



Available online at www.sciencedirect.com

ScienceDirect

Comput. Methods Appl. Mech. Engrg. 354 (2019) 307–330

**Computer methods
in applied
mechanics and
engineering**

www.elsevier.com/locate/cma

A domain decomposition method for the non-intrusive reduced order modelling of fluid flow

D. Xiao^{b,a,c,*}, F. Fang^{a,c}, C.E. Heaney^a, I.M. Navon^d, C.C. Pain^{a,c}

^a Applied Modelling and Computation Group, Department of Earth Science and Engineering, Imperial College London, Prince Consort Road, London, SW7 2BP, UK¹

^b ZCCE, College of Engineering, Swansea University, Bay Campus, Fabian Way, Swansea SA1 8EN, UK

^c Data Assimilation Lab, Data Science Institute, Imperial College London, UK

^d Department of Scientific Computing, Florida State University, Tallahassee, FL, 32306-4120, USA

Received 20 March 2017; received in revised form 23 May 2019; accepted 23 May 2019

Available online 30 May 2019

Highlights

- A domain decomposition non-intrusive reduced order model (DDNIROM) is presented.
- The DDNIROM exhibits good agreement with the high fidelity model (HFM).
- Local hypersurfaces of the DDNIROM are based on radial basis functions.
- These hypersurfaces represent the coupling between the subdomains.
- The DDNIROM can be constructed without modification to the source code of the HFM.

Abstract

In this paper we present a new domain decomposition non-intrusive reduced order model (DDNIROM) for the Navier–Stokes equations. The computational domain is partitioned into subdomains and a set of local basis functions is constructed in each subdomain using Proper Orthogonal Decomposition (POD). A radial basis function (RBF) method is then used to generate a set of hypersurfaces for each subdomain. Each local hypersurface represents, not only the fluid dynamics over the subdomain to which it belongs, but also the interactions with the surrounding subdomains. This implicit coupling between the subdomains provides the global coupling necessary to enforce incompressibility and is a means of providing boundary conditions for each subdomain.

The performance of this DDNIROM is illustrated numerically by three examples: flow past a cylinder, and air flow over 2D and 3D street canyons. The results show that the DDNIROM exhibits good agreement with the high-fidelity full model while the computational cost is reduced by several orders of magnitude. The domain decomposition (DD) method provides the flexibility to choose different numbers of local basis functions for each subdomain depending on the complexity of the flow therein. The fact that the RBF surface representation takes input only from its current subdomain and the surrounding subdomains, means that, crucially, there is a reduction in the dimensionality of the hypersurface when compared with a more traditional, global NIROM. This comes at the cost of having a larger number of hypersurfaces.

© 2019 Elsevier B.V. All rights reserved.

* Corresponding author at: ZCCE, College of Engineering, Swansea University, Bay Campus, Fabian Way, Swansea SA1 8EN, UK.

E-mail address: dunhui.xiao@swansea.ac.uk (D. Xiao).

¹ URL: <http://amcg.ese.imperial.ac.uk>.

Keywords: Domain decomposition; Reduced order modelling; Non-intrusive; Proper orthogonal decomposition

1. Introduction

Reduced order modelling is a powerful tool for real-time analysis as it offers the potential to reduce dimensionality of large complex systems. It has been successfully applied to various fields, for example, fracture modelling [1], fluid dynamics [2–9], air pollution [10], molecular dynamics [11], aerospace design [12], parameter optimisation [13] and haemodynamics [14]. A Reduced Order Model (ROM) can be derived by Proper Orthogonal Decomposition (POD) in combination with Galerkin projection, for example. It can also be derived by model identification methods in conjunction with the POD Galerkin method [15]. ROMs constructed from computational models can be classified as two types according to whether they require modification of the original source code or not; intrusive ROM and non-intrusive ROM (NIROM) respectively. The intrusive ROMs typically require modification of the source code in order to project the discretised system onto the basis functions and form the discretised reduced system. Intrusive ROMs have been reported to suffer from instability, and when applied to non-linear problems, the resulting ROMs can be inefficient [16–20]. To tackle these issues, a number of stabilisation methods have been developed, for example; Petrov–Galerkin methods [21,22], calibration methods [23–25], regularisation methods [26] and Fourier expansion methods [27]. Also, various non-linearity model reduction methods have been proposed such as the empirical interpolation method (EIM) [28], discrete EIM (DEIM) [19], residual DEIM (RDEIM) [29], least squares Petrov–Galerkin projection methods [24], a Gauss–Newton method with approximated tensors (GNAT) [30] and the quadratic expansion method [31].

More recently, non-intrusive reduced order modelling has become popular in various research and engineering fields as its implementation is independent of the original source code. That is, no modifications need be made to the source code in order to construct the non-intrusive reduced order model. All that is required from the full model is its output, *i.e.* the solutions. Non-intrusive reduced order modelling has been demonstrated to be a robust and an efficient approach for the model reduction of general linear and non-linear time-dependent flow dynamical systems, and has been applied to examples in fluid dynamics [32,33], fluid–structure interaction problems [34,35] and multiphase porous media flows [36,37]. Xiao et al. [38] also presented a parameterised NIROM for general time-dependent non-linear partial differential equations. However, POD-based ROMs have difficulty in capturing moving discontinuities, which require larger quantities of data for NIROM to capture the system energy and a larger number of basis functions [39]. This also motivates the development of subdomain ROM technology for complex flows (*e.g.* flows with shocks, turbulent flows) and localised complex physical problems, *e.g.* urban flows with traffic. Using Domain Decomposition (DD) methods allows one to construct local basis functions based on details of local flow solutions over each subdomain. For example, within a subdomain containing a shock there may be many POD basis functions required to capture the physics, whilst in other subdomains with less complex physics, fewer basis functions may be needed.

The domain decomposition method originated in the work of Przemieniecki [40] and has been applied to various fields, such as parallel processing [41,42], shear bands [43], stochastic multi-physics systems [44] and hydrodynamics [45]. Lucia et al. [39] first introduced domain decomposition to ROM for accurately tracking a moving strong shock wave. Baiges et al. [46], Amsallem et al. [47] and Chaturantabut [48] applied domain decomposition to non-linear model reduction. Kerfriden et al. [49] proposed a partitioned ROM strategy for non-linear fracture problems. Pau et al. [50,51] used the POD mapping method to construct ROMs for fine-resolution river basin models. Lucia et al. [52] used domain decomposition methods based ROM to investigate the moving shocks. Antil et al. [53,54] used domain decomposition to form ROMs based on balanced truncation.

This paper presents a new Domain Decomposition NIROM (DDNIROM) for fluid problems using POD and Radial Basis Function (RBF) interpolation methods. This extends our previous work on RBF-based NIROMs [55] by generating local basis functions through the application of domain decomposition. Additionally, the applications in this paper are challenging and based on the multi-dimensional (2D and 3D) Navier–Stokes equations.

Using the RBF interpolation method, a set of single-valued functions is constructed in such a way as to represent the underlying reduced flow dynamics. These functions, herein after known as hypersurfaces, are local to a subdomain.

In DDNIROM, during the offline computational procedure, the solutions to the full model are recorded as a sequence of snapshots over the entire computational domain which is then partitioned into subdomains according to the local flow features. From the local solution, *i.e.* the snapshots over each subdomain, a number of local basis functions are generated using POD. The RBF multi-dimensional interpolation method is then used to construct a set of hypersurfaces representing the local fluid dynamics over this subdomain. When constructing the hypersurfaces for a given subdomain, the solution in the surrounding subdomains is taken into account by introducing their POD coefficients as inputs to the hypersurfaces of this particular subdomain.

During the online computational procedure, for each subdomain, the solution of the DDNIROM at the current time level can be obtained from the reduced solutions at the previous time level for the current subdomain and from the reduced solutions at the current time level for the surrounding subdomains leading to an implicit method. The performance of the new DDNIROM is assessed by applying it to three fluid dynamics test cases: flow past a cylinder, and 2D and 3D flows in street canyons. Comparisons between the high-fidelity full model and the proposed DDNIROM are presented, and in some cases results from NIROM are also shown.

The structure of the present paper is as follows. Section 2 presents the governing equations of fluid problems. Section 3 describes the general theory of the non-intrusive reduced order modelling. Section 4 derives the DDNIROM. Section 5 demonstrates the performance of the DDNIROM for three test cases: flow past a cylinder, and flow in 2D and 3D street canyons. Finally, in Section 6, the summary and conclusions are presented.

2. Governing equations

This work uses the Navier–Stokes equations to describe the fluid dynamics, namely,

$$\nabla \cdot \mathbf{u} = 0, \quad (1)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \nu \nabla^2 \mathbf{u}, \quad (2)$$

where $\mathbf{u} \equiv (u, v, w)^T$ is the velocity vector, p is the pressure and ν is the viscosity.

3. Non-intrusive model reduction method

In POD-based reduced order modelling methods, any variable Ψ (for example, velocity or pressure) can be expressed:

$$\Psi = \sum_{i=1}^m \alpha_i \Phi_i, \quad (3)$$

in which m is the number of POD basis functions, α_i represents the i th POD coefficient (from a total of m POD coefficients ($\alpha_1, \alpha_2, \dots, \alpha_m$)) and Φ_i represents the i th POD basis function. The ensemble average of the variable Ψ is sometimes included on the right-hand side of Eq. (3). We omit this here as we have found its inclusion may lead to inaccuracies [35].

The POD basis functions can be obtained by applying Singular Value Decomposition (SVD) to numerical solutions of the governing equations taken at certain time levels (*i.e.* snapshots), see [55]. If N_s snapshots are produced, the SVD will produce N_s singular values and a corresponding N_s basis functions. In this paper, a solution snapshot was generated at each time level, so the number of snapshots is equal to the number of time levels, N_t . Not all POD basis functions need be used. The square of each singular value represents the amount of information captured by the corresponding basis function, so if the singular value is low its corresponding basis function can be discarded. Based on this, a common criterion for choosing how many POD basis functions to include is as follows: for a tolerance $\eta \lesssim 1$, find the smallest integer m such that

$$\frac{\sum_{i=1}^m \sigma_i^2}{\sum_{i=1}^{N_s} \sigma_i^2} \geq \eta, \quad (4)$$

where $m \leq N_s$.

In non-intrusive reduced order modelling, a set of hypersurfaces is used to represent the dynamics of the reduced system. We construct the hypersurfaces with a radial basis function (RBF) interpolation method, which is a good method for interpolating scattered data in a high dimensional space. There are various types of RBFs such as:

multi-quadratic, plate spline, Gaussian and inverse multi-quadratic. In this paper we use the Gaussian RBF, which has the general form $\phi(r) = e^{-(r/\sigma)^2}$ (σ being the shape parameter and r being the radius). The choice of shape parameter value has an impact on the accuracy of the interpolation function. As discussed in [56], a conflict arises between the accuracy and stability of the approximation both of which depend on the parameter value. Generally, the smaller the value of the shape parameter, the more accurate the approximation, however, the smaller the value of the shape parameter, the more likely the method is to be unstable. There are several automated approaches to find an optimal value for the shape parameter, such as Leave One Out Cross Validation (LOOCV), Generalised Cross Validation (GCV) and Maximum Likelihood Estimator (MLE). References discussing these methods can be found in the work of [56]. In this paper, we base the value on the maximum range of the values of POD coefficients. The exact form of the shape function is given in the following paragraph in Eq. (8).

As explained in [55], the POD coefficient α_i^n associated with the i th POD basis function at any time level n can be obtained from the hypersurface, f_i , and the POD coefficients at the previous time level:

$$\alpha_i^n = f_i(\boldsymbol{\alpha}^{n-1}) = f_i(\alpha_1^{n-1}, \alpha_2^{n-1}, \dots, \alpha_m^{n-1}), \quad n \in \{1, 2, \dots, \mathcal{M}\}, \quad (5)$$

where $\boldsymbol{\alpha}^{n-1}$ denotes the complete set of POD coefficients at time level $n - 1$, f_i is the RBF interpolation function or hypersurface for the i th POD coefficient and \mathcal{M} is the number of time levels. We split our NIROM into two stages: an offline stage, during which the NIROM is constructed, and an online stage, during which the NIROM is solved. Both stages rely upon Eq. (5), and for the offline stage, \mathcal{M} will equal the number of snapshots and for the online stage, \mathcal{M} will equal the desired number of time levels for the NIROM. For the online stage, an initial condition will also be required, and can be calculated by pre-multiplying the high-fidelity full model's initial conditions by the transpose of the basis functions. In order to construct the NIROM, the POD coefficients of each snapshot are found and used to provide data which will determine the hypersurfaces f_i . As previously stated, each hypersurface f_i is approximated by a linear combination of a number of RBFs [55]:

$$f_i(\boldsymbol{\alpha}^{n-1}) = \sum_{j=0}^{N_s-1} w_{i,j} \phi(\|\boldsymbol{\alpha}^{n-1} - \mathbf{c}^j\|_2), \quad n \in \{1, 2, \dots, N_s\} \quad (6)$$

where

- $w_{i,j}$ denotes the weight of the i th POD coefficient associated with the RBF at the j th centre;
- ϕ is the RBF (shown here evaluated at time level $n - 1$ and at the j th centre)

$$\phi(\|\boldsymbol{\alpha}^{n-1} - \mathbf{c}^j\|_2) = \exp\left[-\left(\frac{\|\boldsymbol{\alpha}^{n-1} - \mathbf{c}^j\|_2}{\sigma}\right)^2\right] \quad (7)$$

where the shape parameter is defined as

$$\sigma = \frac{\alpha^{\max} - \alpha^{\min}}{N_s} \quad (8)$$

in which

$$\alpha^{\max} = \max_{\forall i,n} \alpha_i^n \quad \text{and} \quad \alpha^{\min} = \min_{\forall i,n} \alpha_i^n \quad (9)$$

for $n \in \{0, 1, \dots, N_s - 1\}$ and $i \in \{1, 2, \dots, m\}$;

- \mathbf{c}^j is a vector containing the coordinates of the j th centre. In this work, the centres are chosen such that $\mathbf{c}^n = \boldsymbol{\alpha}^n$ for $n \in \{0, 1, \dots, N_s - 1\}$;
- $\|\cdot\|_2$ denotes the ℓ_2 vector norm (or Euclidean norm).

