FOM snapshots associated to sample points in the parameter space. This is a difference with respect to [12,13,15–17], where a POD basis is computed for each parameter in the training set and the basis functions for new parameter values are found via interpolation of the basis functions associated to the training set.

The work in this paper represents an intermediate step towards the development of new FOM and ROM approaches for the quasi-geostrophic equations (see [19] for a recent review) that are usually written in terms of stream function and (potential) vorticity. In particular, in [20] we developed a Large Eddy Simulation model for such equations at the full order level and associated ROMs are currently under investigation. We believe that once our ROM framework is fully developed it could be used for ocean and weather forecast.

* Corresponding author.

E-mail addresses: mgirfogl@sissa.it (M. Girfoglio), quaini@math.uh.edu (A. Quaini), grozza@sissa.it (G. Rozza).

All the FOM simulations presented in this work have been performed with OpenFOAM® [21], an open source Finite Volume C++ library widely used by commercial and academic organizations. For the Navier–Stokes equations in primitive variables, OpenFOAM features several partitioned algorithms (PISO [22], PIMPLE [23] and SIMPLE [24]) based on the Chorin–Temam projection scheme [25]. To the best of our knowledge, no solver for the stream function-vorticity formulation has been shared with the large OpenFOAM® community. Thus, we have implemented such solver at the FOM level. The ROM computations have been carried out with ITHACA-FV [10], an in-house implementation of several ROM techniques within OpenFOAM®.

The rest of this paper is organized as follows. In Section 2, we describe the full order model and the numerical method we use for its time and space discretization. Section 3 presents the reduced order model. The numerical experiments are reported in Section 4. Finally, conclusions and future perspectives are provided in Section 5.

2. The full order model

2.1. The Navier–Stokes equations in stream function-vorticity formulation

We consider the motion of a two-dimensional incompressible, viscous fluid in a fixed domain $\Omega \subset \mathbb{R}^2$ over a time interval of interest (t_0, T) . The flow is described by the incompressible Navier–Stokes equations:

$$\partial_t \mathbf{u} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) - \frac{1}{Re} \Delta \mathbf{u} + \nabla p = \mathbf{f} \quad \text{in } \Omega \times (t_0, T), \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega \times (t_0, T), \quad (2)$$

where Eq. (1) states the conservation of linear momentum and Eq. (2) represents the conservation of mass. Here, $\mathbf{u}(x, y, t) = (u(x, y, t), v(x, y, t), 0)$ is the fluid velocity, ∂_t denotes the time derivative, $p(x, y, t)$ is the pressure and Re is the Reynolds number. In (1), we take into account possible body forces $\mathbf{f}(x, y, t)$. We focus on forcing terms that can be expressed as product of two functions: one function that depends only on space and the other that depends only on time, i.e. $\mathbf{f}(x, y, t) = f_2(t) \mathbf{f}_1(x, y)$. See Remark 3.1 for more details about this choice.

Let ∂_x and ∂_y denote the derivative with respect to the x and y spatial coordinate, respectively. By applying the curl operator $\nabla \times$ to Eq. (1), we obtain the governing equation for the vorticity field $\omega(x, y, t) = \nabla \times \mathbf{u} = (0, 0, \omega) = (0, 0, \partial_x v - \partial_y u)$.

$$\partial_t \omega + \nabla \cdot (\mathbf{u} \omega) - \frac{1}{Re} \Delta \omega = F \quad \text{in } \Omega \times (t_0, T),$$

where $F = (0, 0, F) = \nabla \times \mathbf{f}$. The incompressibility constraint (2) leads to the introduction of the stream function $\psi(x, y, t) = (0, 0, \psi)$ such that $\mathbf{u} = \nabla \times \psi$, or, equivalently, $(\partial_y \psi, -\partial_x \psi) = (u, v)$. The stream function ψ and vorticity ω are linked by a Poisson equation

$$-\Delta \psi = \omega \quad \text{in } \Omega \times (t_0, T).$$

To close problem (1)–(2), we need to provide initial data $\mathbf{u}(x, y, t_0) = \mathbf{u}_0$ and enforce proper boundary conditions. In this work, we consider either homogeneous Dirichlet conditions on the entire boundary, i.e.

$$\mathbf{u} = \mathbf{0} \quad \text{on } \partial\Omega \times (t_0, T), \quad (3)$$

or the following slip condition:

$$\mathbf{u} \cdot \mathbf{n} = 0 \quad \text{and} \quad \partial_n(\mathbf{u} \cdot \mathbf{t}) = 0 \quad \text{on } \partial\Omega \times (t_0, T), \quad (4)$$

where \mathbf{n} is the outward unit normal and \mathbf{t} the unit tangent vector to $\partial\Omega$.

Summarizing, the Navier–Stokes equations in stream function-vorticity formulation, which represent our full order model, are given by

$$\partial_t \omega + \nabla \cdot ((\nabla \times \psi) \omega) - \frac{1}{Re} \Delta \omega = F \quad \text{in } \Omega \times (t_0, T), \quad (5)$$

$$-\Delta \psi = \omega \quad \text{in } \Omega \times (t_0, T), \quad (6)$$

endowed with boundary conditions

$$\psi = 0 \quad \text{on } \partial\Omega \times (t_0, T), \quad (7)$$

$$\partial_n \omega = 0 \quad \text{on } \partial\Omega \times (t_0, T), \quad (8)$$

or

$$\psi = 0 \quad \text{on } \partial\Omega \times (t_0, T), \quad (9)$$

$$\omega = 0 \quad \text{on } \partial\Omega \times (t_0, T), \quad (10)$$

and initial data $\omega(x, y, t_0) = \omega_0$. Notice that while (7)–(8) correspond to (4), homogeneous Dirichlet conditions (9)–(10) do not correspond to (3).

Remark 2.1. As mentioned in Section 1, the stream function-vorticity formulation of the 2D Navier–Stokes equations has important advantages. However, there are some notable difficulties associated with the convection term in the vorticity transport equation, lack of boundary condition for the vorticity at no-slip boundaries, and determination of the value of the stream function at the internal boundaries for multiply connected domains. While these issues have been successfully addressed at the full order level (see, e.g., [26]), they remain open at the reduced order level. Addressing these non-trivial difficulties is beyond the scope of this work, which focuses on an improved POD–Galerkin ROM for the stream function-vorticity formulation in mono-connected domains with boundary conditions (7)–(8) or (9)–(10).

Remark 2.2. If one is interested in the pressure, a possible way to obtain it is with a pressure Poisson equation. At the FOM level, such equation could be solved at each time step once the stream function is computed. At the ROM level, one could collect the pressure snapshots and perform the POD to obtain the pressure modes, onto which the Poisson equation would be projected. We omit the details here and refer the interested reader to, e.g., [27].

2.2. Time and space discretization

Let us start with the time discretization of the FOM (5)–(6). Let $\Delta t \in \mathbb{R}$, $t^n = t_0 + n\Delta t$, with $n = 0, \dots, N_T$ and $T = t_0 + N_T \Delta t$. We denote by y^n the approximation of a generic quantity y at the time t^n . Problem (5)–(6) discretized in time by a Backward Differentiation Formula of order 1 (BDF1) reads: given $\omega^0 = \omega_0$, for $n \geq 0$ find the solution $(\psi^{n+1}, \omega^{n+1})$ of system:

$$\frac{1}{\Delta t} \omega^{n+1} + \nabla \cdot ((\nabla \times \psi^{n+1}) \omega^{n+1}) - \frac{1}{Re} \Delta \omega^{n+1} = b^{n+1}, \quad (11)$$

$$\nabla \psi^{n+1} = \omega^{n+1}, \quad (12)$$

where $b^{n+1} = F^{n+1} + \omega^n / \Delta t$. In order to contain the computational cost required to approximate the solution to problem (11)–(12), we opt for a segregated algorithm. Given the vorticity ω^n , at t^{n+1} such algorithm requires to:

- (i) Find the vorticity ω^{n+1} such that

$$\frac{1}{\Delta t} \omega^{n+1} + \nabla \cdot ((\nabla \times \psi^*) \omega^{n+1}) - \frac{1}{Re} \Delta \omega^{n+1} = b^{n+1}, \quad (13)$$

where ψ^{n+1} in (11) is replaced by an extrapolation ψ^* . Since we are using BDF1, we set $\psi^* = \psi^n$.

- (ii) Find ψ^{n+1} such that

$$\nabla \psi^{n+1} = -\omega^{n+1}. \quad (14)$$

For the space discretization of problem (13)–(14), we adopt a Finite Volume (FV) approximation that is derived directly from the integral form of the governing equations. For this purpose, we partition the computational domain Ω into cells or control volumes Ω_i , with $i =$

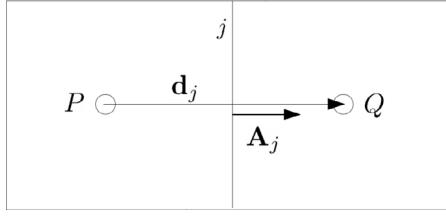


Fig. 1. Close-up view of two orthogonal control volumes in a 2D configuration.

$1, \dots, N_c$, where N_c is the total number of cells in the mesh. The integral form of Eq. (13) for each volume Ω_i is given by:

$$\begin{aligned} \frac{1}{\Delta t} \int_{\Omega_i} \omega^{n+1} d\Omega + \int_{\Omega_i} \nabla \cdot ((\nabla \times \psi^n) \omega^{n+1}) d\Omega \\ - \frac{1}{Re} \int_{\Omega_i} \Delta \omega^{n+1} d\Omega = \int_{\Omega_i} b^{n+1} d\Omega. \end{aligned} \quad (15)$$

By applying the Gauss-divergence theorem, Eq. (15) becomes:

$$\begin{aligned} \frac{1}{\Delta t} \int_{\Omega_i} \omega^{n+1} d\Omega + \int_{\partial\Omega_i} ((\nabla \times \psi^n) \omega^{n+1}) \cdot d\mathbf{A} \\ - \frac{1}{Re} \int_{\partial\Omega_i} \nabla \omega^{n+1} \cdot d\mathbf{A} = \int_{\Omega_i} b^{n+1} d\Omega, \end{aligned} \quad (16)$$

where \mathbf{A} is the surface vector associated with the boundary of Ω_i .

Let \mathbf{A}_j be the surface vector of each face of the control volume, with $j = 1, \dots, M$. Each term in Eq. (16) is approximated as follows:

- Convective term:

$$\int_{\partial\Omega_i} ((\nabla \times \psi^n) \omega^{n+1}) \cdot d\mathbf{A} \approx \sum_j \left((\nabla \times \psi_j^n) \omega_j^{n+1} \right) \cdot \mathbf{A}_j = \sum_j \varphi_j^n \omega_j^{n+1}, \quad (17)$$

$$\varphi_j^n = (\nabla \times \psi_j^n) \cdot \mathbf{A}_j. \quad (18)$$

In (17), $\nabla \times \psi_j^n$ is the extrapolated convective velocity and ω_j^{n+1} is the vorticity, both relative to the centroid of each control volume face. In (18), φ_j^n is the convective flux associated to $\nabla \times \psi^n$ through face j of the control volume. In OpenFOAM® solvers, the convective flux at the cell faces is typically a linear interpolation of the values from the adjacent cells. We also need to approximate ω^{n+1} at cell face j . Different interpolation methods can be applied, including central, upwind, second order upwind, and blended differencing schemes [28]. In this work, we use a Central Differencing (CD) scheme.

