ASIA-PACIFIC JOURNAL OF CHEMICAL ENGINEERING
Asia-Pac. J. Chem. Eng. 2013; **8**: 473–482
Published online 24 September 2012 in Wiley Online Library
(wileyonlinelibrary.com) DOI: 10.1002/apj.1679

Research article

Low-dimensional modeling of linear heat transfer systems using incremental the proper orthogonal decomposition method

Chao Xu[1]* *and Eugenio Schuster*[2]

[1]*The State Key Laboratory of Industrial Control Technology, Institute of Cyber-Systems & Control, Zhejiang University, Hangzhou, 310027, China*
[2]*Department of Mechanical Engineering and Mechanics, Lehigh University, Bethlehem, PA 18015, USA*

Received 6 March 2012; Revised 31 July 2012; Accepted 31 July 2012

ABSTRACT: In this work, we propose the incremental proper orthogonal decomposition (POD) method and the recursive Galerkin projection to achieve model order reduction (MOR) for high-dimensional dynamical systems arising in the processes of heat transfer for green buildings. For MOR of high-dimensional dynamical systems, we use a batch of historic data to initially extract a sequence of POD modes and derive a low-dimensional system to approximate the high-dimensional heat transfer system. Then, we check the prediction error at every subsequent sampling moment by using the obtained POD modes. If the approximation error is larger than the pre-given threshold value, we then add the new snapshot into the collected sampling ensemble. Instead of recalculating the POD-oriented eigenvalue decomposition problem at each ensemble augmentation (which is time-consuming), the incremental POD method applies the updated singular value decomposition approach to increase the number of POD modes and adjust the shape of POD modes, and also change corresponding POD eigenvalues through a matrix rotation transformation. © 2012 Curtin University of Technology and John Wiley & Sons, Ltd.

KEYWORDS: incremental POD; heat transfer; model order reduction

INTRODUCTION

Today, with 80–90% of total worldwide energy consumption being derived from the combustion of fossil fuels,* the world is facing myriad global threats to our unique ecosystem, such as climate changes (e.g. because of carbon dioxide emissions) and environmental pollution. In the long term, an energy shortfall might become another worldwide issue within the next 50 years or so if no effort is taken to change the current energy consumption rate of fossil fuels.† The improvement of energy efficiency, especially developing the state-of-art of smart buildings, is a critical research tendency for researchers in different areas. More specifically, how to simulate heat transfer processes and improve the utility efficiency in buildings are quite important. It is well known that dynamics of the indoor climate is quite complicated, including heat transfer, air flow dynamics and moisture distribution, which can only be described by partial differential equations (PDEs).[1] The discretization of PDEs using various numerical methods (e.g.[2,3]) usually results in large sets of ordinary differential equations (ODEs). However, these high-dimensional dynamical models derived from the spatial, and eventually temporal, discretization procedures are often inappropriate for control synthesis because they are computationally costly in solving controller synthesis-related equations (e.g. Riccati equation, Lyapunov equation, etc.). One may think to tackle the high-dimensionality challenge by using coarse meshes for the domain discretization. However, lower-order models obtained by using coarse meshes may not be accurate enough to capture the essential dynamics. Therefore, model order reduction (MOR) using observational/simulation data becomes an alternative procedure to provide reduced-order models for controller synthesis. This is a typical data-driven model-reduction approach using both physical structures and observational data.

There are several MOR approaches, including the balanced truncation method, the Krylow subspace method and the proper orthogonal decomposition (POD) method. The balanced truncation is an important tool originated in the control systems community.[4] When this method is

**Correspondence to*: Chao Xu, The State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou, 310027, China. E-mail: cxu@zju.edu.cn

*World Energy Resources and Consumption, from Wikipedia, the free encyclopedia.

†Global energy scenarios to 2050 and beyond, available online at http://www.worldenergy.org/wec-geis/edc/scenario.asp

used to derive reduced-order models for high-dimensional systems, Lyapunov equations with the same order of the original systems must be solved numerically. This is a computational challenge, and it finally led to the birth of the balanced POD method, which combines both the POD and the balanced truncation method but without solving high-dimensional Lyapunov equations.[5,6] The Krylov subspace method has been used widely in fast simulations of large-scale linear systems (e.g. discretized PDEs, large-scale integrated circuits, etc.). For a detailed introduction of both the balanced truncation method and the Krylov subspace method for MOR, the reader is referred to introductory books (e.g.[7,8]).

The POD method is also known as the principal component analysis or the Karhunen–Loeve decomposition. The original concept of the POD method goes back to Pearson's work published in 1833.[9] The POD method has been widely used in understanding complex behaviors produced by many high-dimensional dynamical systems, such as fluid flows,[10–12] heat flow,[13,14] nature convection,[15] fusion plasmas,[16] microelectromechanical systems (MEMS),[17] aero-elasticity,[18] model predictive control (MPC),[19] missing point estimation (MPE),[20] and so on. However, most work concerning POD focuses on what is called the *batch* POD method, where modes are extracted from given historic observations without having the capability to incorporate new observations when they become available. A new method, called *incremental* eigenanalysis (e.g.[21]), has been proposed to enable dimension increase, and mode deformations if necessary, when new observations are available. This method has been applied successfully to the multi-media research areas, such as, pattern recognition (e.g.[22]) and visual tracking (e.g.[23]).

In addition to adaptive MOR techniques, an obvious increase of research can be observed from very recent literature. Motivated by the increasing interests in cyber-physical systems, a novel incremental POD (iPOD) method has been proposed in the work of Xu *et al.*[24] using the operator perturbation theory. In the work of Singer and Green[25] to solve the reaction–diffusion equation, adaptive POD projections are constructed from snapshots in different regions of the computational domain due to varying reaction activities. In the work of Varshney *et al.*[26], and Pitchaiah and Armaou[27] adaptive MOR approaches have been used to tackle control and optimization problems of distributed parameter systems arising in chemical engineering. More recent work in terms of adaptive MOR approaches and applications include polycrystalline materials design using statistical learning based POD,[28] particle image velocimetry data reconstruction using adaptive gappy POD,[29] and high-dimensional nonlinear mechanical systems,[30] aeronautic design using adaptive POD/singular value decomposition surrogate model.[31]

The main contribution of this paper is to construct a modified iPOD method for MOR of high-dimensional dynamical systems using the updated singular value decomposition method. A detailed computational complexity is analyzed for the proposal approach, which demonstrates a computational improvement than the standard batch POD method. Without using the mean value of the observations in this work, the proposed iPOD problem has a simpler form than that in the work of Hall and Martin,[21] and it can be extended to multi-snapshot-based iPOD schemes (i.e. rank $q(q > 1)$ iPOD). Because of this new definition, the proposed approach in this paper is straightforward to construct updates for POD modes. The proposed iPOD method is much more technically accessible than introducing more advanced techniques, such as statistical learning,[28] gappy POD[29] tools and so on. To match the iPOD method, an incremental Galerkin projection scheme is developed. This scheme computes system projection matrices recursively based on the existing matrices obtained in the previous mode extraction step. A detailed analysis of the computational complexity is carried out to illustrate the computational efficiency of the iPOD-based MOR approach.

The paper is organized as follows. In Proper Orthogonal Decomposition section, the mathematical derivations of the POD method and the snapshots method are summarized. In Rank-1 Incremental section, the formulation and numerical schemes of the iPOD method are discussed. In Model Order Reduction section, an incremental Galerkin projection method for MOR is proposed. In Arithmetic Complexity Analysis section, the computational complexity of the iPOD method is studied. In Numerical Example section, numerical simulations that validate the effectiveness of the proposed method are provided. The paper is closed by stating conclusions and future research topics in Conclusions section.