Combining Eqs. (5) and (6), and evaluating them at the centres leads to a system of N_s equations and N_s unknowns for the i th POD coefficient which can be written as

$$A\mathbf{w}_i = \mathbf{y}_i, \quad (10)$$

where

$$A = \begin{bmatrix} \phi(\|\alpha^0 - \mathbf{c}^0\|_2) & \phi(\|\alpha^0 - \mathbf{c}^1\|_2) & \cdots & \phi(\|\alpha^0 - \mathbf{c}^{N_s-1}\|_2) \\ \phi(\|\alpha^1 - \mathbf{c}^0\|_2) & \phi(\|\alpha^1 - \mathbf{c}^1\|_2) & \cdots & \phi(\|\alpha^1 - \mathbf{c}^{N_s-1}\|_2) \\ \vdots & \vdots & & \vdots \\ \phi(\|\alpha^{N_s-1} - \mathbf{c}^0\|_2) & \phi(\|\alpha^{N_s-1} - \mathbf{c}^1\|_2) & \cdots & \phi(\|\alpha^{N_s-1} - \mathbf{c}^{N_s-1}\|_2) \end{bmatrix} \quad (11)$$

$$\mathbf{y}_i = (\alpha_i^1, \alpha_i^2, \dots, \alpha_i^{N_s})^T, \quad (12)$$

$$\mathbf{w}_i = (w_{i,0}, w_{i,1}, \dots, w_{i,j}, \dots, w_{i,N_s-1})^T. \quad (13)$$

The hypersurface f_i is obtained by solving equation (10) for the weights. Once a hypersurface for each POD coefficient has been calculated, the offline stage is complete. The online stage consists of solving equations (5) and (6) given an initial condition. The NIROM solution at the current time level is calculated from the known hypersurfaces evaluated at the NIROM solutions at the previous time level. In the online stage the centres are the same as in the offline stage, *i.e.* the POD coefficients of the snapshots.

4. Domain decomposition strategy for non-intrusive reduced order modelling

This section describes how domain decomposition can be applied to NIROM. In the DDNIROM approach, the computational domain Ω is divided into S subdomains, $\Omega^d, d \in \{1, 2, \dots, S\}$. Each subdomain has local unknowns $\Psi^d \in \mathbb{R}^{M^d}$, where M^d is the number of nodes (in the finite element mesh) in subdomain d and $M = M^1 + \dots + M^d + \dots + M^S$ is the total number of nodes. The subdomains are constructed in such a way that each node belongs to just one subdomain.

In this method, the solutions at nodes within subdomain Ω^d are used to form a set of local POD basis functions Φ^d . The local POD basis functions are calculated by performing a SVD on the nodal values of the snapshots from only this local subdomain.

For each subdomain Ω^d , we construct a set of hypersurfaces to represent the underlying dynamical system associated with this subdomain and the surrounding subdomains over the reduced space. By modifying equation (5), each hypersurface maps between POD coefficients as follows:

$$\alpha_i^{d,n} = f_i^d(\alpha^{d,n-1}, \{\alpha^{sd,n}\}_{sd \in \mathcal{D}_d}), \quad (14)$$

where vector $\alpha^{d,n-1}$ denotes the complete set of POD coefficients (for example, velocity α_u , pressure α_p) at time level $n-1$ for subdomain Ω^d , and \mathcal{D}_d denotes the set of neighbours of subdomain d . POD coefficients from neighbouring subdomains $\{\alpha^{sd,n}\}_{sd \in \mathcal{D}_d}$ are included in the hypersurfaces in order to provide information from these subdomains. This can be thought of as a means of enforcing boundary conditions. In 2D, the number of surrounding subdomains associated with a subdomain Ω^d is either 2, 3 or 4, depending on whether subdomain Ω^d is in the corner of the domain, on an edge of the domain or neither of these. This is illustrated in Fig. 1, which shows the four surrounding subdomains associated with a particular subdomain.

Here, we consider only the vertical and horizontal neighbouring subdomains (North, South, West and East). Although the POD coefficients of diagonally neighbouring subdomains (North-West, North-East, South-West and South-East) are not considered explicitly, their influence is felt indirectly. For example, referring to the subdomain numbering given in Fig. 1, subdomain 8 has as its neighbouring subdomains 3, 7, 9 and 13, and subdomains 2, 4, 12 and 14 are not considered to be its neighbours. However, subdomain 12, for instance, is a neighbouring subdomain of subdomains 7 and 13, which are themselves neighbours of subdomain 8. Therefore subdomain 12 affects subdomain 8 indirectly, through subdomains 7 and 13. In more complex realistic cases, all surrounding subdomains could be considered, as is done in the Lattice Boltzmann method, for example.

Once we have the snapshots and the decomposed domain, NIROM is used to construct a set of hypersurfaces for each POD coefficient in each subdomain as previously described. Having the hypersurfaces means that the reduced model can now be run, in place of the discretised governing equations, *i.e.* the full model. These procedures are summarised in algorithms 1 (offline) and 2 (online). Algorithm 1 describes the offline computational procedure of the construction of a set of hypersurfaces for each subdomain. In order to do this a set of snapshots is generated

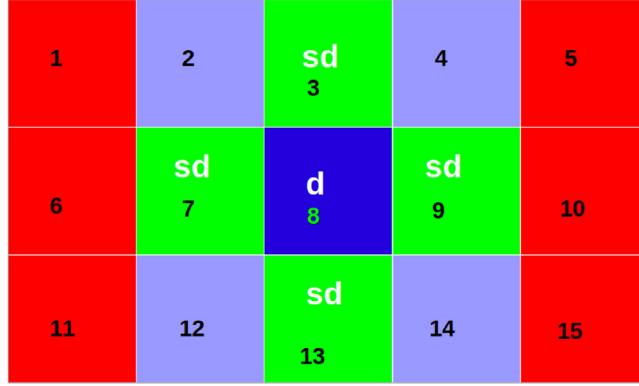


Fig. 1. The figure shows a subdomain d with its four neighbouring subdomains (labelled sd).

from solutions of the full order model; the POD basis functions are found from applying SVD to the snapshots; data on which to base the hypersurfaces is created by projecting the snapshot solutions onto the basis functions; a system relating the weights in the RBF interpolants to this data is solved which fully defines the hypersurfaces. In Eqs. (20), (21) and (22), A^d , $\mathbf{c}^{d,j}$ and $\mathbf{c}^{sd,j}$ have the same structure as A and \mathbf{c}^j in Eq. (11). The difference is that A^d , $\mathbf{c}^{d,j}$ and $\mathbf{c}^{sd,j}$ are restricted to nodal values taken from subdomain d and its neighbours, whereas A and \mathbf{c}^j are constructed using nodal values taken from the entire computational domain.

In Algorithm 1, two fields are considered: velocity and pressure. This algorithm could also be applied to the case where just one field is considered, for example, velocity, or to the case where more than two fields are considered, for example velocity, pressure and density. In the latter case, $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_{\mathbf{u}}^T, \boldsymbol{\alpha}_{\mathbf{p}}^T, \boldsymbol{\alpha}_{\rho}^T)^T$, where $\boldsymbol{\alpha}_{\rho}$ represents the POD coefficients of the density.

Algorithm 2 details the online computation of DDNIROM, in which the interactions between a subdomain and its surrounding subdomains are taken into account. At every time level, information about the incompressibility must be communicated to all the subdomains, therefore we chose an implicit coupling resulting in a recurrent neural network. In this algorithm, the inputs of the function f_i^d include POD coefficients from the previous time level $\boldsymbol{\alpha}^{d,n-1}$ (known) and the current time level from surrounding subdomains $\{\boldsymbol{\alpha}^{sd,n}\}_{sd \in \mathcal{D}_d}$. The latter might be unknown, in which case, an iteration method is introduced to obtain $\boldsymbol{\alpha}^{d,n}$ at current time level n . The iteration loop (**for** $it = 1$ to $N_{iteration}$) ensures that the hypersurface incorporates the flow dynamics over the subdomain d , but also, includes the flow interaction between the subdomain and its neighbouring subdomains.

When we solve the DDNIROM for each subdomain at time level n , at iteration $it > 1$, we need the solutions over the subdomains, $\boldsymbol{\alpha}^{d,n-1}$ and $\{\boldsymbol{\alpha}^{sd,n}\}_{sd \in \mathcal{D}_d}$. After completing the subdomain loop (**for** $d = 1$ to S), the solutions at the current time level for each subdomain are updated. For iteration $it = 1$ however, the current solutions from neighbouring subdomains, $\{\boldsymbol{\alpha}^{sd,n}\}_{sd \in \mathcal{D}_d}$, might not be available. When this is the case we take solutions from the previous time level. For example, consider $it = 1$, $d = 1$ in algorithm 2. The POD coefficients $\boldsymbol{\alpha}^{d,n}$ are calculated solely from POD coefficients from time level $n - 1$:

$$\alpha_i^{d,n} = f_i^d(\boldsymbol{\alpha}^{d,n-1}, \{\boldsymbol{\alpha}^{sd,n}\}_{sd \in \mathcal{D}_d}) \quad \forall i \quad (15)$$

$$\alpha_i^{d=1,n} = f_i^{d=1}(\boldsymbol{\alpha}^{d=1,n-1}, \boldsymbol{\alpha}^{sd=2,n}, \boldsymbol{\alpha}^{sd=6,n}) \quad \forall i \quad (16)$$

$$\approx f_i^{d=1}(\boldsymbol{\alpha}^{d=1,n-1}, \boldsymbol{\alpha}^{sd=2,n-1}, \boldsymbol{\alpha}^{sd=6,n-1}) \quad \forall i, \quad (17)$$

as $\boldsymbol{\alpha}^{sd=2,n}$ and $\boldsymbol{\alpha}^{sd=6,n}$ are not yet known. For the second subdomain, $d = 2$, solutions are available at time level n from neighbouring subdomain 1, but not from subdomain neighbours 3 or 7, hence $\alpha_i^{d=2}$ is calculated as follows

$$\alpha_i^{d=2,n} = f_i^{d=2}(\boldsymbol{\alpha}^{d=2,n-1}, \boldsymbol{\alpha}^{sd=1,n}, \boldsymbol{\alpha}^{sd=3,n}, \boldsymbol{\alpha}^{sd=7,n}) \quad \forall i \quad (18)$$

$$\approx f_i^{d=2}(\boldsymbol{\alpha}^{d=2,n-1}, \boldsymbol{\alpha}^{sd=1,n}, \boldsymbol{\alpha}^{sd=3,n-1}, \boldsymbol{\alpha}^{sd=7,n-1}) \quad \forall i, \quad (19)$$

as $\alpha^{sd=3,n}$ and $\alpha^{sd=7,n}$ are not yet known. Once all subdomains have been looped over, the solutions in every subdomain will be known and available for the second iteration. In this work, the iteration number, $N_{iteration}$, is set to be 4 as we have found that this ensures that the algorithm converges within a time step.

Algorithm 1: Offline: constructing a set of hypersurfaces for DDNIROM

```

(1) Generate  $N_s$  snapshots over the time period  $[0, T]$  by solving the governing equations
(2) Divide the computational domain  $\Omega$  into  $S$  subdomains
(3) Generate POD basis functions and coefficients
for  $d = 1$  to  $S$  do
  (a) Generate a number of POD basis functions  $\Phi_u^d$  and  $\Phi_p^d$  by performing an SVD on the snapshots matrix of subdomain  $d$ ;
  (b) Calculate a set of POD coefficients  $(\alpha_1^d, \alpha_2^d, \dots, \alpha_{m^d}^d)$  for each subdomain by projecting the snapshots into a reduced space;
(4) Obtain a set of hypersurfaces
for  $d = 1$  to  $S$  do
  for  $i = 1$  to  $m^d$  do
    (i) Set the values  $\mathbf{y}_i^d = \{\alpha_i^{d,1}, \alpha_i^{d,2}, \dots, \alpha_i^{d,N_s}\}$  for the  $i$ th POD coefficient
    (ii) Calculate the weights  $\mathbf{w}_i^d$  by solving:
      
$$A^d \mathbf{w}_i^d = \mathbf{y}_i^d, \quad (20)$$

    (iii) Obtain the hypersurface  $f_{u,i}^d$ , for subdomain  $d$  and POD coefficient  $i$ , by substituting the weights obtained in the above step into Eqs. (21) and (22);
      
$$f_{u,i}^d(\alpha_u^{d,n-1}, \alpha_p^{d,n-1}, \{\alpha_u^{sd,n}, \alpha_p^{sd,n}\}_{sd \in \mathcal{D}_d}) = \sum_{j=0}^{N_s-1} w_{i,j}^d \phi \left( \left\| \tilde{\alpha}^{d,n} - \tilde{\mathbf{c}}^{d,j} \right\|_2 \right), \quad (21)$$

      
$$f_{p,i}^d(\alpha_u^{d,n-1}, \alpha_p^{d,n-1}, \{\alpha_u^{sd,n}, \alpha_p^{sd,n}\}_{sd \in \mathcal{D}_d}) = \sum_{j=0}^{N_s-1} w_{i,j}^d \phi \left( \left\| \tilde{\alpha}^{d,n} - \tilde{\mathbf{c}}^{d,j} \right\|_2 \right) \quad (22)$$

      where  $(\tilde{\alpha}^{d,n})^T = ((\alpha^{d,n-1})^T, \{\alpha^{sd,n}\}_{sd \in \mathcal{D}_d})^T$  and  $(\tilde{\mathbf{c}}^{d,j})^T = ((\mathbf{c}^{d,j})^T, \{(\mathbf{c}^{sd,j})^T\}_{sd \in \mathcal{D}_d})^T$ 
    endfor
  endfor

```

5. Numerical examples

In this section, we demonstrate the capability of DDNIROM on three test problems, namely, 2D flow past a cylinder, and 2D and 3D flows in a street canyon. The first problem has flow structures that span most of the domain, whereas the street canyon problems feature localised flow dynamics and it is in such problems that we expect DDNIROM to perform well. The full model solutions were obtained from the finite element code Fluidity [57]. All the simulations were conducted using a shared mesh (for pressure and velocity) of linear triangular or tetrahedral elements and a projection-based solution method which solves for pressure and velocity separately with a second order Crank–Nicolson time-stepping scheme. In this paper, in order to demonstrate our approach, we decompose the domain naïvely as can be seen in Figs. 2, 12 and 20. As future work we plan to develop a method that decomposes the domain more intelligently.