- Diffusion term:

$$\int_{\partial\Omega_i} \nabla \omega^{n+1} \cdot d\mathbf{A} \approx \sum_j (\nabla \omega^{n+1})_j \cdot \mathbf{A}_j,$$

where $(\nabla \omega^{n+1})_j$ is the gradient of ω^{n+1} at face j . Let us briefly explain how $(\nabla \omega^{n+1})_j$ is approximated with second order accuracy on a structured, orthogonal mesh. Let P and Q be two neighboring control volumes (see Fig. 1). The term $(\nabla \omega^{n+1})_j$ is evaluated by subtracting the value of vorticity at the cell centroid on the P -side of the face (denoted with ω_P^{n+1}) from the value of vorticity at the centroid on the Q -side (denoted with ω_Q^{n+1}) and dividing by the magnitude of the distance vector \mathbf{d}_j connecting the two cell centroids:

$$(\nabla \omega^{n+1})_j \cdot \mathbf{A}_j = \frac{\omega_Q^{n+1} - \omega_P^{n+1}}{|\mathbf{d}_j|} |\mathbf{A}_j|.$$

Let us denote with ω_i^{n+1} and b_i^{n+1} the average vorticity and source term in control volume Ω_i , respectively. Moreover, we denote with $\omega_{i,j}^{n+1}$ the vorticity associated to the centroid of face j normalized by the

volume of Ω_i . Then, the discretized form of Eq. (16), divided by the control volume Ω_i , can be written as:

$$\frac{1}{\Delta t} \omega_i^{n+1} + \sum_j \varphi_j^n \omega_{i,j}^{n+1} - \frac{1}{Re} \sum_j (\nabla \omega_i^{n+1})_j \cdot \mathbf{A}_j = b_i^{n+1}. \quad (19)$$

Next, we deal with the space approximation of the Eq. (14). After using Gauss-divergence theorem, the integral form of Eq. (14) reads:

$$-\int_{\partial\Omega_i} \nabla \psi^{n+1} \cdot d\mathbf{A} = \int_{\Omega_i} \omega^{n+1} d\Omega. \quad (20)$$

Once we approximate the integrals and divide by the control volume Ω_i , Eq. (20) becomes:

$$-\sum_j (\nabla \psi_i^{n+1})_j \cdot \mathbf{A}_j = \omega_i^{n+1}. \quad (21)$$

In Eq. (21), $(\nabla \psi_i^{n+1})_j$ is the gradient of ψ^{n+1} at faces j and it is approximated in the same way as $(\nabla \omega_i^{n+1})_j$. Finally, the fully discretized form of problem (13)–(14) is given by system (19), (21).

As mentioned in Section 1, for the implementation of the numerical scheme described in this section we chose the finite volume C++ library OpenFOAM® [21].

3. The reduced order model

The main idea of reduced order modeling for parametrized PDEs is the assumption that solutions live in a low dimensional manifold. Thus, any solution can be approximated as a linear combination of a reduced number of global basis functions.

We approximate vorticity field ω and stream function ψ as linear combinations of the dominant modes (basis functions), which are assumed to be dependent on space variables only, multiplied by scalar coefficients that depend on time and/or parameters. We arrange all the parameters the problem depends upon in a vector $\boldsymbol{\pi}$ that belongs to a d -dimensional parameter space \mathcal{P} in \mathbb{R}^d , where d is the number of parameters. Thus, we have:

$$\omega \approx \omega_r = \sum_{i=1}^{N_\omega^r} \beta_i(\boldsymbol{\pi}, t) \varphi_i(\mathbf{x}), \quad \psi \approx \psi_r = \sum_{i=1}^{N_\psi^r} \gamma_i(\boldsymbol{\pi}, t) \xi_i(\mathbf{x}). \quad (22)$$

In (22), N_Φ^r , $\Phi = \omega, \psi$, denotes the cardinality of a reduced basis for the space field Φ belongs to. We remark that we consider different coefficients for the approximation of the vorticity ω and stream function ψ fields, unlike previous works [12,13,15–18]. This choice will be justified numerically in Section 4.2.

Remark 3.1. As mentioned earlier, we only consider a body force given by the product between a space dependent function and a time dependent function. For the stream function-vorticity formulation, this means:

$$F(x, y, z) = F_2(t) F_1(x, y).$$

Thanks to this assumption, the forcing term is already expressed in the form of (22) and does not require further treatment.

Using (22) to approximate ω^{n+1} and ψ^{n+1} in (13)–(14), we obtain

$$\frac{1}{\Delta t} \omega_r^{n+1} + \nabla \cdot ((\nabla \times \psi_r^*) \omega_r^{n+1}) - \frac{1}{Re} \Delta \omega_r^{n+1} = b_r^{n+1}, \quad (23)$$

$$\nabla \psi_r^{n+1} = -\omega_r^{n+1}, \quad (24)$$

where $b_r^{n+1} = F^{n+1} + \omega_r^n / \Delta t$ and we set $\psi_r^* = \psi_r^n$.

In the literature, one can find several techniques to generate the reduced basis spaces, e.g., Proper Orthogonal Decomposition (POD), the Proper Generalized Decomposition and the Reduced Basis with a greedy sampling strategy. See, e.g., [29–35]. We generate the reduced basis spaces with the method of snapshots. Next, we briefly describe how this method works.

Let $\mathcal{K} = \{\pi^1, \dots, \pi^{N_k}\}$ be a finite dimensional training set of samples chosen inside the parameter space \mathcal{P} . We solve the FOM described in Section 2 for each $\pi^k \in \mathcal{K}$ and for each time instant $t^j \in \{t^1, \dots, t^{N_t}\} \subset (t_0, T]$. The snapshots matrices are obtained from the full-order snapshots:

$$\mathcal{S}_\Phi = [\Phi(\pi^1, t^1), \dots, \Phi(\pi^{N_k}, t^{N_t})] \in \mathbb{R}^{N_\Phi^h \times N_s} \quad \text{for } \Phi = \{\omega_h, \psi_h\}, \quad (25)$$

where $N_s = N_t \cdot N_k$ is the total number of the snapshots, N_Φ^h is the dimension of the space Φ belong to in the FOM, and the subscript h indicates a solution computed with the FOM. The POD problem consists in finding, for each value of the dimension of the POD space $N_{POD} = 1, \dots, N_s$, the scalar coefficients $a_1^1, \dots, a_1^{N_s}, \dots, a_{N_s}^1, \dots, a_{N_s}^{N_s}$ and functions $\zeta_1, \dots, \zeta_{N_s}$, that minimize the error between the snapshots and their projection onto the POD basis. In the L^2 -norm, we have

$$E_{N_{POD}} = \arg \min \sum_{i=1}^{N_s} \|\Phi_i - \sum_{k=1}^{N_{POD}} a_i^k \zeta_k\| \quad \forall N_{POD} = 1, \dots, N_s$$

with $(\zeta_i, \zeta_j)_{L_2(\Omega)} = \delta_{i,j} \quad \forall i, j = 1, \dots, N_s.$ (26)

It can be shown [36] that problem (26) is equivalent to the following eigenvalue problem

$$\mathcal{C}^\Phi \mathbf{Q}^\Phi = \mathbf{Q}^\Phi \Lambda^\Phi, \quad (27)$$

$$C_{ij}^\Phi = (\Phi_i, \Phi_j)_{L_2(\Omega)} \quad \text{for } i, j = 1, \dots, N_s, \quad (28)$$

where \mathcal{C}^Φ is the correlation matrix computed from the snapshot matrix \mathcal{S}_Φ , \mathbf{Q}^Φ is the matrix of eigenvectors and Λ^Φ is a diagonal matrix whose diagonal entries are the eigenvalues of \mathcal{C}^Φ . Then, the basis functions are obtained as follows:

$$\zeta_i = \frac{1}{N_s \Lambda_i^\Phi} \sum_{j=1}^{N_s} \Phi_j Q_{ij}^\Phi. \quad (29)$$

The POD modes resulting from the aforementioned methodology are:

$$\mathcal{L}_\Phi = [\zeta_1, \dots, \zeta_{N_\Phi^h}] \in \mathbb{R}^{N_\Phi^h \times N_\Phi^r}, \quad (30)$$

where the values of $N_\Phi^r < N_s$ are chosen according to the eigenvalue decay of the vectors of eigenvalues Λ . Then, the reduced order model can be obtained through a Galerkin projection of the governing equations onto the POD spaces.

In order to write the algebraic system associated with the reduced problem (23)–(24), we introduce the following matrices:

$$M_{r_{ij}} = (\varphi_i, \varphi_j)_{L_2(\Omega)}, \quad \widetilde{M}_{r_{ij}} = (\xi_i, \varphi_j)_{L_2(\Omega)}, \quad A_{r_{ij}} = (\varphi_i, \Delta \varphi_j)_{L_2(\Omega)}, \quad (31)$$

$$B_{r_{ij}} = (\xi_i, \Delta \xi_j)_{L_2(\Omega)}, \quad G_{r_{ijk}} = (\varphi_i, \nabla \cdot ((\nabla \times \xi_j) \varphi_k))_{L_2(\Omega)}, \quad (32)$$

$$H_{r_{ij}} = (\varphi_i, F_1)_{L_2(\Omega)}, \quad (33)$$

where φ_i and ξ_i are the basis functions in (22). At time t^{n+1} , the reduced algebraic system for (23)–(24) reads: given β^n and γ^n find vectors β^{n+1} and γ^{n+1} containing the values of coefficients β_i and γ_i in (22) at time t^{n+1} such that

$$M_r \left(\frac{\beta^{n+1} - \beta^n}{\Delta t} \right) + (\gamma^n)^T G_r \beta^{n+1} - \frac{1}{Re} A_r \beta^{n+1} = H_r F_2^{n+1}, \quad (34)$$

$$B_r \gamma^{n+1} + \widetilde{M}_r \beta^{n+1} = 0. \quad (35)$$

Finally, the initial conditions for the ROM algebraic system (34)–(35) are obtained performing a Galerkin projection of the initial full order condition onto the POD basis spaces:

$$\beta_i^0 = (\omega_0, \varphi_i)_{L_2(\Omega)}. \quad (36)$$

Remark 3.2. We consider homogeneous boundary conditions (7)–(8) and (9)–(10). So, the approximated vorticity ω , and stream function ψ , automatically satisfy the boundary conditions and no special treatment (such as lifting function and penalty methods [9–11,37]) is necessary.

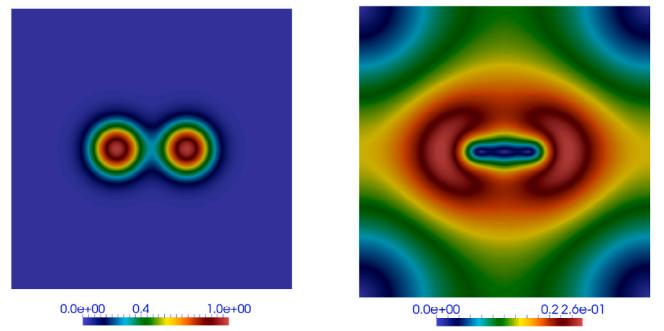


Fig. 2. Vortex merger: initial conditions for ω_0 (left) and u_0 (right).