PROPER ORTHOGONAL DECOMPOSITION

We have N training samples (snapshots) $\mathbf{x}_i \in \mathbb{R}^n (i = 1, 2, \ldots, N)$, and we use $Y = (\mathbf{x}_1, \ldots, \mathbf{x}_N) \in \mathbb{R}^{n\times N}$ to denote the observations (snapshot matrix). Usually, n (the number of spatial discretization nodes) is much larger than N (the number of samples at different time instants). We introduce the covariance matrix of Y,

$$\mathscr{C} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i \mathbf{x}_i^T \tag{1}$$

The POD problem can be stated as the search of a sequence of orthogonal basis functions $\{\phi_k\}_{k=1}^{p} \left(p \leq N, \phi_i^T \phi_j = \delta_{ij}\right)$ to represent each snapshot in the observation set Y, i.e.

DOI: 10.1002/apj

$$\mathbf{x}_i \approx \tilde{\mathbf{x}}_i = \sum_{k=1}^{p} X_{ik}\phi_k, \quad \forall \mathbf{x}_i \in Y, i = 1, 2, \ldots, N \tag{2}$$

where the coefficient X_{ik} can be determined by $X_{ik} = \phi_k^T \mathbf{x}_i$. Then, the basis functions can be determined by the following approximation-error minimization problem:

$$\min_{\phi_1,\ldots,\phi_p} \frac{1}{N}\sum_{i=1}^{N}\left\| \mathbf{x}_i - \sum_{k=1}^{p} X_{ik}\phi_k \right\|_2^2 \tag{3}$$

$$\text{subject to} : \phi_k^T \phi_l = \delta_{kl} = \begin{cases} 1, & \text{if } k = l, \\ 0, & \text{otherwise} \end{cases}$$

where the constraint implies the orthonormality property of the basis vectors.

Lemma 1 [32] Given the observation data set $Y = (\mathbf{x}_1, \ldots, \mathbf{x}_N) \in \mathbb{R}^{n \times N}$ with rank $d \le \min\{n, N\}$, then the optimal solution to the minimization problem in Eqn (3) is given by the first $p(p \le n)$ eigenvectors of the following eigenvalue decomposition problem:

$$\frac{1}{N} YY^T \phi_l = \lambda_l \phi_l, \quad l = 1, \ldots, n \tag{4}$$

The approximation error can be bounded by

$$\varepsilon^2(p) = \min_{\phi_1,\ldots,\phi_p} \frac{1}{N}\sum_{i=1}^{N}\left\| \mathbf{x}_i - \sum_{k=1}^{p} X_{ik}\phi_k \right\|_2^2 \le \sum_{k=p+1}^{n} \lambda_k \tag{5}$$

Remark 1 (Method of snapshots[33]) If $n \gg N$, then it is impractical to solve the eigenvalue decomposition problem in Eqn (4). However, one can solve the following symmetric eigenvalue decomposition $\frac{1}{N} Y^T Y \psi_l = \lambda_l \psi_l, l = 1, \ldots, n$. The POD modes are given by $\phi_l = \frac{1}{\sqrt{\lambda_l}} Y \psi_l, l = 1, 2, \ldots, p$.

RANK-1 IPOD

Given a new observation $\mathbf{x} \in \mathbb{R}^n$, we can use the eigenspace model $\Phi = (\phi_1, \ldots, \phi_p)$ to give an approximation $\mathbf{x} \approx \tilde{\mathbf{x}} = \sum_{i=1}^{p} g_i \phi_i, g_i = \phi_i^T \mathbf{x}$, where the approximation error is given by

$$\mathrm{h} = \mathbf{x}_{\#} - \tilde{\mathbf{x}}_{\#} = \mathbf{x}_{\#} - \Phi \mathbf{g}, \mathbf{g} = \left(g_1, \ldots, g_p\right)^T \tag{6}$$

If the error norm $\|\mathbf{h}\|$ is large, then the observation $\mathbf{x}_{\#}$ is not well represented by eigenspace model $(\phi_1, \ldots, \phi_p)$. We need to include this new observation $\mathbf{x}_{\#}$ and update the eigenspace model.

Remark 2 Shift-window PODThe most straightforward way to achieve this is by adding the latest sample to the end of the snapshot ensemble and dropping the sample at the beginning to retain a fixed length window. Then, the POD problem is solved again for the new data ensemble to update both the eigenvalues and eigenvectors. The shift-window POD process would neglect important formation carried by the data segments in the window removed. However, those features may appear after certain period. Thus, we need to explore the iPOD method.

It is not computationally efficient to carry out repeatedly POD computations at each ensemble update. We present an iPOD method in this section. We consider the eigenvalue decomposition of the updated observation set $Y' \leftarrow (Y, \mathbf{x}) \in \mathbb{R}^{n \times (N+1)}$. First, the covariance matrix becomes

$$\mathscr{C}' = \frac{N}{N+1}\mathscr{C} + \frac{1}{N+1}\mathbf{x}_{\#}\mathbf{x}_{\#}T \tag{7}$$

Remark 3 With the use of the residues $\mathbf{x}_i - \bar{\mathbf{x}}'$ $(i = 1, 2, \ldots, N+1)$, where $\bar{\mathbf{x}}' \triangleq \frac{1}{\mathrm{N}+1}(\mathrm{N}\bar{\mathbf{x}} + \mathbf{x}_{\#})$, to replace the snapshots in Eqn (7), we have

$$\begin{aligned} \tilde{\mathscr{C}}' &= \frac{1}{N+1}\Big(\sum_{i=1}^{N}(\mathbf{x}_i - \bar{\mathbf{x}}')(\mathbf{x}_i - \bar{\mathbf{x}}')^T \\ &\quad + \left(\mathbf{x}_{\#} - \bar{\mathbf{x}}'\right)\left(\mathbf{x}_{\#} - \bar{\mathbf{x}}'\right)^T\Big)(noting\ \mathbf{x}_{N+1} = \mathbf{x}_{\#}) \\ &= \frac{N}{N+1}\tilde{\mathscr{C}} + \frac{N}{(N+1)^2}(\mathbf{x}_{\#} - \bar{\mathbf{x}})(\mathbf{x}_{\#} - \bar{\mathbf{x}})^T \end{aligned} \tag{8}$$

where $\tilde{\mathscr{C}} \triangleq \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}(\mathbf{x}_{\mathrm{i}} - \bar{\mathbf{x}})(\mathbf{x}_{\mathrm{i}} - \bar{\mathbf{x}})^{\mathrm{T}}$ and the cross terms involving $\sum_{i=1}^{N}(\mathbf{x}_i - \mathbf{x})(\mathbf{x} - \mathbf{x}_{\#})^T$ and its transpose are zero. We note that the recursive form of Eqn (8) has the same structure of Eqn (7) where the average value is not used in computing the covariance matrix.