Algorithm 2: Online DDNIROM calculation for the fluid problem

for $n = 1$ to N_t **do**

!! Calculate POD coefficients at the current time step:

for $it = 1$ to $N_{iteration}$ **do**
for $d = 1$ to S **do**
if $it == 1$, then: Initialise POD coefficients for subdomain d : $(\alpha_{u,i}^{d,0}, \alpha_{p,i}^{d,0})$
for $i = 1$ to m^d **do**

(i) Obtain the POD coefficients for associated surrounding subdomains. If there are no solutions for any of the surrounding subdomains, then ignore these particular subdomains at this iteration.

(ii) Evaluate the hypersurface f_i^d by using the complete set of POD coefficients $\alpha_{u,i}^{d,n-1}, \alpha_{p,i}^{d,n-1}$ and, if available, $\{\alpha_{u,i}^{sd,n}, \alpha_{p,i}^{sd,n}\}_{sd \in \mathcal{D}_d}$ which gives the POD coefficients α_u^n and α_p^n at the current time level n :

$$\alpha_u^n = f_{u,i}^d \left(\alpha_{u,i}^{d,n-1}, \alpha_{p,i}^{d,n-1}, \{\alpha_{u,i}^{sd,n}, \alpha_{p,i}^{sd,n}\}_{sd \in \mathcal{D}_d} \right), \quad (23)$$

$$\alpha_p^n = f_{p,i}^d \left(\alpha_{u,i}^{d,n-1}, \alpha_{p,i}^{d,n-1}, \{\alpha_{u,i}^{sd,n}, \alpha_{p,i}^{sd,n}\}_{sd \in \mathcal{D}_d} \right) \quad (24)$$

!! Calculate solutions at the current time step:

for $d = 1$ to S **do**

Calculate the solutions $\mathbf{u}^{d,n}$ and $p^{d,n}$ for each subdomain d on the full space for each time step n by projecting $\alpha_{u,j}^{d,n}$ and $\alpha_{p,j}^{d,n}$ onto the full space.

$$\mathbf{u}^{d,n} = \bar{\mathbf{u}}^d + \sum_{j=1}^{m^d} \alpha_{u,j}^{d,n} \Phi_{u,j}^d, \quad \mathbf{p}^{d,n} = \bar{\mathbf{p}}^d + \sum_{j=1}^{m^d} \alpha_{p,j}^{d,n} \Phi_{p,j}^d \quad (25)$$

5.1. Case 1: flow past a cylinder

The DDNIROM is first applied to flow past a cylinder for a Reynolds number of 1000 ($Re = \frac{UL}{v}$ in which U , L and v are the representative velocity, length scale and viscosity). The cylinder has a diameter (L) of 0.1 units in a computational domain of size $[0, 2.2] \times [0, 0.41]$, see Fig. 2(a). The centre of the cylinder is located at $(0.25, 0.25)$ and an inlet velocity of $U = 0.5$ is applied to the left of the domain. The viscosity v is therefore 0.5×10^{-4} . Zero normal flow and no slip boundary conditions are applied to the cylinder and both lateral sides. The outflow is an open boundary with a natural boundary condition which effectively sets the pressure level to zero at the outlet, see [58]. There are 3213 nodes in the computational domain and 6142 triangular elements. In this example, the global domain is divided into 15 subdomains, see Fig. 2(a). The full model is run over a time period $[0, 6]$ with a time-step size of $\Delta t = 0.01$ and with an initial condition of zero. The inlet boundary velocity is impulsively accelerated from zero to $U = 0.5$ in one time step. Sixty snapshots (solutions) are recorded at regularly-spaced time intervals of 0.1.

Fig. 3 shows the global and local singular values associated with the global domain and the 15 subdomains respectively. The logarithm (base 10) of the singular values is also shown. The local basis function associated with a larger singular value captures more energy (or information) of the flow dynamics of the full model equation, see Eq. (4). It can be seen in Fig. 3 that the magnitude of the local singular values decreases faster than that of the global singular values. Thus, in comparison to the global NIROM, a smaller number of local basis functions is required to capture a given amount of the energy in the original flow system. Consequently this reduces the

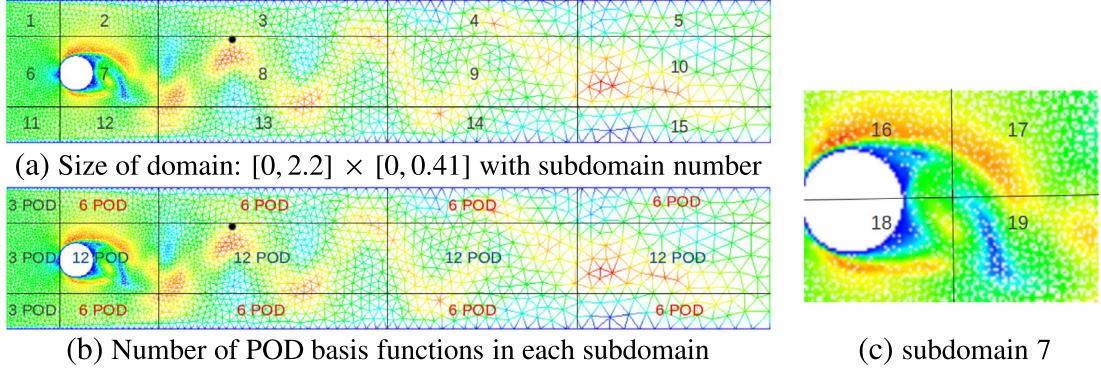


Fig. 2. Case 1: flow past a cylinder. The plots show (a) the computational domain, (b) number of basis functions used in each subdomain, and (c) the subdivision of subdomain 7.

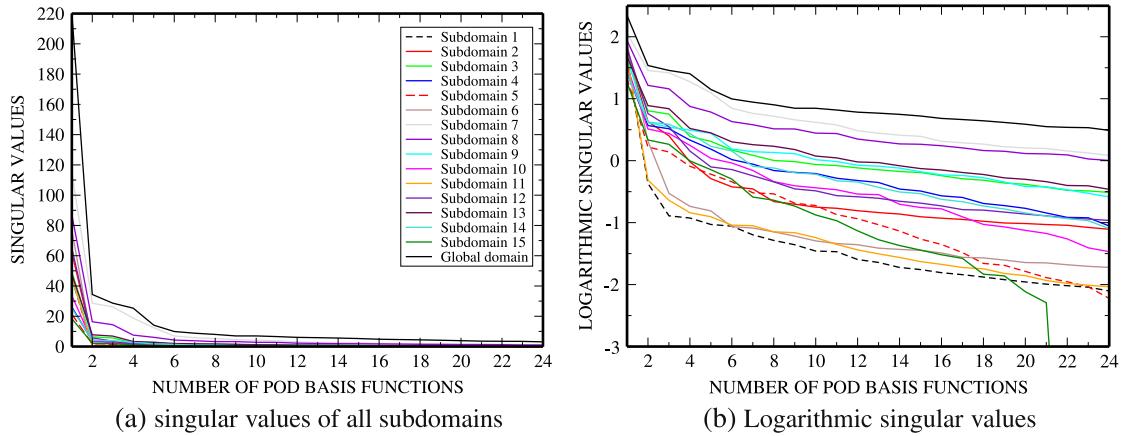


Fig. 3. Case 1: flow past a cylinder. The global and local singular values associated with the global domain and 15 subdomains respectively.

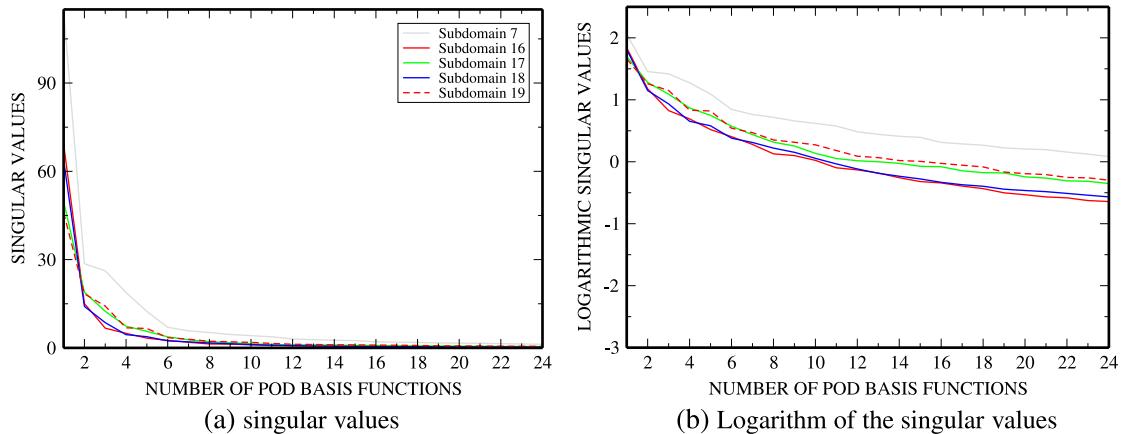


Fig. 4. Case 1: flow past a cylinder. The singular values over subdomain 7, 16, 17, 18 and 19. (Subdomain 7 was split into four subdomains).

dimension of the hypersurfaces required in the NIROM. To improve the accuracy achieved by the reduced order model, subdomain 7 is divided into 4 subdomains, labelled 16, 17, 18 and 19, see Fig. 2(c). Fig. 4 shows the singular values of subdomains 7, 16, 17, 18 and 19. This shows that the singular values associated with the further

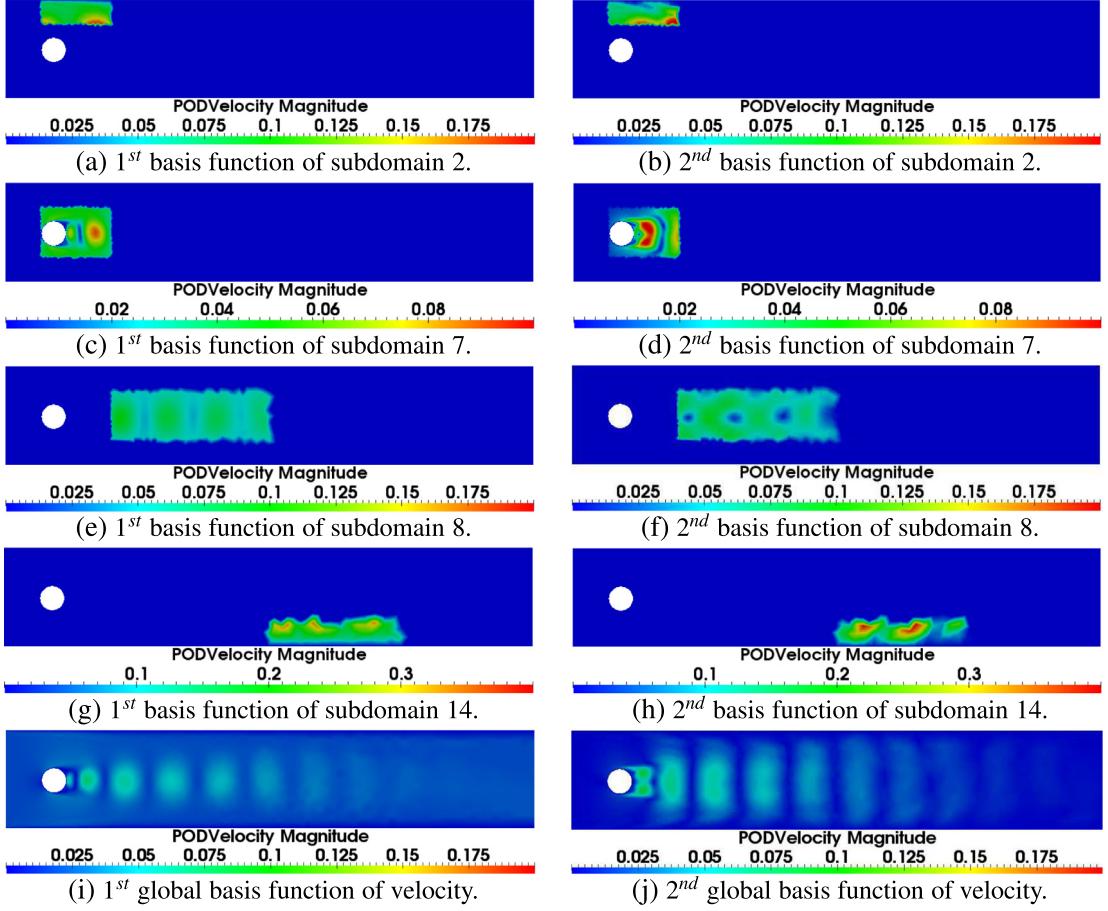


Fig. 5. Case 1: flow past a cylinder. The first and second basis functions for velocity of subdomains 2, 7, 8, 14 (DDNIROM) and of the global domain (NIROM). The basis functions are plotted using Eq. (26).

subdivision into subdomains have a more rapid reduction than subdomain 7, so fewer of them are required for a given accuracy.