4. Numerical results

For the validation of FOM and ROM, we choose the two benchmarks described below.

Vortex merger. This widely used benchmark consists in fluid motion induced by a pair of co-rotating vortices separated from each other by a certain distance. One of the reasons why this test has been extensively investigated in two-dimensions is that it explains the average inverse energy and direct enstrophy cascades observed in two-dimensional turbulence [38]. The computational domain is a $2\pi \times 2\pi$ rectangle. The initial condition for the vortex merger test case is given by:

$$\omega(x, y, 0) = \omega_0(x, y) = \sum_{i=1}^2 e^{-\pi[(x-x_i)^2 + (y-y_i)^2]},$$

where $(x_1, y_1) = (3\pi/4, \pi)$ and $(x_2, y_2) = (5\pi/4, \pi)$ are the initial locations of the centers of the vortices. Fig. 2 shows the initial vorticity ω_0 (left) and corresponding initial velocity u_0 (right). We note that the initial condition $u_0(x, y) = u(x, y, 0) = \nabla \times \psi_0$ is computed by solving $\Delta \psi_0 = \omega_0$. We let the system evolve until time $T = 20$. We enforce boundary conditions (7)–(8) for the $\psi - \omega$ formulation and (4) for the formulation in primitive variables. Following [12,13,15–17], we consider a computational grid with $256^2 = 65536$ cells for all the simulations.

North Atlantic Ocean. This second benchmark features a more complex geometry representing the North Atlantic Ocean. The geometry was reconstructed from satellite images of the geographic area under consideration. More details on this are available in [39–41]. Therein, the North Atlantic Ocean dynamics have been described with the quasi-geostrophic equations discretized in space with finite elements. Since our focus is not on the quasi-geostrophic equations, we will neglect the Coriolis effect. We set $f = F = 0$ and enforce the following initial condition:

$$\omega(x, y, 0) = \omega_0(x, y) = \sum_{i=1}^3 e^{(-7.5e-3[(x-x_i)^2 + (y-y_i)^2])}, \quad (37)$$

where $(x_1, y_1) = (80, 70)$, $(x_2, y_2) = (155, 125)$ and $(x_3, y_3) = (160, 70)$ are the initial locations of the centers of the vortices. Fig. 3 (left) shows the initial vorticity ω_0 and the corresponding initial velocity $u_0 = \nabla \times \psi_0$, with $\Delta \psi_0 = \omega_0$, is reported in Fig. 3 (center). Notice that because of initial condition (37) this test could be considered as an extension of the vortex merger benchmark in a more complex domain and with a more complex dynamics. The associated Reynolds number is about 5, which is a typical value for oceanographic simulations (see, e.g., [42]). We set final time $T = 200$ and focus on the solution in time interval $[100, 200]$, when the vortex merging is at an advanced stage. We enforce boundary condition (3) for the solver in primitive variables as in [39] and boundary conditions (9)–(10) for the stream function-vorticity formulation as in [39–41]. Following [39–41], we use a computational grid with 2425 triangles shown in Fig. 3 (right).

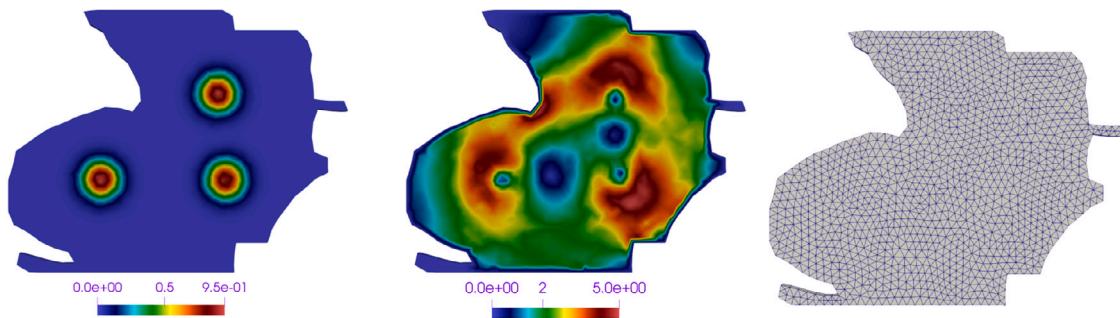


Fig. 3. North Atlantic Ocean: initial conditions for ω_0 (left) and u_0 (center) and computational mesh (right).

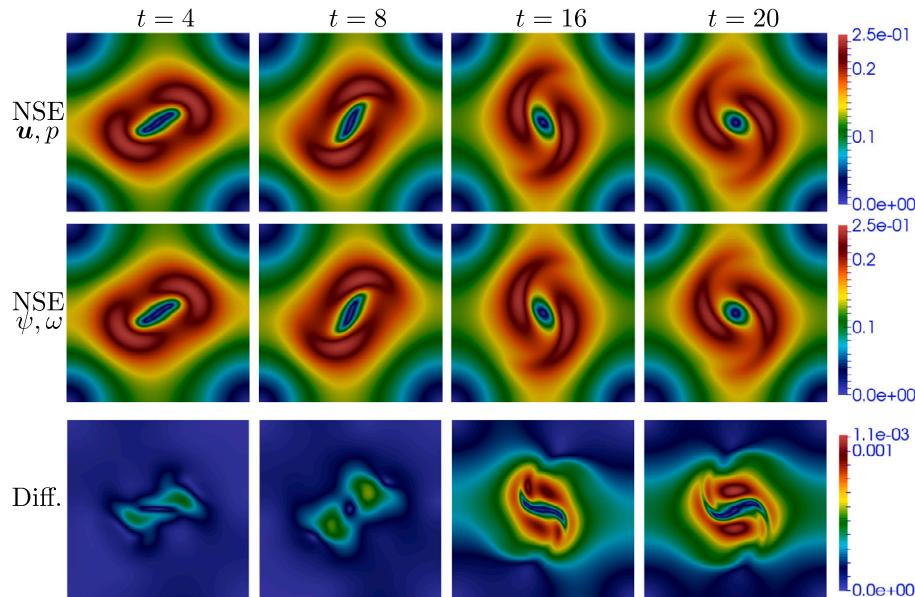


Fig. 4. FOM validation with the vortex merger benchmark: velocity u computed by the solver in velocity–pressure formulation (first row) and stream function-vorticity formulation (second row), and difference between the two fields in absolute value (third row) at $t = 4$ (first column), $t = 8$ (second column), $t = 16$ (third column) and $t = 20$ (fourth column).

The vortex merger benchmark is used for the validation of the FOM in Section 4.1 and to test time reconstruction (Section 4.2) and physical parameterization (Section 4.3) for the ROM. The ROM time reconstruction is further assessed with the more challenging North Atlantic Ocean benchmark. For both benchmarks, we show a comparison between ROM for the Navier–Stokes equations in the stream function-vorticity formulation and ROM for the Navier–Stokes equations in primitive variables. For the latter, we use a supremizer enrichment strategy for the stabilization of the pressure. The reader is referred to [10,11,27] for more details.

4.1. Validation of the FOM with the vortex merger benchmark

Let us start with the validation of our implementation of the stream function-vorticity formulation at the FOM level. We compare the results obtained with such formulation against the results produced by the standard Navier–Stokes solver in OpenFOAM *icoFoam*, which is based on a partitioned algorithm called PISO [22]. We set $\Delta t = 0.01$, $f = F = 0$ and $Re = 800$ [15,16]. For the simulations with *icoFoam*, we enforce boundary condition (4). The partitioned algorithm requires also a boundary condition for the pressure problem: we set $\nabla p \cdot n = 0$ on $\partial\Omega$.

Figs. 4, 5, and 6 display a qualitative comparison in terms of u , ψ , and ω computed by the solvers in primitive variables and in stream function-vorticity formulation at four different times. As we can see from these figures, the solutions are very close to each other with the maximum relative difference in absolute value not exceeding $4.4e - 3$ for u , $2.5e - 3$ for ψ , and $6.8e - 3$ for ω .

With these results, we consider the FOM validated. Next, we are going to validate our ROM approach. Our goal is a thorough assessment of our ROM model on two fronts: (i) the reconstruction of the time evolution of the flow field and (ii) a physical parametric setting. Let us start from the former.

4.2. Validation of the ROM: time reconstruction

Let us start with the vortex merger benchmark. We collect 250 FOM snapshots, one every 0.08, i.e. we use an equispaced grid in time. Fig. 7 shows the eigenvalues decay for the stream function and the vorticity, as well as for the velocity and pressure. We observe that the eigenvalues decay for ω is much slower than the eigenvalues decay for ψ . We suspect that this larger number of modes for ω is due to the richer structure of ω . On the other hand, we see that the eigenvalues decays for u and p are very close to each other. We set the threshold for the selection of the eigenvalues to $1e - 5$, resulting in 6 modes for ψ and 14 modes for ω , and in 9 modes for u and 8 modes for p . We consider the same dimension of the pressure space for the supremizer space.

We calculate the relative L^2 error in percentage:

$$E_\Phi(t) = 100 \cdot \frac{\|\Phi_h(t) - \Phi_r(t)\|_{L^2(\Omega)}}{\|\Phi_h(t)\|_{L^2(\Omega)}}, \quad (38)$$

where Φ_h is a field computed with the FOM (ψ_h, ω_h, u_h or p_h) and Φ_r is the corresponding field computed with the ROM (ψ_r, ω_r, u_r or p_r).

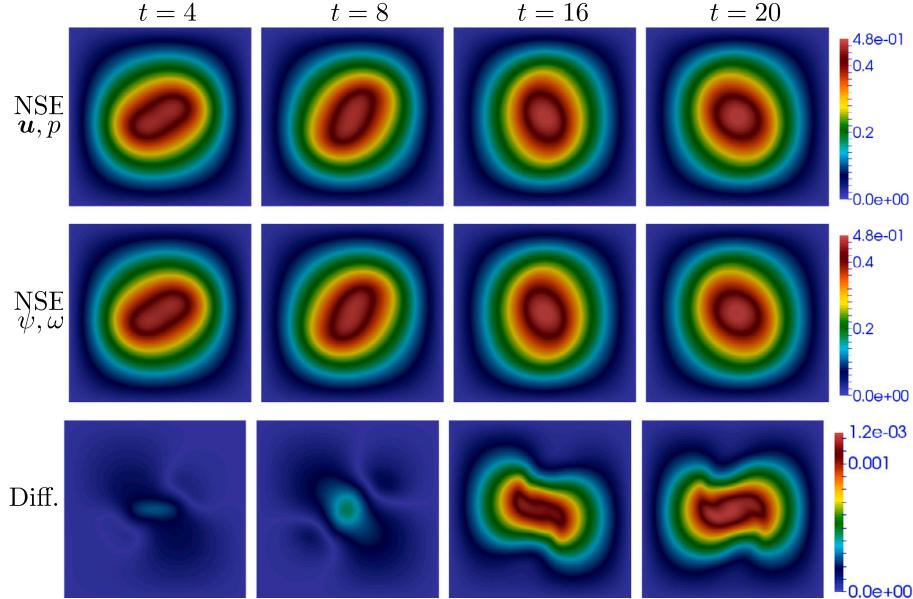


Fig. 5. FOM validation with the vortex merger benchmark: stream function ψ computed by the solver in velocity-pressure formulation (first row) and stream function-vorticity formulation (second row), and difference between the two fields in absolute value (third row) at $t = 4$ (first column), $t = 8$ (second column), $t = 12$ (third column) and $t = 20$ (fourth column).