Remark 4 Given a data sequence $\mathbf{x}_i (i = 1, 2, \ldots, N)$ and the newly collected data sequence $\mathbf{x}_{\#,j} (j = 1, 2, \ldots, \Delta\mathrm{N})$, where the integer $\Delta\mathrm{N} > 1$, we derive the covariance matrix using the following definition:

$$\begin{aligned} \tilde{\mathscr{C}}' &\triangleq \frac{1}{N+\Delta N}\sum_{\mathrm{i}=1}^{N+\Delta N}\left(\mathbf{x}_{\mathrm{i}} - \bar{\mathbf{x}}'\right)\left(\mathbf{x}_{\mathrm{i}} - \bar{\mathbf{x}}'\right)^{\mathrm{T}} \\ &= \frac{1}{N+\Delta N}\sum_{i=1}^{N}\left(\mathbf{x}_i - \bar{\mathbf{x}}'\right)\left(\mathbf{x}_i - \bar{\mathbf{x}}'\right)^T \\ &\quad + \frac{1}{N+\Delta N}\sum_{j=1}^{\Delta N}\left(\mathbf{x}_{\#,j} - \bar{\mathbf{x}}'\right)\left(\mathbf{x}_{\#,j} - \bar{\mathbf{x}}'\right)^T \end{aligned} \tag{9}$$

DOI: 10.1002/apj

Noting that $\bar{\mathbf{x}}' = \frac{1}{N+\Delta N}\left(N\bar{\mathbf{x}} + \sum_{k=1}^{\Delta \mathrm{N}} \mathbf{x}_{\#,k}\right)$, then we have

$$\tilde{\mathscr{C}}' = \frac{1}{N+\Delta N}\sum_{i=1}^{N}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T + \frac{N}{(N+\Delta N)^3} \times \left(\Delta N\bar{\mathbf{x}} - \sum_{k=1}^{\Delta N}\mathbf{x}_{\#,j}\right)\left(\Delta N\bar{\mathbf{x}} - \sum_{k=1}^{\Delta N}\mathbf{x}_{\#,j}\right)^T + \frac{1}{N+\Delta N}\sum_{j=1}^{\Delta N}\left(\mathbf{x}_{\#,j} - \frac{N\bar{\mathbf{x}}}{N+\Delta N} - \frac{\sum_{k=1}^{\Delta N}\mathbf{x}_{\#,k}}{N+\Delta N}\right) \times \left(\mathbf{x}_{\#,j} - \frac{N\bar{\mathbf{x}}}{N+\Delta N} - \frac{\sum_{k=1}^{\Delta N}\mathbf{x}_{\#,k}}{N+\Delta N}\right)^T \tag{10}$$

where we have noted that the terms involving the factor

$$\sum_{i=1}^{N}(\mathbf{x}_i - \bar{\mathbf{x}})\left(\Delta N\bar{\mathbf{x}} - \sum_{k=1}^{\Delta N}\mathbf{x}_{\#,k}\right)^T$$

and its transpose are zero. We note that the recursive form in Eqn (10) cannot be formulated as the one in Eqn (7). However, with the use of the definition in Eqn (1) proposed in this paper, the recursive form corresponding to Eqn (7) for block update ($\Delta N > 1$) becomes

$$\mathscr{C}' = \frac{N}{N+\Delta N}\mathscr{C} + \frac{1}{N+\Delta N}\sum_{j=1}^{\Delta N}\mathbf{x}_{\#,j}\mathbf{x}_{\#,j}^T \tag{11}$$

It is much neater than the recursive form in Eqn (10) and still retain the same structure of Eqn (7) if we introduce a new update black matrix $\mathfrak{X} = \left(\mathbf{x}_{\#,1}, \ldots, \mathbf{x}_{\#,\Delta N}\right)^T$.

The normalized residue vector $\hat{h}$ is a candidate to expand the original eigenspace generated from the observation set $Y \in \mathbb{R}^{n\times N}$:

$$\hat{h} = \begin{cases} \frac{\mathbf{h}}{\|\mathbf{h}\|_2}, & \text{if } \|\mathbf{h}\|_2 > \eta, \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

where η is a small threshold value. A rotation matrix R is used to change the subspace $(\phi_1, \ldots, \phi_p, \hat{h})$ into a solution of the eigenvalue decomposition problem of $\mathscr{C}'$ based on the new observation set Y':

$$\mathscr{C}'\left[\left(\Phi, \hat{h}\right)R\right] = \left[\left(\Phi, \hat{h}\right)R\right]\Lambda' \tag{13}$$

We note that this is an $n \times n$ eigenvalue decomposition problem, and we can multiply $\left(\Phi, \hat{h}\right)^T$ from the left to obtain $\left(\Phi, \hat{h}\right)^T\mathscr{C}'\left(\Phi, \hat{h}\right)R = R\Lambda'$, which is another eigenvalue decomposition problem, but it is $(p+1)$-dimensional. By noting the expression of C' in Eqn (7), we have

$$\left(\Phi, \hat{h}\right)^T\left[\frac{N}{N+1}\mathscr{C} + \frac{1}{N+1}\mathbf{x}_{\#}\mathbf{x}_{\#}T\right]\left(\Phi, \hat{h}\right)R = R\Lambda' \tag{14}$$

We define

$$\left(\Phi, \hat{h}\right)^T\mathscr{C}\left(\Phi, \hat{h}\right) = \begin{pmatrix} \Phi^T\mathscr{C}\Phi & \Phi^T\mathscr{C}\hat{h} \\ \hat{h}^T\mathscr{C}\Phi & \hat{h}^T\mathscr{C}\hat{h} \end{pmatrix} \approx \begin{pmatrix} \Lambda & 0 \\ 0 & 0 \end{pmatrix}$$

$$\Phi,\left(\hat{h}\right)^T\mathbf{x}\mathbf{x}^T\left(\Phi, \hat{h}\right) = \begin{pmatrix} \Phi^T\mathbf{x}\mathbf{x}^T\Phi & \Phi\mathbf{x}\mathbf{x}^T\hat{h} \\ \hat{h}^T\mathbf{x}\mathbf{x}^T\Phi & \hat{h}^T\mathbf{x}\mathbf{x}^T\hat{h} \end{pmatrix} = \begin{pmatrix} \mathbf{g}\mathbf{g}^T & \gamma\mathbf{g} \\ \gamma\mathbf{g}^T & \gamma^2 \end{pmatrix}, \quad \gamma \triangleq \hat{\mathbf{h}}^{\mathrm{T}}\mathbf{x}$$

to rewrite Eqn (14) as the following eigenvalue problem:

$$\left[\frac{N}{N+1}\begin{pmatrix} \Lambda & 0 \\ 0 & 0 \end{pmatrix} + \frac{1}{N+1}\begin{pmatrix} \mathbf{g}\mathbf{g}^T & \gamma\mathbf{g} \\ \gamma\mathbf{g}^T & \gamma^2 \end{pmatrix}\right]R = R\Lambda' \tag{15}$$

Remark 5 When $N \rightarrow \infty$, the incremental eigenvalue decomposition problem in Eqn (15) becomes convergent, and the newly added part can be considered as a small perturbation. Instead of introducing the error vector $\hat{h}$ (i.e. $\hat{h} = 0$) to form the new subspace $\left(\Phi, \hat{h}\right)$, a nonsingular rotation operation of the POD modes Φ, i.e. $\Phi\tilde{R}$ can give an alternative solution. The rotation transformation matrix $\tilde{R}$ can be provided by the eigenvalue problem of the matrix $\frac{N}{N+1}\Lambda + \frac{1}{N+1}\mathbf{g}\mathbf{g}^T$, which is a degenerate case of Eqn (15). Then the matrix perturbation theory[24] can be used to provide a fast (approximate) update.

MODEL ORDER REDUCTION

Let us assume that a high-dimensional linear time-invariant dynamical model

$$\frac{d\mathbf{X}(t)}{dt} = \mathbf{A}\mathbf{X}(t) + \mathbf{B}\mathbf{U}(t), \quad \mathbf{Y}(t) = \mathbf{C}\mathbf{X}(t) \tag{16}$$

is obtained by discretizing linearized PDEs spatially (such as the heat conduction process) using standard numerical methods over a physical domain, where $\mathbf{X} \in \mathbb{R}^n, \mathbf{U} \in \mathbb{R}^m, \mathbf{Y} \in \mathbb{R}^r$ and the initial conditions are stated as $\mathbf{X}(t_0) = \mathbf{X}_0$. $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are system matrices obtained from a standard spatial discretization procedure, and $\mathbf{X} \in \mathbb{R}^n$ is the state vector representing the values at the discrete nodes. Usually, the number of discrete nodes (state dimension) is much larger than the number of sensors, i.e. $n \gg r$. This is a general case while using various standard numerical methods [such as, finite difference method, finite element method (FEM), etc.] on distributed

DOI: 10.1002/apj

parameter systems to obtain discrete lumped parameter systems. For PDEs, it is technically equivalent of both the PDE-discretize-POD and the PDE-POD frameworks. Thus, starting from high-dimensional ODEs derived from PDEs using standard numerical methods in commercial software is more practical and requires less mathematics knowledge of numerical methods but just matrix operations.

We generalize the snapshot expansion in Eqn (2) to a time continuous case $\mathbf{x}(t) \approx \tilde{\mathbf{x}}(t) = \sum_{k=1}^{p} X_k(t)\phi_p$. The reduced-order model is

$$\frac{\mathrm{d}\mathcal{X}(t)}{\mathrm{d}t} = \mathcal{A}\mathcal{X}(t) + \mathcal{B}\mathcal{U}(t), \quad \mathcal{Y}(t) = \mathcal{C}\mathcal{X}(t) \tag{17}$$

where $\mathcal{X} = \left(X_1(t), \ldots, X_p(t)\right)^T \in \mathbb{R}^p, \mathbf{U} = \mathcal{U} \in \mathbb{R}^m$, $\mathcal{Y} \in \mathbb{R}^r$ and $\mathcal{A} = \Phi^T\mathbf{A}\Phi$, $\mathcal{B} = \Phi^T\mathbf{B}$ and $\mathcal{C} = \mathbf{C}\Phi$. Now, we consider the transformation with $\Phi' = \left[\Phi, \hat{\mathbf{h}}\right]R$, then

$$\begin{aligned} \mathcal{A}' &= R^T\begin{pmatrix}\Phi \\ \hat{\mathbf{h}}\end{pmatrix}\mathbf{A}\begin{pmatrix}\Phi & \hat{\mathbf{h}}\end{pmatrix}R \\ &= R^T\begin{pmatrix}\Phi^T\mathbf{A}\Phi & \Phi^T\mathbf{A}\hat{\mathbf{h}} \\ \hat{\mathbf{h}}^T\mathbf{A}\Phi & \hat{\mathbf{h}}^T\mathbf{A}\hat{\mathbf{h}}\end{pmatrix}R \\ &= R^T\begin{pmatrix}\mathcal{A} & \Phi^T\mathbf{A}\hat{\mathbf{h}} \\ \hat{\mathbf{h}}^T\mathbf{A}\Phi & \hat{\mathbf{h}}^T\mathbf{A}\hat{\mathbf{h}}\end{pmatrix}R \end{aligned} \tag{18}$$

$$\mathcal{B}' = R^T\begin{pmatrix}\Phi^T \\ \hat{\mathbf{h}}^T\end{pmatrix}\mathbf{B} = R^T\begin{pmatrix}\Phi^T\mathbf{B} \\ \hat{\mathbf{h}}^T\mathbf{B}\end{pmatrix} = R^T\begin{pmatrix}\mathcal{B} \\ \hat{\mathbf{h}}^T\mathbf{B}\end{pmatrix} \tag{19}$$

$$\begin{aligned} \mathcal{C}' &= \mathbf{C}\begin{pmatrix}\Phi & \hat{\mathbf{h}}\end{pmatrix}R = \begin{pmatrix}\mathbf{C}\Phi & \mathscr{C}\hat{\mathbf{h}}\end{pmatrix}R \\ &= \begin{pmatrix}\mathcal{C} & \mathscr{C}\hat{\mathbf{h}}\end{pmatrix}R \end{aligned} \tag{20}$$

We summarize different POD–MOR procedures as the following algorithms:

Algorithm 1: Shift-window POD–MOR

1. New observation $\mathbf{x}_{\#}$ is available, and compute the approximation error $\mathbf{h}$ on the basis of Eqn (6);
2. If $\|\mathbf{h}\| > \eta$, the snapshot matrix $Y = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ becomes $Y' = (\mathbf{x}_2, \ldots, \mathbf{x}_N, \mathbf{x}_{\#})$ with the new covariance matrix denoted by $C' = \mathbf{Var}\{Y'\}$;
3. We solve the Nth-order eigenvalue problem $\mathscr{C}'Q = Q\Lambda$, where $Q = (\mathbf{q}_1, \ldots, \mathbf{q}_N)$ and $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)$;
4. Extract the first q columns $\mathbf{q}_1, \ldots, \mathbf{q}_q$ (according to $\lambda_1 \geq \ldots \geq \lambda_q$) to form $\Phi' = (\frac{1}{\lambda_1}Y'\mathbf{q}_1, \ldots, \frac{1}{\sqrt{\lambda_q}}Y'\mathbf{q}_q)$ and $\Lambda' = \mathrm{diag}(\lambda_1, \ldots, \lambda_q)$., The truncation order q is chosen as $q = min\{q' | \frac{\sum_{k=1}^{q}\lambda_k}{\sum_{k=1}^{N}\lambda_k} \geq 1 - \varepsilon);$
5. Compute the matrix transformations

$$\mathcal{A}' = \left(\Phi'\right)^T\mathbf{A}\Phi', \quad \mathcal{B}' = \left(\Phi'\right)^T\mathbf{B}, \quad \mathcal{C}' = \mathbf{C}\Phi' \tag{21}$$

Algorithm 2: iPOD–MOR

1. New observation $\mathbf{x}_{\#}$ is available, and compute the approximation error $\mathbf{h}$ on the basis of Eqn (6);
2. If $\|\mathbf{h}\| > \eta$, the snapshot matrix $Y = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ becomes $Y' = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N, \mathbf{x}_{\#})$ with the covariance matrix C' determined by Eqn (7);
3. We solve the $(p+1)$-order eigenvalue problem in Eqn (15);
4. Compute the matrix transformations on the basis of Eqns (18)–(20) or $\mathcal{A}' = \Phi^T\mathbf{A}\Phi'$, $\mathcal{B}' = \left(\Phi'\right)^T\mathbf{B}$ and $\mathcal{C}' = \mathbf{C}\Phi'$.

ARITHMETIC COMPLEXITY ANALYSIS

Now, we study the arithmetic complexity of both the shift-window POD–MOR and the iPOD–MOR approaches. We have to compare the arithmetic flops from two aspects: (i) the flops in computing the transformed matrices; (ii) the flops in computing the POD modes.

Lemma 2 Matrix multiplication flopsGiven two matrices $\mathcal{M}_1 \in \mathbb{R}^{d_1 \times d_2}$ and $\mathcal{M}_2 \in \mathbb{R}^{d_2 \times d_3}$, then the complexity of the matrix multiplication $\mathcal{M}_1\mathcal{M}_2$ is $2d_1d_2d_3$ flops (a flop is a floating point operation[34]).

We first give the algorithm for the matrix multiplication $\mathcal{M}_1\mathcal{M}_2$:

```
(Matrix Multiplication)

(The procedure to compute $\mathcal{M} = \mathcal{M}_1\mathcal{M}_2$)
  $\mathcal{M} = 0$
  for $i = 1 : d_1$
    for $j = 1 : d_2$
      for $k = 1 : d_3$
        $\mathcal{M}(i, j) = \mathcal{M}_1(i, k)\mathcal{M}_2(k, j) + \mathcal{M}(i, j)$
      end
    end
  end
```

For the most deeply nested statement of this algorithm,

$$\mathcal{M}(i,j) = \mathcal{M}_1(i,k)\mathcal{M}_2(k,j) + \mathcal{M}(i,j)$$

there are two flops involved in the operation. In addition, the statement is executed $d_1d_2d_3$ times. Therefore, the matrix multiplication $\mathcal{M}_1\mathcal{M}_2$ requires $2d_1d_2d_3$ flops, which is denoted by $\mathscr{F}(\mathcal{M}) = \mathscr{F}(\mathcal{M}_1 * \mathcal{M}_2) = 2d_1d_2d_3$, where '*' is used to denote the operation whose computational-complexity flops are to be counted.

We multiply matrices from left to right. For example, when we compute $\hat{h}^T\mathbf{A\Phi}$, we first calculate $\mathbf{A\Phi}$ and later $\hat{h}^T(\mathbf{A\Phi})$. Now, we estimate the computational complexity in calculating Eqns (18)–(20). On the basis of Lemma 2, we know that the computational-complexity flops of $\mathbf{A\Phi}$ are $2n \times n \times p$, i.e. $\mathscr{F}(\mathbf{A} * \mathit{\Phi}) = 2pn^2$, where $\mathbf{A} \in \mathbb{R}^{n\times}$ and $\mathbf{\Phi} \in \mathbb{R}^{n\times p}$. Similarly, the computational-complexity flops of $A\hat{h}$ are $\mathscr{F}(\mathbf{A} * \hat{h}) = 2n^2$. Thus, we can obtain the complexity of computing Eqn (18) as

$$\begin{aligned}\mathscr{F}(\mathcal{A}'|_{(18)}) = {} & \mathscr{F}(\mathbf{A} * \mathbf{\Phi}) + \mathscr{F}(\mathbf{A} * \hat{h}) \\ & + \mathscr{F}\left[\hat{h}^T * (\mathbf{A\Phi})\right] + \mathscr{F}\left[\hat{h}^T * (A\hat{h})\right] \\ & + \mathscr{F}\left[\mathbf{\Phi}^T * (A\hat{h})\right] + \mathscr{F}\left[\begin{pmatrix} \mathcal{A} & \mathbf{\Phi}^T A\hat{h} \\ \hat{h}^T\mathbf{A\Phi} & \hat{h}^T A\hat{h} \end{pmatrix} * R\right] \\ & + \mathscr{F}\left\{R^T * \left[\begin{pmatrix} \mathcal{A} & \mathbf{\Phi}^T A\hat{h} \\ \hat{h}^T\mathbf{A\Phi} & \hat{h}^T A\hat{h} \end{pmatrix}\right]\right\} \\ = {} & 2n^2p + 2n^2 + 2np + 2n + 2pn + 2(p+1)^3 \\ & +2(p+1)^3 \\ = {} & 2(p+1)n^2 + 2(2p+1)n + 4(p+1)^3 \end{aligned} \tag{22}$$

Similarly, we have

$$\begin{aligned}\mathscr{F}(\mathcal{B}'|_{(19)}) &= \mathscr{F}\left(\hat{h}^T * \mathbf{B}\right) + \mathscr{F}\left[R^T * \begin{pmatrix} \mathcal{B} \\ \hat{h}^T\mathbf{B} \end{pmatrix}\right] \\ &= 2\mathrm{nm} + 2(p+1)^2\mathrm{m} \end{aligned} \tag{23}$$

and

$$\begin{aligned}\mathscr{F}(\mathcal{C}'|_{(20)}) &= \mathscr{F}(\mathbf{C} * \hat{h}) + \mathscr{F}\left[\begin{pmatrix} \mathcal{C} & \mathscr{C}\hat{h} \end{pmatrix} * R\right] \\ &= 2rn + 2r(p+1)^2 \end{aligned} \tag{24}$$

Therefore, we have the order of the complexity:

$$\mathscr{F}\left(\mathcal{A}'|_{(18)}\right) + \mathscr{F}\left(\mathcal{B}|_{(19)}\right) + \mathscr{F}\left(\mathcal{C}'_{(20)}\right) \sim O\left(n^2\right) \tag{25}$$

We can compare the complexity of Eqns (18)–(20) with that of Eqn (21)

$$\begin{aligned}\mathscr{F}(\mathcal{A}'|_{(21)}) &= \mathscr{F}\left(\mathbf{A} * \mathbf{\Phi}'\right) + \mathscr{F}\left[\left(\mathbf{\Phi}'\right)^T * \left(\mathbf{A\Phi}'\right)\right] \\ &= 2(p+1)n^2 + 2(p+1)^2 n \end{aligned} \tag{26}$$

$$\mathscr{F}\left(\mathcal{B}'|_{(21)}\right) = \mathscr{F}\left(\left(\mathbf{\Phi}'\right)^T * \mathbf{B}\right) = 2(p+1)nr \tag{27}$$

$$\mathscr{F}\left(\mathcal{C}'|_{(21)}\right) = \mathscr{F}\left(\mathbf{C} * \mathbf{\Phi}'\right) = 2rn(p+1) \tag{28}$$

and

$$\mathscr{F}\left(\mathcal{A}|_{(21)}\right) + \mathscr{F}\left(\mathcal{B}'|_{(21)}\right) + \mathscr{F}\left(\mathcal{A}|_{(21)}\right) \sim O\left(n^2\right) \tag{29}$$

Therefore, we can see that the recursive transformations in Eqns (18)–(20) can reduce computational complexity.

Let us now focus on the computation of the POD modes. Given a matrix $\mathcal{M} \in \mathbb{R}^{n\times n}$, the eigenvalue problem is $\mathcal{M}\mathbf{v} = \lambda\mathbf{v}$. There are various methods (e.g. Chapters 7 and 8 in the work of Golub and Loan[34] that numerically provide the eigenvalue pair $(\lambda_i, \mathbf{v}_i), i = 1, 2, \ldots, n$. The computational complexity of the Matlab function eig is $O(n^3)$. Thus, we can note that the eigenvalue decomposition in Eqn (15) of the incremental method has much lower computational complexity ($O((p+1)^3)$) than the eigenvalue decomposition in Eqn (4) of the shift-window method ($O(N^3)$).

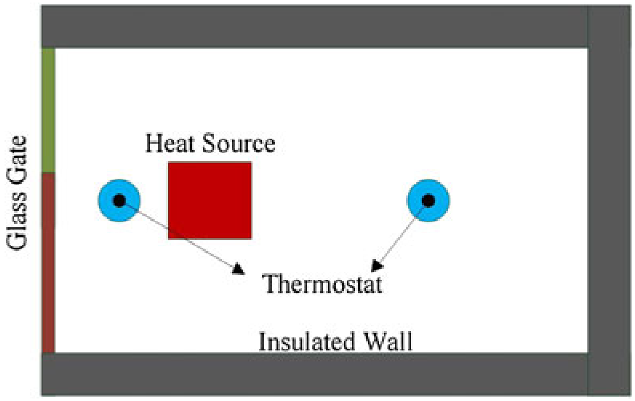

Figure 1. Schematic of a container with a heat source and two thermostats. This figure is available in colour online at www.apjChemEng.com.

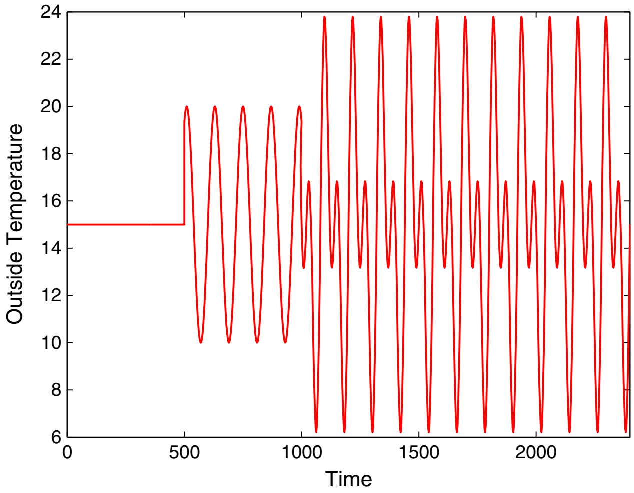

Figure 2. Outside temperature evolution. This figure is available in colour online at www.apjChemEng.com.

DOI: 10.1002/apj

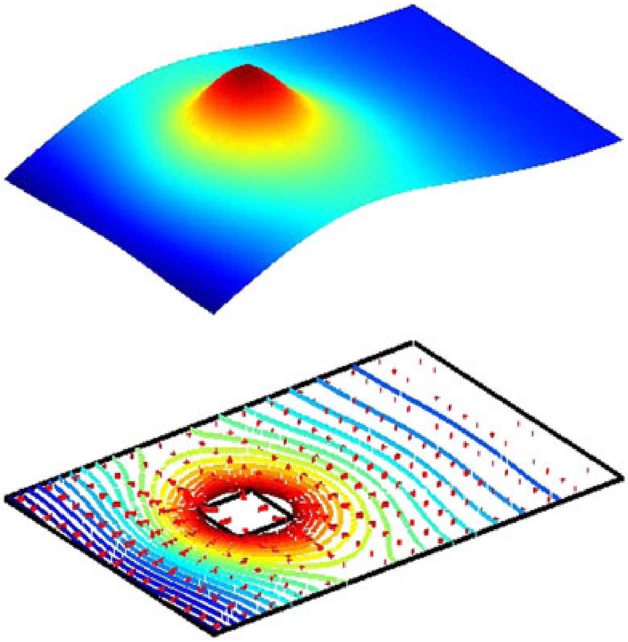

Figure 3. Simulation snapshot at $T = 2400\,\text{s}$ generated by COMSOL Multiphysics. The surface at the top is the three-dimensional plot of the temperature field while the figure at the bottom is the temperature contours within the physical domain. This figure is available in colour online at www.apjChemEng.com.

NUMERICAL EXAMPLE

As shown in Model Order Reduction section, we do not follow a classical integral-type Galerkin approach to derive finite-dimensional models from the projections of the PDEs onto chosen subspaces. We take advantage of the availability of high-dimensional discrete numerical models used for simulations and the generation of data employed for POD mode extraction. As shown earlier, we can obtain low-dimensional models by implementing matrix transformations directly on the discrete numerical models. The projection matrices carry the subspace information extracted from the simulation data. The computation of the integrals in the Galerkin projection has been embedded in various commercial software packages for the derivation of the simulation-oriented highly dimensional discrete models. Instead of obtaining the discrete model directly from weak form integrals, the combination of computational software and MOR matrix operations can save much computational burden for control engineers while dealing with complex physical systems.

This idea motivates one to create a general MOR package to connect with various PDE numerical codes (e.g. Matlab, Fluent, COMSOL Multiphysics, etc.). For the development of such a package, it is necessary to know how the various numerical codes save the discrete

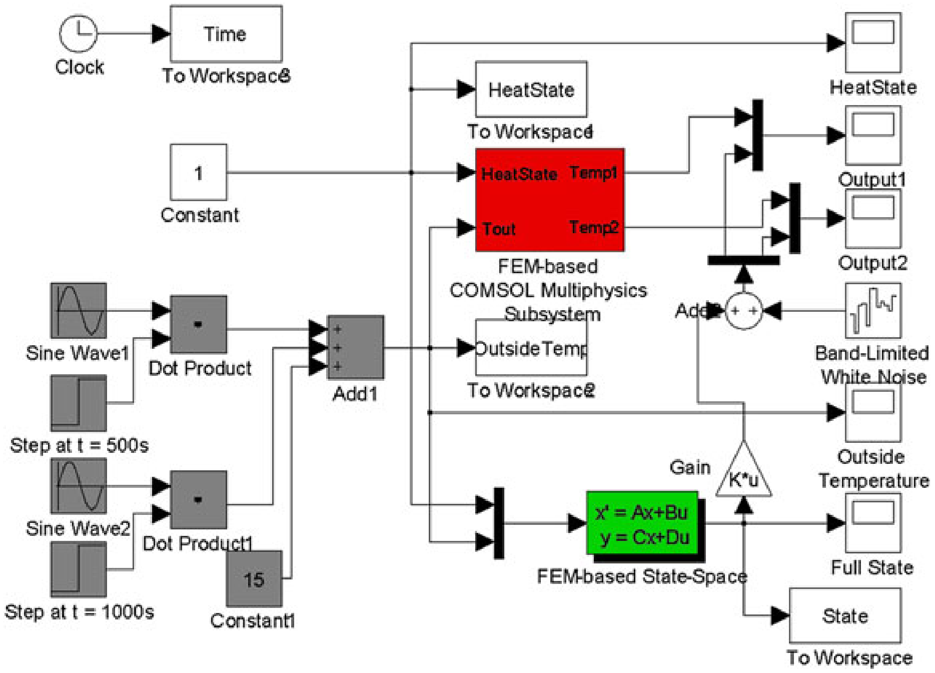

Figure 4. Simulation of the ordinary differential equations extracted from COMSOL Multiphysics using FEM. The red-shaded finite element method (FEM)-based COMSOL Multiphysics Subsystem block is exported using the COMSOL Multiphysics Simulink Model Export, whereas the green-shaded FEM-based State-space block is extracted by the COMSOL Multiphysics State-Space Model Export. This figure is available in colour online at www.apjChemEng.com.

models after implementing the spatial domain decomposition/discretization. In this work, we choose the powerful COMSOL Multiphysics numerical software to connect with our POD–MOR algorithms.

We consider a two-dimensional heat transfer problem in a container with a glass gate on the left side (as shown in Fig. 1). Only heat conduction takes place. Air flows and ventilation are not considered. Then, we can use the heat equation to model the system dynamics, i.e.

$$\rho\mathscr{C}\frac{\partial T}{\partial t} - \nabla\cdot(k\nabla T) = Q \qquad (30)$$

The boundary conditions depend on the level of insulation around the system. On well-insulated sides, the heat flux is zero, which gives the Neumann boundary condition $\mathbf{n}\cdot(k\nabla T)=0$. On poorly insulated sides, the Neumann condition is modified as $\mathbf{n}\cdot(k\nabla T) = \frac{k_g}{l_g}(T_{\text{out}} - T)$, where k_g and l_g are the thermal conductivity and the thickness of the glass sheet that separates the container and the exterior. We use the same parameters defined in the Model Library of COMSOL Multiphysics 3.5 (pages 348–359 of[35]). We assume that the heat source is kept open, but the outside temperature is time-varying. We consider an outside temperature evolution, shown in Fig. 2, of the following form

$$T_{\text{out}}(t) = \begin{cases} 15, & t \in [0, 500], \\ 15 + 5sin\left(\frac{\pi}{60}t\right), & t \in [500, 1000], \\ 15 + 5sin\left(\frac{\pi}{60}t\right) + 5sin\left(\frac{\pi}{30}t\right), & t \geq 1000 \end{cases}$$

Using the FEM implemented by the numerical software package COMSOL Multiphysics, we can generate the temporal-spatial evolution of this process. The snapshot at $t = 2400$ s is shown in Fig. 3. The high-dimensional FEM-based state space representation in Eqn (16) can be exported from COMSOL Multiphysics. Figure 4 compares the FEM-based model and the state space model. The FEM-based COMSOL Multiphysics Subsystem block is extracted using ‘COMSOL Multiphysics Simulink Model Export’. The FEM-based State-Space block is made up of the matrices **A**, **B** and **C**, which are obtained by using ‘COMSOL Multiphysics State-Space Export’.

For the POD mode extraction, we first use the historic sparse data generated from the numerical simulation over the time interval [0,500 s], which is evenly spaced in time of the total simulation time range [0,2400 s]. The first POD mode carries a dominant portion (greater than 90% according to the error estimate criterion in Eqn (5)) of the information in the generated simulation data. It is not surprising to have only one mode to carry a dominant portion of energy because dynamics over [0,500 s] of this simulation is comparably simple. For further prediction

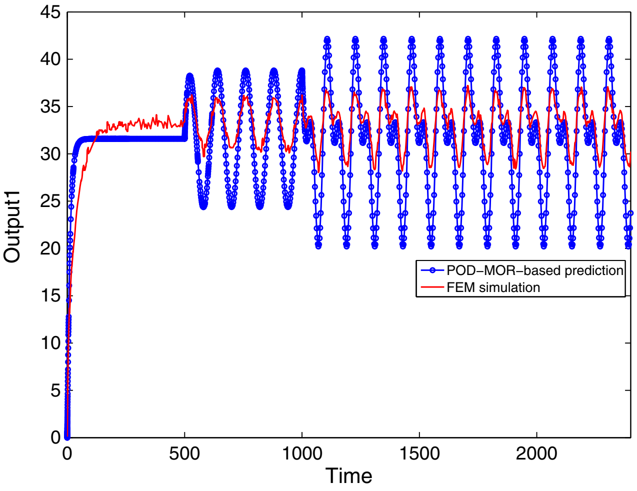

Figure 5. Temperature measured at the left thermostat. This figure is available in colour online at www.apjChemEng.com.

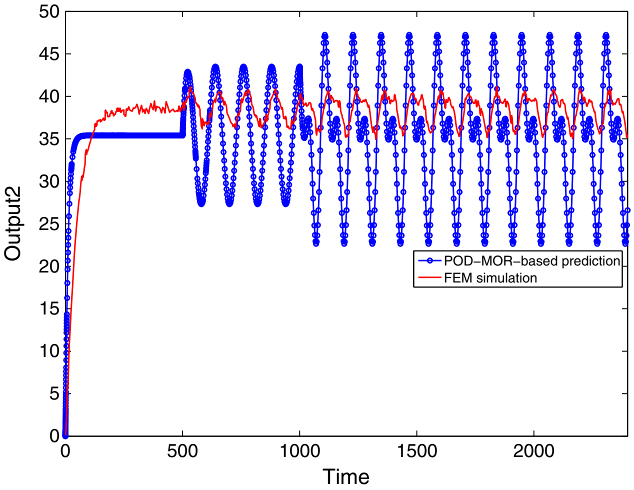

Figure 6. Temperature measured at the right thermostat. This figure is available in colour online at www.apjChemEng.com.

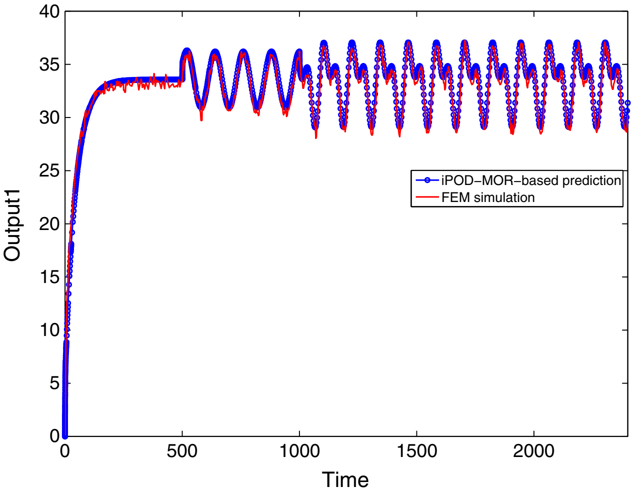

Figure 7. Temperature measured at the left thermostat. This figure is available in colour online at www.apjChemEng.com.

DOI: 10.1002/apj

with a more complex dynamics, only one mode cannot do a good job. We either need more modes or deform the mode to incorporate new features. Then, this mode is used to generate a one-dimensional model to approximate the heat conduction process. However, the error between the dynamics of both the original high-dimensional model and the reduced one-dimensional model, which is shown in Figs 5 and 6 for both thermostat temperatures, is larger than what is desired.

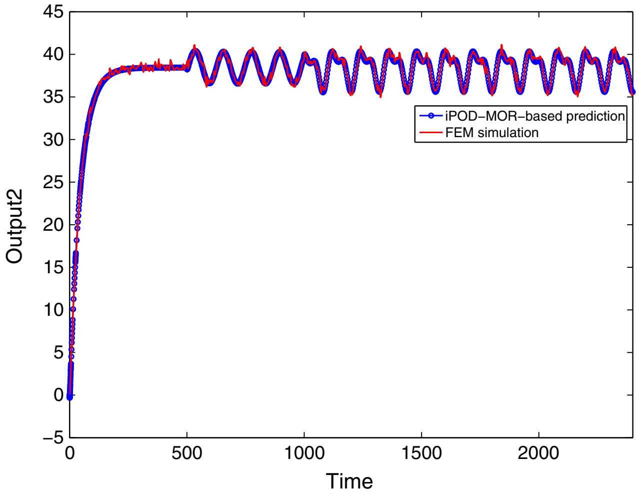

Figure 8. Temperature measured at the right thermostat. This figure is available in colour online at www.apjChemEng.com.

Because of the unsatisfactory approximation accuracy (shown in Figs 5 and 6), we need to use new simulation data to update the POD modes in an incremental way. By setting an appropriate error threshold value η, the proposed iPOD scheme can pick the poorest approximated snapshots to supplement the snapshot ensemble collected over the time interval [0, 500 s] evenly spaced in time of total simulation time range [0, 2400 s]. Two poorly approximated snapshots are chosen on the basis of the criterion in Eqn (12) and added one after another. In each update of the ensemble, an eigenvalue decomposition problem is solved to provide a rotation matrix transformation. After two subsequent iPOD computation procedures, the approximation accuracy is improved, and the output signals at the two measurement points are shown in Figs 7 and 8. The differences between the predictions by the FEM-based high-dimensional model and the iPOD-based low-dimensional model for the temperatures at the two measurement points are reduced to acceptable levels, which are much smaller than those in Figs 5 and 6. This shows the effectiveness of the MOR approach proposed in this paper. The comparisons (Figs 5 and 6 and Figs 7 and 8) are carried out using the Matlab/Simulink block diagram shown in Fig. 9 where the POD/MOR State-Space block and the iPOD/MOR State-Space block represent low-dimensional models obtained by the classic POD method and the iPOD method, respectively. In the sense of extraction accuracy of POD modes, the incremental approach converges to the batch POD one, as mentioned in Remark 5. However, the incremental approach can improve the computational efficiency, especially for a large data size.

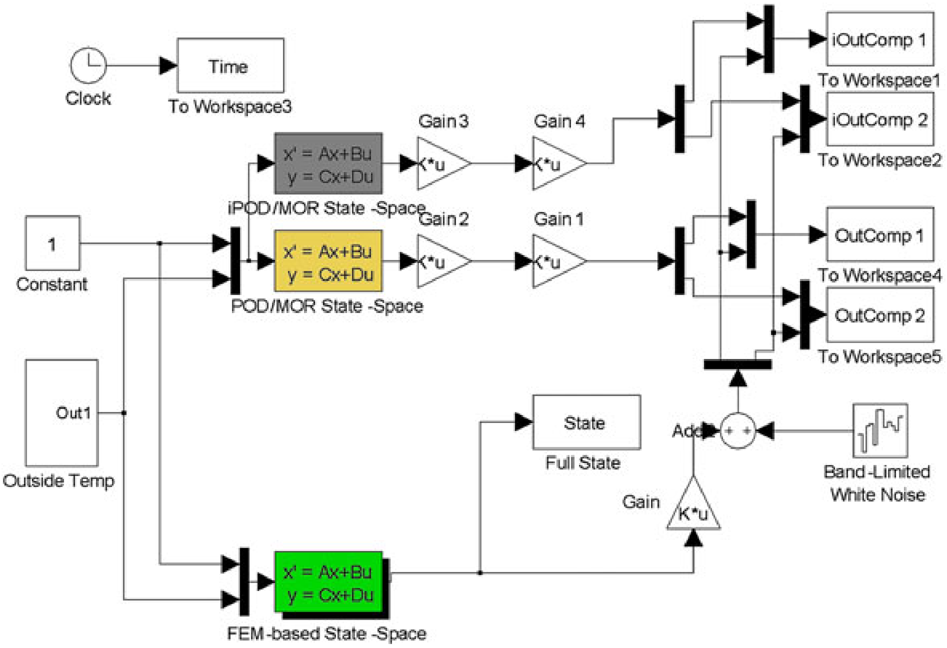

Figure 9. Simulation comparisons of reduced-order models using both the classic proper orthogonal decomposition (POD) method and the incremental POD method. This figure is available in colour online at www.apjChemEng.com.

CONCLUSIONS

An iPOD method is proposed in this paper to provide online POD mode subspace updates when new observation or simulation data become available. Instead of using the weak form of the PDEs, which requires spatial integrations over the whole physical domain, we use high-dimensional ODE models generated by simulation software through spatial discretization. Matrix transformations are used to obtain low-dimensional models. We propose a recursive matrix transformation computation method that can save significant computational effort. We provide a detailed computational analysis in this paper for the calculation of both the POD modes and the matrix transformations. A numerical simulation study based on a two-dimensional heat conduction process in a container has been used to illustrate the effectiveness of the proposed method. For the future work, we would like to extend this method to more complex cases including coupled linear PDEs for the heating, ventilation and air conditioning toward energy buildings.

Acknowledgements

This work was supported by the Fundamental Research Funds for the Central Universities (1A5000-172210101), National Natural Science Foundation of China (F030119) and Zhejiang Provincial Natural Science Foundation (Y1110354).

REFERENCES

[1] A. van Schijndel. *Bldg. Simul.*, **2009**; *2*, 143–155.
[2] J.C. Strikwerda. *Finite Difference Schemes and Partial Differential Equations*, Society for Industrial and Applied Mathematics, Philadelphia, **2004**. Available: http://epubs.siam.org/doi/book/10.1137/1.9780898717938
[3] H. Kardestuncer, D.H. Norrie (Eds). *Finite Element Handbook*, McGraw-Hill, Inc., New York, NY, USA, **1987**.
[4] B. Moore. *IEEE Trans. Autom. Control*, **1981**; *17*, 17–32.
[5] K. Willcox, J. Peraire. *AIAA J.*, **2002**; *40*, 2323–2330.
[6] C. Rowley. *Int. J. Bifurcat. Chaos*, **2005**; *15*, 997–1013.
[7] A. Antoulas. *Approximation of Large-scale Dynamical Systems*, SIAM, Philadelphia, **2005**.
[8] Y. Jiang. *Model Order Reduction Techniques*, Science Press, Beijing, **2010**.
[9] K. Pearson. *Philos. Mag.*, **1833**; *2*, 609–629.
[10] P. Holmes, J. Lumley, G. Berkooz. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, Cambridge University Press, Cambridge, UK, **1996**.
[11] L.G. Bleris, M.V. Kothare. *Comput. Chem. Eng.*, **2005**; *29*, 817–827.
[12] T. John, M. Guay, N. Hariharan, S. Naranayan. *Comput. Chem. Eng.*, **2010**; *34*, 965–975.
[13] M. Efe, H. Ozbay. Proper orthogonal decomposition for reduced order modeling: 2D heat flow. *The Proceedings of the 2003 IEEE Conference on Control Applications*, **2008**; pp. 1273–1277.
[14] L.G. Bleris, M.V. Kothare. *IEEE Trans. Control Syst. Technol.*, **2005**; *13*, 853–867.
[15] S. Sanghi, N. Hasan. *Asia-Pacific J. Chem. Eng.*, **2011**; *6*, 120–128.
[16] C. Xu, Y. Ou, E. Schuster. *Automatica*, **2011**; *47*, 418–426.
[17] Y. Liang, W. Lin, H. Lee, S. Lim, K. Lee, H. Sun. *J. Sound Vib.*, **2002**; *256*, 515–532.
[18] D.J. Lucia, P.S. Beran, W.A. Silva, R. Florea, J.J. Alonso, M. Love, R.C. Maple. *J. Aircraft*, **2005**; *42*(2), 509–518. DOI: 10.2514/1.2176.
[19] O.M. Agudelo, M. Baes, J.J. Espinosa, M. Diehl, B.D. Moor. *IEEE Trans. Automat. Control*, **2009**; *54*, 988–999.
[20] P. Astrid, S. Weiland, K. Willcox, T. Backx. *IEEE Trans. Automat. Control*, **2008**; *53*, 2237–2251.
[21] P. Hall, R. Martin. *Proc. British Machine Vision Conf.*, **1998**; *1*, 286–295.
[22] Y. Li. *Pattern Recogn.*, **2004**; *37*, 1509–1518.
[23] D. Ross, J. Lim, R. Lin, M. Yang. *Int. J. Comput. Vision*, **2007**; *77*, 125–141.
[24] C. Xu, L. Luo, E. Schuster. On the recursive proper orthogonal decomposition method and applications to distributed sensing in cyber-physical systems. *The Proceedings of the 2010 American Control Conference, 2010*, **2010**; pp. 4905–4910.
[25] M.A. Singer, W.H. Green. *Appl. Numeric. Math.*, **2009**; *59*, 272–279.
[26] A. Varshney, S. Pitchaiah, A. Armaou. *AIChE J.*, **2009**; *55* (4), pp. 906–918. [Online]. Available: http://dx.doi.org/10.1002/aic.11770
[27] S. Pitchaiah, A. Armaou. *Ind. Eng. Chem. Res.*, **2010**; *49*, 10 496–10 509.
[28] V. Sundararaghavan, N. Zabaras. *Stat. Anal. Data Mining*, **2009**; *1*, 306–321.
[29] S.G. Raben, J.J. Charonko, P.P. Vlachos. *Meas. Sci. Technol.*, **2012**; *23*(2), p. 025303. [Online]. Available: http://stacks.iop.org/0957-0233/23/i=2/a=025303
[30] P. Kerfriden, P. Gosselet, S. Adhikari, S. Bordas. *Comput. Methods Appl. Mech. Eng.*, **2011**; *200*, 850–866.
[31] T. Braconnier, M. Ferrier, J.-C. Jouhaud, M. Montagnac, P. Sagaut. *Comput. Fluids*, **2011**; *40*, 195–209.
[32] K. Kunisch, S. Volkwein. *SIAM J. Numer. Anal.*, **2002**; *40*, 492–515.
[33] L. Sirovich. *Quarterly Appl. Math., XLV*, **1987**; *42*, 561–590.
[34] G. Golub, C.V. Loan. *Matrix Computation*, 2nd edn, The John Hopkins University Press, Baltimore, **1989**.
[35] www.comsol.com, Comsol Multiphysics Model Library. Comsol Version 3.5, **2008**.

DOI: 10.1002/apj