Figs. 5 and 6 show the first four basis functions associated with the global domain and with subdomains 2, 7, 8 and 14. To visualise the velocity basis functions we plot the following

$$\widehat{\Phi}_{i,k} = \sqrt{(\Phi_{i,k}^u)^2 + (\Phi_{i,k}^v)^2 + (\Phi_{i,k}^w)^2} \quad \forall k \in \{1, 2, \dots, M\} \quad (26)$$

where $\Phi_{i,k}^u$ is the k th nodal value of the i th POD basis function corresponding to the x -component of velocity, $\Phi_{i,k}^v$ corresponds to the y -component and $\Phi_{i,k}^w$ corresponds to the z -component. In 2D problems the final term is omitted. For NIROM, M is the number of nodes in the finite element mesh. For DDNIROM, M^d should replace M in Eq. (26) as M^d is the number of nodes in subdomain d . The structure of the POD basis functions shown in Figs. 5 and 6 is designed to represent the vortex shedding past a cylinder. The first basis function associated with the subdomains seems more uniform than the first basis function of the global POD approach possibly because the latter has to capture more variation in the solution. However, more structure emerges in the subdomain basis functions associated with the higher order POD modes (basis functions) so they are able to represent the more subtle dynamics of the system.

Fig. 7 shows the magnitude of the velocity solutions obtained from the full model and DDNIROM at time level 5.0. The numbers of local basis functions used within each subdomain are given in Fig. 2(b). These numbers were based loosely on the criterion given in Eq. (4). As shown in Fig. 7, the solutions from the DDNIROM are close to those from the full model. The x -component of the velocity at a particular point $(x, y) = (0.718, 0.287)$ located

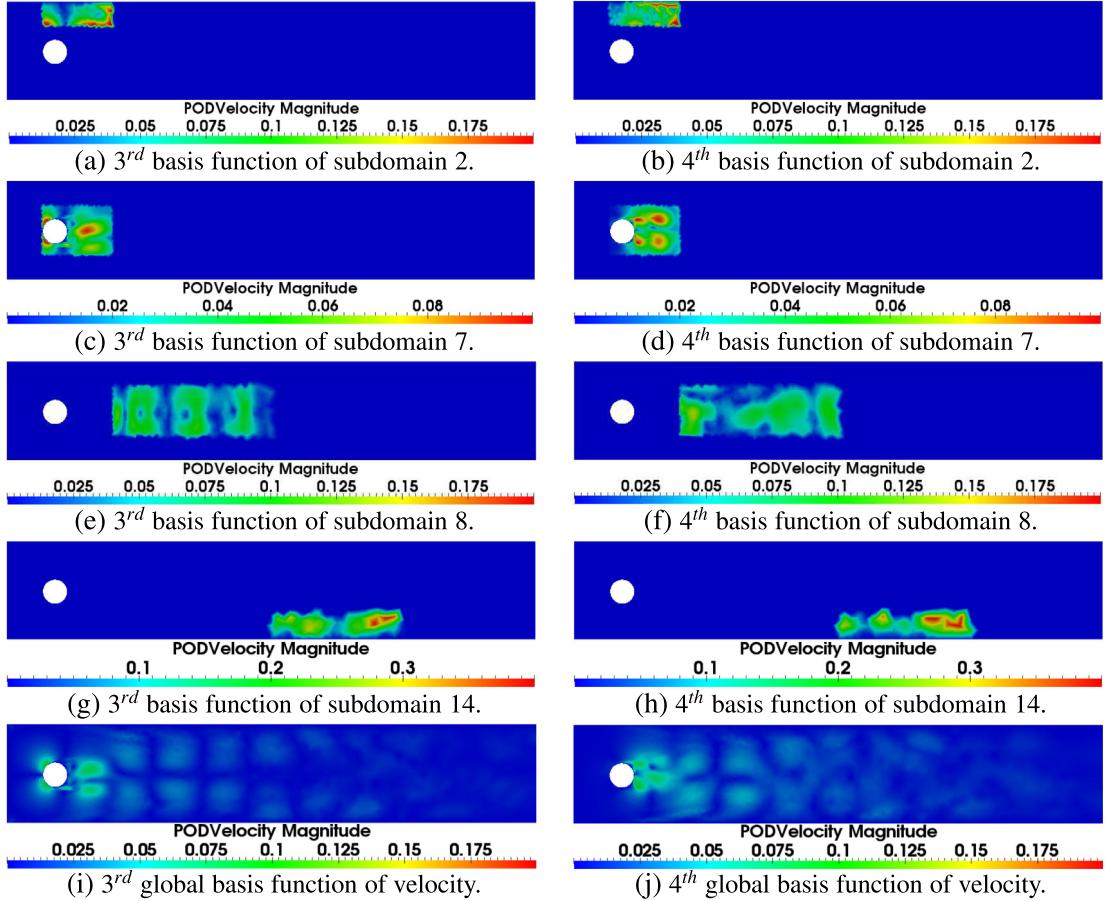


Fig. 6. Case 1: flow past a cylinder. The third and fourth basis functions for velocity of subdomains 2, 7, 8, 14 (DDNIROM) and of the global domain (NIROM). The basis functions are plotted using Eq. (26).

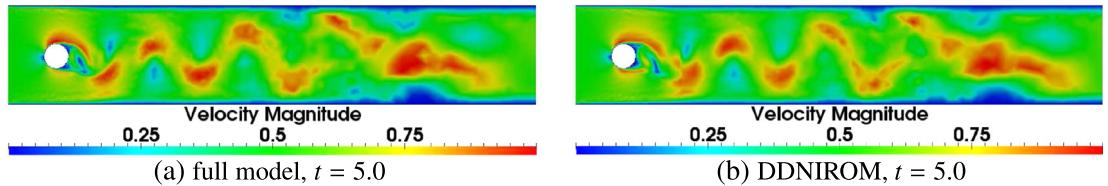


Fig. 7. Case 1: flow past a cylinder. The magnitude of the velocity solutions obtained from the full model and from DDNIROM at time level 5.0. Note that the numbers of local basis functions for each subdomain are labelled in Fig. 2(b).

in subdomain 8 (see Fig. 2(b)), is shown in Fig. 8. We compare the full model against a NIROM (without the domain decomposition) and a DDNIROM (with the domain decomposition). As one can see from this figure, the DDNIROM with a maximum of 12 local basis functions in each subdomain (see Fig. 2(b) for the number of basis functions in each subdomain) is more accurate than the NIROM with 18 global basis functions. There will be a difference in the CPU cost of NIROM and DDNIROM in general, although for problems with so few degrees of freedom, the difference is very small.

In order to test the predictive capability of the DDNIROM, the simulation was run to a time level of 7, beyond the training period [0, 6] of the DDNIROM. Fig. 9(a)–(c) show the velocity solutions obtained from the full model, NIROM with 12 global basis functions and DDNIROM with 6 local basis functions in each subdomain, at time level $t = 6.5$ (where the DDNIROM is predicting). Fig. 9(d) shows the x -component of velocity obtained from

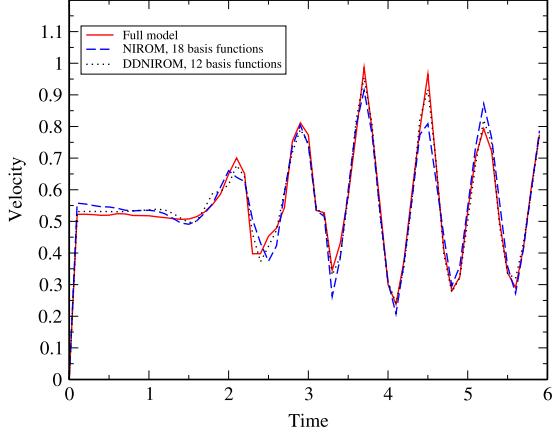


Fig. 8. Case 1: flow past a cylinder. The time series of the x -component of velocity solutions obtained from the full model, global NIROM with 18 basis functions and DDNIROM (with 15 subdomains, the number of POD functions used within each subdomain is shown in Fig. 2(b)) at a particular point $(x, y) = (0.718, 0.287)$, which is located in subdomain 8 in Fig. 2(b).

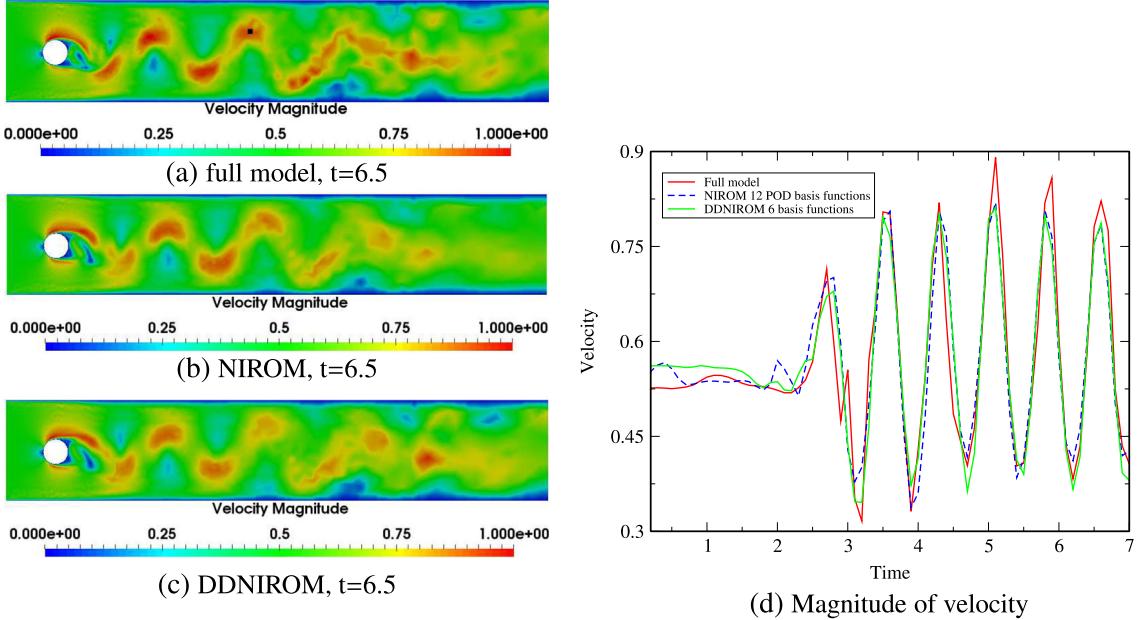


Fig. 9. Case 1: flow past a cylinder. The graphs (a)–(c) show the velocity solutions obtained from the full model, NIROM with 12 global basis functions and DDNIROM with various numbers local basis functions (given in Fig. 2b) at predicted time level $t = 6.5$. Graph (d) shows the x -component of velocity obtained from the full model, NIROM with 12 basis functions and DDNIROM with 6 local basis functions at a particular point $(x, y) = (0.989, 0.283)$, the location of this point is shown in plot (a) as a black point.

the full model, NIROM with 12 basis functions and DDNIROM with 6 local basis functions at a particular point $(x, y) = (0.989, 0.283)$, the location of this point is shown in Fig. 9(a) as a black point. Fig. 10 plots the same quantities as Fig. 9, but at time level $t = 7$, and the time series is taken at a point nearer the cylinder. It can be seen that the DDNIROM predicts well at time levels $t = 6.5$ and $t = 7$. Furthermore, the accuracy of the DDNIROM is similar to the global NIROM despite the small number of DDNIROM basis functions. DDNIROM is not ideally suited to this application (vortex shedding past a cylinder) as the NIROM basis functions decrease rapidly anyway. It may be better suited to ‘hard’ problems in which the singular values, associated with global NIROM, do not

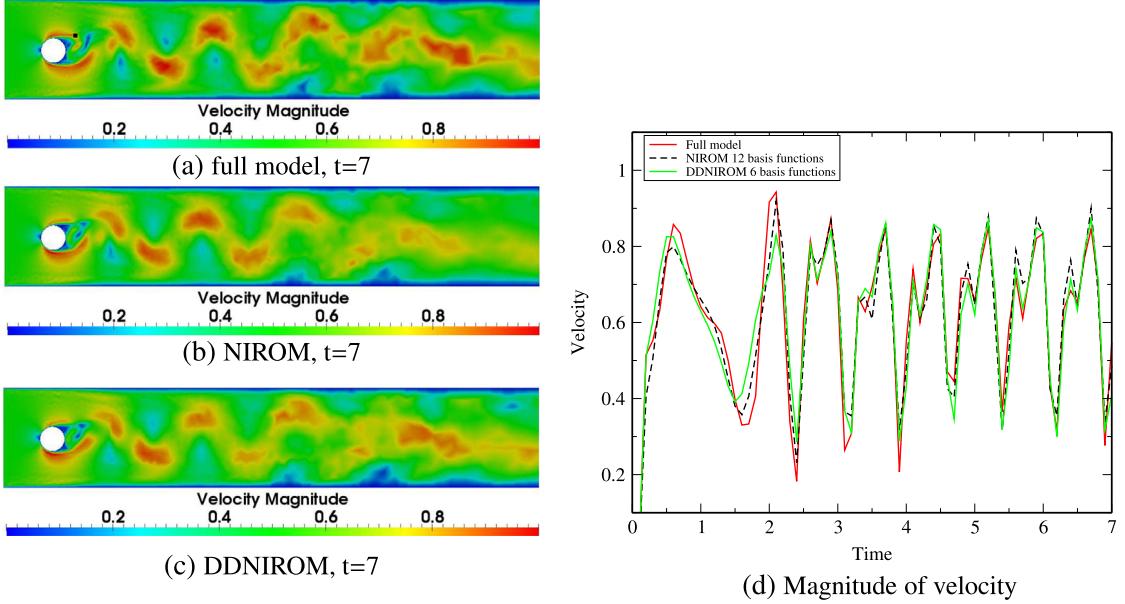


Fig. 10. Case 1: flow past a cylinder. The graphs (a)–(c) show the velocity solutions obtained from the full model, NIROM with 12 global basis functions and DDNIROM with various numbers local basis functions (given in Fig. 2b) at predicted time level $t = 7$. Graph (d) shows the x -component of velocity obtained from the full model, NIROM with 12 basis functions and DDNIROM with 6 local basis functions at a particular point $(x, y) = (0.290, 0.262)$, the location of this point is shown in plot (a) as a black point.