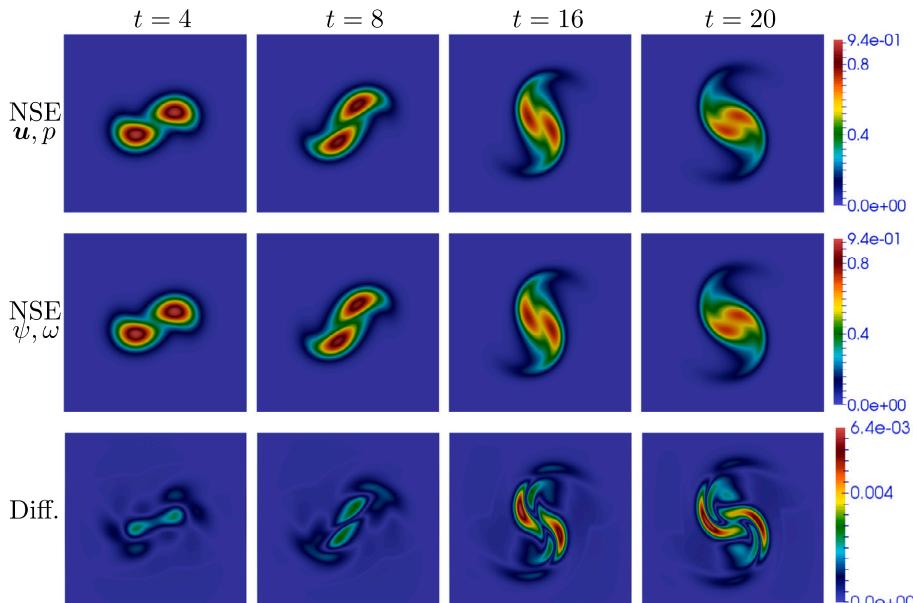


Fig. 6. FOM validation with the vortex merger benchmark: vorticity ω computed by the solver in velocity-pressure formulation (first row) and stream function-vorticity formulation (second row), and difference between the two fields in absolute value (third row) at $t = 4$ (first column), $t = 8$ (second column), $t = 16$ (third column) and $t = 20$ (fourth column).

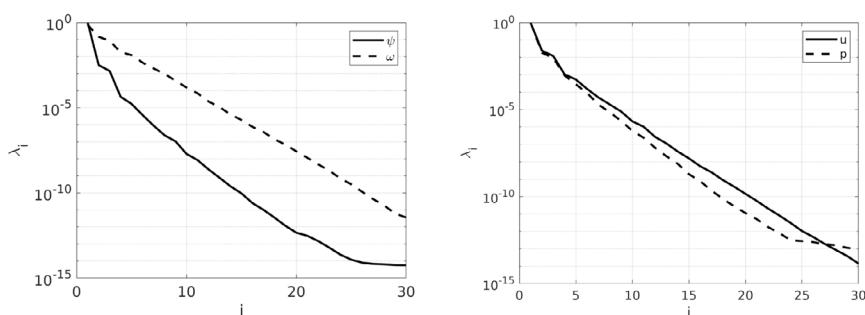


Fig. 7. ROM validation — time reconstruction for the vortex merger benchmark: eigenvalue decay for the stream function and the vorticity (left) and for the velocity and pressure (right).

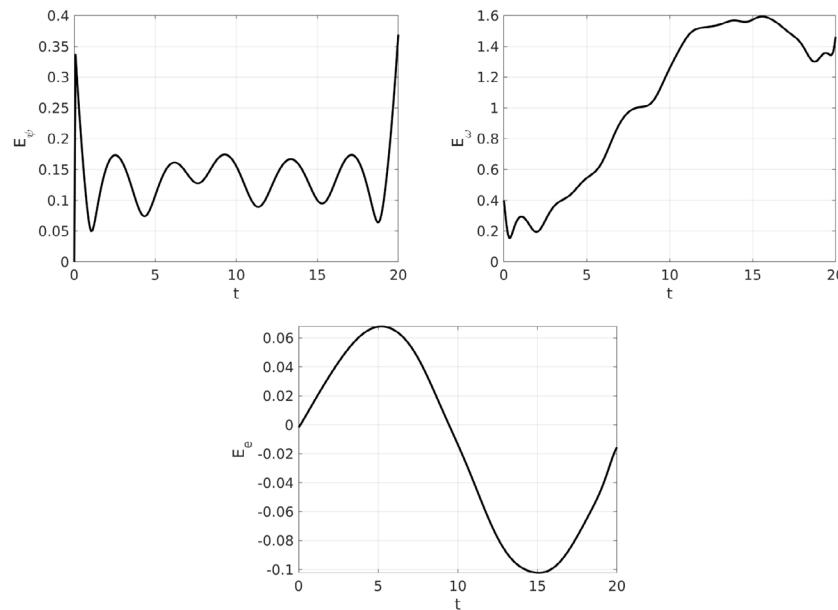


Fig. 8. ROM validation — time reconstruction for the vortex merger benchmark: evolution of error (38) for stream function ψ (top left) and vorticity ω (top right), and error for enstrophy e (39) (bottom).

p_r). Moreover, we evaluate the relative error in percentage for the enstrophy $e = \int_{\Omega} \omega^2 d\Omega$, i.e.:

$$E_e(t) = 100 \cdot \frac{e_h(t) - e_r(t)}{e_h(t)}, \quad (39)$$

and the kinetic energy $k = \int_{\Omega} |\mathbf{u}|^2 d\Omega$, i.e.

$$E_k(t) = 100 \cdot \frac{k_h(t) - k_r(t)}{k_h(t)}, \quad (40)$$

where e_h and e_r , k_h and k_r are the values of the enstrophy and kinetic energy computed by the FOM and the ROM.

Fig. 8 shows error (38) for the stream function and vorticity and error for the enstrophy (39) over time. Fig. 9 shows error (38) for the velocity and pressure and error for the kinetic energy (40) over time. We see that all relative errors in percentage achieve very low values. In particular, over the entire time interval the error for ψ is lower than 0.4%, the error for ω is lower than 1.6%, and the error for the enstrophy is lower than 0.1% in absolute value. Similarly, the error for \mathbf{u} is lower than 0.6%, the error for p is lower than 0.7%, and the error for the kinetic energy is lower than 0.04% in absolute value. So, we conclude that the performance of the ROM for the stream function-vorticity formulation is comparable to the ROM for the formulation in primitive variables.

In order to justify our choice to use different reduced coefficients to approximate stream-function and vorticity in (22), we show in Fig. 10 the time evolution of the first three reduced coefficients for ψ and ω : the differences are significant. Thus, using the same reduced coefficients would lead to a less accurate reconstruction of ψ and ω .

Finally, we compare the solutions computed by FOM and ROM. Figs. 11 and 12 display such comparison for ψ and ω at four different times, respectively. From these figures, we see that our ROM approach provides a good global reconstruction of both stream function and vorticity. In fact, the maximum relative difference in absolute value does not exceed $4.7e-3$ for ψ and $1.7e-2$ for ω .

We conclude this subsection by proving some information about the efficiency of our ROM approach for the $\psi-\omega$ formulation. The total CPU time required by a FOM simulation is about 64 s, while the computation of the modal coefficients over the entire time window of interest takes 0.47 s. The resulting speed-up is about 136.

Next, we consider the North Atlantic Ocean benchmark. We set $\Delta t = 0.001$ and collect 1000 FOM snapshots, one every 0.1, in the time

interval [100, 200]. Fig. 13 shows the eigenvalues decay for the stream function and the vorticity, as well as for the velocity and pressure. Also for this benchmark, the eigenvalues decay for ω is much slower than the eigenvalues decay for ψ and the eigenvalues decays for \mathbf{u} and p are close to each other. We set the threshold for the selection of the eigenvalues to $1e-5$, resulting in 9 modes for ψ and 35 modes for ω , and in 16 modes for \mathbf{u} and 12 modes for p and supremizer. The number of modes to be retained for all the variables is larger than for the vortex merger benchmark. This is an indication of the fact that the system has a richer modal content and more complex flow patterns.

Fig. 14 shows error (38) for the stream function and vorticity and error for the enstrophy (39) over time. Fig. 15 shows error (38) for velocity and pressure and error for the kinetic energy (40) over time. As for the vortex merger benchmark, we see all the relative errors in percentage achieve very low values. In particular, over the entire time interval the error for ψ is lower than 1.9%, the error for ω is lower than 2.3%, and the error for the enstrophy is lower than 0.5% in absolute value. The error for \mathbf{u} is lower than 3.2%, the error for p is lower than 4.4%, and the error for the kinetic energy is lower than 3.5% in absolute value. Also for this benchmark, we can conclude that the ROM for the stream function-vorticity formulation and the ROM for primitive variables are comparable in terms of accuracy.

Finally, we compare the solutions computed by FOM and ROM. Figs. 16 and 17 display such comparison for ψ and ω at three different times. From these figures, we see that our ROM approach provides a good global reconstruction of both stream function and vorticity. In fact, the maximum relative difference in absolute value does not exceed $5.4e-1$ for ψ and $1.1e-2$ for ω .

4.3. Validation of the ROM: physical parameterization for the vortex merger benchmark

In this section, we are going to consider a physical parametric setting. We set an arbitrary array of decaying Taylor–Green vortices as source term in the vorticity Eq. (5) given by

$$F = -\gamma e^{-t/Re} \cos(3x) \cos(3y), \quad (41)$$

where γ is the strength of the source term. We consider Re and γ as the control parameters. We remark that a Re parameterization has been considered in [13] and parameterization with respect to both Re and

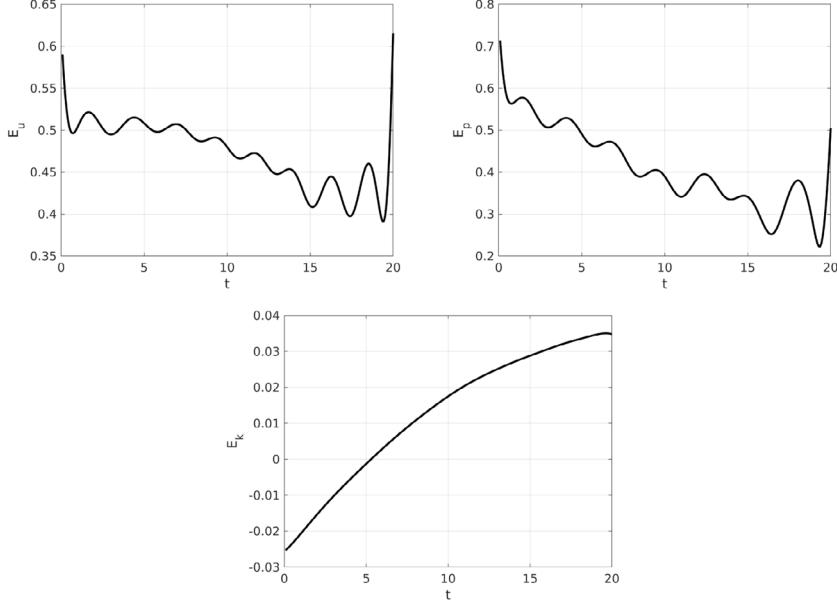


Fig. 9. ROM validation — time reconstruction for the vortex merger benchmark: evolution of error (38) for velocity u (top left) and pressure p (top right), and error for kinetic energy k (40) (bottom).

