TU Delft

Delft University of Technology

A fully adaptive nonintrusive reduced-order modelling approach for parametrized time-dependent problems

Alsayyari, Fahad; Perkó, Zoltán; Tiberga, Marco; Kloosterman, Jan Leen; Lathouwers, Danny

DOI
10.1016/j.cma.2020.113483

Publication date
2021

Document Version
Final published version

Published in
Computer Methods in Applied Mechanics and Engineering

Citation (APA)
Alsayyari, F., Perkó, Z., Tiberga, M., Kloosterman, J. L., & Lathouwers, D. (2021). A fully adaptive nonintrusive reduced-order modelling approach for parametrized time-dependent problems. *Computer Methods in Applied Mechanics and Engineering*, *373*, Article 113483. https://doi.org/10.1016/j.cma.2020.113483

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 373 (2021) 113483
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

A fully adaptive nonintrusive reduced-order modelling approach for parametrized time-dependent problems

Fahad Alsayyari*, Zoltán Perkó, Marco Tiberga, Jan Leen Kloosterman, Danny Lathouwers

Delft University of Technology, Faculty of Applied Sciences, Department of Radiation Science and Technology, Mekelweg 15, Delft, 2629JB, The Netherlands

Received 16 March 2020; received in revised form 28 September 2020; accepted 2 October 2020
Available online 3 November 2020

Abstract

We present an approach to build a reduced-order model for nonlinear, time-dependent, parametrized partial differential equations in a nonintrusive manner. The approach is based on combining proper orthogonal decomposition (POD) with a Smolyak hierarchical interpolation model for the POD coefficients. The sampling of the high-fidelity model to generate the snapshots is based on a locally adaptive sparse grid method. The novelty of the work is in the adaptive sampling of time, which is treated as an additional parameter. The goal is to have a robust and efficient sampling strategy that minimizes the risk of overlooking important dynamics of the system while disregarding snapshots at times when the dynamics are not contributing to the construction of the reduced model. The developed algorithm was tested on three numerical tests. The first was an advection problem parametrized with a five-dimensional space. The second was a lid-driven cavity test, and the last was a neutron diffusion problem in a subcritical nuclear reactor with 11 parameters. In all tests, the algorithm was able to detect and include more snapshots in important transient windows, which produced accurate and efficient representations of the high-fidelity models.
© 2020 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Keywords: Proper orthogonal decomposition; Data-driven; Greedy; Time-adaptive; Locally adaptive sparse grid

1. Introduction

In science and engineering applications, dynamic models can be described by time-dependent mathematical models. Often, these models are written as parametrized partial (integro-) differential equations (PDE). Applications such as uncertainty and sensitivity analysis and design optimization require solving the equations repeatedly for different values of the PDE parameters. For complex, large-scale problems, applications of repeated evaluations demand excessive computational power and memory resources. In such cases, *model reduction* techniques are used to overcome the computational burden. Model reduction methods aim to replace the high-fidelity model with an efficient, low-dimensional reduced-order model (ROM) capturing the main dynamics of the system with a controlled level of accuracy. Model reduction methods can be classified into intrusive and nonintrusive methods. Intrusive

* Corresponding author.
E-mail address: f.s.s.alsayyari@tudelft.nl (F. Alsayyari).

https://doi.org/10.1016/j.cma.2020.113483
0045-7825/© 2020 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

methods are mainly projection-based, where the high-dimensional model is approximated by projecting the original equations onto a reduced subspace. For an overview of different projection-based approaches, see [1,2].

Intrusive approaches require access to the operator of the high-fidelity model, which can be limiting for applications where the numerical solver is closed-source or legacy, coupled multi-physics code. In these cases, nonintrusive methods are applicable where the reduced model is built using only data generated from the high-fidelity model. For this reason, they are also called data-driven methods. A class of nonintrusive methods aims at recovering part of the problem's physical structure by inferring an assumed operator from the data. In the Loewner framework [3], a reduced model is built by interpolating measurements of the transfer function in the frequency domain. This approach was extended to construct a reduced model from time-domain data [4]. However, reduced models in the Loewner framework are applied to linear, time-invariant systems (or linear PDEs). For non-parametrized PDEs, dynamic mode decomposition (DMD) [5] learns a linear operator by fitting a sequence of time snapshots data in an optimal least square sense. However, this approach cannot be directly applied to parametrized problems.

A different line of research attempts to construct a nonintrusive reduced model for general (nonlinear) parametrized PDEs without an operator inference. The PDE solver is considered as a black-box. This class of methods uses generated data to fit a model mapping a defined input space to the desired output space. Hence, they are closer to machine learning techniques. However, while machine learning methods are typically applied in settings where data are abundant, in numerical and experimental computational science and engineering applications, data are typically expensive to generate [6]. Therefore, an important challenge to overcome for nonintrusive ROM methods is to build an accurate model using minimum data size. One effective black-box ROM method adapts the projection-based proper orthogonal decomposition (POD) method to be a nonintrusive approach. This nonintrusive version starts in a similar way to the projection-based version by constructing a reduced basis space. However, instead of projecting the high-fidelity model equations onto the reduced basis space to solve for the POD coefficients, data-fit surrogate models for the POD expansion coefficients are used. Different routes can be followed to construct the models for the expansion coefficients. One can interpolate with splines [7] or, more commonly, use radial basis function (RBF) [8,9]. Additionally, neural networks can be used to learn a surrogate model for the coefficients [10]. Gaussian process regression (or kriging) is another option to build the surrogate model [11,12]. We have presented an approach using locally adaptive sparse grid and hierarchical interpolation [13], which was then applied to perform analysis of the uncertainties in a coupled multi-physics model of a nuclear reactor system [14].

Most of the work on nonintrusive ROM methods has been developed either for parametrized time-independent problems (steady-state solutions) or time-dependent non-parametrized problems. Generalizing a ROM method to address both spatiotemporal discretization as well as the parameter space is not trivial. As a direct approach, one could build a separate ROM model for each time instance of interest using any of the (steady-state) nonintrusive ROM methods. However, such an approach is computationally unfeasible for the entire discretized time series. The challenge is even more complicated if the boundary and initial conditions are also parameter and time-dependent or if the parameter space is of a high dimension. Audouze et al. (2013) [9] suggested a nonintrusive ROM approach for time-dependent PDE problems using a two-level RBF–POD technique. This approach constructs two reduced basis spaces, one for spatial basis and a second for temporal basis. The authors use a coarse grid discretization of the spatial coordinates, time, and parameter spaces to sample the high fidelity model and generate the snapshots for the POD. Chen et al. (2018) [15] extended this work to include adaptive sampling for the parameter space using an RBF error estimator based on the distance between the RBF coefficients. The adaptivity in this approach cannot easily be extended to higher dimensional parameter spaces. Xiao et al. (2017) [16] presented an approach to tackle the high dimensional parameter space challenge using (non-adaptive) sparse grid to generate the sampling points. Their approach is also based on RBF–POD, but only one reduced basis space is constructed offline while a two-level RBF interpolation is used online; the first layer generates interpolated coefficients in parameter space then a second RBF layer propagate these coefficients in time. Peherstorfer and Willcox (2016) [17] proposed a nonintrusive operator inference ROM approach that can be applied to linear systems or systems with nonlinear terms of low order polynomials. As an extension of this work, Qian et al. (2020) [18] proposed first to lift the generated data from the high-fidelity model to a quadratic form using auxiliary variables. Then apply the operator inference approach to the lifted system. However, defining the lifting maps is problem specific and requires characterization of the nonlinear term, which is an intrusive step. Guo et al. (2019) [19] proposed an approach based on Gaussian process regression models for the POD coefficients. Time is treated as a parameter, and the snapshots for the POD basis

construction are generated based on parameter and time tensorization. Recently, several studies investigated ROM approaches based on Artificial Neural Networks (ANN) [20–26]. Swischuk et al. [6] compared between different machine learning methods for POD-based ROM modelling. They found ANN to be underperforming in cases where data is scarce. Their finding is in line with ANN literature, where it has been established that successful training of the ANN model requires a minimum data size that is a multiple of the complexity of ANN structure [27,28].

In all of the studied ROM methods, one must select the time snapshots of the high-fidelity model a priori. If the snapshots are too close to each other in time, the computational burden is unnecessarily increased. On the other hand, defining coarse time intervals risks overlooking important system dynamics. This issue has been identified by the projection-based community [29–32], where an adaptive selection of the time snapshots has been proposed to improve the projection-based POD modelling. However, the selection of the snapshot is imposed based on criteria that require knowledge of the governing equations, which is unfeasible when the system's precise dynamics are unknown, such as our nonintrusive setting.

In the present work, we aim to develop a general nonintrusive approach to identify and select the snapshots for any parametrized, time-dependent system. We build on our previous work for steady-state systems where we presented an adaptive sparse grid approach combined with POD [13]. We extend the adaptivity in parameter space to the time domain. We consider time as a parameter and use our adaptive technique to choose the important snapshots both in time and parameter spaces. This approach assumes a bounded time window of interest (i.e., $t \in [0, T]$). Therefore, the reduced model has no predictive capabilities beyond the defined end time T. However, the reduced model is able to simulate the spatiotemporal evolution of the system as a function of system parameters up to the end time T. We present three numerical tests for our adaptive approach. The first is a two-dimensional linear unsteady advection problem (Molenkamp test) that has an exact solution. This problem has five input parameters to investigate. In this test, we compare between the direct (fixed time grid) method and our time-adaptive approach. The second test is a lid-driven cavity problem, which was solved as a non-parametrized model (i.e., only time was considered as a parameter). The third is a two-dimensional time-dependent neutron diffusion problem in a subcritical reactor, which was parametrized with an 11-dimensional space. This problem is challenging due to the higher dimensionality of the parameter space and the abrupt response of the system during the transient.

The remainder of this paper is organized as follows: Problem formulation is introduced in Section 2, along with a summary of the adaptive-POD algorithm and the time treatment approach. The numerical tests are presented in Section 3. Our conclusions are discussed in Section 4.