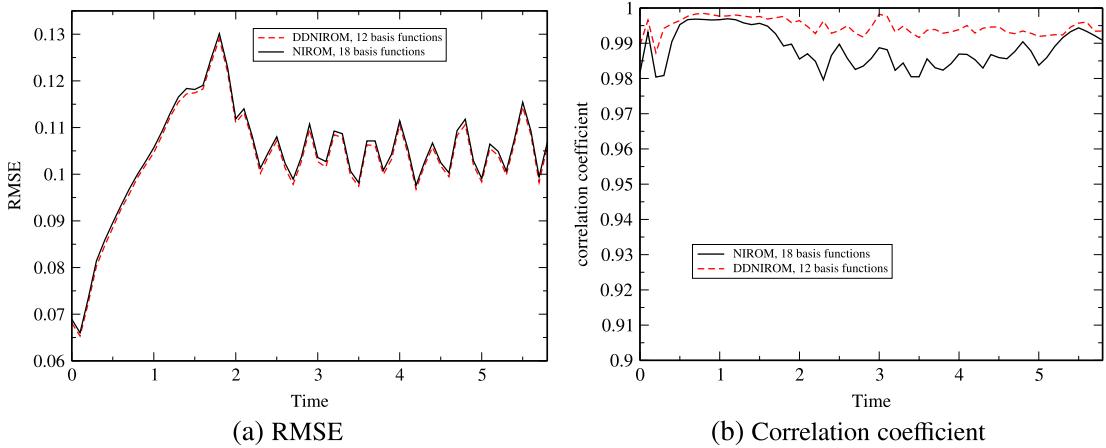


Fig. 11. Case 1: root-mean-square error and correlation coefficient of velocity solutions between the full model and NIROM with 18 global basis functions (black line); RMSE and CC of velocity solutions between the full model and DDNIROM with 12 local basis functions (red dotted line).

decrease rapidly and also, to problems with a larger number of degrees of freedom. The next two problems are slightly harder than this one, mainly, in terms of the rate of decrease in the global singular values.

Fig. 11 presents the root-mean-square error (RMSE) and the correlation coefficient (CC) of the solutions of the full model and the reduced order models (NIROM and DDNIROM respectively). The correlation coefficient is computed for each time step, and is defined as

$$\text{CC}(\psi(t), \psi_o(t)) = \frac{\text{cov}(\psi(t), \psi_o(t))}{\sigma_{\psi(t)} \sigma_{\psi_o(t)}} = \frac{\text{E}(\psi(t) - \mu_{\psi(t)})(\psi_o(t) - \mu_{\psi_o(t)})}{\sigma_{\psi(t)} \sigma_{\psi_o(t)}}. \quad (27)$$

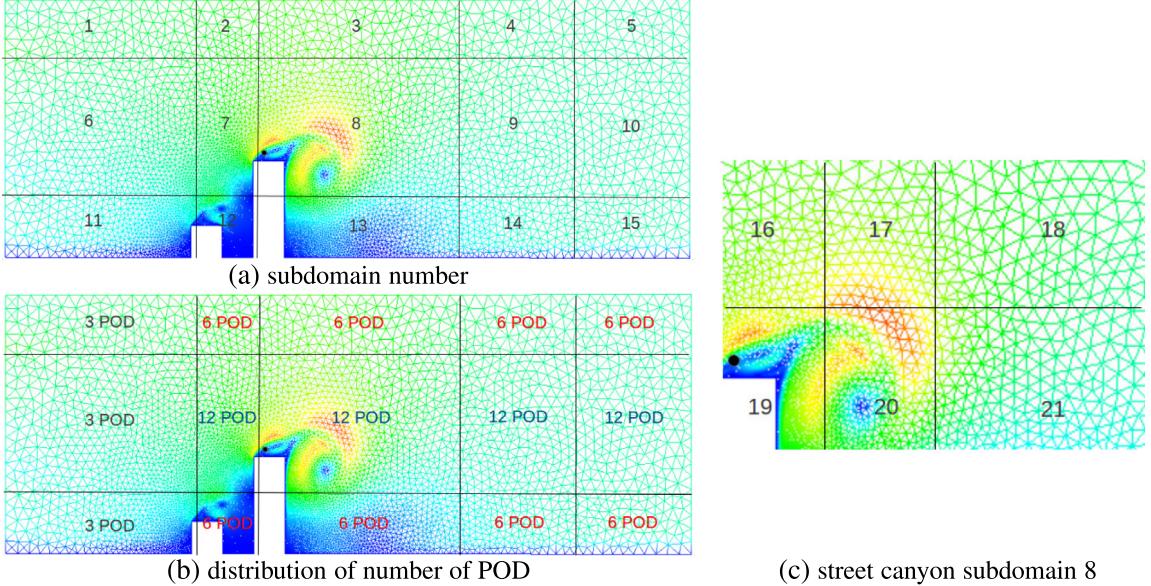


Fig. 12. Case 2: 2D urban street canyon test case. The 2D urban street canyon case: (a) the computational domain and 15 subdomains; (b) the number of POD basis functions chosen for each subdomain; and (c) 6 sub-subdomains within subdomain 8.

where $\mu_{\psi(t)}$ and $\mu_{\psi_o(t)}$ are expected values; $\sigma_{\psi(t)}$ and $\sigma_{\psi_o(t)}$ are standard deviations; $\psi_i(t)$ and $\psi_{o,i}(t)$ denote the DDNIROM solution (mapped onto the finite element mesh) and the full model solution at node i and time t , respectively; and M represents the number of nodes on the finite element mesh. The RMSE is defined as

$$\text{RMSE}(t) = \sqrt{\frac{\sum_{i=1}^M (\psi_i(t) - \psi_{o,i}(t))^2}{M}}, \quad (28)$$

The black line shows the RMSE and CC between the full model and NIROM with 18 global basis functions. The red dotted line shows the RMSE and CC of solutions between the full model and DDNIROM with 12 local basis functions. We can see that DDNIROM is in closer agreement with the full model than the global NIROM, as the correlation coefficient of DDNIROM is closer to a value of 1. There is not much difference between RMSE for NIROM and DDNIROM.

5.2. Case 2: 2D urban street canyon

In the second example we construct a DDNIROM of a 2D urban street canyon with two buildings. The computational domain has a size of $[0, 2] \times [0, 1]$ and comprises of an unstructured finite element mesh with 8264 nodes, see Fig. 12. A uniform velocity of 1 is set as the inflow boundary condition (left boundary). The simulation is impulsively initialised (in the same way as for case 1) from an initial velocity of zero. The top and bottom boundary conditions are free-slip and no-slip respectively. No-slip boundary conditions are prescribed to all the building surfaces. An open boundary condition is used at the outflow. A Reynolds number of 3000 was used where the length scale was given by the tallest building (0.3 units) and the viscosity was therefore 1×10^{-4} . The full model was run for a time interval of $[0, 0.8]$ with a time-step size of $\Delta t = 0.01$ and zero initial conditions. Forty snapshots were taken at regularly-spaced time levels of 0.02. In this example, the global domain was partitioned into 15 different subdomains, see Fig. 12. In the full model, the anisotropic Smagorinsky large eddy simulation (LES) model was used [59] in which the Smagorinsky coefficient was set to be 0.1.

The flow structure within subdomain 8 is complex. Thus, to capture the details of eddies, subdomain 8 is further subdivided into 3×2 subdomains labelled 16 to 21 in Fig. 12. Fig. 13 shows the global and local singular values over the global domain and 15 subdomains respectively. Fig. 14 shows the singular values over subdomain 8 and the subdomains 16 to 21 that fit within subdomain 8. In this figure it can be seen that the singular values of

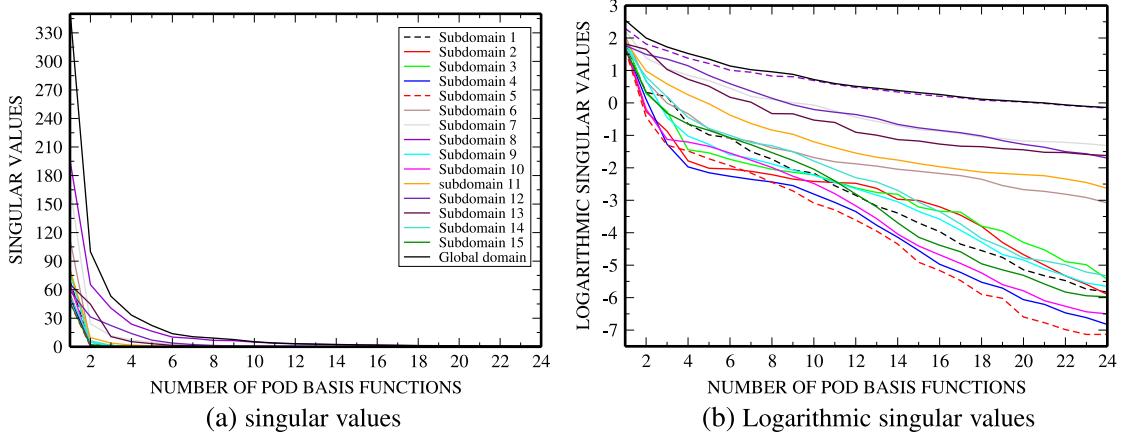


Fig. 13. Case 2: 2D urban street canyon test case. The global and local singular values over the global domain and 15 subdomains respectively.

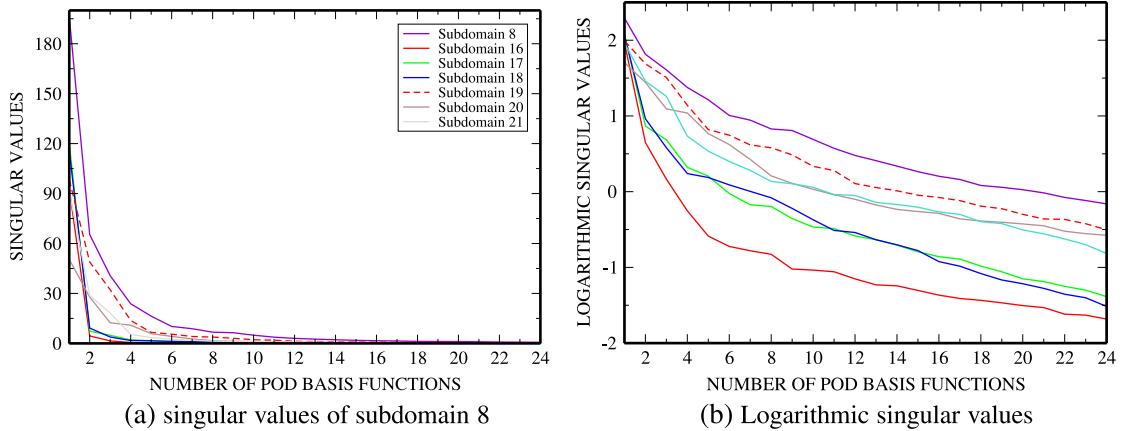


Fig. 14. Case 2: 2D urban street canyon test case. The singular values associated with subdomain 8 and the subdomains inside the subdomain 8.

subdomains 16 to 21 decay faster than those of subdomain 8 and the global domain. Different numbers of local basis functions are thus chosen for different subdomains so that most of energy is captured, as indicated in Fig. 12. The number of POD basis functions are again chosen, loosely based on Eq. (4). The graphs of the two most significant basis functions (*i.e.* the ones with the highest singular values) over the global domain and subdomains 2, 7, 8 and 12 are displayed in Fig. 15. These figures show that the local basis functions capture, and global basis functions reflect, the vortex shedding past the tallest building. The result of subdividing subdomain 8 into six subdomains 16 to 21 is that the singular values decrease even more rapidly for these subdomains than for subdomain 8 as seen in Fig. 14.

Fig. 16 presents the velocity magnitude distributions at time level 0.6, as obtained from the full model and DDNIROM using the numbers of local functions chosen for each subdomain shown in Fig. 12. There are only minor visual differences between the solutions of the two models. It can be seen from Fig. 16 that the DDNIROM captures well the small structures of the flow around the two buildings.

To further demonstrate the capability of the DDNIROM, first one and then three local basis functions (for each of the two velocity components and for pressure) are used for every subdomain to construct the DDNIROM. Fig. 17 shows the velocity solutions at time level 0.4, for the full model and for both of the global NIROM and the DDNIROM. Fig. 17 (f) and (g) show the error of velocity solutions from the NIROM and DDNIROM with 3 basis functions at time level 0.4. It can be seen that the DDNIROM captures well the eddies above the two buildings even using just 3 local basis functions, and performs better than the global NIROM. This is further confirmed by

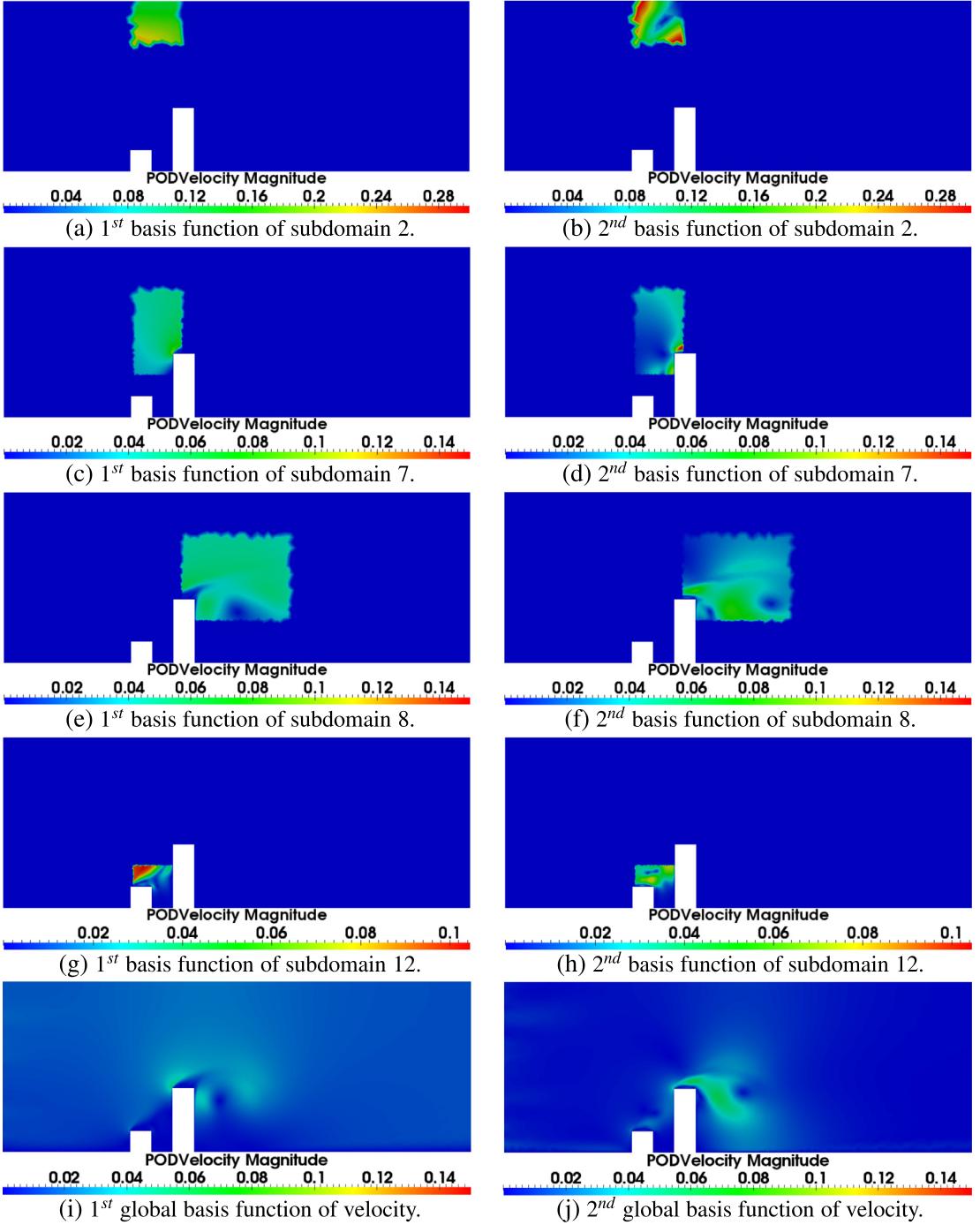


Fig. 15. Case 2: 2D urban street canyon test case. The first and second basis functions and global functions of velocity. The basis functions are plotted using Eq. (26).

the velocity solution at a particular point $(x, y) = (0.332, 0.328)$ shown in Fig. 18. With an increased number of basis functions, the accuracy of solutions from both the NIROM and DDNIROM has been improved. Again the DDNIROM with a maximum of 12 basis functions in each subdomain provides better results than the global NIROM with 18 basis functions, especially around the peak at $t = 0.04$. This is also highlighted by the correlation

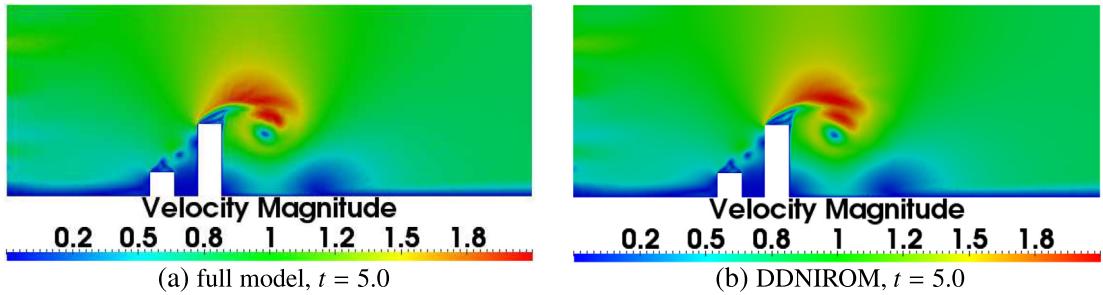


Fig. 16. Case 2: 2D urban street canyon test case. The magnitude of velocity from the full model and DDNIROM at time level 5. The numbers of local basis functions chosen for each subdomain are shown in Fig. 12.

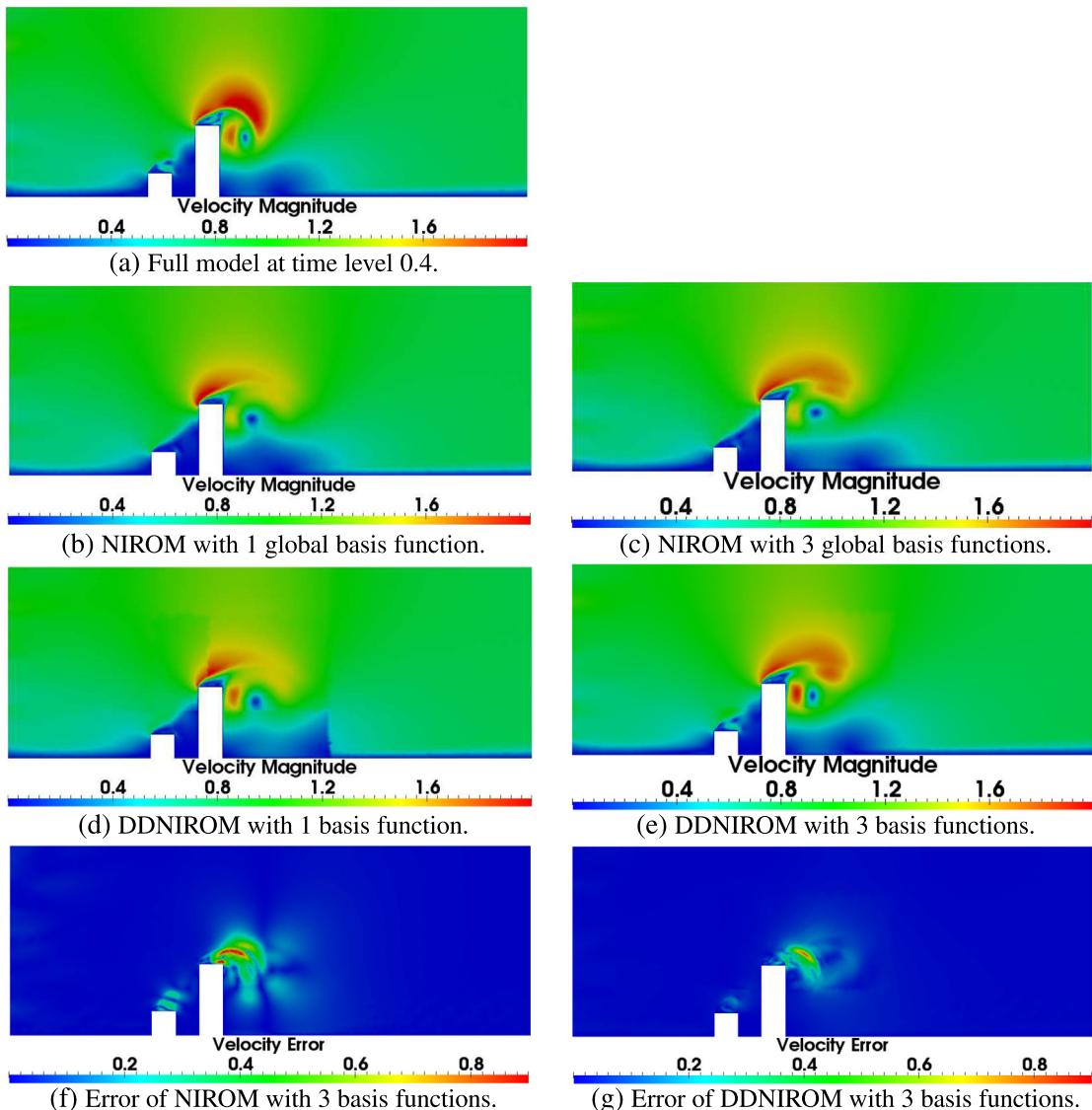


Fig. 17. Case 2: 2D urban street canyon test case. Figures (a) – (e) show the magnitude of velocity at time level 0.4, as calculated from the full model, global NIROM with 1 basis function and local DDNIROM with 3 basis functions for all the subdomains; and figures (f) and (g) show the velocity errors from both the global NIROM and DDNIROM with 3 basis functions at time level 0.4.

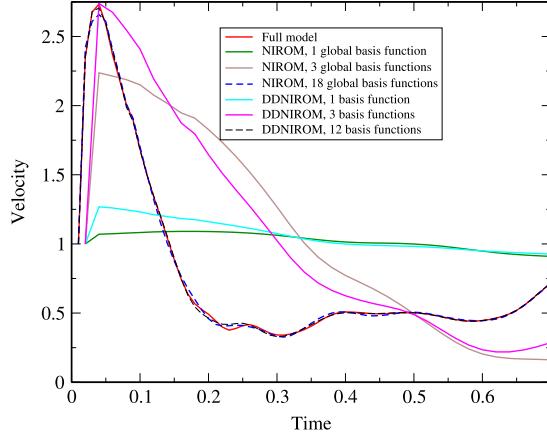


Fig. 18. Case 2: 2D urban street canyon test case. The x -component of velocity of the full model, the global NIROM with 1, 3 and 18 global basis functions, as well as the DDNIROM with 1, 3 and 12 basis functions at a particular point $(x, y) = (0.332, 0.328)$.

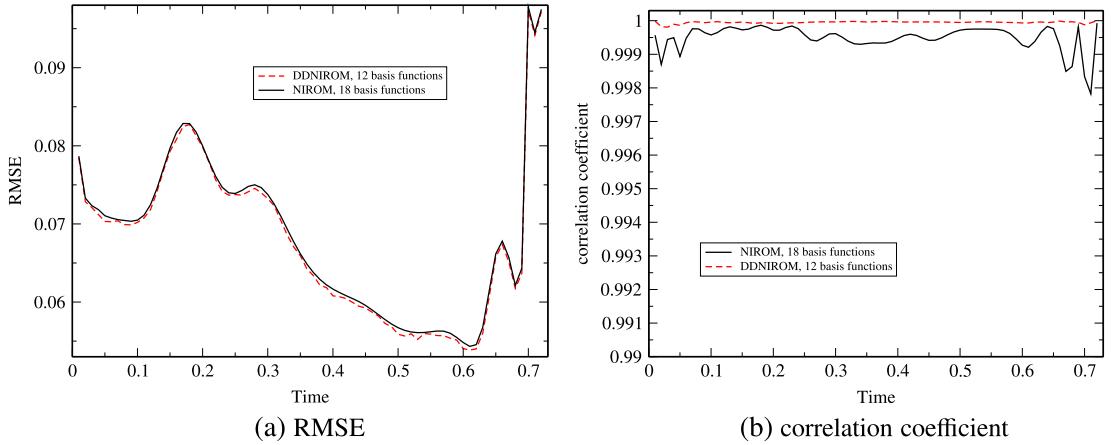


Fig. 19. Case 2: 2D urban street canyon test case. RMSE and CC of velocity solutions.

coefficient of velocity solutions, see Fig. 19. The reduced number of basis functions in DDNIROM leads to a decreased dimension of the hypersurfaces, thus decreasing the computational cost. Decreasing the dimension of the hypersurface also enables us to more accurately represent it.

The offline computational cost of the reduced order model is the sum of costs of calculating the snapshots, forming the POD basis functions and constructing the hypersurfaces. The time taken to generate the snapshots is just over 2 and a half minutes. Obtaining the basis functions and the hypersurfaces both take less than a second, giving an offline time of less than 3 minutes. As the problem solved here is small (a few thousand nodes and fewer than 20 basis functions), there is no significant difference in times between NIROM and DDNIROM.

The online computational cost includes the time for solving the DDNIROM and projecting the reduced order results onto the full space. For this example, the online cost amounts to approximately 0.02 seconds, giving a speed up of several orders of magnitude. The problems presented in this paper are very small, and it would be interesting to compare computational cost of NNIROM and DDNIROM for more challenging problems.

5.3. Case 3: 3D urban street canyon test case

In the third example the DDNIROM is applied to the simulation of a 3D street canyon with two buildings. The computational domain is shown in Fig. 20, consisting of a street canyon between two buildings. The domain

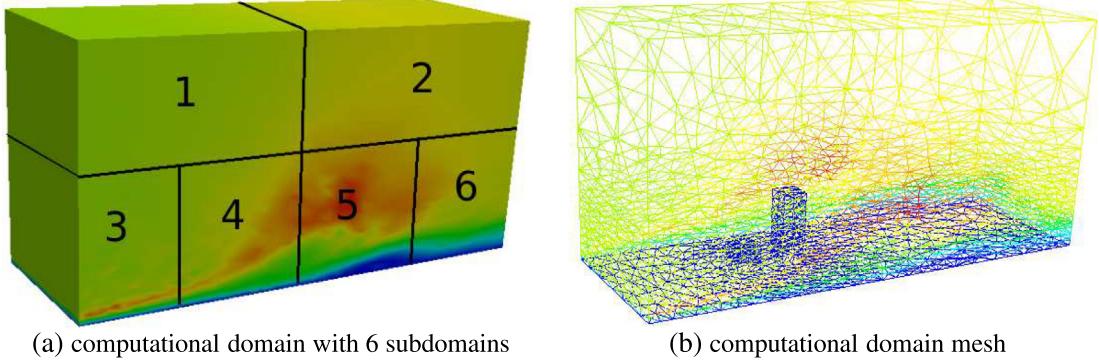


Fig. 20. Case 3: 3D urban street canyon. Computational domain and distribution of 6 subdomains. The velocity magnitude is shown.