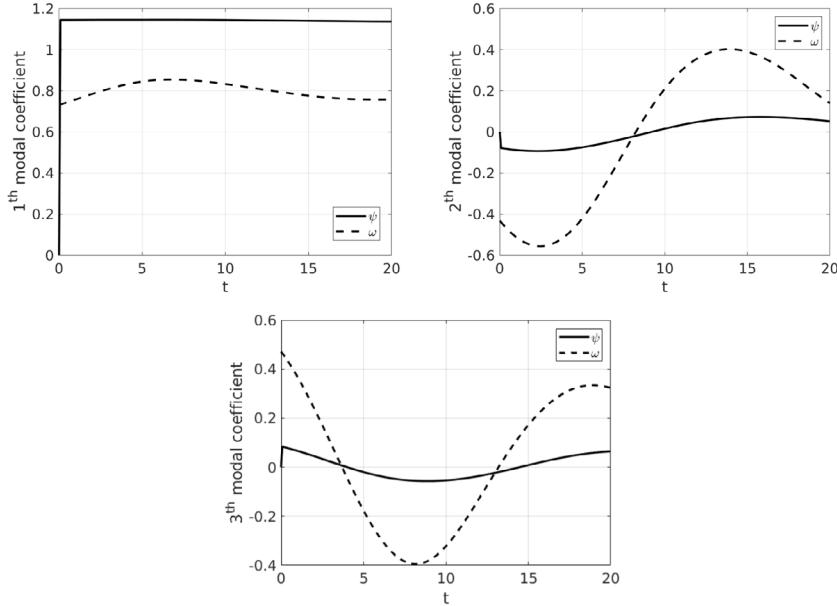


Fig. 10. ROM validation — time reconstruction for the vortex merger benchmark: evolution of the first three reduced coefficients for ψ and ω .

γ has been studied in [15–17] where the aim was to develop a ROM framework to account for hidden physics through data-driven strategies based on machine learning techniques. We will proceed by fixing one parameter (e.g., γ) and varying the other (Re if γ is fixed). Since time is treated as a parameter, when we vary one physical parameter we are effectively running a study in a two-dimensional parameter space. This is what is typically done for a parametric study involving a time dependent PDE problem [10,11,13,15–17].

In order to reduce the offline cost, i.e. the time needed to collect the FOM snapshots and the time required to compute the POD basis space, we will focus on the first half of the time interval of interest considered in Section 4.2, i.e. $(0, 10]$. In addition, we recall that the ROM time reconstruction has been thoroughly validated in Section 4.2.

Let us start with the parameterization with respect to Re and set $\gamma = 0.09$. To train the ROM, we choose a uniform sample distribution

in the range $Re \in [200, 800]$ with 4 sampling points: 200, 400, 600, and 800. For each value of the Reynolds number in the training set, a simulation is run over time interval $(0, 10]$. Fig. 18 displays the stream function and vorticity fields computed by the FOM throughout the training set under consideration at $t = 10$. We observe that the stream function does not significantly vary as Re changes, while the differences in the vorticity are more evident.

Based on the results presented for the time reconstruction (Section 4.2), the snapshots are collected every 0.08 s. So, we collect a total of $4 \times 125 = 500$ snapshots. We set the threshold for the selection of the eigenvalues to $1e-5$, resulting in 6 modes for ψ and 11 modes for ω . We take three different test values to evaluate the performance of the parametrized ROM: one value ($Re = 500$) in the range under consideration but not in the training set and two values ($Re = 100, 1000$) outside the range under consideration. The latter cases

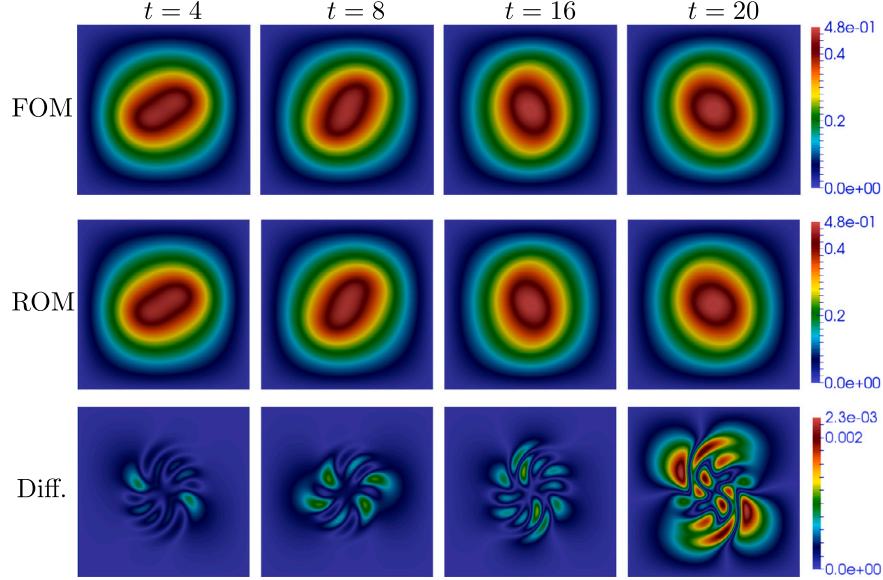


Fig. 11. ROM validation — time reconstruction for the vortex merger benchmark: stream function ψ computed by the FOM (first row) and the ROM (second row), and difference between the two fields in absolute value (third row) at times $t = 4$ (first column), $t = 8$ (second column), $t = 16$ (third column) and $t = 20$ (fourth column). We consider 6 modes.

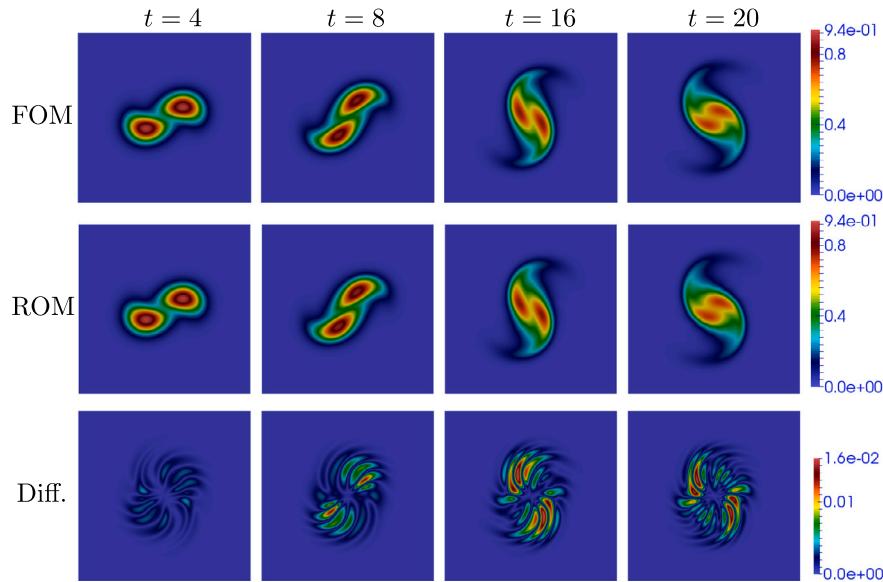


Fig. 12. ROM validation — time reconstruction for the vortex merger benchmark: vorticity ω computed by the FOM (first row) and the ROM (second row), and difference between the two fields in absolute value (third row) at times $t = 4$ (first column), $t = 8$ (second column), $t = 16$ (third column) and $t = 20$ (fourth column). We consider 14 modes.

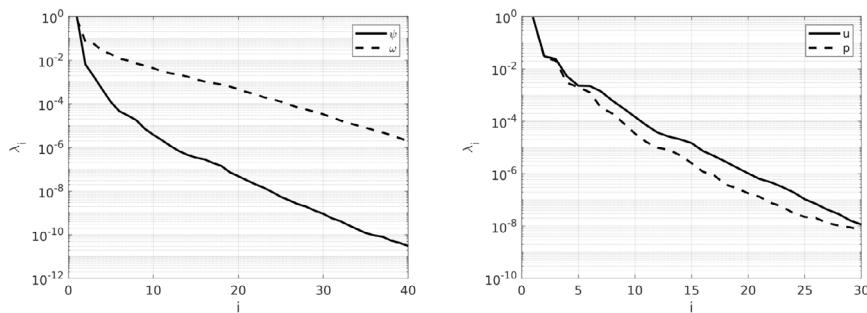


Fig. 13. ROM validation — time reconstruction for the North Atlantic Ocean benchmark: eigenvalue decay for the stream function and the vorticity (left) and for the velocity and pressure (right).

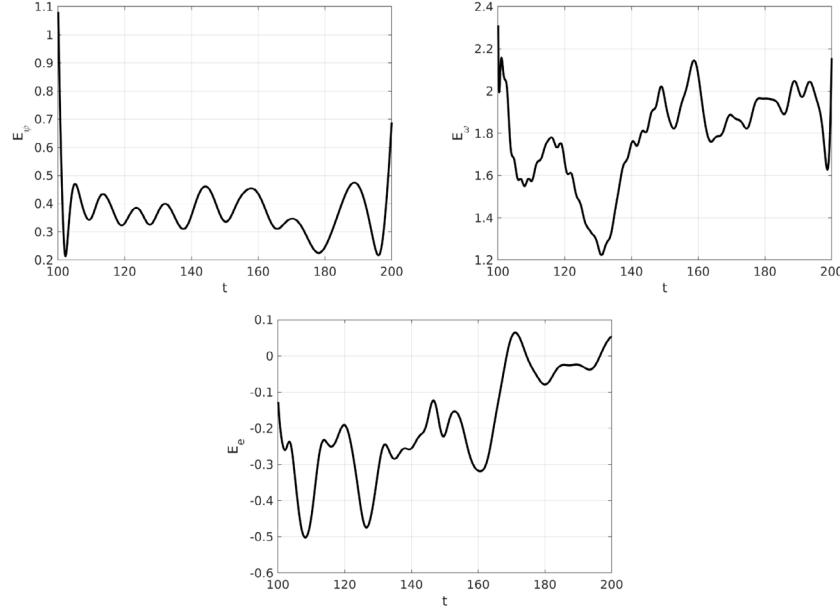


Fig. 14. ROM validation — time reconstruction for the North Atlantic Ocean benchmark: evolution of error (38) for stream function ψ (top left) and vorticity ω (top right), and error for enstrophy e (39) (bottom).

