INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2014; **00**:1–18
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld

Model identification of reduced order fluid dynamics systems using deep learning

Z. Wang[1], D. Xiao [2 4 5 *], F. Fang[2 5], R. Govindan[3], C.C. Pain[2 5], Y. Guo[5]

[1]*School of Earth Sciences and Engineering, Sun Yat-Sen University*
[2]*Applied Modelling and Computation Group, Department of Earth Science and Engineering,Imperial College London,Prince Consort Road, London, SW7 2BP, UK.URL:* http://amcg.ese.imperial.ac.uk
[3]*College of Science and Engineering, Hamad Bin Khalifa University, Qatar Foundation, Doha, Qatar*
[4]*Department of Chemical Engineering, Imperial College London*
[5]*Data assimilation lab, Data Science Institute, Imperial College London*

SUMMARY

This paper presents a novel model reduction method: Deep Learning Reduced Order Model (DLROM), which is based on proper orthogonal decomposition (POD) and deep learning methods. The deep learning approach is a recent technological advancement in the field of artificial neural networks. It has the advantage of learning the non-linear system with multiple levels of representation and predicting data. In this work, the training data are obtained from high fidelity model solutions at selected time levels. The Long Short-Term Memory network (LSTM) is used to construct a set of hypersurfaces representing the reduced fluid dynamic system. The model reduction method developed here is independent of the source code of the full physical system.
The reduced order model (ROM) based on deep learning has been implemented within an unstructured mesh finite element fluid model. The performance of the new ROM is evaluated using two numerical examples: an ocean gyre and flow past a cylinder. These results illustrate that the CPU cost is reduced by several orders of magnitude whilst providing reasonable accuracy in predictive numerical modelling. Copyright © 2014 John Wiley & Sons, Ltd.

Received . . .

KEY WORDS: deep learning, LSTM, POD, ROM

1. INTRODUCTION

Model identification is a method for constructing mathematical models of dynamic physical systems, and it involves numerical simulations. Numerical simulations play an important part in analyses of many natural systems in physics, climatology, astrophysics, biology, chemistry, economics, psychology and engineering. These simulations generally involve solving partial differential equations (PDEs) with suitable initial and boundary conditions using discretisation methods (for example, finite element methods). However, the discretised dynamical system for complex problems often has a huge number of degrees of freedom (*e.g.* $10^7 - 10^{10}$). Thus, even with moderate complexity problems, the computation cost can still be prohibitive. A number of model identification methods have been presented to improve the capabilities of the mathematical models such as state calibration with 4D Var[1], neural networks[2], dynamics with radial basis functions[3], genetic programming [4, 2, 5] and the traditional Galerkin method [1, 6]. Reduced-order modeling, one of model identification methods, has proven to be a powerful tool to reduce the

*Correspondence to: Applied Modelling and Computation Group, Department of Earth Science and Engineering, Imperial College London. E-mail: dh.xiao@imperial.ac.uk

Copyright © 2014 John Wiley & Sons, Ltd.
Prepared using fldauth.cls *[Version: 2010/05/13 v2.00]*

complexity and large dimensional size of the full discretised dynamical system. It is also often used in uncertainty quantification or optimal design where many simulations (*e.g.* hundreds or thousands) are needed to analyse model parameters (see [7, 8, 9, 10, 11]).

ROM offers the potential to simulate physical and dynamic systems with substantially increased computational efficiency whilst maintaining reasonable accuracy [12]. ROM is a rapidly growing discipline, with significant potential advantages in: interactive use, emergency response, ensemble calculations, and data assimilation. ROM is expected to play a major role in facilitating real time turn-around of computational results. It has been applied into a number of fields, for example, nonlinear large-scale systems [13], ocean modelling [14, 15], sensor location optimisation [16], air pollution modelling [17], shape optimisation [18], porous media problems [19], aerospace [20, 21], optimal control [22, 23], multi-scale fracture [24], shallow water [25, 26, 27] and neutron problems [28].

Reduced order models (ROMs) can be broadly divided into two types:intrusive ROMs (IROMs) and non-intrusive ROMs (NIROMs) [29]. IROMs are dependent on the source code of the governing system and suffer from instability and non-linearity efficiency issues [30, 31, 32, 33, 34, 35, 36, 37]. In addition, the source code of IROMs is difficult to modify [38]. The NIROMs become popular recently since they avoid modification of the original source code for complex dynamical systems. Audouze *et al.* proposed a NIROM for nonlinear parameterized time-dependent PDEs using RBF and POD [39, 40]. Xiao *et al.* presented three types of non-intrusive model reduction methods: (a) POD and Smolyak sparse grid interpolation; (b) POD and Taylor series expansion; and (c) POD and radial basis function interpolation [41, 42]. Chen *et al.* proposed a NIROM based on black-box stencil interpolation method [38]. Xiao *et al.* also presented a parameterized NIROM for general time-dependent nonlinear PDEs [3]. The NIROM method has been applied successfully into a number of fields such as fluid-structure interactions [43, 44], ocean modelling [42] and multiphase flow in porous media [45, 46]. These applications were carried out using POD and RBF. However, their predictive ability is dependent on the choice of interpolation functions and distribution of the sample data points.

Deep learning technology is a recent advancement in artificial neural networks which is capable of finding more hidden information from the data. It has the advantage of processing data in their raw form, learning the non-linear system with multiple levels of representation and predicting data [47]. Recently, it has resulted in breakthroughs in various areas like image processing, video and speech recognition, genetics and disease diagnosis [48, 49, 50]. More recently, an article about the application of deep learning in fluid dynamics has also been reported [51].

The deep learning method is also used to construct a reduced order model. In the work of [52], Deep Auto-Encoder (DAE) is used for dimensionality reduction of distributed parameter systems. Then, the low-dimensional representations are used to establish a reduced-order method. In that work, the deep learning method is used to do dimensionality reduction, but it does not take the advantage of its predictive capability.

The LSTM was presented by Sepp Hochreiter and Jurgen Schmidhuber in 1997 [53], and it was improved by Gers *et al.* in 2000. Around 2007, LSTM outperformed traditional neural networks in speech recognition [54]. In 2009, it won a number of pattern recognition competitions [55]. More recently, it has become a very popular method in major IT companies such as Google, Microsoft, Apple, Baidu and Amazon [56, 57, 58, 59].

In this work the authors have developed a novel NIROM based on LSTM deep learning and proper orthogonal decomposition (POD) method. LSTM is used to construct a set of hypersurfaces that represent the reduced fluid dynamic system. In this method, the governing equations are firstly discretised by a finite element Bubnov Galerkin discretisation using the Fluidity model [60] and the NIROM is generated through the solution snapshots taken at regular time intervals. In this method, solutions of the full model are recorded as a set of snapshots, and from these snapshots, appropriate basis functions optimally representing the problem are generated. LSTM is then used to learn the underlying physical dynamics comprised in the solutions of the full high fidelity model. The predictive capability is also compared with the widely used NIROM based on POD and RBF.

Copyright © 2014 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

The remainder of this paper is organised as follows. Section 2 presents the theory involved in this work, which includes the governing equations of the fluid problem and methods for deriving the formulation of the reduced order system. Section 3 illustrates the capability of the newly derived model reduction method by two numerical examples: a gyre and flow past a cylinder problem. Finally in Section 4, discussion and conclusions are presented.