2. Adaptive-POD approach

2.1. Problem formulation

We are interested in building a reduced model for a general parametrized time-dependent problem. Due to our nonintrusive approach, the governing equations are unknown. Therefore, a general form for the problem under an unknown nonlinear operator $\mathcal{N}(\cdot)$ can be written as

$$\mathcal{N}(y(\boldsymbol{x}, t, \boldsymbol{\alpha}), \boldsymbol{x}, t, \boldsymbol{\alpha}) = s(\boldsymbol{x}, t, \boldsymbol{\alpha}), \tag{1}$$

where $y(\boldsymbol{x}, t, \boldsymbol{\alpha})$ is the solution of the system, $\boldsymbol{x}$ is the state independent variable (e.g., spatial coordinates, energy, or angular directions), t is time, $\boldsymbol{\alpha} \in \mathbb{R}^d$ is a vector of d parameters representing properties of the system (e.g., material, geometry, or boundary conditions), and $s(\boldsymbol{x}, t, \boldsymbol{\alpha})$ is a source function. We aim at building a reduced model to capture the dynamics of the solution $y(\boldsymbol{x}, t, \boldsymbol{\alpha})$ within a defined range of the parameter $\boldsymbol{\alpha}$. We assume the availability of a numerical solver for the discretized version of the problem. That is

$$\mathcal{N}(\boldsymbol{y}(t, \boldsymbol{\alpha}), t, \boldsymbol{\alpha}) = \boldsymbol{s}(t, \boldsymbol{\alpha}), \tag{2}$$

where $\boldsymbol{y} \in \mathbb{R}^n$ is a vector with n state variables. The computational burden usually scales with the dimension of the state vector n. Our approach is based on POD method, where we seek to approximate $\boldsymbol{y}(\boldsymbol{\alpha}, t)$ using an expansion of the form

$$\boldsymbol{y}(\boldsymbol{\alpha}, t) \approx \sum_{j=1}^{r} c_j(\boldsymbol{\alpha}, t)\boldsymbol{v}_j, \tag{3}$$

where $\boldsymbol{v}_j \in \mathbb{R}^n$ is the basis vector (or POD mode) and $c_j(\boldsymbol{\alpha}, t)$ is its coefficient that depends on parameter $\boldsymbol{\alpha}$ and the time (t). The POD method extracts a reduced basis space for the system using the left singular vectors of the singular value decomposition (SVD) applied to the snapshot matrix, which is a matrix containing an ensemble of solutions at different states of the system. The basis space can be truncated at the first r left singular vectors such that the truncation error is below a cut-off threshold γ_{tr}. That is

$$\frac{\sum_{j=r+1}^{n} \sigma_j^2}{\sum_{j=1}^{n} \sigma_j^2} < \gamma_{\text{tr}}, \tag{4}$$

where σ_j is the singular value of the left singular vector $\boldsymbol{v}_j$.

We have proposed in [13] an iterative algorithm to build a reduced model by adaptively selecting important points from the parameter space and updating the snapshot matrix. In this work, we propose to deal with time as a parameter. That is, we consider time to be an additional input parameter such that the solution $\boldsymbol{y}(\boldsymbol{\alpha}, t) = \boldsymbol{y}(\boldsymbol{\alpha}^*)$, where $\boldsymbol{\alpha}^* = [\boldsymbol{\alpha}^\top, t]^\top$ and the symbol $^\top$ denotes the transpose. The dimension of the parameter space becomes $d^* = d + 1$ and Eq. (3) becomes

$$\boldsymbol{y}(\boldsymbol{\alpha}^*) \approx \sum_{j=1}^{r} c_j(\boldsymbol{\alpha}^*)\boldsymbol{v}_j. \tag{5}$$

Formulating the problem in this way allows us to directly use the previously developed adaptive tool. Once the orthonormal basis is known, the coefficient values at the sampled point $\boldsymbol{\alpha}_q^*$ can be computed as

$$c_j(\boldsymbol{\alpha}_q^*) = \langle \boldsymbol{v}_j, \boldsymbol{y}(\boldsymbol{\alpha}_q^*) \rangle, \tag{6}$$

where $\langle ., . \rangle$ indicates the scalar product.

2.2. Smolyak interpolation

To compute the coefficient at any non-sampled point, we use the Smolyak iterative interpolant developed in [13]. Here, we only present a summary of the adaptive algorithm. At iteration k, the d^*-dimensional interpolant $A_{k,d^*}(c)(\boldsymbol{\alpha}^*)$ is given by

$$A_{k,d^*}(c)(\boldsymbol{\alpha}^*) = A_{k-1,d^*}(c)(\boldsymbol{\alpha}^*) + \Delta A_{k,d^*}(c)(\boldsymbol{\alpha}^*) \ , \tag{7}$$

with $A_{0,d^*}(c)(\boldsymbol{\alpha}^*) = 0$, and

$$\Delta A_{k,d^*}(c)(\boldsymbol{\alpha}^*) = \sum_{n=1}^{m_k^\Delta} w_n^k \Theta_n(\boldsymbol{\alpha}^*), \tag{8}$$

where m_k^Δ is the cardinality of the so-called important set $\mathcal{Z}^k$. The important set contains the parameter points $\boldsymbol{\alpha}^*$ at which the interpolant was found to have an error greater than a pre-defined threshold γ_{int}. In the next iteration, the algorithm refines the sampling scheme in the neighbourhood of the points in the important set. The d^*-variate basis function $\Theta_n(\boldsymbol{\alpha}^*)$ is defined for every point $\boldsymbol{\alpha}_n^* = (\alpha_{n,1}^{i_1}, \ldots, \alpha_{n,d}^{i_d}) \in \mathcal{Z}^k$ as

$$\Theta_n(\boldsymbol{\alpha}^*) = \prod_{p=1}^{d^*} a^{i_p}_{\alpha^{i_p}_{n,p}}(\alpha_p), \tag{9}$$

where $\boldsymbol{\alpha}^*$ has support nodes $= (\alpha_1, \ldots, \alpha_{d^*})$, and i_p is the level (tree depth) index along dimension p. The unidimensional interpolant $a^{i_p}_{\alpha^{i_p}_{n,p}}(\alpha_p)$ is defined as

$$\begin{aligned} a^i_{\alpha_n^i}(\alpha) &= 1 \quad \text{if } i = 1, \\ a^i_{\alpha_n^i}(\alpha) &= \begin{cases} 1 - (m^i - 1) \cdot |\alpha - \alpha_n^i|, & \text{if } |\alpha - \alpha_n^i| < \dfrac{1}{m^i - 1}, \\ 0, & \text{otherwise,} \end{cases} \end{aligned} \tag{10}$$

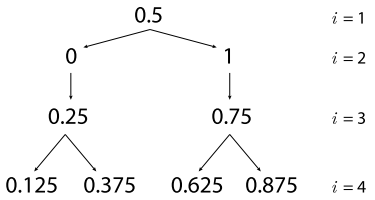


Fig. 1. Illustration of the first 4 levels of the tree structure where 0.5 is the root of the tree and nodes are added at half the distance between the previous nodes. Each node has 2 children except the nodes at level 2 where each has one child only.

and

$$m^i = \begin{cases} 1 & \text{if } i = 1, \\ 2^{i-1} + 1 & \text{if } i > 1, \end{cases} \tag{11}$$

where the dependence on the dimension p is dropped for notational convenience.

The surplus w_n^k is defined as the difference between the interpolated value and the true value of the coefficient at $\boldsymbol{\alpha}_n^*$. That is

$$w_n^k = c(\boldsymbol{\alpha}_n^*) - A_{k-1,d^*}(c)(\boldsymbol{\alpha}_n^*). \tag{12}$$

The reduced model is then built by using the interpolant of Eq. (7) as a surrogate model for the POD coefficients in Eq. (5), yielding

$$\boldsymbol{y}(\boldsymbol{\alpha}^*) \approx \sum_{j=1}^{r} A_{k,d^*}(c_j)(\boldsymbol{\alpha}^*)\boldsymbol{v}_j. \tag{13}$$

2.3. Adaptive sampling strategy

The adaptive sparse grid algorithm is based on arranging the nodes along each dimension in a tree structure, as shown in Fig. 1. Each node has two children and one father with an exception at the boundary nodes in level 2 where each has one child only. The nodes are then tensorized to form points in parameter space. To maximize the separation of points in parameter space, we choose the equidistant rule for the unidimensional nodes. Each point in parameter space has *forward* and *backward* points. The forward points for a point $\boldsymbol{\alpha} = (\alpha_1, \alpha_2 \ldots, \alpha_{d^*})$ is generated by tensorizing the children of each node with the rest of the nodes. That is, the first forward point of $\boldsymbol{\alpha}$ is $(b_1(\alpha_1), \alpha_2 \ldots, \alpha_{d^*})$, where $b_1(\alpha)$ is a function that returns the first child from the tree. The second forward point is $(b_2(\alpha_1), \alpha_2 \ldots, \alpha_{d^*})$, where $b_2(\alpha)$ is a function that returns the second child. By applying $b_1(\alpha)$ and $b_2(\alpha)$ to α_2, the third and fourth forward points are generated and so forth for the rest of the dimensions. Therefore, for any point in parameter space we can generate up to $2d$ forward points. The backward points are generated in the same manner but by using a function that returns the father of a node instead of the child function. Hence, each point has at most d backward points. By generating the backward points recursively, the set of *ancestors* are created. Note that a forward point can be shared between two different backward points. Therefore, points in parameter space do not form a classical tree structure but rather are connected as a network.

The parameter space is bounded by the defined upper and lower values for each dimension in $\boldsymbol{\alpha}^*$. This space is mapped to a unitary hypercube with dimension d^*, where 1 is mapped to the upper value and 0 represents the lower value of the range. In the initialization step ($k = 0$), the algorithm selects the central point in the hypercube and adds it to the important set $\mathcal{Z}^0$. The high-fidelity model is then sampled at that point. Then, a reduced model is built using Eq. (13). In the first iteration ($k = 1$), a trial set is generated from the forward points of the points in $\mathcal{Z}^0$. The algorithm then samples the high-fidelity model and computes the error of the reduced model at each point in the trial set. Points with an error above a pre-defined threshold (γ_{int}) are considered important points. Then, in any iteration k, the trial set is generated from the forward points of $\mathcal{Z}^{k-1}$. For each point in the trial set, if the error is found to be above γ_{int}, this point is marked as a candidate point. The algorithm then considers the ancestors of each candidate point. If all ancestors were included in $\bigcup_{l=0}^{k-1} \mathcal{Z}^l$, that candidate point is added to the important set. On the other hand, if the candidate point has an ancestor that was not included in $\bigcup_{l=0}^{k-1} \mathcal{Z}^l$, that ancestor is

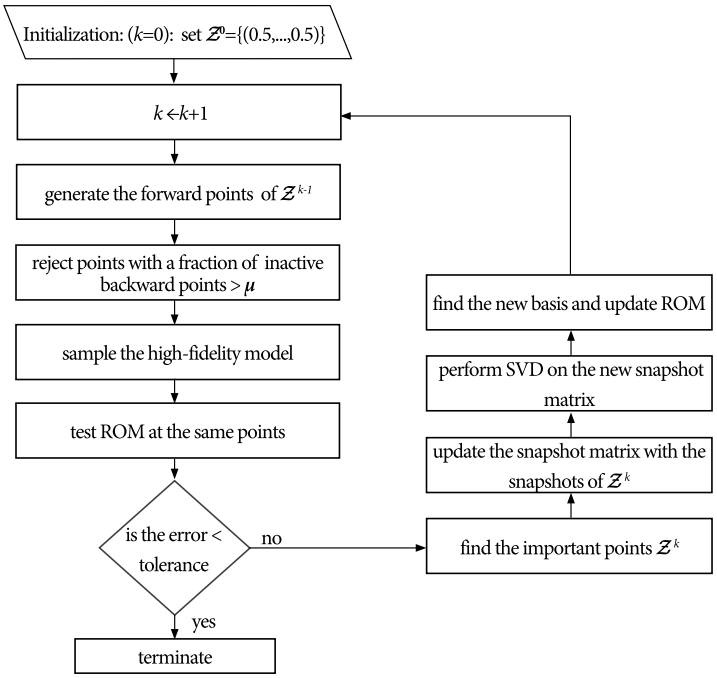


Fig. 2. A flow chart illustrating the adaptive-POD algorithm.

marked important and the candidate point is stored and tested again in the next iteration. Points that are not marked important are added to the inactive set.

To control the efficiency of the sampling scheme, a generated forward point is excluded from the trial set if it has a fraction of inactive backward points above a predefined parameter $\mu \in [0, 1]$. For $\mu = 1$, all forward points are sampled and the algorithm is more exploratory whereas for $\mu = 0$, the algorithm is more efficient by only sampling points which have all their backward points in the important set. Fig. 2 summarizes the algorithm with a flow chart. For a detailed description of the algorithm, we refer the reader to [13].

Clearly, time is not an input parameter and special attention has to be taken with such an approach. This is because numerical solvers are discretized in time. The algorithm could request a snapshot at a certain time t_l, which could be a time instance in-between the solver's default time steps. This can be addressed either by solving up to the last default time step before t_l then modifying the time step to reach t_l or interpolating between two time steps before and after t_l. Additionally, for every request of t_l, the high-fidelity model will solve for all time instances from $t = 0$ up to t_l. Management of the interface with the solver is important to avoid redundant simulations. If at one iteration, the algorithm requests $\boldsymbol{\alpha}_q$ with t_l, we can store all generated snapshots for $t < t_l$ in a library. In this manner, the algorithm can recall from this library instead of rerunning the high-fidelity solver with each $\boldsymbol{\alpha}_q$ call. Likewise, If the algorithm requests $t > t_l$ with $\boldsymbol{\alpha}_q$, the solver needs only to be restarted from t_l instead of the initial conditions $t = 0$. This strategy saves computational resources.

However, in cases where memory is limited, one might opt to store only a certain percentage of the generated snapshots. Then, restart a requested simulation from the closest stored snapshot. Note that storing all generated snapshots is not an integral part of the algorithm. The only snapshots that need to be stored are the ones marked as important and included in the snapshot matrix. Stored snapshots that are not used during the construction stage can serve as testing points for the reduced model once the algorithm is terminated. The snapshot matrix is the only memory consuming step in the algorithm. One can reduce the memory burden of the snapshot matrix with the use of an SVD updating algorithm [33] instead of the full SVD at each iteration. In our implementation for this work, however, we have used full SVD at each iteration.

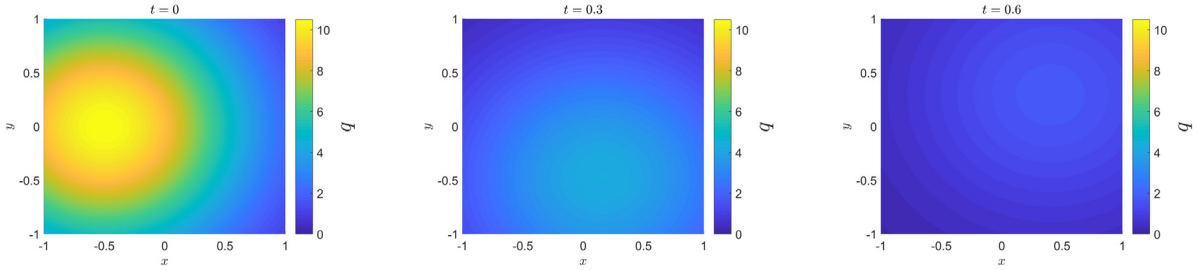


Fig. 3. Snapshots of the solution for the smooth Molenkamp problem ($\lambda_2 = 0.15$) at selected time steps ($t \in \{0, 0.3, 0.6\}$).

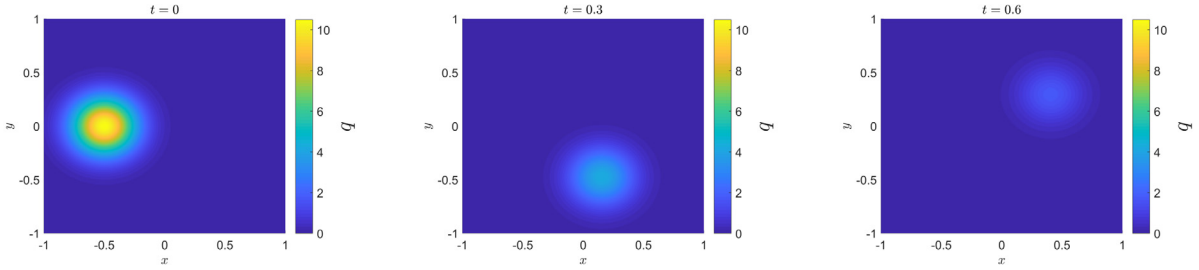


Fig. 4. Snapshots of the solution for the steep Molenkamp problem ($\lambda_2 = 3$) at selected time steps ($t \in \{0, 0.3, 0.6\}$).

We use the ℓ^2 norm to compute the relative error for any point $\boldsymbol{\alpha}_q^*$ as

$$e_q^k = \frac{\|\boldsymbol{y}(\boldsymbol{\alpha}_q^*) - \sum_{j=1}^{r} A_{k,d^*}(c_j)(\boldsymbol{\alpha}_q^*)\boldsymbol{v}_j\|_{\ell^2}}{\|\boldsymbol{y}(\boldsymbol{\alpha}_q^*)\|_{\ell^2} + \epsilon}, \tag{14}$$

where ϵ is introduced as an offset for cases when $\|\boldsymbol{y}(\boldsymbol{\alpha}_q^*)\|_{\ell^2}$ has near zero magnitude. A point $\boldsymbol{\alpha}_q^*$ is marked as a candidate point when e_q^k is above the threshold γ_{int}. The iterative algorithm is terminated when e_q^k for all points in the trial set of iteration k is below a global tolerance ζ.

3. Applications

Our proposed algorithm is tested on three time-dependent problems. The first is a two-dimensional linear unsteady advection problem that has an exact solution. This problem is also called the Molenkamp test [34]. We parametrize this problem on a five-dimensional space. The second is a lid-driven cavity test. This problem is not parametrized but tests the ability of the algorithm to detect the important transient window. The third is a challenging 11-dimensional transient nuclear reactor problem. This problem simulates a subcritical reactor with an external source.

3.1. Molenkamp test

The original Molenkamp test [34] is a two-dimensional advection problem which has an exact solution as a Gaussian cloud of material being transported in a circular path without changing its shape. However, in order to create a more challenging setting for the adaptive-POD algorithm, we modified this problem to include an additional reaction term, which in effect causes the amplitude of the solution to decay over time. The dimensionless advection–reaction equation is

$$\frac{\partial q(x, y, t)}{\partial t} + u\frac{\partial q(x, y, t)}{\partial x} + v\frac{\partial q(x, y, t)}{\partial y} + \lambda_3 q(x, y, t) = 0, \qquad (x, y) \in [-1, 1], \tag{15}$$

where the velocity field describes a solid body rotation $u = -2\pi y$ and $v = 2\pi x$. The initial condition is

$$q(x, y, 0) = \lambda_1 0.01^{\lambda_2 h(x,y,0)^2}, \qquad h(x, y, 0) = \sqrt{(x - \lambda_4 + \frac{1}{2})^2 + (y - \lambda_5)^2}. \tag{16}$$

Table 1
Range of variations for each parameter in the Molenkamp test.

Parameter	Lower bound	Upper bound
λ_1	1	20
λ_2	0.1 (2 for steep setting)	0.2 (4 for steep setting)
λ_3	1	5
λ_4	-0.1	0.1
λ_5	-0.1	0.1

The exact solution is imposed on the inflow boundary condition as

$$q(x, y, t) = \lambda_1 0.01^{\lambda_2 h(x,y,t)^2} e^{-\lambda_3 t}, \quad h(x, y, t) = \sqrt{(x - \lambda_4 + \frac{1}{2}\cos 2\pi t)^2 + (y - \lambda_5 + \frac{1}{2}\sin 2\pi t)^2}. \tag{17}$$

The exact solution is evaluated on a Cartesian uniform 100×100 grid. Therefore, the model has 10,000 degrees of freedom. Note that in this problem, evaluating the solution is computationally efficient and a reduced model is not necessary. However, this problem is selected to test the ability of the developed algorithm in capturing such dynamics.

The problem is parametrized with a 5 dimensional space λ_i for $i = 1, \ldots, 5$. Figs. 3 and 4 show selected time snapshots of the solution for different values of λ_2. The snapshots show the Gaussian cloud initially centred at $x = -0.5$ and $y = 0$. Over time, the cloud is transported in a circle which completes a full rotation at $t = 1$. The cloud also decays to reach a near-zero magnitude after a full rotation. The parameter λ_1 is a linear scaling factor that controls the magnitude of the initial cloud, λ_4 and λ_5 control the initial coordinates of the centre of the cloud with respect to the domain centre. The parameter λ_3 is the decay constant of the cloud that controls the speed of the decay. The parameter λ_2 controls the size of the cloud. For smaller values of λ_2, the cloud size is bigger and the solution is smoother over the domain as shown in Fig. 3 whereas for higher values, the solution has a steeper gradient (spike-like) as shown in Fig. 4.

We test two different settings of the problem. The first is a smooth solution by varying λ_2 between 0.1 and 0.2 and the second is a steep solution with values of λ_2 between 2 and 4. The steep solution is more challenging to capture for a POD-based ROM because as the solution becomes steeper (or closer to being orthogonal), the required basis space becomes larger. That is, the rank of the snapshot matrix is higher for snapshots that are already orthogonal or near orthogonal, which entails more POD modes for an accurate representative model.

Table 1 summarizes the range of variation for each parameter. We are interested in building a reduced model that can reproduce the solution $q(x, y, t)$ over the spatial domain and time $t \in [0, 1]$ for any values of the parameters within the defined range.

We compare two approaches for this test. The first is the developed time-adaptive approach as described in Section 2. The second is the more direct approach by defining a fixed time grid then building a separate reduced model for each time instance in the grid. To reproduce the solution at any time t, the solution of the ROM models on the time grid is interpolated. However, to select the snapshots in parameter space, we still use the adaptive algorithm for each model on the grid. For this approach, a single basis space is constructed for all ROM models on the grid. A point in parameter space is selected to be included in the important set if any of the ROM models marked that point as important. Thus, the basis space for all ROM models on the fixed grid is updated with any point marked important by at least one of the ROM models. We initially defined the fixed time grid with 11 times points uniformly separated in the time window of interest (i.e., $t \in [0, 1]$).

For both approaches, we choose a greediness value of $\mu = 0$ and require the reduced model to have a maximum of 1% ℓ^2 norm error. Therefore, we set the global tolerance (ζ) to be 1% and the adaptive threshold (γ_{int}) to 0.1%. The POD truncation threshold (γ_{tr}) was set to 10^{-12}. The results are summarized in Table 2. The table presents a comparison between the two approaches for both the smooth and the steep solution settings in the number of calls to run the high-fidelity model, the total number of snapshots resulted from these runs, the number of POD modes after truncation, and the maximum relative error resulted from testing the model on 1000 randomly generated points using latin hypercube sampling (LHS) (i.e., snapshots generated by random point in the space formed by the parameters λ_i and time t, which were not part of the snapshot matrix). In Table 2, we report the maximum error results for the model of the fixed grid approach in two separate occasions: The maximum error at the predefined

Table 2
Results for the Molenkamp problem for the smooth and steep setting comparing time-adaptive and fixed grid approaches. The number of model runs indicates the number of calls to run the high-fidelity model. The number of snapshots indicates the total number of snapshots resulted from the high-fidelity model runs. The number of POD modes indicates the number of basis vectors selected after truncation. The maximum relative error reports the maximum computed error from testing the model on 1000 randomly generated points that are not part of the snapshots. For the fixed grid, the errors at the defined grid points and at interpolated time instances are reported separately.

Problem setting	Approach	Number of model runs	Number of snapshots	Number of POD modes	Maximum relative ℓ^2 error	
Smooth Molenkamp	Time-adaptive	775	6369	33	0.5%	
	Fixed grid (11 points)	1379	1379 × 11	33	At grid points Interpolated	0.17% 1.1%
Steep Molenkamp	Time-adaptive	2944	78 035	238	1.4%	
	Fixed grid (11 points)	5093	5093 × 11	223	At grid points Interpolated	0.33% 76%
	Fixed grid (101 points)	5093	5093 × 101	234	At grid points Interpolated	0.33% 0.34%

fixed grid instances and the maximum error at interpolated points in-between these instances. The maximum of the two values is the more relevant result to be compared to the error results of the adaptive approach. The interpolated values were obtained using splines interpolation.

For the smooth Molenkamp setting, the time-adaptive approach needed 775 high-fidelity model runs and computed 6369 snapshots. Out of the total number of snapshots, 4692 were marked important and included in the snapshot matrix. The number of POD modes after truncation was 33. On the other hand, the fixed time grid approach needed 1379 model runs resulting in a total of 15 169 snapshots, out of which 9889 snapshots were important. The result of the test on the 1000 random points showed the time-adaptive model having a maximum error of 0.5%, which is less than the set tolerance of 1%. The fixed grid model resulted in a maximum error of 0.17% at the grid points. However, at interpolated points (time instances in-between the defined grid points), the maximum error was 1.1%. The adaptive model has a clear advantage in this test as the error was lower and the model was more efficient in the number of high-fidelity model calls. Note that for the time-adaptive approach, not all high-fidelity model calls are simulated up to the end time T (where in this case $T = 1$). This is because the time-adaptive algorithm requests some high-fidelity model runs with a time t_l that is less than T. Therefore, the efficiency in the time-adaptive model is not only in the reduced number of high-fidelity model runs but also in the reduced computational burden of each model run.

For the steep solution test, we notice that both the time-adaptive and fixed grid approaches needed an increased number of model runs and a larger POD basis space compared to the smooth solution setting. For the time-adaptive approach, a total of 2944 high-fidelity model runs were requested with 78 035 snapshots sampled and 64 379 of them were marked as important. The algorithm selceted 238 POD modes. A conclusion similar to the smooth setting can be drawn in this case about the time-adaptive model being more efficient and more accurate. In fact, the fixed grid model captured the dynamics of the solution at the grid points but the error was as high as 76% at the interpolated points. In order to produce a more accurate model, we built another fixed grid model with 101 time points uniformly distributed in time $t \in [0, 1]$). This model reduced the maximum error at the interpolated points to about 0.34%. However, this was achieved with about twice as much model runs compared to the time-adaptive approach and about 6 times more snapshots.

Table 3 summarizes the number of projected important points on each dimension. This number represents the linearity of the output with respect to each dimension. A value of 1 signals that the algorithm considered that dimension to be constant. In other words, varying the value of that parameter has a negligible effect on the output of the model with respect to the defined tolerance. A value of 3 means that the algorithm considered the output of the model to be linear or piecewise linear with respect to that dimension. A higher value implies a nonlinear parameter, and the degree of the non-linearity scales with the value. It can be seen that the algorithm recognized λ_1 as a linear scaling parameter in both settings. The decay constant λ_3 was the most sampled parameter and was not affected by the change in the shape of the solution controlled by λ_2. This can be confirmed from the exact solution in Eq. (17). The parameters λ_4 and λ_5, on the other hand, were affected by the shape as they control the location of

Table 3
Number of unique important nodes along each dimension for the Molenkamp test.

Parameter	Smooth setting	Steep setting
λ_1	3	3
λ_2	9	13
λ_3	33	33
λ_4	5	17
λ_5	5	17
t	129	513

the cloud. For this reason, the algorithm added more points along the dimensions of these parameters in the steep setting, where the cloud size is smaller compared to the smooth setting.

3.2. Lid-driven cavity test

In this test, the incompressible Navier–Stokes equations are solved in a two-dimensional lid-driven cavity. Zero-velocity boundary conditions are assumed around the cavity except at the top lid, where a velocity equal to v_{lid} is imposed. The model equations read

$$\begin{aligned}
&\frac{\partial \boldsymbol{u}(x,y,t)}{\partial t} + \nabla\cdot(\boldsymbol{u}(x,y,t)\otimes\boldsymbol{u}(x,y,t)) - \nabla\cdot\left[\nu(\nabla\boldsymbol{u}(x,y,t) + \nabla(\boldsymbol{u}(x,y,t))^{\top})\right] = -\nabla p, \quad \text{in } \Omega = [0,1]^2,\\
&\nabla\cdot\boldsymbol{u}(x,y,t) = 0 \qquad \text{in } \Omega,\\
&\boldsymbol{u}(x,y,t) = \boldsymbol{0} \qquad \text{on } \Gamma_1,\\
&\boldsymbol{u}(x,y,t) = (v_{\text{lid}},0)^{\top} \qquad \text{on } \Gamma_2,\\
&\boldsymbol{u}(x,y,0) = \boldsymbol{0},
\end{aligned} \tag{18}$$

where $\boldsymbol{u}(x,y,t)$ is the flow velocity, ν is the viscosity, p is the pressure, and v_{lid} is the velocity of the top cavity wall, which was imposed as a ramp according to

$$v_{\text{lid}} = \begin{cases} -t & \text{if } 0 \le t < 1,\\ -1 & \text{if } t \ge 1. \end{cases} \tag{19}$$

The domain is illustrated in Fig. 5. We consider a laminar flow with Reynolds number of 1000 and are interested in the velocity field within a time range $t \in [0, 100]$. An in-house Navier–Stokes solver was used as the high-fidelity model [35]. The system of equations is solved with a pressure-correction method, discretizing the equations in space with a discontinuous Galerkin finite element method and in time with the implicit Euler scheme. A fixed time-step size of 10^{-3} was chosen. The domain was discretized on a structured non-uniform (finer near the walls) mesh of 60×60 elements. The velocity field was discretized using a second-order polynomial, which leads to a total of 43 200 degrees of freedom for the high-fidelity model. A single high-fidelity simulation to $t = 100$ requires about 35 CPU-hours.

Lorenzi et al. (2016) [36] has presented an approach to build a reduced model for this benchmark using a projection-based POD approach. To select the snapshots, the authors sampled the velocity field using a fixed grid of 1000 equally spaced time points. As pointed out in their work, this test is challenging for projection-based POD methods due to the potential instability of the reduced model induced by truncating modes that have small energy magnitudes but are important for dissipating the energy of the system. Nonintrusive approaches do not face such an issue.

We aim to build a non-parametrized reduced model that captures the velocity evolution with time as a response to v_{lid}. We require a 0.5% maximum ℓ^2 norm error and set a POD truncation threshold γ_{tr} to 10^{-12}. The algorithm selected 463 snapshots and marked 232 of them as important. The number of POD modes was 229. The selected points are shown in Fig. 6. The algorithm was successful in identifying the first part of the transient to be more important than the last. This is because most of the changes to the velocity field occur within the first few seconds and then gradually stabilize until a steady-state is reached. In fact, the algorithm recognizes that the flow is in a steady-state by $t = 50$ and no snapshots were marked important between $t = 50$ and $t = 100$.

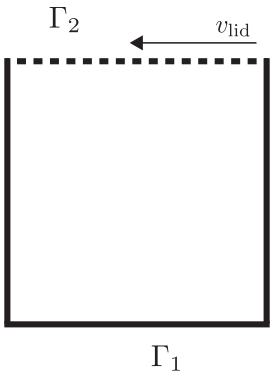


Fig. 5. Illustration of the domain for the lid-driven cavity problem. The boundaries around the cavity are marked by Γ_1 except at the top lid where Γ_2 labels the boundary there.

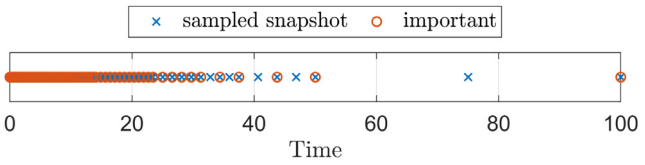


Fig. 6. Time instances selected by the time-adaptive algorithm for the lid-driven cavity test. Points added to the important set are marked with a red circle.

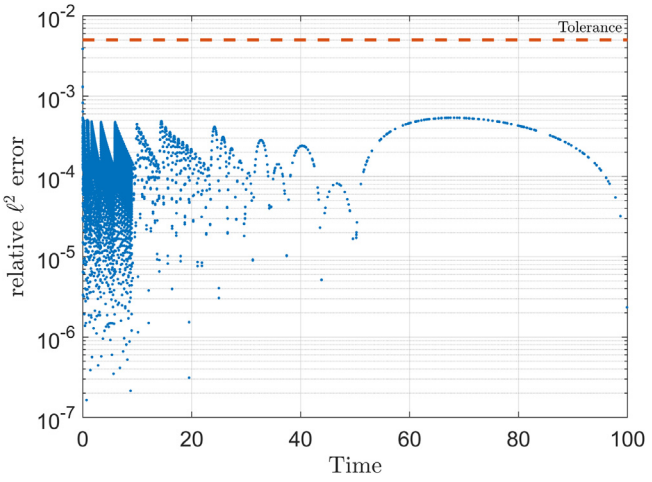
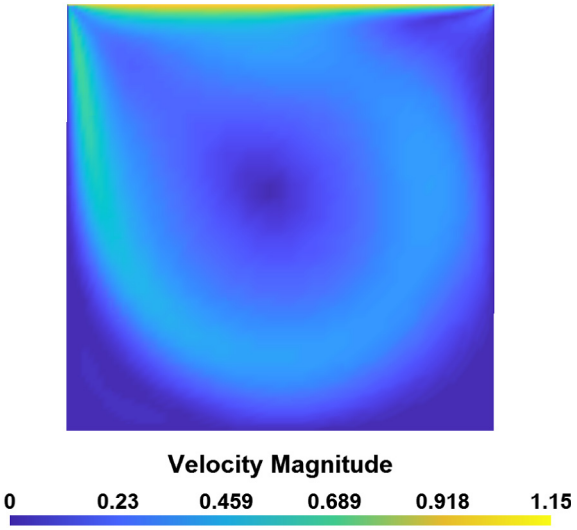
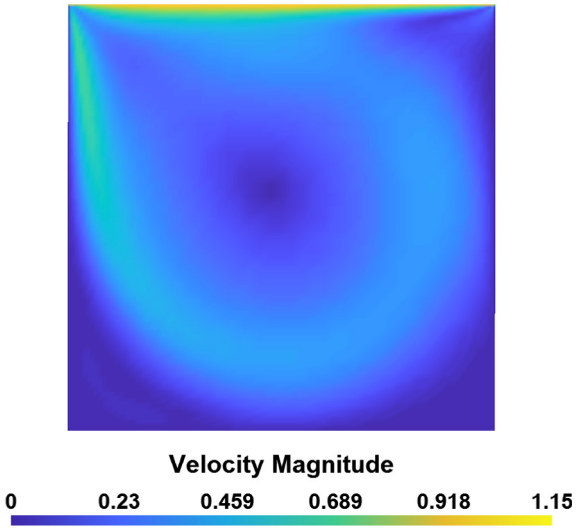


Fig. 7. Relative ℓ^2 error in the reduced model for the lid-driven cavity problem tested on 10,000 points that were not part of the snapshot matrix. The global tolerance was set at 0.5%.

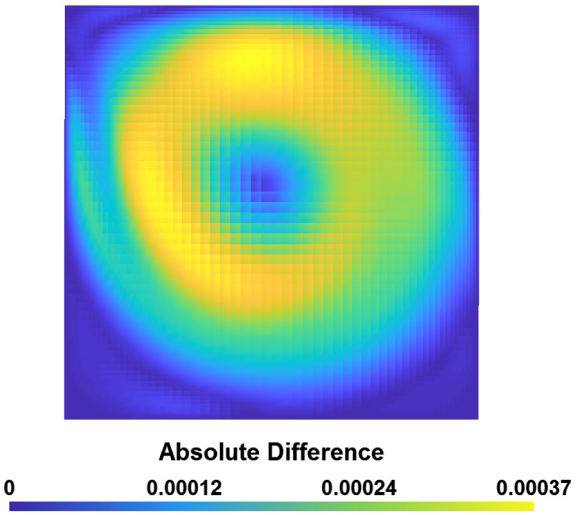
To test the reduced model, Fig. 7 shows the computed relative ℓ^2 norm error between the reduced model and the high-fidelity model at 10,000 randomly generated points in time. The tested points were not part of the snapshots matrix. It can be seen that all tested points resulted in an error below the set tolerance of 0.5%. The figure shows the error to oscillate between 10^{-7} and the tolerance (5×10^{-3}), with the oscillation frequency being higher in the first part of the time domain. These oscillations are due to the fact that some of the tested points are very close to points that were marked important and included in the snapshot matrix. The error for reconstructing a point in the snapshot matrix can be estimated with the POD truncation threshold $\gamma_{\text{tr}} = 10^{-12}$. Therefore, γ_{tr} can be considered as a lower bound of the error in the ROM model. When a point is tested near a point included in the snapshot matrix, the error can be expected to approach γ_{tr}. On the other hand, when the tested point is further from any



(a) Velocity magnitude of HFM.



(b) Velocity magnitude of ROM.

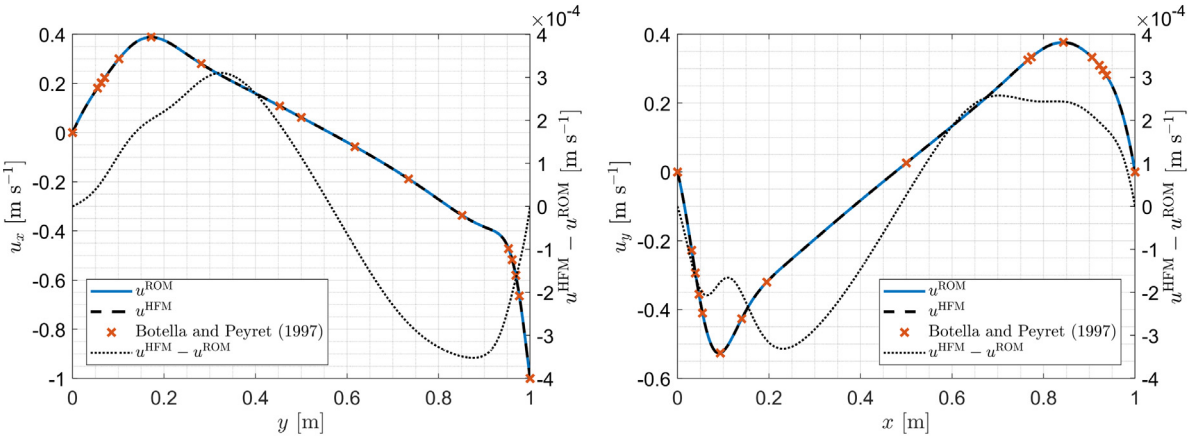


(c) Spatial absolute difference between HFM and ROM.

Fig. 8. Comparison between the reduced model (ROM) and the high-fidelity model (HFM) for the lid-driven cavity problem at $t = 68.157$ where the relative ℓ^2 error was found to be 0.054%.

point in the snapshot matrix, the error increases to the tolerance. This is evident by considering Fig. 6 where the frequency of the selected important points correlated well with the frequency of the oscillation in the error shown in Fig. 7.

The figure shows the maximum error to be 0.38%. In fact, this maximum is at the first time step of the high-fidelity model. This high error is attributed to the discontinuity in the velocity field between the initial conditions (null velocity everywhere) and the first time step (velocity almost zero except at the very top of the cavity where v_{lid} is introduced). The relative error is magnified by the near zero ℓ^2 norm of the solution at this first step. The maximum absolute difference between the reduced model and the high-fidelity model at this point was about 6×10^{-6}, while the magnitude of the maximum velocity at the top of the cavity was 8×10^{-4}. Beyond $t = 1$ (when the input ramp ends), the highest error is observed to be 0.054% at $t = 68.157$ s. A comparison between the high-fidelity model and the reduced model at this point is shown in Fig. 8. We also plot the velocity components along the horizontal



(a) u_x profile along vertical centerline.

(b) u_y profile along horizontal centerline.

Fig. 9. Velocity components profile along the central line for both the high-fidelity (HFM) and the reduced model (ROM) for lid-driven cavity problem at the time $t = 68.157$ s. Benchmark data from Botella and Peyret (1997) [37] are also marked. The right axes of the figures show the difference between HFM and ROM ($u^{\mathrm{HFM}} - u^{\mathrm{ROM}}$).

and vertical central lines in Fig. 9 at $t = 68.157$. The figures show that the reduced model produced an accurate representation of the high-fidelity model despite the fact that no snapshots were selected in the important set between $t = 50$ and $t = 100$. In addition, Fig. 9 compares the results with steady-state benchmark data from Botella and Peyret (1997) [37] to verify that the algorithm was successful in recognizing the flow to be in a steady-state beyond $t = 50$. Simulating the flow for the 10,000 tested points required about 10 s on a personal computer with the reduced model compared to the 35 CPU-hours needed by the high-fidelity model to simulate the flow to $t = 100$.

3.3. Subcritical reactor test

Nuclear reactors are complex systems with multiple interacting physical phenomena. A standard model to describe the neutron flux dynamics inside a reactor is the time-dependent diffusion equation [38]

$$\frac{1}{v}\frac{\partial \phi(\boldsymbol{x},t)}{\partial t} - \nabla \cdot D(\boldsymbol{x})\nabla\phi(\boldsymbol{x},t) + \Sigma_a(\boldsymbol{x})\phi(\boldsymbol{x},t) = S(\boldsymbol{x},t), \qquad \boldsymbol{x} \in \Omega \tag{20}$$

where $\phi(\boldsymbol{x},t)$ is the one-speed neutron flux with speed $v = 300{,}000$ cm/s, $D(\boldsymbol{x})$ is the diffusion coefficient, and Σ_a is the absorption (removal) cross section. The source term $S(\boldsymbol{x},t)$ is defined as

$$S(\boldsymbol{x},t) = (1-\beta)\nu\Sigma_f(\boldsymbol{x})\phi(\boldsymbol{x},t) + \lambda C(\boldsymbol{x},t) + q(\boldsymbol{x},t), \tag{21}$$

where β is the delayed neutron fraction, ν is the number of neutrons emitted per fission, Σ_f is the fission cross section, $q(\boldsymbol{x},t)$ is the external neutron source, and λ is the decay constant of the precursors $C(\boldsymbol{x},t)$. The dynamics of the precursors is governed by

$$\frac{\partial C(\boldsymbol{x},t)}{\partial t} = -\lambda C(\boldsymbol{x},t) + \beta\nu\Sigma_f(\boldsymbol{x})\phi(\boldsymbol{x},t). \tag{22}$$

We consider a two-dimensional domain (i.e., $\boldsymbol{x} = (x,y)$) divided into 4 regions as shown in Fig. 10. The dimensions of the reactor (including the extrapolated length) were set to $x, y \in [-109.36, 109.36]$, which were chosen such that the flux $\phi(\boldsymbol{x},t)$ is zero at the boundary Γ. Each region has uniform material properties such that $D(\boldsymbol{x}), \Sigma_a(\boldsymbol{x}), \Sigma_f(\boldsymbol{x}) \rightarrow \boldsymbol{D}, \boldsymbol{\Sigma}_a, \boldsymbol{\Sigma}_f \in \mathbb{R}^4$. The external source $q(\boldsymbol{x},t)$ is assumed to be present only in the lower left corner of the domain,

$$q(\boldsymbol{x},t) = \begin{cases} q_{\mathrm{ext}}(t) & \forall \boldsymbol{x} \in \text{Region 1}, \\ 0 & \text{elsewhere.} \end{cases} \tag{23}$$

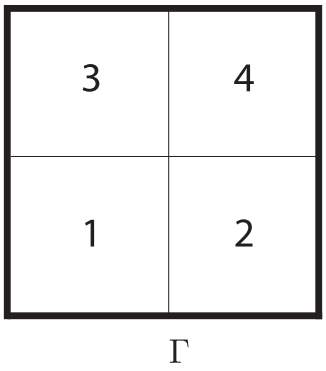


Fig. 10. Domain of the subcritical reactor test showing the boundary Γ and the 4 regions. The neutron source is present only in Region 1.

Table 4
Nominal values and range of variations of each parameter in the subcritical reactor test.

Parameter	Nominal value	% variation	Parameter	Nominal value	% variation
D_1	9.21 cm	±5%	Σ_{a3}	0.153 cm^{-1}	±5%
D_2	9.21 cm	±5%	Σ_{a4}	0.153 cm^{-1}	±5%
D_3	9.21 cm	±5%	β	0.00686	±20%
D_4	9.21 cm	±5%	λ	0.08 s^{-1}	±20%
Σ_{a1}	0.153 cm^{-1}	±5%	q_1	2.5 n/cm^3 s	±100%
Σ_{a2}	0.153 cm^{-1}	±5%			

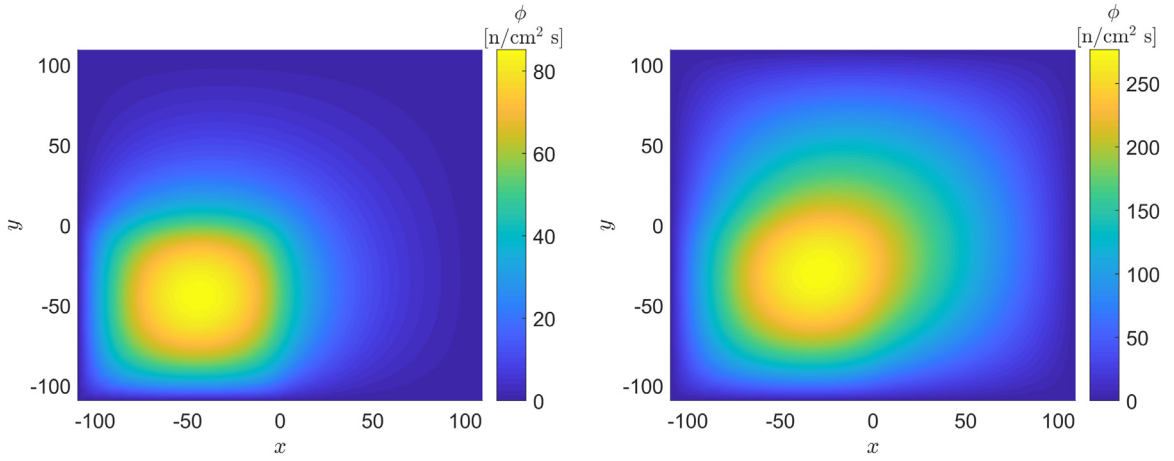
The multiplication factor of a reactor (k_{eff}) is the ratio of the neutrons produced from fission in one generation to the neutrons lost in the previous generation. For $k_{\text{eff}} < 1$, a fission chain reaction cannot be sustained and the reactor is said to be subcritical, while for $k_{\text{eff}} > 1$, the reactor is supercritical since the neutron population is multiplying over time. For $k_{\text{eff}} = 1$, the reactor is critical and the neutron population is constant in time. In our test, the reactor is assumed to be in a subcritical condition with a multiplication factor $k_{\text{eff}} = 0.94$ in the nominal state. The neutron population is kept in a steady-state due to the external source $q_{\text{ext}}(t) = 1$. At time $t = 100$ s, the source intensity is perturbed. This is equivalent to a step-change in the source at time $t = 100$ s,

$$q_{\text{ext}}(t) = \begin{cases} 1 & \text{for } 0 \leq t < 100, \\ q_1 & \text{for } t \geq 100, \end{cases} \tag{24}$$

where $q_1 \in [0, 5]$. The neutron flux response is then observed as a function of time and space. We aim to build a reduced model that captures the dynamics of flux under different conditions of material properties $\boldsymbol{D} = [D_1, D_2, D_3, D_4]^\top$, $\boldsymbol{\Sigma}_a = [\Sigma_{a1}, \Sigma_{a2}, \Sigma_{a3}, \Sigma_{a4}]^\top$, λ, β and source intensity q_1. Therefore, the model is parametrized with an 11-dimensional space. The nominal values and range of variations of each parameter are summarized in Table 4.

The parameters range of variations was chosen such that the reactor is kept in a subcritical condition ($k_{\text{eff}} < 1$) in all cases. The level of the subcritical condition is controlled with $\boldsymbol{\Sigma}_a$ and $\boldsymbol{D}$, which also set the initial flux value. As the reactor gets closer to criticality, the response following a perturbation becomes steeper and the transient becomes longer. Therefore, the time until reaching a new steady state is a function of the material properties. The flux response to the perturbations can be described by two main parts. First, an initial abrupt response due to the prompt neutrons, which has a magnitude controlled by β, $\boldsymbol{\Sigma}_a$ and $\boldsymbol{D}$. This prompt response has a duration in the order of $1/v\Sigma_a = 2 \times 10^{-5}$ s. The second is the response due to the delayed neutrons emitted from the decay of the precursors, which is governed by a time in the order of $1/\lambda = 12$ s. The final steady-state value scales linearly with the external source q_1. Therefore, this test poses a challenge for any nonintrusive approach because different parameters affect the dynamics at different timescales.

The model was solved using a finite element method with an unstructured mesh discretizing the spatial domain such that the model has 1084 degrees of freedom. An implicit Euler discretization for time was employed with variable step sizes. The step size following the perturbation was taken as 10^{-5} s for the first 0.1 s to resolve steep



(a) Initial flux distribution in the case of all parameters at nominal values.

(b) Initial flux distribution in the case of $\mathbf{\Sigma}_a$ being reduced by 5% while all other parameters are at nominal values.

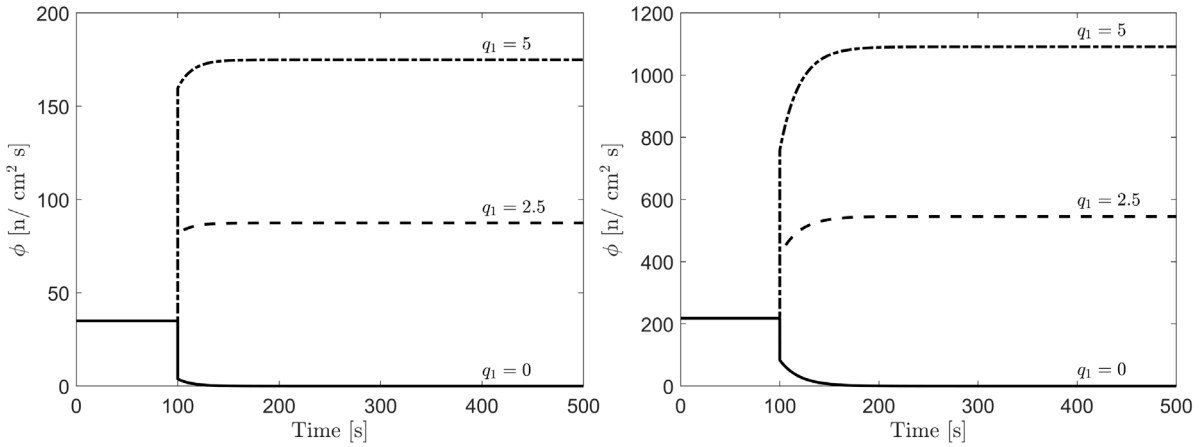
Fig. 11. Comparison of the distribution of the initial neutron flux ($t = 0$) for the subcritical reactor test showing the difference in the flux intensity and shape between the case of all parameters at nominal values and a case of $\mathbf{\Sigma}_a$ reduced by 5% of the nominal value while all other parameters are kept at nominal values.

variations. Then, a step size of 10^{-2} s was employed for the remainder of the transient. A single simulation of the transient takes about 30 s on a personal computer. Therefore, this model is not computationally demanding. However, we considered this test to challenge the algorithm in capturing the effect of the 11 parameters on the complete transient. The initial flux distribution before perturbing the source ($t = 0$) for the nominal case is shown in Fig. 11a while Fig. 11b shows the initial flux distribution when reducing $\mathbf{\Sigma}_a$ by 5%. It can be seen that reducing $\mathbf{\Sigma}_a$ caused the flux intensity to increase and the shape to broaden over the spatial domain, which is expected since fewer neutrons are being absorbed in this case. Fig. 12 shows the transient tracking the flux at the centre of the reactor following three different source perturbations $q_1 \in \{0, 2.5, 5\}$ and compares the case of all parameters at the nominal values with the case of only reducing $\mathbf{\Sigma}_a$ by 5% of the nominal value. The figure shows that by reducing $\mathbf{\Sigma}_a$, the reactor is closer to criticality, which not only has an effect on the initial flux value but also resulted in a slower response to reach a new steady-state following a perturbation.

We built a reduced model with a global tolerance of 1% and a POD truncation threshold of 10^{-12}. The algorithm ran 3295 high-fidelity simulations and selected 155 270 snapshots, where 52 710 were marked important. The number of POD modes was 294 after truncation. The high number of selected snapshots is a result of the high dimensional parameter space combined with the complex dynamics of the problem. Note that the number of selected snapshots is not a function of the number of degrees of freedom. Therefore, scaling this problem to larger degrees of freedom will not affect the selection of the snapshots.

The projection of the important points onto each dimension is given in Table 5. The table shows that the algorithm considered the diffusion coefficient to be linear within the defined range of ±5% while the absorption cross section was the most nonlinear parameter. This is expected because the absorption cross section has a direct effect on the subcriticality level of the reactor. In addition, it is shown that Σ_{a1} was considered the most nonlinear parameter and was sampled more densely because it belongs to the region that contains the external source. On the other hand, Σ_{a4} belongs to the region furthest from the source and was sampled the least. The parameters λ and β had 3 unique nodes each, which implies that within the defined ranges of ±20%, the effect of these parameters on the dynamics of the reactor is linear. The external source intensity was correctly identified as a linear scaling factor. The time parameter was considered important at 110 time instances.

Fig. 13 shows the projection of the sampled points onto the (Σ_{a1}, t) plane. It can be seen that the algorithm sampled most of the points during the period from $t = 100$ s to $t = 250$ s, which is the transient time following the source perturbation. Along the Σ_{a1} dimension, most of the sampled points were in the lower range of the domain. This is expected, because for lower values of the absorption cross section, the reactor is closer to criticality and the response becomes more nonlinear.



(a) selected transients of the flux following different source perturbations at $t = 100$ s in the case of all parameters at nominal values.

(b) selected transients of the flux following different source perturbations at $t = 100$ s in the case of $\mathbf{\Sigma}_a$ being reduced by 5% while all other parameters are at nominal values.

Fig. 12. Transients for the subcritical reactor test at a point in the centre of the reactor following selected perturbation of $q_1 \in \{0, 2.5, 5\}$ showing the difference in response between the case of all parameters at nominal values and a case of reducing $\mathbf{\Sigma}_a$ by 5% of the nominal value.

Table 5
Number of unique important nodes along each dimension for the subcritical reactor test.

Parameter	Number of unique nodes	Parameter	Number of unique nodes
D_1	3	Σ_{a3}	12
D_2	3	Σ_{a4}	8
D_3	3	β	3
D_4	3	λ	3
Σ_{a1}	22	q_1	3
Σ_{a2}	11	t	110

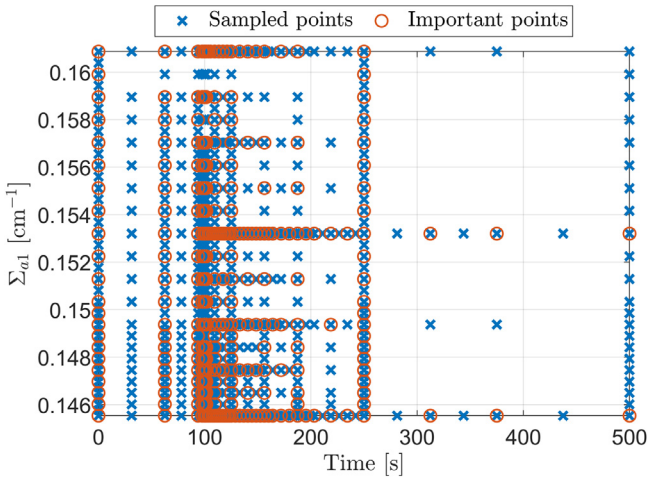


Fig. 13. Projection of the sampled point onto the (Σ_{a1}, t) plane for the subcritical reactor test. Points included in the important set are marked with a circle.

