Advanced Modeling and Simulation in Engineering Sciences
a SpringerOpen Journal

RESEARCH ARTICLE

Open Access

CrossMark

PEBL-ROM: Projection-error based local reduced-order models

David Amsallem[1*†] and Bernard Haasdonk[2†]

*Correspondence:
amsallem@stanford.edu
†David Amsallem and Bernard
Haasdonk equally contributed to
this work
[1]Department of Aeronautics and
Astronautics, Stanford University,
Stanford, CA 94305, USA
Full list of author information is
available at the end of the article

Abstract

Projection-based model order reduction (MOR) using local subspaces is becoming an increasingly important topic in the context of the fast simulation of complex nonlinear models. Most approaches rely on multiple local spaces constructed using parameter, time or state-space partitioning. State-space partitioning is usually based on Euclidean distances. This work highlights the fact that the Euclidean distance is suboptimal and that local MOR procedures can be improved by the use of a metric directly related to the projections underlying the reduction. More specifically, scale-invariances of the underlying model can be captured by the use of a true projection error as a dissimilarity criterion instead of the Euclidean distance. The capability of the proposed approach to construct local and compact reduced subspaces is illustrated by approximation experiments of several data sets and by the model reduction of two nonlinear systems.

Keywords: Model order reduction, Reduced basis methods, Local bases

Background

Projection-based model-order reduction (MOR) is an indispensable tool for accelerating large-scale computational procedures and enabling their solutions in real-time. This class of approaches proceeds by restricting the solution to a subspace of the entire solution space, resulting in a much smaller set of equations. Many problems, however, are characterized by distinct physical regimes within a given simulation. Among those, one can mention the transition from laminar to turbulent flows, bifurcation of solutions and moving features such as shocks and discontinuities. These simulations are particularly difficult to reduce using classical projection-based MOR as they may require the projection onto large subspaces. These considerations have motivated the recent development of novel local model reduction approaches in which smaller local subspaces are defined and the reduced-order models marches from one subspace to another one within each single simulation [1–3]. Local subspaces can be defined in time [1,2], parameter space [4–6], solution features [7] or state-space [3,6,8–10].

In the local MOR context, many approaches are based on a notion of distance in order to (1) partition solutions and construct local subspaces offline and (2) determine online which subspace is currently used to define the reduced order model (ROM) solution. Although the choice of distance measure is particularly important in these procedures, this choice has not been yet the subject of detailed studies. More specifically, most approaches are

based on the Euclidean distance and this choice may be suboptimal as a dissimilarity measure in the context of local MOR. For instance, a Euclidean distance defined in time typically fails at recognizing periodic phenomena as well as phase shift. Similarly, a basis selection using Euclidean or anisotropic distances in the parameter space cannot identify cases where different parameters lead to identical solutions. On the other hand, a Euclidean distance in the state-space is able to recognize the two aforementioned classes of phenomena leading to similar or identical solutions. However, a Euclidean distance in the state-space does not recognize the linear nature of projections. More specifically, if two snapshots are scaled versions of each other, they can be captured by a unique low-dimensional subspace but the two snapshots may be very distant in the state-space when the measure of distance is the Euclidean norm.

These considerations underline the fact that current local MOR procedures may result in approximating local subspaces that are suboptimal or redundant, leading to unnecessarily large reduced bases. In the present work, a novel local MOR approach is presented. It closely follows the general locality in state-space approach developed in [3,8,9], but is here based on the true projection error as a natural dissimilarity measure. The proposed approach both reflects the nature of approximation in linear spaces, as well as explicitly captures effects of scale-invariance in models. It is based on an extension of the *hp*-RB approach [11–13] now using the true projection error as a partitioning criterion for a given set of snapshots. The procedure partitions the set of snapshot by the construction of a binary tree structure. Each leaf is a cluster of snapshots which is subsequently reduced by proper orthogonal decomposition (POD).

This paper is organized as follows. In the next section the proposed projection error based local ROM approach, PEBL-ROM, is developed and is compared to the *k*-means based local ROM procedure, KML-ROM. Numerical experiments are conducted in the subsequent section, highlighting the capability of the proposed PEBL-ROM approach to construct small and optimal local reduced-order models. In particular, approximation experiments on toy and real simulation data are presented together with MOR results for two nonlinear dynamical systems. Finally, conclusions are given in the last section.

Methods

Data approximation and nonlinear MOR with local bases

The local MOR framework is presented in this section together with notations and notions in the context of data approximation and nonlinear MOR. A set of training data $\mathcal{U} = \{\mathbf{u}_j\}_{j=1}^{n_u} \subset \mathbb{R}^n$ of n_u instances or so-called snapshots in state space of dimension n is assumed to be available. In the context of nonlinear MOR, such a dataset is typically obtained by suitable sampling of the solution trajectory of a—say time discrete—dynamical system of the form

$$\mathbf{u}(t_{i+1}) = \mathbf{f}(\mathbf{u}(t_i)), \quad \mathbf{u}(0) = \mathbf{u}_{init}, \tag{1}$$

for $i = 0, \ldots, K - 1$, where $0 = t_0 < t_i < \ldots < t_K = T$ denote the time instances, $\mathbf{f}$ represents a general nonlinear mapping and $\mathbf{u}_{init}$ is the initial condition. In addition to this single deterministic system, parameters can also enter the system and hence the sampling of snapshots does typically involve both the choice of time instances and parameter values. The choice of sampling parameters is crucial to the definition of an accurate parametric ROM but is not the focus of the present paper and the reader is referred to the following

references [14–16]. The dynamical system, suitable solver and sampling procedure are assumed to be given in the present work. The goal of this paper is to present a framework in order to approximate the data and associated dynamical system by local approximation spaces. Computationally, such approximation spaces are represented by a suitable basis matrix $\boldsymbol{\Phi} \in \mathbb{R}^{n \times r}$ and the approximation space Y is the column span of this basis matrix $Y = \mathrm{colspan}(\boldsymbol{\Phi})$. Such a matrix $\boldsymbol{\Phi}$ is then subsequently referred to as a reduced order basis (ROB). A typical method for constructing a global approximation space is the proper orthogonal decomposition (POD) [17]. In this context, $\boldsymbol{\Phi} = \mathrm{POD}(\mathcal{U}, \varepsilon_{POD}) \subset \mathbb{R}^{n \times r}$ will subsequently denote the computation of a POD of the snapshot set $\mathcal{U}$ with relative energy error $\varepsilon_{POD} > 0$:

$$\mathrm{POD}(\mathcal{U}, \varepsilon_{POD}) := \arg \min_{\substack{\boldsymbol{\Phi} \in \mathbb{R}^{n \times r(\varepsilon_{POD})} \\ \boldsymbol{\Phi}^T \boldsymbol{\Phi} = I}} \sum_{i=1}^{n_u} \|\boldsymbol{u}_i - \mathbf{P}_Y \boldsymbol{u}_i\|_2^2.$$

Here, Y denotes the space spanned by the basis $\boldsymbol{\Phi}$ and $\mathbf{P}_Y = \boldsymbol{\Phi}\boldsymbol{\Phi}^T$ is the orthogonal projection operator $\mathbf{P}_Y : \mathbb{R}^n \to Y$. In particular, $\boldsymbol{\Phi}$ is a matrix with orthogonal columns minimizing the mean projection error of the given data projected onto the subspace spanned by $\boldsymbol{\Phi}$. The dimension $r = r(\varepsilon_{POD})$ is in practice chosen such that the ratio of the sum of the squared projection errors divided by the sum of the squared norm of the data is smaller than ε_{POD}. For more details on POD, also known as Principal Component Analysis, the reader is referred to [17–19]. With such a basis matrix $\boldsymbol{\Phi}$ at hand, $E_P(\mathbf{u}, \boldsymbol{\Phi})$ denotes the true orthogonal projection error of a state vector $\mathbf{u}$ onto the space Y spanned by the basis $\boldsymbol{\Phi}$ as

$$E_P(\mathbf{u}, \boldsymbol{\Phi}) := \min_{\mathbf{v} \in \mathrm{colspan}(\boldsymbol{\Phi})} \|\mathbf{u} - \mathbf{v}\|_2 = \|\mathbf{u} - \mathbf{P}_Y \mathbf{u}\|_2.$$

A typical nonlinear ROM is obtained by a Petrov Galerkin projection procedure. First, an approximation $\hat{\mathbf{u}} := \boldsymbol{\Phi}\mathbf{u}_r$ is chosen for the state where the vector $\mathbf{u}_r \in \mathbb{R}^r$ of reduced coordinates is introduced for all time instants. Then, a second projection basis $\boldsymbol{\Psi} \in \mathbb{R}^{n \times r}$ is chosen. Two popular choices arise in practice for nonlinear dynamical systems: (a) the first choice is to consider $\boldsymbol{\Psi} = \boldsymbol{\Phi}$ simplifying the reduction to a Ritz–Galerkin projection and (b) the second choice is to use a least-square residual minimization [20] arising from the choice $\boldsymbol{\Psi} = \frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\mathbf{u})\boldsymbol{\Phi}$. This second choice will be considered in the nonlinear MOR numerical experiments of this paper.

The choice of local projection bases depends in practice on the current snapshot $\boldsymbol{\Phi}_i := \boldsymbol{\Phi}(\mathbf{u}(\hat{t}_i))$, hence the projected local reduced system is solved for each time step $i = 1, \ldots, K$

$$\boldsymbol{\Psi}_i^T \boldsymbol{\Phi}_i \mathbf{u}_r(t_{i+1}) = \boldsymbol{\Psi}_i^T \mathbf{f}(\boldsymbol{\Phi}_i \mathbf{u}_r(t_i)), \quad i = 0, \ldots K - 1, \tag{2}$$

$$\mathbf{u}_r(t_0) = \boldsymbol{\Phi}_i^T \mathbf{u}_{init}. \tag{3}$$

This reduced system is denoted as ROM and essentially is a low dimensional system of r equations where typically $r \ll n$. It is well known, however, that a computational acceleration is still not always obtained by this procedure, as the approximate state $\boldsymbol{\Phi}_i \mathbf{u}_r(t_i)$ needs to be reconstructed and the full nonlinearity evaluated. Several sparse sampling procedures, sometimes denoted hyperreduction techniques, have been developed that allow to approximate the evaluation of $\mathbf{f}$ in order to accelerate these computations such as the Empirical Interpolation [21,22] or GNAT [23]. However, this additional approximation

step is omitted in this paper as its main purpose is to assess the approximation quality of the local MOR approach of interest. The four-step structure as developed in [3] for general local ROM approaches is recalled as follows:

1. Collection of snapshots from training simulations.
2. Clustering of the snapshots into k clusters.
3. Construction of a local reduced basis for each cluster using POD.
4. Construction of a ROM for each cluster.

The main focus of the current proposed PEBL-ROM approach is step two and three, the partitioning and construction of local reduced order models making use of the projection error. The method is based on a hierarchical partitioning of the state space based on a binary tree structure. Subsequentially, as a reference method, we recall a procedure developed in [3,8,9] which is based on the classical k-means clustering (KML-ROM).

Projection-error based local ROM (PEBL-ROM)

In this section, a new approach for local MOR using the true projection error is proposed as a variant of the hp-RB approach [11,12] in combination with POD. Note that other types of error measures already have been used in partitioning procedures, i.e., an RB error estimator in the hp-RB approach [11] or the empirical interpolation error in the implicit partitioning approach for function approximation [10].

The offline phase of the PEBL-ROM procedure consists of two stages and is summarized in the pseudo-code of Algorithm 1. As input quantities, the proposed algorithm requires the set of snapshots to be processed and accuracy thresholds for the bisection procedure and POD be specified. In stage 1 of the algorithm, a binary tree structure is constructed. Its nodes consist of anchor points that are a subset of the training snapshots. This tree is associated to a non-regular consecutive bisection of the state space. The bisection is defined by comparing the projection errors of new vectors to the corresponding 1D spaces spanned by the anchor points. This partitioning of the state space therefore defines a partitioning of the training snapshots. In stage 2 of the procedure, local bases are generated by POD separately applied to each of the leaf snapshot sets.

The output of the proposed algorithm consists of the binary tree composed of the anchor points and the local bases associated to the leaves of the tree.

The online phase then directly follows and is given in Algorithm 2: Given the bisection tree and the set of local bases constructed in the offline phase as well as a query current state **u**, the tree is traversed depending on the minimum projection error associated with the 1D spaces corresponding to the two candidate anchor points. When reaching a leaf node the corresponding local basis is returned as ROB and can be used for approximation. We anticipate two possibile applications and possible choices for these query states u: First pure function approximation, where u is a given function, the local basis is determined by Algorithm 2 and is used for approximation by orthogonal projection. Second, in dynamical ROM simulation, the current state might be the query, the online algorithm then determines the local basis, which then is used for computing the next time step.

The practical choice of ε_{bisect} is problem dependent. One way of motivated choice for this parameter is realizing the monotonicity of the map $\varepsilon_{bisect} \mapsto k$ by the PEBL-ROM offline phase, and choose ε_{bisect} via the number of desired clusters k_d. This means one can

---

Algorithm 1 Offline phase of the PEBL-ROM procedure

---

function $[\mathcal{T}, \{\boldsymbol{\Phi}^{(i)}\}_{i=1}^{k}] = \text{PEBL\_ROM\_offline}(\mathcal{U}, \varepsilon_{bisect}, \varepsilon_{POD})$
Input: training snapshot set $\mathcal{U}$, target accuracy for bisection ε_{bisect} and POD ε_{POD}
Output: binary tree of anchor points $\mathcal{T}$ and set of local bases $\{\boldsymbol{\Phi}^{(i)}\}_{i=1}^{k}$

 1: $\{$ Stage 1: generation of bisection tree with accuracy $\varepsilon_{bisect}\}$
 2: $\mathbf{u}^{(1)} := \text{argmax}_{\mathbf{u}\in\mathcal{U}} \|\mathbf{u}\|_2, \mathcal{T} := \{\mathbf{u}^{(1)}\}$
 3: $\mathcal{U}^{(1)} := \mathcal{U}, \varepsilon^{(1)} := \max_{\mathbf{u}\in\mathcal{U}^{(1)}} E_P(\mathbf{u}, \mathbf{u}^{(1)})$
 4: **while** $\{i = 1, \dots, |\text{leaves}(\mathcal{T})|$ with $\varepsilon^{(i)} > \varepsilon_{bisect}\}$ is not empty **do**
 5:    search maximum error leaf index $i = \text{argmax}_{i=1,\dots,|\text{leaves}(\mathcal{T})|} \varepsilon^{(i)}$
 6:    search next anchor point $\mathbf{u}_{new} = \text{argmax}_{\mathbf{u}\in\mathcal{U}^{(i)}} E_P(\mathbf{u}, \mathbf{u}^{(i)})$
 7:    divide $\mathcal{U}^{(i)}$ into $\mathcal{U}_{new} := \{\mathbf{u} \in \mathcal{U}^{(i)}|E_P(\mathbf{u}, \mathbf{u}^{(i)}) > E_P(\mathbf{u}, \mathbf{u}_{new})\}$ and $\mathcal{U}^{(i)} := \mathcal{U}^{(i)}\backslash\mathcal{U}_{new}$
 8:    insert $(\mathbf{u}^{(i)}, \mathbf{u}_{new})$ into $\mathcal{T}$ as children of leaf anchor point $\mathbf{u}^{(i)}$ and associate their snapshot sets $\mathcal{U}^{(i)}$ and $\mathcal{U}_{new}$
 9:    recompute leaf errors $\varepsilon^{(i)} := \max_{\mathbf{u}\in\mathcal{U}^{(i)}} E_P(\mathbf{u}, \mathbf{u}^{(i)})$ for $i = 1, \dots, |\text{leaves}(\mathcal{T})|$
10: **end while**
11: $\{$ Stage 2: generation of local POD bases$\}$
12: $k := |\text{leaves}(\mathcal{T})|$
13: **for** $i = 1, \dots k$ **do**
14:    compute $\boldsymbol{\Phi}^{(i)} := \text{POD}(\mathcal{U}^{(i)}, \varepsilon_{POD})$
15: **end for**

Algorithm 2 Online phase of the PEBL-ROM procedure

---

function $\boldsymbol{\Phi} = \text{PEBL\_ROM\_online}(\mathbf{u}, \mathcal{T}, \{\boldsymbol{\Phi}^{(i)}\}_{i=1}^{k})$
Input: State $\mathbf{u}$, binary tree of anchor point $\mathcal{T}$, set of local bases $\{\boldsymbol{\Phi}^{(i)}\}_{i=1}^{k}$
Output: Local basis $\boldsymbol{\Phi} = \boldsymbol{\Phi}^{(i)}$

 1: $\mathbf{u}^* = \text{root}(\mathcal{T})$
 2: **while** not isleaf$(\mathbf{u}^*)$ **do**
 3:    $[\mathbf{u}_0^*, \mathbf{u}_1^*] = \text{children}(\mathbf{u}^*, \mathcal{T})$
 4:    **if** $E_P(\mathbf{u}, \mathbf{u}_0^*) < E_P(\mathbf{u}, \mathbf{u}_1^*)$ **then**
 5:       $\mathbf{u}^* := \mathbf{u}_0^*$
 6:    **else**
 7:       $\mathbf{u}^* := \mathbf{u}_1^*$
 8:    **end if**
 9: **end while**
10: $i := \text{leafindex}(\mathbf{u}^*) \in 1, \dots, k$

---

start from some extremely large value (resulting in $k = 1$) and some close to zero value for ε_{bisect} (resulting in $k = n_u$), perform some (logarithmical) interval division algorithm while repeatedly performing the offline phase to detect the parameter ε_{bisect} that gives $k = k_d$. We also adopted to this procedure in our experiments.

In the case of the MOR for instationary dynamical systems, switching between clusters must be ensured. It is obvious that with the fully refined tree and resulting 1D spaces, the space resulting from a given query snapshot will exactly be the 1D space consisting of that snapshot. Such a dynamic simulation will never result in a different space and no switching will occur. Therefore, in the MOR framework and following [3,8,9], a variant of the algorithm will be considered in which all local snapshot sets used for POD will be

based on increments of snapshots at the point considered. This means, that the anchor points are still selected from state snapshots, but the local ROB are generated from a POD of the corresponding increment snapshots, where an increment snapshot is simply the difference between two consecutive snapshots of the dynamic simulation. The use of increment snapshots is demonstrated theoretically and in practice in [3,8,9] for MOR using local bases.

In the case of dynamical ROM simulation, it is essential that no computational step in the online phase scales with the dimension n of the high-dimensional space. Algorithm 2 can be executed by computing projection errors with a complexity that does not scale with n by one of two approaches: (1) by an offline-online decomposition or (2) by introducing a surrogate inner product acting only on the sampled mesh elements as developed in [9]. For completeness, we give the essential idea for the offline-online decomposition of the projection error computation: Assuming that the anchor point $\boldsymbol{u}^* \in \mathbb{R}^n$ is normalized, the projection error of a reduced online query state $\boldsymbol{\Phi u}_r$ with some basis $\boldsymbol{\Phi} \in \mathbb{R}^{n \times r}$ and reduced coefficient vector $\boldsymbol{u}_r \in \mathbb{R}^r$ can be explicitly computed by an orthogonal projection:

$$E_P(\boldsymbol{\Phi u}_r, \boldsymbol{u}^*)^2 = \|\boldsymbol{\Phi u}_r - \langle \boldsymbol{\Phi u}_r, \boldsymbol{u}^* \rangle \boldsymbol{u}^*\|^2 = \boldsymbol{u}_r^T \boldsymbol{\Phi}^T \boldsymbol{\Phi u}_r - ((\boldsymbol{u}^*)^T \boldsymbol{\Phi u}_r)^2$$

So offline, the inner product matrix $\boldsymbol{\Phi}^T \boldsymbol{\Phi}$ and all anchor point projections $\boldsymbol{\Phi}^T \boldsymbol{u}^*$ need to be precomputed with storage complexity $\mathcal{O}(r^2 + kr)$. Online, the projection error computations can be realized in $\mathcal{O}(r^2)$ per node.

As mentioned in the previous section, in case of nonlinear ROM simulation, hyperreduction needs to be performed in order to obtain computational acceleration. First, if using a global hyperreduction Ansatz (i.e., single global sample mesh for GNAT, or single collateral basis set and interpolation points for DEIM), no changes in Algorithm 1 or 2 are required. as they only address the (Galerkin) projection stage, but not the nonlinearity approximation. However, the use of local hyperreduction (i.e., local interpolation bases, submeshes, etc.) would require essential extensions of the offline and online phases. We refrain from detailed presentation of these extensions, as we do not make use of that in the experiments, but the extensions can be obtained by following the ideas of [9].

k-means local ROM (KML-ROM)

As a reference method, the KML-ROM approach is considered. It proceeds by applying the k-means algorithm using the Euclidean error for clustering the training state-snapshot set, then using POD on each leaf snapshot set used for constructing local spaces. This approach has been successfully applied in [3,8,9] for the reduction of nonlinear computational fluid dynamics problems. The number of clusters is here specified as an input parameter. Then, the iterative k-means procedure clusters snapshots that are close in the Euclidean norm sense. For dynamical systems, in addition to the snapshots on each cluster, a fraction f_{add} (typically $f_{\text{add}} \approx 10\,\%$) of neighboring snapshots is added to each snapshot set, resulting in overlapping clusters [3,8]. This choice was demonstrated to result in more robust local ROMs. After clustering, POD is applied to compress each local snapshots set. The approach is summarized in Algorithm 3.

Similarly as for the PEBL-ROM approach, two use cases of this algorithm can be distinguished. First, for approximation experiments, the POD is applied to state snapshots. Second, for dynamic MOR experiments, the POD is applied to local increment snapshots and not the state snapshots themselves.

Algorithm 3 Offline phase of the KML-ROM procedure

---

function $[\{\mathbf{u}_c^{(i)}\}_{i=1}^k \{\mathbf{\Phi}^{(i)}\}_{i=1}^k] = \text{KML\_ROM\_offline}(\mathcal{U}, k, \varepsilon_{POD})$
Input: training snapshot set $\mathcal{U} = \{\mathbf{u}_j\}_{j=1}^{n_u}$, number of clusters k and POD accuracy ε_{POD}
Output: set of cluster centroids $\{\mathbf{u}_c^{(i)}\}_{i=1}^k$ and set of local bases $\{\mathbf{\Phi}^{(i)}\}_{i=1}^k$

1: Randomly choose k snapshots as centroids $\mathbf{u}_c^{(i)}$, $i = 1, \ldots, k$
2: **while** the centroids have not converged **do**
3:     **for** $j = 1, \ldots, |\mathcal{U}|$ **do**
4:         Set $c_j = \text{argmin}_{\ell=1,\ldots,k} \|\mathbf{u}_j - \mathbf{u}_c^{(\ell)}\|_2$
5:     **end for**
6:     **for** $i = 1, \ldots, k$ **do**
7:         Update the centroid

$$\mathbf{u}_c^{(i)} = \left(\sum_{j=1}^{|\mathcal{U}|} \mathbf{1}_{c_j=i} \mathbf{u}_j \right) \Big/ \left(\sum_{j=1}^{|\mathcal{U}|} \mathbf{1}_{c_j=i} \right)$$

8:     **end for**
9: **end while**
10: **for** $j = 1, \cdots, |\mathcal{U}|$ **do**
11:     Assign $\mathbf{u}_j$ to the cluster $\mathcal{U}^{(c_j)}$
12: **end for**
13: **for** $i = 1, \ldots k$ **do**
14:     compute $\mathbf{\Phi}^{(i)} := \text{POD}(\mathcal{U}^{(i)}, \varepsilon_{POD})$
15: **end for**

In the online phase, the current cluster is determined by computing the Euclidean distance of the current state to each cluster centroid and selecting the closest one. This step is summarized in Algorithm 4.

Algorithm 4 Online phase of the KML-ROM procedure

---

function $\mathbf{\Phi} = \text{KML\_ROM\_online}(\mathbf{u}, \{\mathbf{u}_c^{(i)}\}_{i=1}^k, \{\mathbf{\Phi}^{(i)}\}_{i=1}^k)$
Input: State $\mathbf{u}$, cluster centroids $\{\mathbf{u}_c^{(i)}\}_{i=1}^k$, set of local bases $\{\mathbf{\Phi}^{(i)}\}_{i=1}^k$
Output: Local basis $\mathbf{\Phi} = \mathbf{\Phi}^{(i)}$

1: **for** $\ell = 1, \ldots, k$ **do**
2:     Compute the Euclidean distance

$$d_\ell = \|\mathbf{u} - \mathbf{u}_c^{(\ell)}\|_2$$

3: **end for**
4: Find $i = \text{argmin}_{\ell=1\cdots,k} d_\ell$

Conceptional discussion

The POD in its elementary definition reflects the linear approximation nature of MOR by minimizing the mean true projection error of a given set of snapshots. Therefore, the PEBL-ROM procedure seems a natural extension in the context of local projection-

based approximation. The PEBL-ROM approach fully reflects the projection nature of the approximation task both in the partitioning, the local space construction as well as the online partition selection.

Some remarks can be made when comparing the PEBL-ROM and the KML-ROM algorithms. First some limiting cases can be considered: In the case of $k = n_u$, both procedures generate the maximum number of k clusters and optimal 1D approximation spaces allowing the training error to be zero. This highlights the asymptotic optimality of both approaches.

Further, a remark concerning the computational complexity can be made. The tree-structure allows a traversal of the local spaces with a lower computational complexity (logarithmic complexity for a perfectly balanced tree and linear complexity in the worst case) when compared to the linear search in a cluster list generated by k-means. Still, as typical values of k are usually modest in the local MOR context, a large CPU discrepancy at the traversal level should not be expected and is not observed in practice.

A final remark can be made about the nestedness property of the local bases. The PEBL-ROM procedure results in a hierarchical partitioning of the training snapshots. Indeed, a fine binary tree can be coarsened by merging children nodes at the parent node level. This constitutes an advantage over the KML-ROM procedure for which the clusters are not nested when varying k. In the case of the PEBL-ROM procedure, the local ROBs are only nested themselves when ε_{POD} is small enough to result in no truncation of the snapshots space.

Results and discussion
Approximation of toy data
In the first set of experiments, the properties of the algorithms are illustrated on artificially generated data of random clouds in $\mathbb{R}^n$ for $n = 1000$.

The first *unimodal* dataset consists of 500 points drawn from a single normal random distribution. The mean is set to $\mathbf{0}$ and the covariance matrix is diagonal with variance $0.1e^{10}$ in the first two dimensions, then exponentially decaying as $0.1e^{10(2-i)}$ for $i = 2, \ldots, n$.

The second *multimodal* dataset consists of a mixture of four normal distributions, each with the same covariance matrix as the unimodal dataset, but different mean values. From each of the four normal distributions, 100 points are drawn.

Figure 1 illustrates the results of the PEBL-ROM procedure on the unimodal dataset for $\varepsilon_{POD} = 10^{-5}$ and different bisection accuracies, i.e., by lowering ε_{bisect} from 0.25 to 0.15. These values were chosen such that 2, 3, 4 and 5 parts were respectively obtained. In the left column of Fig. 1, the 1000-dimensional training data is depicted by projecting it on the first two dimensions corresponding to the directions of maximal variance of the normal distributions. Each point is plotted according to a color chosen for the corresponding local part. The corresponding anchor points are also represented as colored circles with black boundary. In the right column of Fig. 1, the structure of the corresponding trees is displayed with the final local snapshot number and basis sizes indicated at each leaf.

As expected, the number of parts is increasing with lower bisection tolerances. Also, one can observe (despite the differing colors) that the partitions are hierarchical in the sense that a coarser anchor point set is a subset of the refined anchor set. Hence, each part of the refined partition always is completely contained in one part of a coarser partition of
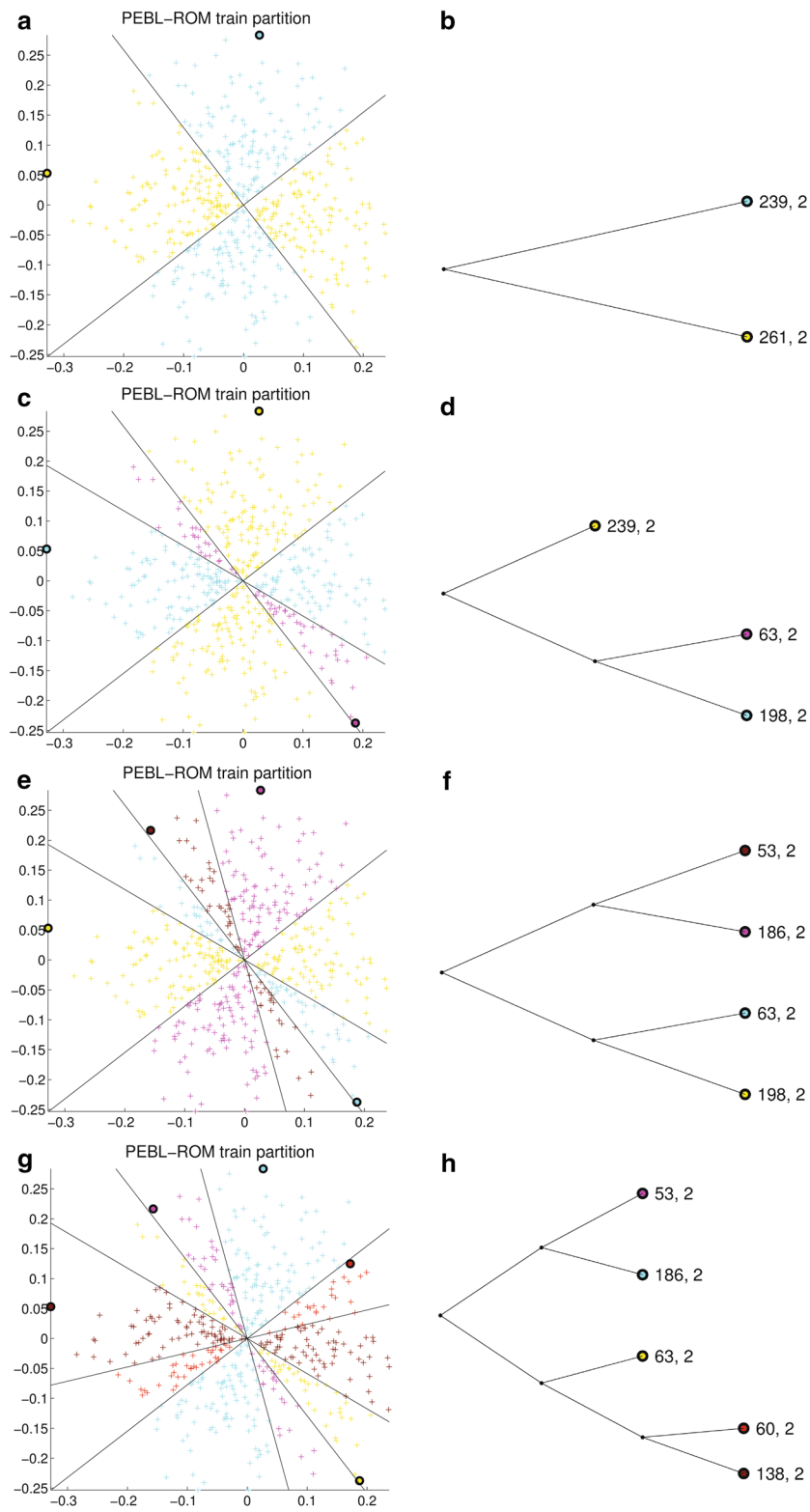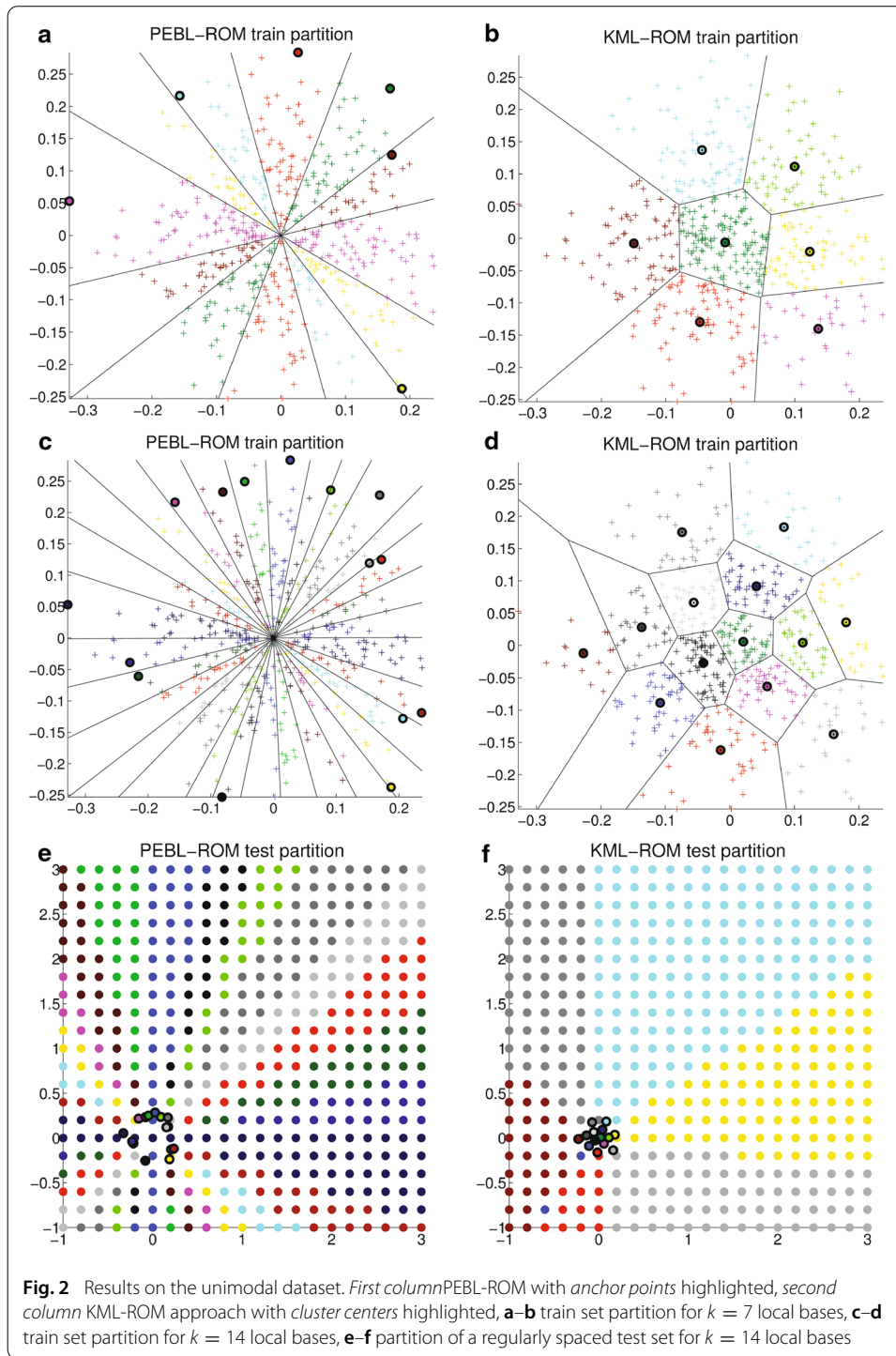
Fig. 1 Results on the unimodal dataset for improving accuracy of the PEBL-ROM procedure. *Left column* training partitions and anchor points for $\varepsilon_{bisect} = 0.25, 0.2, 0.17$ and 0.15, resulting in $k = 2, 3, 4, 5$ parts. *Second column* the corresponding trees with snapshot number and resulting local basis sizes

state space. The partitioning is based on the true projection error, which is reflected in the fact that all clusters are geometrically double-cones centered in the origin. This illustrates the scale invariance of the parts. In particular, points at the opposite side of an anchor point are assigned to the cluster of that anchor point, although these points are maximally distant to this anchor point with respect to the Euclidean distance. Hence, the projection error has a completely different characteristic as the Euclidean distance. As samples with current worst projection error are chosen as new anchor point, it is understandable that these tend to lie at the boundary of the point set and not in the interior.

Figure 2 illustrates a comparison of both the PEBL-ROM and the KML-ROM algorithms on the unimodal dataset. For each of the following experiments, a value ε_{bisect} for the PEBL-ROM procedure is chosen so that it results in an equivalent number k of local bases chosen as input for the KML-ROM procedure. This ensures comparability in terms of identical number of local bases. For subplot a, ε_{bisect} is chosen as $\varepsilon_{bisect} = 0.1$, resulting in $k = 7$ local bases, hence $k = 7$ for the KML-ROM in plot b. For plots c and e, $\varepsilon_{bisect} = 0.05$, resulting in $k = 14$ local spaces, hence the target number of clusters is $k = 14$ for the KML-ROM in plots d and f.

Plot a and c again confirm the insights obtained from the previous refinement experiment, now with slightly larger number of parts $k = 7$ and $k = 14$. In contrast to this, plot b and d illustrate the training set partitions obtained by the KML-ROM algorithm. One can observe how the cluster centers for the k-means based procedure tend to be distributed uniformly with respect to the Euclidean distance. The clusters are actually Voronoi cells of a corresponding Voronoi partitioning. With increasing target cluster number k, the clusters are not nested but rather independent. The rather "circular" shape of the clusters of the KML-ROM algorithm in contrast to the "lengthy" clusters in the PEBL-ROM procedure might indicate that these k-means clusters require more basis vectors than the clusters obtained from the tree-based procedure. This will indeed be visible with subsequent approximation experiments. Comparing the partitions on a test set of regularly distributed points over a considerably larger square domain in plot e and f reveals that the PEBL-ROM procedure makes full use of the k different clusters in the far field, while the k-means algorithm only uses fewer number of clusters in the outer regions, as some clusters are bounded and compact and completely lying in the range of the original training set. This motivates the expectation that the PEBL-ROM procedure might be better generalizing on solution regimes, which have not been included in the training data (e.g., scaled snapshots).
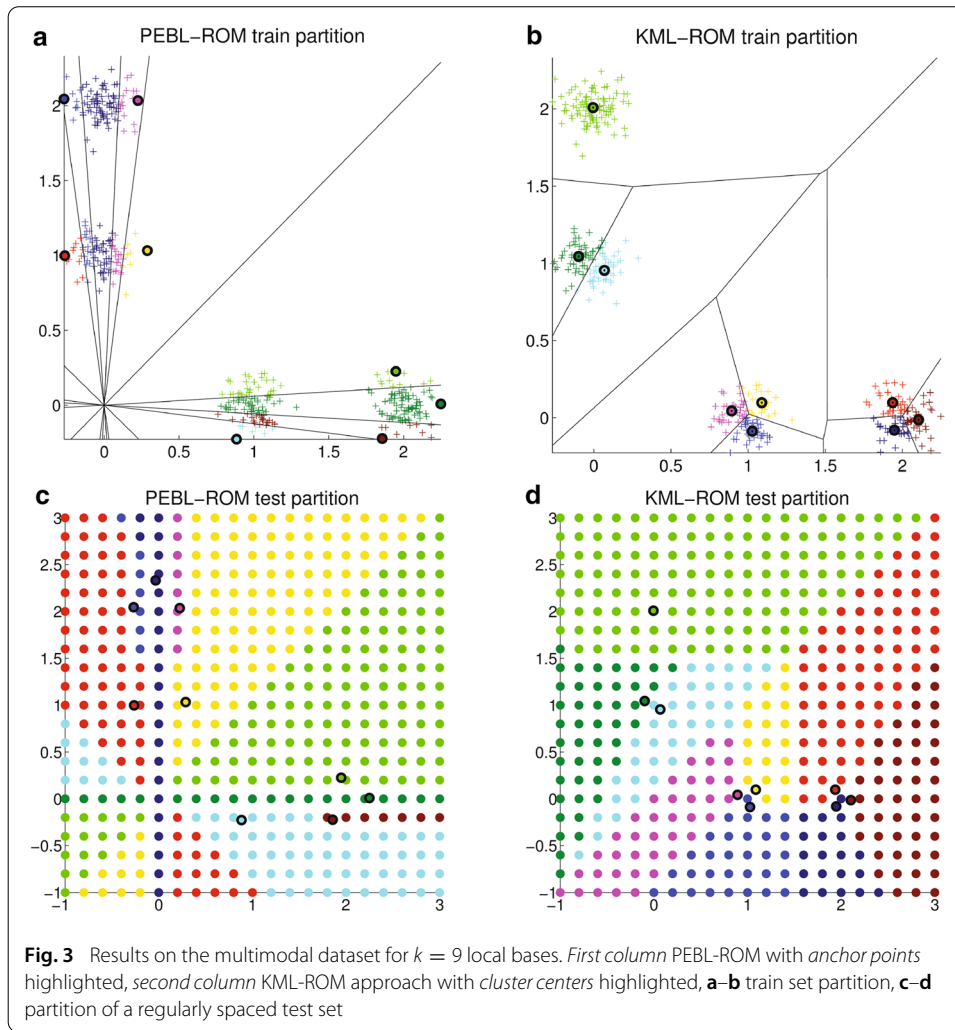
In Figure 3, results are illustrated for the multimodal dataset. The accuracy is set to $\varepsilon_{bisect} = 0.2$, resulting in $k = 9$ clusters. Both algorithms are applied and the training set and the testing set assignments are plotted. Again, one can observe the cone structure and scale-invariance of the PEBL-ROM procedure clusters, while the clusters of the KML-ROM algorithm are differently shaped. The difference between the procedures gets very clear when considering the top left cluster: the KML-ROM algorithm in plot b assigns this to one cluster, while the PEBL-ROM procedure in plot a splits it into 3 subparts, the latter promising better approximation. Indeed, due to the scaling nature of the two left clouds, points from the upper can be very well approximated by points from the lower and vice versa. This is however not captured by the Euclidean distance as the k-means algorithm produced disjoint parts of these two point clouds. In contrast, the PEBL-ROM procedure

Fig. 2 Results on the unimodal dataset. *First column* PEBL-ROM with *anchor points* highlighted, *second column* KML-ROM approach with *cluster centers* highlighted, **a**–**b** train set partition for $k = 7$ local bases, **c**–**d** train set partition for $k = 14$ local bases, **e**–**f** partition of a regularly spaced test set for $k = 14$ local bases

indeed assigns points from the lower left cloud to the same cluster as some points from the upper left cloud.

Approximation of Burgers equation data

Next, approximation experiments are performed on data obtained from a dynamical system, i.e., snapshots of a parameterized 1D Burgers equation. The one-dimensional Burgers

Fig. 3 Results on the multimodal dataset for $k = 9$ local bases. *First column* PEBL-ROM with *anchor points* highlighted, *second column* KML-ROM approach with *cluster centers* highlighted, **a**–**b** train set partition, **c**–**d** partition of a regularly spaced test set

equation with parameterized boundary condition is investigated as in [24]. The equation is

$$\frac{\partial u}{\partial t} + \frac{1}{2}\frac{\partial \left(u^2\right)}{\partial x} = 0, \quad t \in [0, 2.5], \ x \in [0, 5], \tag{4}$$

with Dirichlet boundary condition

$$u(x = 0, t) = u_{BC}. \tag{5}$$

At time $t = 0$, the initial condition is $u(x, t = 0) = 0$. As a result of the non-zero boundary condition, a shock with speed equal to the left boundary value is propagated into the computational domain.

In the subsequent numerical experiments, three training simulations are conducted for the three parameter values $u_{BC} \in \{2, 3, 5\}$. The accuracy of the local model reduction methods will then be assessed for those three conditions as well as three additional testing parameter values $u_{BC} \in \{1.5, 4, 5.5\}$.

The PDE is discretized in space by upwind finite differences with $n = 1000$ nodes and in space by the backward Euler finite difference scheme with a time step $dt = 0.0125$.

Representative solutions for the parameters considered are depicted in Fig. 4 at times $t \in \{0.5, 1.0, 1.5, 2.0, 2.5\}$.

The use of the resulting local bases in MOR will be dealt with later. Here the approximation properties of the local bases based on the true projection error are investigated.

Using the tolerances $\varepsilon_{POD} = 10^{-5}$ and $\varepsilon_{bisect} = 10$ the results for the PEBL-ROM approach are given in the left column of Table 1. Again, in order to be able to compare qualitative results, a value of k for the KML-ROM procedure is chosen so that it generates the same number of partitions. The corresponding results are indicated in the right column of Table 1. The maximum basis size is slightly larger for the PEBL-ROM approach than for the KML-ROM. In contrast, the overall basis size sum and the mean basis size is

Fig. 4 Solutions of the parameterized Burgers equation at $t \in \{0.5, 1.0, 1.5, 2.0, 2.5\}$ for different boundary values u_{BC}: **a–c** training parameters, **d–f** test parameters
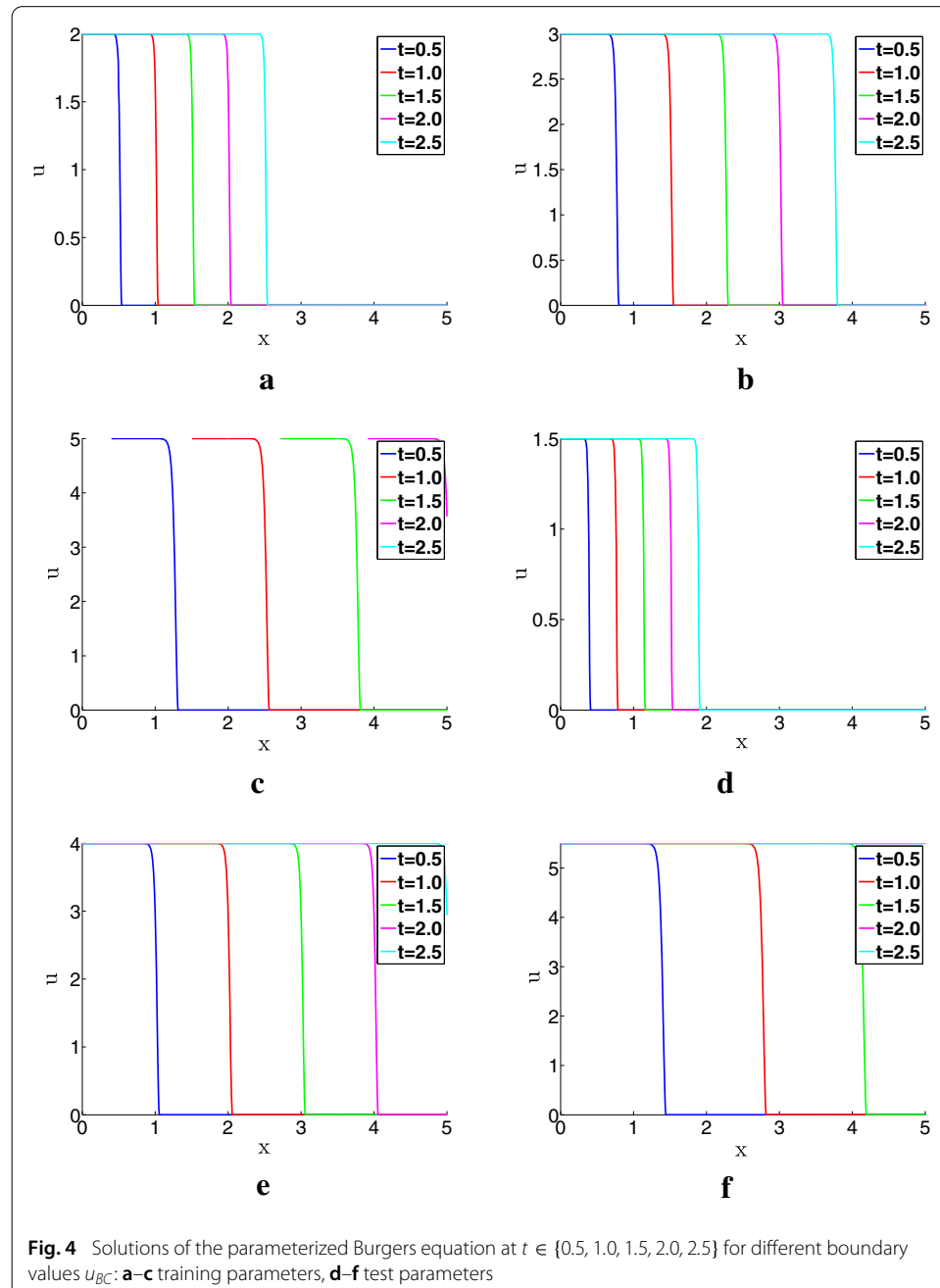
Table 1 Results of the offline phases for the approximation of Burgers data

	PEBL-ROM	KML-ROM
Number of bases	12	12
Sum of overall basis sizes	187	301
Maximum basis size	68	54
Min of basis sizes	4	4
Mean of basis sizes	15.5833	25.0833
Variance of basis sizes	295.3561	217.9015

much larger for the KML-ROM approach. This indicates that the PEBL-ROM procedure generates more compact ROBs than the KML-ROM procedure and few large bases. This is also reflected in the larger variance for the PEBL-ROM procedure. Consequently, with the PEBL-ROM procedure a set of local reduced approximation spaces that requires overall smaller storage is obtained.

In order to illustrate the resulting partitions, some properties of the PEBL-ROM procedure are reported in Fig. 5. In plot a, the shape of the resulting binary tree with the local snapshot set is indicated together with the local basis sizes for each leaf node. An unbalanced binary tree is generated as locally repeated refinements are required. Overall a compression by the POD of about a factor 2–3 is observed. The "lowest" node (dark green) can be observed to compress a large set of 45 snapshots to merely 4 POD modes, which corresponds to the local basis with shock position at the end of the interval, i.e., the snapshots, where the shock has already left the interval and the solutions are expected to be rather smooth. Plot b illustrates for a set of test snapshots, depicted by their shock position and by color coding, which local space is chosen for approximation. It is clearly visible, that the test data is consisting of consecutive snapshots of few number of trajectories. Now, when comparing the colors, one realizes that indeed, the PEBL-ROM procedure chooses similar spaces for snapshots with similar shock position from trajectories of different parameters. For fast moving shocks, the time range is small enough such that the shocks leave the computational domain and, again, many snapshots at these final times are assigned to one cluster representing such "smooth" solutions. Also, the blue node in the tree is outstanding, as it used a remarkable large set of 247 snapshots which cannot be compressed very well, resulting in a 68 dimensional local basis. This node corresponds to early time snapshots over a remarkably large shock position interval (0–300). This is resulting from the fact, that these initial snapshots have quite small norms (the value behind the shock being zero), hence allow quite good approximation by a single basis for a long time. In contrast, the "later" shock positions are represented by much finer clusters, as the snapshots have larger norms, hence larger projection errors indicate an earlier refinement for these shock position regimes.

The corresponding experiment for the test data and the KML-ROM algorithm is given in Fig. 6. One can clearly observe that the partitioning using the Euclidean error considers several shapshots with identical shock position as being dissimilar and assigns different local bases, although essentially, they mainly differ by a scaling.

Quantitative results are now compared concerning approximation quality by determining the relative sum of squared training errors. Using the notation introduced before, the absolute squared training error is defined as
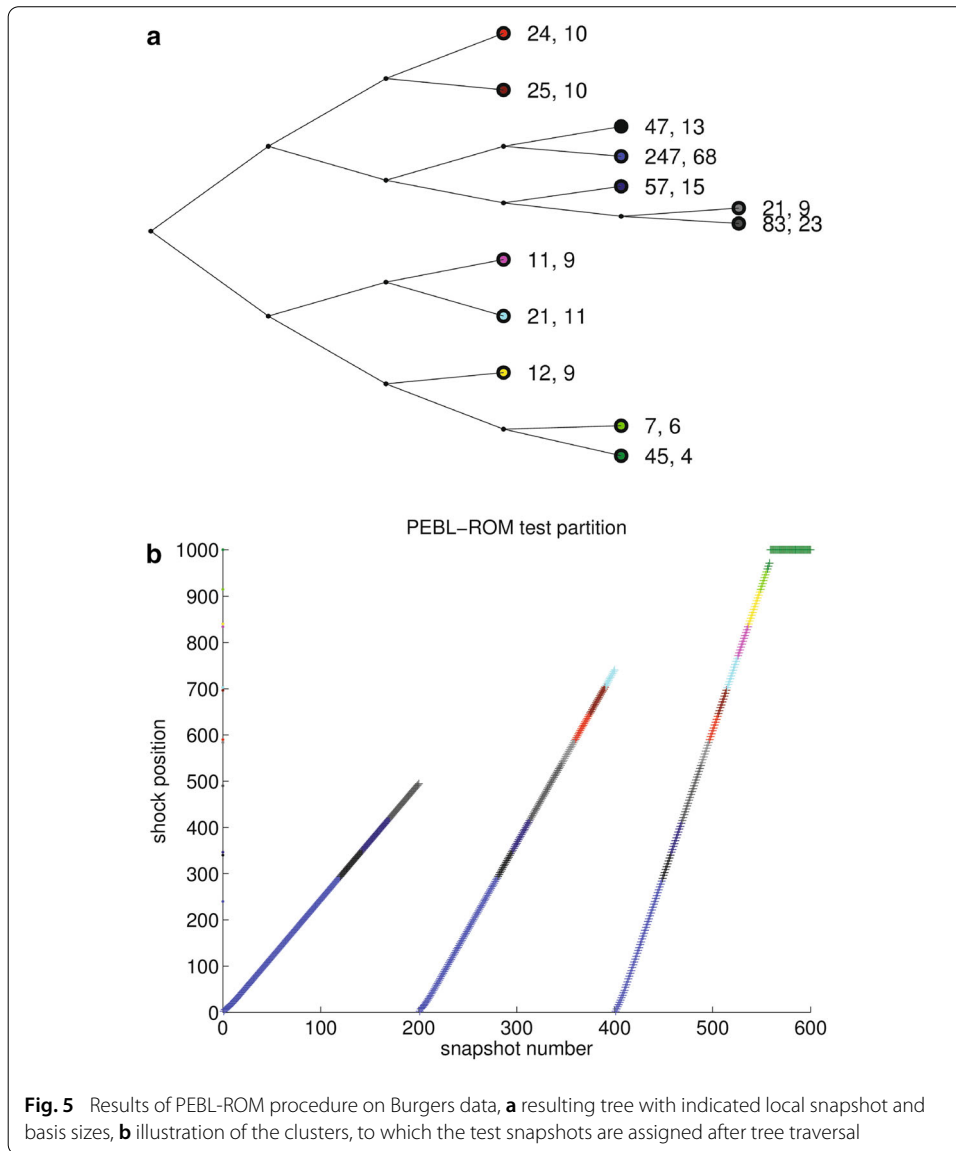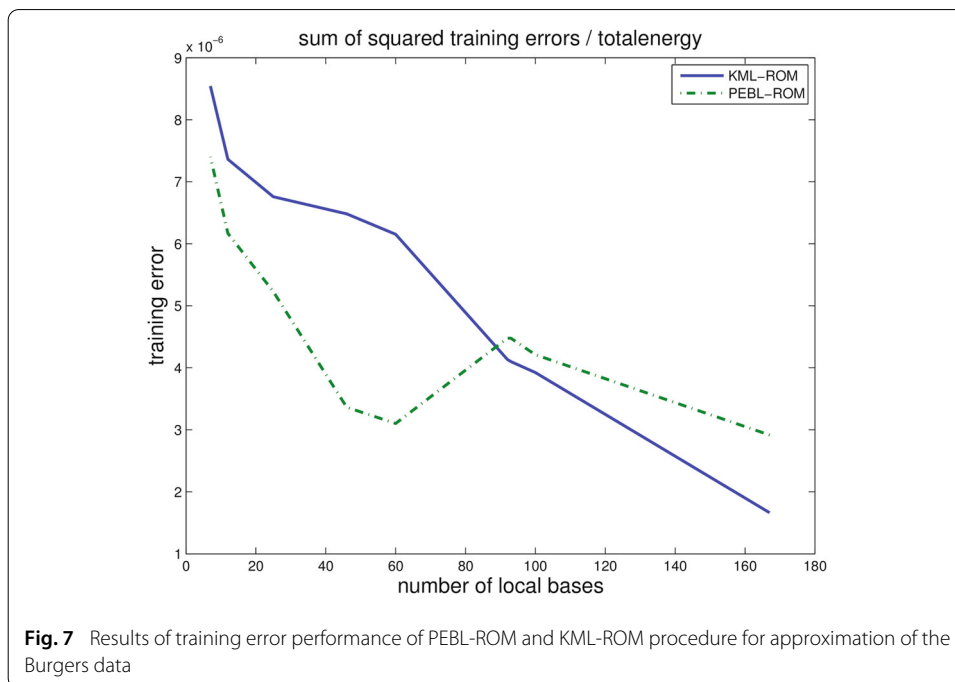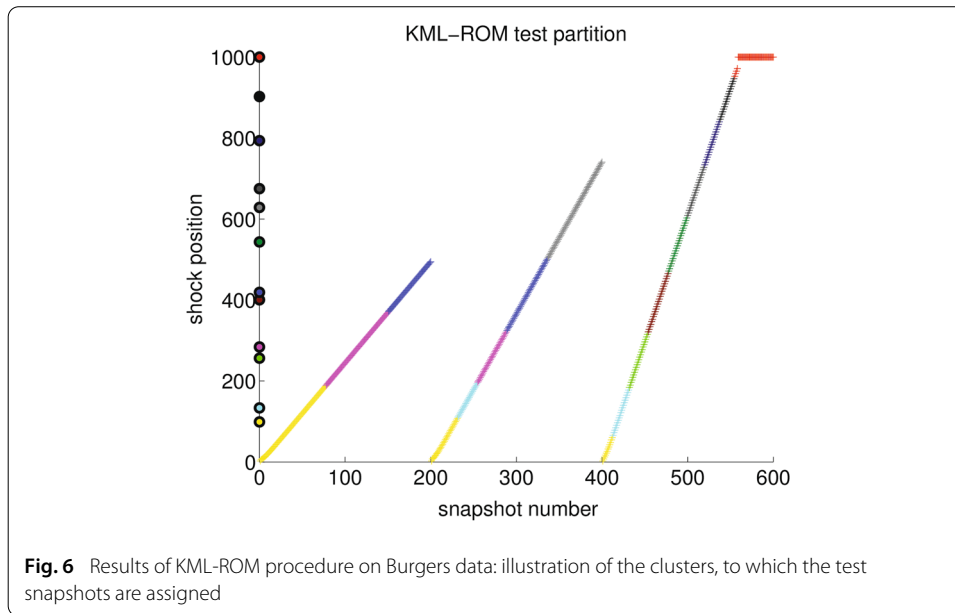
Fig. 5 Results of PEBL-ROM procedure on Burgers data, **a** resulting tree with indicated local snapshot and basis sizes, **b** illustration of the clusters, to which the test snapshots are assigned after tree traversal

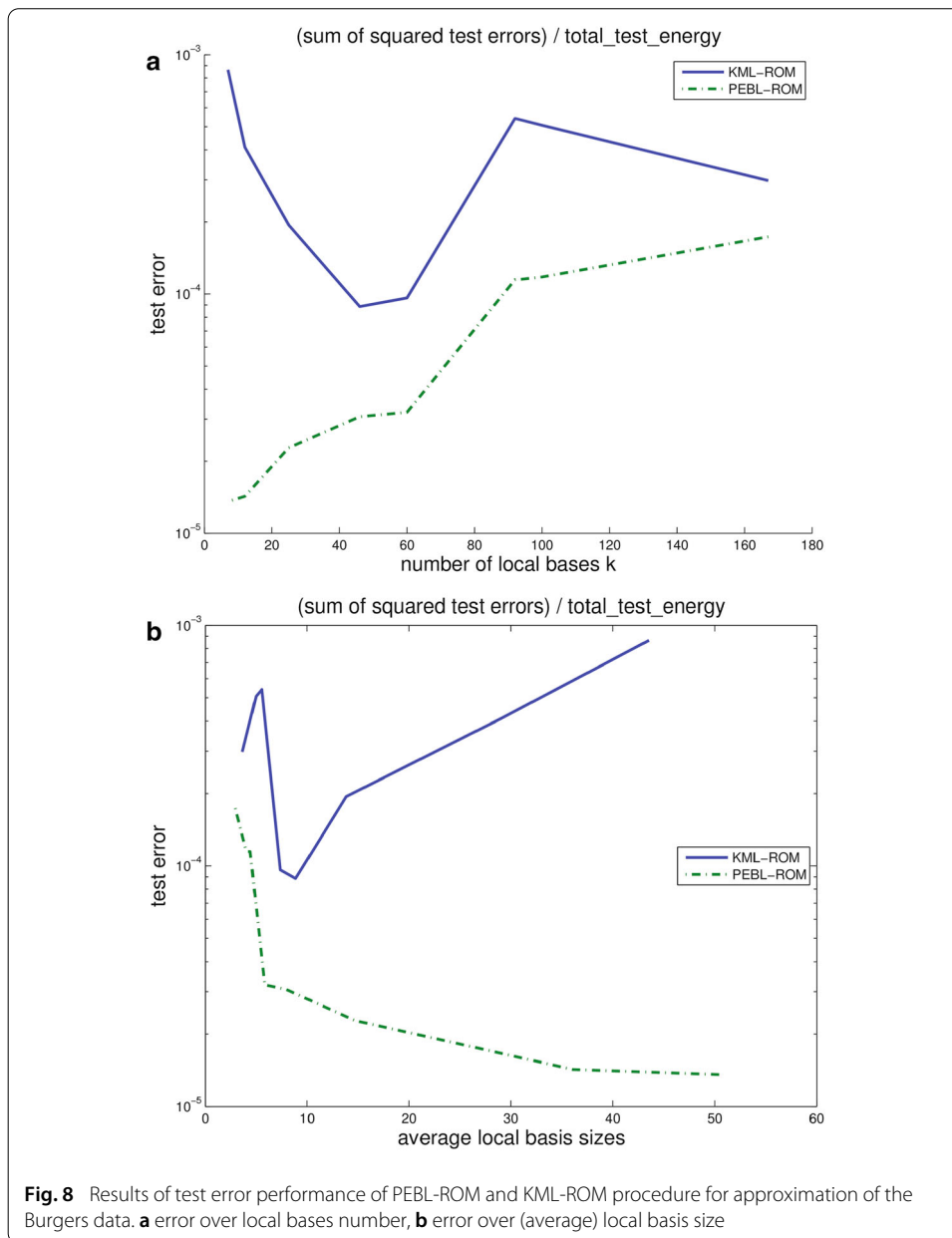$$\sum_{i=1}^{n_u} \|\mathbf{u}_i - P_{Y_i}(\mathbf{u}_i)\|_2^2,$$

where $Y_i := \text{colspan}(\mathbf{\Phi}(\mathbf{u}_i))$ denotes the local space associated with the i th snapshot. This error measure is exactly the quantity that is minimized by the POD procedure. This measure does not involve the approximation results from solving the reduced dynamical system, but purely measures the approximation quality of the local spaces.

While varying the number of local bases (by varying k for the KML-ROM, and by choosing ε_{bisect} for the PEBL-ROM procedure), the results obtained are summarized in Fig. 7. Both approaches result in training errors below the POD accuracy $\varepsilon_{POD} = 10^{-5}$ confirming the training stage correctness. Otherwise, the approaches are very comparable, the PEBL-ROM perhaps being slightly more accurate.

However, this relation becomes much more expressive, if considering a predictive scenario. In the predictive context, analogous experiments are performed using the set of

Fig. 6 Results of KML-ROM procedure on Burgers data: illustration of the clusters, to which the test snapshots are assigned

Fig. 7 Results of training error performance of PEBL-ROM and KML-ROM procedure for approximation of the Burgers data

test snapshots. The results are given in Fig. 8. In a, the relative summed squared test error performance is reported as a function of the number of local bases. In b the error is plotted over the average local basis size. One can observe that the PEBL-ROM procedure clearly outperforms the KML-ROM algorithm by almost one order of magnitude in the relative squared error. This relation is even more clear in case of a small number of local bases or a higher average local basis size. Inspecting the diagram more carefully indicates an increase of the test-error for the PEBL-ROM with increasing number of local bases. We expect that this indicates an overfitting effect, as the training error in the previous figure is simultaneously decreasing.

Fig. 8 Results of test error performance of PEBL-ROM and KML-ROM procedure for approximation of the Burgers data. **a** error over local bases number, **b** error over (average) local basis size

Overall, it can be concluded from these numerical experiments that the PEBL-ROM procedure provides more compact approximation models in the sense of ROB size versus test error. This is due to the expected scaling properties of the Burgers snapshots. This scaling invariance is captured by the true projection error, while it is overseen by the Euclidean distance.

Nonlinear MOR for the Burgers equation

Now, the use of the local reduced bases is investigated in dynamical problems for reduced order simulations. The experiments here are exactly using the same trajectory snapshots from the Burgers model as in the previous section. As explained in the method section clustering is performed on snapshot increments of the training trajectories.

In a first set of experiments, the number of local ROBs is fixed to $k = 7$. The POD energy level of the truncation, ε_{POD}, is varied and the following two local model reduction approaches are compared to build the local bases: (1) KML-ROM clustering with overlapping clusters ($f_{add} = 10\%$) and (2) the proposed PEBL-ROM approach. The accuracy of the resulting reduced-order model solutions is depicted in Fig. 9 for the following mean relative error measuring the discrepancy between the high-dimensional and the reduced trajectories

$$\frac{1}{K+1} \sum_{i=0}^{K} \frac{\|\mathbf{u}(t_i) - \mathbf{\Phi}_i \mathbf{u}_r(t_i)\|_2}{\|\mathbf{u}(t_i)\|_2}$$

In consistency with the paper title, we denote this error "Local ROM Error" in subsequent plots.

One can observe that, for small values of ε_{POD}, the PEBL-ROM approach generally results in more accurate reduced-order models than its k-means counterpart, both for training (top row) and testing parameters (bottom row). Figure 10 shows the error as a function of the average basis size. Again, the PEBL-ROM approach leads to more accurate ROMs.

In a second set of numerical experiments, the truncated POD energy level is fixed to $\varepsilon_{POD} = 10^{-8}$. In that case, the number of local bases is varied from $k = 2$ to 10 and the KML-ROM approach compared to the PEBL-ROM approach. Figure 11 depicts the error as a function of the number of local ROBs. Again, the PEBL-ROM method leads to more

Fig. 9 Local ROM error for Burgers equation as a function of the POD truncated energy for seven local ROBs: KML-ROM (*solid line*) and PEBL-ROM (*dashed dotted line*) for training (*top row*) and test parameters (*bottom row*)

Fig. 10 Local ROM error for Burgers equation as a function of the average ROB size (by varying ε_{POD}) for $k = 7$ local ROBs: KML-ROM (*solid line*) and PEBL-ROM (*dashed dotted line*) for training (*top row*) and test parameters (*bottom row*)

accurate reduced-order models. Figure 12 reports the average ROB size as a function of the number of local bases. It can be observed that the PEBL-ROM approach leads to smaller bases for the same truncation criterion ε_{POD}. This is confirmed by inspecting Fig. 13, where the error is reported as a function of the average ROB dimensionality. The PEBL-ROM approach leads to both smaller and more accurate ROBs.

Nonlinear MOR for a chemical reaction problem

In this second MOR application, the reaction of a premixed H_2-air flame model is studied in two space dimensions. The reaction, $2H_2 + O_2 \rightarrow 2H_2O$, is modeled by the following nonlinear unsteady advection-diffusion-reaction equation [25]:

$$\frac{\partial w}{\partial t} + u \cdot \nabla w - \kappa \Delta w = s(w), \quad x \in [0, L_x] \times [0, L_y], t \in [0, T_{\max}] \tag{6}$$
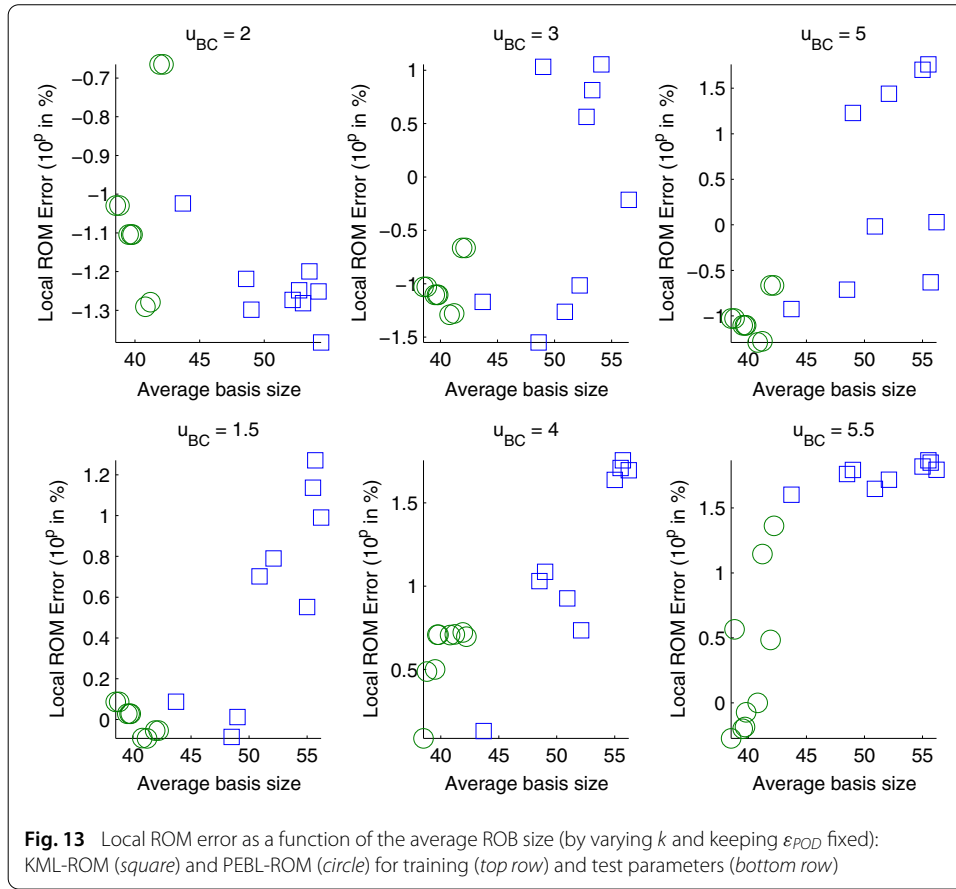
where the state vector

$$w(\mathbf{x}, t) = [T(x, t), Y_{H_2}(x, t), Y_{O_2}(x, t), Y_{H_2O}(x, t)]^T \in \mathbb{R}^4 \tag{7}$$

contains the temperature T and the mass fraction Y_i of the three species $i \in \{H_2, O_2, H_2O\}$. L_x and L_y denote the length and width of the geometrical rectangular domain. The nonlinear reaction source term

$$s(w) = [s_T(w), s_{H_2}(w), s_{O_2}(w), s_{H_2O}(w)]^T \tag{8}$$

Fig. 11 Local ROM error as a function of the number of local bases: KML-ROM (*solid line*) and PEBL-ROM (*dashed dotted line*) for training (*top row*) and test parameters (*bottom row*)

Fig. 12 Average ROB size as a function of number of local bases: KML-ROM (*square*) and PEBL-ROM (*circle*)

Fig. 13 Local ROM error as a function of the average ROB size (by varying k and keeping ε_{POD} fixed): KML-ROM (*square*) and PEBL-ROM (*circle*) for training (*top row*) and test parameters (*bottom row*)

is of Arrhenius type and

$$s_i(w) = -v_i \frac{W_i}{\rho} \left(\frac{\rho Y_{H_2}}{W_{H_2}}\right)^{v_{H_2}} \left(\frac{\rho Y_{O_2}}{W_{O_2}}\right)^{v_{O_2}} A \exp\left(-\frac{E}{RT}\right),$$

$$i \in \{H_2, O_2, H_2O\} \tag{9}$$

$$s_T(w) = Q\mathcal{S}_{H_2O}(w)$$

with the stoichiometric coefficients $v_{H_2} = 2$, $v_{O_2} = 1$ and $v_{H_2O} = -2$. The molecular weights of the three species are $W_{H_2} = 2.016\,\mathrm{g\,mol^{-1}}$, $W_{O_2} = 31.9\,\mathrm{g\,mol^{-1}}$ and $W_{H_2O} = 18\,\mathrm{g\,mol^{-1}}$. The density of the mixture is $\rho = 1.39 \times 10^{-3}\,\mathrm{g\,cm^{-3}}$. The universal gas constant is $R = 8.314\,\mathrm{J\,mol^{-1}\,K^{-1}}$ and the heat of the reaction is $Q = 9800$ K. The diffusivity is $\kappa = 2\,\mathrm{cm^2\,s^{-1}}$. The activation energy is here $E = 5.5 \times 10^3$ J.mol^{-1}. The advection velocity is chosen as $u = 0.5$ m.s^{-1}.

A Dirichlet boundary condition $T(x) = 950$ K is enforced in the middle of the left boundary. Everywhere else on the left boundary, $T(x) = 300$ K. Homogeneous Neumann boundary conditions are enforced at all other three boundaries of the computational domain which is depicted in Fig. 14. The boundary conditions for the mass fractions are chosen as $Y_i = 0$ on the left boundary and homogeneous Neumann everywhere else.

The PDE is discretized by the finite differences method in space, resulting in a solution vector of dimension $n = 23,104$ and by backward Euler finite differences in time with uniform time step $dt = 6 \times 10^{-4}$ s.

Fig. 14 Computational domain for the reactive flow problem

Fig. 15 Training solutions of the parameterized reaction equation at $t = 0.06$ s for configuration T1
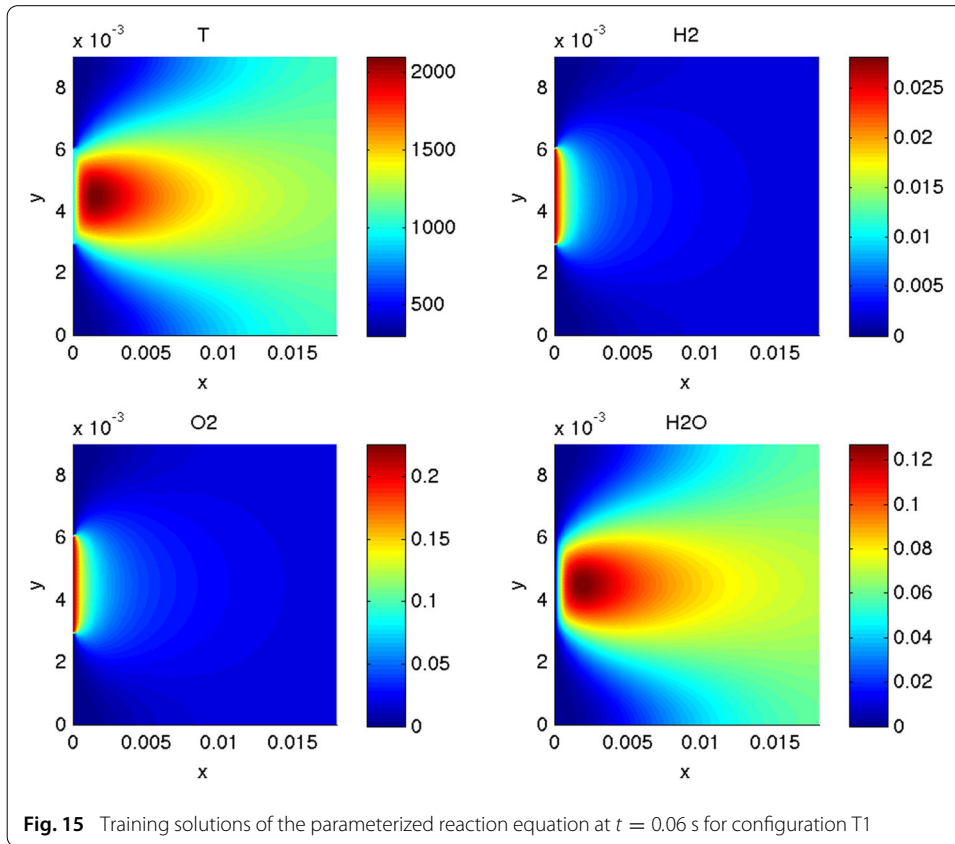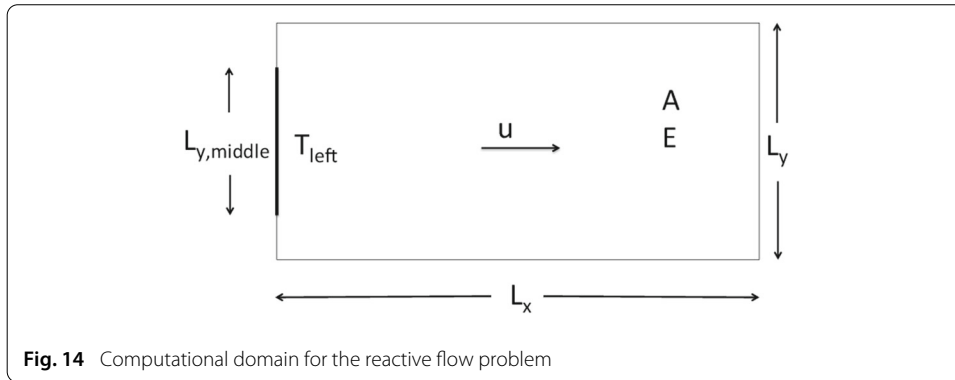
The pre-exponential factor A will be allowed to vary in the following study. More specifically, two training conditions and one testing predictive condition are considered:

- T1 for which $A = 7$
- T2 for which $A = 10$
- P1 for which $A = 8.5$

The steady-state solution associated with the configuration T1 is depicted in Fig. 15. The two training simulations result in the collection of $n_u = 200$ snapshots. Figure 16 displays the temperature field for all three configurations. One can observe that the magnitude of the solution differs in each configuration but the shape of the solution is similar across configurations.
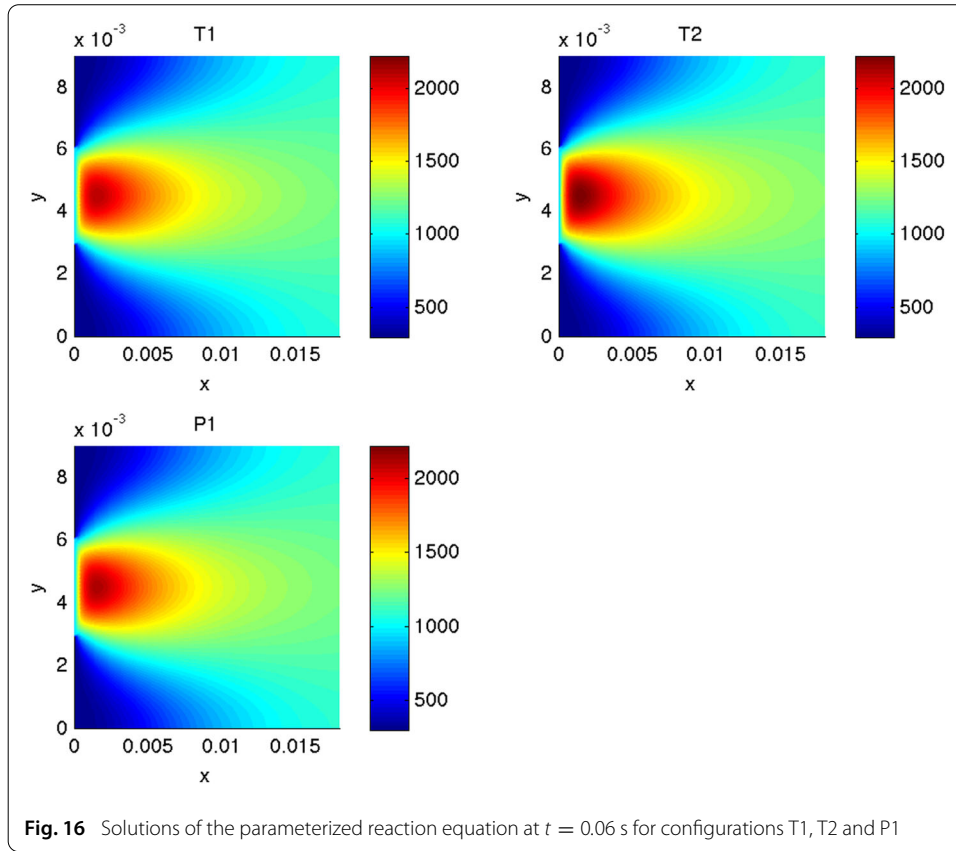
Fig. 16 Solutions of the parameterized reaction equation at $t = 0.06$ s for configurations T1, T2 and P1

The POD truncation is set to $\varepsilon_{POD} = 10^{-12}$ and the number of local bases is varied from $k = 2$ to 10. The accuracy of local ROMs obtained with the two approaches (KML-ROM with clustering overlap of $f_{\mathrm{add}} = 10\,\%$ and PEBL-ROM) are then computed in each case and reported in Fig. 17 for configurations T1 and P1. One can observe that the KML-ROM algorithm is more accurate for the training configuration with an average error of 10^{-4}, versus 0.015 % for the PEBL-ROM approach. However, all models are very accurate here.

On the other hand the PEBL-ROM approach leads to much more accurate predictions for the predictive configuration for $k \leq 7$ (average error of 0.66 versus 1.37 % for KML-ROM procedure) and similar accuracy for $k \geq 8$. This emphasizes the fact that the PEBL-ROM procedure is more suited for clustering snapshots of similar shapes but different magnitudes and might be less prone to overfitting.

Conclusions

A PEBL-ROM approach for local nonlinear model reduction is presented in this work. It relies on a dissimilarity measure defined as the true projection error. The approach proceeds by building offline a binary tree that is used to determine online the local ROB of interest. On a set of toy data, numerical experiments verify that the projection-error based partitioning creates partitions that are independent of intuitive "Euclidean" cluster structure. In the approximation this is reflected in segments being double-cones instead of Voronoi tessellation for a KML-ROM approach. The projection-error partition generates large "generalization" regions outside of any training samples. The clusters are naturally

Fig. 17 Local ROM error for the parametrized reaction equation as a function of the number of local ROBs for $\varepsilon_{POD} = 10^{-12}$: KML-ROM (*solid line*) and PEBL-ROM (*dashed dotted line*)
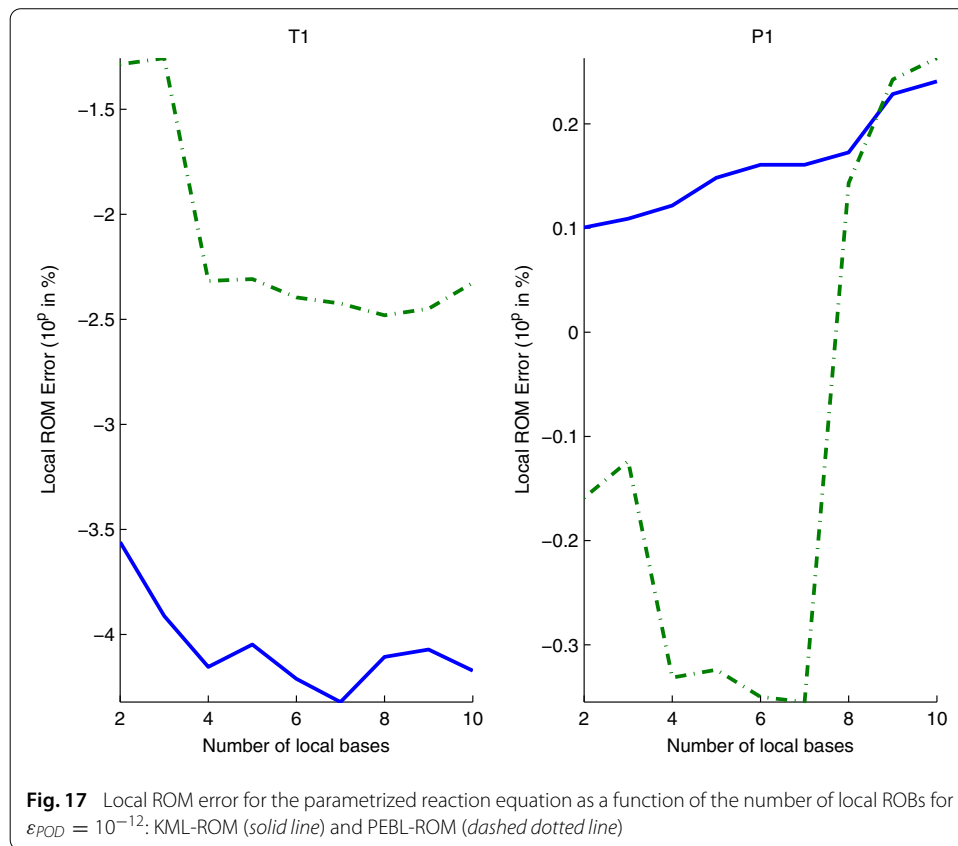
scale invariant, nicely fitting to the projection nature and not available for other local basis approaches so far. In addition to these approximation experiments, MOR experiments are also performed, illustrating the capability of the proposed PEBL-ROM approach to generate accurate local reduced bases in dynamical simulations that are more robust to changes in parameters than existing approaches. Overall we see a very good performance of the PEBL-ROM over the KML-ROM. The situations, where the former is inferior to the latter are mainly situations of large k (i.e., small sets of snapshots per subset) and regions of high POD truncation value. Both situations are not considered to be of major relevance, as accurate ROMs, i.e., models with low POD truncation value are of practical interest. Also the large k (i.e., the case where the ratio of k over the number of snapshots is getting close to 1 is not of interest, as in the limit this would imply clusters of single training snapshots, where all clustering procedures and subspaces coincide.

Author details
[1]Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305, USA , [2]Institute for Applied Analysis and Numerical Simulation, University of Stuttgart, Pfaffenwaldring 57, 70569 Stuttgart, Germany.

Competing interests

The authors declare that they have no competing interests.

References

1. Dihlmann M, Drohmann M, Haasdonk B. Model reduction of parametrized evolution problems using the reduced basis method with adaptive time-partitioning. In: Proc. of ADMOS 2011, International Conference on Adaptive Modeling and Simulation. 2011.
2. Drohmann M, Haasdonk B, Ohlberger M. Adaptive reduced basis methods for nonlinear convection-diffusion equations. In: Proc. FVCA6, Finite Volumes and Complex Applications. 2011.
3. Amsallem D, Zahr M, Farhat C. Nonlinear model order reduction based on local reduced-order bases. Int J Numerical Methods Eng. 2012;92(10):891–916.
4. Haasdonk B, Dihlmann M, Ohlberger M. A training set and multiple basis generation approach for parametrized model reduction based on adaptive grids in parameter space. MCMDS. 2011;17:423–42.
5. Maday Y, Stamm B. Locally adaptive greedy approximations for anisotropic parameter reduced basis spaces. SIAM J Sci Comput. 2013;35(6):2417–41.
6. Peherstorfer B, Butnaru D, Willcox K, Bungartz HJ. Localized discrete empirical interpolation method. SIAM J Sci Comput. 2014;36(1):168–92.
7. Redeker M, Haasdonk B. A POD-EIM reduced two-scale model for crystal growth. Adv Comput Math. 2014;1–27. doi:10.1007/s10444-014-9367-y.
8. Washabaugh K, Amsallem D, Zahr MJ, Farhat C. Nonlinear model reduction for CFD problems using local reduced order bases. AIAA Paper 2012–2686, 42nd AIAA Fluid Dynamics Conference and Exhibit 25–28, New Orleans. Louisiana. 2012;1–16.
9. Amsallem D, Zahr MJ, Washabaugh K. Fast local reduced basis updates for the efficient reduction of nonlinear systems with hyper-reduction. Special issue on Model Reduction of Parameterized Systems (MoRePaS). Adv Comput Math. 2015;1–34.
10. Wieland B. Implicit partitioning methods for unknown parameter sets. Adv Comput Math. 2015;41:1159–86.
11. Eftang JL, Patera AT, Rønquist EM. An *hp* certified reduced basis method for parametrized elliptic partial differential equations. SIAM J Sci Comput. 2010;32(6):3170–200.
12. Eftang JL, Knezevic DJ, Patera AT. An *hp* certified reduced basis method for parametrized parabolic partial differential equations. MCMDS. 2011;17(4):395–422.
13. Eftang J, Stamm B. Parameter multi-domain *hp* empirical interpolation. Int J Numerical Methods Eng. 2012;90(4):412–28.
14. Grepl MA, Patera AT. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. ESAIM. 2005;39(1):157–81.
15. Kunisch K, Volkwein S. Optimal snapshot location for computing POD basis functions. ESAIM. 2010;44(3):509–29.
16. Paul-Dubois-Taine A, Amsallem D. An adaptive and efficient greedy procedure for the optimal training of parametric reduced-order models. Int J Numerical Methods Eng. 2015;102(5):1262–92.
17. Berkooz G, Holmes P, Lumley JL. The proper orthogonal decomposition in the analysis of turbulent flows. Annu Rev Fluid Mech. 1993;25:539–75.
18. Jolliffe IT. Principal component analysis. Berlin-Heidelberg:Springer; 2002. doi:10.1007/b98835.
19. Volkwein S. Proper orthogonal decomposition: theory and reduced-order modelling. 2012. doi:http://www.math.uni-konstanz.de/numerik/personen/volkwein/teaching/POD-Book.
20. LeGresley PA, Alonso JJ. Airfoil design optimization using reduced order models based on proper orthogonal decomposition. AIAA Paper 2000–2545 Fluids, Conference and Exhibit, Denver. CO. 2000;1–14.
21. Barrault M, Maday Y, Nguyen NC, Patera AT. An "empirical interpolation" method: application to efficient reduced-basis discretization of partial differenti al equations. Comptes Rendus de l'Académie des Sciences, Series. 2004;I(339):667–72.
22. Chaturantabut S, Sorensen D. Nonlinear model reduction via discrete empirical interpolation. SIAM J Sci Comput. 2010;32(5):2737–64. doi:10.1137/090766498.
23. Carlberg K, Farhat C, Cortial J, Amsallem D. The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows. J Comput Phys. 2013;242:623–47. doi:10.1016/j.jcp.2013.02.028.
24. Rewienski M. A trajectory piecewise-linear approach to model order reduction of nonlinear dynamical systems. Ph.D. thesis, Massachussets Institute of Technology. 2003.
25. Buffoni M, Willcox K. Projection-based model reduction for reacting flows. AIAA Paper 2010–5008, 40th Fluid Dynamics Conference and Exhibit, 28 June1–July 2010, Chicago, IL. 2010.