2. THEORY AND METHOD

2.1. Governing equation of the physical system

This article considers the three dimensional non-hydrostatic Navier-Stokes equations describing the conservation of mass and momentum of a fluid,

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

$$\frac{\partial \underline{\mathbf{u}}}{\partial t} + \underline{\mathbf{u}} \cdot \nabla \underline{\mathbf{u}} + f\mathbf{k} \times \underline{\mathbf{u}} = -\nabla p + \nabla \cdot \tau, \tag{2}$$

where the terms $\underline{\mathbf{u}} \equiv (u_x, u_y, u_z)^T$ denote the velocity vector, p the perturbation pressure ($p := p/\rho_0$, ρ_0 is the constant reference density) and f the Coriolis inertial force. The stress tensor τ represents the viscous forces.

In this work, a finite element Bubnov-Galerkin discretisation is employed for the Navier Stokes equations [60]. After weighting and integrating, the discretised form of the system can be obtained,

$$C^T \mathbf{u} = 0,$$

$$N\frac{\partial \mathbf{u}}{\partial t} + A(\mathbf{u})\mathbf{u} + K\mathbf{u} + Cp = \mathbf{s}, \tag{3}$$

where C denotes the pressure gradient matrix, N is the mass matrix involving the finite element basis functions N_i, $A(\mathbf{u})$ is the solution dependent discretised streaming operator, K is the matrix related to the rest of the linear terms of velocity, and $\mathbf{s}$ is the vector accounting for the forces acting upon the solution.

The degree of freedom (DOF) of the matrices (N, A, K and C) in equation (3) is huge. In some realistic cases, it can be up to 10^{10} or higher. The computational cost of solving the equations is intensive. In the following sections, a reduced system is derived by using POD and LSTM deep learning methods.

2.2. The Reduced Order System of Equations

In the POD approach, a sequence of basis functions is constructed from snapshots that are taken at a number of time levels of the high fidelity full model given by equation (3).

For each solution field, *e.g.*, velocity or pressure components, the sampled values at the snapshot i are stored in the vectors $\mathcal{V}_i^k$ (where the superscript k denotes space direction (x, y, z or pressure component) with the number of nodes $\mathcal{N}$. Each snapshot matrix for velocity components or pressure are treated separately in the discretised formulation. Thus the procedure is implemented on the general snapshot matrix $\mathcal{V}$ in order to derive the reduced system.

A set of basis functions $\{\phi_i\}$, $i \in \{1, 2, \ldots, I\}$, are obtained by POD, which involves performing a singular value decomposition of the snapshot matrix $\mathcal{V}$,

$$\mathcal{V} = U\Sigma V^T. \tag{4}$$

The terms $U \in R^{\mathcal{N}\times\mathcal{N}}$ and $V \in R^{I\times I}$ are the matrices that consist of the orthogonal vectors for $\mathcal{V}\mathcal{V}^T$ and $\mathcal{V}^T\mathcal{V}$, respectively and Σ is a diagonal matrix of size $\mathcal{N} \times I$. The non zero values of Σ are the singular values of $\mathcal{V}$, and these are assumed to be listed in order of their decreasing magnitude. It can be shown [61] that the POD vectors are defined to be the column vectors of the matrix V,

$$\phi_i = \mathcal{V}V_i/\sqrt{\lambda_i}, \qquad \text{for } i \in \{1, 2 \ldots I\}, \tag{5}$$

Copyright © 2014 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

and the optimal basis set of size P consists of the functions corresponding to the largest P singular values (*i.e.* the first P columns of U). λ denotes singular values. These vectors are optimal in the sense that no other rank P set of basis vectors can be closer to the snapshot matrix $\mathcal{V}$ in the Frobenius norm.

In POD, any variable ψ (for example, the velocity and pressure components) can be expressed by the expansion,

$$\psi = \sum_{j=1}^{P} \alpha_j \phi_j + \overline{\psi} \tag{6}$$

where α_j denote the coefficients of the POD expansion and $\overline{\psi}$ is the mean of the ensemble of snapshots for the variable ψ. In this work, the mean of snapshots were not considered in the equation (6), *i.e.*, $\overline{\psi}$=0. It was previously found that in some cases, considering the snapshot mean may lower the accuracy [44].

The loss of information due to the truncation of the POD expansion set to P vectors can be quantified by the following ratio,

$$E = \frac{\sum_{j=1}^{P} \lambda_j^2}{\sum_{j=1}^{I} \lambda_j^2}, \tag{7}$$

The value of E will tend to 1 as P is increased to the value I, this would imply no loss of information.

2.3. *Modelling fluids using deep learning and POD*

Substituting equation (6) into equation (3) and taking the POD basis function as the test function, then integrating equation (2) over the computational domain, the reduced order equations are then obtained,

$$\begin{bmatrix} B^{POD} & C^{POD} \\ (C^{POD})^T & 0 \end{bmatrix} \begin{bmatrix} \alpha^{u,n+\Delta n} \\ \alpha^{p,n+\Delta n} \end{bmatrix} = \begin{bmatrix} B'^{POD} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha^{u,n} \\ \alpha^{p,n} \end{bmatrix} + \begin{bmatrix} s^{POD} \\ 0 \end{bmatrix}, \tag{8}$$

where $\alpha^{u,n}$ is the vector of POD coefficients containing all velocity components at time level n. Δn is the size of the timestep. In order to keep consistent, $n + \Delta n$ is treated as next timestep $t + 1$. Equation (8) can also be rewritten in the general form of equation (9):

$$\alpha_j^{t+1} = f_j(\alpha^t), \quad j \in \{1, 2, \dots m\}, \tag{9}$$

In this article, the function $\tilde{f}_j(\alpha^t)$, which approximates the $f_j(\alpha^t)$ in equation (9), is constructed using LSTM deep network. The structure of a deep recurrent neural network used in this work is shown in figure 1. This type of network has cyclic connections, which make the network a powerful method to model temporal data since it has an internal memory system to deal with temporal sequence inputs. The LSTM has a special memory block in the hidden layer of the recurrent neural network. The architecture of a single memory block is shown in figure 2. The input gate of each memory block controls the information transmitting from the input activations into the cell and the output gate controls the information transmitting from the memory cell activations into other nodes.

In the LSTM network, a map from the input $x = (x^1, ..., x^{N_t})$ to the output $y = (y^1, ..., y^{N_t})$ can be calculated using the following equations:

$$\begin{aligned} i^t &= \varrho(W_{ix}x^t + W_{im}m^{t-1} + W_{ic}c^{t-1} + b_i), \\ r^t &= \varrho(W_{rx}x^t + W_{rm}m^{t-1} + W_{rc}c^{t-1} + b_r), \\ o^t &= \varrho(W_{ox}x^t + W_{om}m^{t-1} + W_{oc}c^t + b_o), \\ c^t &= r^t \odot c^{t-1} + i^t \odot g(W_{cx}x^t + W_{cm}m^{t-1} + b_c), \\ m^t &= o^t \odot h(c^t), \\ y^t &= \varsigma(W_{rm}m^t + b_r) \end{aligned} \tag{10}$$

where i, r and o denote the input, forget and output gate vectors respectively, c is the cell activation vector, b is the bias vector. ϱ is the activation function, W denote the weight matrices (*e.g.* W_{ix} is

Copyright © 2014 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

the weight matrix from the input gate to the input), $\odot$ is the element wise product of the vectors, h and g are the cell output and cell input activation functions respectively and ς is the network output activation function.

