



HAL
open science

Error-based efficient parameter space partitioning for mesh adaptation and local reduced order models

Sourabh Bhat, Nicolas Barral, Mario Ricciuto

► **To cite this version:**

Sourabh Bhat, Nicolas Barral, Mario Ricciuto. Error-based efficient parameter space partitioning for mesh adaptation and local reduced order models. *Computer Methods in Applied Mechanics and Engineering*, 2025, 435, pp.117649. 10.1016/j.cma.2024.117649 . hal-04851883

HAL Id: hal-04851883

<https://inria.hal.science/hal-04851883v1>

Submitted on 20 Dec 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Error-based Efficient Parameter Space Partitioning for Mesh Adaptation and Local Reduced Order Models

Sourabh P. Bhat^{a,b}, Nicolas Barral^{a,b}, Mario Ricchiuto^{a,b}

^aUniv. Bordeaux, CNRS, Bordeaux INP, IMB, UMR 5251, F-33400 Talence, France

^bCentre Inria de l'université de Bordeaux, 33405 Bordeaux, France

Abstract

The resolution and accuracy of numerical partial differential equation solvers are governed by the mesh density and the order of accuracy of the solver. Anisotropic mesh adaptation combined with a posteriori error estimation is known to be a powerful tool to enhance the efficiency of the solvers. However, in engineering applications involving multiple complex configurations, optimization or uncertainty quantification, generating a single adapted mesh to efficiently capture features of the solutions for all possible choices of the problem parameters (geometry, boundary conditions, etc.) is not feasible. This is particularly true when the solution is very sensitive to the problem parameters and changes in the topology of the solution features (e.g. one vs two shocks) occur for small changes in the computational setting. This paper presents a novel algorithm [coupling partitioning](#) the parameter space for efficient parameter clustering and local model order reduction [and anisotropic mesh adaptation](#). The hierarchical partitioning is performed by sampling the parameter space using a full-order model on a relatively coarse mesh. The variance of the physical solutions is reconstructed in the parameter space using proper orthogonal decomposition and Taylor's series to guide each partition's size and shape. Subsequently, the solutions are sampled within the optimal partitions to compute the locally adapted mesh and optimal reduced-order model for each partition. The efficacy of the new algorithm is demonstrated by solving heat conduction with material discontinuity and compressible flow problems.

Keywords: Local reduced order models, Parameter space partitioning, Anisotropic mesh adaptation, Mesh clustering, Hierarchical clustering

Notations and symbols used in the paper

Notation	Description
d	: number of physical dimensions

*Corresponding author: Nicolas Barral

Email addresses: sourabh.bhat@inria.fr (Sourabh P. Bhat), nicolas.barral@inria.fr (Nicolas Barral), mario.ricchiuto@inria.fr (Mario Ricchiuto)

Notation	Description
$\Omega \in \mathbb{R}^d$: physical domain
k	: number of parameters
$\Omega_\mu \in \mathbb{R}^k$: parameter space
x, y, z	: Cartesian coordinate directions for the physical space
u_x, u_{xy}, \dots	: derivatives of function u with respect to x and second derivative of u with respect to x then y
$\mu_1, \mu_2, \dots, \mu_k$: parameters of the problem
\mathcal{M}	: metric tensor used for mesh adaptation
$\mathcal{M}_{1 \cap 2}$: metric obtained after intersection of two metrics \mathcal{M}_1 and \mathcal{M}_2
$u(x, y, z; \mu_1, \mu_2, \dots, \mu_k)$: high-fidelity solution
$S \in \mathbb{R}^{n \times s}$: snapshot matrix whose columns are the sampled high-fidelity solutions at s sampling points in Ω_μ
$U \in \mathbb{R}^{n \times s}$: matrix whose columns are left singular vectors of matrix S
$\Sigma \in \mathbb{R}^{s \times s}$: diagonal matrix, $diag(\sigma_1, \sigma_2, \dots, \sigma_s)$, containing singular values arranged in the order of decreasing magnitude, $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_s \geq 0$
$V \in \mathbb{R}^{s \times s}$: matrix whose columns are right singular vectors of matrix S . In the paper, V^T has rows as the right singular vectors of matrix S
$\Phi \in \mathbb{R}^{n \times r}$: matrix whose columns $(\phi_1, \phi_2, \dots, \phi_r)$ are the reduced number of solution basis vectors with n degrees of freedom, arranged in the order of decreasing importance
σ'	: normalized singular value = σ/σ_1 , with respect to the largest singular value
$\hat{\sigma}$: cutoff normalized singular value, σ' , used as convergence criterion during partitioning
$\bar{\sigma}$: normalized singular value = $\sigma/\sum_i \sigma_i$, with respect to sum of all the singular values

1. Introduction

The accuracy of numerical methods for solving partial differential equations (PDE), such as finite volume and finite element methods, is governed by the quality and resolution of the mesh used for the simulation. In particular, for optimal accuracy, it is computationally efficient to have a dense mesh in regions with a rapidly changing solution and a coarse mesh in smooth regions. Moreover, anisotropic meshes can improve

the computational efficiency further by taking into account the directional changes in the solution. This is particularly true for problems with discontinuities in the solution, such as problems with material discontinuities or shocks in compressible flows. The optimal mesh being solution dependent, a uniform mesh is seldom optimal for most of the interesting engineering problems. Therefore, a solution dependent mesh adaptation combined with some a posteriori error estimator has been widely used in computationally efficient, stable and accurate numerical PDE solvers [1–6].

In optimization problems and Model Order Reduction (MOR), the problem depends on a number of parameters and the parameter space of the problem is explored to obtain optimal solutions and to generate a Reduced Order Model (ROM) respectively. Another possibility is to generate an offline ROM to be used for very fast optimization loops [7–15]. MOR is also widely applied in digital twins due to the speedup obtained by the ROM [16–26]. All these applications require to a large extent the computation of accurate solutions of the PDEs which may intervene both in the generation of the ROM and in the optimization process. Clearly, the use of some aggressive mesh adaptation method would allow to push the accuracy of such simulations whenever relevant. However, it is impractical to generate an optimal mesh for each parameter combination while exploring the parameter space for generating the ROM, as the construction requires the solution snapshots on a common mesh. It is also computationally intractable to have a suitably refined mesh to capture all the features in the global parameter space as this will result in a mesh with a large number of degrees of freedom, thus requiring a huge computational effort to generate high-fidelity solutions during the offline stage, and relatively expensive ROMs in the online stage. A more practical approach is to generate a small cluster of meshes, each mesh being tailored for a few local sub-spaces within the global space. This approach is further referred to as mesh clustering and keeps the size and number of meshes manageable, and still maintains the accuracy of the solution everywhere in the global parameter space. So, the next question which arises is how to partition the global parameter space such that the accuracy is uniformly improved in all the partitions.

In the literature, two families of methods have been explored for local reduced order models in an attempt to generate computationally efficient ROMs. The first family uses pre-computed solutions in the global parameter space, which are then clustered into separate groups. One of such widely used method is the k -means partitioning, where the solutions in the complete parameter space are computed and grouped into a number k of clusters such that each group’s average (i.e. mean) forms the center of the cluster [27–29]. The distance norm (such as Euclidean norm) between the cluster’s mean and the solutions is iteratively minimized to obtain k optimal clusters. The optimization process can converge to local minima depending on the starting solution mean. Therefore the process is repeated several times, with a different starting mean solution, to obtain the best possible clusters. In this class of methods one needs to choose in advance the expected number

of clusters for a given problem, which may not always be obvious. The silhouette method [30–32] and other optimization techniques [33] have been used to identify the optimal number of clusters, which further increases the computational cost. Furthermore, for classification, a separate machine learning model may be necessary to identify appropriate clusters during the online phase [34]. Another limitation of this approach is that the resolution of the parameter space sampling must be predecided and solutions precomputed. A clever greedy approach [35, 36] for building local reduced order models can mitigate some of these limitations. However, the greedy approach is possible only for problems where a partially built reduced order model can be used to efficiently estimate the error of the model. The commonly used measure in k -means clustering, i.e. the Euclidean norm in the state space, may be a suboptimal metric as solutions which are scaled versions of each other (and therefore having similar bases for ROMs) can be very far in the state space [37]. Other metrics, such as the DEIM-residual-based metric [34], may be used for computing the distance from the cluster’s center to overcome this limitation. The second family of methods use a hierarchical tree structure to identify local sub-spaces with similar solution bases [38, 39]. This approach can be computationally efficient as the necessary resolution of sampling is learned by the algorithm during partitioning. The tree-based algorithms are also efficient during classification in the online phase because of their logarithmic computational time complexity when choosing the appropriate ROM, due to the binary search tree data structure used for storing the partitioned parameter space. Other works can be mentioned, e.g. related to the use of the variance in parameter space to produce overlapping clusters [40], allowing smoother transitions between two different local reduced models, or inspired by h -adaptation techniques to produce adaptive sub-clusters also using some measure of the variation of the output with the parameters [41].

In this work we focus more on using local bases in MOR in combination with mesh adaptation in the physical space to enhance the resolution of the outputs, for a given size of the spatial mesh. To this end, we use a binary-tree-based algorithm to generate local partitions in the parameter space. On each partition we can produce a specific mesh adapted in the physical space which is optimal in the sense that the approximation error in the physical space is minimized on this mesh for all values of the parameters within the partition. The subdivision method is an extension of the k -dimensional tree algorithm (see [42]), modified for parameter-space partitioning. The original algorithm is modified to learn the required sampling resolution for the given problem and to optimally partition the parameter space. The algorithm is designed to minimize the number of sampling points, therefore minimizing the computational cost incurred by the PDE solver, and reuse most of the previous solutions during the sub-division process. The resulting partitioning of the parameter space is combined with anisotropic mesh adaptation with a posteriori error estimation to maximize the efficiency of the overall computational method. The efficacy of the algorithm is evaluated by solving heat conduction problems with material discontinuities and compressible flow problems with shocks.

The paper is organized as follows. The next section provides a high-level overview of the problem being solved, the techniques of mesh adaptation, metric intersection and model order reduction. Section 3 presents a detailed description of the partitioning algorithm. This section also provides a brief discussion on the governing equations used in the test problems and the full order PDE solver. In Section 4 the errors with and without partitioning are compared using heat conduction problems with analytic solutions. Section 5 uses the new algorithm for partitioning the parameter space with unknown solutions, computed by the full-order PDE solver. Various configurations of the heat conduction problems in two and three dimensions with material discontinuities are solved in Subsection 5.1. Problems involving compressible flow over two-dimensional and three-dimensional bodies are solved in Subsection 5.2. Finally, Section 6 concludes the paper with useful remarks on the presented work and scope for future research.

2. Problem definition

Consider a steady-state Equation (1) defined on a d -dimensional physical domain $\Omega \in \mathbb{R}^d$,

$$F(u, u_x, \dots, u_{xx}, u_{xy}, \dots; \mu_1, \mu_2, \dots, \mu_k) = 0. \quad (1)$$

The solution, u depends on the parameters of the problem, $\mu_1, \mu_2, \dots, \mu_k$, which form a global parameter space $\Omega_\mu \in \mathbb{R}^k$. A schematic showing the problem with $d = 2$ and parameter space¹ with $k = 2$ and parameters μ_1 and μ_2 is shown in Figure 1. When solving the problem numerically, the solution has a finite number of degrees of freedom n depending on the resolution of the mesh. Thus, for numerical solutions $u \in \mathbb{R}^n$, the solution varies across the parameter space. Therefore, the optimal mesh capturing the solutions is also different at every point in the parameter space. In this work, the global parameter space is divided into smaller sub-spaces (called partitions) such that all the parameters of a partition share a common optimal mesh, which may differ from the neighbouring partitions. Moreover, the process also produces optimal partitions for building reduced order models.

2.1. Mesh adaptation

The resolution of the mesh plays a vital role in the accuracy of PDE solvers. A uniform mesh in the complete domain is generally not capable of efficiently resolving all the solution features. Error-based mesh adaptation is a well-tested technique to optimally use the available computing resources, such that high accuracy is maintained in the complete domain [1, 2]. In our work, we use metric-based anisotropic mesh

¹The symbol k here should not be confused with the same symbol used in k -means method, where it has a different meaning.

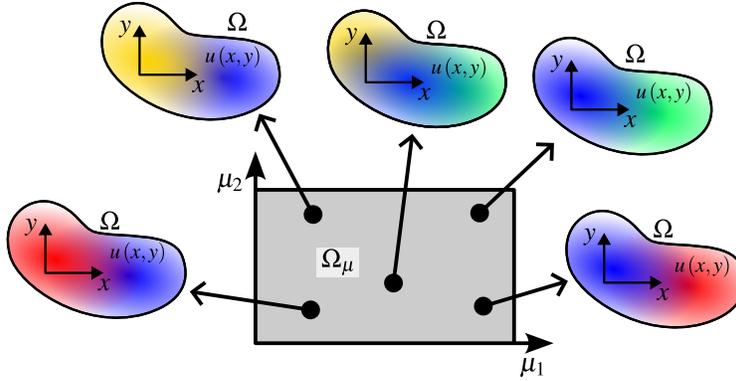


Figure 1: Schematic representation of the solution space (x, y) and parameter space (μ_1, μ_2)

adaptation, where the adaptation is driven by a Riemannian metric tensor derived from an estimate of the interpolation error based on the hessian of the solution.

The metric tensor (or metric in short) prescribes the mesh element sizes along its d principal directions, an information which the meshing software can use to create a new anisotropic mesh capable of accurately capturing the solution. For a given solution at a point in the parameter space Ω_μ , the field of symmetric positive-definite metric tensor \mathcal{M}_μ is defined in the physical space Ω . In practice, \mathcal{M}_μ is defined at each vertex of the mesh. In the rest of the paper, we drop the subscript μ for readability reasons, but the parametric dependence is always implied. In this work, the metric tensor is derived from a numerically computed Hessian matrix of the solution at each vertex of the mesh. The second-order derivatives of the solution are computed using a weighted least-square approach [43, 44] by computing the first-order derivatives twice. The principal directions of anisotropy are given by the eigenvectors of the symmetric Hessian matrix. A so-called multiscale metric, that is able to accurately capture different scales of the solution, is computed from the Hessian matrix to optimize the L_2 -norm of the error with a target number of mesh elements [45, 46]. The mesh element sizes along the eigenvector directions are inversely proportional to the square roots of the the eigenvalues of the metric tensor.

As the metric tensor is solution dependent, it changes with changing parameters, thus the ideal adapted mesh is different at every point in the parameter space. A globally refined mesh for the complete parameter space is not practical for most of the realistic problems, as a very fine mesh will be necessary to capture the features in the complete parameter space. This is especially true in problems with moving material interfaces and shocks in compressible flows. To overcome this limitation, we analyze the parameter space using a proper orthogonal decomposition of the sampled solutions and subdivide the global parameter space into smaller partitions, thus creating local parameter spaces, as discussed in Subsection 3.1. An optimal mesh specific to each partition is computed by intersecting the metrics at various points in the local parameter space.

Metric intersection [46] is a procedure that combines two metric tensors ensuring that the sizes prescribed by the result are smaller than the sizes of the two operands. Let \mathcal{M}_1 and \mathcal{M}_2 be symmetric positive definite matrices defined at a mesh vertex (in the physical space), corresponding to the metric at two points in the parameter space. Here we write the formulas in 3D, the 2D case can easily be deduced. The intersection of \mathcal{M}_1 and \mathcal{M}_2 produces a resultant metric $\mathcal{M}_{1\cap 2}$ such that the size prescribed by $\mathcal{M}_{1\cap 2}$ is smaller in each direction than that prescribed by either \mathcal{M}_1 or \mathcal{M}_2 . The metric intersection is computed as:

$$\mathcal{M}_{1\cap 2} = \left(\mathcal{M}_1^{1/2} \right)^T \overline{\mathcal{M}_{1\cap 2}} \mathcal{M}_1^{1/2}, \quad (2)$$

with $\mathcal{M}_1^{1/2} = \mathcal{R}_1 \Lambda_1^{1/2} \mathcal{R}_1^T$, where columns of \mathcal{R}_1 are the right eigenvectors and $\Lambda_1^{1/2} = \text{diag}(\sqrt{\lambda_{11}}, \sqrt{\lambda_{12}}, \sqrt{\lambda_{13}})$ are the square roots of eigenvalues of \mathcal{M}_1 . Matrix $\overline{\mathcal{M}_{1\cap 2}}$ is defined as:

$$\overline{\mathcal{M}_{1\cap 2}} = \overline{\mathcal{R}_2} \begin{pmatrix} \max(\overline{\lambda}_{21}, 1) & 0 & 0 \\ 0 & \max(\overline{\lambda}_{22}, 1) & 0 \\ 0 & 0 & \max(\overline{\lambda}_{23}, 1) \end{pmatrix} \overline{\mathcal{R}_2}^T. \quad (3)$$

The columns of $\overline{\mathcal{R}_2}$ are the right eigenvectors and $\overline{\Lambda}_2 = \text{diag}(\overline{\lambda}_{21}, \overline{\lambda}_{22}, \overline{\lambda}_{23})$ are the eigenvalues of matrix $\overline{\mathcal{M}_2}$, which is given as

$$\overline{\mathcal{M}_2} = \left(\mathcal{M}_1^{-1/2} \right)^T \mathcal{M}_2 \mathcal{M}_1^{-1/2}, \quad (4)$$

with $\mathcal{M}_1^{-1/2} = \mathcal{R}_1 \Lambda_1^{-1/2} \mathcal{R}_1^T$, where $\Lambda_1^{-1/2} = \text{diag}(1/\sqrt{\lambda_{11}}, 1/\sqrt{\lambda_{12}}, 1/\sqrt{\lambda_{13}})$.

In the context of this article, we seek to adapt the mesh to a set of solutions at different points in Ω_μ . A resultant metric is computed by combining the metrics of each solution. A reduction operation, intersecting two metrics at a time, is performed over all the metrics in the set to obtain the resultant metric tensor capable of capturing all the solutions in the set. The adapted mesh is obtained by passing the resultant metric to a remeshing software to obtain the new mesh. In this work the adapted mesh is generated using an open source remeshing library called MMG [47–49].

2.2. Model order reduction

In projection-based Model Order Reduction (MOR), the solution $u \in \mathbb{R}^n$ is defined using a linear combination of a small number of global basis functions:

$$u = \Phi \hat{u} = \sum_{i=1}^r \hat{u}_i \phi_i, \quad (5)$$

where $\Phi \in \mathbb{R}^{n \times r}$ is a matrix containing the r most dominant global basis functions arranged as column vectors $(\phi_1, \phi_2 \dots \phi_r)$ and \hat{u} is a vector of weights to reconstruct the solution. The PDE solver is modified to work in this global basis by using Galerkin projection, i.e. pre-multiplying Φ^T to the system of linear equations generated by the PDE solver. This produces a system of r equations instead of n . Therefore, if $r \ll n$, it results in a significantly faster solver. This method is an intrusive method as it requires a change of the underlying solver. Moreover, a technique called hyper-reduction is essential for solving non-linear problems to favorably scale the reduced order model (ROM). The method of hyper-reduction uses specially designed techniques [50–55] for sparse sampling and evaluation of non-linear terms. The global basis functions are computed in an offline stage, by solving the problem at s different sampled points in the parameter space. The solution snapshots are stacked as columns vectors to create a snapshot matrix $S \in \mathbb{R}^{n \times s}$, which is decomposed using singular vector decomposition (SVD) $S = U \Sigma V^T$. The diagonal matrix Σ contains the singular values arranged in order of decreasing magnitude $\sigma_1 > \sigma_2 > \dots > \sigma_s$, while $U \in \mathbb{R}^{n \times s}$ and $V^T \in \mathbb{R}^{s \times s}$ contain the left and the right singular vectors respectively. The left singular vectors provide an optimal basis for the column space (i.e. the physical space) of S , while the right singular vectors provide an optimal basis for row space (i.e. the parameter space) of S [56, 57]. Therefore, each column of the matrix V^T corresponds to the sampling point column in the solution snapshot matrix.

In many problems only the first few basis functions $r < s \ll n$ are sufficient to effectively reconstruct the solutions in a large parameter space. The number of basis functions required to effectively reconstruct the full order solution is decided by analyzing the decay of the singular values. Various techniques are used in the literature [57] to decide the reduced number of basis functions r . Some of these methods include truncation of SVD at a rank r corresponding to a predefined percentage of variance or energy, identifying features such as “elbow” in the singular value distribution, a hard cutoff on the normalized singular values, or fitting a signal-noise model to identify the optimal threshold [58].

The online evaluation of the ROM grows rapidly with r . Therefore, ROMs become less useful when a large number of basis functions is required to obtain a required accuracy. Moreover, the upfront costs involved in computing basis functions, building the ROM and the changes required in the solver in case of intrusive ROMs, make them less attractive if the savings are minimal. This is particularly true in problems with moving solution features, such as shocks or wave fronts, and when the solution has a strong non-linear dependence on the parameters. Such problems require a large number of basis functions to effectively capture the solutions. It is therefore advantageous to work with smaller parameter sub-spaces to ensure each local ROM i.e. specific to a sub-space requires few basis functions, making each local ROM very efficient. In some cases, such as hyperbolic equations with discontinuities, one may want to align the snapshots to a reference solution to facilitate the ROM construction. A registration procedure was recently developed for

this purpose [59]. This procedure benefits from the identification of changes in the solution structure and the use of parametric clustering [55].

Another advantage of using local ROMs is that we can generate a mesh tailored to the solutions captured by each local ROM. Mesh adaptation compounds the benefits of ROMs because for a reduced computational cost, we obtain a very accurate solution for every local ROM. The process of metric based mesh adaptation and intersection of metrics to obtain a mesh suitable for all the solutions generated by the local ROM are discussed in Subsection 2.1. The process of optimally partitioning the global parameter space into smaller non-uniform regions so as to minimize the value of r for all the local ROMs is presented in Subsection 3.1.

3. Methods

3.1. Algorithm for optimal partitioning

The partitioning of the parameter space is determined by the number of dominant basis functions necessary to capture the solutions within the partition. The parameter space is split recursively in two, along one of the axes, following a k-dimensional binary tree structure.

The overall procedure for parameter space partitioning is presented in Algorithm 1. A summary of the algorithm is provided in this paragraph, so that the reader can understand the overall idea behind it. First, sample solutions of the full order model are computed at evenly distributed points in the parameter space. The same coarse global mesh is used to discretize the physical space for every parameter. The resulting solution vectors are then stacked as columns to form the snapshot matrix. Singular Value Decomposition is used to obtain singular vectors and singular values of the snapshot matrix. The sorted singular values are normalized by order of magnitude. For a problem with k parameters, if the $(k + 2)^{\text{th}}$ singular value is smaller than a chosen cutoff value the partition is said to be converged (see Step **Step 4** below). If the partition is not converged, it is divided into two sub-partitions by splitting along the parameter dimension with the largest variation. This results in a recursive process of optimally splitting the global parameter space into smaller sub-spaces (i.e. partitions) of minimal variation. Finally, the global mesh is adapted within each partition, resulting in a series of local meshes capturing adequately the solutions. These local adapted meshes can then be used for solving optimization problems or generating highly accurate and economical reduced order models. These steps are further elaborated below.

The global parameter space is defined by the problem being solved. This is the complete space for which the ROM is being developed, or the global space considered for the optimization problem. As a first step, an initial mesh is generated that is used on the whole (global) parameter space. This mesh must be fine enough to capture the salient features of the solution for each parameter in the physical domain, however it must be

Algorithm 1 Recursive mesh clustering and local model order reduction algorithm

```
1:  $Level \leftarrow 0$ 
2: Input parameter space
3: if  $Level \geq Level_{\max}$  then
4:   Save partition data
5:   Return
6: end if
7: Sample points in parameter space
8: Solve the problems at sampled points
9: Stack solution snapshots as columns and compute SVD  $\rightarrow U, \sigma, V$ 
10: if  $\sigma'_{k+2} \leq \hat{\sigma}$  then
11:   Save partition data
12:   Return
13: end if
14:  $Level \leftarrow Level + 1$ 
15: Compute Hessian matrix:  $V \rightarrow H$ 
16: Compute the variation along all dimensions using second order derivatives
17: Split the partition into two by dividing the dimension with maximum variation
18:   left branch: Goto 2
19:   right branch: Goto 2
```

as coarse as possible to reduce the computational cost. In cases where the mesh refinement to adequately capture the solution is unknown, the initial mesh can be generated by sparse sampling of the parameter space with a posteriori error estimator. For example, this technique is used in Subsection 5.2 to reduce the overall computational cost. The algorithm begins at $Level = 0$.

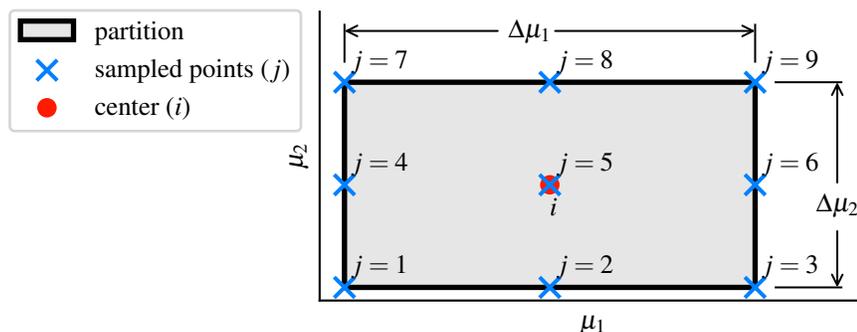


Figure 2: A schematic of the sampled points with $p = 3$, inside a partition of the parameter space with $k = 2$ and parameters μ_1, μ_2 .

Step 1: Sample the parameter space and obtain solution snapshots (using a PDE solver) at sampled points inside the partition using the global mesh. For a given k -dimensional parameter space, various points are sampled so as to uniformly span the space. This can be easily done by a Cartesian product of p locations along each of the k directions, resulting in $s = p^k$ sampling points inside the partition. In this work we choose $p = 3$ along each direction. The points are chosen such that the first and the last points are on the

boundary of the partition, which allows us to cache and reuse the solutions between neighboring partitions. Also, choosing $p = 2^b + 1$ is recommended as it allows for maximum reuse of cached solutions in subsequent steps of the algorithm, where $b \geq 1$. Moreover, one may also choose a larger value of b if $Level = 0$, say $b = 3$, to capture the variations in the global parameter space and then reduce to, say $b = 2$, if $Level = 1$, and finally to $b = 1$ if $Level \geq 2$. A schematic of the sampled points for $k = 2$ and $p = 3$ is shown in Figure 2.

Step 2: The solution vectors are stacked as columns of a snapshot matrix S . This results in a matrix of shape $n \times s$ where n is the number of vertices of the global mesh and $s = p^k$ is the number of sampling points. As the solutions are obtained at sampled points using the same mesh, each column of matrix S corresponds to a sampling point in the parameter space and each row of the matrix corresponds to a mesh vertex in the physical space.

Step 3: The snapshot matrix is decomposed using compact Singular Value Decomposition (SVD) into three matrices $S = U \Sigma V^T$. The columns of matrix U , also called left singular vectors, are orthonormal vectors which form the basis of the solutions in the physical space. The diagonal matrix $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_s)$ contains non-negative singular values sorted in descending order. The columns of matrix V , also called right singular vectors², are orthonormal vectors which form the basis of the solutions in the parametric space.

Step 4: Projection based ROMs use a linear combination of basis functions for approximating solutions in the parameter space. Such an approximation in a k -dimensional parameter space requires at most $k + 1$ basis functions for a linear approximation, as one can easily verify by analyzing e.g. a linear truncated Taylor expansion in k dimensions. Therefore, we need to ensure that the $(k + 2)^{\text{th}}$ singular value is small. The partition is said to be converged if the $(k + 2)^{\text{th}}$ normalized singular value, $\sigma' = \sigma/\sigma_1$, is below the specified threshold (cutoff criterion) $\hat{\sigma}$, and the partition becomes a terminal node. Otherwise, the partition is split into two sub-partitions as discussed in the next steps. Once the partition converges, a dedicated folder is created for the partition to save the information about the bounds of the partition and the adapted mesh using solutions sampled (possibly at points other than the ones used for SVD) within the partition. The mesh adaptation process is discussed in Section 2.1.

Step 5: If the partition is not converged ($\sigma' > \hat{\sigma}$), the tree branch is split into two (left branch and right branch) with each sub-partition half the parent size. This is done by halving one parameter's range, while keeping the other ranges identical to those of the parent partition. The direction of the splitting (i.e. the parameter to halve) is decided based on the second order derivatives of the reconstructed right singular

²Please note that depending on the library used for computing the SVD, the library may return V^T or V . For example, Python's NumPy library returns V^T matrix while MATLAB returns V matrix. In this work we use Python's Numpy library.

vectors, as they project the solution in the parameter space. A truncated Taylor's series is used to fit a quadratic polynomial to the discrete components of right singular vector in the parent partition by using a least-square approximation.

Consider the q^{th} right singular vector $v^{(q)}$ i.e. row q of the V^T matrix. The components of the vector correspond to the s sampling points in a k -dimensional parameter space. For notational simplicity and without loss of clarity we drop the superscript of v in **Step 5**, thus $v^{(q)} \equiv v$. The second order accurate Taylor's series approximation at a sampling point j in the parameter space about the center of the partition i can be written as:

$$v_j = v_i + \sum_{l=1}^k \left(\frac{\partial v}{\partial \mu_l} \right)_i (\Delta \mu_l)_j + \frac{1}{2} \sum_{l=1}^k \sum_{m=1}^k \left(\frac{\partial^2 v}{\partial \mu_l \partial \mu_m} \right)_i (\Delta \mu_l)_j (\Delta \mu_m)_j . \quad (6)$$

The center of the partition is denoted by index i and the approximation is written for a sampled point j (refer to Figure 2). The expression is written for a k -dimensional parameter space, with l and m used for indexing dimensions. v_j is the j^{th} component of the right singular vector which corresponds to sampling point j . The distance $(\Delta \mu_l)_j$ is the normalized distance of parameter μ_l between the sampling point j and the center i . Thus, $(\Delta \mu_l)_j = \left((\mu_l)_j - (\mu_l)_i \right) / |\Delta \mu_l|$, where $(\mu_l)_i$ and $(\mu_l)_j$ are the value of parameter μ_l at point i and j respectively. $|\Delta \mu_l|$ is the width of the partition along dimension l . The unknowns in Equation (6) are v_i and the first and second order derivatives of v evaluated at point i . Writing Equation (6) for a fixed q and all the s sampling points results in the following system of equations (note that depending on how sampling

points are chosen, some v_i may be known already):

$$\begin{pmatrix}
1 & (\Delta\mu_1)_1 & \dots & (\Delta\mu_k)_1 & 1/2(\Delta\mu_1)_1 & (\Delta\mu_1)_1 & (\Delta\mu_1)_1(\Delta\mu_2)_1 & \dots & 1/2(\Delta\mu_k)_1 & (\Delta\mu_k)_1 \\
1 & (\Delta\mu_1)_2 & \dots & (\Delta\mu_k)_2 & 1/2(\Delta\mu_1)_2 & (\Delta\mu_1)_2 & (\Delta\mu_1)_2(\Delta\mu_2)_2 & \dots & 1/2(\Delta\mu_k)_2 & (\Delta\mu_k)_2 \\
\vdots & \vdots \\
1 & (\Delta\mu_1)_s & \dots & (\Delta\mu_k)_s & 1/2(\Delta\mu_1)_s & (\Delta\mu_1)_s & (\Delta\mu_1)_s(\Delta\mu_2)_s & \dots & 1/2(\Delta\mu_k)_s & (\Delta\mu_k)_s
\end{pmatrix}
\begin{pmatrix}
v_i \\
(\partial v/\partial\mu_1)_i \\
\vdots \\
(\partial v/\partial\mu_k)_i \\
(\partial^2 v/\partial\mu_1^2)_i \\
(\partial^2 v/\partial\mu_1\partial\mu_2)_i \\
\vdots \\
(\partial^2 v/\partial\mu_k^2)_i
\end{pmatrix}
=
\begin{pmatrix}
v_{j=1} \\
v_{j=2} \\
\vdots \\
v_{j=s}
\end{pmatrix}
. \quad (7)$$

The over-determined system of equations is solved using a least-square approximation (Moore–Penrose inverse[60]) to obtain the unknowns v_i and the first and second order derivatives of v evaluated at the center of the partition. The second derivatives of the reconstructed polynomial for a right singular vector q are assembled into a Hessian matrix. Equation (7) is written and solved for each of the right singular vectors $q = 1 \dots s$ (i.e. each row of the matrix V^T) to obtain s Hessian matrices.

Step 6: The one-dimensional splitting of a Cartesian cell in the parameter space requires some measure of the direction along which the largest variation of the output is observed. To this end in this work we propose a heuristic definition which makes use of one-dimensional second derivatives in each of the parameter space directions $\partial^2 v/\partial\mu_l^2$. This quantity is a classical measure of the approximation error for linear polynomials, as discussed in section §2.1 (see [1, 61]). To be more precise, for each direction μ_l we construct a sensor which is defined as a weighted average of the second derivatives of each right singular vector $v^{(q)}$, with weights $\bar{\sigma}_q \in [0, 1]$ defined from the relative magnitude of the singular values σ_q . In formulas we have

$$\left| \left(\frac{\partial^2 v}{\partial\mu_l^2} \right)_i \right| \leftarrow \sum_{q=1}^s \bar{\sigma}_q \left| \left(\frac{\partial^2 v^{(q)}}{\partial\mu_l^2} \right)_i \right| \quad l = 1, \dots, k, \quad \bar{\sigma}_q = \frac{\sigma_q}{\sum_{\gamma=1}^s \sigma_\gamma}. \quad (8)$$

Equation (8) is based on the understanding that the contribution of the right singular vectors in reconstructing the solution matrix is proportional to the corresponding singular value. The partition is split by halving the parameter l corresponding to the maximum second order derivative calculated using Equation (8). The *Level*

is incremented by 1 for both the sub-partitions.

Step 7: For each of the sub-partitions **Step 1** through **Step 6** are followed until it converges in **Step 4** or the binary tree reaches the maximum level $Level_{max}$. Therefore, the recursive algorithm is guaranteed to terminate either by converging to specified error (i.e. cutoff singular value) or by reaching the maximum specified level of recursion. In both these cases the mesh is adapted for the partition (see Section 2.1) and stored in a folder specific to the partition.

Step 8: Once the partitioning algorithm is converged, optimal adapted meshes of the physical domain Ω are generated on each partition, using metric intersection following the approach detailed in Section 2.1. Each partition now has a unique mesh that is adapted to the whole partition, that can be used to compute high-fidelity solutions which are then used to train a local Reduced Order Model on the partition.

The cutoff criterion (i.e. the threshold used for the decision of splitting in Algorithm 1 line 10) is an important specification for the algorithm, as it decides the resulting number of partitions. A lower cutoff criterion leads to a higher number of partitions and vice versa. The cutoff criterion to generate a given number of partitions for a given problem and parameter ranges cannot be predetermined as it depends on the variations of the solutions within the parameter space. A smaller cutoff criterion leads to efficient ROMs and highly tailored meshes for the partitions, however it may also lead to a large number of partitions which requires more disk space to store the meshes and POD modes for the ROM's library. As the disk storage resources are limited, it is recommended to start with a higher cutoff value, say $\hat{\sigma} = 5\% = 0.05$, and perform the partitioning using the prescribed algorithm. The algorithm can then be repeated by reducing the cutoff criterion until the desired number of partitions are reached. The solutions should be cached and reused at each stage to avoid repeated computational work as it is guaranteed that the algorithm will take the same path through the binary tree each time, splitting the branches further as needed. This leads to a very efficient greedy approach with sparse sampling of the parameter space.

During the classification step, i.e. when using the resulting partitions and mesh for solving problems, it is crucial to quickly locate the cluster to which the queried parameter belongs to. This search is of order $\mathcal{O}(\log_2 N)$, where N is the number of partitions, due to the binary-tree data structure. It is further bounded by the value chosen for $Level_{max}$. This results in a very fast partition lookup, which is important when the resulting number of partitions are large in case of highly non-linear problems.

3.2. Governing equations for test cases

The governing partial differential equation (PDE) for the steady state heat conduction problem with source term can be written as:

$$\nabla \cdot (\alpha \nabla T) = f . \quad (9)$$

Here, $T = T(x, y, z)$ is the temperature, $\alpha = \alpha(x, y, z)$ is the coefficient of diffusivity and $f = f(x, y, z)$ is the heat source term. Dirichlet boundary conditions are used in this work.

The governing equations for the steady-state inviscid compressible flow, i.e. the steady-state Euler equations, can be written in a conservative form as:

$$\begin{aligned} \nabla \cdot (\rho \mathbf{V}) &= 0 \\ \nabla \cdot (\rho \mathbf{V} \otimes \mathbf{V} + p \mathbf{I}) &= 0 \\ \nabla \cdot ((\rho E + p) \mathbf{V}) &= 0, \end{aligned} \quad (10)$$

where, $\rho = \rho(x, y, z)$ is the density of the fluid, $\mathbf{V} = \mathbf{V}(x, y, z)$ is the velocity vector field and $p = p(x, y, z)$ is the pressure field. The total specific energy is given as:

$$E = \frac{p}{\rho(\gamma - 1)} + \frac{\mathbf{V} \cdot \mathbf{V}}{2} . \quad (11)$$

The ratio of specific heats γ is 1.4. Supersonic inflow and wall boundary conditions are used in this work.

The high-fidelity numerical solutions to the PDEs are computed using a second-order accurate Finite Volume Method [62]. The heat conduction equation is solved in its hyperbolic form [63] to improve the solver's convergence and stability on high aspect ratio anisotropic tetrahedral meshes. The HLLC Riemann solver [64, 65] is used for computing convection fluxes when solving the compressible flow problems. In the following examples, the parameters are contained in the boundary conditions and/or the source terms and will be specified for each case.

4. Error analysis

In this section, we assess the behavior of the algorithm on 2D analytical cases. The key metric of success is its ability to build partitions resulting in a consistently smaller numerical errors for a given computational cost. The error analysis of the newly developed algorithm for optimal parameter space partitioning and mesh clustering is performed by solving steady state heat conduction PDE from Equation (9) for following problems with manufactured solutions. The manufactured solutions are generated with artificial discontinuities so that

average error of the numerical scheme can be computed and compared. The average error in the parameter space is defined as:

$$\text{average error} = \frac{\int_{\Omega_\mu} \int_{\Omega} |u(\mu) - u_h(\mu)| d\Omega d\Omega_\mu}{|\Omega| |\Omega_\mu|}, \quad (12)$$

where u is the known manufactured solution, u_h is the numerical solution, $|\Omega|$ is the total physical volume (or area for 2D problems) of the domain Ω , and $|\Omega_\mu|$ is the total hypervolume of the parameter space Ω_μ . The error function $|u - u_h|$ is numerically integrated in the physical space using a quadrature rule with a very high-order approximation [66] (which produces an exact integration for up to degree 13 polynomial functions) over triangular elements of the mesh. A composite 3-point Gauss-Legendre quadrature with $N_\mu = 32$ uniform subdivisions is used in the global parameter space. The resulting approximation of the error can be written as:

$$\text{average error} \approx \frac{\Delta\mu}{2 |\Omega| |\Omega_\mu|} \sum_{j=1}^{N_\mu} \sum_{l=1}^3 w_l f_\Omega \left(\frac{\mu_j - \mu_{j-1}}{2} \xi_l + \frac{\mu_j + \mu_{j-1}}{2} \right) \quad (13)$$

where $\{(w_l, \xi_l)\}_{l=1,3}$ denote the Gauss-Legendre quadrature weights and coordinates on the reference interval $[-1, 1]$, and $f_\Omega(\mu) = \sum_{i=1}^{N_T} \int_{\Delta_i} |u(\mu) - u_h(\mu)| dA$ is the integrated error for each triangle Δ_i , aggregated over all the N_T triangles of the mesh. The uniformly subdivided parameter value $\mu_j = \mu_0 + j \Delta\mu$, with $\Delta\mu = (\mu_{N_\mu} - \mu_0) / N_\mu$, where μ_0 and μ_{N_μ} are minimum and maximum parameter values respectively. The error between the partitioned and unpartitioned approaches is compared for a few test problems in the following subsections.

4.1. Heat conduction in square plate with moving wave discontinuity

In this test problem, the 2D manufactured analytical solution in a square plate $\Omega = [-1, 1] \times [-1, 1]$ is given as:

$$u_1(x, y) = T(x, y) = \frac{1}{2} \left(\tanh \left(\frac{2x - 2\bar{x} + w}{2dw} \right) - \tanh \left(\frac{2x - 2\bar{x} - w}{2dw} \right) \right), \quad (14)$$

where \bar{x} is the wave location, $w = 1/4$ is the width of the wave and $dw = 1/20$ is the width of the discontinuity. The corresponding heat source term f in Equation (9) can be calculated analytically using a constant diffusivity $\alpha = 1$, as:

$$f_1(x, y) = \frac{\text{sech}^2 \left(\frac{2x - 2\bar{x} - w}{2dw} \right) \tanh \left(\frac{2x - 2\bar{x} - w}{2dw} \right) - \text{sech}^2 \left(\frac{2x - 2\bar{x} + w}{2dw} \right) \tanh \left(\frac{2x - 2\bar{x} + w}{2dw} \right)}{dw^2},$$

The solution is parametrized by \bar{x} . The wave is simply translated with changing parameter, therefore we expect a uniform and symmetric partitioning of the parameter space as each solution is captured by a simply translated basis. Thus, for a chosen threshold, the parameter space is expected to converge at the same partition size everywhere.

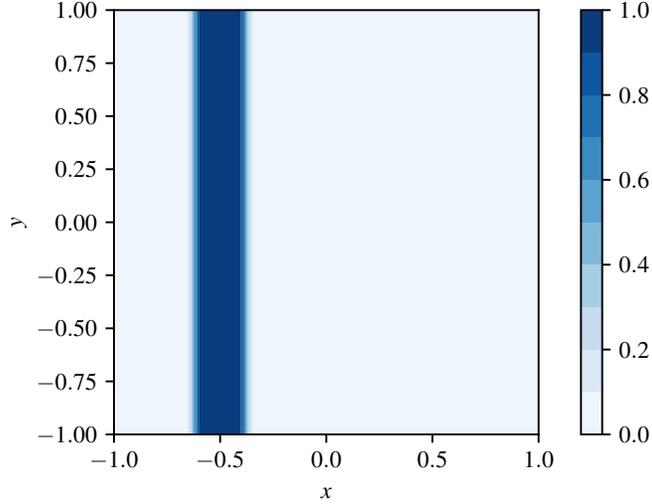


Figure 3: Plot of the manufactured solution (14), $u_1(x, y)$ with $\bar{x} = -0.5$ for the heat conduction problem with moving wave discontinuity.

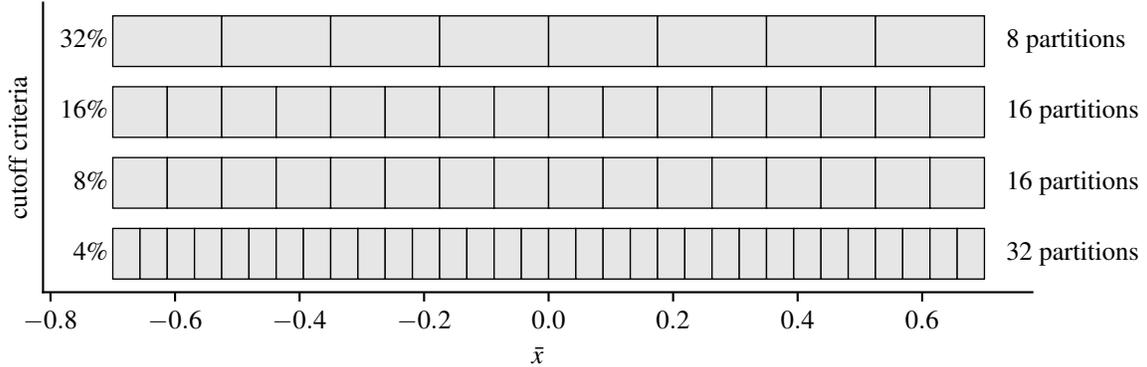


Figure 4: Partitions created by the algorithm using various cutoff criteria for the heat conduction problem with moving wave discontinuity.

The manufactured solution at $\bar{x} = -0.5$ is plotted in Figure 3. The parameter space $-0.7 \leq \bar{x} \leq 0.7$ is partitioned using the algorithm and models described in Section 3. The partitioning algorithm is used with various cutoff criteria of 4%, 8%, 16% and 32%, which results in the partitions shown in Figure 4. An adapted mesh is generated in each partition (with a number of elements similar to the global mesh $N_T \approx 10000$) for solving the problem and error calculation. The number of partitions generated and the average error (see Equation (13)) for each of the cutoff criterion is tabulated in Table 2, and compared with the error on the global mesh. It can be seen that each of the partitions are of the same size for a chosen threshold, as expected. As seen in Table 2 and Figure 5, the error is also much smaller for the partitioned parameter space, due to refined meshes capturing the behavior of the solution more accurately in each of the partition.

Table 2: Number of partitions and the average error with different cutoff criteria for the heat conduction problem with moving wave discontinuity.

Cutoff criteria	No. of partitions	Average error (with partitions)	Average global error (without partitions)
32%	8	8.94604×10^{-4}	2.78704×10^{-3}
16%	16	7.90366×10^{-4}	
8%	16	7.90366×10^{-4}	
4%	32	7.38330×10^{-4}	

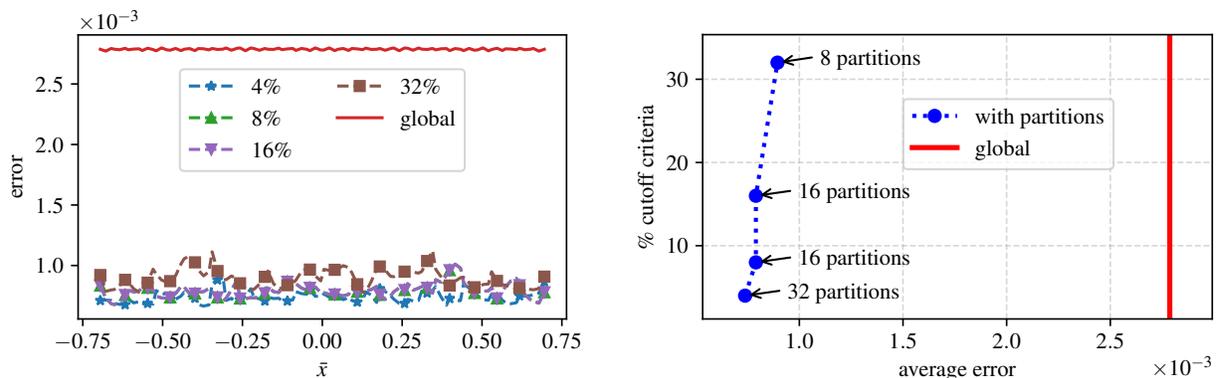


Figure 5: Plot of the error with different cutoff criteria for the heat conduction problem with moving wave discontinuity.

4.2. Heat conduction in square plate with radial discontinuity

In this test problem, the 2D analytical temperature distribution in a square plate is given as:

$$u_3(x, y) = T(x, y) = \frac{1}{2} \left(\tanh \left(\frac{r - \bar{r}}{dw} \right) + 3 \right), \quad (15)$$

where $r = \sqrt{x^2 + y^2}$, \bar{r} is the radial location of discontinuity and $dw = 1/20$ is the width of the discontinuity. The corresponding heat source term f in Equation (9) can be calculated analytically using a constant diffusivity $\alpha = 1$ as:

$$f_3(x, y) = \frac{1}{2r dw^2} \operatorname{sech}^2 \left(\frac{r - \bar{r}}{dw} \right) \left(dw - 2r \tanh \left(\frac{r - \bar{r}}{dw} \right) \right) \quad (16)$$

The solution is parametrized by \bar{r} . Unlike the previous test case in Subsection 4.1, this problem exhibits non-uniform behavior in the parameter space. This is because when \bar{r} is small, changing its value has a minimal impact on the solution and the corresponding basis functions compared to when the value of \bar{r} is large. Therefore, we expect non-uniform partitioning of the parameter space, with smaller partitions at larger \bar{r} to capture the quickly changing basis functions.

The manufactured solution at $\bar{r} = 0.65$ is plotted in Figure 6. The parameter space $0 \leq \bar{r} \leq 1$ is partitioned using the algorithm and models described in Section 3. The partitioning algorithm is used with various cutoff criteria of 0.5%, 1%, 2% and 4%, which results in the partitions shown in Figure 7. A refined

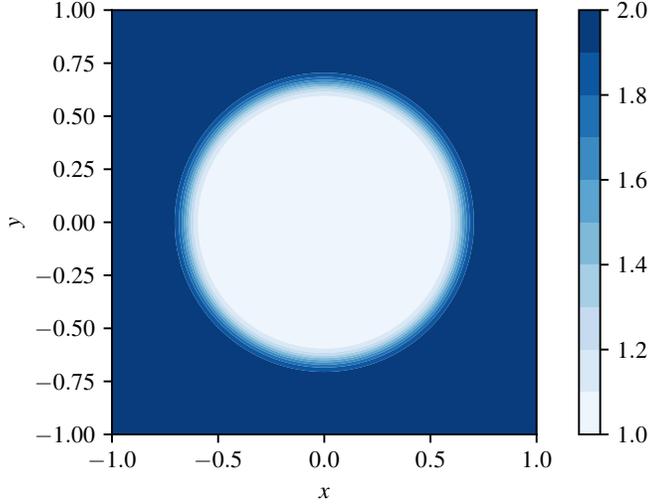


Figure 6: Plot of the manufactured solution (15), $u_3(x, y)$ with $\bar{r} = 0.65$ for the heat conduction problem with radial discontinuity.

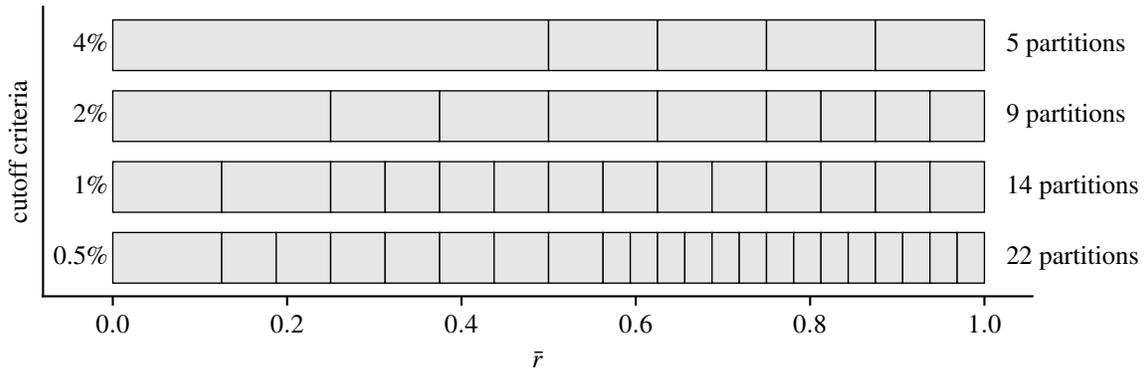


Figure 7: Partitions created by the algorithm using various cutoff criteria for the heat conduction problem with radial discontinuity.

mesh is generated in each partition (with a number of elements similar to the global mesh $N_T \approx 10000$) for solving the problem and error calculation. The number of partitions generated and the average error (see Equation (12)) for each of the cutoff criterion is tabulated in Table 3, plotted in Figure 8 and is compared with the error on the global mesh. As expected, here we distinctly observe larger partitions when \bar{r} is close to zero and smaller partitions when \bar{r} is close to 1.

The average errors are also much reduced with partitioning, as the mesh is adapted for each partition according to the specific behavior of the solution within each partition. In the left handside plot of Figure 8, which shows the error distribution at various locations in the parameter space, we see that the error on the adapted mesh is higher when $\bar{r} \approx 0.37$ at 4% cutoff criterion. This happens because the mesh is constructed on a large partition with sampling points away from this location. This can be easily overcome by using more sampling points while computing the metric intersection for mesh adaptation. However, we use only 3

Table 3: Number of partitions and the average error with different cutoff criteria for the heat conduction problem with radial discontinuity.

Cutoff criteria	No. of partitions	Average error (with partitions)	Average global error (without partitions)
4%	5	1.34026×10^{-3}	2.23625×10^{-3}
2%	9	7.39198×10^{-4}	
1%	14	6.43605×10^{-4}	
0.5%	22	6.44268×10^{-4}	

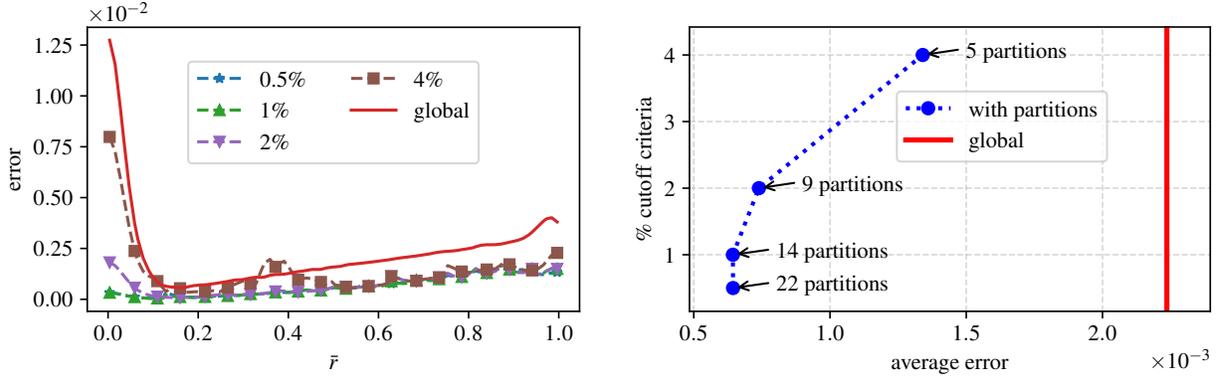


Figure 8: Plot of the error with different cutoff criteria for the heat conduction problem with radial discontinuity.

points per partition (as shown in Figure 2) for metric intersection to maintain uniformity in the results.

In all the test cases used for error analysis, we observe that the error stops dropping beyond certain cutoff criterion. This is due to the predefined number of elements for global and local adapted meshes. To reduce the error further a larger number of elements is needed, thus increasing the computational cost of the simulations.

5. Application to Heat Conduction and Flow Problems

In this section we discuss applications to more interesting problems where no closed-form analytical solution exist. The algorithm can work with any number of parameters. However, we will primarily present results for two- and three-dimensional parameter spaces for better visualization. We then critically analyze and justify the resulting partitions based on the physical understanding of the problem.

In all the following cases, once the snapshots are computed, building the k-d tree takes a couple of minutes on a recent laptop.

5.1. Heat conduction problems

Heat conduction in annular cylinder. In this test problem, heat transfer takes place in a two-dimensional annular domain with an inner radius $r_i = 0.25$ and an outer radius $r_o = 1.0$ as shown in Figure 9. The

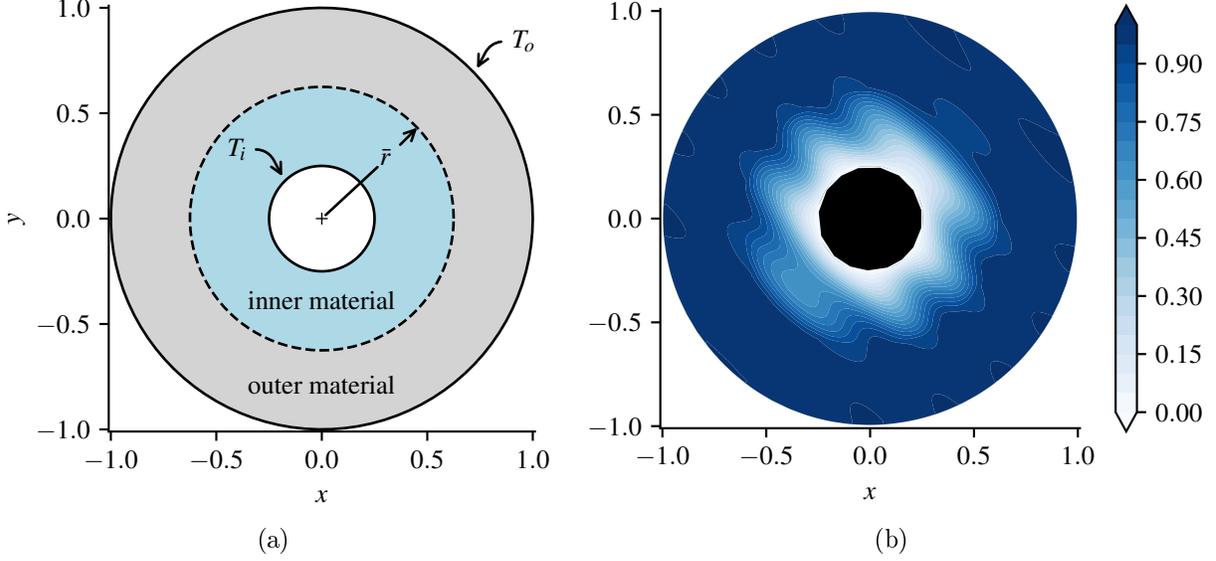


Figure 9: (a) Geometry of the annular cylinder used for the heat conduction problem with inner wall temperature T_i , outer wall temperature T_o and material discontinuity at \bar{r} (b) A sample solution at $T_i = 0$, $T_o = 1$, $a = 50$, $\omega = 15$, $\alpha_{in} = 1$, $\alpha_{out} = 10$, $\bar{r} = 0.625$.

physics of the heat conduction is governed by Equation (9). The heat source is chosen as:

$$f(x, y) = a \sin(\omega(x + y)), \quad (17)$$

where a is the amplitude and ω is the wave number of the source term. The temperature at the inner diameter and outer diameter are T_i and T_o respectively. The coefficient of thermal diffusivity is given by:

$$\alpha(x, y) = \begin{cases} \alpha_{in} & \text{if } \sqrt{x^2 + y^2} \leq \bar{r} \\ \alpha_{out} & \text{if } \sqrt{x^2 + y^2} > \bar{r}, \end{cases} \quad (18)$$

where α_{in} is the thermal diffusivity of the inner region and α_{out} is the thermal diffusivity of the outer region. The coefficient of thermal diffusivity is discontinuous at a radial distance of \bar{r} . The parameter space of the problem is defined by the parameters $T_i, T_o, a, \omega, \alpha_{in}, \alpha_{out}$ and \bar{r} . In the parameter space, the solution varies non-linearly with respect to ω and \bar{r} , while the solution has a linear dependence on the remaining parameters.

The first test case uses $\bar{r} - T_i$ as a parameter space, with $0.1 \leq \bar{r} \leq 1.1$ and $0 \leq T_i \leq 10$. The remaining parameters are maintained constant: $T_o = 1$, $a = 50$, $\omega = 10$, $\alpha_{in} = 1$ and $\alpha_{out} = 10$. The non-linear dependence of the solution is only in the \bar{r} direction. Thus, the expectation is that the space would be split only in the \bar{r} direction. Figure 10(a) shows the partitioned $\bar{r} - T_i$ parameter space using a cutoff criterion of 1%. The partition is large for very small values of \bar{r} because when the radius of discontinuity is smaller than the inner radius, the problem depends only on T_i which makes the problem linear. The figure also shows

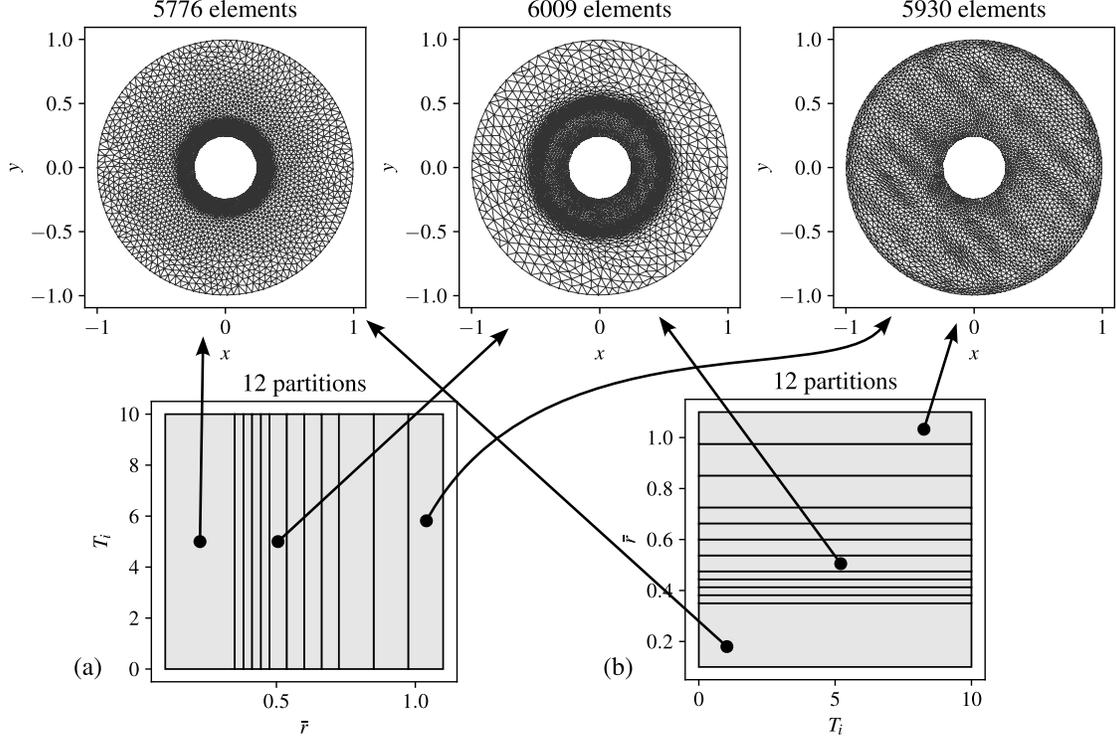


Figure 10: The partitions and meshes for the $\bar{r} - T_i$ parameter space using a cutoff criterion of 1%.

the adapted mesh in different regions of the parameter space. We see that the partitions correspond to very different meshes, hence a gain in computing time and accuracy. Moreover, as the algorithm is symmetric, the partitions swap directions when the parameters are swapped and the new couple of parameters considered is $T_i - \bar{r}$, as seen in Figure 10(b). The local singular values are computed using sampled solutions on the adapted mesh for each local partition, while the global SVD is performed using a uniform mesh with a similar number of elements. The singular values for the global and local parameter spaces are plotted in Figure 11. It can be observed that the singular values for the local partitions decay very quickly resulting in a far efficient reduced order model. Moreover, as the mesh is tailored for each partition, the solutions are accurate compared to a global uniform mesh with the same number of elements, thus leading to further computational efficiency.

The second variation of the test case uses the parameters $\omega - \bar{r}$. The solution has a non-linear dependence on both ω and \bar{r} . Thus, we expect to obtain a significantly more intricate partitioning in this case. Figure 12 shows the partitioned $\omega - \bar{r}$ parameter space with the mesh at few selected regions. It can be seen that the mesh is adapted according to the behavior of the physics within the partition, thus resolving the solutions accurately. It is interesting to observe (in Figure 12) that the higher wave numbers are concentrated in a larger partition. This is because at lower wave numbers the solution changes rapidly with changing wave number, compared to that at higher wave numbers. Similarly, at larger radii of discontinuity the solution changes rapidly with changing radius, compared to smaller radii of discontinuity. Therefore, the partitions are smaller

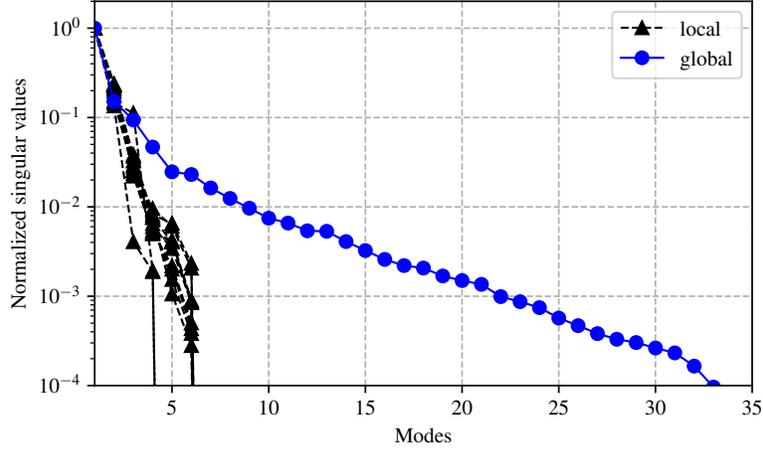


Figure 11: The decay of global and local singular values for $\bar{r} - T_i$ parameter space.

at larger radius and smaller wave numbers, as more basis functions are necessary to capture the solution with a given cutoff criterion. As before, it can be seen in Figure 13 that the singular values decay rapidly in all the local partitioned spaces compared to those of the global parameter space. The algorithm produces efficient partitions of the parameter space so as to obtain the modes for reduced order models and meshes suitable for capturing the local physics accurately.

The third variation of the test case uses two non-linear parameters ω and \bar{r} and a linear parameter T_i . The three-dimensional parameter space is therefore given by $\omega - \bar{r} - T_i$. As expected the partitioning algorithm only partitions in the directions of the non-linear parameters ω and \bar{r} as seen in Figure 14. Moreover, we obtain smaller partitions when the discontinuity radius is large and the wave number is small, which is similar to the observation in the previous test case using $\omega - \bar{r}$. Also, as can be seen in Figure 15, the singular values decay rapidly in all the local partitioned spaces compared to those of the global parameter space, thus resulting in an optimal reduced order model with desired accuracy.

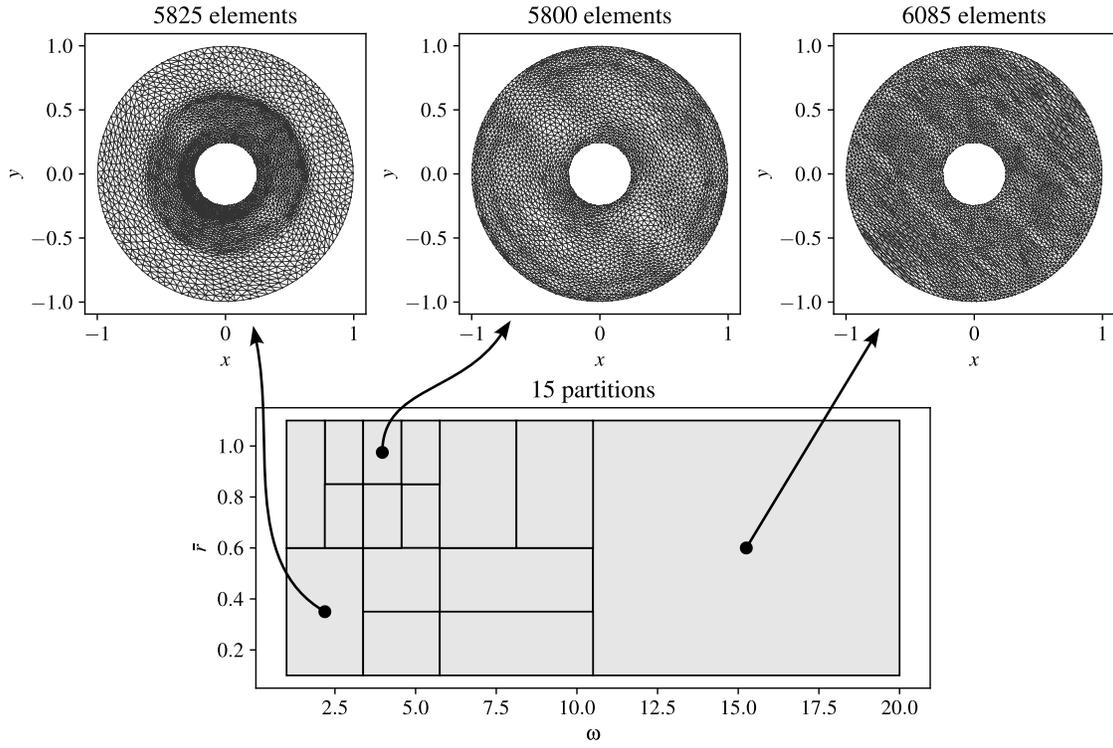


Figure 12: The partitions and mesh for the $\omega - \bar{r}$ parameter space using a cutoff criterion of 5%.

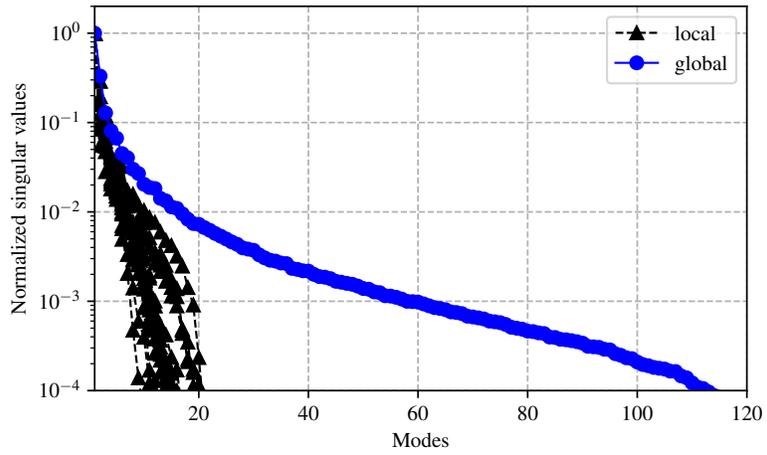


Figure 13: The decay of global and local singular values for $\omega - \bar{r}$ parameter space.

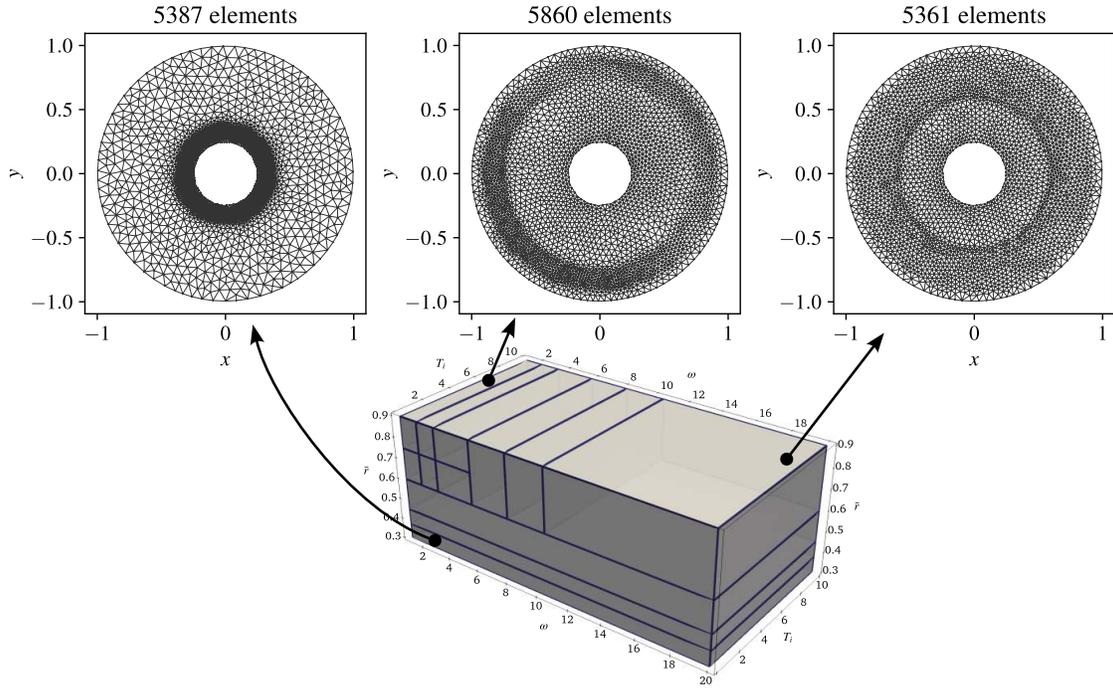


Figure 14: The partitions and mesh for the $\omega - \bar{r} - T_i$ parameter space using a cutoff criterion of 2.5%.

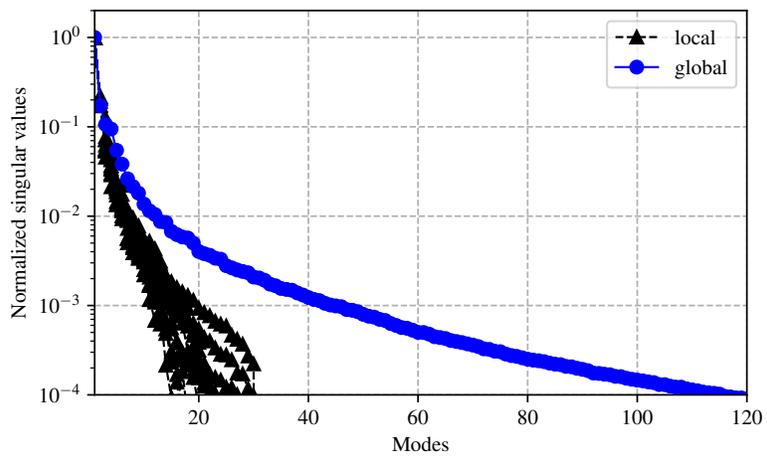


Figure 15: The decay of global and local singular values for $\omega - \bar{r} - T_i$ parameter space.

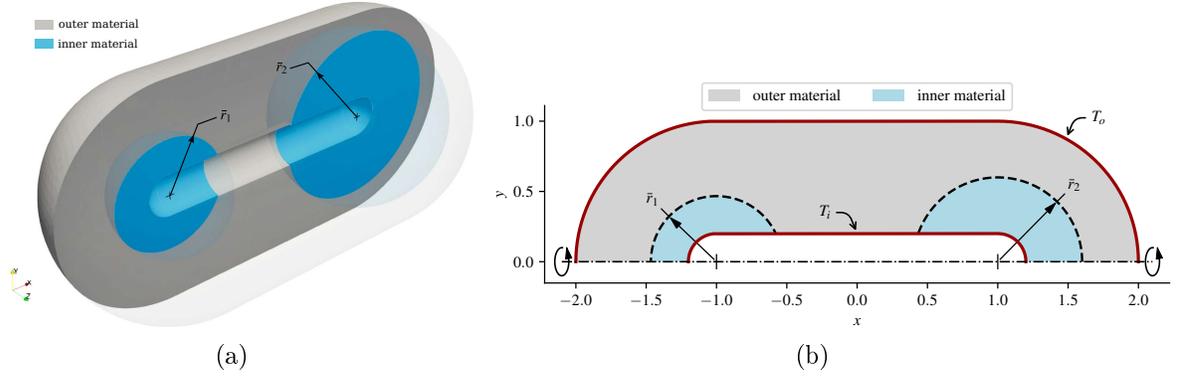


Figure 16: Geometry of the 3D cylinder with two materials where the parameters are the radii of the two inner material regions, \bar{r}_1 and \bar{r}_2 . (a) Cut-section of three-dimensional geometry, (b) Area of revolution with dimensions.

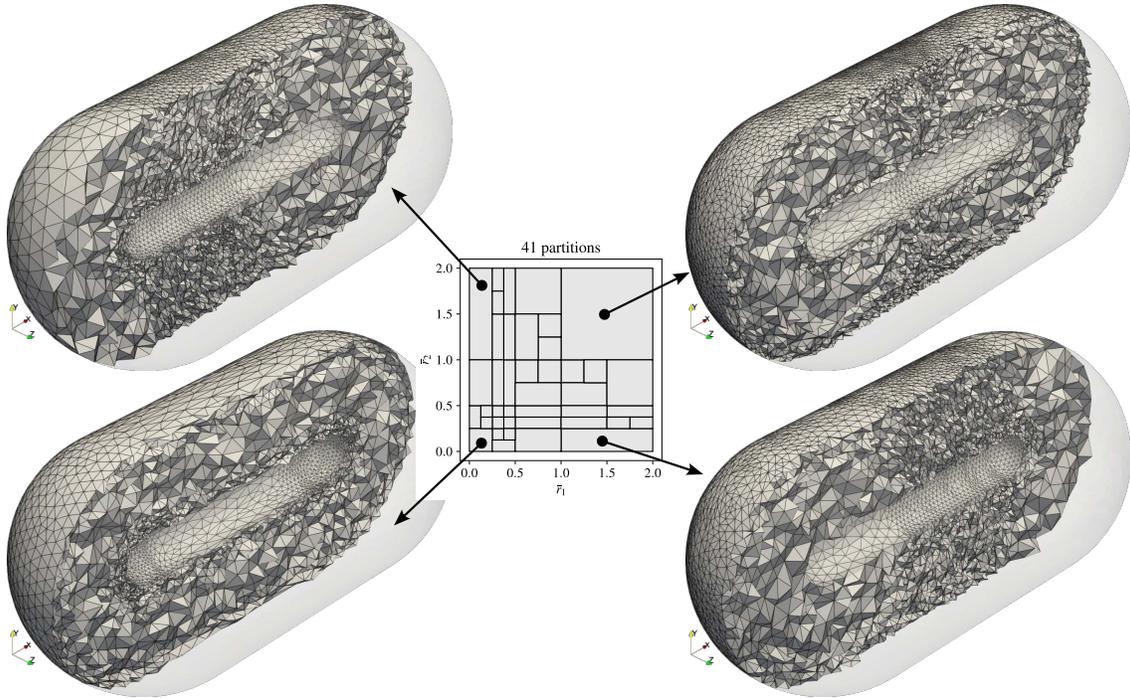


Figure 17: The partitions and mesh for the $\bar{r}_1 - \bar{r}_2$ parameter space using a cutoff criterion of 1.5%.

Heat conduction in a 3D cylinder. This test problem is an extension of the previous heat conduction problem using a three-dimensional geometry for the physical domain. The geometry is cylindrical with hemispherical ends and a hollow slot in the middle as shown in Figure 16 (a). The area of revolution and detailed dimensions are shown in Figure 16 (b). The inner wall of the cylinder is maintained at a temperature of $T_i = 0$ and the outer wall is maintained at a temperature of $T_o = 1$. The computational domain is composed of two materials, the inner and the outer material, as shown in Figure 16. The thermal diffusivity of the inner material is $\alpha_{in} = 1$ and the outer material has thermal diffusivity of $\alpha_{out} = 10$. The thermal diffusivity of

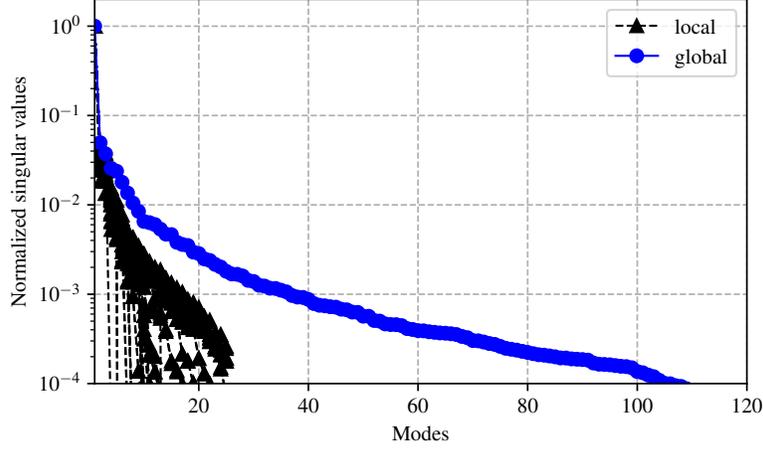


Figure 18: The decay of global and local singular values for $\bar{r}_1 - \bar{r}_2$ parameter space.

the domain is defined as:

$$\alpha(x, y, z) = \begin{cases} \alpha_{\text{in}} & \text{if } \sqrt{(x+1)^2 + y^2 + z^2} \leq \bar{r}_1 \\ \alpha_{\text{in}} & \text{if } \sqrt{(x-1)^2 + y^2 + z^2} \leq \bar{r}_2 \\ \alpha_{\text{out}} & \text{otherwise} \end{cases} .$$

The discontinuities of the material are at a spherical radial distance of \bar{r}_1 and \bar{r}_2 for the left and right handside inner materials, respectively. The radii of the discontinuities, $0 < \bar{r}_1 \leq 2$ and $0 < \bar{r}_2 \leq 2$ define the problem's parameter space. It can be seen that there are three material regions when $\bar{r}_1 + \bar{r}_2 < 2$ and two discontinuous regions when the inner materials merge at $\bar{r}_1 + \bar{r}_2 > 2$. The parameters of the problem are the radii of the two inner material regions, \bar{r}_1 and \bar{r}_2 .

The problem is symmetric about the $\bar{r}_1 = \bar{r}_2$ line in the $\bar{r}_1 - \bar{r}_2$ parameter space. The partitions, dividing the parameter space, are therefore expected to exhibit symmetry about the $\bar{r}_1 = \bar{r}_2$ line. We can see in Figure 17 that, except for one partition, the global parameter space is divided into symmetric sub-spaces about the 45° line. The 41 partitions in Figure 17 are generated using a cutoff criterion of 1.5%. In fact, we have observed that reducing the cutoff criterion to 1%, the algorithm subdivides these partitions further leading to 92 perfectly symmetric partitions. Figure 17 also shows some of the meshes generated, targeting 100,000 elements, using metric intersection and multiscale metric tensor (discussed in Subsection 2.1) using sampled solutions within the partition.

Figure 18 depicts the decay of singular values corresponding to the POD modes for the global parameter space and partitioned local parameter spaces. It can be observed that the normalized singular values decay very rapidly for the partitioned spaces, thus resulting in significantly more efficient reduced-order models

compared to a single model for the complete parameter space.

5.2. Compressible flow problems

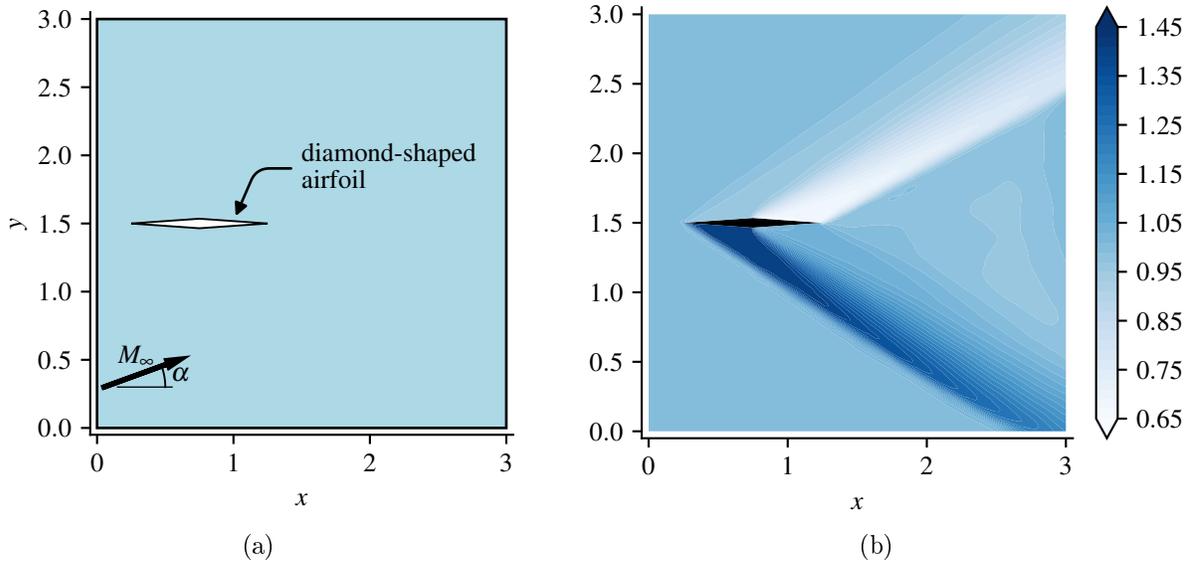


Figure 19: (a) Geometry of the diamond-shaped airfoil placed in supersonic flow (b) A sample solution of density at $M_\infty = 2$, $\alpha = 5^\circ$.

Supersonic flow over a 2D diamond-shaped airfoil. In this test problem, a diamond-shaped airfoil with chord length $c = 1$ and thickness $t = 0.07c$, as shown in Figure 19, is placed in a supersonic flow at various angles of attack. The physics of the compressible flow is governed by Equation (10). The left boundary is a supersonic inlet, while non-reflecting boundary conditions are applied at the top, bottom and right boundaries. The parametric space is defined by the freestream Mach number $1.5 \leq M_\infty \leq 3$ and angle of attack $-10^\circ \leq \alpha \leq 10^\circ$ which are used in the inflow boundary conditions of Equation (10). The solution snapshots are created using the density of the fluid ρ as it captures the variations due to shocks, contact discontinuities and the expansion waves. The sparse sampling of the solution in the parameter space is guided by the partitioning algorithm. The partitions, using 1% cutoff criterion, and some of the representative meshes are shown in Figure 20. As expected the partitioning and the meshes are symmetric about the zero angle of attack, since the problem is symmetric about $\alpha = 0$. The algorithm generates smaller partitions at larger angles of attack and lower Mach numbers. This is because the oblique shock angle changes rapidly when the supersonic Mach number is close to unity and the flow physics also exhibits greater sensitivity at high angles of attack. This behavior can be verified using the theoretical relation between deflection angle and oblique shock angle with varying Mach numbers (see for example [67]). These characteristics of the solution in the parameter space are automatically identified by the partitioning algorithm. Figure 21 shows the decay of POD modes of all the partitions and compares them with the decay of the global POD modes. As seen

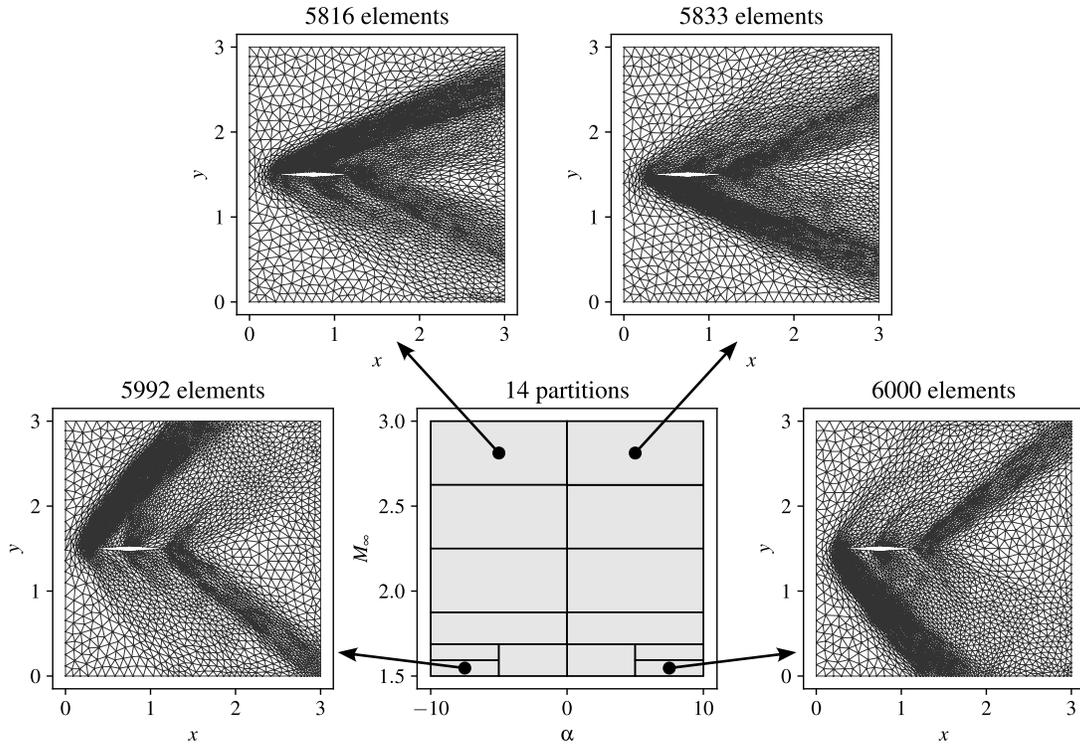


Figure 20: The partitions and meshes for the $\alpha - M_\infty$ parameter space using a cutoff criterion of 1%.

in the figure the local modes decay very quickly compared to the global modes, enabling the development of significantly more efficient local reduced-order models. The tailored meshes for each partition further enhances the accuracy of the local ROMs.

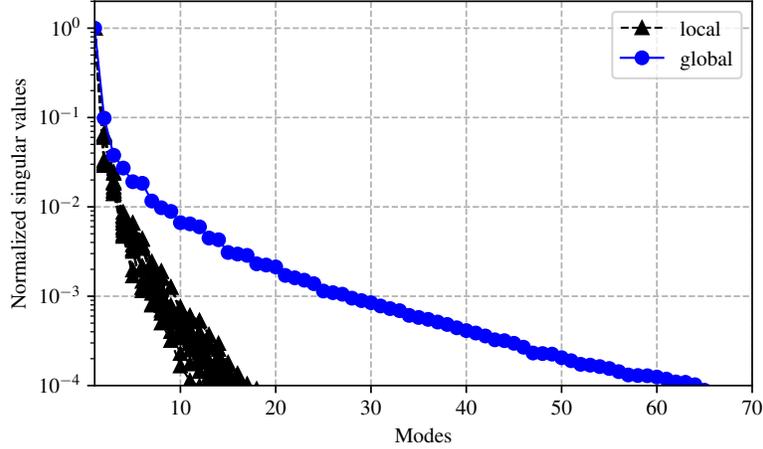


Figure 21: The decay of global and local singular values for $\alpha - M_\infty$ parameter space.

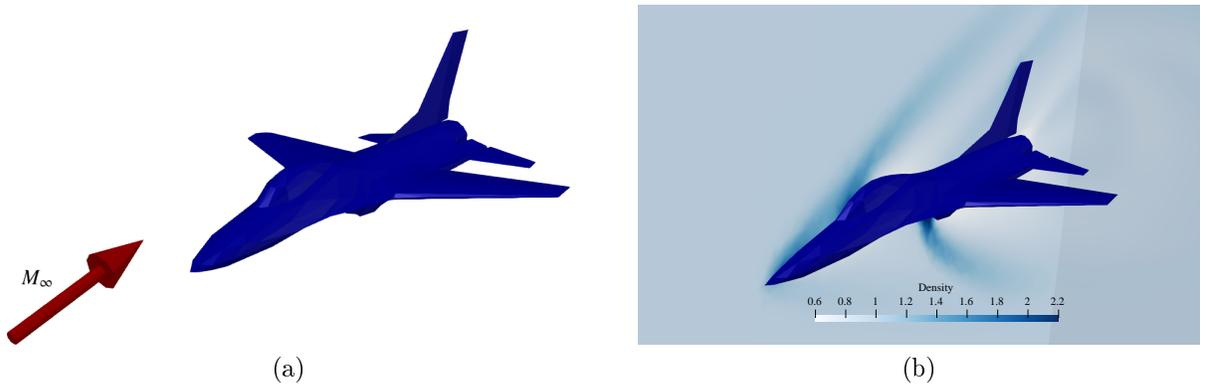


Figure 22: Supersonic flow over an airplane model in a wind tunnel. (a) Airplane model, (b) A sample solution of density at $M_\infty = 2.5$, $\alpha = -2^\circ$, $\beta = 0^\circ$.

Supersonic flow over a 3D airplane. In this problem, we consider a three-dimensional supersonic flow over an airplane in a wind tunnel. The geometric model is based on the simplified F-16 aircraft without its engine. The geometry of the airplane and a sample solution are shown in Figure 22 (a) and (b) respectively. The three parameters of this problem are the freestream Mach number M_∞ , the airplane's angle of attack α , and the airplane's angle of yaw β . The parameter space is defined as $1.5 \leq M_\infty \leq 3$, $-5^\circ \leq \alpha \leq 5^\circ$ and $-5^\circ \leq \beta \leq 5^\circ$. In the parameter space, the solutions produce diverse and intricate flow features resulting from interactions between the oblique shock waves, contact discontinuities and expansion waves. Therefore, the solutions cannot be efficiently captured by a single mesh across the entire parameter space.

In this problem, as the flow is supersonic, the flow field only changes downstream of the airplane. Therefore, it is computationally wasteful to use a uniformly refined mesh in the complete flow domain during partitioning. Hence, we do not use a uniform mesh for parameter space partitioning for this problem. Instead, the solution is uniformly sampled across the global parameter space to generate an appropriate adapted

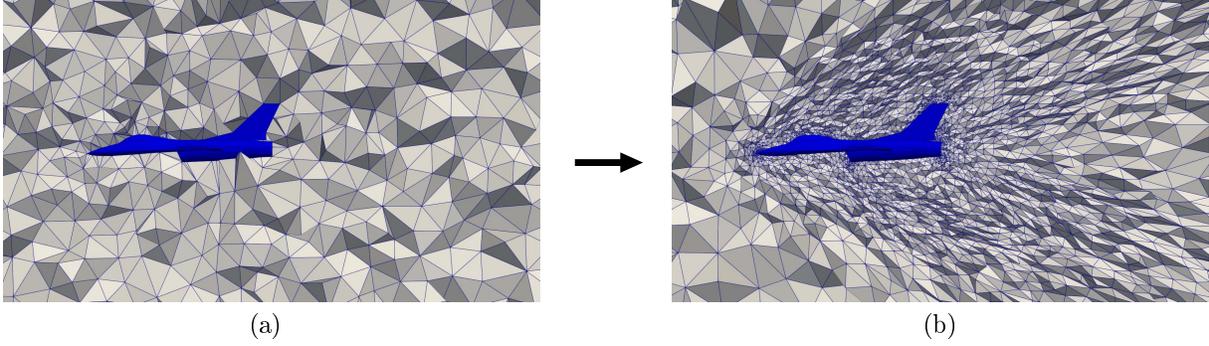


Figure 23: Cut section of the meshes used for the supersonic flow over the airplane (a) Starting uniform coarse mesh, capturing only the airplane geometry. (b) Mesh used for the partitioning algorithm, obtained by intersection of solutions sampled from the global parameter space.

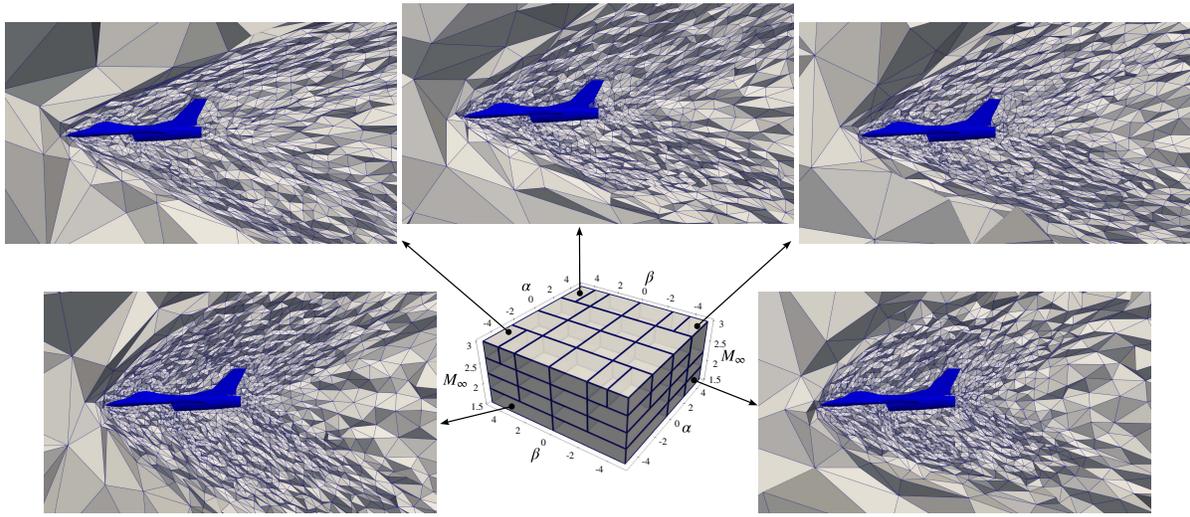


Figure 24: The partitions and mesh for the $M_\infty - \alpha - \beta$ parameter space using a cutoff criterion of 0.5%.

mesh using metric intersection, as discussed in Section 2.1. This provides a well refined mesh in the domain of influence, and a very coarse mesh in the zone of silence, for all the simulations run by the partitioning algorithm. The starting mesh and the adapted mesh used for parameter space partitioning are shown in Figure 23 (a) and (b) respectively.

The solution snapshots for the partitioning algorithm are created using the density ρ of the fluid and the sparse greedy sampling of the parameter space is guided by the partitioning algorithm. The partitioning algorithm is used to subdivide the parameter space with a cutoff criterion of 0.5%. This splits the parameter space into 52 partitions, such that each partition needs only a few POD modes to capture the solutions within the partition. The partitions generated by the algorithm for the airplane problem and few representative meshes, created by sampling solutions within the partition, are shown in Figure 24. For each partition, the mesh is generated to have about 100,000 elements by adjusting the normalisation factor in the metric as described in Subsection 2.1.

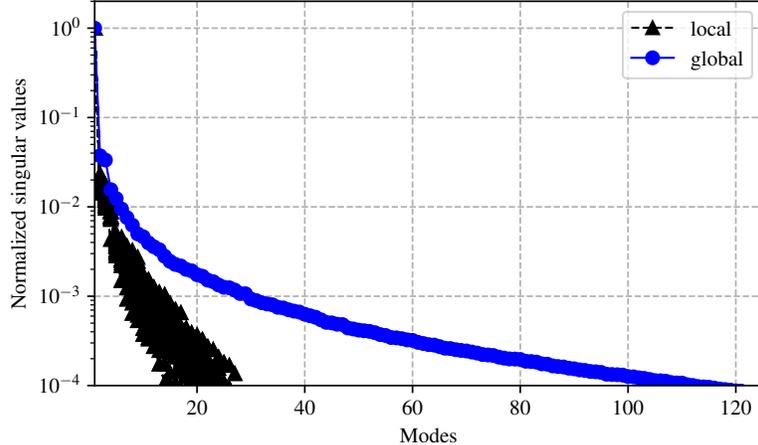


Figure 25: The decay of global and local singular values for $M_\infty - \alpha - \beta$ parameter space.

As observed in Figure 24, the algorithm produces symmetric partitioning about $\beta = 0$ plane. This is because the problem is symmetric about zero angle of yaw due to symmetry of the airplane. On the other hand, the partitions are not symmetric about $\alpha = 0$ plane. This is due to the airplane’s asymmetric features, such as the cockpit and engine intake. The intricate interactions between the shocks, expansion waves and contact discontinuities, are effectively captured by the meshes produced for each partition. Also, the number of modes necessary to capture the physics is much smaller for all the partitions, as seen by the rapid decay of the singular values in Figure 25.

6. Conclusion

In this work we have developed an algorithm for efficient partitioning the parameter space for mesh clustering and model order reduction (MOR). The algorithm analyzes the variation of solutions in the parameter space using proper orthogonal decomposition (POD). The parameter space is recursively split into smaller sub-spaces until the solutions in the resulting sub-space are captured by a linear reduced order model (ROM), with a chosen cutoff criterion. At each level of the recursive algorithm, the direction of maximum variation is identified using the right singular vectors, which are used to decide the direction of the split. The resulting local ROMs, i.e. ROM generated for each partition, are demonstrated to require significantly fewer modes to capture the solutions, hence making them computationally efficient compared to the global ROM. The recursive procedure used by the algorithm requires only sparse sampling of the parameter space thus optimizing the number of expensive solution computations. The caching and reuse of the solutions in the neighboring partitions further optimizes the computational cost. An anisotropic metric intersection procedure is used to generate a mesh tailored to each partition to increase accuracy and further reduce the computational cost.

The effectiveness of the algorithm is demonstrated through error analysis on various test cases using

manufactured solutions. It is observed that the algorithm significantly improves the accuracy of the solutions in the complete parameter space. The algorithm is also applied on a more complicated multi-parameter heat conduction problem with material discontinuities. The algorithm correctly identifies the non-linear parameters, and optimally partitions the non-linear parameter spaces without splitting the linear parameters. Finally, the algorithm is used to partition the parameter space of compressible supersonic flow over an airfoil and an airplane. The algorithm again optimally splits the parameter space, generating smaller partitions where the solution changes rapidly and larger partitions where it varies slowly. In all the studied problems, the modes for building local and global ROMs are also computed, and the associated normalized singular values are analyzed. It is observed that the local ROMs, built for partitioned parameter space, require 5 to 10 times less modes compared to the global ROM, built for the whole parameter space. Therefore, the computational cost of the ROMs in the online stage is reduced by a significant factor ranging from 25 to 100. The computational cost can be reduced further by choosing a smaller cutoff criterion, at an expense of disk space for more partitions. Moreover, the ROMs are also more accurate as the mesh is specific to the local parameter space.

An important feature of the algorithm is that it is non intrusive, and therefore can be easily coupled with any solver or any parameterized function in general. Moreover, the application of the partitioned parameter space and customized meshes is not limited to just POD based reduced order models. The partitioned parameter space can also be used for efficiently solving optimization problems or generating highly optimized neural network based autoencoders, as these may also require computing the solution of the same problem for many parameter.

In the current work we have focused on steady-state problems with hierarchical clustering based on optimally-aligned splitting directions. In the current work we have chosen the k -dimensional tree clustering method which has the advantages of being simple and have a low complexity since cells in parameter spaces are always split in two independently on the dimension of the space. In the future, this approach can be extended to time-dependent problems with evolving geometries. Moreover, to enhance the convergence of the refinement method other techniques will be explored as e.g. k -means, or the use of cuts in directions non parallel to the axes, as well as agglomeration (i.e. regrouping) of partitions can be used to optimise the number of resulting clusters. The main challenge will be the trade-off between accuracy, and the cost of producing the final partitioned adaptive approximation, especially when the number of parameters becomes large.

Acknowledgments

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 955558. The JU receives support from the European Union’s Horizon 2020 research and innovation programme and Spain, Germany, France, Italy, Poland, Switzerland, Norway. Experiments presented in this paper were carried out using the PlaFRIM experimental testbed, supported by Inria, CNRS (LABRI and IMB), Université de Bordeaux, Bordeaux INP and Conseil Régional d’Aquitaine.

CRedit author statement

Sourabh P. Bhat: Methodology, Software, Validation, Investigation, Writing - Original Draft. **Nicolas Barral:** Conceptualization, Methodology, Investigation, Writing - Review & Editing. **Mario Richiuto:** Conceptualization, Methodology, Investigation, Writing - Review & Editing.

References

- [1] P. Frey, F. Alauzet, Anisotropic mesh adaptation for CFD computations, *Computer Methods in Applied Mechanics and Engineering* 194 (2005) 5068–5082. [doi:10.1016/j.cma.2004.11.025](https://doi.org/10.1016/j.cma.2004.11.025).
- [2] F. Alauzet, A. Loseille, A decade of progress on anisotropic mesh adaptation for computational fluid dynamics, *Computer-Aided Design* 72 (2016) 13–39. [doi:10.1016/j.cad.2015.09.005](https://doi.org/10.1016/j.cad.2015.09.005).
- [3] L. Frazza, A. Loseille, F. Alauzet, Anisotropic mesh adaptation for turbomachinery applications, *American Institute of Aeronautics and Astronautics*, 2017. [doi:10.2514/6.2017-3299](https://doi.org/10.2514/6.2017-3299).
- [4] I. Kissami, Numerical assessment of criteria for mesh adaptation in the finite volume solution of shallow water equations, *Advances in Applied Mathematics and Mechanics* 12 (2020) 503–526. [doi:10.4208/aamm.OA-2019-0011](https://doi.org/10.4208/aamm.OA-2019-0011).
- [5] M. Giacomini, R. Sevilla, A second-order face-centred finite volume method on general meshes with automatic mesh adaptation, *International Journal for Numerical Methods in Engineering* 121 (2020) 5227–5255. [doi:10.1002/nme.6428](https://doi.org/10.1002/nme.6428).
- [6] Y. Xia, J. Zheng, J. Zou, J. Zhang, G. Wang, Y. Zheng, Mesh adaptation for curing the pathological behaviors of an upwind scheme, *Communications in Computational Physics* 31 (2022) 626–644. [doi:10.4208/cicp.OA-2021-0131](https://doi.org/10.4208/cicp.OA-2021-0131).

- [7] P. LeGresley, J. Alonso, Airfoil design optimization using reduced order models based on proper orthogonal decomposition, American Institute of Aeronautics and Astronautics, 2000. doi:10.2514/6.2000-2545.
- [8] Y. dong Lang, A. Malacina, L. T. Biegler, S. Munteanu, J. I. Madsen, S. E. Zitney, Reduced order model based on principal component analysis for process simulation and optimization, Energy & Fuels 23 (2009) 1695–1706. doi:10.1021/ef800984v.
- [9] D. Amsallem, M. Zahr, Y. Choi, C. Farhat, Design optimization using hyper-reduced-order models, Structural and Multidisciplinary Optimization 51 (2015) 919–940. doi:10.1007/s00158-014-1183-y.
- [10] J. D. Jansen, L. J. Durlofsky, Use of reduced-order models in well control optimization, Optimization and Engineering 18 (2017) 105–132. doi:10.1007/s11081-016-9313-6.
- [11] W. Keiper, A. Milde, S. Volkwein, Reduced-order modeling (ROM) for simulation and optimization: powerful algorithms as key enablers for scientific computing, Springer, 2018.
- [12] Y. Choi, G. Oxberry, D. White, T. Kirchdoerfer, Accelerating design optimization using reduced order models, Preprint arXiv:1909.11320 (9 2019).
URL <http://arxiv.org/abs/1909.11320>
- [13] C. Mou, Z. Wang, D. R. Wells, X. Xie, T. Iliescu, Reduced order models for the quasi-geostrophic equations: A brief survey, Fluids 6 (2020) 16. doi:10.3390/fluids6010016.
- [14] D. Butti, S. K. Mangipudi, S. Rayapudi, Model order reduction based power system stabilizer design using improved whale optimization algorithm, IETE Journal of Research 69 (2023) 2144–2163. doi:10.1080/03772063.2021.1886875.
- [15] V. Dahlberg, A. Dalkint, M. Spicer, O. Amir, M. Wallin, Efficient buckling constrained topology optimization using reduced order modeling, Structural and Multidisciplinary Optimization 66 (2023) 161. doi:10.1007/s00158-023-03616-7.
- [16] D. Hartmann, M. Herz, U. Wever, Model Order Reduction a Key Technology for Digital Twins, Springer International Publishing, 2018, pp. 167–179. doi:10.1007/978-3-319-75319-5_8.
- [17] G. Aversano, A. Bellemans, Z. Li, A. Coussement, O. Gicquel, A. Parente, Application of reduced-order models based on pca & kriging for the development of digital twins of reacting flow applications, Computers & Chemical Engineering 121 (2019) 422–441. doi:10.1016/j.compchemeng.2018.09.022.

- [18] M. G. Kapteyn, D. J. Knezevic, K. Willcox, [Toward predictive digital twins via component-based reduced-order models and interpretable machine learning](#), American Institute of Aeronautics and Astronautics, 2020. doi:10.2514/6.2020-0418.
URL <https://arc.aiaa.org/doi/10.2514/6.2020-0418>
- [19] D. Hartmann, M. Herz, M. Paffrath, J. Rommes, T. Tamarozzi, H. V. der Auweraer, U. Wever, Model Order Reduction: Applications, Vol. 3, De Gruyter, 2021. doi:10.1515/9783110499001.
- [20] S. J. Salinger, Toward predictive digital twins for self-aware unmanned aerial vehicles : non-intrusive reduced order models and experimental data analysis (2021). doi:10.26153/tsw/14557.
- [21] M. Kapteyn, D. Knezevic, D. Huynh, M. Tran, K. Willcox, Data-driven physics-based digital twins via a library of component-based reduced-order models, International Journal for Numerical Methods in Engineering 123 (2022) 2986–3003. doi:10.1002/nme.6423.
- [22] S. Maulik, D. Riordan, J. Walsh, Dynamic reduction-based virtual models for digital twins—a comparative study, Applied Sciences 12 (2022) 7154. doi:10.3390/app12147154.
- [23] H. Gong, S. Cheng, Z. Chen, Q. Li, Data-enabled physics-informed machine learning for reduced-order modeling digital twin: Application to nuclear reactor physics, Nuclear Science and Engineering 196 (2022) 668–693. doi:10.1080/00295639.2021.2014752.
- [24] C. Zhong, S. Cheng, M. Kasoar, R. Arcucci, Reduced-order digital twin and latent data assimilation for global wildfire prediction, Natural Hazards and Earth System Sciences 23 (2023) 1755–1768. doi:10.5194/nhess-23-1755-2023.
- [25] V. Zambrano, I. Viejo, J. M. Rodriguez, G. Lopez, J. Alfonso, D. Caceres, M. Lopez-Blanco, P. Talasila, M. Sanchez, S. Calvo, [Interpretable and reusable reduced order models for digital twins in manufactory as a service](#), ADMOS2023, 2023.
URL https://www.scipedia.com/public/_2023a
- [26] F. Zhao, X. Zhou, S. He, C. Wang, L. Dong, S. N. Atluri, B-spline surface-based reduced-order modeling of nonplanar crack growth in structural digital twins, AIAA Journal 62 (2024) 360–373. doi:10.2514/1.J062959.
- [27] D. Amsallem, M. J. Zahr, C. Farhat, Nonlinear model order reduction based on local reduced-order bases, International Journal for Numerical Methods in Engineering 92 (2012) 891–916. doi:10.1002/nme.4371.

- [28] K. Washabaugh, D. Amsallem, M. Zahr, C. Farhat, Nonlinear model reduction for CFD problems using local reduced-order bases, American Institute of Aeronautics and Astronautics, 2012. doi:[10.2514/6.2012-2686](https://doi.org/10.2514/6.2012-2686).
- [29] E. A. Cutillo, G. Petite, K. Bizon, G. Continillo, Analysis of an innovative sampling strategy based on k -means clustering algorithm for pod and pod-deim reduced order models of a 2-d reaction-diffusion system, *Combustion Theory and Modelling* 27 (2023) 508–535. doi:[10.1080/13647830.2023.2174451](https://doi.org/10.1080/13647830.2023.2174451).
- [30] P. J. Rousseeuw, Silhouettes: A graphical aid to the interpretation and validation of cluster analysis, *Journal of Computational and Applied Mathematics* 20 (1987) 53–65. doi:[10.1016/0377-0427\(87\)90125-7](https://doi.org/10.1016/0377-0427(87)90125-7).
- [31] E. Hruschka, L. de Castro, R. Campello, Evolutionary algorithms for clustering gene-expression data, *IEEE*, 2004, pp. 403–406. doi:[10.1109/ICDM.2004.10073](https://doi.org/10.1109/ICDM.2004.10073).
- [32] F. Wang, H.-H. Franco-Penya, J. D. Kelleher, J. Pugh, R. Ross, An analysis of the application of simplified silhouette to the evaluation of k-means clustering validity, in: *Machine Learning and Data Mining in Pattern Recognition: 13th International Conference, MLDM 2017, New York, NY, USA, July 15-20, 2017, Proceedings 13*, Springer, 2017, pp. 291–305.
- [33] M. P. Tan, J. R. Broach, C. A. Floudas, A novel clustering approach and prediction of optimal number of clusters: global optimum search with enhanced positioning, *Journal of Global Optimization* 39 (2007) 323–346. doi:[10.1007/s10898-007-9140-6](https://doi.org/10.1007/s10898-007-9140-6).
- [34] B. Peherstorfer, D. Butnaru, K. Willcox, H.-J. Bungartz, Localized discrete empirical interpolation method, *SIAM Journal on Scientific Computing* 36 (2014) A168–A192. doi:[10.1137/130924408](https://doi.org/10.1137/130924408).
- [35] A. Paul-Dubois-Taine, D. Amsallem, An adaptive and efficient greedy procedure for the optimal training of parametric reduced-order models, *International Journal for Numerical Methods in Engineering* 102 (2015) 1262–1292. doi:[10.1002/nme.4759](https://doi.org/10.1002/nme.4759).
- [36] E. Agouzal, J.-P. Argaud, M. Bergmann, G. Ferté, T. Taddei, A projection-based reduced-order model for parametric quasi-static nonlinear mechanics using an open-source industrial code, *International Journal for Numerical Methods in Engineering* 125 (2 2024). doi:[10.1002/nme.7385](https://doi.org/10.1002/nme.7385).
- [37] D. Amsallem, B. Haasdonk, PEBL-ROM: Projection-error based local reduced-order models, *Advanced Modeling and Simulation in Engineering Sciences* 3 (2016) 6. doi:[10.1186/s40323-016-0059-7](https://doi.org/10.1186/s40323-016-0059-7).

- [38] J. L. Eftang, A. T. Patera, E. M. Rønquist, An "*hp*" certified reduced basis method for parametrized elliptic partial differential equations, *SIAM Journal on Scientific Computing* 32 (2010) 3170–3200. doi:[10.1137/090780122](https://doi.org/10.1137/090780122).
- [39] K. Carlberg, Adaptive h-refinement for reduced-order models, *International Journal for Numerical Methods in Engineering* 102 (2015) 1192–1210. doi:[10.1002/nme.4800](https://doi.org/10.1002/nme.4800).
- [40] Y.-E. Kang, S. Shon, K. Yee, Local non-intrusive reduced order modeling based on soft clustering and classification algorithm, *International Journal for Numerical Methods in Engineering* 123 (10) (2022) 2237–2261. doi:<https://doi.org/10.1002/nme.6934>.
- [41] B. P. Ferreira, F. Andrade Pires, M. Bessa, Adaptivity for clustering-based reduced-order modeling of localized history-dependent phenomena, *Computer Methods in Applied Mechanics and Engineering* 393 (2022) 114726. doi:<https://doi.org/10.1016/j.cma.2022.114726>.
- [42] M. de Berg, O. Cheong, M. van Kreveld, M. Overmars, *Computational Geometry*, Springer Berlin Heidelberg, 2008. doi:[10.1007/978-3-540-77974-2](https://doi.org/10.1007/978-3-540-77974-2).
- [43] T. Barth, A 3-d upwind euler solver for unstructured meshes, American Institute of Aeronautics and Astronautics, 1991. doi:[10.2514/6.1991-1548](https://doi.org/10.2514/6.1991-1548).
- [44] S. P. Bhat, J. Mandal, [An improved hllc-type solver for incompressible two-phase fluid flows](#), *Computers & Fluids* 244 (2022) 105570. doi:[10.1016/j.compfluid.2022.105570](https://doi.org/10.1016/j.compfluid.2022.105570).
URL <https://linkinghub.elsevier.com/retrieve/pii/S0045793022001876>
- [45] A. Loseille, F. Alauzet, Continuous mesh framework part ii: Validations and applications, *SIAM Journal on Numerical Analysis* 49 (2011) 61–86. doi:[10.1137/10078654X](https://doi.org/10.1137/10078654X).
- [46] N. Barral, [Time-accurate anisotropic mesh adaptation for three-dimensional moving mesh problems](#) (2016).
URL <https://hal.inria.fr/tel-01284113v2>
- [47] C. Dapogny, C. Dobrzynski, P. Frey, Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems, *Journal of Computational Physics* 262 (2014) 358–378. doi:[10.1016/j.jcp.2014.01.005](https://doi.org/10.1016/j.jcp.2014.01.005).
- [48] L. Arpaia, H. Beaugendre, L. Cirrottola, A. Froehly, M. Lorini, L. Nouveau, M. Ricchiuto, h- and r-Adaptation on Simplicial Meshes Using MMG Tools, 2022, pp. 183–208. doi:[10.1007/978-3-030-92540-6_9](https://doi.org/10.1007/978-3-030-92540-6_9).

- [49] Mmg platform - upgrade your meshes, accessed: 2024-03-05.
URL <https://www.mmgtools.org/>
- [50] R. Everson, L. Sirovich, Karhunen-loève procedure for gappy data, *Journal of the Optical Society of America A* 12 (1995) 1657. doi:10.1364/JOSAA.12.001657.
- [51] M. Barrault, Y. Maday, N. C. Nguyen, A. T. Patera, An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations, *Comptes Rendus Mathématique* 339 (2004) 667–672. doi:10.1016/j.crma.2004.08.006.
- [52] K. Willcox, Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition, *Computers & Fluids* 35 (2006) 208–226. doi:10.1016/j.compfluid.2004.11.006.
- [53] S. Chaturantabut, D. C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, *SIAM Journal on Scientific Computing* 32 (2010) 2737–2764. doi:10.1137/090766498.
- [54] Z. Drmač, S. Gugercin, A new selection operator for the discrete empirical interpolation method—improved a priori error bound and extensions, *SIAM Journal on Scientific Computing* 38 (2016) A631–A648. doi:10.1137/15M1019271.
- [55] N. Barral, T. Taddei, I. Tifouti, Registration-based model reduction of parameterized pdes with spatio-parameter adaptivity, *Journal of Computational Physics* 499 (2024) 112727. doi:10.1016/j.jcp.2023.112727.
- [56] L. David, S. Lay, J. McDonald, *Linear Algebra and Its Applications*, 5th Edition, Pearson Education, 2021.
- [57] S. L. Brunton, J. N. Kutz, *Data-Driven Science and Engineering*, Cambridge University Press, 2022. doi:10.1017/9781009089517.
- [58] M. Gavish, D. L. Donoho, The optimal hard threshold for singular values is $4/\sqrt{3}$, *IEEE Transactions on Information Theory* 60 (2014) 5040–5053. doi:10.1109/TIT.2014.2323359.
- [59] T. Taddei, A registration method for model order reduction: data compression and geometry reduction, *SIAM Journal on Scientific Computing* 42 (2) (2020) A997–A1027.
- [60] R. Penrose, On best approximate solutions of linear matrix equations, *Mathematical Proceedings of the Cambridge Philosophical Society* 52 (1956) 17–19. doi:10.1017/S0305004100030929.
- [61] A. Ern, J.-L. Guermond, *Theory and practice of finite elements*, Vol. 159 of Applied Mathematical Sciences, Springer, 2004.

- [62] C. Hirsch, *Numerical Computation of Internal and External Flows*, Elsevier, 2007. doi:10.1016/B978-0-7506-6594-0.X5037-1.
URL <https://linkinghub.elsevier.com/retrieve/pii/B9780750665940X50371>
- [63] H. Nishikawa, A hyperbolic poisson solver for tetrahedral grids, *Journal of Computational Physics* 409 (2020) 109358. doi:10.1016/j.jcp.2020.109358.
- [64] E. F. Toro, M. Spruce, W. Speares, Restoration of the contact surface in the hll-riemann solver, *Shock Waves* 4 (1994) 25–34.
- [65] E. F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics*, Springer Berlin Heidelberg, 2009. doi:10.1007/b79761.
URL <http://www.springer.com/materials/mechanics/book/978-3-540-25202-3>
<http://link.springer.com/10.1007/b79761>
- [66] J. Berntsen, T. O. Espelid, Algorithm 706: Dcutri: an algorithm for adaptive cubature over a collection of triangles, *ACM Transactions on Mathematical Software* 18 (1992) 329–342. doi:10.1145/131766.131772.
- [67] J. D. Anderson, *Modern Compressible Flow: With Historical Perspective*, McGraw-Hill Education, 2003.