The model was tested on 1000 randomly generated points using LHS method. The histogram of the relative errors in Fig. 14 shows that 99.5% of the points were below the tolerance. The maximum relative error was found

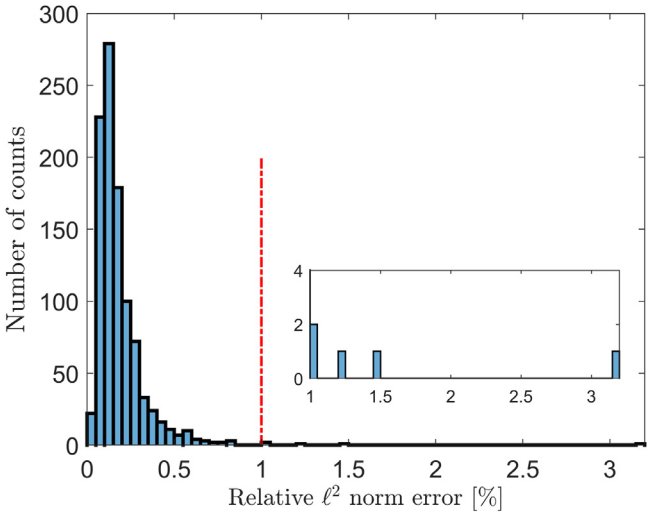


Fig. 14. Histogram of the relative error resulting from testing the reduced model for the subcritical reactor on 1000 random points. The maximum error was 3.2%. 99.5% of the points resulted in errors below the tolerance of 1%. A close up of the histogram for the values above the tolerance is shown in the box.

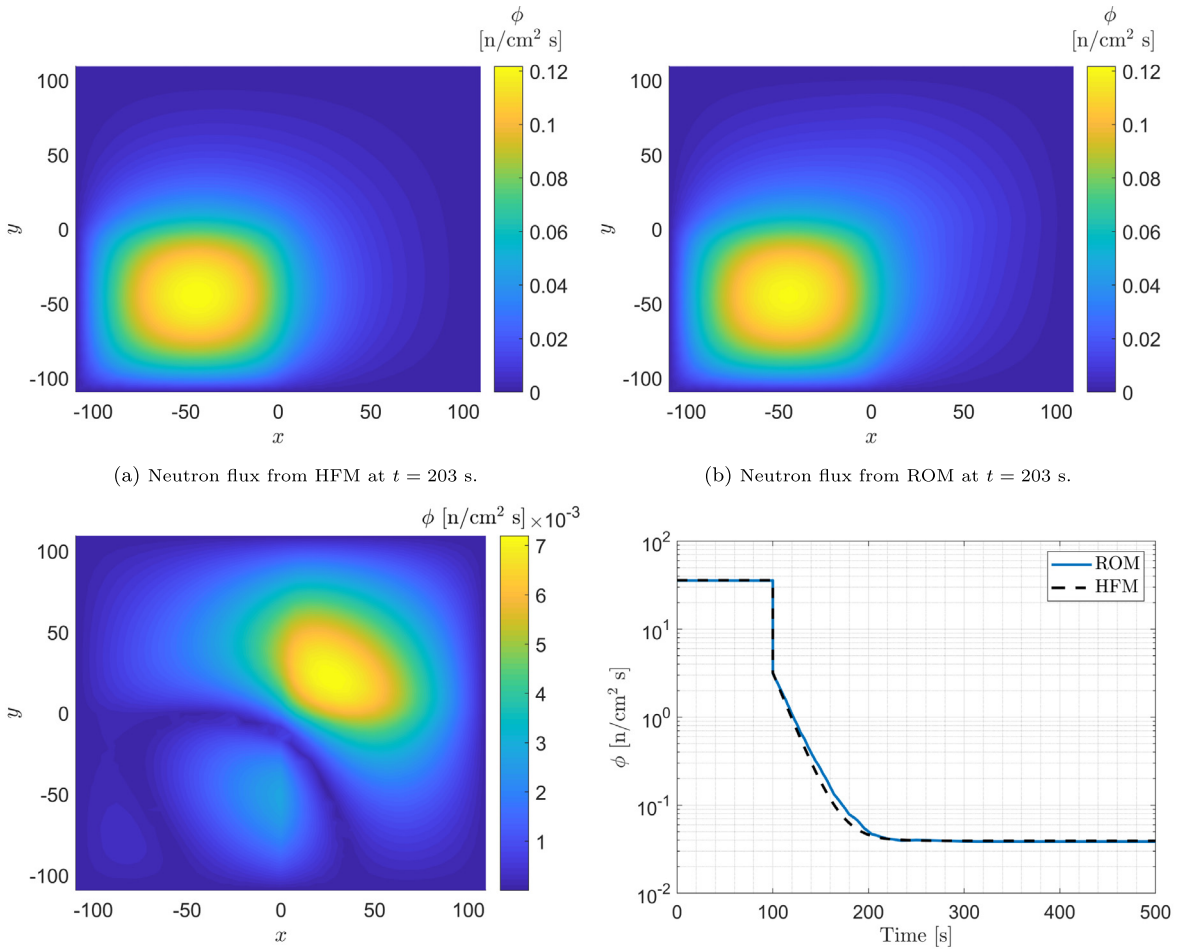
to be 3.2%. The point with the maximum error corresponds to a case where the source value $q_1 = 1.1 \times 10^{-3}$ n/cm^3 s and time $t = 203$ s. The solutions of the reduced and high-fidelity models are compared for this case in Fig. 15. It can be seen from the figure that the flux at this case is almost zero at $t = 203$ s. The maximum absolute difference between the reduced and high-fidelity models was found to be 7×10^{-3} n/cm^2 s. The complete transient for this case is also shown in Fig. 15d. The figure shows that the ROM model was able to track the reference solution with great accuracy at the initial and final steady-state while most of the discrepancy was contained in the transient. The second highest error was 1.5%, which was also a point with q_1 near zero ($q_1 = 7 \times 10^{-4}$ n/cm^3 s). The third highest error was 1.2%, which was found at $q_1 = 1.76$ n/cm^3 s and time $t = 470$ s. This case is shown in Fig. 16, which shows that the error in this case was in the steady-state value rather than the transient. Simulating the 1000 points needed 10 s with the reduced model while the high-fidelity model required about 6 h for the same points.

4. Conclusions

This work presented an approach for time and parameter adaptivity to build a nonintrusive reduced-order model. The approach is an extension of our sparse grid adaptive-POD algorithm to time-dependent parametrized problems. Time was considered as an additional parameter, which enabled the locally adaptive sparse grid algorithm to be applied directly. The adaptivity provided a tool to include more snapshots from important time windows, which reduces the probability of overlooking crucial dynamics in the POD snapshot matrix. Moreover, the efficiency of the construction phase (offline phase) is improved by sampling the high-fidelity model less in time periods of steady-state or slow (smooth) changes.

Three numerical problems were presented to test the proposed approach. The first was a Molenkamp problem with five parameters, which was solved in two settings: a smooth solution and a more challenging steep solution. In this test, we compared the time-adaptive approach with an a priori fixed sampling approach of the time domain. The results in both settings showed that the time-adaptive approach was more efficient without compromising the accuracy. Additionally, the algorithm was able to identify the linearity of the response with respect to each parameter. The second test was a standard lid-driven cavity problem. For this problem, only time was considered as a parameter. The adaptive algorithm was able to identify that the important time period was the first few seconds of the transient when the flow is still developing.

Moreover, the algorithm recognized that after about t = 50, the flow was fully developed and no important snapshots were selected between $t = 50$ and $t = 100$. The reduced model was able to simulate the flow in 10 s compared to the 35 CPU-hours needed by the high-fidelity model. The last subcritical reactor test was challenging



(a) Neutron flux from HFM at $t = 203$ s.

(b) Neutron flux from ROM at $t = 203$ s.

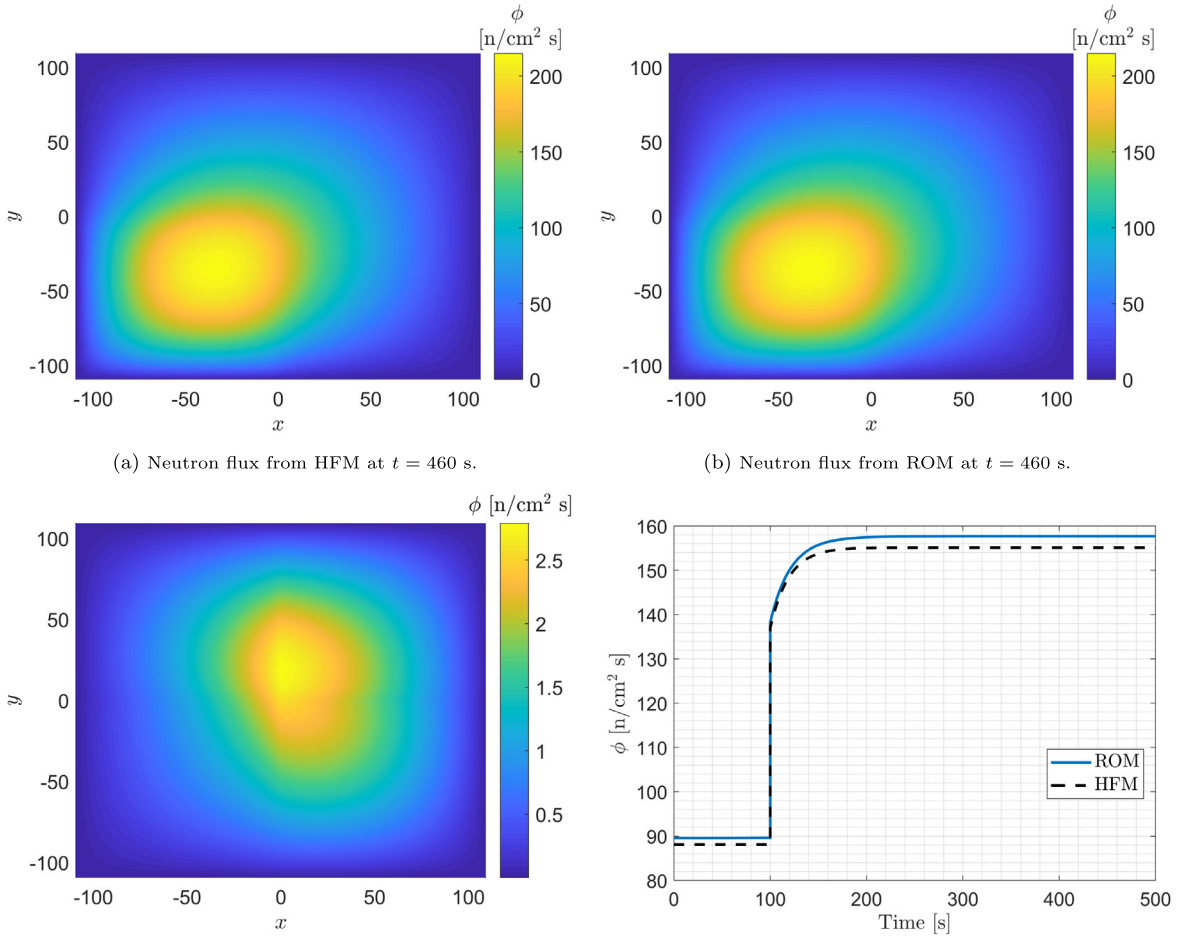
(c) Absolute spatial difference between HFM and ROM at $t = 203$ s.

(d) Complete transient at the center of the reactor.

Fig. 15. Comparison between ROM and HFM for the subcritical reactor test at the point with the maximum error (3.2%), $q_1 = 1.1 \times 10^{-3}$ n/cm^3 s and time $t = 203$ s.

not only due to the higher dimensionality of the parameter space but also due to the abrupt dynamics at small timescales. The algorithm correctly recognized the time of the important transient following the source perturbation. In addition, the algorithm revealed the region of importance of each parameter and correspondingly concentrated the sampling of the points in these discovered regions. This improved the efficiency of the approach compared to non-adaptive techniques. The model was tested on 1000 randomly generated points which were simulated in 10 s while the reference model needed about 6 h to simulate the same points. In all tests, the reduced models built with the time-adaptive approach captured the dynamics of the model with an accuracy that fell within the defined tolerances.

Our approach was nonintrusive which can be applied to a wide range of problems. Despite the fact that nonintrusive approaches do not preserve the physical structure of the system, using adaptive approaches, such as the one presented in this work, provides an insight into the physics of the system by ranking the importance of the parameters or exploring linearity. A challenge for any adaptive method is to scale efficiently to higher dimensional spaces. This issue was addressed in our approach by using the locally adaptive sparse grid approach. However, for the Molenkamp and subcritical reactor tests, the algorithm required a high number of snapshots compared to the number of POD modes selected after truncation. This is an indication that most of the sampled snapshots were needed for the construction of the surrogate model of the POD coefficient more than revealing additional dynamics



(a) Neutron flux from HFM at $t = 460$ s.

(b) Neutron flux from ROM at $t = 460$ s.

(c) Absolute spatial difference between HFM and ROM at $t = 460$ s.

(d) Complete transient at the center of the reactor.

Fig. 16. Comparison between ROM and HFM for the subcritical reactor test at the point with error 1.2%, $q_1 = 1.76$ n/cm^3 s and time $t = 460$ s.

of the system. Therefore, an area to study in future work is the use of higher order interpolation models for the POD coefficients with the aim to reduce the number of snapshots and further improve the efficiency. Another interesting area of research to achieve this goal is investigating a space–time decomposition of the basis space or the construction of several local basis spaces tailored to different dynamics instead of a single global basis space.

Declaration of competing interest

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

Acknowledgement

The authors acknowledge the support of King Abdulaziz City for Science and Technology (KACST) for this work.

References

[1] A.C. Antoulas, D.C. Sorensen, S. Gugercin, A survey of model reduction methods for large-scale systems, Contemp. Math. 280 (2001) 193–219.

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

[3] A. Antoulas, A. Ionita, S. Lefteriu, On two-variable rational interpolation, Linear Algebra Appl. 436 (8) (2012) 2889–2915, http://dx.doi.org/10.1016/j.laa.2011.07.017.

[4] B. Peherstorfer, S. Gugercin, K. Willcox, Data-driven reduced model construction with time-domain loewner models, SIAM J. Sci. Comput. 39 (5) (2017) A2152–A2178, http://dx.doi.org/10.1137/16m1094750.

[5] P.J. Schmid, Dynamic mode decomposition of numerical and experimental data, J. Fluid Mech. 656 (2010) 5–28, http://dx.doi.org/10.1017/s0022112010001217.

[6] R. Swischuk, L. Mainini, B. Peherstorfer, K. Willcox, Projection-based model reduction: Formulations for physics-based machine learning, Comput. & Fluids 179 (2019) 704–717, http://dx.doi.org/10.1016/j.compfluid.2018.07.021.

[7] H.V. Ly, H.T. Tran, Modeling and control of physical processes using proper orthogonal decomposition, Math. Comput. Modelling 33 (1–3) (2001) 223–236.

[8] V. Buljak, Inverse Analyses with Model Reduction: Proper Orthogonal Decomposition in Structural Mechanics, Springer, Berlin, 2011.

[9] C. Audouze, F.D. Vuyst, P.B. Nair, Nonintrusive reduced-order modeling of parametrized time-dependent partial differential equations, Numer. Methods Partial Differential Equations 29 (5) (2013) 1587–1628, http://dx.doi.org/10.1002/num.21768.

[10] J. Hesthaven, S. Ubbiali, Non-intrusive reduced order modeling of nonlinear problems using neural networks, J. Comput. Phys. 363 (2018) 55–78.

[11] N. Nguyen, J. Peraire, Gaussian functional regression for output prediction: Model assimilation and experimental design, J. Comput. Phys. 309 (2016) 52–68.

[12] M. Xiao, P. Breitkopf, R.F. Coelho, C. Knopf-Lenoir, M. Sidorkiewicz, P. Villon, Model reduction by CPOD and kriging, Struct. Multidiscip. Optim. 41 (4) (2009) 555–574, http://dx.doi.org/10.1007/s00158-009-0434-9.

[13] F. Alsayyari, Z. Perkó, D. Lathouwers, J.L. Kloosterman, A nonintrusive reduced order modelling approach using proper orthogonal decomposition and locally adaptive sparse grids, J. Comput. Phys. 399 (2019) 108912, http://dx.doi.org/10.1016/j.jcp.2019.108912.

[14] F. Alsayyari, M. Tiberga, Z. Perkó, D. Lathouwers, J.L. Kloosterman, A nonintrusive adaptive reduced order modeling approach for a molten salt reactor system, Ann. Nucl. Energy 141 (2020) 107321, http://dx.doi.org/10.1016/j.anucene.2020.107321.

[15] W. Chen, J.S. Hesthaven, B. Junqiang, Y. Qiu, Z. Yang, Y. Tihao, Greedy nonintrusive reduced order model for fluid dynamics, AIAA J. 56 (12) (2018) 4927–4943, http://dx.doi.org/10.2514/1.j056161.

[16] D. Xiao, F. Fang, C. Pain, I. Navon, A parameterized non-intrusive reduced order model and error analysis for general time-dependent nonlinear partial differential equations and its applications, Comput. Methods Appl. Mech. Engrg. 317 (2017) 868–889, http://dx.doi.org/10.1016/j.cma.2016.12.033.

[17] B. Peherstorfer, K. Willcox, Data-driven operator inference for nonintrusive projection-based model reduction, Comput. Methods Appl. Mech. Engrg. 306 (2016) 196–215, http://dx.doi.org/10.1016/j.cma.2016.03.025.

[18] E. Qian, B. Kramer, B. Peherstorfer, K. Willcox, Lift & learn: Physics-informed machine learning for large-scale nonlinear dynamical systems, Physica D 406 (2020) 132401, http://dx.doi.org/10.1016/j.physd.2020.132401.

[19] M. Guo, J.S. Hesthaven, Data-driven reduced order modeling for time-dependent problems, Comput. Methods Appl. Mech. Engrg. 345 (2019) 75–99, http://dx.doi.org/10.1016/j.cma.2018.10.029.

[20] A.T. Mohan, D.V. Gaitonde, A deep learning based approach to reduced order modeling for turbulent flow control using lstm neural networks, 2018, arXiv:1804.09269.

[21] F. Regazzoni, L. Dedè, A. Quarteroni, Machine learning for fast and reliable solution of time-dependent differential equations, J. Comput. Phys. 397 (2019) 108852, http://dx.doi.org/10.1016/j.jcp.2019.07.050.

[22] R. Hu, F. Fang, C. Pain, I. Navon, Rapid spatio-temporal flood prediction and uncertainty quantification using a deep learning method, J. Hydrol. 575 (2019) 911–920, http://dx.doi.org/10.1016/j.jhydrol.2019.05.087.

[23] O. San, R. Maulik, M. Ahmed, An artificial neural network framework for reduced order modeling of transient flows, Commun. Nonlinear Sci. Numer. Simul. 77 (2019) 271–287, http://dx.doi.org/10.1016/j.cnsns.2019.04.025.

[24] Z. Deng, Y. Chen, Y. Liu, K.C. Kim, Time-resolved turbulent velocity field reconstruction using a long short-term memory (lstm)-based artificial intelligence framework, Phys. Fluids 31 (7) (2019) 075108.

[25] S. Pawar, S.M. Rahman, H. Vaddireddy, O. San, A. Rasheed, P. Vedula, A deep learning enabler for nonintrusive reduced order modeling of fluid flows, Phys. Fluids 31 (8) (2019) 085101, http://dx.doi.org/10.1063/1.5113494.

[26] H.F.S. Lui, W.R. Wolf, Construction of reduced-order models for fluid flows using deep feedforward neural networks, J. Fluid Mech. 872 (2019) 963–994, http://dx.doi.org/10.1017/jfm.2019.358.

[27] A. Alwosheel, S. van Cranenburgh, C.G. Chorus, Is your dataset big enough? Sample size requirements when using artificial neural networks for discrete choice analysis, J. Choice Model. 28 (2018) 167–182, http://dx.doi.org/10.1016/j.jocm.2018.07.002.

[28] S.O. Haykin, Neural Networks and Learning Machines, third ed., Pearson, 2008.

[29] K. Kunisch, S. Volkwein, Optimal snapshot location for computing POD basis functions, ESAIM Math. Model. Numer. Anal. 44 (3) (2010) 509–529, http://dx.doi.org/10.1051/m2an/2010011.

[30] O. Lass, S. Volkwein, Adaptive POD basis computation for parametrized nonlinear systems using optimal snapshot location, Comput. Optim. Appl. 58 (3) (2014) 645–677, http://dx.doi.org/10.1007/s10589-014-9646-z.

[31] G.M. Oxberry, T. Kostova-Vassilevska, W. Arrighi, K. Chand, Limited-memory adaptive snapshot selection for proper orthogonal decomposition, Internat. J. Numer. Methods Engrg. 109 (2) (2016) 198–217, http://dx.doi.org/10.1002/nme.5283.

[32] A. Alla, C. Gräßle, M. Hinze, A residual based snapshot location strategy for POD in distributed optimal control of linear parabolic equations, IFAC-PapersOnLine 49 (8) (2016) 13–18, http://dx.doi.org/10.1016/j.ifacol.2016.07.411.
[33] M. Brand, Fast low-rank modifications of the thin singular value decomposition, Linear Algebra Appl. 415 (1) (2006) 20–30, http://dx.doi.org/10.1016/j.laa.2005.07.021.
[34] C. Vreugdenhil, B. Koren (Eds.), Numerical Methods for Advection-Diffusion Problems, Vieweg, Germany, 1993.
[35] M. Tiberga, A. Hennink, J.L. Kloosterman, D. Lathouwers, A high-order discontinuous Galerkin solver for the incompressible RANS equations coupled to the k-ϵ turbulence model, Comput. & Fluids 212 (2020) 104710, http://dx.doi.org/10.1016/j.compfluid.2020.104710.
[36] S. Lorenzi, A. Cammi, L. Luzzi, G. Rozza, POD-galerkin method for finite volume approximation of navier–stokes and RANS equations, Comput. Methods Appl. Mech. Engrg. 311 (2016) 151–179, http://dx.doi.org/10.1016/j.cma.2016.08.006.
[37] O. Botella, R. Peyret, Benchmark spectral results on the lid-driven cavity flow, Comput. & Fluids 27 (4) (1998) 421–433, http://dx.doi.org/10.1016/s0045-7930(98)00002-4.
[38] J.J. Duderstadt, L.J. Hamilton, Nuclear Reactor Analysis, John Wiley & Sons, Inc., 1976.