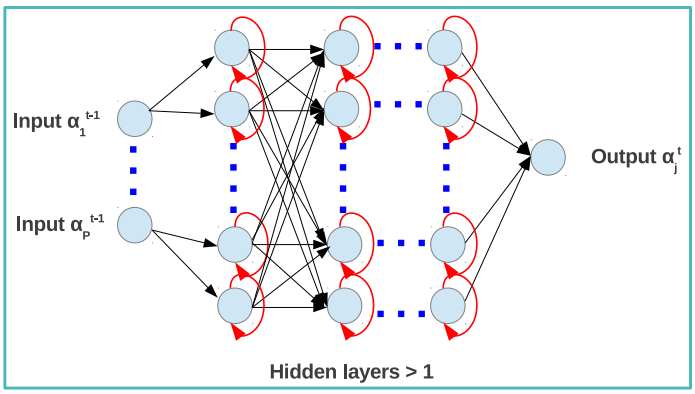


Figure 1. LSTM deep learning architecture

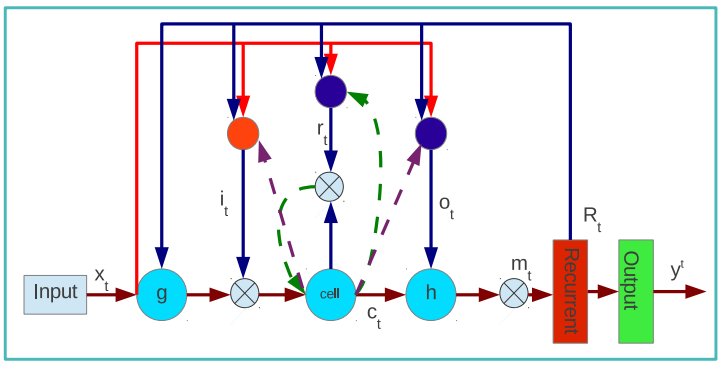


Figure 2. One memory node in the deep learning architecture

In this work, the inputs $x = (x^1, ..., x^{N_t})$ are the POD coefficients at the previous time level, $\alpha = (\alpha^{t-1}, ..., \alpha^{N_t-1})$ and the outputs are the POD coefficients at the current time level, $\alpha = (\alpha^t, ..., \alpha^{N_t})$. The offline procedure of constructing a ROM using POD and LSTM can be summarised as following,

(a) Generate a sequence of snapshots over the time period $[0, N^t]$ by solving the Fluidity model;

Copyright © 2014 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

(b) Obtain a set of POD basis functions for velocity and pressure using SVD of the snapshot matrix;

(c) Obtain the functional values $y_{t,j}$ at the data point $(\alpha_{\mathbf{u}}^t, \alpha_p^t)$ via the solutions from the high fidelity full model, where $t \in \{1, 2, \dots N^t\}$ and $j \in \{1, 2, \dots P\}$;

(d) Use the data points $(\alpha_{\mathbf{u}}^t, \alpha_p^t)$ as the input data for (11) and the functional values as the output for (11), and train the LSTM deep network.

(e) After training the LSTM network, treat the network as a function, that is, $\alpha_j^{t+1} = f_j(\alpha_{\mathbf{u}}^t, \alpha_p^t)$

After obtaining the function f_j, it can then be used to predict the POD coefficients at current time level. The procedure of online prediction using the NIROM can be summarised as following,

Copyright © 2014 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

Algorithm 1: Online NIROM calculation algorithm

(1) Initialisation.
for $j = 1$ *to* P **do**
Initialize $\alpha^0_{\mathbf{u},j}$ and $\alpha^0_{p,j}$;
endfor

(2) Calculate solutions at current time step:
for $t = 1$ *to* N^t **do**
for $j = 1$ *to* P **do**

(i) Evaluate the function f at the previous time step $t-1$ by using the complete set of POD coefficients $\alpha^{t-1}_{u,j}$ and $\alpha^{t-1}_{p,j}$:

$$f_{\mathbf{z},j} \leftarrow (\alpha^{t-1}_{\mathbf{u}}, \alpha^{t-1}_{p}), \tag{11}$$

(ii) Calculate the POD coefficients $\alpha^t_{\mathbf{u}}$ and α^t_p at the current time step t using the following equations:

$$\alpha^{t+1}_j = f_j(\alpha^t_{\mathbf{u}}, \alpha^t_p) \tag{12}$$

endfor

Obtain the solutions $\mathbf{u}^t$ and p^t on the full space for current time step t by projecting $\alpha^t_{\mathbf{u},j}$ and $\alpha^t_{p,j}$ onto the full space.

$$\mathbf{u}^t = \sum_{j=1}^{P} \alpha^t_{\mathbf{u},j} \Phi_{\mathbf{u},j}, \quad p^t = \sum_{j=1}^{P} \alpha^t_{p,j} \Phi_{p,j},$$

.
endfor

Copyright © 2014 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

3. NUMERICAL SIMULATIONS

In this section two numerical simulations, namely a flow past a cylinder and a gyre, were conducted. The first numerical experiment shows the novel NIROM is capable of predicting the flows, and the second experiment compares the results with those obtained from NIROM based on POD/RBF. In all experiments, the presented novel NIROM based on deep learning was implemented under the framework of an adaptive mesh unstructured finite element computational fluid dynamics tool Fluidity [60]. In this work, the Lapack and Keras libraries are used to perform the SVD and LSTM [62, 63]. Fluidity provided the exact solutions for model comparison as well as the snapshots for the ROM POD basis functions generation. These Fluidity solutions, when mapped into the reduced space, also provided the training data for the deep learning approach.

The root mean square error (RMSE) and correlation coefficient are used to evaluate the model performance, *i.e.*,

$$RMSE^t = \sqrt{\frac{\sum_{i=1}^{F}(\psi_i^t - \psi_{o,i}^t)^2}{F}}, \tag{13}$$

where ψ_i^t and $\psi_{o,i}^t$ denote the ROM (mapped onto the full mesh) and full model solution at the node i and the time level t, respectively, and F represents the number of nodes on the full mesh.

The correlation coefficient (CC) is computed for each time step, and is defined for given expected values μ_{ψ^t} and $\mu_{\psi_o^t}$ and standard deviations σ_{ψ^t} and $\sigma_{\psi_o^t}$,

$$CC(\psi^t, \psi_o^t)^t = \frac{cov(\psi^t, \psi_o^t)}{\sigma_{\psi^t}\sigma_{\psi_o^t}} = \frac{E(\psi^t - \sigma_{\psi^t})(\psi_o^t - \sigma_{\psi_o^t})}{\sigma_{\psi^t}\sigma_{\psi_o^t}}. \tag{14}$$

Copyright © 2014 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

3.1. Ocean gyre simulation

In the first numerical example, the NIROM based on POD and LSTM was applied to the gyre problem. The ocean gyre is circular ocean currents driven by the wind, and the forces are created by the rotation of Earth. It transports water to long distances and helps develop the large scale mixing of the ocean, transport salt and chemicals, and adjust the weather and ecosystems. The eddies caused by gyres also influence the shipping routes and damage the oil platforms. In this experiment, a NIROM for the gyre was set up in order to predict the fluid dynamics in real-time. The gyre has a computational domain of 1000×1000 km and has a depth of 500m. The solution's free surface is driven by the wind with a force strength given by the expression,