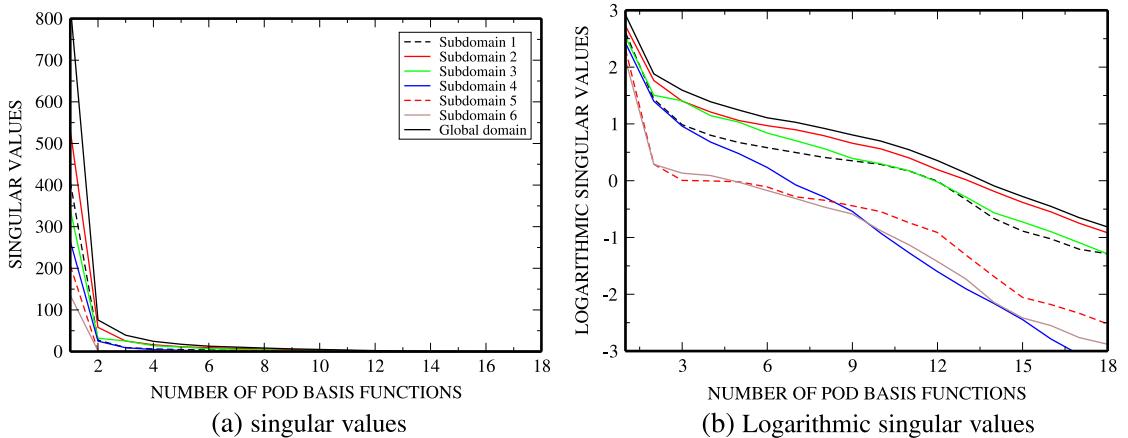


Fig. 21. Case 3: 3D urban street canyon test case. The global and local singular values over the global domain and 6 subdomains respectively.

has a dimensionless size of $[0, 2] \times [0, 0.7] \times [0, 1]$ and comprises of an unstructured finite element mesh with 9905 nodes, see Fig. 20. A uniform velocity of $(1, 0, 0)^T$ is given to the left side of the computational domain as the inflow boundary condition. Again the simulation is impulsively initialised from zero (at $t = 0$) in the same way as for the other two simulations. The sides and bottom boundary conditions are free-slip and no-slip respectively. The top boundary is a free-slip boundary condition. No-slip boundary conditions are prescribed to all the building surfaces. An open boundary condition is used at the outflow. The viscosity is 1×10^{-5} and the Reynolds number is 30000. The full model was simulated over a time interval of $[0, 0.312]$, and the solutions at time level 0.312 are chosen as the initial condition. The full model was then simulated for a time interval of $[0.312, 1.6]$ with a time-step size of $\Delta t = 0.008$. Eighty snapshots were stored at regularly-spaced time levels of 0.016. The anisotropic Smagorinsky large eddy simulation (LES) model is used with a second order Smagorinsky coefficient of 0.1, see [59]. In this example, the global domain was partitioned into 6 different subdomains, see Fig. 20. The domain is divided equally into two horizontal subdomains, and the top and bottom half are further divided equally into two and four subdomains respectively.

Fig. 21 shows the global and local singular values over the global domain and 6 subdomains respectively. In this figure the singular values of subdomains 1 to 6 decay faster than those of the global domain. Different numbers of local basis functions are thus chosen for different subdomains, loosely based on Eq. (4). Fig. 22 shows the first and second basis functions and global functions of velocity. In this case, 6 basis functions are used in subdomains 4 and 5, and two basis functions are used for other subdomains (1, 2, 3, and 6). Fig. 23(a)–(c) presents the velocity solutions at time level 0.632, as calculated from the full model and the NIROM with 6 global basis functions and the DDDNIROM with different number of local basis functions (six basis functions are used in subdomains 4 and 5, see Fig. 20; three basis functions are used in other subdomains); Fig. 23(d) and (e) show velocity errors from the

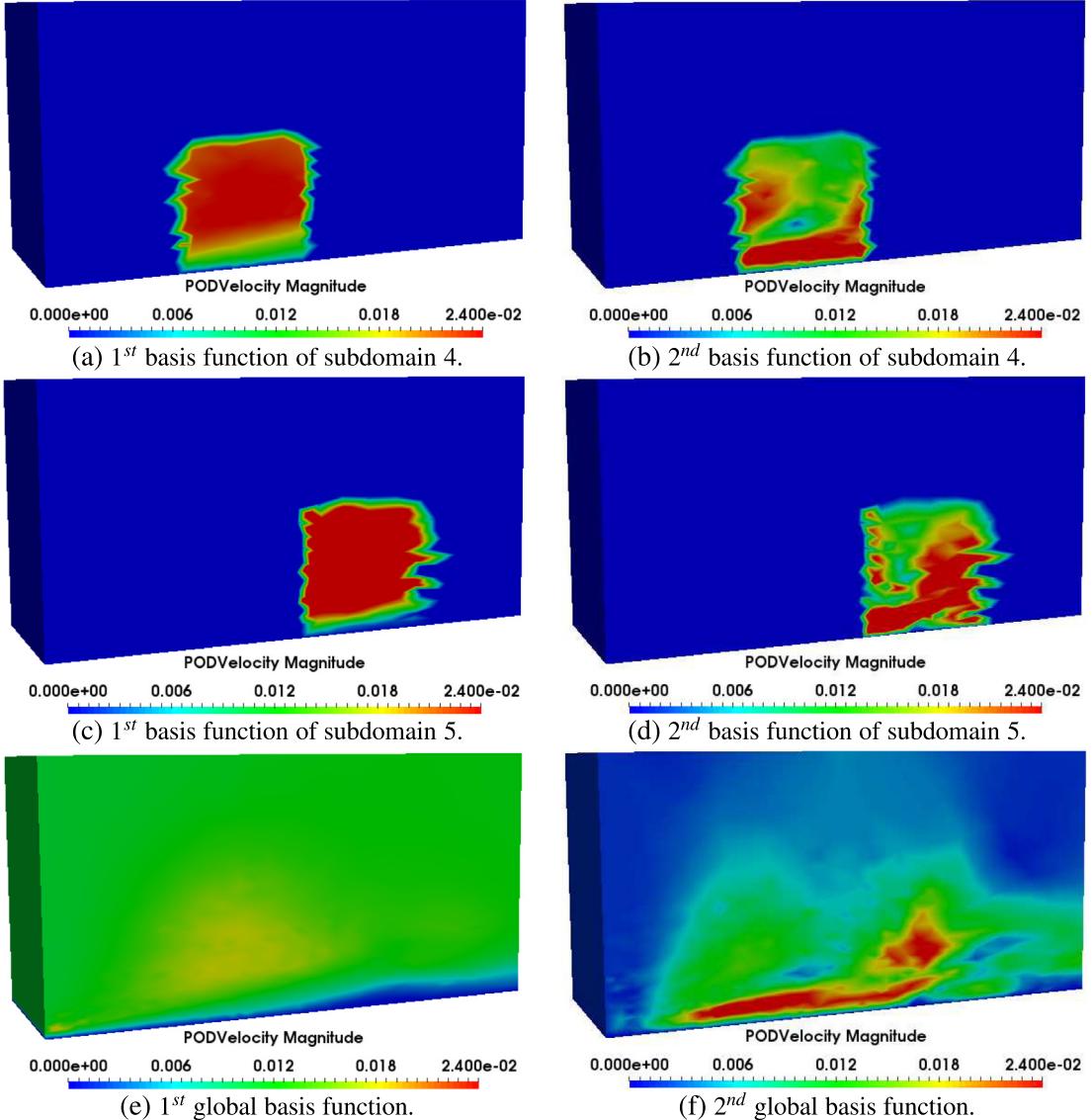


Fig. 22. Case 3: 3D urban street canyon. The first and second basis functions and global functions of velocity. The basis functions are plotted using Eq. (26).

NIROM with global basis functions and the DDNIROM with local basis functions at time level 0.632. As expected the DDNIROM errors are smaller than those of the global NIROM results (see Fig. 22).

6. Conclusions

In this article, a new Domain Decomposition Non-Intrusive Reduced Order Model (DDNIROM) is presented. The motivation of this development is to improve the capability of our recently developed NIROM [32,55] for complex flow problems over widely varying ranges of scales, for example, from modelling buildings to streets to cities. In this paper, we achieve this by partitioning the spatial computational domain into subdomains with a domain decomposition approach. Using Proper Orthogonal Decomposition, the local basis functions, within each subdomain, are generated based on the local solution snapshots over each subdomain. The Radial Basis Function interpolation method is then used to construct a set of hypersurfaces for each subdomain. These hypersurfaces

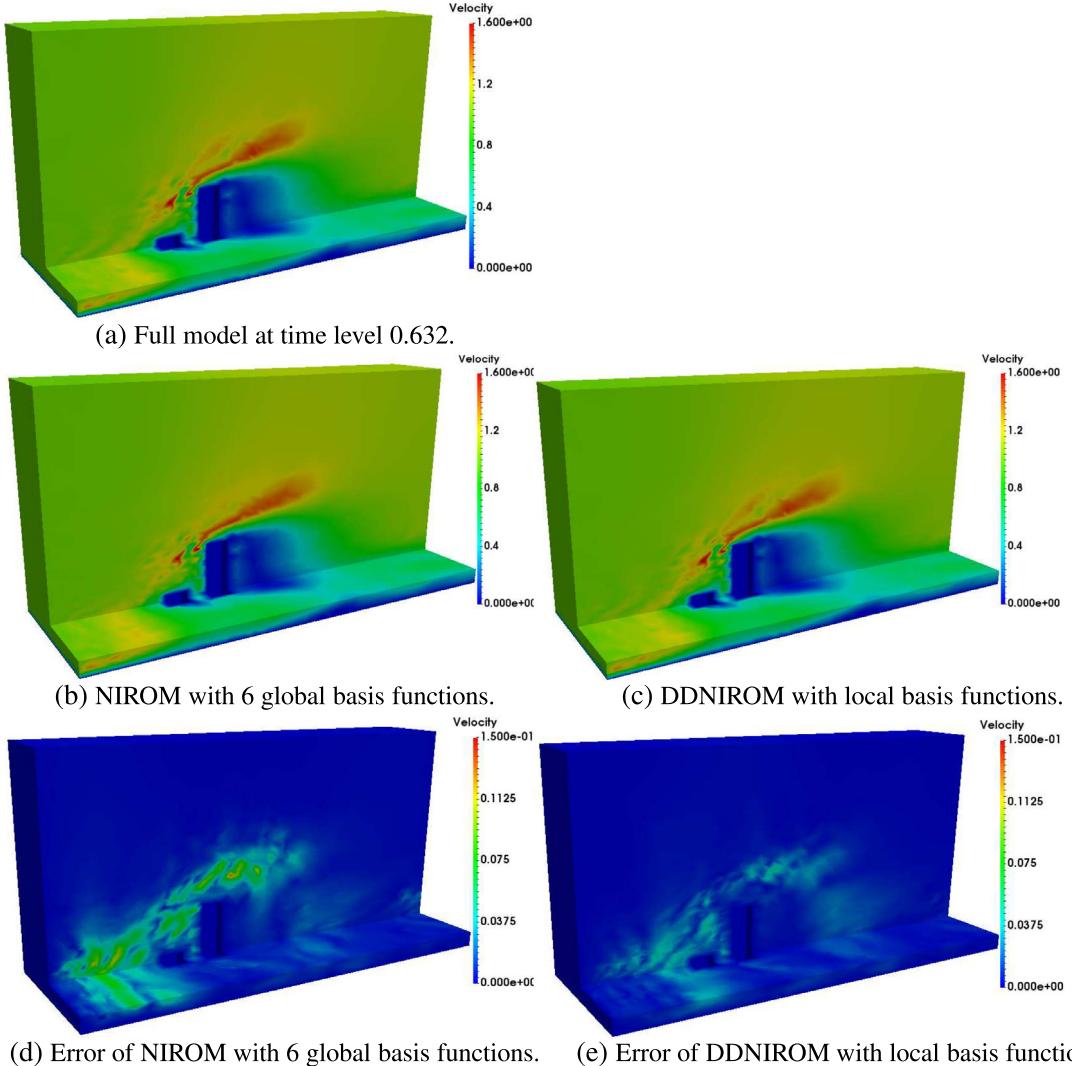


Fig. 23. Case 3: 3D urban street canyon test case. Figures (a) – (c) show the velocity solutions at time level 0.632, from the full model and from both of the NIROM with 6 global basis functions and DDNIROM with different numbers of local basis functions (six basis functions in subdomain 4 and 5, three basis functions in other subdomains, see Fig. 20); figures (d) and (e) show velocity errors from both the NIROM with global basis functions and DDNIROM with local basis functions at time level 0.632.

are designed to include the fluid dynamics not only from the subdomain itself, but also from the neighbouring subdomains. DDNIROMs have been constructed for three test cases: flow past a cylinder, and flows in 2D and 3D street canyons. Different numbers of basis functions were used in each subdomain depending on how rapidly the singular values decrease within each subdomain. A comparison between the high-fidelity full model, NIROM and DDNIROM has been conducted. The numerical results show that the DDNIROM exhibits comprehensive good agreement with the full model.

The DDNIROM is a generic and efficient approach for model reduction of general linear and non-linear time-dependent flow dynamical systems, and can be applied whether or not the original source code is available. Compared to existing ROMs that use global basis functions, the DDNIROM proposed here, is able to; (1) calculate more details of local flows since local POD basis functions are generated based on local flow solutions over each subdomain; (2) reduce the need for large multi-dimensional hypersurfaces representing the fluid dynamics using the global NIROM; and (3) reduce the size of the singular value decomposition and provide a greater scope for parallelisation of this. The DDNIROM is an essential step towards applying NIROMs to demanding real-life

scenarios with arbitrarily complex physics, e.g. air flows with traffic, chemistry etc. We conjecture that, to fully exploit DDNIROM, one would require ‘hard’ problems in which the global singular values do not necessarily decrease rapidly, but if the subdomains are sufficiently small and well chosen then the singular values associated with their POD basis functions would decrease more rapidly. Thus, next steps would be to apply this approach to much more complex problems.

Acknowledgements

The authors are grateful for the support of the following: Managing Air for Green Inner Cities (MAGIC), UK (EP/N010221/1); the EPSRC multi-phase flow programme, UK grant MEMPHIS (EP/K003976/1); the Innovate UK Smart-GeoWells consortium; funding from the European Union Seventh Frame work Programme (FP7/20072013) under grant agreement no.603663 for the research project PEARL (Preparing for Extreme And Rare events in coastal regions); the EPSRC MUFFINS project (MULTiphase Flow-induced Fluid-flexible structure InteractioN in Subsea applications EP/P033148/1) and the IMPACT research institute (Innovative Materials, Processing and Numerical Technologies) at Swansea University.