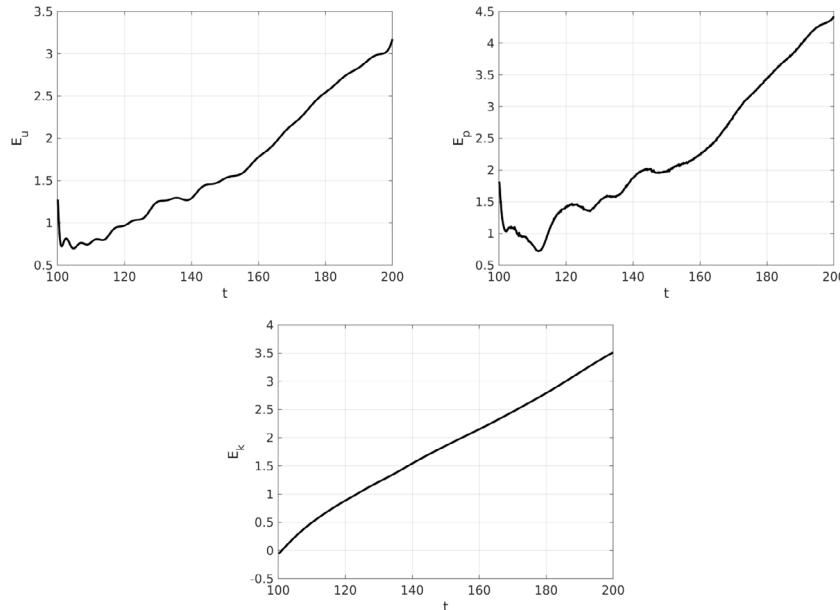


Fig. 15. ROM validation — time reconstruction for the North Atlantic Ocean benchmark: evolution of error (38) for velocity u (top left) and pressure p (top right), and error for kinetic energy k (39) (bottom).

are more challenging. Fig. 19 shows error (38) for the stream function and vorticity over time for the three values of Re . We see that for the interpolatory test value $Re = 500$ both the errors are below 1% over the entire time interval. As for the extrapolatory test values, the errors for $Re = 1000$ are comparable to the errors for $Re = 500$, while the errors for $Re = 100$ are much larger for $Re = 100$ (up to about 3% for ψ and up to about 7% for ω). The poorer performance of our ROM at $Re = 100$ could be due to the fact that the vorticity computed for $Re = 200$ looks pretty different from the vorticity at the higher Re included in the offline database (see Fig. 18). Thus, we suspect that more solutions for lower values of Re would have to be included into the training set in order to obtain a more accurate reconstruction of the flow field at $Re = 100$. Finally, we note that for a given Re the errors for the vorticity are larger than the errors for the stream function. Once

again, we suspect that this is due to the richer structure of ω (compare the bottom row of Fig. 18 with the top row) that might require a larger number of modes to obtain errors comparable to those for ψ .

Figs. 20 and 21 present a qualitative comparison of the solutions computed by FOM and ROM at $t = 10$ for the three test value of Re . We observe that our ROM approach provides a good global reconstruction of both stream function and vorticity. In fact, the maximum relative difference in absolute value does not exceed $3.1e-2$ for ψ and $6.7e-2$ for ω .

Next, we consider γ as a variable parameter and fix Re to 800. Similarly to the Re parameterization, the training of the ROM is carried out by a uniform sample distribution in the range $\gamma \in [0.06, 0.09]$ consisting of 4 sampling points: 0.06, 0.07, 0.08 and 0.09. For each value of γ inside the training set, a simulation is run over time interval

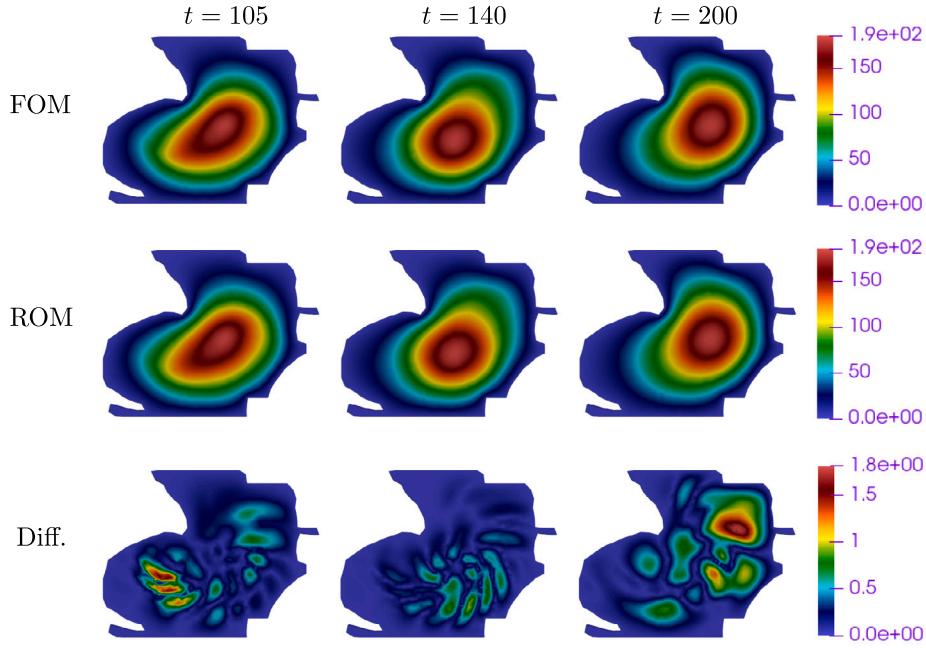


Fig. 16. ROM validation — time reconstruction for North Atlantic Ocean benchmark: stream function ψ computed by the FOM (first row) and the ROM (second row), and difference between the two fields in absolute value (third row) at times $t = 105$ (first column), $t = 140$ (second column) and $t = 200$ (third column). We consider 9 modes.

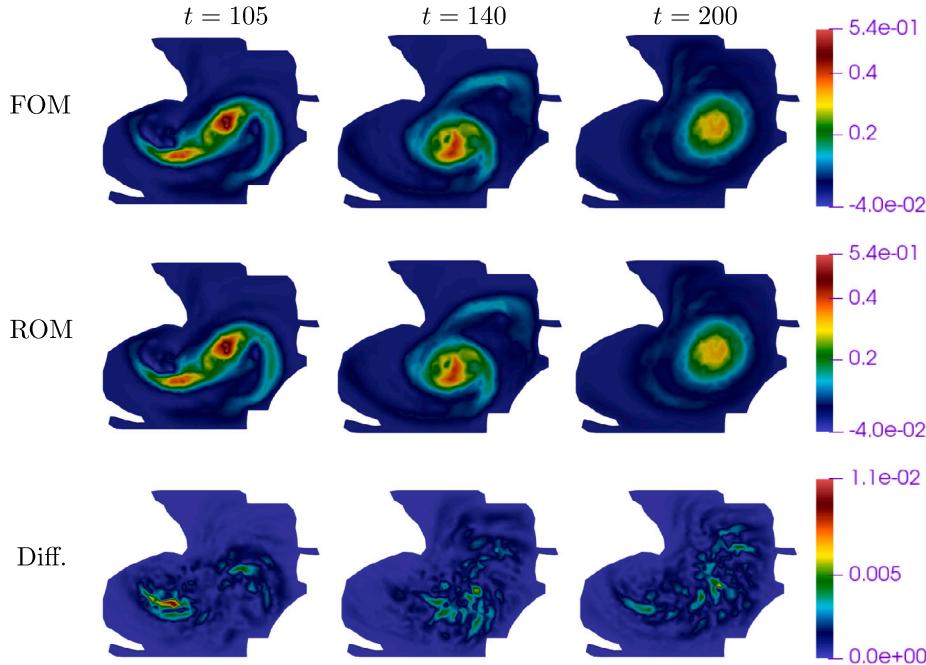


Fig. 17. ROM validation — time reconstruction for North Atlantic Ocean benchmark: vorticity ω computed by the FOM (first row) and the ROM (second row), and difference between the two fields in absolute value (third row) at times $t = 105$ (first column), $t = 140$ (second column) and $t = 200$ (third column). We consider 35 modes.

(0, 10]. Fig. 22 shows the stream function and vorticity fields computed by the FOM at $t = 10$ for the four sampling values of γ . We see that both stream function and vorticity do not significantly vary as γ changes value.

Analogously to what we have done for the previous parametric test case, the snapshots are collected every 0.08 s for a total of $4 \times 125 = 500$ snapshots. We set the threshold for the selection of the eigenvalues to $1e - 5$, which results in 6 modes for ψ and 12 modes for ω . Once again, we take three different test values to evaluate the performance of the parametrized ROM: $\gamma = 0.075$ (in the range under consideration but not in the training set) and $\gamma = 0.05, 0.1$ (outside the range under

consideration). Fig. 23 shows error (38) for the stream function and vorticity over time for these three values of γ . For the interpolatory test value ($\gamma = 0.075$), the error for ψ is lower than 0.4% and the error for ω is lower than 1% over the entire time interval of interest. As for the extrapolatory test values ($\gamma = 0.05, 0.1$), the error for ψ is lower than 1.2% and the error for ω is lower than 2.5%. So, unlike the Re parametric case, the errors obtained for all test values are comparable. This is obviously due to the fact that there is no abrupt change in the solution as γ varies in [0.06, 0.09] (see Fig. 22) and the POD space seems to include enough information for a very good reconstruction of the flow field also at values of γ right outside the training set.

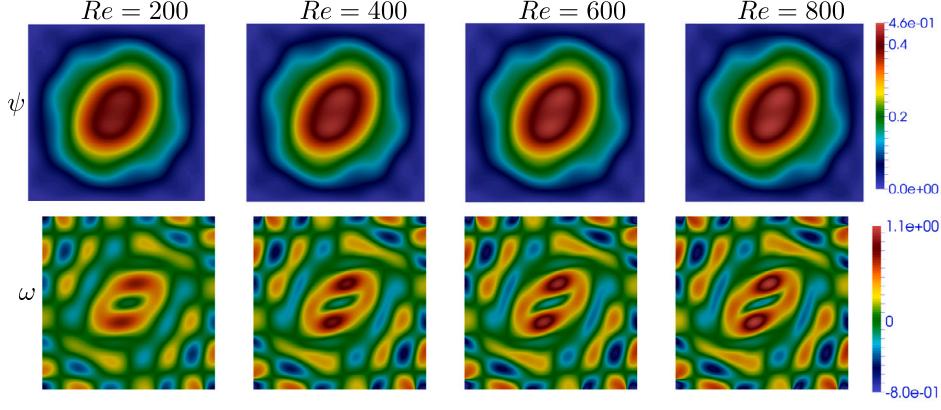


Fig. 18. ROM validation — Re parameterization: stream function ψ (first row) and vorticity ω (second row) computed by the FOM at $Re = 200$ (first column), $Re = 400$ (second column), $Re = 600$ (third column), and $Re = 800$ (fourth column), at time $t = 10$ for $\gamma = 0.09$.

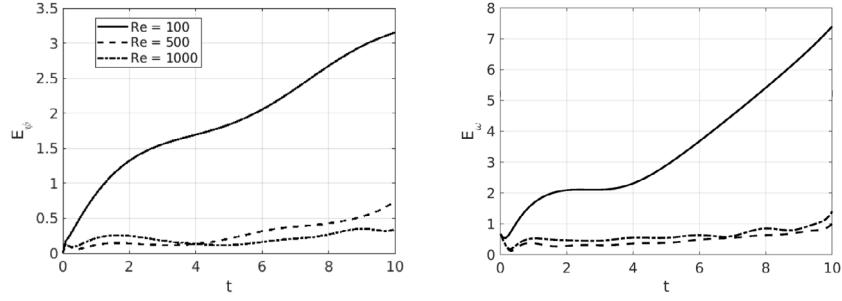


Fig. 19. ROM validation — Re parameterization: time history of error (38) for stream function ψ (left) and vorticity ω (right) for three different test values: $Re = 100$, $Re = 500$ and $Re = 1000$.

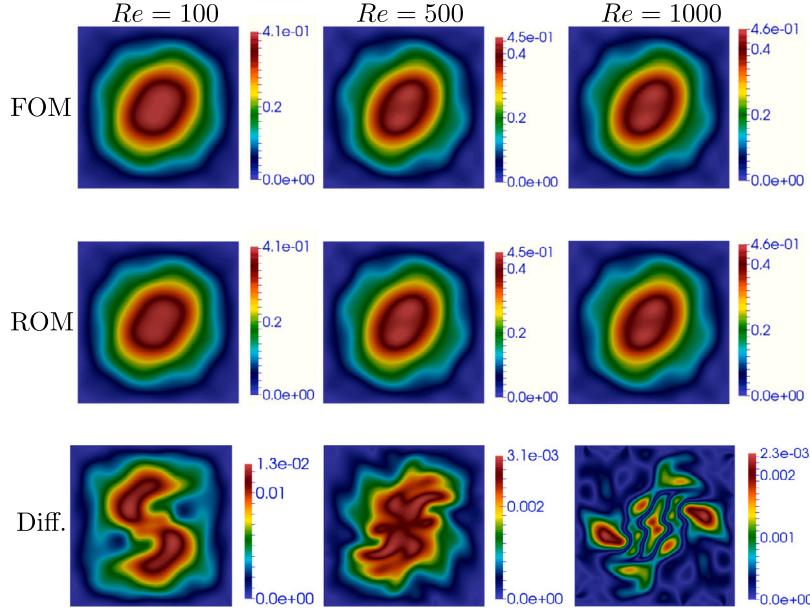


Fig. 20. ROM validation — Re parameterization: stream function ψ computed by the FOM (first row) and the ROM (second row), and difference between the two fields in absolute value (third row) for $Re = 100$ (first column), $Re = 500$ (second column) and $Re = 1000$ (third one) at time $t = 10$. Six modes for ψ were considered.

To conclude, we qualitatively compare the solutions computed by FOM and ROM for the three test values of γ at $t = 10$ in Figs. 24 and 25. Once again, we see that our ROM approach provides a good global reconstruction of both stream function and vorticity. In fact, the maximum relative difference in absolute value does not exceed $6.3e-3$ for ψ and $6.1e-2$ for ω .

5. Conclusions and future developments

This work presents a POD-Galerkin based Reduced Order Method for the Navier-Stokes equations in stream function-vorticity formulation within a Finite Volume environment. The main novelties of the proposed ROM approach are: (i) the use of different coefficients to approximate the stream function and vorticity fields and (ii) the use

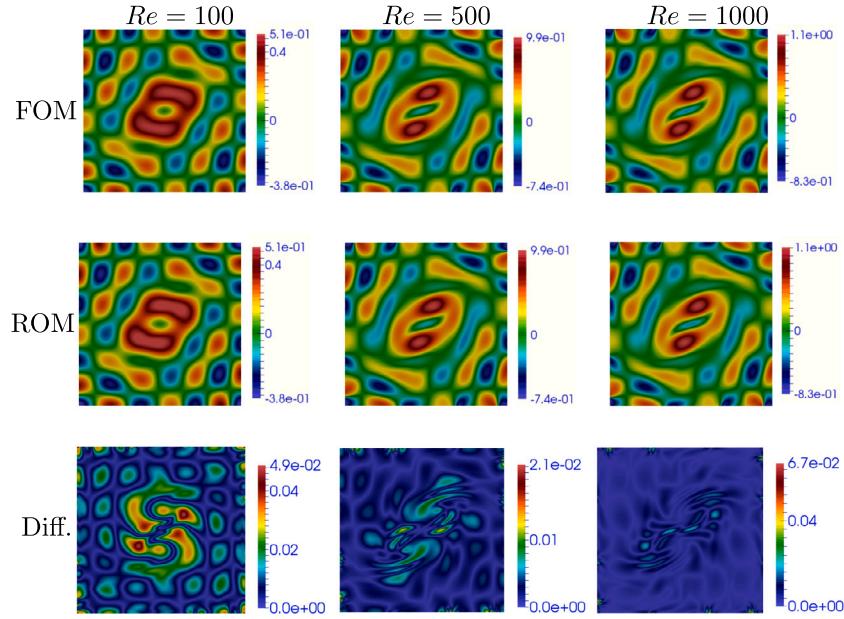


Fig. 21. ROM validation — Re parameterization: vorticity ω computed by the FOM (first row) and the ROM (second row), and difference between the two fields in absolute value (third row) for $Re = 100$ (first column), $Re = 500$ (second column) and $Re = 1000$ (third one) at time $t = 10$. Eleven modes for ω were considered.

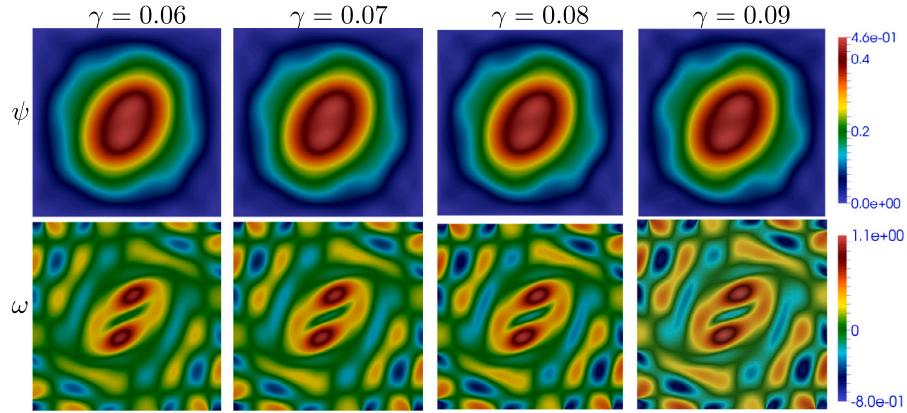


Fig. 22. ROM validation — γ parameterization: stream function ψ (first row) and vorticity ω (second row) computed by the FOM for $Re = 800$ and $\gamma = 0.06$ (first column), $\gamma = 0.07$ (second column), $\gamma = 0.08$ (third column), and $\gamma = 0.09$ (fourth column) at time $t = 10$.

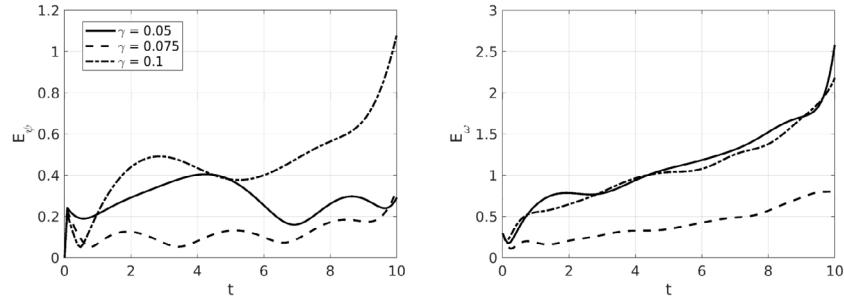


Fig. 23. ROM validation — γ parameterization: time history of error (38) for stream function ψ (left) and vorticity ω (right) for the three different test values $\gamma = 0.05$, $\gamma = 0.075$ and $\gamma = 0.1$ at $Re = 800$.

of a global POD basis space for parametric studies. We assessed our ROM approach with the vortex merger problem, a classical benchmark used for the validation of numerical methods for the stream function-vorticity formulation of the Navier-Stokes equations, and with an extension of the vortex merger problem to a more complex domain representing the North Atlantic Ocean. The numerical results show that our ROM is able to capture the flow features with a good accuracy

both in the reconstruction of the flow field evolution and in a physical parametric setting. In addition, for the simple vortex merger problem we observed that our ROM enables substantial computational time savings.

Next, we are going to extend the ROM approach presented here to the Quasi-Geostrophic Equations (see [20] for the development of the

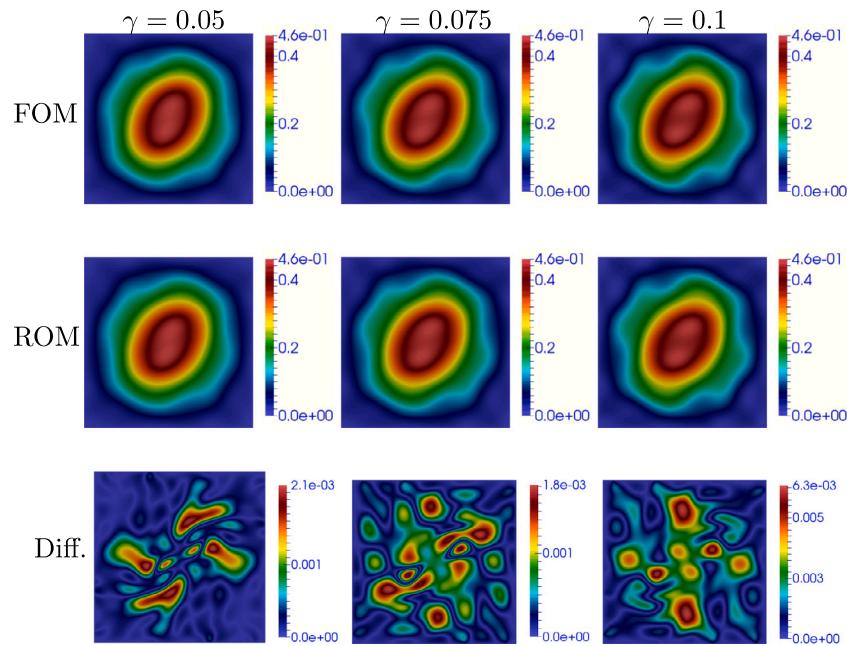


Fig. 24. ROM validation — γ parameterization: stream function ψ computed by the FOM (first row) and the ROM (second row), and difference between the two fields in absolute value (third row) for $\gamma = 0.05$ (first column), $\gamma = 0.075$ (second column) and $\gamma = 0.1$ (third one) at time $t = 10$. Six modes for ψ were considered.

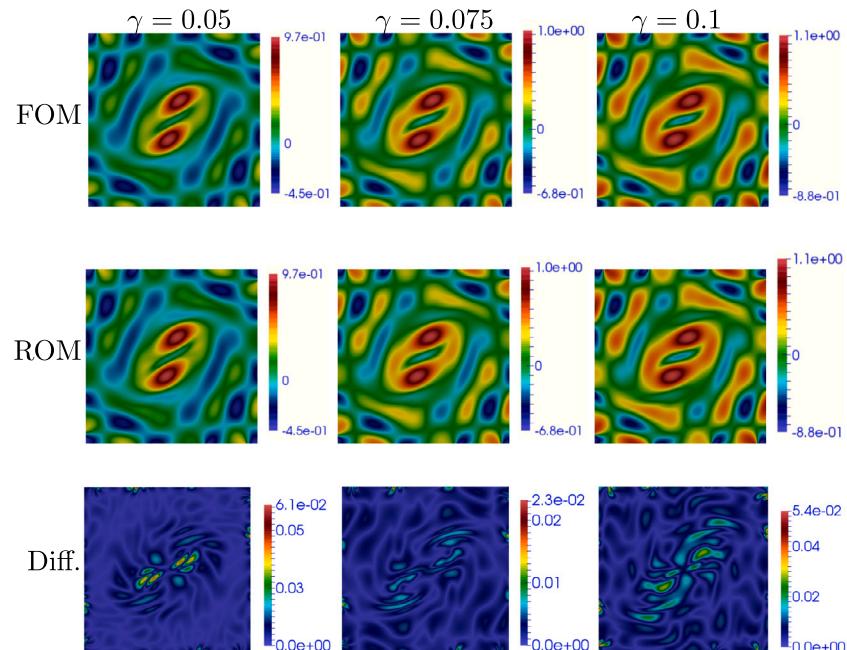


Fig. 25. ROM validation — γ parameterization: vorticity ω computed by the FOM (first row) and the ROM (second row), and difference between the two fields in absolute value (third row) for $\gamma = 0.05$ (first column), $\gamma = 0.075$ (second column) and $\gamma = 0.1$ (third one) at time $t = 10$. Twelve modes for ω were considered.

FOM). In particular, we intend to couple such equations with a differential filter [11,27,43–45] in order to simulate two-dimensional turbulent geophysical flows on under-refined meshes in the spirit of [46,47].

CRediT authorship contribution statement

Michele Girfoglio: First draft, Numerical simulations and tests. **Annalisa Quaini:** First revision, Funding, Test case design. **Gianluigi Rozza:** Final revision, Funding, Reporting.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Gianluigi Rozza reports financial support was provided by European Research Council.

Acknowledgments

We acknowledge the support provided by the European Research Council Executive Agency by the Consolidator Grant project AROMA-CFD "Advanced Reduced Order Methods with Applications in Computational Fluid Dynamics" - GA 681447, H2020-ERC CoG 2015 AROMA-CFD, PI G. Rozza, and INdAM-GNCS 2019–2020 projects. This work was also partially supported by US National Science Foundation through grant DMS-1953535 (PI A. Quaini). A. Quaini acknowledges support from the Radcliffe Institute for Advanced Study at Harvard University where she has been the 2021–2022 William and Flora Hewlett Foundation Fellow.

References

- [1] Behr M, Liou R, Tezduyar T. Vorticity-streamfunction formulation of unsteady incompressible flow past a cylinder: Sensitivity of the computed flow field to the location of the outflow boundary. *Internat J Numer Methods Fluids* 1991;12:323–42.
- [2] Lequeurre J, Munnier A. Vorticity and stream function formulations for the 2D Navier–Stokes equations in a bounded domain. *J Math Fluid Mech* 2020;22.
- [3] Minev P, Vabishchevich P. An operator-splitting scheme for the stream function–vorticity formulation of the unsteady Navier–Stokes equations. *J Comput Appl Math* 2015;293:147–63.
- [4] Sousa E, Sobey I. Effect of boundary vorticity discretization on explicit stream-function vorticity calculations. *Internat J Numer Methods Fluids* 2005;49:371–93.
- [5] Tezduyar T, Liou J, Ganjoo D. Incompressible flow computations based on the vorticity-stream function and velocity-pressure formulations. *Comput Struct* 1990;35:445–72.
- [6] Veroy K, Patera AT. Certified real-time solution of the parametrized steady incompressible Navier–Stokes equations: Rigorous reduced-basis a posteriori error bounds. *Internat J Numer Methods Fluids* 2005;47:773–88.
- [7] Burkardt J, Gunzburger M, Lee H-C. POD And CVT-based reduced-order modeling of Navier–Stokes flows. *Comput Methods Appl Mech Engrg* 2006;196:337–55.
- [8] Ballarin F, Manzoni A, Quarteroni A, Rozza G. Supremizer stabilization of POD–Galerkin approximation of parametrized steady incompressible Navier–Stokes equations. *Internat J Numer Methods Engrg* 2014;102:1136–61.
- [9] Lorenzi S, Cammi A, Luzzi L, Rozza G. POD–Galerkin method for finite volume approximation of Navier–Stokes and RANS equations. *Comput Methods Appl Mech Engrg* 2016;311:151–79.
- [10] Stabile G, Rozza G. Finite volume POD–Galerkin stabilised reduced order methods for the parametrised incompressible Navier–Stokes equations. *Comput Fluid* 2018;173:273–84.
- [11] Girfoglio M, Quaini A, Rozza G. A POD–Galerkin reduced order model for a LES filtering approach. *J Comput Phys* 2021;436:110260.
- [12] Ahmed S, Rahman SM, San O, Rasheed A, Navon I. Memory embedded non-intrusive reduced order modeling of non-ergodic flows. *Phys Fluids* 2019;31:126602.
- [13] Ahmed S, San O, Rasheed A, Iliescu T. A long short-term memory embedding for hybrid uplifted reduced order models. *Physica D* 2020;409:132471.
- [14] Dumon A, Allery C, Ammar A. Proper Generalized Decomposition method for incompressible flows in stream-vorticity formulation. *Eur J Comput Mech* 2010;19:591–617.
- [15] Pawar S, Ahmed S, San O, Rasheed A. Data-driven recovery of hidden physics in reduced order modeling of fluid flows. *Phys Fluids* 2020;32:036602.
- [16] Pawar S, Ahmed S, San O, Rasheed A. An evolve-then-correct reduced order model for hidden fluid dynamics. *Mathematics* 2020;8:570.
- [17] Pawar S, San O, Nair A, Rasheed A, Kvamsdal T. Model fusion with physics-guided machine learning: Projection-based reduced-order modeling. *Phys Fluids* 2021;33:067123.
- [18] San O, Borggaard J. Principal interval decomposition framework for POD reduced-order modeling of convective Boussinesq flows. *Internat J Numer Methods Fluids* 2015;78:37–62.
- [19] Mou Z, Wells D, Xie X, Iliescu T. Reduced order models for the quasi-geostrophic equations: A brief survey. *Fluids* 2020;6:16.
- [20] Girfoglio M, Quaini A, Rozza G. A novel large Eddy simulation model for the quasi-geostrophic equations in a finite volume setting. 2022, <https://arxiv.org/abs/2202.00295>.
- [21] Weller HG, Tabor G, Jasak H, Fureby C. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Comput Phys* 1998;12(6):620–31.
- [22] Issa RI. Solution of the implicitly discretised fluid flow equations by operator-splitting. *J Comput Phys* 1986;62:40–65.
- [23] Moukalled F, Mangani L, Darwish M. The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM and Matlab. 1st ed. Springer Publishing Company, Incorporated; 2015.
- [24] Patankar SV, Spalding DB. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *Int J Heat Mass Transfer* 1972;15:1787–806.
- [25] Temam R. *Navier–Stokes Equations: Theory and Numerical Analysis*. Cambridge University Press; 2001.
- [26] Tezduyar TE, Glowinski R, Liou J. Petrov–Galerkin Methods on multiply connected domains for the vorticity-stream function formulation of the incompressible Navier–Stokes equations. *Internat J Numer Methods Fluids* 1988;8(10):1269–90.
- [27] Girfoglio M, Quaini A, Rozza G. Pressure stabilization strategies for a LES filtering reduced order model. *Fluids* 2021;6:302.
- [28] Jasak H. *Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows*. (Ph.D. thesis), Imperial College, University of London; 1996.
- [29] Benner P, Schilders W, Grivet-Talocia S, Quarteroni A, Rozza G, Silveira LM. *Model order reduction*. Berlin, Boston: De Gruyter; 2020.
- [30] Chinesta F, Huerta A, Rozza G, Willcox K. *Model order reduction*. Encycl Comput Mech 2016.
- [31] Chinesta F, Ladeveze P, Cueto E. A short review on model order reduction based on proper generalized decomposition. *Arch Comput Methods Eng* 2011;18:395.
- [32] Dumon A, Allery C, Ammar A. Proper general decomposition (PGD) for the resolution of Navier–Stokes equations. *J Comput Phys* 2011;230:1387–407.
- [33] Kalashnikova I, Barone MF. On the stability and convergence of a Galerkin reduced order model (ROM) of compressible flow with solid wall and far-field boundary treatment. *Internat J Numer Methods Engrg* 2010;83:1345–75.
- [34] Quarteroni A, Manzoni A, Negri F. *Reduced Basis Methods for Partial Differential Equations*. Springer International Publishing; 2016.
- [35] Rozza G, Huynh DBP, Patera AT. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Arch Comput Methods Eng* 2008;15:229.
- [36] Kunisch K, Volkwein S. Galerkin Proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM J Numer Anal* 2002;40:492–515.
- [37] Star SK, Stabile G, Belloni F, Rozza G, Degroote J. A novel iterative penalty method to enforce boundary conditions in finite volume POD–Galerkin reduced order models for fluid dynamics problems. *Commun Comput Phys* 2021;30:34–66.
- [38] Reinaud J, Dritschel D. The critical merger distance between two co-rotating quasi-geostrophic vortices. *J Fluid Mech* 2005;522:357–81.
- [39] Strazzullo M, Mosetti R. FEniCS application to a finite element quasi-geostrophic model. Linear and non-linear analysis of munk-like solutions and data assimilation implementation through adjoint method, [arXiv:1705.02013](https://arxiv.org/abs/1705.02013).
- [40] Strazzullo M, Ballarin F, Mosetti R, Rozza G. Model reduction for parametrized optimal control problems in environmental marine sciences and engineering. *SIAM J Sci Comput* 2018;40(4):B1055–79.
- [41] Carere G, Strazzullo M, Ballarin F, Rozza G, Stevenson R. A weighted POD-reduction approach for parametrized PDE-constrained optimal control problems with random inputs and applications to environmental sciences. *Comput Math Appl* 2021;102:261–76.
- [42] Foster E, Iliescu T, Wang Z. A finite element discretization of the stream-function formulation of the stationary quasi-geostrophic equations of the ocean. *Comput Methods Appl Mech Engrg* 2013;261:105–17.
- [43] Girfoglio M, Quaini A, Rozza G. Fluid–structure interaction simulations with a LES filtering approach in solids4Foam. *Commun Appl Indus Math* 2021;12:13–28.
- [44] Girfoglio M, Quaini A, Rozza G. A finite volume approximation of the Navier–Stokes equations with nonlinear filtering stabilization. *Comput & Fluids* 2019;187:27–45.
- [45] Girfoglio M, Quaini A, Rozza G. A hybrid reduced order model for nonlinear LES filtering. 2021, <https://arxiv.org/abs/2107.12933>.
- [46] Holm D, Nadiga B. Modeling mesoscale turbulence in the barotropic double-gyre circulation. *J Phys Oceanogr* 2003;33:2355–65.
- [47] Monteiro I, Manica C, Rebholz L. Numerical study of a regularized barotropic vorticity model of geophysical flow. *Numer Methods Partial Differential Equations* 2015;31:1492–514.