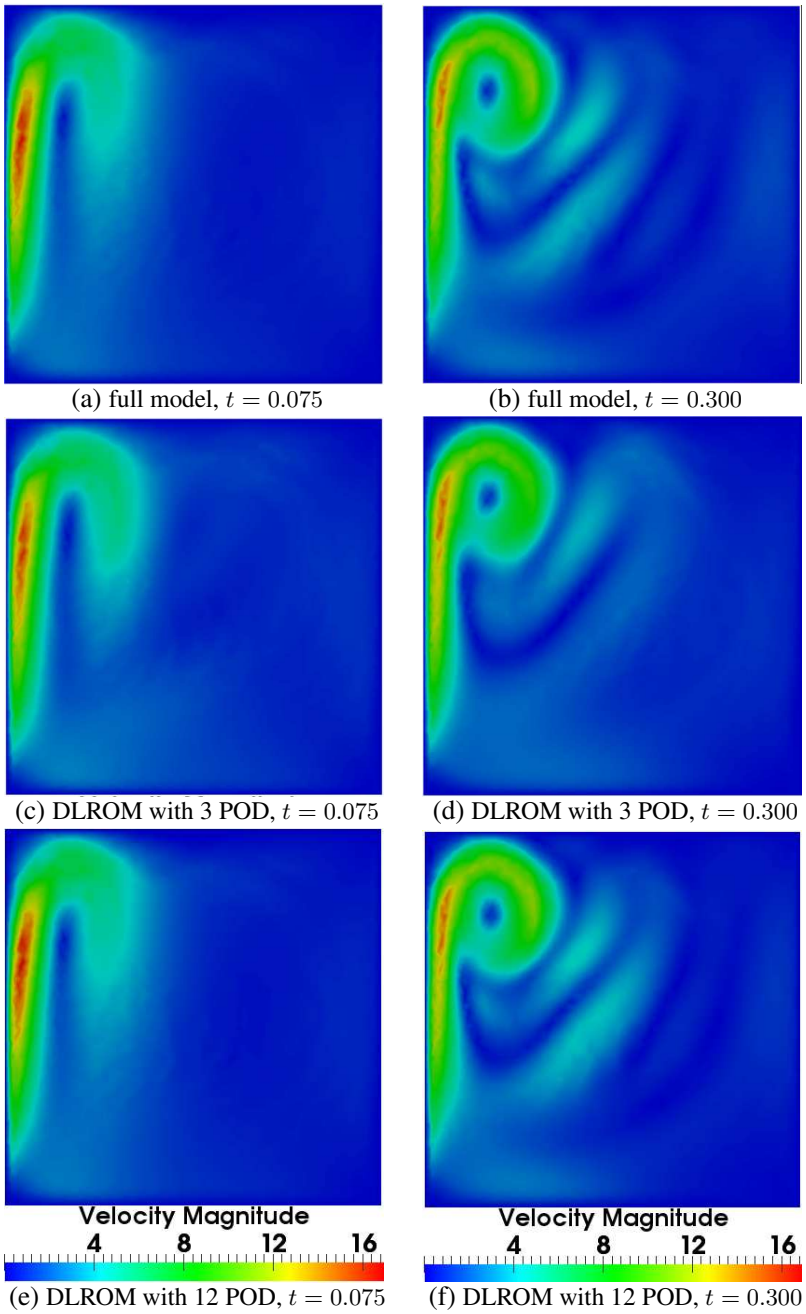
$$\tau_y = \tau_0 cos(\pi y/L) \text{ and } \tau_x = 0.0, \tag{15}$$

where L is the problem's length scale given by $L = 1000\ km$. The τ_x and τ_y are the wind stresses on the free surface along the x and y directions respectively. In the example, the maximum zonal wind stress is $\tau_0 = 0.1\ Nm^{-1}$ in the latitude (y) direction. The Coriolis terms are taken into account with the beta-plane approximation(f=βy) where $\beta = 1.8 \times 10^{-11}$. The reference density of the fluid ρ_0 is set to $1000\ kgm^{-1}$ and the Reynolds number was set to be $R_e = 250$. The gyre was simulated through the full finite element model for a period of non-dimensional unit of [0-0.21] using an non-dimensional time step size of $\Delta t = 0.001$. From this simulation 42 snapshots were taken at a regular time interval $\Delta t = 0.005$. The snapshots contribute the training data for the LSTM deep network. Then the NIROM predicts the solution further to 0.3 to investigate the capability of the NIROM. In order to compare the results between the high fidelity full model and the NIROM, the full model was simulated to obtain 60 snapshots over the simulation period [0,0.3]. The training data uses 70% of the data from the full model and uses the remaining 30% to do the comparison. In order to avoid overfitting, the dropout method was used [64]. The dropout was added to the input and recurrent connections.

Figure 3 shows the velocity solutions (a combination of two directions: $\mathbf{u} = \sqrt{\mathbf{u}_x^2 + \mathbf{u}_y^2}$) of the gyre problem at time instances 0.075 and 0.3. The solutions compare the solutions between the full model and deep learning ROM (DLROM) using 3 and 12 POD basis functions. It can be seen from the figure that NIROM using both 3 and 12 POD basis functions predict well, and it performs better when choosing larger number of POD basis functions. In order to further investigate the accuracy of the NIROM, the solutions at a particular point in the computational domain $(x = 0.19682, y = 0.91654)$ is presented in figure 4. This point is a relatively sensitive point, therefore, it is easy to show the difference of solutions between the full model and ROMs. It shows again that the NIROM performs well using both 3 and 12 POD basis functions, and the NIROM using 12 POD basis functions gets closer agreements with the exact solutions. The errors of the NIROMs is validated by RMSE and correlation coefficient, which consider errors from all the nodes in the computational domain, see figure 5.

Copyright © 2014 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld



(a) full model, $t = 0.075$ (b) full model, $t = 0.300$

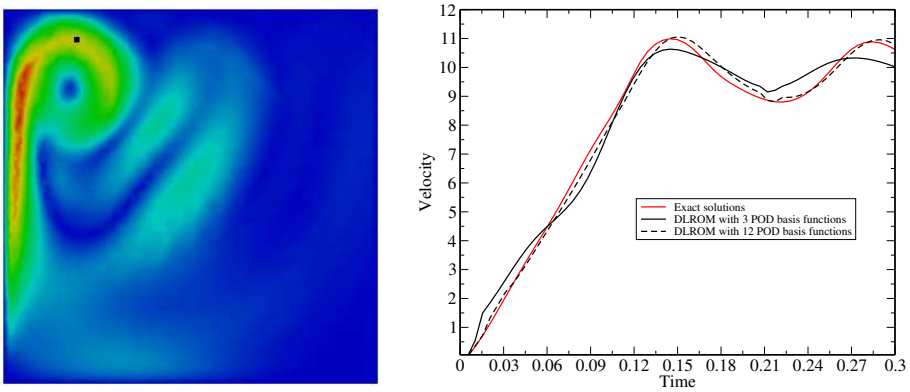
(c) DLROM with 3 POD, $t = 0.075$ (d) DLROM with 3 POD, $t = 0.300$

(e) DLROM with 12 POD, $t = 0.075$ (f) DLROM with 12 POD, $t = 0.300$

Figure 3. The figures displayed above show the velocity solutions (a combination of two spatial directions) of the gyre problem at time instances 0.075 and predicted 0.3. The solutions compare the solutions of full model and predictions from DLROM using 3 and 12 POD basis functions.

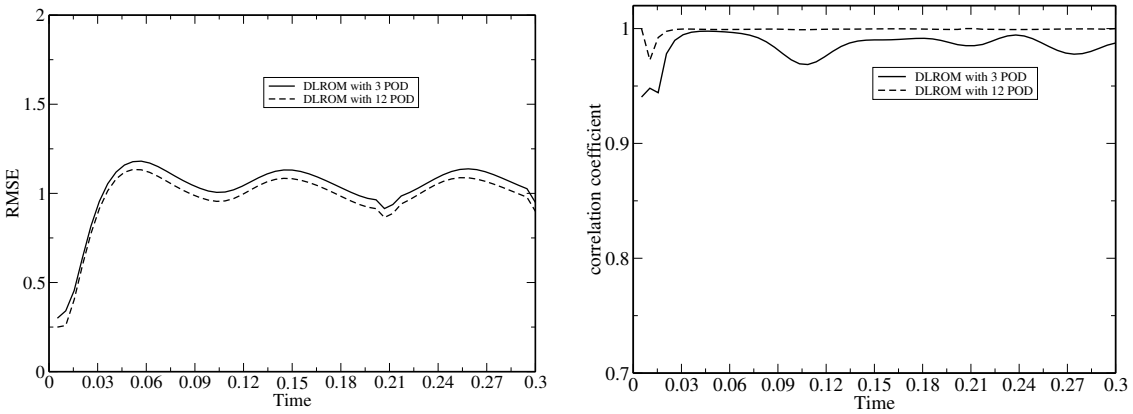
Copyright © 2014 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld



(a) location of the point (b) solutions comparison

Figure 4. Gyre case: the graphs shows the solution velocities obtained from the full model, DLROM with 3 POD basis functions and 12 POD basis functions at a point in figure (a) : (x=0.19682, y=0.91654).