References

- [1] J. Oliver, M. Caicedo, A.E. Huespe, J.A. Hernández, E. Roubin, Reduced order modeling strategies for computational multiscale fracture, *Comput. Methods Appl. Mech. Engrg.* 313 (2017) 560–595.
- [2] B.R. Noack, K. Afanasiev, M. Morzyński, G. Tadmor, F. Thiele, A hierarchy of low-dimensional models for the transient and post-transient cylinder wake, *J. Fluid Mech.* 497 (2003) 335–363.
- [3] B.R. Noack, W. Stankiewicz, M. Morzyński, P.J. Schmid, Recursive dynamic mode decomposition of transient and post-transient wake flows, *J. Fluid Mech.* 809 (2016) 843–872.
- [4] X. Xie, D. Wells, Z. Wang, T. Iliescu, Approximate deconvolution reduced order modeling, *Comput. Methods Appl. Mech. Engrg.* 313 (2017) 512–534.
- [5] S. Lorenzi, A. Cammi, L. Luzzi, G. Rozza, POD-Galerkin method for finite volume approximation of Navier–Stokes and RANS equations, *Comput. Methods Appl. Mech. Engrg.* 311 (2016) 151–179.
- [6] Y. Wang, I.M. Navon, X. Wang, Y. Cheng, 2D Burgers Equation with large Reynolds number using POD/DEIM and calibration, *Internat. J. Numer. Methods Fluids* 82 (12) (2016) 909–931.
- [7] L. Cordier, B.R. Noack, M. Morzyński, G. Tadmor, O. Lehmann, S. Siegel, R. King, *Reduced-Order Modelling for Flow Control*, Vol. 528, Springer, 2011.
- [8] M. Bergmann, L. Cordier, Optimal control of the cylinder wake in the laminar regime by trust-region methods and POD reduced-order models, *J. Comput. Phys.* 227 (16) (2008) 7813–7840.
- [9] M. Bergmann, L. Cordier, J.P. Brachter, Optimal rotary control of the cylinder wake using proper orthogonal decomposition reduced-order model, *Physics of fluids* 17 (9) (2005) 097101.
- [10] F. Fang, T. Zhang, D. Pavlidis, C.C. Pain, A.G. Buchan, I.M. Navon, Reduced order modelling of an unstructured mesh air pollution model and application in 2D/3D urban street canyons, *Atmos. Environ.* 96 (2014) 96–106.
- [11] K.C. Hoang, Y. Fu, J.H. Song, An hp -proper orthogonal decomposition-moving least squares approach for molecular dynamics simulation, *Comput. Methods Appl. Mech. Engrg.* 298 (2016) 548–575.
- [12] A. Manzoni, F. Salmoiraghi, L. Heltai, Reduced basis isogeometric methods (RB-IGA) for the real-time simulation of potential flows about parametrized NACA airfoils, *Comput. Methods Appl. Mech. Engrg.* 284 (2015) 1147–1180.
- [13] H.A. Akhras, T. Elguedj, A. Gravouil, M. Rochette, Towards an automatic isogeometric analysis suitable trivariate models generation–application to geometric parametric analysis, *Comput. Methods Appl. Mech. Engrg.* 316 (2017) 623–645.
- [14] F. Ballarin, E. Faggiano, S. Ippolito, A. Manzoni, A. Quarteroni, G. Rozza, R. Scrofani, Fast simulations of patient-specific haemodynamics of coronary artery bypass grafts based on a POD–Galerkin method and a vascular shape parametrization, *J. Comput. Phys.* 315 (2016) 609–628.
- [15] L. Cordier, B.R. Noack, G. Tissot, G. Lehnasch, J. Delville, M. Balajewicz, G. Daviller, R.K. Niven, Identification strategies for model-based control, *Exp. Fluids* 54 (8) (2013) 1580.
- [16] M. Schlegel, B.R. Noack, On long-term boundedness of Galerkin models, *J. Fluid Mech.* 765 (2015) 325–352.
- [17] J. Osth, B.R. Noack, S. Krajnovic, D. Barros, J. Boree, On the need for a nonlinear subscale turbulence term in POD models as exemplified for a high-reynolds-number flow over an ahmed body, *J. Fluid Mech.* 747 (2014) 518–544.
- [18] L.P. Franca, S.L. Frey, Stabilized finite element methods: II. The incompressible Navier-Stokes equations, *Comput. Methods Appl. Mech. Engrg.* 99 (2) (1992) 209–233.
- [19] S. Chaturantabut, D.C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, *SIAM J. Sci. Comput.* 32 (2010) 2737–2764.
- [20] D. Xiao, Non-Intrusive Reduced Order Models and Their Applications (Ph.D. thesis), Imperial College London, 2016.
- [21] F. Fang, C.C. Pain, I.M. Navon, A.H. Elsheikh, J. Du, D. Xiao, Non-linear Petrov–Galerkin methods for reduced order hyperbolic equations and discontinuous finite element methods, *J. Comput. Phys.* 234 (2013) 540–559.

- [22] D. Xiao, F. Fang, J. Du, C.C. Pain, I.M. Navon, A.G. Buchan, A.H. ElSheikh, G. Hu, Non-linear Petrov–Galerkin methods for reduced order modelling of the Navier–Stokes equations using a mixed finite element pair, *Comput. Methods Appl. Mech. Engrg.* 255 (2013) 147–157.
- [23] L. Cordier, B. Abou El Majd, J. Favier, CaLibration of POD reduced-order models using tikhonov regularization, *Internat. J. Numer. Methods Fluids* 63 (2) (2010) 269–296.
- [24] K. Carlberg, C. Bou-Mosleh, C. Farhat, Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations, *Internat. J. Numer. Methods Engrg.* 86 (2) (2011) 155–181.
- [25] Y. Chu, M. Serpas, J. Hahn, State-preserving nonlinear model reduction procedure, *Chem. Eng. Sci.* 66 (17) (2011) 3907–3913.
- [26] F. Sabetghadam, A. Jafarpour, α Regularization of the POD-Galerkin dynamical systems of the kuramoto–sivashinsky equation, *Appl. Math. Comput.* 218 (10) (2012) 6012–6026.
- [27] K. Willcox, A. Megretski, Model reduction for large-scale linear applications, in: Proc.of 13th IFAC Symposium on System Identification, Rotterdam, Netherlands, 2003, pp. 1431–1436.
- [28] M. Barrault, Y. Maday, N.C. Nguyen, A.T. Patera, An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations, *C. R. Math.* 339 (9) (2004) 667–672.
- [29] D. Xiao, F. Fang, A.G. Buchan, C.C. Pain, I.M. Navon, J. Du, G. Hu, Non-linear model reduction for the Navier–Stokes equations using residual DEIM method, *J. Comput. Phys.* 263 (2014) 1–18.
- [30] K. Carlberg, C. Farhat, J. Cortial, D. Amsallem, The GNAT method for nonlinear model reduction: Effective implementation and application to computational fluid dynamics and turbulent flows, *J. Comput. Phys.* 242 (2013) 623–647.
- [31] J. Du, F. Fang, C.C. Pain, I.M. Navon, J. Zhu, D.A. Ham, POD Reduced-order unstructured mesh modeling applied to 2d and 3D fluid flow, *Comput. Math. Appl.* 65 (3) (2013) 362–379.
- [32] D. Xiao, F. Fang, A.G. Buchan, C.C. Pain, I.M. Navon, A. Muggeridge, Non-intrusive reduced order modelling of the Navier–Stokes equations, *Comput. Methods Appl. Mech. Engrg.* 293 (2015) 522–541.
- [33] D.A. Bistrian, I.M. Navon, Randomized dynamic mode decomposition for non-intrusive reduced order modelling, *Internat. J. Numer. Methods Engrg.* 112 (2017) 3–25.
- [34] D. Xiao, P. Yang, F. Fang, J. Xiang, C.C. Pain, I.M. Navon, Non-intrusive reduced order modeling of fluid-structure interactions, *Comput. Methods Appl. Mech. Engrg.* 303 (2016) 35–54.
- [35] D. Xiao, P. Yang, F. Fang, J. Xiang, C.C. Pain, I.M. Navon, M. Chen, A non-intrusive reduced-order model for compressible fluid and fractured solid coupling and its application to blasting, *J. Comput. Phys.* 330 (2017) 221–244.
- [36] D. Xiao, Z. Lin, F. Fang, C.C. Pain, I.M. Navon, P. Salinas, A. Muggeridge, Non-intrusive reduced-order modeling for multiphase porous media flows using smolyak sparse grids, *Internat. J. Numer. Methods Fluids* 83 (2) (2017) 205–219.
- [37] D. Xiao, F. Fang, C.C. Pain, I.M. Navon, A. Muggeridge, Non-intrusive reduced order modelling of waterflooding in geologically heterogeneous reservoirs, in: ECMOR XV-15th European Conference on the Mathematics of Oil Recovery, 2016.
- [38] D. Xiao, F. Fang, C.C. Pain, I.M. Navon, A parameterized non-intrusive reduced order model and error analysis for general time-dependent nonlinear partial differential equations and its applications, *Comput. Methods Appl. Mech. Engrg.* 317 (2017) 868–889.
- [39] D.J. Lucia, P.I. King, P.S. Beran, Reduced order modeling of a two-dimensional flow with moving shocks, *Comput. & Fluids* 32 (7) (2003) 917–938.
- [40] J.S. Przemieniecki, Matrix structural analysis of substructures, *AIAA J.* 1 (1) (1963) 138–147.
- [41] I.M. Navon, Y. Cai, Domain decomposition and parallel processing of a finite element model of the shallow water equations, *Comput. Methods Appl. Mech. Engrg.* 106 (1–2) (1993) 179–212.
- [42] Y. Cai, I.M. Navon, Parallel block preconditioning techniques for the numerical simulation of the shallow water flow using finite element methods, *J. Comput. Phys.* 122 (1) (1995) 39–50.
- [43] L. Berger-Vergiat, H. Waisman, An overlapping domain decomposition preconditioning method for monolithic solution of shear bands, *Comput. Methods Appl. Mech. Engrg.* 318 (2017) 33–60.
- [44] S. Taverniers, D.M. Tartakovsky, A tightly-coupled domain-decomposition approach for highly nonlinear stochastic multiphysics systems, *J. Comput. Phys.* 330 (2017) 884–901.
- [45] X. Bian, Z. Li, G.E. Karniadakis, Multi-resolution flow simulations by smoothed particle hydrodynamics via domain decomposition, *J. Comput. Phys.* 297 (2015) 132–155.
- [46] J. Baiges, R. Codina, S. Idelsohn, A domain decomposition strategy for reduced order models. application to the incompressible Navier–Stokes equations, *Comput. Methods Appl. Mech. Engrg.* 267 (2013) 23–42.
- [47] D. Amsallem, M.J. Zahr, C. Farhat, Nonlinear model order reduction based on local reduced-order bases, *Internat. J. Numer. Methods Engrg.* 92 (10) (2012) 891–916.
- [48] S. Chaturantabut, Temporal localized nonlinear model reduction with a priori error estimate, *Appl. Numer. Math.* 119 (2017) 225–238.
- [49] P. Kerfriden, O. Goury, T. Rabczuk, S.P.A. Bordas, A partitioned model order reduction approach to rationalise computational expenses in nonlinear fracture mechanics, *Comput. Methods Appl. Mech. Engrg.* 256 (2013) 169–188.
- [50] G.S.H. Pau, C. Shen, W.J. Riley, Y. Liu, Accurate and efficient prediction of fine-resolution hydrologic and carbon dynamic simulations from coarse-resolution models, *Water Resour. Res.* 52 (2) (2016) 791–812.
- [51] G.S.H. Pau, G. Bisht, W.J. Riley, A reduced-order modeling approach to represent subgrid-scale hydrological dynamics for land-surface simulations: application in a polygonal tundra landscape, *Geosci. Model Dev.* 7 (5) (2014) 2091–2105.
- [52] D.J. Lucia, P.I. King, P.S. Beran, Domain decomposition for reduced-order modeling of a flow with moving shocks, *AIAA J.* 40 (11) (2002) 2360–2362.
- [53] H. Antil, M. Heinkenschloss, R.H.W. Hoppe, D.C. Sorensen, Domain decomposition and model reduction for the numerical solution of PDE constrained optimization problems with localized optimization variables, *Comput. Vis. Sci.* 13 (6) (2010) 249–264.

- [54] H. Antil, M. Heinkenschloss, R.H.W. Hoppe, Domain decomposition and balanced truncation model reduction for shape optimization of the Stokes system, *Optim. Methods Softw.* 26 (4–5) (2011) 643–669.
- [55] D. Xiao, F. Fang, C.C. Pain, G. Hu, Non-intrusive reduced order modelling of the Navier-Stokes equations based on RBF interpolation, *Internat. J. Numer. Methods Fluids* 79 (11) (2015) 580–595.
- [56] M. Mongillo, Choosing basis functions and shape parameters for radial basis function methods, *SIAM Undergrad. Res. Online* 4 (2011) 190–209.
- [57] C.C. Pain, M.D. Piggott, A.J.H. Goddard, F. Fang, G.J. Gorman, D.P. Marshall, M.D. Eaton, P.W. Power, C.R.E. de Oliveira, Three-dimensional unstructured mesh ocean modelling, *Ocean Model.* 10 (1–2) (2005) 5–33.
- [58] C.C. Pain, A.P. Umpleby, C.R.E. de Oliveira, A.J.H. Goddard, Tetrahedral mesh optimisation and adaptivity for steady-state and transient finite element calculations, *Comput. Methods Appl. Mech. Engrg.* 190 (29–30) (2001) 3771–3796.
- [59] E. Aristodemou, L.M. Boganegra, L. Mottet, D. Pavlidis, A. Constantinou, C.C. Pain, A. Robins, H. ApSimon, How tall buildings affect turbulent air flows and dispersion of pollution within a neighbourhood, *Environ. Pollut.* 233 (2018) 782–796.