(a) RMSE (b) Correlation coefficient

Figure 5. Gyre case: the figures displayed above show the RMSE and correlation coefficient of DLROM with 3 POD basis functions and 12 POD basis functions of the gyre case.

Copyright © 2014 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

3.2. *Flow past a cylinder*

In the second numerical example, a flow past a cylinder was simulated. The problem domain is non-dimensional 50 units in length and non-dimensional 10 units in width, and it possesses a cylinder of radius non-dimensional 3 units positioned over the point (5,5). Figure 6 shows the computational domain, with a mesh of 3213 nodes.

The simulation period is non-dimensional unit of 5.6 units, and for all models a time step size of $\Delta t = 0.01$ was used. 280 snapshots were obtained at regularly spaced time intervals for each of the velocity and pressure solution variables. The NIROM is used to predict the simulation period [5.7, 8] units. In order to compare the results between the high fidelity full model and the NIROM, the full model was simulated to obtain 400 snapshots over the simulation period [0,8]. Thus, 70% of the data from the full model is used as the training data, while the remaining 30% is used for cross-validation/comparison. In order to avoid overfitting, the dropout was also added to the input and recurrent connections.

The dynamics of the fluid flow is driven by an in-flowing liquid with velocity 1 unit, and this enters the domain through the left boundary and passes right boundary. No slip and zero outward flow conditions are applied to the upper and lower edges of the problem whilst the Dirichlet boundary conditions are applied to the cylinder's wall. In this example, the NIROM using LSTM deep learning method is compared with the NIROM using RBF interpolation methods, which is a popular method used in the literature [42, 43, 44].

Figure 7 shows the predicted velocity solutions of the flow past a cylinder problem at time instances 4.0 and 7.6. The solutions compare the solutions of full model and predictions from ROM/RBF and DLROM using 18 and 36 POD basis functions. As shown in this figure, the ROM based on LSTM using 36 POD performs better than that using 18 POD basis functions. At some areas in the computational domain, see figure (7 (f) and (h)), the DLROM performs better than ROM/RBF. In order to see the differences between different models, the solutions at two particular points (labelled 1055 and 2556 respectively in figure 6) are given in figure 8. This figure further shows the DLROM has a better predictive capability than ROM/RBF, especially in the simulation period [5.7, 8] units. The errors for DLROM and ROM/RBF are compared using RMSE and correlation coefficients, which consider all the nodes in the computational domain and all the time levels. The error in figure 9 shows that both the DLROM and ROM/RBF are acceptable, but DLROM shows better predictive capabilities.

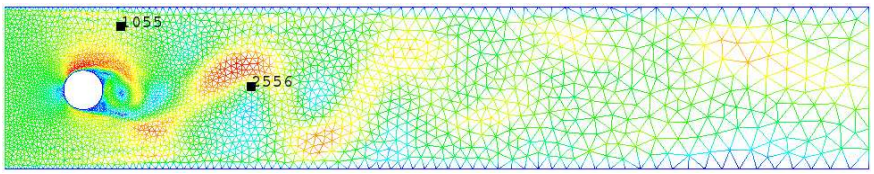


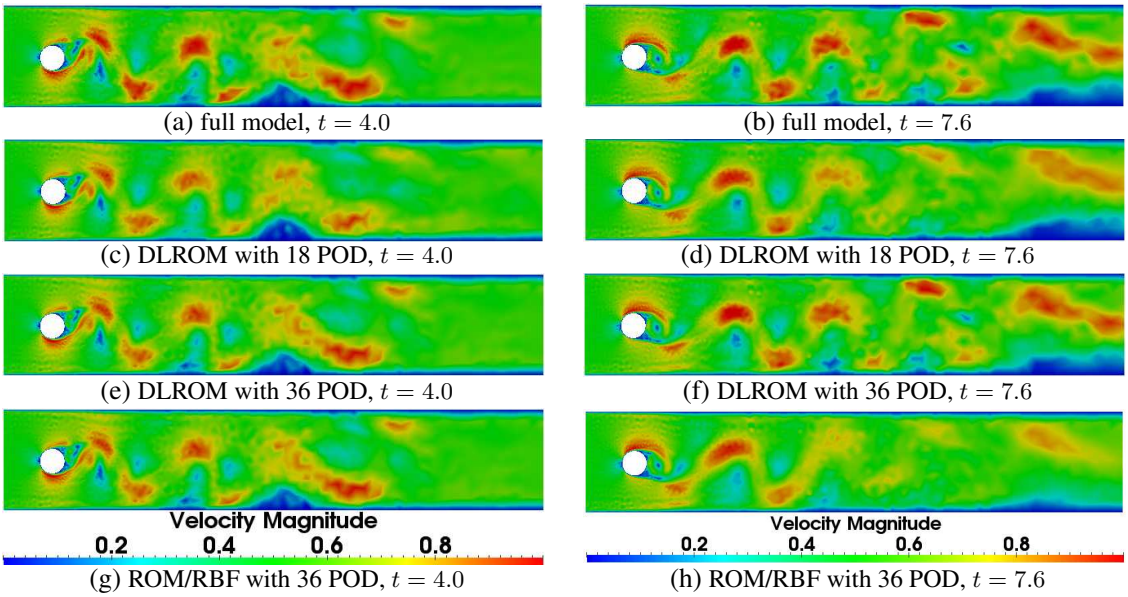
Figure 6. The graph shows the computational domain with a mesh of 3213 nodes.

As for the computational efficiency, it includes includes both offline and online CPU cost. The offline CPU cost includes the process of forming POD basis functions and training process of LSTM deep network. The offline CPU cost is given in Table I. The training CPU cost of the network is expensive, however, it can be worth investing such a time to construct the DLROM. Moreover, once the DLROM is constructed, the simulation is normally very fast - the training time is much less than the simulation time of the full model in complicated realistic cases. In addition, this offline process is precomputed and only calculated once. The online CPU time includes the prediction time and the time projecting back to the full system, which are needed for every time step. The simulations were carried out on a computer with 4 cores ($Intel_R Core^{TM}$ i7-3537U CPU @ 2.00GHz 4) and 8GB RAM. Only one of the cores was used when running the examples.

Table II shows the online CPU cost (non-dimensional unit) required for running the high fidelity full model and DLROM for one snapshot. It shows that the CPU time of DLROM is reduced by

Copyright © 2014 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld



(a) full model, $t = 4.0$ (b) full model, $t = 7.6$

(c) DLROM with 18 POD, $t = 4.0$ (d) DLROM with 18 POD, $t = 7.6$

(e) DLROM with 36 POD, $t = 4.0$ (f) DLROM with 36 POD, $t = 7.6$

(g) ROM/RBF with 36 POD, $t = 4.0$ (h) ROM/RBF with 36 POD, $t = 7.6$

Figure 7. The figures displayed above show the velocity solutions (a combination of two spatial directions) of the flow past a cylinder problem at time instances 4.0 and predicted 7.6. The solutions compare the solutions of full model and predictions from ROM/RBF and DLROM using 18 and 36 POD basis functions.

three orders of magnitude. It is worth mentioning that the ROM speed up, over the full model, can be greater the if the number of nodes in the computational domain is large.

Table I. Offline CPU cost required for calculating POD basis functions and deep learning training time

Number of POD bases	12	18	36	nodes	training time	snapshots
Gyre	0.698	0.728	0.8839	2823	630	60
Number of POD bases	12	18	50	nodes	training time	snapshots
Flow past a cylinder	48.7	50.38	83.38	3213	1273	400

Table II. Comparison of the online CPU cost (non-dimensional unit) required for running the high fidelity full model and DLROM for one snapshot.

Cases	Model	Assembling and Solving	Projection	prediction	Total
Gyre	Full model	3.0677	0	0	3.0677
	DLROM	0	0.0003	0.0001	0.00040
Flow past	Full model	3.1267	0	0	3.1267
a cylinder	DLROM	0	0.0003	0.0001	0.00040

4. DISCUSSION AND CONCLUSIONS

In this article, the authors for the first time used a deep learning method (LSTM) together with POD methods to construct a reduced order model. The DLROM is non-intrusive, which means it is independent of the source code of the full physical system and easy to modify and extend to other complex problems such as real time response of the natural hazards. The method has been tested using two numerical flow problems, flow past a cylinder and a wind driven ocean gyre. The underlying full model is an unstructured mesh finite element model (Fluidity). The results

Copyright © 2014 John Wiley & Sons, Ltd.

Prepared using fldauth.cls
DOI: 10.1002/fld

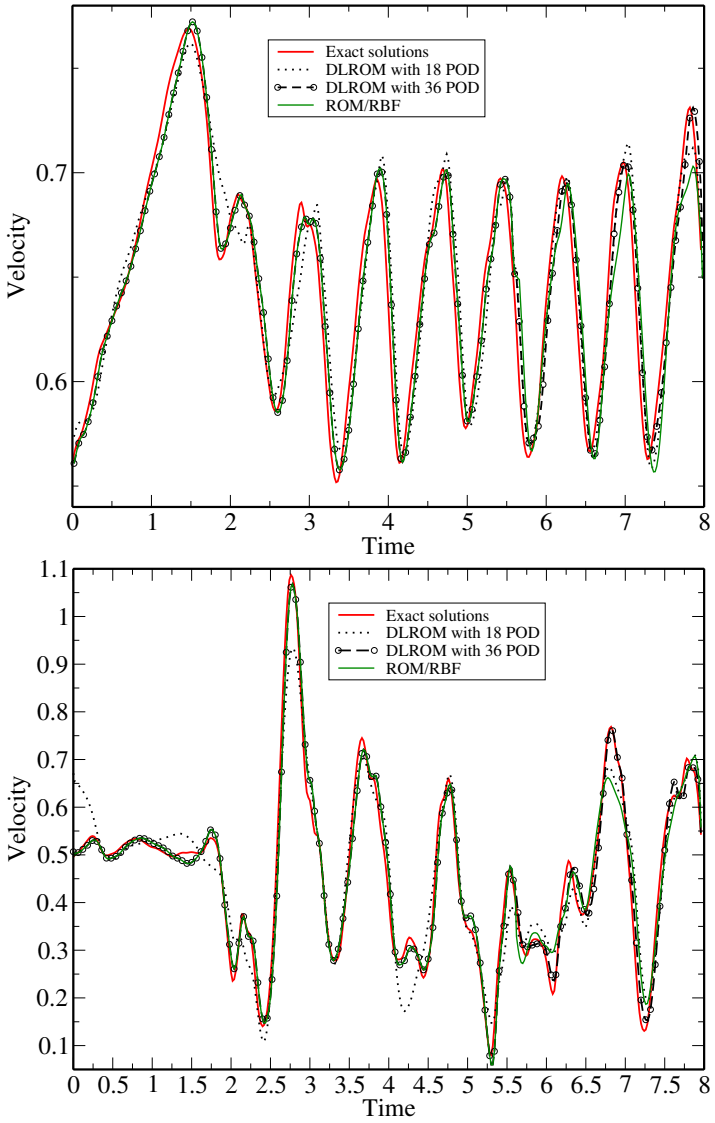


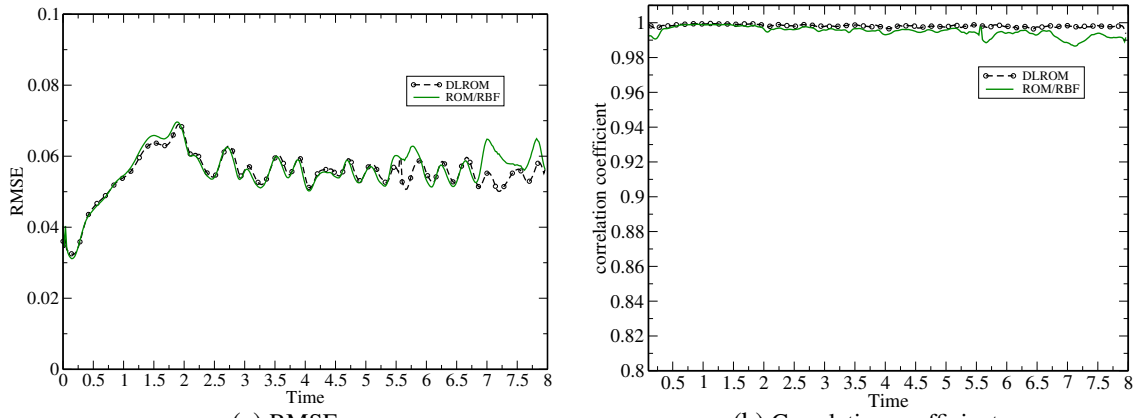
Figure 8. The graphs shows the solution velocities predicted by the full model, DLROM with 18 POD basis functions, DLROM with 36 POD basis functions and ROM/RBF models at positions (1055): (x=0.29608, y=0.36) and (2556): (x=0.62679, y=0.20903).

show that the DLROM is capable of capturing the complex fluid dynamics with less than 1/1000 CPU cost. The DLROM is also compared with the high fidelity full physical model and the now popular ROM/RBF model. The comparison shows that the DLROM has potentially better predictive capabilities. An error analysis has also been carried out for the validation and accuracy assessment of the DLROM through RMSE and correlation coefficient.

It is worth pointing out that it is difficult to choose the best model identification method for reduced order model because of various criteria [1, 65]. In this paper, we discuss the different model identification methods in terms of the temporal behaviour criteria. The model identification method using strong constraint 4D-Var is capable of predicting the dynamics of the system after the data assimilation [1]. The radial basis function interpolation based model identification method for ROM has an advantage of predicting the varying parameter problems such as varying initial and boundary conditions. However, the temporal behaviour prediction is not as good as machine learning methods. Machine learning methods have tremendous potential in data-driven engineering

Copyright © 2014 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld



(a) RMSE (b) Correlation coefficient

Figure 9. case two: the figures displayed above show the RMSE and correlation coefficient of ROM/RBF and DLROM for flow past a cylinder case.

problems. Genetic programming is especially promising since it optimises both the parameters and structure associated with the system [2]. Neural network method is capable of approximating most input-output functions, but it is prone to overfitting. The neural network method needs tremendous amounts of training data if better solutions are expected. Deep learning approach has tremendous potential in the challenging complex problems such as turbulence modelling, and control and transient dynamics. In regards to the application of deep learning to transient dynamics, the time step δt needs to be sufficiently small to provide sufficient snapshots training data (tiny changes of the fluids). The recursive dynamic model decomposition (RDMD) method can also be combined with the deep learning approach to improve the capability of capturing the transient dynamics. Like POD method, RDMD keeps orthonormal modes. In addition to orthogonality, it also includes purer frequency content [66, 67].

The ROM with machine learning methods is suitable for various modelling applications such as ocean modelling, climate modelling, air pollution and environment modelling, shape optimisation, fluids dynamics, aerospace design, neutron/photon transport problems and so on. Future work will investigate the effects of applying this model to varying parametric problems and more complex problems, such as the real time response to natural hazards (landslides, flooding etc).

ACKNOWLEDGEMENTS

The authors would like to thank the reviewers for their valuable comments and effort to improve the manuscript. The Authors acknowledge the support of the EPSRC grant: Managing Air for Green Inner Cities (MAGIC)(EP/N010221/1), the NSFC grant 11502241, funding from the European Union Seventh Framework Programme (FP7/20072013) under grant agreement NO. 603663 for the research project PEARL (Preparing for Extreme And Rare events in coastaL regions), the Newton funding: Smart technologies for optimal design, drilling, completion and management of geothermal wells (EP/R005761/1) and the EPSRC MEMPHIS multi-phase flow programme grant (EP/K003976/1).

Copyright © 2014 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

REFERENCES

1. Laurent Cordier, Bernd R Noack, Gilles Tissot, Guillaume Lehnasch, Joël Delville, Maciej Balajewicz, Guillaume Daviller, and Robert K Niven. Identification strategies for model-based control. *Experiments in fluids*, 54(8):1580, 2013.
2. Thomas Duriez, Steven L Brunton, and Bernd R Noack. Machine learning control–taming nonlinear dynamics and turbulence. *Fluid mechanics and its applications (ISSN 0926-5112*, 116, 2016.
3. D. Xiao, F. Fang, C.C. Pain, and I.M. Navon. A parameterized non-intrusive reduced order model and error analysis for general time-dependent nonlinear partial differential equations and its applications. *Computer Methods in Applied Mechanics and Engineering*, 317:868 – 889, 2017.
4. Antoine Debien, Kai AFF von Krbek, Nicolas Mazellier, Thomas Duriez, Laurent Cordier, Bernd R Noack, Markus W Abel, and Azeddine Kourta. Closed-loop separation control over a sharp edge ramp using genetic programming. *Experiments in fluids*, 57(3):1–19, 2016.
5. Ruiying Li, Bernd R Noack, Laurent Cordier, Jacques Borée, Fabien Harambat, Eurika Kaiser, and Thomas Duriez. Drag reduction of a car model by linear genetic programming control. *arXiv preprint arXiv:1609.02505*, 2016.
6. Bernd R Noack and Robert K Niven. Maximum-entropy closure for a galerkin model of an incompressible periodic wake. *Journal of Fluid Mechanics*, 700:187–213, 2012.
7. David Amsallem, Matthew Zahr, Youngsoo Choi, and Charbel Farhat. Design optimization using hyper-reduced-order models. *Structural and Multidisciplinary Optimization*, 51(4):919–940, 2015.
8. Andrea Saltelli. Sensitivity analysis for importance assessment. *Risk Analysis*, 22(3):579–590, 2002.
9. Andrea Saltelli, Marco Ratto, Terry Andres, Francesca Campolongo, Jessica Cariboni, Debora Gatelli, Michaela Saisana, and Stefano Tarantola. *Global sensitivity analysis: the primer*. John Wiley & Sons, 2008.
10. Ronald L Iman and Jon C Helton. An investigation of uncertainty and sensitivity analysis techniques for computer models. *Risk analysis*, 8(1):71–90, 1988.
11. Jerome Sacks, William J Welch, Toby J Mitchell, and Henry P Wynn. Design and analysis of computer experiments. *Statistical science*, pages 409–423, 1989.
12. Wilhelmus HA Schilders, Henk A Van der Vorst, and Joost Rommes. *Model order reduction: theory, research aspects and applications*, volume 13. Springer, 2008.
13. Gabriel Dimitriu, Răzvan Ştefănescu, and Ionel M Navon. Comparative numerical analysis using reduced-order modeling strategies for nonlinear large-scale systems. *Journal of Computational and Applied Mathematics*, 310:32–43, 2017.
14. F.Fang, C.Pain, I.M. Navon, A.H. Elsheikh, J. Du, and D.Xiao. Non-linear Petrov-Galerkin methods for Reduced Order Hyperbolic Equations and Discontinuous Finite Element Methods. *Journal of Computational Physics*, 234:540–559, 2013.
15. Z Lin, D Xiao, F Fang, CC Pain, and Ionel M Navon. Non-intrusive reduced order modelling with least squares fitting on a sparse grid. *International Journal for Numerical Methods in Fluids*, 83(3):291–306, 2017.
16. F Fang, CC Pain, Ionel M Navon, and D Xiao. An efficient goal-based reduced order model approach for targeted adaptive observations. *International Journal for Numerical Methods in Fluids*, 83(3):263–275, 2017.
17. F Fang, T Zhang, D Pavlidis, C.C. Pain, AG Buchan, and I.M. Navon. Reduced order modelling of an unstructured mesh air pollution model and application in 2d/3d urban street canyons. *Atmospheric Environment*, 96:96–106, 2014.
18. Matteo Diez, Emilio F Campana, and Frederick Stern. Design-space dimensionality reduction in shape optimization by karhunen–loève expansion. *Computer Methods in Applied Mechanics and Engineering*, 283:1525–1544, 2015.
19. Manal Alotaibi, Victor M Calo, Yalchin Efendiev, Juan Galvis, and Mehdi Ghommem. Global–local nonlinear model reduction for flows in heterogeneous porous media. *Computer Methods in Applied Mechanics and Engineering*, 292:122–137, 2015.
20. Andrea Manzoni, Filippo Salmoiraghi, and Luca Heltai. Reduced basis isogeometric methods (rb-iga) for the real-time simulation of potential flows about parametrized naca airfoils. *Computer Methods in Applied Mechanics and Engineering*, 284:1147–1180, 2015.
21. David Amsallem and Charbel Farhat. Interpolation method for adapting reduced-order models and application to aeroelasticity. *AIAA journal*, 46(7):1803–1813, 2008.
22. Manal Alotaibi, Victor M. Calo, Yalchin Efendiev, Juan Galvis, and Mehdi Ghommem. Globallocal nonlinear model reduction for flows in heterogeneous porous media. *Computer Methods in Applied Mechanics and Engineering*, 292:122 – 137, 2015. Special Issue on Advances in Simulations of Subsurface Flow and Transport (Honoring Professor Mary F. Wheeler).
23. A. Corigliano, M. Dossi, and S. Mariani. Model order reduction and domain decomposition strategies for the solution of the dynamic elasticplastic structural problem. *Computer Methods in Applied Mechanics and Engineering*, 290:127 – 155, 2015.
24. J Oliver, M Caicedo, AE Huespe, JA Hernández, and E Roubin. Reduced order modeling strategies for computational multiscale fracture. *Computer Methods in Applied Mechanics and Engineering*, 313:560–595, 2017.
25. Alexander Lozovskiy, Matthew Farthing, Chris Kees, and Eduardo Gildin. Pod-based model reduction for stabilized finite element approximations of shallow water flows. *Journal of Computational and Applied Mathematics*, 302:50–70, 2016.
26. Razvan Stefanescu, Adrian Sandu, and I.M. Navon. Comparison of POD reduced order strategies for the nonlinear 2D shallow water equations. *International Journal for Numerical Methods in Fluids*, 76(8):497–521, 2014.
27. D.N. Daescu and I.M. Navon. A dual-weighted approach to order reduction in 4d-var data assimilation. *Monthly Weather Review*, 136(3):1026–1041, 2008.
28. AG Buchan, AA Calloo, MG Goffin, S Dargaville, F Fang, CC Pain, and IM Navon. A pod reduced order model for resolving angular direction in neutron/photon transport problems. *Journal of Computational Physics*, 296:138–157, 2015.

Copyright © 2014 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

29. Dunhui Xiao. *Non-intrusive reduced order models and their applications*. PhD thesis, Imperial College London, 2016.
30. Michael Schlegel and Bernd R. Noack. On long-term boundedness of galerkin models. *Journal of Fluid Mechanics*, 765:325–352, 2 2015.
31. Jan Osth, Bernd R. Noack, Siniśa Krajnović, Diogo Barros, and Jacques Borée. On the need for a nonlinear subscale turbulence term in pod models as exemplified for a high-reynolds-number flow over an ahmed body. *Journal of Fluid Mechanics*, 747:518–544, 5 2014.
32. David Amsallem and Charbel Farhat. Stabilization of projection-based reduced-order models. *International Journal for Numerical Methods in Engineering*, 91(4):358–377, 2012.
33. Leopoldo P. Franca and Sergio L. Frey. Stabilized finite element methods: Ii. the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 99(2-3):209–233, 1992.
34. S. Chaturantabut and D.C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM J. Sci. Comput*, 32:2737–2764, 2010.
35. Alireza Jafarpour Feriedoun Sabetghadam. α Regularization of the POD-Galerkin dynamical systems of the Kuramoto-Sivashinsky equation. *Applied Mathematics and Computation*, 218:6012–6026, 2012.
36. D. Xiao, F. Fang, A. G. Buchan, C.C. Pain, I.M. Navon*, J. Du, , and G. Hu. Non-linear model reduction for the Navier-Stokes equations using Residual DEIM method. *Journal of Computational Physics*, 263:1–18, 2014.
37. D. Xiao, F. Fang, J. Du, C.C. Pain, I.M. Navon, A. G. Buchan, A.H. ElSheikh, and G. Hu. Non-linear Petrov-Galerkin methods for reduced order modelling of the Navier-Stokes equations using a mixed finite element pair. *Computer Methods In Applied Mechanics and Engineering*, 255:147–157, 2013.
38. Chen Han. Blackbox stencil interpolation method for model reduction. Master's thesis, Massachusetts Institute of Technology, 2012.
39. C.Audouze, F.D.Vuyst, and P.B.Nair. Nonintrusive reduced-order modeling of parametrized time-dependent partial differential equations. *Numerical Methods for Partial Differential Equations*, 29(5):1587–1628, 2013.
40. C Audouze, F De Vuyst, and PB Nair. Reduced-order modeling of parameterized pdes using time–space-parameter principal component analysis. *International journal for numerical methods in engineering*, 80(8):1025–1057, 2009.
41. D. Xiao, F. Fang, A.G. Buchan, C.C. Pain, I.M. Navon, and A. Muggeridge. Non-intrusive reduced order modelling of the Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 293:552–541, 2015.
42. D. Xiao, F. Fang, C. Pain, and G. Hu. Non-intrusive reduced order modelling of the Navier-Stokes equations based on RBF interpolation. *International Journal for Numerical Methods in Fluids*, 79(11):580–595, 2015.
43. D Xiao, P Yang, F Fang, J Xiang, CC Pain, and IM Navon. Non-intrusive reduced order modeling of fluid-structure interactions. *Computer Methods in Applied Mechanics and Engineering*, 303:35–54, 2016.
44. D Xiao, P Yang, F Fang, J Xiang, CC Pain, IM Navon, and M Chen. A non-intrusive reduced-order model for compressible fluid and fractured solid coupling and its application to blasting. *Journal of Computational Physics*, 330:221–244, 2017.
45. D Xiao, Z Lin, F Fang, C Pain, IM Navon, P Salinas, and A Muggeridge. Non-intrusive reduced-order modeling for multiphase porous media flows using smolyak sparse grids. *International Journal for Numerical Methods in Fluids*, 83(2):205–219, 2017. fld.4263.
46. D Xiao, F Fang, C Pain, IM Navon, and 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.
47. Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
48. Tara N Sainath, Abdel-rahman Mohamed, Brian Kingsbury, and Bhuvana Ramabhadran. Deep convolutional neural networks for lvcsr. In *Acoustics, speech and signal processing (ICASSP), 2013 IEEE international conference on*, pages 8614–8618. IEEE, 2013.
49. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
50. Hui Y Xiong, Babak Alipanahi, Leo J Lee, Hannes Bretschneider, Daniele Merico, Ryan KC Yuen, Yimin Hua, Serge Gueroussov, Hamed S Najafabadi, Timothy R Hughes, et al. The human splicing code reveals new insights into the genetic determinants of disease. *Science*, 347(6218):1254806, 2015.
51. J Nathan Kutz. Deep learning in fluid dynamics. *Journal of Fluid Mechanics*, 814:1–4, 2017.
52. Mingliang Wang, Han-Xiong Li, Xin Chen, and Yun Chen. Deep learning-based model reduction for distributed parameter systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 46(12):1664–1674, 2016.
53. Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
54. Santiago Fernández, Alex Graves, and Jürgen Schmidhuber. An application of recurrent neural networks to discriminative keyword spotting. *Artificial Neural Networks–ICANN 2007*, pages 220–229, 2007.
55. Alex Graves and Jürgen Schmidhuber. Offline handwriting recognition with multidimensional recurrent neural networks. In *Advances in neural information processing systems*, pages 545–552, 2009.
56. Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.
57. Amir Efrati. Apples machines can learn too. *The Information*, 2016.
58. Werner Vogels. Bringing the magic of amazon ai and alexa to apps on aws. - all things distributed. *www.allthingsdistributed.com*, 2016.
59. Schmidhuber Jrgen. Why use recurrent networks at all? and why use a particular deep learning recurrent network called long short-term memory or lstm. *http://people.idsia.ch/ juergen/rnn.html*, 2017.
60. CC Pain, MD Piggott, AJH Goddard, F Fang, GJ Gorman, DP Marshall, MD Eaton, PW Power, and CRE De Oliveira. Three-dimensional unstructured mesh ocean modelling. *Ocean Modelling*, 10(1):5–33, 2005.
61. S. Chaturantabut. Dimension reduction for unsteady nonlinear partial differential equations via empirical interpolation methods. Master's thesis, Rice university, 2008.

Copyright © 2014 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

62. E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999.
63. Franois Chollet. keras. https://github.com/fchollet/keras, 2015.
64. Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
65. Zhu Wang, Imran Akhtar, Jeff Borggaard, and Traian Iliescu. Proper orthogonal decomposition closure models for turbulent flows: a numerical comparison. *Computer Methods in Applied Mechanics and Engineering*, 237:10–26, 2012.
66. Bernd R Noack, Witold Stankiewicz, Marek Morzyński, and Peter J Schmid. Recursive dynamic mode decomposition of transient and post-transient wake flows. *Journal of Fluid Mechanics*, 809:843–872, 2016.
67. Bernd R Noack, Konstantin Afanasiev, Marek Morzynski, Gilead Tadmor, and Frank Thiele. A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *Journal of Fluid Mechanics*, 497:335–363, 2003.

Copyright © 2014 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld