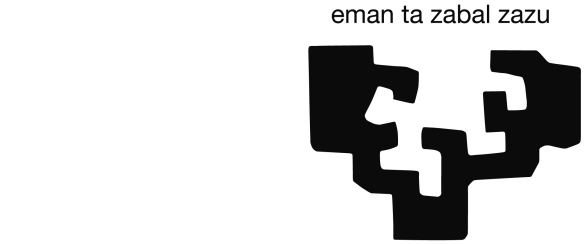
Doctoral Thesis

Snapshots location for Reduced Order Models: An approach based on Proper Orthogonal Decomposition and Mesh Adaptivity Techniques



Universidad del País Vasco

Euskal Herriko Unibertsitatea

Author:

Iñigo Bidaguren Diego

Supervisors:

Prof. Jesús María Blanco Ilzarbe

Prof. Lakhdar Remaki

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

2019

Doctoral Thesis

Snapshots location for Reduced Order Models: An approach based on Proper Orthogonal Decomposition and Mesh Adaptivity Techniques

Tesis Doctoral

Localización de los Snapshots en Modelos de Orden Reducido: Un enfoque basado en Proper Orthogonal Decomposition y técnicas de Mallado Adaptativo

Doktorego Tesia

Maila Murriztua duten Ereduen "Snapshot"-en kokapena: "Proper Orthogonal Decomposition" eta sare moldakorren tekniketan oinarritutako ikuspuntua

Author:
Iñigo Bidaguren Diego

Supervisors:
Prof. Jesús María Blanco Ilzarbe
Prof. Lakhdar Remaki

2019

Doctor of Philosophy

Snapshots location for Reduced Order Models: An approach based on Proper Orthogonal Decomposition and Mesh Adaptivity Techniques

Iñigo Bidaguren Diego

Abstract

Solving Partial Differential Equations (PDE), is a key issue in science and engineering since they are widely used in many real problem modelling including fluid mechanics, acoustics, heat and mass transfer ...etc. Exact solutions for PDEs and Ordinary Differential Equations (ODE) can be obtained in a very few and simplified cases only, numerical approximations are used instead (i.e. Finite Volume Method (FEM), Finite Volume Method (FVM)...). However, and despite the availability of important supercomputing facilities, due to the huge number of degrees of freedom these methods have, they may still suffer from cost-effectiveness performance. They are mainly two contexts where one need to solve PDEs with a lower computational cost: Real-time context and Many-query context. Examples for the former are: parameter-estimation, control, flying simulator..., and for the latter: optimization, multi model/scale simulation... To reduce significantly simulation time (often on the expense of accuracy) Reduced Order Modelling (ROM) techniques are introduced. The main idea is to reduce the initial solving space dimension (as in finite elements method) to a subspace with a significantly reduced dimension and then solve for the projected solutions.

Proper Orthogonal Decomposition (POD) is one of the more used ROM strategies. The presented work focuses on this technique, which has two challenging steps: (i) the snapshot location and (ii) the error estimate on the parameter space that drive the process to search new snapshot location. As a consequence of these two steps, POD applied to PDEs is considered as belonging to the well-known Greedy Algorithm family. This thesis brings a mesh adaptivity approach as the process to find the new parameter space locations. This process will be driven by a new error estimate based on *Leave One Out Cross Validation* (LOOCV) technique. We could say that this error estimate is universal, in the sense that it is not problem dependent. In addition, it is well known POD lack of accuracy when dealing with PDEs which solution contains shocks. Here, a new interpolation approach improves this, for shocked solutions. Finally, we present that the POD reduced basis is optimal just in average, and a new local basis is presented (Sorted Gram-Schmidt (SGS)) to be coupled with POD one. The criteria to decide which basis is better to be used for each new parameter value is defined as well. The whole proposed strategy efficiency is validated against a mathematical (exact) solutions of an incompressible, steady state flow equations, and on CFD solutions of an inviscid flow around a NACA0012 airfoil.

Acknowledgements

I would like to give gratitude to everyone who contributed to this thesis, making these years memorable.

The IT781-13 Basque Government consolidated Research group (A type, 2013-2018) for their support and for accepting me as team member, and to the next proposed IT1314-19 Research Group. This research was also supported by the grant BFA/DFB-6/12/TK/2012/0020 (Diputación Foral de Bizkaia) by financing the BBIPED-CFD platform development, by the Basque Government through the BERC 2018-2021 program and by the Spanish Ministry of Science, Innovation and Universities: BCAM Severo Ochoa accreditation SEV-2017-0718.

My supervisors Jesús María Blanco and Lakhdar Remaki, without them I could have never made this work. Research opened to me thanks to your permanent support and motivation, from the beauty of science on the first day, to communication and dissemination skills on the last one. Our Thesis.

The whole CFDMS team: Lakhdar, Ali, Goran, Carmen, Imanol, Laura, and Alireza. We spent time as colleagues during work time, and also as friends in our private time.

All BCAM staff for your care towards my well-being, and your support with all my difficulties.

All the researchers I had the luck to come across. Reda Mekhlouf in Laval University. Many researchers in UPV/EHU, especially Iñigo Albaina, Aitor Sarrionandia and Pedro Arriaga. UPV Laboratory technicians Iker, Iñigo, Cesar and Sergio, I hope I will not fall into the dark side. Finally, the reviewers and the Jury for their effort.

Many people and friends should be listed during these Ph.D years. Arkaitz, Miren, Iker G., Iker A., Jarno, June, Mikel, Aida, Xabi, Aitziber, Nagore, Izaskun, Miren, Eneko, Gorka, Unai, Maitane, Kepa, Irama, Mikel, Izaskun, Lander, Amaia, Deustuko Loosers, Javi, Sonia, Txoni, Igor, Txomin, Turrillas dynasty, Tijana, Arrate, Irati, Simone, Mario and old BCAM members. Also my great office mates.

Azkenean Oihane, BETI nire ondoan zaudelako. Y a mi familia, eskerrik asko Ama, Aita, Idoia, Anton, Markel, Paul, Izeko Juana, Maite, Victor, Olatz, Andoni, Lea eta Malen.

Iñigo Bidaguren Diego
May 14, 2019

Contents

List of Figures

Oihane, Uxue eta Peiorentzat

Resumen

La resolución de Ecuaciones Diferenciales Parciales (EDP), y mas concretamente, los sistemas dinámicos paramétricos, que representan la física real, es un tema clave en ciencia e ingeniería. La evaluación de esas ecuaciones es necesaria en muchos campos: mecánica clásica, mecánica de fluidos, acústica, transferencia de calor y masa... Se han utilizado varias técnicas para lograr esta tarea: tradicionalmente se ha recurrido a métodos de alto orden, como Método de Elementos Finitos (FEM) o Método de Volúmenes Finitos (FVM), y que debido a la gran cantidad de grados de libertad no se pueden resolver siempre tan rápido como uno hubiese deseado. En general, hay dos contextos principales en los que se necesita resolver estos Sistemas en EDP con un costo computacional más bajo: "contexto en tiempo real" y "contexto de muchas consultas". Ejemplos del primero podrían ser: estimación de parámetros, control, simulador de vuelo... Son aplicaciones en las que los parámetros que entran en juego varían con frecuencia y es necesario disponer de la solución correspondiente con rapidez para poder evaluar su resultado y actuar en consecuencia. Ejemplos del segundo contexto podrían ser: optimización, simulación de múltiples modelos/escalas... Son aplicaciones en las que se desea explorar todas las zonas del espacio paramétrico.

Los Modelos de Orden Reducido (MOR) son la forma principal para afrontar esta clase de problemas. Ambos contextos son cruciales para la ingeniería computacional y para una adopción y aplicación más generalizada de métodos numéricos para EDPs en la práctica de ingeniería. Estos dos contextos son un desafío para los métodos numéricos clásicos (FEM, FVM) que requieren un alto coste computacional cuando la dimensión de la discretización crece excesivamente. Los métodos de Base Reducida (RB) no son competidores de los modelos clásicos de alto orden, sino que son complementarios y es obligatorio usar ambos en este tipo de contextos, para lograr construir modelos eficientes.

Las técnicas MOR reducen la dimensión del problema y permiten una resolución más rápida que las técnicas de alto orden (alta fidelidad) manteniendo sin embargo aún un cierto nivel de precisión. Para las EDP paramétricas, ser capaz de evaluar la solución del problema de baja complejidad, para cualquier nueva instancia, nuevo valor de los parámetros, a un coste que sea independiente de la dimensión del problema original de alta fidelidad, es la clave para el éxito computacional de cualquier MOR.

El trabajo presentado en esta tesis trata sobre varios aspectos del Modelado de Orden Reducido (MOR) utilizando la Descomposición Ortogonal Propia, mas conocida por su denominación en inglés *Proper Orthogonal Decomposition* (POD) como Base Reducida (RB). La construcción de la base reducida del POD es un proceso clásico de diagonalización de covarianza que utiliza un conjunto de soluciones del modelo de alta fidelidad (FEM, FVM...), *Snapshots*, para identificar los componentes principales y generar los modos ortogonales. Al igual que en la descomposición del valor singular (SVD, del inglés *Singular Value Decomposition*), basta con un pequeño número de modos para reproducir el modelo de alto orden con un nivel de aproximación aceptable. En POD, una combinación de *Snapshots* construye una base ortogonal de bajo orden donde se proyectará la Ecuación Diferencial Parcial que describe el problema. La proyección de POD clásica está en L^2,

sin embargo, el análisis de la literatura muestra que H^1 podría ser una alternativa más precisa tal y como se describe en el Capítulo 2.

El enfoque de descomposición computacional *Offline-Online* muestra un buen desempeño en ROM. La fase Offline es donde se calculan todas las soluciones a escala completa (*Snapshots*), y así se construye el MOR. Esta es la fase más larga y la más cara debido a la cobertura de muestreo del espacio paramétrico. Después, en la fase Online, se resuelve una nueva solución para una nueva ubicación no explorada del espacio paramétrico. La segunda etapa es mucho más rápida que la primera, ya que no estamos resolviendo el problema original, sino que tenemos una dimensión reducida, pero lo suficientemente precisa. Es la fase en la que obtenemos el rédito del esfuerzo invertido en la fase anterior. Es un equilibrio entre la exploración y la explotación del espacio de diseño.

En el MOR, un sistema mucho más pequeño reemplaza al algebraico grande, conservando las características esenciales del mapeo y garantizando un error por debajo de un umbral deseado. La precisión del POD-ROM se basa en tres cuestiones principales: (*i*) número de muestras de instantáneas (*Snapshots*), (*ii*) ubicación de los *Snapshots* en el espacio paramétrico y (*iii*) cantidad de modos ortogonales de la base reducida.

Las Estimaciones de Error *a posteriori* se han utilizado como criterio en los procesos de decisión de estas tres cuestiones en diferentes trabajos, como se describe en la revisión de la literatura del Capítulo 1. En esta Tesis, se presenta un nuevo enfoque para determinar la Estimación del Error. Se trata de un método basado en la técnica bien conocida de *Leave One Out Cross Validation* (LOOCV) que se ha modificado para no tener que recalcular la base reducida en el análisis de la sensibilidad (o relevancia) de cada *Snapshot*.

Según lo declarado por Quarteroni et al. en [74]: Los ingredientes esenciales de la metodología RB son: una proyección de Galerkin en un espacio tridimensional de funciones de base correctamente seleccionadas, una dependencia paramétrica afín que permite realizar una división *Offline-Online* competitiva en el procedimiento computacional, y una Estimación de error *a posteriori* rigurosa, utilizada tanto para la selección de la base como para la certificación de la solución.

La revisión del estado del arte realizada ha mostrado algunos aspectos interesantes del método POD en los que se puede profundizar y así contribuir al desarrollo de este tipo de herramientas MOR para la resolución de ecuaciones en derivadas parciales paramétricas. Además de una introducción teórica al método POD, en el capítulo 2, se presentan las alternativas de producto escalar L^2 y H^1 para realizar la proyección sobre la base reducida.

La calidad de la base reducida POD va a depender de la información que aporten los *Snapshots*, por lo que es de vital importancia decidir la cantidad de *Snapshots* que se requieren para alcanzar una mínima precisión en la solución aproximada y qué valores tomarán sus parámetros (localización en el espacio paramétrico). Se trata de recoger la mayor parte de la información disponible, utilizando el menor numero de *Snapshots* y barriendo el espacio parámetrico de una forma inteligente y eficiente. Esta es la fase "Offline", la mas costosa desde el punto de vista computacional. Este ha sido uno de los principales ámbitos en los que se ha desarrollado el capítulo 3: a partir de una distribución uniforme inicial de una serie de *Snapshots*, se ha propuesto un método que proporciona una búsqueda justificada, eficiente y fiable de la localización de nuevos *Snapshots* en el espacio paramétrico.

Como consecuencia de estos dos pasos: (i) distribución inicial y (ii) búsqueda de nuevas localizaciones, el POD aplicado a la resolución de EDP pertenece al conocido ámbito de los Algoritmos Voraces (*Greedy Algorithm*). La tesis recupera una conocida técnica de adaptividad de malla como método para la localización de los nuevos *Snapshots*. El Algoritmo Voraz es guiado por una Estimación del Error *a priori*. La revisión bibliográfica ha detectado varias estrategias para la definición de la Estimación del Error, y su obtención, generalmente, no es sencilla debido a que es dependiente del tipo de problema que se quiere resolver. En esta tesis se presenta una nueva estrategia para calcular la Estimación del Error que no es dependiente del problema que se quiere resolver. Para ello se parte de una técnica *Leave One Out Cross Validation* (LOOCV), que cuantifica la importancia que cada uno de los *Snapshots* tiene al generar el modelo reducido, algo así como la cantidad de información que aporta cada uno y asi poder determinar qué zonas del espacio paramétrico son mas relevantes y han de ser exploradas con mayor detalle. Para calcular la Estimación del Error de cada *Snapshot*, el método LOOCV, anula la información aportada por cada uno de ellos y calcula su solución sobre el modelo reducido utilizando el resto de *Snapshots*. El error calculado será respecto de su solución "verdadera" (el propio *Snapshot* anulado). Este proceso se realiza para cada uno de ellos. El resultado es conocido como "*a priori* Error Estimate", y es el valor que guía el Algoritmo Voraz.

La técnica LOOCV descrita puede ser costosa: para cada *Snapshot* analizado se debe crear un modelo reducido diferente, con una base creada a partir del resto de *Snapshots*. El trabajo que se presenta en esta tesis desarrolla, a partir del método LOOCV, un teorema nuevo para el cálculo de la Estimación de Error de cada *Snapshot*.

El **Teorema** dice: si ϕ_i, $1 \leq i \leq N$, son los *Snapshots* y si Ψ_i, $1 \leq i \leq N$, son los modos ortogonales de la base POD reducida, el enfoque LOOCV para la formulación POD no depende de la base POD. Es mas,

$$\|\phi_i - \sum_{j=1, j\neq i}^{N} (\phi_i, \Psi_j)\Psi_j\|^2 \leq \|\phi_i\|^2 - \frac{1}{trace(K)} \sum_{j=1, j\neq i}^{N} (\phi_i, \phi_j)^2$$

Este nuevo método solo depende de los *Snapshots* disponibles calculados en la fase *Offline* y no necesita recalcular el sistema POD para obtener la Estimación del Error de cada uno de ellos. Solo se necesita crear la base reducida una vez.

Se ha utilizado una localización inicial de los *Snapshots* uniformemente distribuida a lo largo del espacio paramétrico. Se puede considerar como una malla estructurada. Para la localización de los nuevos *Snapshots* se ha traído una técnica común de adaptividad de malla como Algoritmo Voraz, guiado por la Estimación del Error desarrollada. Se puede decir que la malla se va a adaptar y refinar allá donde el error calculado sea mayor, la zona del espacio paramétrico donde mas relevante sea la nueva información a aportar al modelo.

Es conocido el mal comportamiento que tiene el método POD para resolver de manera reducida problemas cuyas soluciones contienen choques. Se puede explicar este problema: el POD es como una interpolación "inteligente" de los resultados obtenidos por cualquier método numérico tradicional, por ejemplo, CFD. De esta manera, cuando se desea resolver el problema reducido para un nuevo valor de los parámetros, la solución obtenida no presenta el choque en su posición adecuada sino

que lo resuelve como una "media" de los valores disponibles. La tesis presenta una técnica para la solución POD de este tipo de problemas.

Para demostrar es falta de precisión del POD en este tipo de problemas que presentan choques en su solución, también en el capítulo 3, se ha desarrollado un ejemplo a partir de un problema de valor inicial convectivo puro. Aquí se demuestra cómo el POD en su formulación clásica no se comporta bien cuando se analiza la zona con choques y es necesario plantear un nuevo enfoque.

Primero, recordemos brevemente el cálculo de la solución aproximada en POD:

$$u(x,t) = \sum_{j=1}^{N} y_j(t)\Psi_j(x)$$

donde $u(x,t)$ es la solución buscada, $\Psi_j(x)$ son los modos ortogonales de la base reducida y los coeficientes $y_j(t)$ son la proyección de la solución sobre los modos, q tal y como se muestra, son dependientes de los parámetros.

El cálculo de estos coeficientes $y_j(t)$ requiere resolver un sistema dinámico. Puede ser cuadrático, cúbico... o algebraico, en los casos en los que la EDP no sea dependiente del tiempo. Habitualmente puede resultar un trabajo costoso. Cabe la posibilidad de estimar $y_j(t)$ como interpolación de los coeficientes obtenidos de la proyección de cada *Snapshot* sobre la base reducida. En el proceso propuesto en el capítulo 3, para problemas con choques en la solución, en lugar de interpolar, se propone buscar el *Snapshot* mas cercano, pero no es una cercanía obtenida en el espacio paramétrico, sino cercano en términos de los coeficientes $y_j(t)$. Se va a utilizar un proceso predictor-corrector. Para ello, en el paso predictor, primero se calculan los coeficientes $y_j(t)$ de la solución buscada mediante cualquier método de interpolación (cubic, spline...), y a continuación se localiza el *Snapshot* cuyos coeficientes sean los mas cercanos (corrector).

La validación de las contribuciones realizadas en el capítulo 3 se ha realizado en dos partes. Primero, se ha validado el método de localización de nuevos *Snapshots* sobre unas ecuaciones de un flujo incompresible y no viscoso con una solución conocida. La solución forzada en este caso no responde a una física real, se han creado unas oscilaciones importantes para los casos de una zona concreta del espacio paramétrico. Esa será la zona con los *Snapshots* mas relevantes, donde mas se debe explorar el espacio paramétrico en la fase Offline. A continuación, se ha validado también el nuevo enfoque para tratar los casos con choques sobre un caso CFD de un flujo en régimen permanente y no viscoso de un perfil aerodinámico simétrico tipo NACA0012. Los parámetros del modelo son el numero de Mach (M) y el ángulo de ataque (AoA). Se han validado casos subsónicos y transónicos. El código BBIPED (www.bcamath.org), basado en SU2, se ha utilizado como CFD para crear los *Snapshots*. El dominio espacial 2D está discretizado por una malla circular con 3762 elementos triangulares. Un código de MATLAB programado desde cero se ha acoplado con BBIPED para ejecutar el proceso completo:

1. Comienza con una distribución inicial de *Snapshots*uniforme.

2. Resuelve esos *Snapshots* utilizando la plataforma BBIPED CFD.

3. Construye una base reducida (RB) de POD y un modelo reducido (POD-ROM inicial).

4. Establece la estimación de error segun el nuevo enfoque propuesto.

5. Ejecuta un proceso de adaptabilidad de malla conducido por esa Estimación de Error, obteniendo las nuevas ubicaciones mejoradas de los *Snapshots*.

6. Resuelve nuevos *Snapshots* mejorados (utilizando BBIPED).

7. Construye un nuevo POD-ROM enriquecido.

8. Resuelve ambos enfoques POD-ROM: inicial y enriquecido, para nuevas ubicaciones del espacio paramétrico no exploradas.

9. Ambos enfoques POD-ROM son comparados.

La base reducida de POD (POD-RB) es óptima en promedio (Teorema de Mercer). Es decir, para una dimensión determinada de la base reducida, el procedimiento POD obtiene la mejor representación de los *Snapshots*, en promedio. En el capítulo 4 se muestra cómo es posible encontrar una mejora local de esa RB. Existen otras bases reducidas que mejoran el resultado del modelo reducido POD para ciertos valores de los parámetros. La tesis muestra un nuevo enfoque denominado algoritmo "Gramm Schmidt clasificado" (SGS) para la obtención de otra base reducida. Finalmente, se define un criterio para discriminar entre ambas bases, POD o SGS, que determina a la hora de explotar el modelo reducido, cual de las dos se debe utilizar.

La validación del acoplamiento de ambas estrategias ROM, SGS y POD, se ha realizado también sobre unas ecuaciones de flujo con solución conocida. Ademas, se ha evaluado el criterio de selección de base reducida local entre SGS y POD.

En Resumen. El trabajo presentado contribuye al método ROM POD en algunos de sus aspectos. El POD consta de dos pasos transcendentales: la distribución inicial de *Snapshots* y la localización de los nuevos que vayan a enriquecer el modelo reducido. La busqueda de los nuevos *Snapshots* se basa en un Algoritmo Voraz basado en una Estimación del Error *a priori*. La técnica LOOCV puede obtener ese Error. Se ha desarrollado un teorema que demuestra la universalidad de ese Error, desde el punto de vista que no es dependiente del problema a resolver, y que no hace falta recalcular la base reducida cada vez que se desea obtener el Error de cada uno de los *Snapshots*. Para el proceso que determina las nuevas localizaciones se ha traído al espacio paramétrico un método de refinamiento de mallado espacial (*mesh adaptivity*) comunmente utilizado en métodos de alto orden (FEM, FVM...). En el ejemplo mostrado, el proceso se ha iniciado con una distribución uniforme de los *Snapshots* (como una malla estructurada) para, después, hallar una nueva distribución de una manera mas inteligente y eficiente. Por otro lado, se ha presentado el calculo de los coeficientes POD $y_j(t)$ (dependientes de los parámetros) usando métodos de interpolación. Este enfoque no da buenos resultados para los casos en los que la solución presenta choques. Se ha desarrollado una estrategia predictor-corrector de estos coeficientes y se ha demostrado en un flujo transónico en torno a un perfil aerodinámico. Finalmente, la base POD es optima en promedio, pero se ha demostrado que existen otras bases reducidas que para ciertas zonas del espacio paramétrico mejoran su comportamiento. Se ha desarrollado un algortimo para crear una nueva base, denominado SGS. Ambas bases trabajan acopladas, y es necesario elegir una de las dos a la hora de explotar el ROM para nuevos valores de los parámetros. Se ha creado un criterio para la toma de esta decisión.

Chapter 1

Introduction

Solving Partial Differential Equations (PDEs), is a key issue in science and engineering since they are widely used in many real problem modelling including fluid mechanics, acoustics, heat and mass transfer ...etc. Exact solutions for PDEs can be obtained in a very few and simplified cases only, numerical approximations are used instead. Several numerical techniques have been developed and deeply analysed in literature starting in the early beginning of the 20th century with Richardson(1910); Phillips and Wiener(1923); Courant,Friedrichsand Lewy(1928). The numerical simulation interest growth raised exponentially up in the last decade with the advent of supercomputers and parallel programming techniques. This allowed using high dimension methods (Full order scale techniques), like Finite Element Method (FEM), Finite Volume Method (FVM), discontinuous Galerkin (DG), Spectral Methods and so on. However and despite the availability of important supercomputing facilities, due to the huge number of degrees of freedom these methods for some problems still suffering from cost-effectiveness performance. They are mainly two contexts where one need to solve PDEs with a low computational cost: Real-time context and Many-query context [74, 43, 78]. Examples for the first case include parameter-estimation, control, Flying simulators ... etc. And for the second case applications include optimization, multi model/scale simulation among others.

To reduce significantly simulation time (often on the expense of accuracy) Reduced Order Modelling (ROM) techniques are introduced. The main idea is to reduce the initial solving space dimension (as in Finite Elements Method) to a subspace with a significantly reduced dimension and then solve for the projected solutions. The dimension reduction will result naturally in a significant computing cost reduction.

Many techniques were proposed to achieve ROMs, the most popular being Reduce Basis (RB) [9, 46, 67, 64], and Proper Orthogonal Decomposition (POD) [53, 1, 14, 93, 54] techniques, specially in the context of PDEs. Note that both methods are projection methods: a subspace of a reduced dimension is first built on which the unknown solution is then projected.

A considerable amount of work can be found in literature concerning different aspect of ROM technique, one can refer to papers [75, 4, 78, 17, 11, 13, 91]. more detailed can be found in books [73, 81, 40] and PhD Thesis [20, 5, 6].

POD [56, 42, 83, 69, 88, 50, 49, 75, 24, 19, 18] as a particular strategy to build a ROM, which is the scope of this thesis, is a projection Galerkin method where a reduced basis system is built such that the spanned subspace captures the maximum

information (physics in case of PDE-based physical model) of the parametric solution (parameters could be time, design parameters, boundary conditions, and so on). Then an unknown solution for any given set of parameters is projected onto the obtained basis and substituted in the original equation to solve for the projection coefficients as in Galerkin approach.

As a first consequence using the Galekin approach in POD will result as for Finite Element Method in a stability issue. Many solutions are proposed to remedy to this shortcoming, we can cite the generalization of the Stream-line Upwind Petrov-Galerkin method refereed to as SUPG-POD see [90, 31], the regularization approach [80] and the stabilization by dissipation [21, 25, 7, 47].

Three other aspects, that are the main scope of this thesis, are crucial for an effective POD system and received a big interest in literature.

Error Bounds and Snapshots Locations (Sampling)

The basis system is built from a set of off-line computed PDE parametric solutions called snapshots. Therefore, the efficiency of the basis system depends clearly on the choice of the snapshots. in another word, the choice of the initial parameters distribution (snapshots locations) is crucial and will define the performance of the POD to solve the PDE problem. The most popular approach is to derive an error bound then introduce the snapshots one by one in order to reduce the error following the spirit of the Greedy algorithm reviewed later.

A priori error estimate called in general error bound [70, 87, 36, 78, 32] is then the starting point for an efficient POD system. The error bound must be valid for any basis size and for all parameter values in the parameter domain, and efficient in the sense that the error tends to zero as the number of snapshots is increased. This is problem dependent and very hard to obtain. Other alternative are proposed for a universal error bounds estimate, like LOOCV (Leave One Out Cross Validation) technique, that will be the basis of the proposed error estimate in this work.

Note that the POD approach is proven to be optimal (in average), in this work and based on error estimates, a strategy to locally couple POD with an appropriate ROM is proposed to improve the classical POD outputs

The choice of parameter sample points is a critical and quite general concern not only restricted to POD. For instance in the rational interpolation methods [12] , one must select parameter samples at which interpolation conditions are applied; for balanced truncation [59], one must select parameter samples to generate the local LTI (Linear Time Invariant) systems at which balanced truncation is applied; and in the POD [83], one must select parameter samples at which snapshots are computed.

For problems with a small number of parameters, a structured or random sampling method is a reasonable approach and, with a sufficiently high number of samples, will generate a rich set of data that covers the parameter space. For a moderate number of parameters, full grid sampling quickly becomes expensive, since the number of points in the grid grows exponentially with the dimension d. Then, it becomes challenging to balance sampling cost with coverage of the parameter space. In another word, and in the case of POD, to balance the dimension of the POD subspace with the maximum information capturing. This case a more sophisticated sampling approaches are required, such as a problem-aware adaptive search of the parameter space (see [13]). For POD many approaches for snapshots locations are

proposed based on error estimate and following the Greedy algorithm spirit. In the same idea we propose in this work the use of techniques for local mesh adaptation widely used in Computational Fluid Dynamics (CFD) [33, 27, 39, 77, 76, 66, 85, 8, 48] driven by the proposed error estimate to enrich and initially and uniformly snapshots distribution.

Projection coefficients estimation

Another technical challenge is to estimate the projected POD coefficients. The natural way to proceed is to substitute the projected solution into the original PDE and then, solve (numerically) the resulting dynamical system (for time dependent PDE) or algebraic system. However this is not always easy, specially for nonlinear PDEs, where the resulting system can be too expensive, reducing the attractive cost-effectiveness of POD. As an alternative an interpolation techniques are proposed and many works have been done on the subject [18, 10, 73, 72, 23, 73].

Greedy Algorithm

The greedy algorithm [15] philosophy, is to make local improvement hopping that this will lead to a globally-optimal solution. In case of availability of a posteriori error estimators, the Greedy procedure can be applied [87, 36, 64], and accumulatively determines snapshots based on a (typically large) set of training parameters [38]

The main step consists of repeatedly searching for the currently worst resolved parameter (as indicated in the error estimator), and computing and including solution snapshots for this parameter into the basis. This process can loop and extend the basis in each step until certain a priori error estimate threshold is reached.

Another possible method of selecting the parameter values for sampling, proposed in [16], is to use local sensitivity analysis to estimate whether a change in parameter value will result in states that are not well represented in the current reduced basis. That work uses a first-order Taylor series expansion to approximate the state solution as a function of small changes in parameters about the current sample point. The sensitivities of the state variables with respect to the parameters are then obtained by solving a large sparse linear system.

1.1 Thesis Contributions

The main contributions of the thesis are summarized as follows:

1. A Leave-One-Out Cross-Validation method (LOOCV) technique is used to estimate a universal upper bound error, this error is estimated empirically by re-building the POD system for each disregarded snapshot. This process is prohibitive for large systems (when a big number of snapshot is needed). A theorem is proven providing a formula to estimate the LOOCV error avoiding recalculating the POD system. The theorem states: if $\phi_i 1 \leq i \leq N$ are the snapshot (solutions of a given PDE for different parameters) and if $\Psi i 1 \leq i \leq N$ are the POD basis elements, then

Theorem. *The LOOCV approach for POD formulation don't depend on the POD basis. Moreover*

$$LOOCV - Error = \|\phi_i - \sum_{j=1, j\neq i}^{N} (\phi_i, \Psi_j)\Psi_j\|^2 \leq \|\phi_i\|^2 - \frac{1}{trace(K)} \sum_{j=1, j\neq i}^{N} (\phi_i, \phi_j)^2 \tag{1.1}$$

Note that this formula does not depend on the POD basis nor on the correlation matrix which means no need to build the basis system to estimate the error for each snapshot location.

2. A mesh adaptivity inspired approach is proposed to drive the insertion of a new shots location process for the Greedy algorithm. The error defined in 1. can be used as is or its Hessian. For the performed tests no significant difference is observed whether the error is used or its Hessian.

3. The POD is performing badly for shocked solutions as mentioned by many authors. This can be explained by the fact that the POD solution is a 'smart' averaging of solutions obtained by any traditional CFD solver, therefore averaging shocks will results necessary in creating oscillations and wrong position of the real shock. This is first proved using a simple convection equation. To remedy this problem, a new interpolation (for shocked solutions) approach is proposed by considering the nearest shot based on the POD coefficient not on the parameters. In another word, a POD solution is first calculated using any standard interpolation method, and then the final coefficients of the POD solution are compared to the snapshot coefficients, to find the closest one.

4. A validation of the proposed POD system against an exact solution of the following incompressible inviscid flow equations

$$\begin{cases} v\left(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}\right) = f_1 \\ u\left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}\right) = f_2 \end{cases}$$

where

$$\begin{aligned} f_1 &= \cos^2(\alpha(x^2+y^2))(x+y)(4\alpha\sin(\alpha(x^2+y^2))\cos(\alpha(x^2+y^2)) \\ f_2 &= \sin^2(\alpha(x^2+y^2))(x+y)(-4\alpha\sin(\alpha(x^2+y^2))\cos(\alpha(x^2+y^2)) \end{aligned}$$

is achieved.

5. A validation against a CFD solution of an inviscid flow around a NACA0012 is achieved as well demonstrating the efficiency of the proposed approach by comparison to the numerical result obtained by SU2 solver (a finite volume solver developed by Stanford University)

6. The contribution presented in chapter 4 is motivated by the optimality theorem for POD method that states

Theorem. *Let $V_M = Span\{\Psi_1, ..., \Psi_M\} \subset H = L^2(\Omega)$, then for a subspace $Q_M = Span\{W_1, ..., W_M\} \subset H$ we have:*

$$\int_\Lambda d_H(u(.,.,\gamma), V_M)\mathrm{d}\gamma \leq \int_\Lambda d_H(u(.,.,\gamma), Q_M)\mathrm{d}\gamma \tag{1.2}$$

where $d_H(u(.,.,\gamma), V_M)$ is a distance from $u(.,.,\gamma)$ to the subspace V_M

As we can see from the theorem, POD optimality is guaranteed in the sense of average over the parameter space. This optimality is not satisfied pointwise, as we demonstrated with through some tests. So, this open a room for coupling the POD with another suitable ROM in an appropriate way to improve the results (locally) without contradicting the optimality theorem.

7. As a consequence, it is proposed in this chapter 4 a new ROM based on a sorted Gram Schmidt (SGS) algorithm. Then the POD and SGS approaches are coupled to improve the final result. First an error distribution is calculated over the parameter space for both POD and SGS. Note that in this chapter a projection-error is proposed since the proved theorem (chapter 3) for LOOCV based error is not valid for the SGS. The projection-error is defined as the projection of each shot on the space spanned by the POD and SGS basis respectively. To couple the two methods, when estimating a new solution (for a new parameters value), and to decide which method to use (POD or SGS) the error is interpolated for the current parameters using the error distributions defined above and then the method with smallest error is selected.

8. The efficiency of the coupled POD-SGS is demonstrated for the exact solution of the following uncompressible inviscid flow equations

$$\begin{cases} v\left(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}\right) = f_1 \\ u\left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}\right) = f_2 \end{cases}$$

with source terms

$$f_1 = (x-y)\quad \frac{2\beta^2}{\left((x-a)^2+(y-a)^2-\beta^2\right)^2}\quad e^{\frac{2\beta^2}{(x-a)^2+(y-a)^2-\beta^2}}$$

$$f_2 = (y-x)\quad \frac{2\beta^2}{\left((x-a)^2+(y-a)^2-\beta^2\right)^2}\quad e^{\frac{2\beta^2}{(x-a)^2+(y-a)^2-\beta^2}}$$

It is demonstrated that results are improved when POD-SGS is used comparing to when POD or SGS is used separately

1.2 Thesis Outline

The thesis is organized as follow: In Chapter 2, the POD method is introduced in the general framework and in the context of PDEs. In Chapter 3, two contributions of the thesis are developed. First, based on LOOCV technique, a new a priori error estimate formula is proposed through a proved theorem. Then snapshot locations technique using a well known mesh adaptivity strategy is proposed. Then, a new interpolation approach for POD projection coefficients is presented, in order to remedies the lack of accuracy of the POD method when dealing with shocked solution. The chapter ends with a validation section against exact (mathematical) solutions of incompressible inviscid flow equations and against CFD solutions of inviscid flow around a NACA0012. Chapter 4 proposes a strategy to couple POD with appropriate ROM (SGS). In chapter 5, conclusions are drawn.

Chapter 2

Proper Orthogonal Decomposition (POD)

Solving large-scale problems is one of the biggest challenge in computational science in general and in PDEs (partial differential equations) based problems in particular. Despite the huge progress in computers capability and parallel architectures, solving cost-effectively the original problems still unaffordable in almost complex cases. One of the most attracting methods to handle and overcome such limitations are the so-called reduced-order models (ROMs) that reduce the dimension of the original problem such that the obtained reduced model is solved with much less complexity and in a reasonable processing time. Almost of ROMs models are snapshots based and the most popular one is undoubtedly the Proper Orthogonal Decomposition (POD) [53, 1, 14, 93], know as well as Karhunen-Loeve's expansion (KLE) [54]. The ROMs snapshots based received a lot of attention in the literature due to their success in solving many problems in a wide range of fields. The principle of the technique consists in representing a parametric family of functions, that could be solutions of PDEs, belonging to a given functional space, by their projections on a subspace of reduced dimension spanned by a reduced number of orthogonal basis functions built from the snapshots and referred to, in some areas, as modes. This reduces for instance solving Navier-Stokes equations by finite elements that requires a subspace spanned by a basis of size of order of millions to solve a small dynamical system of size smaller than hundred [55, 82, 41, 26].

In the following, a detailed overview of the the proper orthogonal decomposition (POD) technique, the purpose of this thesis, is provided in either the general framework and the particular context of PDEs.

2.1 POD: General framework

2.1.1 General idea

In the general context of data analysis, POD is a technique that aims to represent a huge amount of data by a reduced number of basis element built from the data. In another word if the data are represented in a $m \times n$ matrix A with n being a large number, it is possible to represent the main information stored in the n columns A by a p orthonormal vectors with p very small comparing to n.

2.1.2 Mathematical formulation

If $A = [V_1, V_2, ..., V_n]$, the problem is to find $B = [W_1, W_2, ..., W_p]$ an orthonormal set of p vectors that better represent the n columns A in the following sense,

W_1 satisfying the optimization problem,

$$W_1 = Arg\left[Max_{W \in IR^n}\left(\sum_{i=1}^{n} |\langle W_1, V_i\rangle|^2\right) \quad subject\ to \quad \|W_1\|^2 = 1\right] \tag{2.1}$$

This constrained optimization problem can be solved using Lagrange multipliers:

$$L(W, \lambda) = \sum_{i=1}^{n} |\langle W, V_i\rangle|^2 + \lambda(1 - \|W\|^2) \quad (W, \lambda) \in IR^n \tag{2.2}$$

Then find W_2 satisfying

$$W_2 = Arg\left[Max_{W \in IR^n}\left(\sum_{i=1}^{n} |\langle W_2, V_i\rangle|^2\right) \quad subject\ to \quad \|W_2\|^2 = 1 \ \ and \ \ \langle W_2, W_1\rangle = 0\right] \tag{2.3}$$

And so on...

By Nullifying the gradient of the Lagrange multipliers $L(W, \lambda_1, ..., \lambda_p)$, namely $\nabla L(W, \lambda) = 0$ we can show (see [88] for details) that $\{W_1, W_2, ..., W_p\}$ are the eigenvectors of the positive semi-definite matrix AA^T and the $\lambda' s$ are the associated eigenvalues. Moreover this establishes a clear connection of the POD approach to the SVD (singular values decomposition) technique [34, 52]. A deeper description of the relation between POD and SVD based on the approaches of Nobach in [62] and Kutz in [52] is available at Appendix B.

2.2 POD: PDEs context

Proper Orthogonal Decomposition for PDEs is based on a Galerkin projection of the governing equations onto the subspace spanned by POD basis functions (also known as modes) yielding a simple set of ordinary differential equations (ODEs).

Let

$$\frac{\partial u}{\partial t} + \mathcal{L}(u) = f \tag{2.4}$$

represent an evolutionary PDE, where $\mathcal{L}$ being any space differential operator and f is a source term.

Let Λ be a space of parameters (design parameters, flying conditions, etc...) on which the solution depends. For the sake of simplicity, let's consider a one dimensional case. So, $u = u(t, x, \gamma) \quad \gamma \in \Lambda$. Let $\phi_i, 1 \leq i \leq N$, be a set of solutions (snapshots) of equation (2.4), where the index i refers to different values of the parameter γ. The goal of the POD method is to reconstruct the solution of (2.4) for any parameter value (within a certain range), without solving the original problem (2.4). To achieve this objective a discrete vector space with a low dimension (as low as possible) is built from the snapshots, then the solution of (2.4) is projected onto this subspace and as a result, rather than solving (2.4), an approximated POD solution is obtained by solving a simple Ordinary Differential Equation (ODE) system.

2.2.1 POD basis system

To build this POD basis system we proceed as follows:

Let K be the correlation matrix given by $K_{i,j} = (\phi_i, \phi_j)$ where $(,)$ is any inner product (L^2, H^1, or any other depending on the functional space to wich belongs the solution u). Then the eigenvalues and associated eigenvectors are calculated. Let V a matrix which columns are eigenvecors of K, then the basis functions Ψ_i, called in this case *modes*, are given by:

$$\Psi_i = \sum_{i=1}^{N} v_i^j \, \phi_i \tag{2.5}$$

where v_i^j are the components of the j^{th} eigenvector of K

Let's now calculate the inner product (Ψ_i, Ψ_j)

$$(\Psi_i, \Psi_j) = (\sum_{l=1}^{N} v_l^i \phi_l, \sum_{k=1}^{N} v_k^j, \phi_k) = \sum_{l,k} v_l^j v_k^j (\phi_l, \phi_k) = \sum_{l,k} v_l^j v_k^j K_k^l = KV^i.V^j = \lambda_i V^i.V^j \tag{2.6}$$

This shows that the functions Ψ_i, $1 \leq i \leq N$, are orthogonal if and only if the eigenvectors V^i, $1 \leq i \leq N$ are orthogonal, which is general the case. Now if $i = j$ we get the norm $\|\Psi_i\| = \lambda_i$. This is coherent, since the matrix K is positive semi-definite which imply that its eigenvalues are all non-zero. The norm of ϕ_i represents part of the energy of the system. We can conclude that the modes with low energy (low corresponding eigenvalues) can be neglected reducing the POD basis to only modes with significant energy.

2.2.2 From PDE to reduced dimension Dynamical system

Now multiply both sides of equation (2.4) by Ψ_i

$$(\frac{\partial u}{\partial t}, \Psi_i) + (\mathcal{L}(u), \Psi_i) = (f, \Psi_i) \tag{2.7}$$

Then extend the solution of problem (2.4) on the POD basis

$$u(x,t) = \sum_{j=1}^{N} y_j(t) \Psi_j(x) \tag{2.8}$$

And by substituting this expression in (2.4) and using the fact that Ψ_j are orthonormal (we can always normalize the modes), we get

$$(\frac{\partial \sum_{j=1}^{N} Y_j(t) \Psi_j(x)}{\partial t}, \Psi_i) + (\mathcal{L}(\sum_{j=1}^{N} Y_j(t) \Psi_j(x)), \Psi_i) = (f, \Psi_i) \tag{2.9}$$

Then

$$\frac{\partial Y_i(t)}{\partial t} + (\mathcal{L}(\sum_{j=1}^{N} Y_j(t) \Psi_j(x)), \Psi_i) = (f, \Psi_i) \tag{2.10}$$

Now if $\mathcal{L}$ is a linear operator we get

$$\frac{\partial Y_i(t)}{\partial t} + \sum_{j=1}^{N} Y_j(t)(\mathcal{L}(\Psi_j(x)), \Psi_i) = (f, \Psi_i) \tag{2.11}$$

Which is a linear dynamical system in $Y_j(t)'s$ of the form

$$Y' = A + (BY) \qquad 0 \leq i \leq N \tag{2.12}$$

Here the vector A contains the source term and the boundary conditions and B contain the terms of the form $(\mathcal{L}(\Psi_j(x)), \Psi_i)$

If $\mathcal{L}$ is a quadratic operator we obtain a quadratic dynamical system in $Y_j(t)$.

$$Y' = A + (BY) + (CY, Y) = 0 \qquad 0 \leq i \leq N \tag{2.13}$$

Here C is a $N \times N \times N$ tensor.

More details for builded dynamical system in this thesis developped by POD projection of Euler equations onto an orthonormal basis can be found in appendix A.3.

2.2.3 Choice of the inner product

The choice of the inner product $(,)$ when POD is applied to solve PDEs is important, it is pointed out in literature [44, 45, 62] add references and confirmed by our tests, that we get better solutions using the H^1 inner product defined by

$$(u, v)_{H^1} = \int_\Omega uv\mathrm{d}x + \int_\Omega (\nabla u \cdot \nabla v)\mathrm{d}x \tag{2.14}$$

Than using L^2 inner product

$$(u, v)_{L^2} = \int_\Omega (u_1 v_1^T + u_2 v_2^T + u_3 v_3^T)\mathrm{d}x; \tag{2.15}$$

This is due, among others, to the contribution of the derivatives that brings information about the solution regularity. However this is on the expense of the cost-effectiveness.

Chapter 3

POD: Effective Snapshots Location and Interpolation Strategy

Proper Orthogonal Decomposition (POD), described in Chapter 2 gained a large interest in the contest of solving PDEs thanks to its strong mathematical background and practical implementation. The key for an efficient POD design (as is the case for all ROMs methods) is the sampling strategy consisting in an optimal snapshots (PDEs solutions) selection that capture a maximum information and allow building a POD subspace on which the projected PDEs solutions are representative of the actual solution to a desired accuracy. Several sampling strategies have been proposed and used in the literature [37, 18, 20, 51, 2, 61, 79]. The main trend is to set an initial Snapshots locations distribution (computed at an Offline stage), then increase the number of Snapshots driven by an *a posteriori error estimator*. This strategy falls into the well known Greedy Algorithm general framework [15, 36, 87, 86, 64, 63]. Note that this error estimator is in general problem dependent and not easy to obtain.

In this chapter we propose to first derive an error estimator based on the Leave-One-Out Cross-Validation strategy (LOOCV) (described later), and then use the well known local mesh adaptivity technique to drive the process of inserting and improving snapshots locations. Note that the LOOCV based error estimate combined to machine learning is used successfully in the context of POD in [92]. However and as reported by the authors, this process is prohibitive for large snapshots samples since it requires re-building the POD basis for each removed snapshot. To overcome this shortcoming, we proved a theorem that provides a formula of the LOOCV that is independent of the POD basis, as a consequence no need to re-build the basis for each disregarded snapshot. The advantage of using LOOCV based error, is it universality (it is not a problem dependent) and the proposed theorem reducing drastically its cost-effectiveness makes it even more attractive.

3.1 Shots Location Algorithm

In this section the proposed algorithm for an optimal shots locations is described. This is based on two ingredients, the technique of local mesh adaptivity, and LOOCV based error estimate. First a short overview of local mesh adaptivity technique is

given, then the LOOCV based error is described and a theorem providing a POD-basis independent LOOCV formula is proved.

3.1.1 Local mesh adaptivity technique

In numerical simulation of PDEs-based physical phenomena, it is well known that the accuracy of any numerical method used for such a purpose depends strongly on the mesh quality. To improve the final solution, one of the mostly used technique is the mesh adaptivity which received a constant interest growth in literature see for instance [33, 27, 39, 77, 76, 66, 85, 8, 48]. The most popular approach of mesh adaptivity is the so called Hessian-based local mesh adaptation technique. The adaptivity process is driven by an *a posteriori* error estimate based on the Hessian of a selected scalar of the field variables solution. When the Hessian H is positive definite it defines a metric T, therefore the error along a given vector could be interpreted as the length of the vector measured by the Hessian metric. Consequently, and to ensure positivity, the absolute value of the the Hessian is considered. Resulting in a metric $T = |H| = R^t|\Lambda|R$, with $|R|$ being the eigenvectors matrix of $|H|$ and $\Lambda = diag\{\lambda_1, \lambda_2, \lambda_3, ...\}$ the matrix composed by the absolute values of the eigenvalues of $|H|$. The eigenvectors provide the local direction of the stretching, while the eigenvalues give its local magnitude, thus gradually creating anisotropy.

The goal of the adaptation is to equi-distribute the error on the adapted grid, where the error along an edge in the *Riemannian* metric is computed as:

$$\varepsilon = \int_0^1 \sqrt{\overrightarrow{x}^t(s)T(s)\overrightarrow{x}(s)} \tag{3.1}$$

where $\overrightarrow{x}$ is the vector that defines the edge.

The algorithm starts with a uniform mesh, and then a first solution is estimated by the solver and used to estimate the error metric. The mesh adaptation operations include node movement, edge refinement, coarsening and swapping. The error on the adapted mesh is interpolated on the original one considered as a background mesh.

The sequence of operations begins with node movement, edge refinement and edge swapping on solid boundaries, to satisfy a minimum and maximum edge length constraint, as well as a curvature constraint, yielding substantial surface CAD improvements. The process then continues with node movement in the entire domain, followed by refinement and coarsening, then swapping, before concluding with additional node movement. The method preserves CAD integrity by re-projecting boundary points onto the original surfaces during the adaptation process.

This technique has been widely tested in CFD applications and its efficiency demonstrated over a considerable set of test cases. In this work the technique is adapted to optimize snapshots locations. The space of parameters is considered as a mesh and referred to it by a P-mesh. To drive the whole process and error based on the LOOCV technique is derived and described in the following subsection.

3.1.2 LOOCV based error estimate

Introduced by [57], and well known in statistics, the method computes each snapshot error estimate as follows.

Let ϕ_i be a given snapshot of the set $S = \{\phi_1, ..., \phi_N\}$, consider the new set $\overline{S}_i = S \backslash \{\phi_i\}$, by taking out the shot ϕ_i. Then generate the POD system $B = \{\Psi_k(x)\}_{k=1}^{N-1}$ from the snapshots of $\overline{S}$. The error $Error(i)$ is then calculated as the L^2 norm of the difference between ϕ_i and its projection $\overline{\phi}_i$ onto B namely

$$Error(i) = \frac{\sqrt{\int_\Omega (\phi_i(x,y) - \overline{\phi}_i(x,y))^2 dxdy}}{\sqrt{\int_\Omega (\phi_i(x,y))^2 dxdy}} \tag{3.2}$$

With

$$\overline{\phi}_i = \sum_{k=1}^{N-1} (\phi_k, \Psi_k)\Psi_k \tag{3.3}$$

This formula could be used as is and then the error length ε_i of an edge $e_{i,j}$ connecting two nodes in the parameter mesh (P-mesh) is given simply by

$$\varepsilon_i = \frac{Error(i) + Error(j)}{2} e_{i,j} \tag{3.4}$$

or we can look at the curvature of the error by taking its Hessian and then use formula (3.1) and proceed as in subsection 3.1.1.

Now let's prove a theorem that provides a LOOCV-based error formula independent of the POD basis independent

Theorem. *The LOOCV approach for POD formulation don't depend on the POD basis. Moreover*

$$LOOCV - Error = \|\phi_i - \sum_{j=1, j\neq i}^{N} (\phi_i, \Psi_j)\Psi_j\|^2 \leq \|\phi_i\|^2 - \frac{1}{trace(K)} \sum_{j=1, j\neq i}^{N} (\phi_i, \phi_j)^2 \tag{3.5}$$

Proof Let's estimate the quantity

$$\|\phi_i - \sum_{j=1}^{N} (\phi_i, \Psi_j)\Psi_j\|^2 \tag{3.6}$$

$$\begin{aligned} &\|\phi_i - \sum_{j=1}^{N} (\phi_i, \Psi_j)\Psi_j\|^2 = \\ &= \Big(\phi_i - \sum_{j=1}^{N} (\phi_i, \Psi_j)\Psi_j \ , \ \phi_i - \sum_{j=1}^{N} (\phi_i, \Psi_j)\Psi_j\Big) = \\ &= \|\phi_i\|^2 + \|\sum_{j=1}^{N} (\phi_i, \Psi_j)\Psi_j\|^2 - 2\Big(\phi_i, \sum_{j=1}^{N} (\phi_i, \Psi_j)\Psi_j\Big) \end{aligned} \tag{3.7}$$

let's develop second and third terms from equation (3.7).
The second term:

$$\|\sum_{j=1}^{N}(\phi_i,\Psi_j)\Psi_j\|^2 = \Big(\sum_{j=1}^{N}(\phi_i,\Psi_j)\Psi_j \ , \ \sum_{j=1}^{N}(\phi_i,\Psi_j)\Psi_j\Big) =$$

$$= \sum_{j,k=1}^{N}\Big((\phi_i,\Psi_j)\Psi_j \ , \ (\phi_i,\Psi_k)\Psi_k\Big) = \sum_{j,k=1}^{N}(\phi_i,\Psi_j)(\phi_i,\Psi_k)(\Psi_j,\Psi_k) =$$

$$= \sum_{j=1}^{N}(\phi_i,\Psi_j)^2$$

since for every $j \neq k$, $(\Psi_j,\Psi_k) = 0$, by orthogonality.
and for $j = k$, $(\Psi_j,\Psi_k) = 1$, we assume that the modes are normalized ($\Psi_j = \frac{\Psi_j}{\|\Psi_j\|} = \frac{\Psi_j}{\lambda_j}$

Now, we develop the third term:

$$2\Big(\phi_i, \sum_{j=1}^{N}(\phi_i,\Psi_j)\Psi_j\Big) =$$

$$2\sum_{j=1}^{N}(\phi_i,\Psi_j)(\phi_i,\Psi_j) = 2\sum_{j=1}^{N}(\phi_i,\Psi_j)^2$$

substituting in (3.7) we get

$$\|\phi_i - \sum_{j=1}^{N}(\phi_i,\Psi_j)\Psi_j\|^2 = \|\phi_i\|^2 - \sum_{j=1}^{N}(\phi_i,\Psi_j)^2 \tag{3.8}$$

Now let K^i be the correlation matrix given by $K_{k,j} = (\phi_k,\phi_j)$ obtained with the snapshots excluding ϕ_i, and V^i the matrix of corresponding eigenvectors. We know that

$$\Psi_j = \frac{1}{\sqrt{\lambda_j}}\sum_{\substack{k=1\\k\neq i}}^{N} v_k^{i,j}\,\phi_k \tag{3.9}$$

where $v_k^{i,j}$ are the k components of the j^{th} eigenvector of K^i.
Set

$$\Phi_i = \begin{pmatrix}(\phi_i,\phi_1)\\ \vdots \\ (\phi_i,\phi_k)\\ \vdots \\ (\phi_i,\phi_N)\end{pmatrix} \quad \text{for } k \neq i \text{ and note that } \mathit{trace}(K^i) \leq \mathit{trace}(K)$$

Now, expend the second term of (3.8)

$$\sum_{j=1,j\neq i}^{N}(\phi_i,\Psi_j)^2=\sum_{j=1,j\neq i}^{N}(\phi_i,\sum_{\substack{k=1\\k\neq i}}^{N}\frac{1}{\sqrt{\lambda_j}}v_k^{i,j}\,\phi_k)^2=\sum_{j=1,j\neq i}^{N}\frac{1}{\sqrt{\lambda_j}^2}(\Big(\sum_{\substack{k=1\\k\neq i}}^{N}v_k^j(\phi_i,\phi_k)\Big)^2=$$

$$\geq\frac{1}{trace(K_i)}\sum_{j=1,j\neq i}^{N}\Big((V^i)^T,\Phi_i\Big)^2=$$

$$\geq\frac{1}{trace(K)}\sum_{j=1,j\neq i}^{N}\Big((V^i)^T,\Phi_i\Big)^2=$$

$$\frac{1}{trace(K)}\sum_{j=1,j\neq i}^{N}\Big((V^i)^T\Phi_i,(V^i)^T\Phi_i\Big)=$$

$$\frac{1}{trace(K)}\sum_{j=1,j\neq i}^{N}\Big((V^i)(V^i)^T\Phi_i,\Phi_i\Big)=$$

$$\frac{1}{trace(K)}\sum_{j=1,j\neq i}^{N}\Big(\Phi_i,\Phi_i\Big)=$$

$$\frac{1}{trace(K)}\sum_{j=1,j\neq i}^{N}(\phi_i,\phi_j)^2 \tag{3.10}$$

Substituting back in (3.8) :

$$\|\phi_i-\sum_{j=1,j\neq i}^{N}(\phi_i,\Psi_j)\Psi_j\|^2\leq\|\phi_i\|^2-\frac{1}{trace(K)}\sum_{j=1,j\neq i}^{N}(\phi_i,\phi_j)^2 \tag{3.11}$$

Which achieves the proof.

3.1.3 The Algorithm

The proposed algorithm steps are summarized as follow:

1. Start with a random number of uniformly distributed snapshots (parameters), the parameter space could include time, Mach number, angle of attack, design parameters and others, depending on the targeted application

2. Consider the parameter space as a mesh (P-mesh) then for each snapshot calculate the proposed error at snapshot location (node) by

$$Err_i=\|\phi_i\|^2-\frac{1}{trace(K)}\sum_{j=1,j\neq i}^{N}(\phi_i,\phi_j)^2 \tag{3.12}$$

3. estimate the error $\varepsilon_{i,j}$ along each edge $e_{i,j}$ either by taking the Hessian (with the absolute value as described in subsection 3.1.1) of Err_i and Err_j or use

$$\varepsilon_{i,j}=\frac{Err_i+Err_j}{2}\|e_{i,j}\| \tag{3.13}$$

4. Apply the local mesh adaptivity algorithm described in subsection 3.1.1, till a uniform error distribution is obtained and equal to prescribed error threshold. This mechanism will move, add, and delete snapshots locations in order to put more snapshots where needed and remove unnecessary snapshots for an optimal POD basis system

3.2 POD Projection Coefficients Estimate

Another challenging task when POD is applied to solve PDEs (Chapter 2) is to estimate the projection coefficients onto POD basis system, indeed if $\{\Psi_j(x)\}_{j=1}^N$ is the POD basis and $u(x,t)$ is the POD solution of the concerned PDE, then

$$u(x,t) \approx \sum_{j=1}^{N} y_j(t)\Psi_j(x) \tag{3.14}$$

To estimate the coefficient $\{y_j(t)\}_{j=1}^N$ the natural approach is to substitute the POD solution in the PDE equation and then solve the obtained dynamical system in $y_j(t)$ (or an algebraic system if the PDE is time independent) (see Chapter 2). This dynamical system (as mentioned in Chapter 2) can be quadratic, cubic etc... system and therefore not easy to solve and time consuming, for this reason in literature the coefficients are obtained by interpolation(cubic, bi-splines,...) rather than solving the nonlinear dynamical system. However, and whatever the used approach, it is pointed out in the literature [60] and verified by our tests, difficulties are observed for shocked solution. In such a problem is demonstrated on a simple example.

3.2.1 POD for shocked solutions

Let's consider the simple pure parabolic convection initial value problem,

$$\begin{cases} \frac{\partial Q}{\partial t} + \frac{\partial Q}{\partial x} = 0, \quad (t,x) \in [0,\infty) \times IR \\ Q(0,x) = H(x) = \begin{cases} 1 & if\ x \geq 0 \\ 0 & if\ x < 0 \end{cases} \end{cases} \tag{3.15}$$

The exact solution is given by $Q(t,x) = H(x-t)$ (see Figs. 3.1 and 3.2)

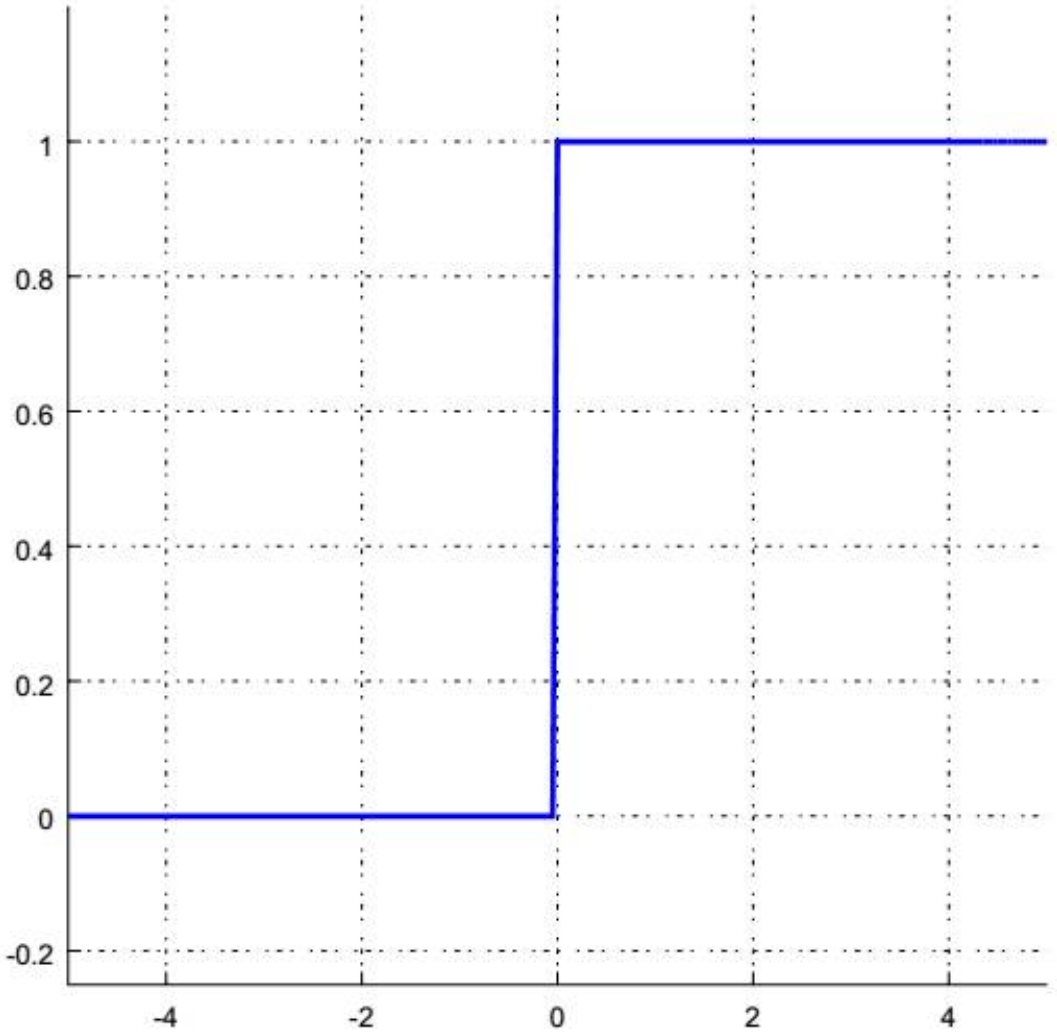


Figure 3.1: Initial condition of the convection parabolic equation

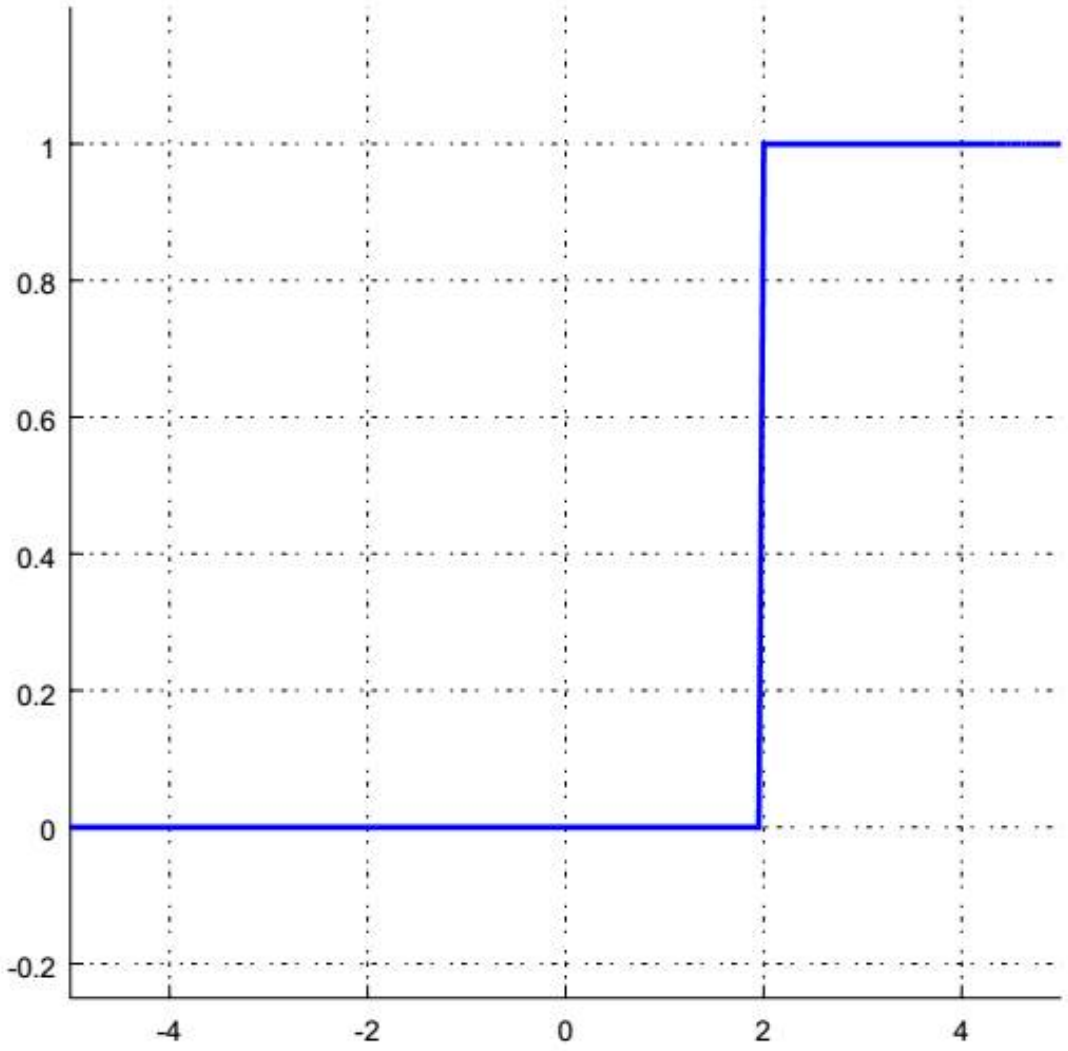


Figure 3.2: Solution of the convection parabolic equation at $t = 2$

Now apply the POD technique to solve the equation with the time t being a parameter.

Let $\{Q_i(x) = H(x - t_i)\}_{i=1}^{N}$ N solutions (snapshots) at times $t_1, t_2, ..., t_N$ Note that the POD solutions are 'smart' averaging of the snapshots, and if we need to estimate the solution at a time $\bar{t} \in (t_k, t_{k+1})$ we will get

$$Q_{POD}(\bar{t}, x) \approx \sum_{j=1}^{N} y_j(\bar{t})\Psi_j(x) \tag{3.16}$$

And this can be written as

$$Q_{POD}(\bar{t}, x) \approx \sum_{i=1}^{N} \alpha_i(\bar{t})Q_i(x) \tag{3.17}$$

since Ψ_j's are linear combinations of Q_i's

Now we want that $Q_{POD}(\bar{t}, x) \approx H(x - \bar{t})$ which means that

$$Q_{POD}(\bar{t}, x) \approx 1 \quad for \quad x \geq \bar{t} \tag{3.18}$$

in another hand

$$H(x - t_i) = 0 \qquad if x < t_i \tag{3.19}$$

This implies that for any $x \in (t_k, t_{k+1})$ (see Fig. 3.3)

$$Q_{POD}(\bar{t}, x) \approx \sum_{i=1}^{k} \alpha_i(\bar{t}) Q_i(x) + \sum_{i=k+1}^{N} \alpha_i(\bar{t}) Q_i(x) = \sum_{i=1}^{k} \alpha_i(\bar{t}) Q_i(x) \tag{3.20}$$

But we need $Q_{POD}(\bar{t}, x) \approx 0$ if $x < \bar{t}$ and $Q_{POD}(\bar{t}, x) \approx 1$ if $x \geq \bar{t}$ to be an acceptable solution. This leads to

$$\sum_{i=1}^{k} \alpha_i \approx 0 \qquad and \qquad \sum_{i=1}^{N} \alpha_i \approx 1 \tag{3.21}$$

Which is impossible to obtain.

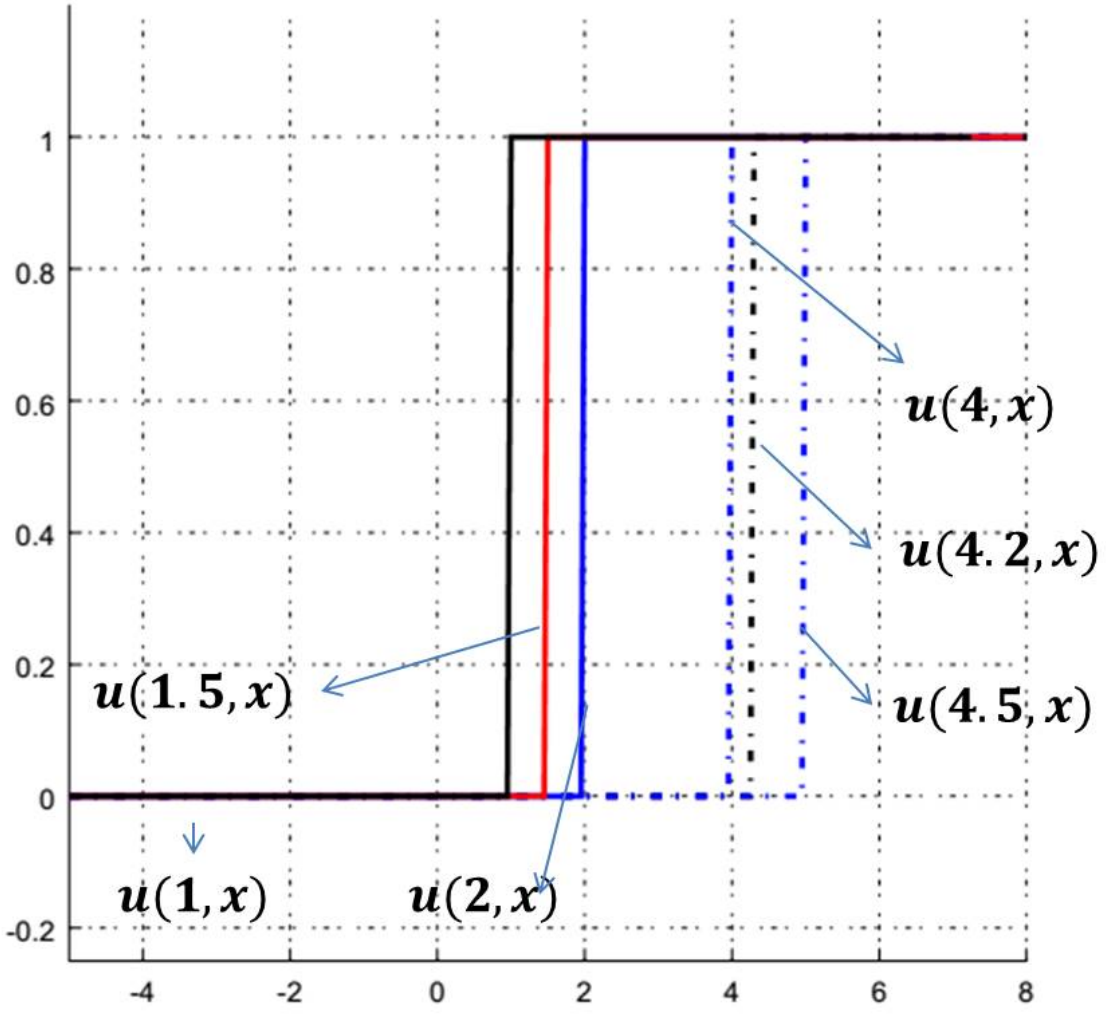


Figure 3.3: Solutions of the convection parabolic equation for different times. Here the interval $[t_1, t_2] = [2, 4]$

3.2.2 Closest Snapshot: prediction-correction approach

The previous example shows that it really hard to capture shocks properly with POD as long as we consider through any interpolation process the contribution of all snapshots. Based on that, we propose to consider the closest snapshot rather than interpolating. That is

$$u(x,t) \approx Arg\left(Min\{\|u(x,t)-\phi_j\|\}_{j=1}^{N}\right) \tag{3.22}$$

Of course we don't have $u(x,t)$ to take the min, we can think about taking the closest parameters, but as we will demonstrate in the validation section, closest parameters doesn't necessary mean closest solutions. Therefore we propose the following closest snapshot measure in the light of the prediction-correction spirit approach.

We assume that we already built the POD system with the proposed adaptive technique. We then proceed using the commonly used interpolation (cubic, spline,..). For such a purpose we need first to project the snapshots onto the POD basis system (Modes) ${\Psi_j}_{j=1}^{N}$, then interpolate component by component (prediction step).

Any snapshot can $\phi_i(x,t)$ be projected onto the POD basis ${\Psi_j}_{j=1}^{N}$ as follows,

$$\phi_i(x,t) = \sum_{j=1}^{N} y_j^i(t)\Psi_j(x) \tag{3.23}$$

Note that in general modes with low energy are neglected then the previous sum is reduced to

$$\phi_i(x,t) = \sum_{j=1}^{M} y_j^i(t)\Psi_j(x) \tag{3.24}$$

with $M << N$. But without loss of generality let's keep the sum to N

The coefficients $y_j^i(t)$ are given by (use orthonormality of Ψ_j)

$$y_j^i(t) = \Big(\phi_i(x,t), \Psi_j\Big) \tag{3.25}$$

$(\,,)$ being the inner product used to build the correlation $N \times N$ matrix K. In practice the inner product (generally L^2 or H^1) are computed numerically, for more accuracy we propose in the following exact value of the coefficients y_j^i .

Let V the matrix of eigenvectors of K and $\lambda_1, \lambda_2, ..., \lambda_N$ the associated eigenvalues (see chapter 2 for details).

Set

$$V = \begin{pmatrix} v_1^1 & v_1^2 & \dots & v_1^N \\ v_2^1 & v_2^2 & \dots & v_1^N \\ \vdots & \vdots & \vdots & \\ v_N^1 & v_N^2 & \dots & v_N^N \end{pmatrix}, \tag{3.26}$$

The modes Ψ_j are the projection of the snapshots $\phi_i(x,t)$ onto the eigenvectors,

$$\Psi_j = \sum_{k=1}^{N} v_k^j \phi_k \tag{3.27}$$

Substituting Ψ_j in equation 3.25 we obtain

$$y_j^i(t) = (\phi_i(x,t), \Psi_j) = (\phi_i(x,t), \sum_{k=1}^{N} v_k^j \phi_k) = \sum_{k=1}^{N} v_k^j (\phi_i, \phi_k) \tag{3.28}$$

The last expression is the product of the i^{th} row of the correlation matrix K with the j^{th} eigenvector. we deduce (since $KV^j = \lambda_j V^j$) that

$$y_j^i(t) = \lambda_j v_i^j \tag{3.29}$$

Now to estimate any new solution with the POD system, apply any interpolation method to the components y_j^i as function of the parameters space, and obtain y_j^{POD}. The next step (correction step) is to take the closest snapshot based on POD projection coefficients of the snapshots and the interpolated ones,

if we refer by $y_j^{POD-final}$ to the POD solution output, then

$$y_j^{POD-final} = Arg\left(Min_i(\sum_{j=1}^{N}(y_j^i - y_j^{POD})^2)\right), \quad for \quad j = 1, N \tag{3.30}$$

3.3 Validation

The proposed method is implemented in MATLAB environment for validation. First the method is assessed against exact solutions, the goal of this step is to better (since we have a full control on the solution) demonstrate the effectiveness of the proposed approach including the accuracy of the reconstructed solutions and more precisely the validity of the proposed error distribution and adaptivity process, a corner stone of an effective snapshots locations distribution. In a second step and for a validation on a practical physical problems, the MATLAB code is coupled with an open source CFD solver SU2 (Stanford university unstructured) developed by Stanford university [84, 68, 35].

3.3.1 Validation on Exact Solutions

Let us consider the following incompressible, steady state flow given by the following governing equations:

$$\begin{cases} v\left(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}\right) = 0 \\ u\left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}\right) = 0 \end{cases} \tag{3.31}$$

Then let us define on the whole plan $\mathbb{R}^2$ the following analytical velocity field

$$\begin{cases} u = \sin^2(\alpha(x^2 + y^2)) \\ v = \cos^2(\alpha(x^2 + y^2)) \end{cases} \tag{3.32}$$

with

$$\alpha = \frac{1}{\beta((a - a_0)^2 + (b - b_0)^2} + \varepsilon \tag{3.33}$$

ε and β are given values, a, b, a_0 and b_0 belong to the parameter space. Note that the selected velocity field didn't correspond to a physical solution, this is motivated

in one hand to obtain an exact solution for the considered PDEs system, in another hand to create important oscillations around a_0, b_0 to show clearly the effect of adaptivity . In our tests, we set $\beta = 10$, $\varepsilon = 0.1$, $a_0 = 0.4$ and $b_0 = 0.4$

Now, to force the above functions to be solutions of (3.31) with appropriate source terms f_1 and f_2, we substitute their expressions into the equations and then obtain the following new system:

$$\begin{cases} v\left(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}\right) = f_1 \\ u\left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}\right) = f_2 \end{cases} \tag{3.34}$$

where

$$f_1 = \cos^2(\alpha(x^2+y^2))(x+y)(4\alpha\sin(\alpha(x^2+y^2))\cos(\alpha(x^2+y^2))$$
$$f_2 = \sin^2(\alpha(x^2+y^2))(x+y)(-4\alpha\sin(\alpha(x^2+y^2))\cos(\alpha(x^2+y^2))$$

Consider now the problem (3.34) in a rectangular domain and impose boundary conditions extracted from the exact solution (3.32). The rectangle is discretized using uniform 20×20 quads. We first start with a uniform distribution of 49 initial points from the parameter space, referred to as P-mesh for simplicity, shown in Fig.3.4. The deriving error is estimated using the proposed LOOCV formula. Then, the adaptivity algorithm described in section 3.1.1 is applied.

Fig.3.5 shows the adapted P-mesh, in which we can see that the mesh is enriched and the nodes are more concentrated in the area of high oscillations, as expected.

To assess the impact of the adapted P-mesh, error distribution is calculated by considering the L^2 norm of the difference between the exact solution and estimated POD solution. Fig.3.6 and Fig.3.7 show the error distribution for the original P-mesh and the adapted one. We can see that the error distribution (POD-Exact solutions) using the original P-mesh (light gray) is sensibly reduced (dark gray) after adaptivity.

The same tests are repeated using the Hessian-based metric of the LOOCV-based error. The results, depicted in Figs. 3.8 and 3.9, showed similar results. Note that in situations where the relevant parameters have an anisotropic distribution, the use of the Hessian-based metric approach will allow capturing anisotropy and then obtaining more efficient snapshots locations with a reduced number of snapshots. Cross sections of error surfaces at different parameter values are depicted in Figs. 3.10, 3.11 and 3.12.

The performed tests demonstrated the efficiency of the proposed snapshots locations algorithm and the accuracy of the reconstructed POD solutions.

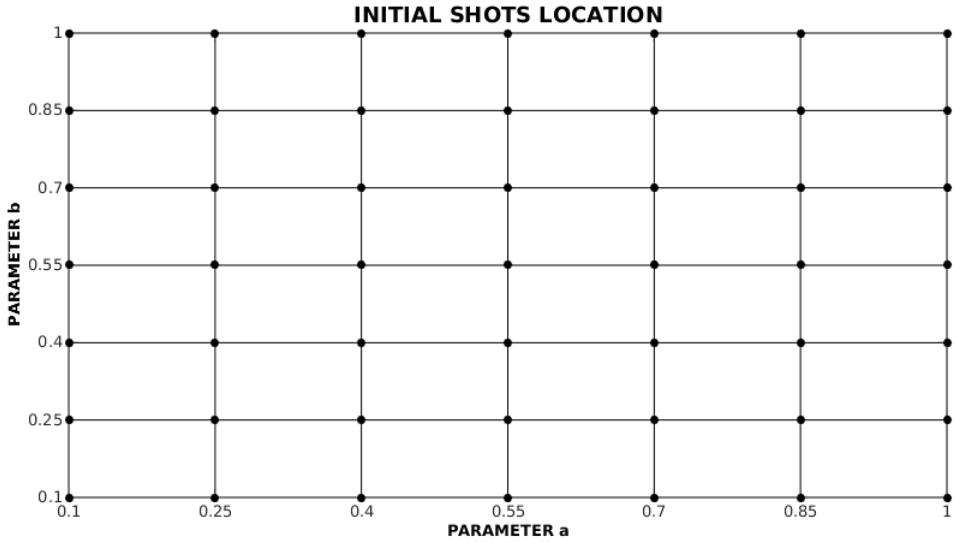


Figure 3.4: Original P-Mesh: Uniform distribution

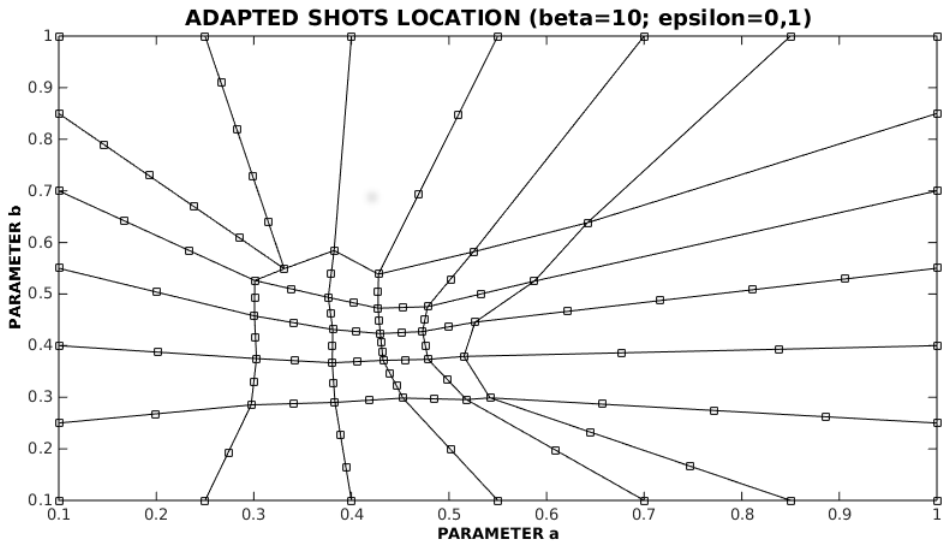


Figure 3.5: Adapted P-mesh using LOOCV-based error

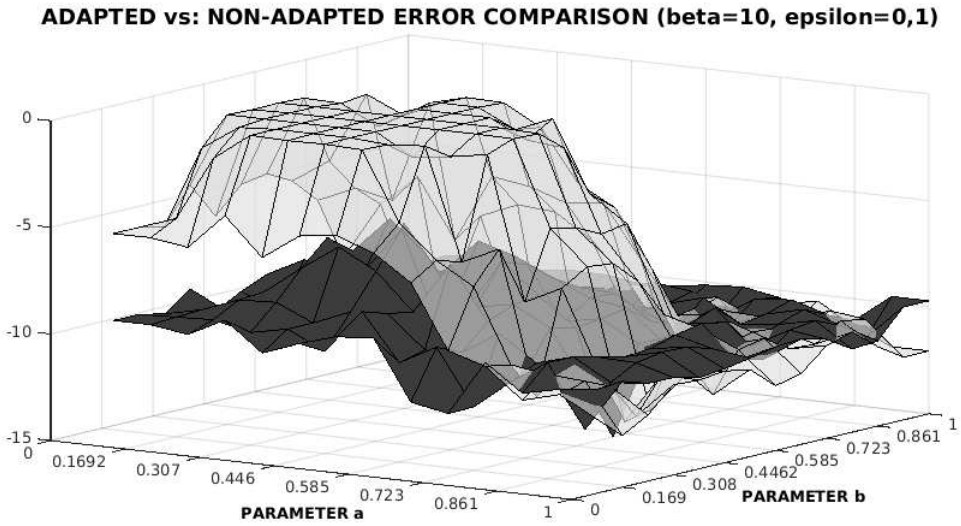


Figure 3.6: Logarithmic POD-Exact solution error distribution before(light gray) and after (dark gray) adaptivity for $\beta = 10$, $\varepsilon = 0.1$

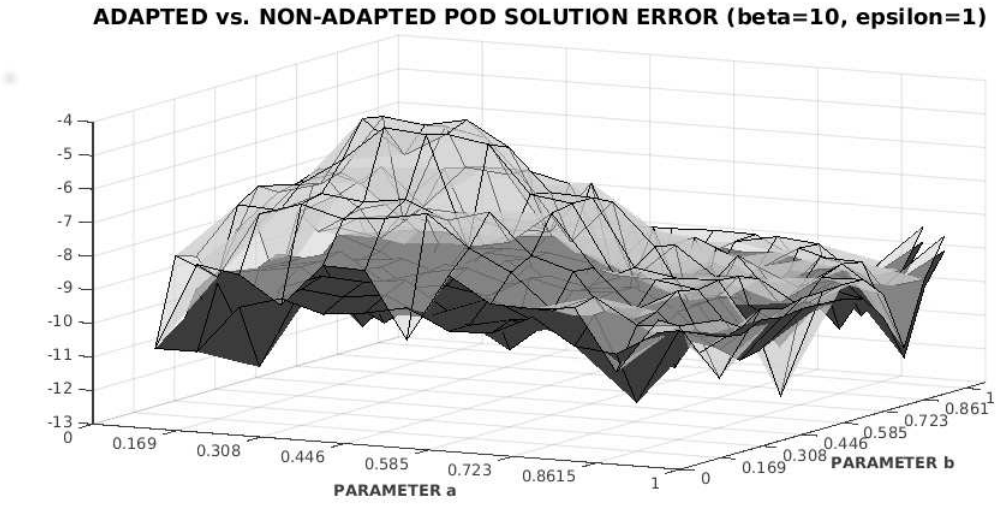


Figure 3.7: Logarithmic POD-Exact solution error distribution before(light gray) and after (dark gray) adaptivity for $\beta = 10$, $\varepsilon = 1$

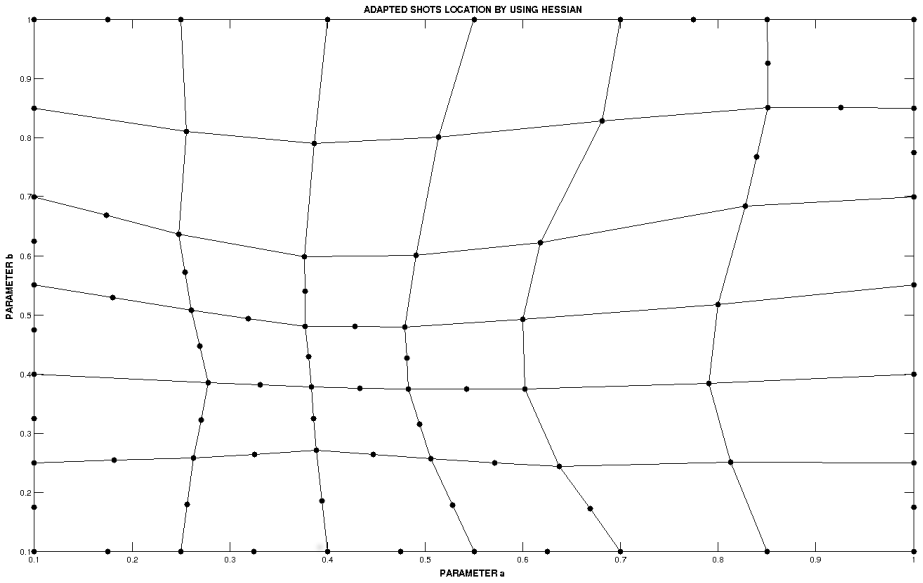


Figure 3.8: Adapted P-mesh using the LOOCV-based error Hessian

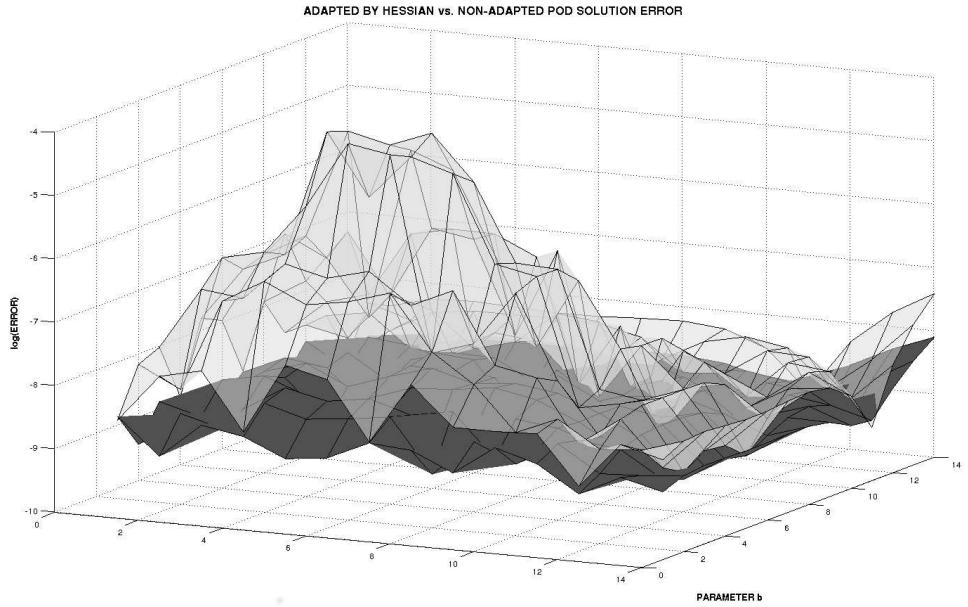


Figure 3.9: Using Hessian-based error: Logarithmic POD-Exact solution error distribution before(light gray) and after (dark gray) adaptivity

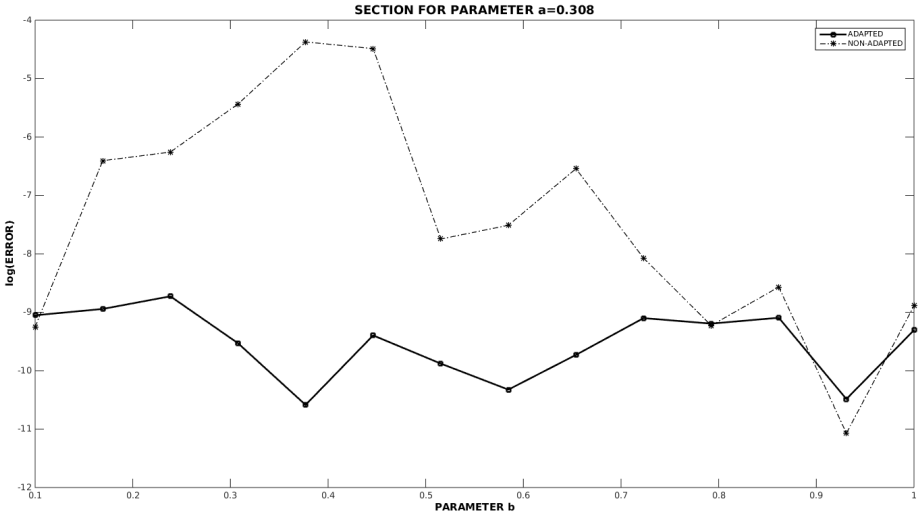


Figure 3.10: Cross section of POD-Exact solutions error distribution before(dashed line) and after (solid line)

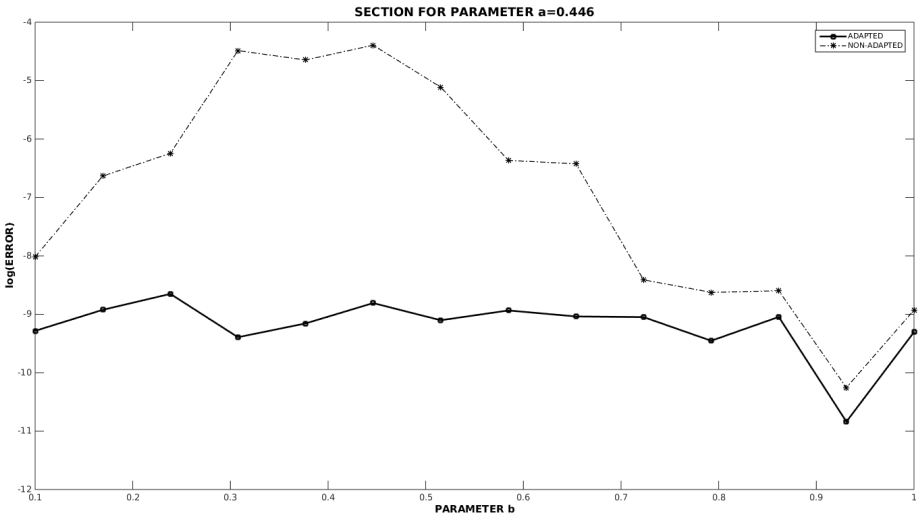


Figure 3.11: Cross section of POD-Exact solutions error distribution before(dashed line) and after (solid line)

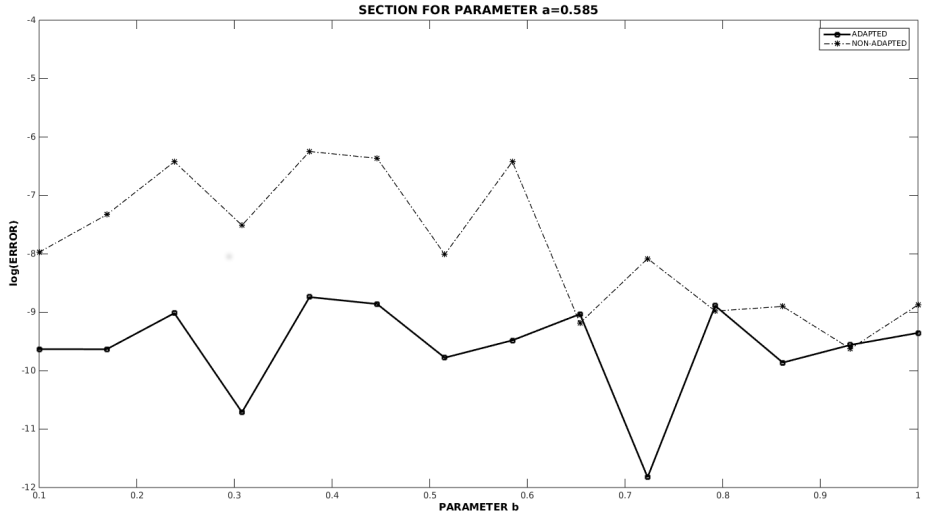


Figure 3.12: Cross section of POD-Exact solutions error distribution before(dashed line) and after (solid line)

3.3.2 Validation on CFD solutions: Flow Around Airfoil

For a validation against a physical CFD solutions, inviscid flow around a NAC0012 is considered. The following Euler equations are solved

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u)}{\partial x} + \frac{\partial (\rho v)}{\partial y} = 0 \tag{3.35}$$

$$\frac{\partial (\rho u)}{\partial t} + \frac{\partial}{\partial x}(\rho u u) + \frac{\partial}{\partial y}(\rho u v) + \frac{\partial p}{\partial x} = 0 \tag{3.36}$$

$$\frac{\partial (\rho v)}{\partial t} + \frac{\partial}{\partial x}(\rho v u) + \frac{\partial}{\partial y}(\rho v v) + \frac{\partial p}{\partial y} = 0 \tag{3.37}$$

$$\frac{\partial E}{\partial t} + \frac{\partial}{\partial x}\Big((E+p)\, u \Big) + \frac{\partial}{\partial y}\Big((E+p)\, u \Big) = 0 \tag{3.38}$$

Pressure is related to the conservative flow variables, ρ, ρu, ρv and E, by the equation of state

$$p = (\gamma - 1)\Big(E - \frac{1}{2}\rho \left(u^2 + v^2\right) \Big) \tag{3.39}$$

where γ is the ratio of specific heats, generally taken as $\gamma = 1.4$.

The coefficients of the dynamical system obtained after substituting the projected solution on POD basis for each field variable are given in appendix A after neglecting third order terms in projection coefficients. Details are not given here since and as we mentioned along this manuscript, the dynamical system is not solved, interpolation approach is used instead to estimate the projection coefficients (coordinates on POD basis)

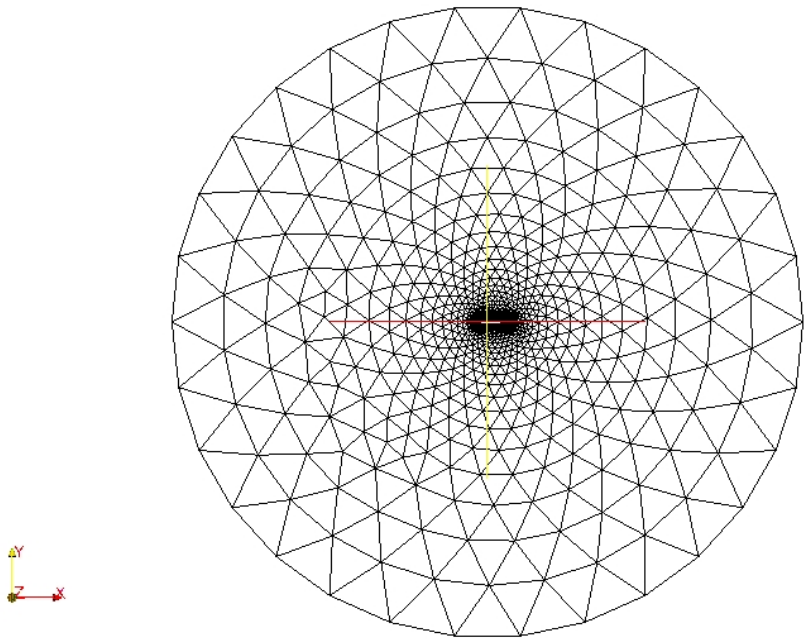


Figure 3.13: CFD mesh

The parameter space is formed by AoA and Mach number and the range $(\theta, M) \in [0, 5] \times [0.3, 0.9]$ is selected. An unstructured mesh of 3762 triangular elements (see Fig. 3.13 and Fig. 3.14) is used for all simulations. An initial uniform P-mesh (parameters space) is obtained by subdividing the interval $\in [0, 5] \times [0.3, 0.9]$ into 10 sub-intervals,then the initials snapshots (CFD solutions) at each parameter node are obtained using SU2 solver.The proposed algorithm is then run to improve/enrich snapshots locations and new snapshots are calculated at new nodes. Results before and after P-mesh improvement are compared to confirm the efficiency of the proposed method already demonstrated on exact solutions in previous subsection.

Before showing results and to confirm the interpolation issue discussed in section 3.22, tests are performed on the uniform P-mesh and POD projection coefficients are calculated using cubic-spline interpolation. Fig. 3.15 shows that for subsonic solution (smooth solutions) the interpolation can provide a good solution reconstruction (not always unfortunately). However Fig. 3.16 shows that for transonic solutions (shocked solution) the method faces real difficulties to reconstruct the solution as it was pointed out before. Of course result can be improved by increasing considerably the number of snapshots which is not a desired thing to do.

To demonstrate the efficiency the proposed snapshots locations, Fig. 3.17 shows the uniform P-mesh and the adapted one, as one can expect the P-mesh is enriched in the area of high Mach and High AoA for which the physics is more complex. This confirms the validity of the proposed a posteriori error LOOCV proposed formula.

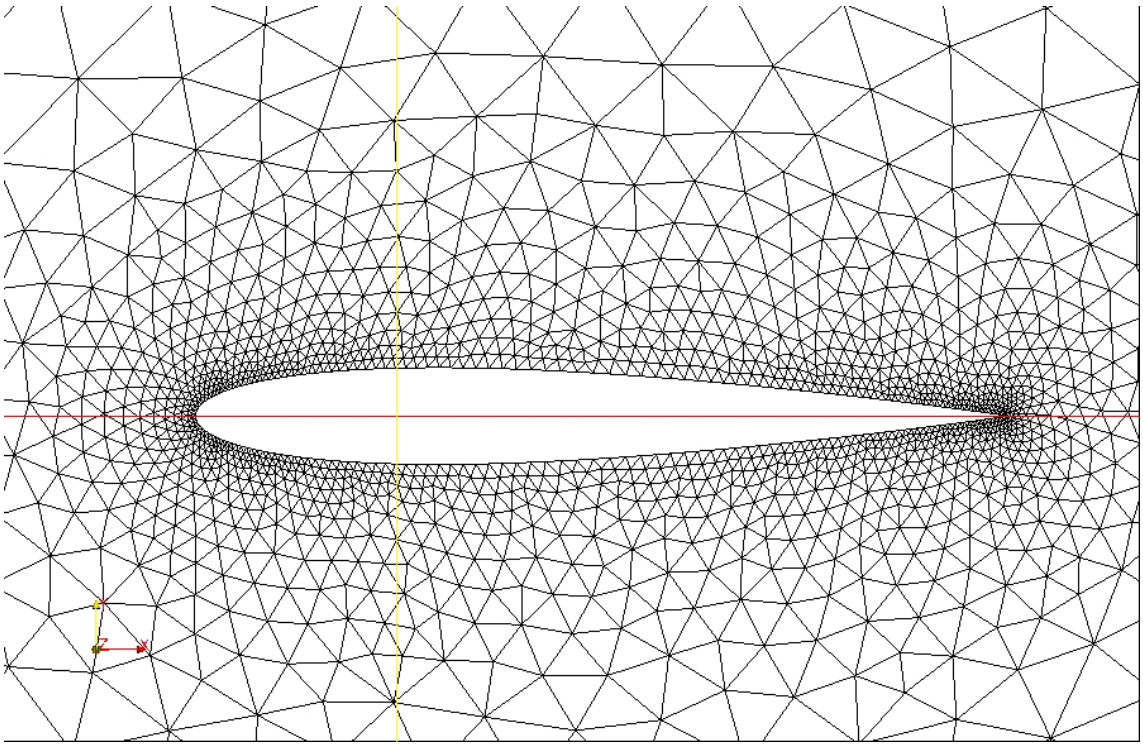


Figure 3.14: CFD mesh -zoom

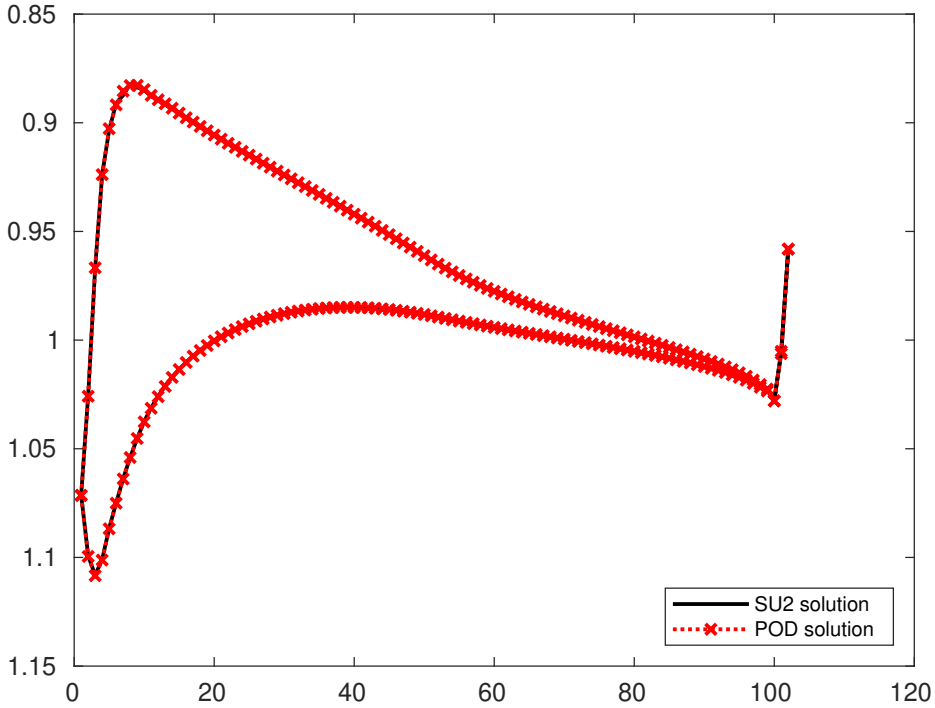


Figure 3.15: Airfoil Pressure (AoA=3.125, M=0.375):Subsonic CFD and POD solutions

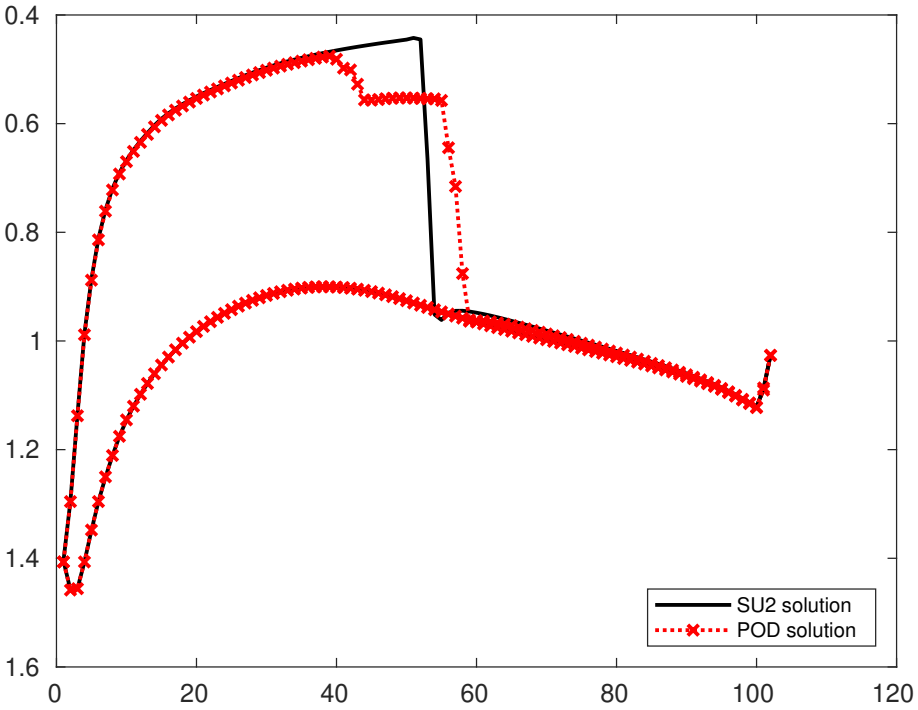


Figure 3.16: Airfoil Pressure (AoA=3.125, M=0.75):Transonic CFD and POD solutions

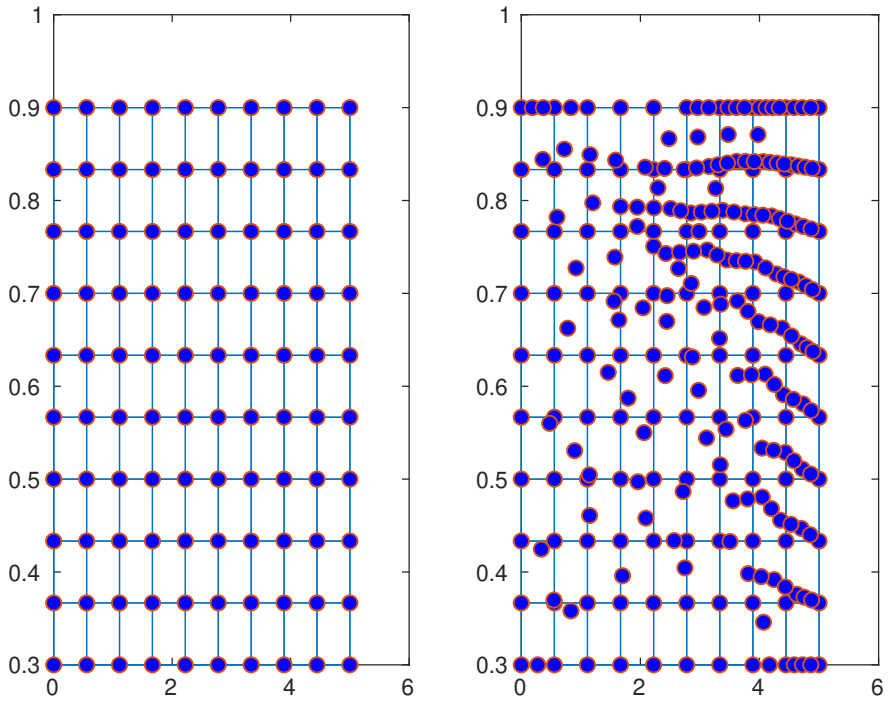


Figure 3.17: Initial (left) and Adapted (right) P-mesh: Snapshot locations insertion and improvement

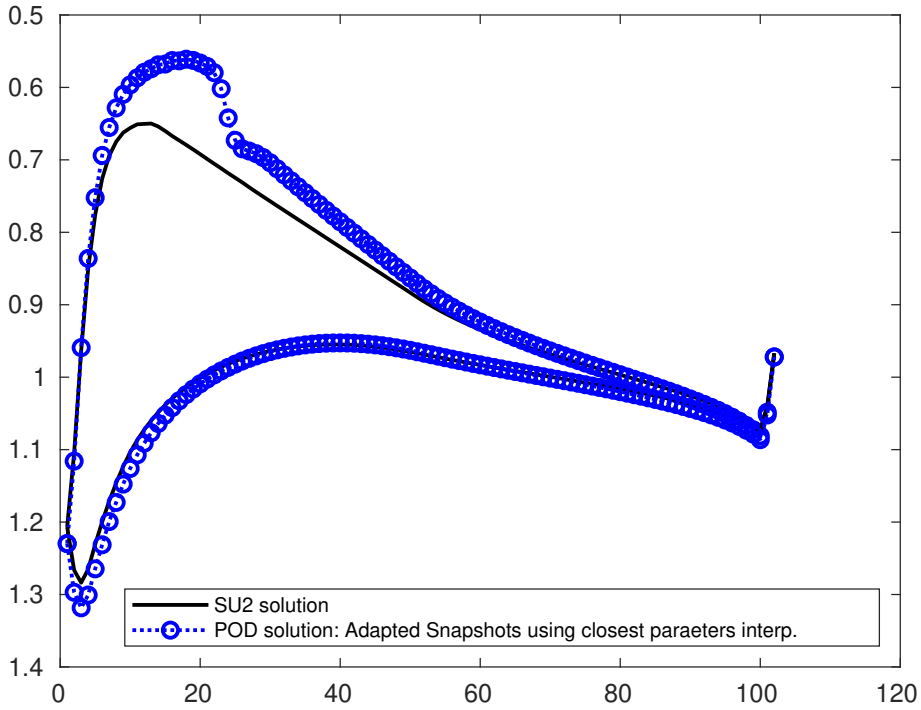


Figure 3.18: Airfoil Pressure(AoA=3.125, M=0.6): Before and after snapshots locations improvement

As we mentioned in this chapter taking the closest snapshot based on the closest parameters is not accurate, Figs. 3.18, 3.19 and 3.20 depict airfoil pressure distributions for different AoA and MAch numbers on adapted . The results show a clear discrepancy between POD solutions and CFD solutions. This demonstrates that the use of closest snapshot based on closest parameter is not the best choice. In the following this results will be repeated with the proposed closest snapshot estimation method

For the rest of the tests, the proposed closest snapshot (subsection 3.2.2) is used. Figs.3.24, 3.21, 3.25, 3.22 and 3.23 show for different values of AoA and Mach numbers (from subsonic to supersonic regimes) the reconstructed POD solution before and after adaptation. First Figs.3.21, 3.22, 3.23 are the same tests done above to evaluate closest snapshot parameters based, we can see that using the proposed 'interpolation' (prediction-correction based) results are sensibly improved.

The whole showed tests confirm the effectiveness of the proposed strategy to build a POD method for PDEs already demonstrated with the exact solutions tests.

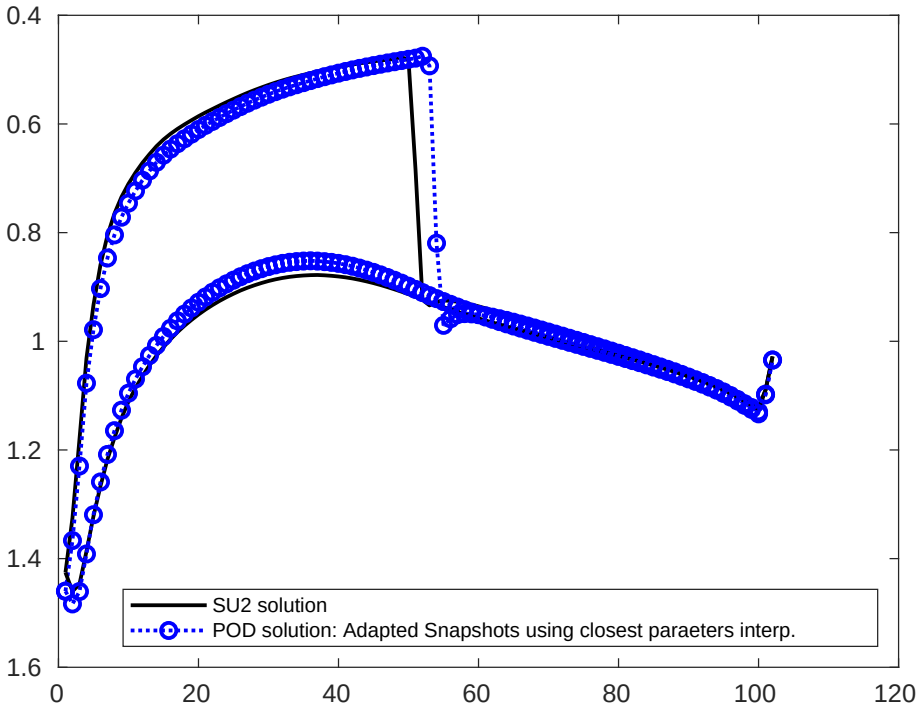


Figure 3.19: Airfoil Pressure (AoA=2.5, M=0.75): Before and after snapshots locations improvement

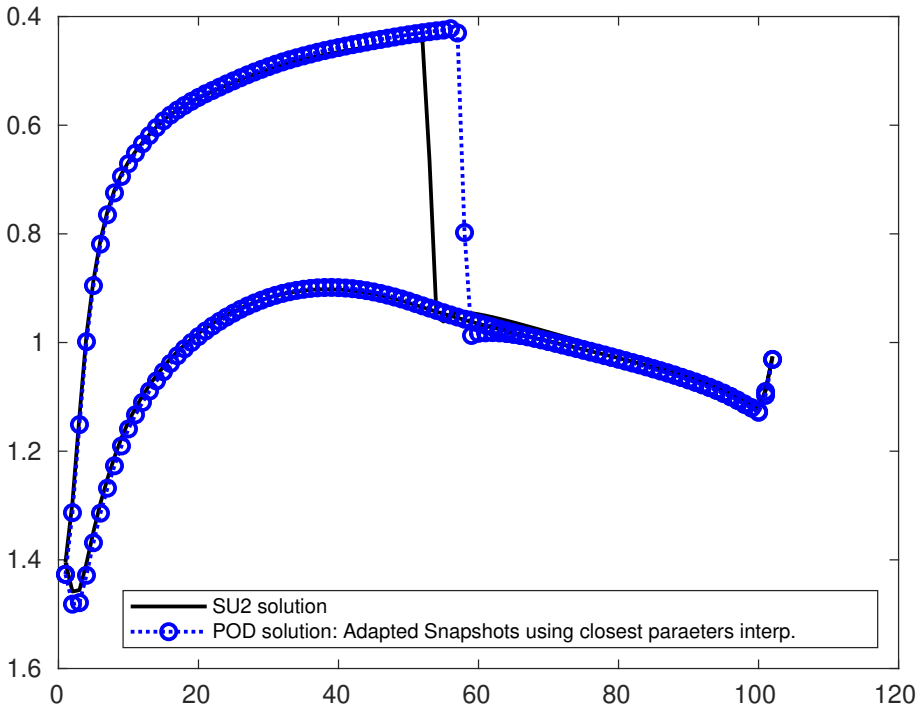


Figure 3.20: Airfoil Pressure (AoA=3.125, M=0.75): Before and after snapshots locations improvement

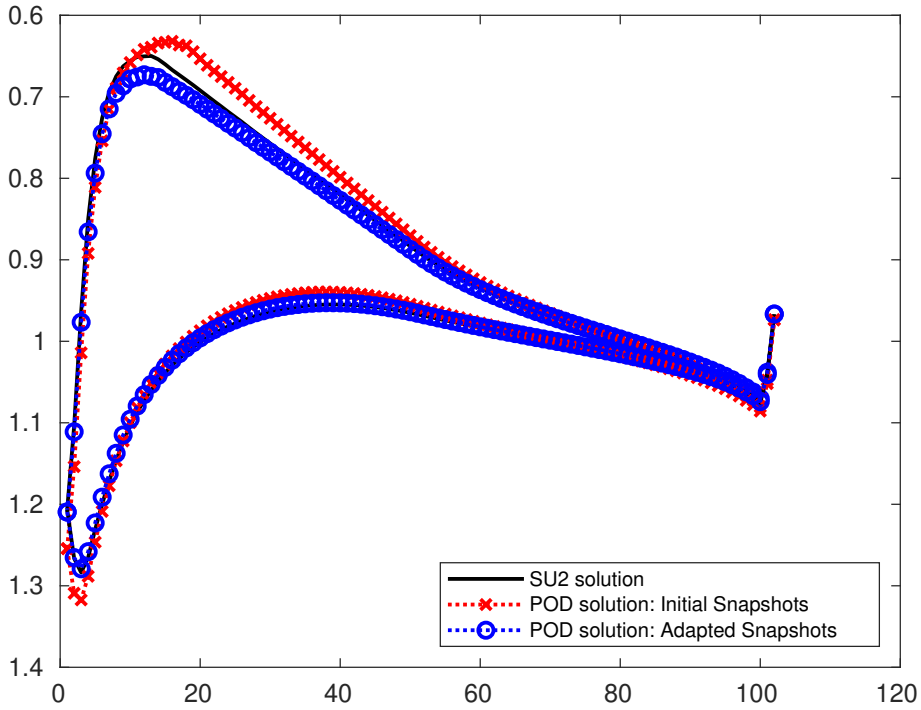


Figure 3.21: Airfoil Pressure (AoA=3.125, M=0.6): Before and after snapshots locations improvement

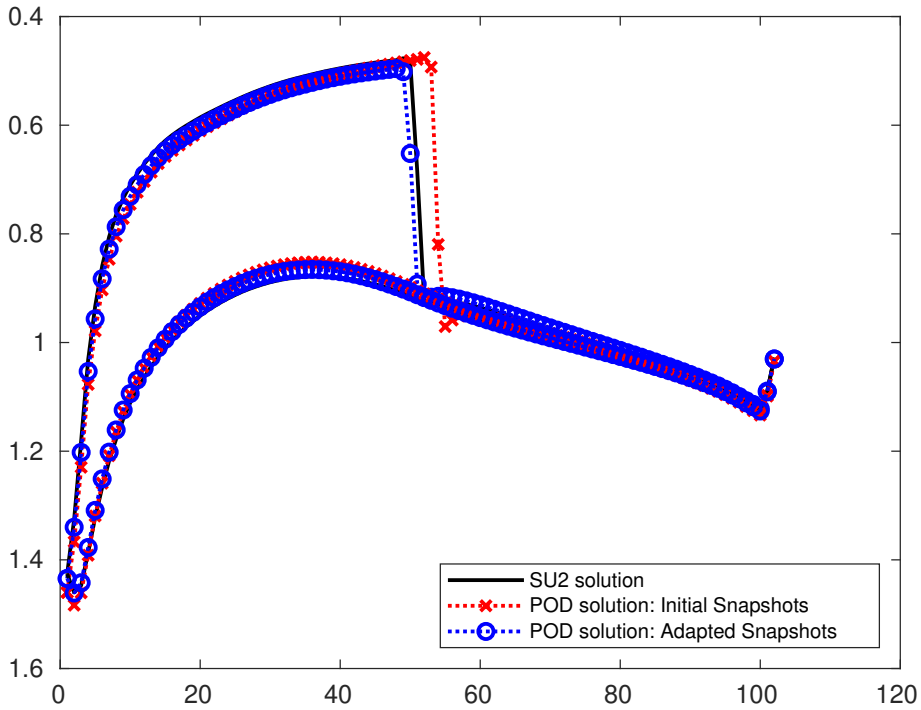


Figure 3.22: Airfoil Pressure(AoA=2.5, M=0.75): Before and after snapshots locations improvement

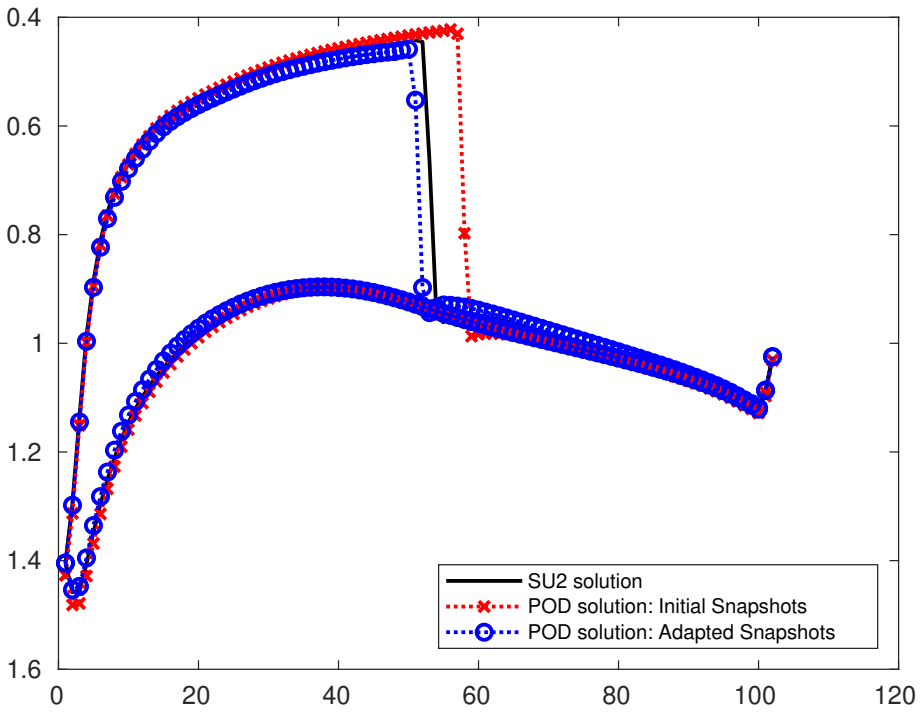


Figure 3.23: Airfoil Pressure (AoA=3.125, M=0.75): Before and after snapshots locations improvement

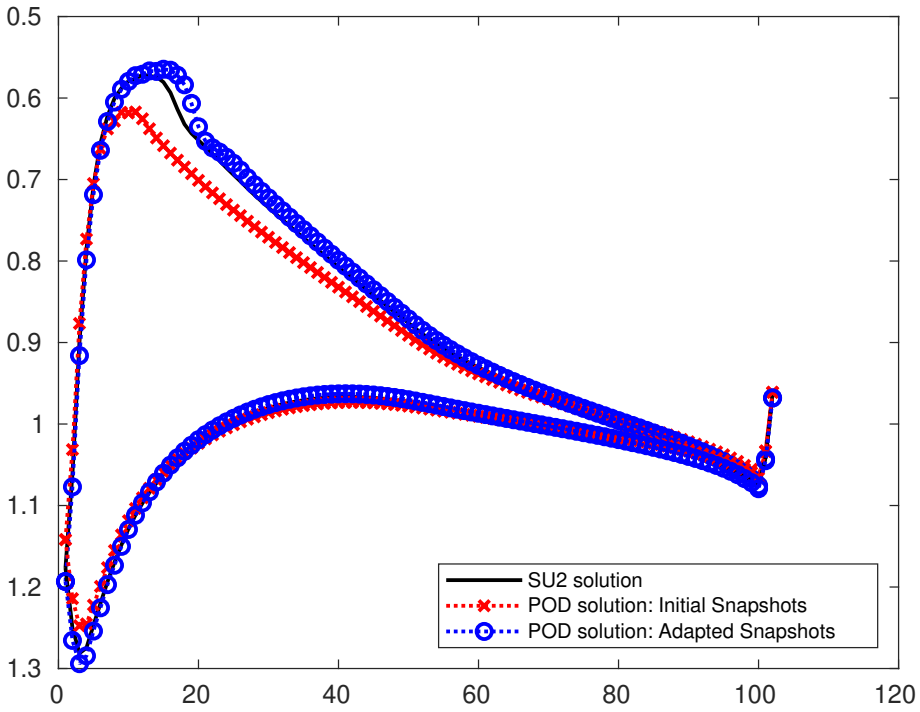


Figure 3.24: Airfoil Pressure (AoA=3.75, M=0.6): Before and after snapshots locations improvement

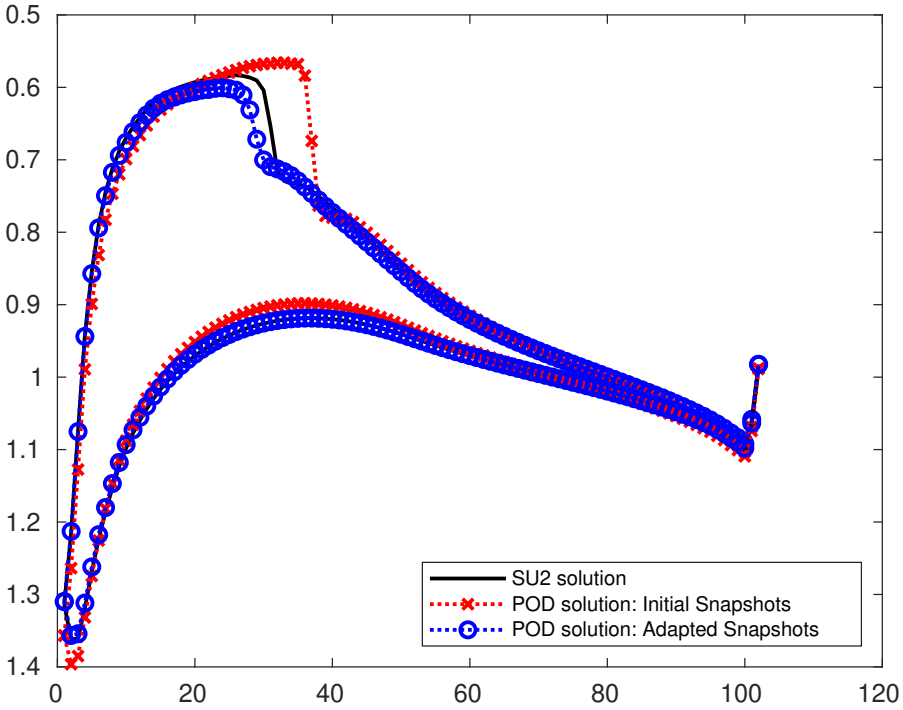


Figure 3.25: Airfoil Pressure (AoA=2.5, M=0.675): Before and after snapshots locations improvement

Chapter 4

ROMs Coupling Strategy

It was observed in literature that using POD (or any ROM method), results can be improved by decomposing the parameter domain into subdomains and then build different POD basis with low dimensions for each subdomain, this will reduce the prohibitive cost of a global high dimension POD basis (for high dimension parameter space) while keeping the same accuracy. Many strategies are proposed [3, 28, 29, 30, 37, 65, 89, 22]. In this chapter, we are not concerned by the cost but rather by unceasing the accuracy by coupling different ROMs methods on the same domain and this, to our knowledge, was never done. As we mentioned in the introduction, the POD method yields an optimal set of basis functions in the sense that no other decomposition of the same order captures an equivalent amount of kinetic energy [62, 58]. So any attempt to couple the method with any other ROM will result in degrading the accuracy. This is not completely true, let' first recall the POD optimality theorem,

Theorem. *Let $V_M = Span\{\Psi_1, ..., \Psi_M\} \subset H = L^2(\Omega)$, then for a subspace $Q_M = Span\{W_1, ..., W_M\} \subset H$ we have:*

$$\int_\Lambda d_H(u(.,.,\gamma), V_M)\mathrm{d}\gamma \leq \int_\Lambda d_H(u(.,.,\gamma), Q_M)\mathrm{d}\gamma \tag{4.1}$$

where $d_H(u(.,.,\gamma), V_M)$ is a distance from $u(.,.,\gamma)$ to the subspace V_M

As we can see from the theorem, POD optimality is guaranteed in the sense of average over the parameter space. This optimality is not satisfied pointwise, as we will demonstrate later on. So, the idea is to build another basis under some criterion such that the POD results could be improved in some areas (locally) of the parameter space Λ by coupling without contradicting the optimality theorem.

Let to illustrate the idea on a very simple example: consider 3 vectors in the $\mathbb{R}^3$, where two of them are orthogonal and span the horizontal plan as in Fig.4.1. The third one is has an angle of 45 degrees with respect to the horizontal plane. Assume that we want to build a POD model of two dimensions (a plane) from the given vectors to represent the three vectors. Applying the POD method will result in the light grey plane passing between the horizontal one and the third vector as in Fig.4.2. Now, it is clear that three vectors close or belonging to the horizontal plane will be better represented by the horizontal plane. So, the idea is then to build another basis (ROM) that spans the horizontal plane and to find a mechanism such each vector of the initial space is projected on the appropriate plane. The first

challenge is to build a basis system (ROM) that could locally improve the results of the classical POD method, to take advantage of the coupling. In the following section we propose an example of a basis system that fulfils such a condition.

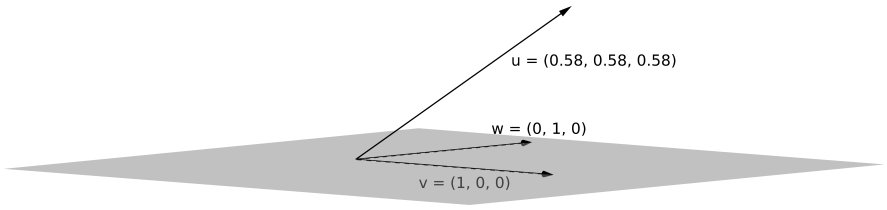


Figure 4.1: Three vectors in R^3 and the horizontal plan

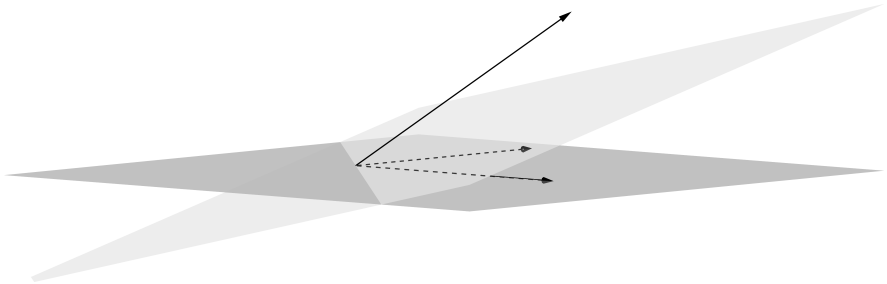


Figure 4.2: Light-gray plan shows the 2D plan obtained using POD

4.1 Locally competitive ROM

As mentioned above, before considering any coupling, we need first to build a ROM or equivalently a basis system from the snapshots that can locally improve the POD results. For such a purpose we need a criterion prior building the targeted basis system. First, note that the best scenario in building any snapshot-based basis is to have snapshots that are already orthogonal, assuming of course that the snapshots locations are optimal (in the broad sense), which can be obtained with the proposed adaptivity process (using the already built POD system). This case, the basis elements are solutions of the considered PDE providing consequently a maximum (local) information of the PDE. Consequently we can advance the following criterion; when building a basis system for a competitive ROM, the basis elements have to stay as close as possible to the snapshots, in another word, the desired basis system should be built with a minimum change of the original system of snapshots. Mathematically we can formulate this criterion as:

If we refer by $W_1, ..., W_d$ to the desired basis, and $\phi_1, ..., \phi_d$ to the snapshots, then

$$\{W_1, ..., W_d\} = Arg\left[\max_{U,\sigma} \sum_{k=1}^{d} \frac{|\left(U_k, \phi_{\sigma(k)}\right)_H|}{\|U_k\|_H \|\phi_{\sigma(k)}\|_H}, \quad subject\,to \;\; U_1, ..., U_d \quad \text{orthogonal}\right] \tag{4.2}$$

where H is a functional space, in general $H = L^2$ or $H = H^1$, and σ are all possible one-to-one functions from a natural number set of dimension d (the desirable dimension) onto a natural numbered set of dimension N (the initial shots set size).

It is very difficult to solve the optimization problem (4.2) exactly, instead we will provide an algorithm that attempts to follow the spirit of the criterion, after which some validation will be provided.

4.1.1 Sorted Gram Schmidt (SGS) algorithm

We propose to build new basis from the snapshots using the classical Gram Schmidt process but after sorting the snapshots according to problem (4.2). We refer to this procedure as the Sorted Gram Schmidt (SGS) algorithm.

The algorithm is summarized as follows.

1. Find the least correlated two snapshots among the set of snapshots.
2. Apply Gram-Schmidt (GS) orthogonalization for the two snapshots found in step 1.
3. Find the least correlated snapshot to the previous subspace spanned by the basis elements built so far.
4. Complete GS orthogonalization with the selected snapshot in 3.
5. Use the correlation computed in 3 as a stopping test. If it is bigger than a given threshold stop, otherwise repeat from step 3.

4.2 POD-SGS coupling

To couple the POD and the SGS methods, we need first to have an error distribution on the P-mesh nodes for each method, such that for any new solution estimate (at new parameter node) we first interpolate the error for both method and then go with the lowest one. Of course the LOOCV can be used however we cannot use the proposed cost-effective formula for SGS since it is valid for POD only. Therefore we propose to use a project-based error. This is is defined as follow:

Let ϕ_i be a given shot of the set $S = \{\phi_1, ..., \phi_N\}$, let $B = \{\Psi_1, ..., \Psi_M\}$, be the corresponding POD or SGS basis. The error at $Node\ \ i$ is taken to be,

$$Err_i = \frac{\|\phi_i - \sum_{j=1}^{M} \big((\phi_i, \Psi_j)\, \Psi_j \big)\|}{\|\phi_i\|} \tag{4.3}$$

Of course we need to have in general $M < N$.

This error distribution is calculated for both method POD and SGS. This error is estimating how well each snapshot is represented by the POD or SGS subspace. Now, by interpolating on the P-mesh, we can estimate the error for a new targeted solution (prior to solving) for both POD and SGS and then select the more appropriate method to use based on the smallest error.

The proposed coupling strategy efficiency is validated on an exact (mathematical) solution in the following subsection.

4.3 Validation

Considering the same system as in the chapter 3 (3.31):

$$\begin{cases} v\left(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}\right) = 0 \\ u\left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}\right) = 0 \end{cases} \tag{4.4}$$

And the following velocity field

$$u = v = \begin{cases} e^{\frac{\beta^2}{(x-a)^2 + (y-a)^2 - \beta^2}}, & if (x-a)^2 + (y-a)^2 < \beta^2 \\ 0, & else \end{cases} \tag{4.5}$$

where, a is a parameter and β is the width of the bell. In our tests we set $\beta = 0.4$. Using the same process, substituting this expression into equation (4.4), we force it to be the exact solution of similar equations

$$\begin{cases} v\left(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}\right) = f_1 \\ u\left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}\right) = f_2 \end{cases} \tag{4.6}$$

with source terms

$$f_1 = (x-y) \quad \frac{2\beta^2}{\left((x-a)^2 + (y-a)^2 - \beta^2\right)^2} \quad e^{\frac{2\beta^2}{(x-a)^2 + (y-a)^2 - \beta^2}}$$

$$f_2 = (y-x) \quad \frac{2\beta^2}{\left((x-a)^2 + (y-a)^2 - \beta^2\right)^2} \quad e^{\frac{2\beta^2}{(x-a)^2 + (y-a)^2 - \beta^2}}$$

The goal of this test is to demonstrate the possibility of coupling two ROMs (here POD and SGS) using the proposed strategy to achieve a local improvements of the efficiency of both methods. The POD method and the proposed SGS method were coupled as described in section 4.2 using the POD-projection and SGS-projection based errors.

Fig.4.3 shows the error-distribution solution for both methods, which clearly demonstrates that POD optimality is not satisfied pointwise.

Fig.4.4 shows that the errors in the POD and SGS solutions behave with perfect agreement with the corresponding POD-projection and SGS-projection errors, shown in the previous picture. This confirms our expectations and justifies the coupling procedure. Fig.4.5 shows the solution error of the coupled POD-SGS, we can see that the coupling is effective, the coupled method takes the best solutions from each method.

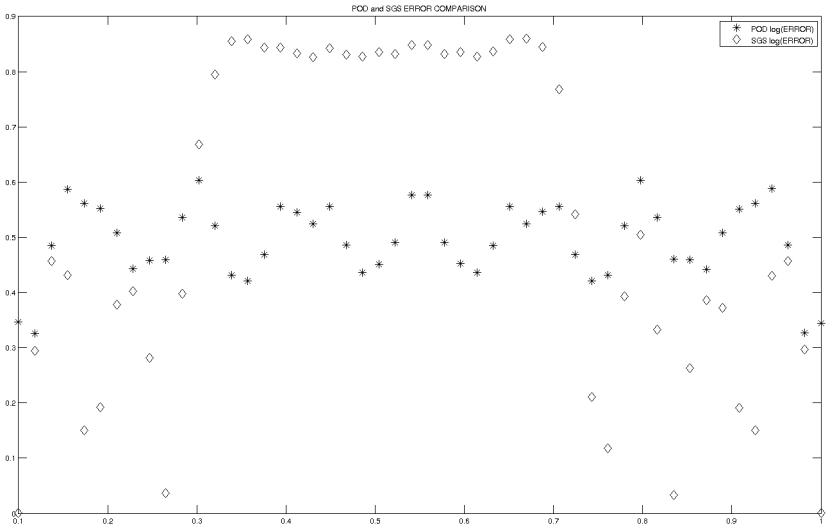


Figure 4.4: Solution error distribution for the POD and SGS based ROM

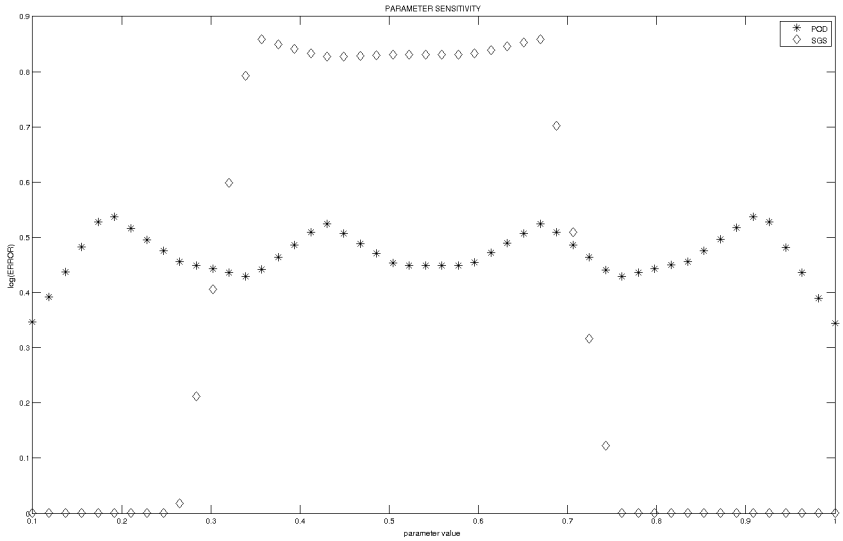


Figure 4.3: POD-projection and SGS-projection error distribution

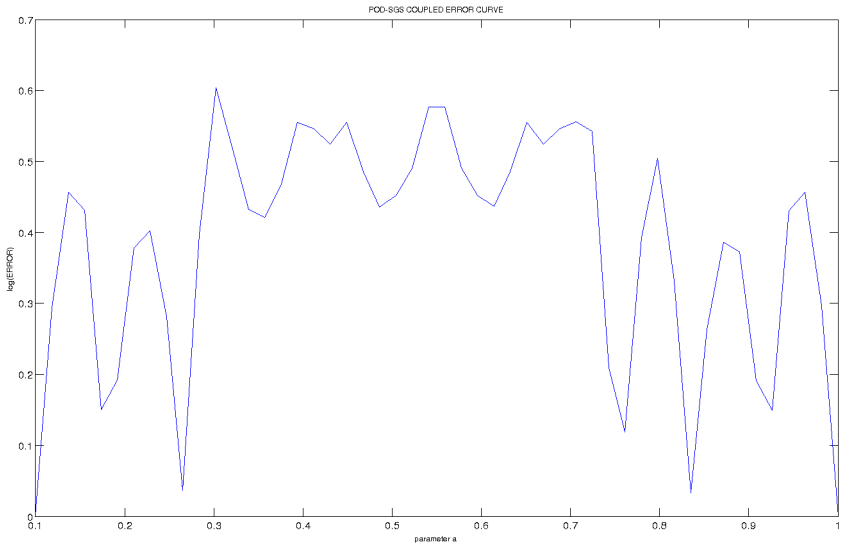


Figure 4.5: Solution error using coupled POD-SGS

Chapter 5

Conclusion

Reduction Order Methods (ROMS), received a sensitive interest growth last two decades, the principle is to reduce a problem initially defined in a high dimension space to an equivalent problem (up to a desired accuracy) defined in a sensitively low dimension for a cost-effectiveness solving and better analysis of inherent phenomena. This work deals with one of most popular ROMs, the Proper Orthogonal Decomposition (POD) in the context of partial differential equations (PDEs). Where parameters dependent solutions are first solved for a given set of parameters (called snapshots solutions) using a classical (yet expensive) CFD solver and then a POD subspace (with a reduced dimension) is built, spanned by a basis system obtained from the snapshots. Any new solution (for new parameters value) is then projected onto the POD subspace and then reconstructed with low cost. To build an efficient POD system, two challenging steps that received a big interest in literature are crucial, first the initial selection of the snapshots or equivalently the initial distribution of parameters, and alternatively, the mostly used terminology (adopted in this work), snapshots locations. Second the error estimate on the parameters space that drive the snapshots locations process. Note that as a consequence of these two steps, POD applied to PDEs is considered as belonging to the well-known Greedy algorithm family.

In this work we proposed an error estimate based on the a Leave-One-Out Cross-Validation method (LOOCV) technique. Each snapshot is disregarded and then the POD system is built with the remaining snapshots and the error is calculated at the considered parameter node to be the sensibility of the POD system to the absence of the considered snapshot. This has the advantage of being a universal upper bound error in the sense it is problem independent. However this is prohibitive for large systems due to the fact that the POD basis needs to be re-built for each disregarded node. To remedy to this shortcoming, a theorem is proven providing a formula to estimate the LOOCV error avoiding recalculating the POD system.

To drive the snapshot location selection an adaptive approach base on the well-known mesh adaptivity in CFD is proposed.

In this work we showed as well that using interpolation to estimate the projection coefficients on the POD basis has as a consequence a bad shocks capturing, as mentioned by many authors. A new interpolation (for shocked solutions) approach is then proposed based on the prediction-correction principle. The projection coefficients are first estimated with any interpolation technique, then the closest snapshot (based on the projection coefficients, not parameters) is taken as the POD solution.

The whole proposed strategy efficiency is validated against a mathematical (exact) solutions of an incompressible, steady state flow equations, and on CFD solutions of an inviscid flow around a NACA0012.

Another contribution of the thesis is a strategy to couple ROMs. It is pointed-out first that the POD is optimal in an average sense which mean that there is a room for a local improvement if it is coupled properly with another ROMs. We showed how to build a new basis for a ROM, based on a small modification of the Gram Schmidt algorithm that can do better locally than the POD. Then we provide how to do the coupling using a projection-based error. This strategy is validated on mathematical (exact) solutions of an incompressible, steady state flow equations.

Future work

As a future work we will intend to achieve more CFD tests for a more complete validation on more complex geometries and flows. We will investigate the exact solution of the challenging optimization problem we proposed for the coupling process.

Appendix A

Euler Equations projection onto Orthogonal POD Ψ_i modes

A.1 Euler Equations

Find here below stated the Euler equations for a two dimensional case. First conservation of mass, second and third conservation of momentum and finally conservation of energy. See [71] for more details.

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u)}{\partial x} + \frac{\partial (\rho v)}{\partial y} = 0 \tag{A.1}$$

$$\frac{\partial (\rho u)}{\partial t} + \frac{\partial}{\partial x}(\rho u u) + \frac{\partial}{\partial y}(\rho u v) + \frac{\partial p}{\partial x} = 0 \tag{A.2}$$

$$\frac{\partial (\rho v)}{\partial t} + \frac{\partial}{\partial x}(\rho v u) + \frac{\partial}{\partial y}(\rho v v) + \frac{\partial p}{\partial y} = 0 \tag{A.3}$$

$$\frac{\partial E}{\partial t} + \frac{\partial}{\partial x}\Big((E+p)\, u\Big) + \frac{\partial}{\partial y}\Big((E+p)\, u\Big) = 0 \tag{A.4}$$

Pressure is related to the conservative flow variables, ρ, ρu, ρv and E, by the equation of state

$$p = (\gamma - 1)\Big(E - \frac{1}{2}\rho\left(u^2 + v^2\right)\Big) \tag{A.5}$$

where γ is the ratio of specific heats, generally taken as $\gamma = 1.4$.

In next, $m_1 = \rho u$ and $m_2 = \rho v$. Then

$$\frac{\partial \rho}{\partial t} + \frac{\partial (m_1)}{\partial x} + \frac{\partial (m_2)}{\partial y} = 0 \tag{A.6}$$

$$\frac{\partial (m_1)}{\partial t} + \frac{\partial}{\partial x}(m_1 u) + \frac{\partial}{\partial y}(m_1 v) + \frac{\partial p}{\partial x} = 0 \tag{A.7}$$

$$\frac{\partial (m_2)}{\partial t} + \frac{\partial}{\partial x}(m_2 u) + \frac{\partial}{\partial y}(m_2 v) + \frac{\partial p}{\partial y} = 0 \tag{A.8}$$

$$\frac{\partial E}{\partial t} + \frac{\partial}{\partial x}\Big((E+p)\, u\Big) + \frac{\partial}{\partial y}\Big((E+p)\, u\Big) = 0 \tag{A.9}$$

A.2 Vector integration by parts. Green's Theorem

$$\int_{\Omega} u\nabla \cdot \mathbf{F} \mathrm{d}\Omega = \int_{S} u\mathbf{F} \cdot \mathbf{n} \mathrm{d}S - \int_{\Omega} (\nabla u) \cdot \mathbf{F} \mathrm{d}\Omega \tag{A.10}$$

A.3 Euler Equations projection onto Orthogonal POD Ψ_i modes

Now, as detailed in section 2.2.2, and in equation 2.11, Euler equations will be projected onto the orthogonal POD reduced basis, formed by modes Ψ_i,

$$\begin{aligned} \int_{\Omega} \frac{\partial \rho}{\partial t} \Psi_i^{\rho} + \int_{\Omega} \frac{\partial m_1}{\partial t} \Psi_i^{m_1} + \int_{\Omega} \frac{\partial m_2}{\partial t} \Psi_i^{m_2} + \int_{\Omega} \frac{\partial E}{\partial t} \Psi_i^{E} + \\ \int_{\Omega} \frac{\partial m_1}{\partial x} \Psi_i^{\rho} + \int_{\Omega} \frac{\partial m_2}{\partial y} \Psi_i^{\rho} + \\ \int_{\Omega} \frac{\partial u m_1}{\partial x} \Psi_i^{m_1} + \int_{\Omega} \frac{\partial v m_1}{\partial y} \Psi_i^{m_1} + \int_{\Omega} \frac{\partial p}{\partial x} \Psi_i^{m_1} + \\ \int_{\Omega} \frac{\partial u m_2}{\partial x} \Psi_i^{m_2} + \int_{\Omega} \frac{\partial v m_2}{\partial y} \Psi_i^{m_2} + \int_{\Omega} \frac{\partial p}{\partial y} \Psi_i^{m_2} + \\ \int_{\Omega} \frac{\partial}{\partial x} \Big((E+p)\, u \Big) \Psi_i^{E} + \int_{\Omega} \frac{\partial}{\partial y} \Big((E+p)\, v \Big) \Psi_i^{E} = 0 \end{aligned} \tag{A.11}$$

$$\begin{aligned} \int_{\Omega} \frac{\partial \rho}{\partial t} \Psi_i^{\rho} + \int_{\Omega} \frac{\partial m_1}{\partial t} \Psi_i^{m_1} + \int_{\Omega} \frac{\partial m_2}{\partial t} \Psi_i^{m_2} + \int_{\Omega} \frac{\partial E}{\partial t} \Psi_i^{E} + \\ \int_{\Omega} \nabla \cdot \begin{pmatrix} m_1 \\ m_2 \end{pmatrix} \Psi_i^{\rho} + \int_{\Omega} \nabla \cdot \left(m_1 \begin{pmatrix} u \\ v \end{pmatrix} \right) \Psi_i^{m_1} + \int_{\Omega} \nabla \cdot \left(m_2 \begin{pmatrix} u \\ v \end{pmatrix} \right) \Psi_i^{m_2} + \\ \int_{\Omega} \frac{\partial p}{\partial x} \Psi_i^{m_1} + \int_{\Omega} \frac{\partial p}{\partial y} \Psi_i^{m_2} + \int_{\Omega} \nabla \cdot \left((E+p) \begin{pmatrix} u \\ v \end{pmatrix} \right) \Psi_i^{E} = 0 \end{aligned} \tag{A.12}$$

Applying Green's Theorem vector integration by parts A.10 to equation A.12,

$$\begin{aligned}
\int_\Omega \frac{\partial \rho}{\partial t}\Psi_i^\rho + \int_\Omega \frac{\partial m_1}{\partial t}\Psi_i^{m_1} + \int_\Omega \frac{\partial m_2}{\partial t}\Psi_i^{m_2} + \int_\Omega \frac{\partial E}{\partial t}\Psi_i^E + \\
\int_{\partial\Omega} \left(\begin{pmatrix} m_1 \\ m_2 \end{pmatrix} \mathbf{n} \right) \Psi_i^\rho - \int_\Omega \begin{pmatrix} m_1 \\ m_2 \end{pmatrix} \cdot \nabla \Psi_i^\rho + \\
\int_{\partial\Omega} \left(\begin{pmatrix} m_1 u \\ m_1 v \end{pmatrix} \mathbf{n} \right) \Psi_i^{m_1} - \int_\Omega \begin{pmatrix} m_1 u \\ m_1 v \end{pmatrix} \cdot \nabla \Psi_i^{m_1} + \\
\int_{\partial\Omega} \left(\begin{pmatrix} p \\ 0 \end{pmatrix} \mathbf{n} \right) \Psi_i^{m_1} - \int_\Omega p \cdot \frac{\partial}{\partial x}\Psi_i^{m_1} + \\
\int_{\partial\Omega} \left(\begin{pmatrix} m_2 u \\ m_2 v \end{pmatrix} \mathbf{n} \right) \Psi_i^{m_2} - \int_\Omega \begin{pmatrix} m_2 u \\ m_2 v \end{pmatrix} \cdot \nabla \Psi_i^{m_2} + \\
\int_{\partial\Omega} \left(\begin{pmatrix} 0 \\ p \end{pmatrix} \mathbf{n} \right) \Psi_i^{m_2} - \int_\Omega p \cdot \frac{\partial}{\partial y}\Psi_i^{m_2} + \\
\int_{\partial\Omega} \left(\begin{pmatrix} (E+p)\,u \\ (E+p)\,v \end{pmatrix} \mathbf{n} \right) \Psi_i^E - \int_\Omega \begin{pmatrix} (E+p)\,u \\ (E+p)\,v \end{pmatrix} \cdot \nabla \Psi_i^E
\end{aligned} \tag{A.13}$$

where $\partial\Omega$ are all the boundaries which are formed by two: the Farfield Γ_1 and the airfoil surface Γ_2. Remember that all integrals related to velocity and belonging to airfoil surface boundary Γ_2 are zero, since velocities there are orthogonal to $\mathbf{n}$ vector. Finally, find here all the terms of the Dynamical System with its respective boundary terms:

$$\begin{aligned}
\int_\Omega \frac{\partial \rho}{\partial t}\Psi_i^\rho + \int_\Omega \frac{\partial m_1}{\partial t}\Psi_i^{m_1} + \int_\Omega \frac{\partial m_2}{\partial t}\Psi_i^{m_2} + \int_\Omega \frac{\partial E}{\partial t}\Psi_i^E + \\
- \int_\Omega \begin{pmatrix} m_1 \\ m_2 \end{pmatrix} \cdot \nabla \Psi_i^\rho - \int_\Omega \begin{pmatrix} m_1 u \\ m_1 v \end{pmatrix} \cdot \nabla \Psi_i^{m_1} - \int_\Omega p \cdot \frac{\partial}{\partial x}\Psi_i^{m_1} - \\
\int_\Omega \begin{pmatrix} m_2 u \\ m_2 v \end{pmatrix} \cdot \nabla \Psi_i^{m_2} - \int_\Omega p \cdot \frac{\partial}{\partial y}\Psi_i^{m_2} - \int_\Omega \begin{pmatrix} (E+p)\,u \\ (E+p)\,v \end{pmatrix} \cdot \nabla \Psi_i^E + \\
\int_{\Gamma_1} \left(\begin{pmatrix} m_1 \\ m_2 \end{pmatrix} \mathbf{n} \right) \Psi_i^\rho + \int_{\Gamma_1} \left(\begin{pmatrix} m_1 u \\ m_1 v \end{pmatrix} \mathbf{n} \right) \Psi_i^{m_1} + \int_{\Gamma_1} p\,\mathbf{n}_x \Psi_i^{m_1} + \\
\int_{\Gamma_1} \left(\begin{pmatrix} m_2 u \\ m_2 v \end{pmatrix} \mathbf{n} \right) \Psi_i^{m_2} + \int_{\Gamma_1} p\,\mathbf{n}_y \Psi_i^{m_2} + \int_{\Gamma_1} \left(\begin{pmatrix} (E+p)\,u \\ (E+p)\,v \end{pmatrix} \mathbf{n} \right) \Psi_i^E + \\
\int_{\Gamma_2} p\,\mathbf{n}_x \Psi_i^{m_1} + \int_{\Gamma_2} p\,\mathbf{n}_y \Psi_i^{m_2} = 0
\end{aligned} \tag{A.14}$$

Now we introduce the separation of variables by POD modes $\Psi_i(x)$for each of the conservative variables:

$$\begin{aligned}
\rho(x,t) &= \bar{\rho} + \sum_{j=1}^{N} y_j(t)\Psi_j^{\rho}(x) \\
m_1(x,t) &= \bar{m_1} + \sum_{j=1}^{N} y_j(t)\Psi_j^{m_1}(x) \\
m_2(x,t) &= \bar{m_2} + \sum_{j=1}^{N} y_j(t)\Psi_j^{m_2}(x) \\
E(x,t) &= \bar{E} + \sum_{j=1}^{N} y_j(t)\Psi_j^{E}(x)
\end{aligned} \tag{A.15}$$

where N is the number of modes of the Reduced Basis, $\bar{\rho}, \bar{m_1}, \bar{m_2}, \bar{E}$ are the average values and $\Psi_j^{\rho}(x), \Psi_j^{m_1}(x), \Psi_j^{m_2}(x), \Psi_j^{E}(x)$ are the modes separated for each of the conservative variables respectively. Notice that in this formulation just one $y_j(t)$ has stated common to all the variables.

Find below the development for each of the terms fo equation A.14 separately:

$$\int_{\Omega} \frac{\partial \rho}{\partial t}\Psi_i^{\rho} = \int_{\Omega} \frac{\partial}{\partial t}\left(\bar{\rho} + \sum_{j=1}^{N} y_j \Psi_j^{\rho}\right)\Psi_i^{\rho} = \frac{\partial}{\partial t}\sum_{j=1}^{N} y_j \int_{\Omega} \Psi_j^{\rho}\Psi_i^{\rho} = \frac{\partial}{\partial t} y_i \tag{A.16}$$

since

$$\frac{\partial \bar{\rho}}{\partial t} = 0 \tag{A.17}$$

and

$$\int_{\Omega} \Psi_i \Psi_j = \delta_{ij} \tag{A.18}$$

where

$$\delta_{ij} = \begin{cases} 0 & fori \neq j \\ 1 & fori = j \end{cases} \tag{A.19}$$

is the Kroenecker delta symbol. Next three terms of (A.14) are developed in a similar way. Shortly:

$$\int_{\Omega} \frac{\partial m_1}{\partial t}\Psi_i^{m_1} = \frac{\partial}{\partial t} y_i \tag{A.20}$$

$$\int_{\Omega} \frac{\partial m_2}{\partial t}\Psi_i^{m_2} = \frac{\partial}{\partial t} y_i \tag{A.21}$$

$$\int_{\Omega} \frac{\partial E}{\partial t}\Psi_i^{E} = \frac{\partial}{\partial t} y_i \tag{A.22}$$

Fifth term of (A.14):

$$\begin{aligned}
\int_\Omega \begin{pmatrix} m_1 \\ m_2 \end{pmatrix} \cdot \nabla \Psi_i^\rho = \int_\Omega m_1 \frac{\partial}{\partial x}\Psi_i^\rho + \int_\Omega m_2 \frac{\partial}{\partial y}\Psi_i^\rho = \\
\int_\Omega \left(\bar{m_1} + \sum_{j=1}^{N} y_j \Psi_j^{m_1} \right) \frac{\partial}{\partial x}\Psi_i^\rho + \int_\Omega \left(\bar{m_2} + \sum_{j=1}^{N} y_j \Psi_j^{m_2} \right) \frac{\partial}{\partial y}\Psi_i^\rho = \\
\sum_{j=1}^{N} y_j \left(\int_\Omega \Psi_j^{m_1} \frac{\partial}{\partial x}\Psi_i^\rho + \int_\Omega \Psi_j^{m_2} \frac{\partial}{\partial y}\Psi_i^\rho \right) + \int_\Omega \bar{m_1} \frac{\partial}{\partial x}\Psi_i^\rho + \int_\Omega \bar{m_2} \frac{\partial}{\partial y}\Psi_i^\rho
\end{aligned} \tag{A.23}$$

In a similar way are developed the next terms. Remember that $u = \frac{m_1}{\rho}$ and $v = \frac{m_2}{\rho}$, and pressure p is related to the conservative flow variables, ρ, ρu, ρv and E, by the equation of state A.5. Just take in to account that for the sake of simplicity, variables appearing in any denominator have not been expanded by equations A.15 but approximated by its average values $\bar{\rho}, \bar{m_1}, \bar{m_2}$ and $\bar{E}$ respectively.

$$\begin{aligned}
\int_\Omega \begin{pmatrix} m_1 u \\ m_1 v \end{pmatrix} \cdot \nabla \Psi_i^{m_1} = \int_\Omega m_1\, u \frac{\partial}{\partial x}\Psi_i^{m_1} + \int_\Omega m_1\, v \frac{\partial}{\partial y}\Psi_i^{m_1} = \\
\int_\Omega \frac{m_1^2}{\bar{\rho}} \frac{\partial}{\partial x}\Psi_i^{m_1} + \int_\Omega \frac{m_1 m_2}{\bar{\rho}} \frac{\partial}{\partial y}\Psi_i^{m_1} = \\
\int_\Omega \frac{\left(\bar{m_1} + \sum_{j=1}^{N} y_j \Psi_j^{m_1} \right)^2}{\bar{\rho}} \frac{\partial}{\partial x}\Psi_i^{m_1} + \\
\int_\Omega \frac{\left(\bar{m_1} + \sum_{j=1}^{N} y_j \Psi_j^{m_1} \right) \left(\bar{m_2} + \sum_{j=1}^{N} y_j \Psi_j^{m_2} \right)}{\bar{\rho}} \frac{\partial}{\partial y}\Psi_i^{m_1} = \\
\sum_{j,k=1}^{N} y_j y_k \left(\int_\Omega \frac{1}{\bar{\rho}} \Psi_j^{m_1} \Psi_k^{m_1} \frac{\partial}{\partial x}\Psi_i^{m_1} + \int_\Omega \frac{1}{\bar{\rho}} \Psi_j^{m_1} \Psi_k^{m_2} \frac{\partial}{\partial y}\Psi_i^{m_1} \right) + \\
\sum_{j=1}^{N} y_j \left(\int_\Omega \frac{2\bar{m_1}}{\bar{\rho}} \Psi_j^{m_1} \frac{\partial}{\partial x}\Psi_i^{m_1} + \int_\Omega \frac{\bar{m_1}}{\bar{\rho}} \Psi_j^{m_2} \frac{\partial}{\partial y}\Psi_i^{m_1} + \int_\Omega \frac{\bar{m_2}}{\bar{\rho}} \Psi_j^{m_1} \frac{\partial}{\partial y}\Psi_i^{m_1} \right) + \\
\int_\Omega \frac{\bar{m_1^2}}{\bar{\rho}} \frac{\partial}{\partial x}\Psi_i^{m_1} + \int_\Omega \frac{\bar{m_1}\bar{m_2}}{\bar{\rho}} \frac{\partial}{\partial y}\Psi_i^{m_1}
\end{aligned} \tag{A.24}$$

$$\begin{aligned}
\int_\Omega p \cdot \frac{\partial}{\partial x}\Psi_i^{m_1} = \cdots = \\
0.4 \Bigg[\sum_{j,k=1}^{N} y_j y_k \left((-) \int_\Omega \frac{1}{2\bar{\rho}} \Psi_j^{m_1} \Psi_k^{m_1} \frac{\partial}{\partial x}\Psi_i^{m_1} - \int_\Omega \frac{1}{2\bar{\rho}} \Psi_j^{m_2} \Psi_k^{m_2} \frac{\partial}{\partial x}\Psi_i^{m_1} \right) + \\
\sum_{j=1}^{N} y_j \left(\int_\Omega \Psi_j^E \frac{\partial}{\partial x}\Psi_i^{m_1} - \int_\Omega \frac{\bar{m_1}}{\bar{\rho}} \Psi_j^{m_1} \frac{\partial}{\partial x}\Psi_i^{m_1} - \int_\Omega \frac{\bar{m_2}}{\bar{\rho}} \Psi_j^{m_2} \frac{\partial}{\partial x}\Psi_i^{m_1} \right) + \\
\int_\Omega \bar{E} \frac{\partial}{\partial x}\Psi_i^{m_1} - \int_\Omega \frac{\bar{m_1}^2}{2\bar{\rho}} \frac{\partial}{\partial x}\Psi_i^{m_1} - \int_\Omega \frac{\bar{m_2}^2}{2\bar{\rho}} \frac{\partial}{\partial x}\Psi_i^{m_1} \Bigg]
\end{aligned} \tag{A.25}$$

$$
\begin{aligned}
\int_\Omega \begin{pmatrix} m_2 u \\ m_2 v \end{pmatrix} \cdot \nabla \Psi_i^{m_2} = \cdots = \\
\sum_{j,k=1}^{N} y_j y_k \left(\int_\Omega \frac{1}{\bar{\rho}} \Psi_j^{m_1} \Psi_k^{m_2} \frac{\partial}{\partial x} \Psi_i^{m_2} + \int_\Omega \frac{1}{\bar{\rho}} \Psi_j^{m_2} \Psi_k^{m_2} \frac{\partial}{\partial y} \Psi_i^{m_2} \right) + \\
\sum_{j=1}^{N} y_j \left(\int_\Omega \frac{\bar{m_1}}{\bar{\rho}} \Psi_j^{m_2} \frac{\partial}{\partial x} \Psi_i^{m_2} + \int_\Omega \frac{\bar{m_2}}{\bar{\rho}} \Psi_j^{m_1} \frac{\partial}{\partial x} \Psi_i^{m_2} + \int_\Omega \frac{2\bar{m}_2}{\bar{\rho}} \Psi_j^{m_2} \frac{\partial}{\partial y} \Psi_i^{m_2} \right) + \\
\int_\Omega \frac{\bar{m_1} \bar{m_2}}{\bar{\rho}} \frac{\partial}{\partial x} \Psi_i^{m_2} + \int_\Omega \frac{\bar{m_2}^2}{\bar{\rho}} \frac{\partial}{\partial y} \Psi_i^{m_2}
\end{aligned}
\tag{A.26}
$$

$$
\begin{aligned}
\int_\Omega p \cdot \frac{\partial}{\partial y} \Psi_i^{m_2} = \cdots = \\
0.4 \left[\sum_{j,k=1}^{N} y_j y_k \left(- \int_\Omega \frac{1}{2\bar{\rho}} \Psi_j^{m_1} \Psi_k^{m_1} \frac{\partial}{\partial y} \Psi_i^{m_2} - \int_\Omega \frac{1}{2\bar{\rho}} \Psi_j^{m_2} \Psi_k^{m_2} \frac{\partial}{\partial y} \Psi_i^{m_2} \right) + \right. \\
\sum_{j=1}^{N} y_j \left(\int_\Omega \Psi_j^E \frac{\partial}{\partial y} \Psi_i^{m_2} - \int_\Omega \frac{\bar{m_1}}{\bar{\rho}} \Psi_j^{m_1} \frac{\partial}{\partial y} \Psi_i^{m_2} - \int_\Omega \frac{\bar{m_2}}{\bar{\rho}} \Psi_j^{m_2} \frac{\partial}{\partial y} \Psi_i^{m_2} \right) + \\
\left. \int_\Omega \bar{E} \frac{\partial}{\partial y} \Psi_i^{m_2} - \int_\Omega \frac{\bar{m_2}^2}{2\bar{\rho}} \frac{\partial}{\partial y} \Psi_i^{m_2} - \int_\Omega \frac{\bar{m_1}^2}{2\bar{\rho}} \frac{\partial}{\partial y} \Psi_i^{m_2} \right]
\end{aligned}
\tag{A.27}
$$

$$
\begin{aligned}
\int_\Omega \begin{pmatrix}(E+p)\,u\\(E+p)\,v\end{pmatrix}\cdot\nabla\Psi_i^E = \cdots =\\
\sum_{j,k=1}^{N} y_j y_k \Bigg(-\int_\Omega \frac{0.2\bar{m}_1}{\bar{\rho}^2}\Psi_j^{m_1}\Psi_k^{m_1}\frac{\partial}{\partial x}\Psi_i^E - \int_\Omega \frac{0.2\bar{m}_1}{\bar{\rho}^2}\Psi_j^{m_2}\Psi_k^{m_2}\frac{\partial}{\partial x}\Psi_i^E -\\
\int_\Omega \frac{0.4\bar{m}_1}{\bar{\rho}^2}\Psi_j^{m_1}\Psi_k^{m_1}\frac{\partial}{\partial x}\Psi_i^E - \int_\Omega \frac{0.4\bar{m}_2}{\bar{\rho}^2}\Psi_j^{m_2}\Psi_k^{m_1}\frac{\partial}{\partial x}\Psi_i^E -\\
\int_\Omega \frac{0.2\bar{m}_2}{\bar{\rho}^2}\Psi_j^{m_1}\Psi_k^{m_1}\frac{\partial}{\partial y}\Psi_i^E - \int_\Omega \frac{0.2\bar{m}_2}{\bar{\rho}^2}\Psi_j^{m_2}\Psi_k^{m_2}\frac{\partial}{\partial y}\Psi_i^E -\\
\int_\Omega \frac{0.4\bar{m}_1}{\bar{\rho}^2}\Psi_j^{m_1}\Psi_k^{m_2}\frac{\partial}{\partial y}\Psi_i^E - \int_\Omega \frac{0.4\bar{m}_2}{\bar{\rho}^2}\Psi_j^{m_2}\Psi_k^{m_2}\frac{\partial}{\partial y}\Psi_i^E +\\
\int_\Omega \frac{1.4}{\bar{\rho}}\Psi_j^E\Psi_k^{m_1}\frac{\partial}{\partial x}\Psi_i^E + \int_\Omega \frac{1.4}{\bar{\rho}}\Psi_j^E\Psi_k^{m_2}\frac{\partial}{\partial y}\Psi_i^E\Bigg) +\\
\sum_{j=1}^{N} y_j \Bigg(\int_\Omega \frac{1.4}{\bar{\rho}}\Psi_j^E\bar{m}_1\frac{\partial}{\partial x}\Psi_i^E - \int_\Omega \frac{0.4}{\bar{\rho}^2}{\bar{m}_1}^2\Psi_j^{m_1}\frac{\partial}{\partial x}\Psi_i^E -\\
\int_\Omega \frac{0.4}{\bar{\rho}^2}\bar{m}_2\bar{m}_1\Psi_j^{m_2}\frac{\partial}{\partial x}\Psi_i^E - \int_\Omega \frac{0.2}{\bar{\rho}^2}{\bar{m}_1}^2\Psi_j^{m_1}\frac{\partial}{\partial x}\Psi_i^E -\\
\int_\Omega \frac{0.2}{\bar{\rho}^2}{\bar{m}_2}^2\Psi_j^{m_1}\frac{\partial}{\partial x}\Psi_i^E + \int_\Omega \frac{1.4}{\bar{\rho}}\bar{m}_2\Psi_j^E\frac{\partial}{\partial y}\Psi_i^E -\\
\int_\Omega \frac{0.4}{\bar{\rho}^2}\bar{m}_2\bar{m}_1\Psi_j^{m_1}\frac{\partial}{\partial y}\Psi_i^E - \int_\Omega \frac{0.4}{\bar{\rho}^2}{\bar{m}_2}^2\Psi_j^{m_2}\frac{\partial}{\partial y}\Psi_i^E -\\
\int_\Omega \frac{0.2}{\bar{\rho}^2}{\bar{m}_1}^2\Psi_j^{m_2}\frac{\partial}{\partial y}\Psi_i^E - \int_\Omega \frac{0.2}{\bar{\rho}^2}{\bar{m}_2}^2\Psi_j^{m_2}\frac{\partial}{\partial y}\Psi_i^E +\\
\int_\Omega \frac{1.4}{\bar{\rho}}\bar{E}\Psi_j^{m_1}\frac{\partial}{\partial x}\Psi_i^E + \int_\Omega \frac{1.4}{\bar{\rho}}\bar{E}\Psi_j^{m_2}\frac{\partial}{\partial y}\Psi_i^E\Bigg) +\\
\int_\Omega \frac{1.4}{\bar{\rho}}\bar{E}\bar{m}_1\frac{\partial}{\partial x}\Psi_i^E - \int_\Omega \frac{0.2}{\bar{\rho}^2}{\bar{m}_1}^3\frac{\partial}{\partial x}\Psi_i^E -\\
\int_\Omega \frac{0.2}{\bar{\rho}^2}{\bar{m}_2}^2\bar{m}_1\frac{\partial}{\partial x}\Psi_i^E + \int_\Omega \frac{1.4}{\bar{\rho}}\bar{E}\bar{m}_2\frac{\partial}{\partial y}\Psi_i^E -\\
\int_\Omega \frac{0.2}{\bar{\rho}^2}{\bar{m}_2}^3\frac{\partial}{\partial y}\Psi_i^E - \int_\Omega \frac{0.2}{\bar{\rho}^2}{\bar{m}_1}^2\bar{m}_2\frac{\partial}{\partial y}\Psi_i^E
\end{aligned}
\tag{A.28}
$$

Integrals over boundaries are treated next. First find the only two terms related to the airfoil boundary Γ_2:

$$
\begin{aligned}
&\int_{\Gamma_2} p\mathbf{n}_x\Psi_i^{m_1} = \cdots = \\
0.4\Bigg[&\sum_{j,k=1}^{N} y_j y_k \left((-)\int_{\Gamma_2} \frac{1}{2\bar{\rho}}\Psi_j^{m_1}\Psi_k^{m_1}\mathbf{n}_x\Psi_i^{m_1} - \int_{\Gamma_2} \frac{1}{2\bar{\rho}}\Psi_j^{m_2}\Psi_k^{m_2}\mathbf{n}_x\Psi_i^{m_1} \right) + \\
&\sum_{j=1}^{N} y_j \left(\int_{\Gamma_2} \Psi_j^{E}\mathbf{n}_x\Psi_i^{m_1} - \int_{\Gamma_2} \frac{\bar{m_1}}{\bar{\rho}}\Psi_j^{m_1}\mathbf{n}_x\Psi_i^{m_1} - \int_{\Gamma_2} \frac{\bar{m_2}}{\bar{\rho}}\Psi_j^{m_2}\mathbf{n}_x\Psi_i^{m_1} \right) + \\
&\int_{\Gamma_2} \bar{E}\mathbf{n}_x\Psi_i^{m_1} - \int_{\Gamma_2} \frac{\bar{m_1}^2}{2\bar{\rho}}\mathbf{n}_x\Psi_i^{m_1} - \int_{\Gamma_2} \frac{\bar{m_2}^2}{2\bar{\rho}}\mathbf{n}_x\Psi_i^{m_1} \Bigg]
\end{aligned}
\tag{A.29}
$$

$$
\begin{aligned}
&\int_{\Gamma_2} p\mathbf{n}_y\Psi_i^{m_2} = \cdots = \\
0.4\Bigg[&\sum_{j,k=1}^{N} y_j y_k \left((-)\int_{\Gamma_2} \frac{1}{2\bar{\rho}}\Psi_j^{m_1}\Psi_k^{m_1}\mathbf{n}_y\Psi_i^{m_2} - \int_{\Gamma_2} \frac{1}{2\bar{\rho}}\Psi_j^{m_2}\Psi_k^{m_2}\mathbf{n}_y\Psi_i^{m_2} \right) + \\
&\sum_{j=1}^{N} y_j \left(\int_{\Gamma_2} \Psi_j^{E}\mathbf{n}_y\Psi_i^{m_2} - \int_{\Gamma_2} \frac{\bar{m_1}}{\bar{\rho}}\Psi_j^{m_1}\mathbf{n}_y\Psi_i^{m_2} - \int_{\Gamma_2} \frac{\bar{m_2}}{\bar{\rho}}\Psi_j^{m_2}\mathbf{n}_y\Psi_i^{m_2} \right) + \\
&\int_{\Gamma_2} \bar{E}\mathbf{n}_y\Psi_i^{m_2} - \int_{\Gamma_2} \frac{\bar{m_1}^2}{2\bar{\rho}}\mathbf{n}_y\Psi_i^{m_2} - \int_{\Gamma_2} \frac{\bar{m_2}^2}{2\bar{\rho}}\mathbf{n}_y\Psi_i^{m_2} \Bigg]
\end{aligned}
\tag{A.30}
$$

And all integrals over Farfield boundary Γ_1,

$$
\begin{aligned}
&\int_{\Gamma_1} \left(\begin{pmatrix} m_1 \\ m_2 \end{pmatrix} \mathbf{n} \right) \Psi_i^{\rho} + \int_{\Gamma_1} \left(\begin{pmatrix} m_1 u \\ m_1 v \end{pmatrix} \mathbf{n} \right) \Psi_i^{m_1} + \int_{\Gamma_1} p\,\mathbf{n}_x\Psi_i^{m_1} + \\
&\int_{\Gamma_1} \left(\begin{pmatrix} m_2 u \\ m_2 v \end{pmatrix} \mathbf{n} \right) \Psi_i^{m_2} + \int_{\Gamma_1} p\,\mathbf{n}_y\Psi_i^{m_2} + \int_{\Gamma_1} \left(\begin{pmatrix} (E+p)\,u \\ (E+p)\,v \end{pmatrix} \mathbf{n} \right) \Psi_i^{E} = \\
&\int_{\Gamma_1} m_1\mathbf{n}_x\Psi_i^{\rho} + \int_{\Gamma_1} m_2\mathbf{n}_y\Psi_i^{\rho} + \int_{\Gamma_1} \frac{m_1^2}{\bar{\rho}}\mathbf{n}_x\Psi_i^{m_1} + \\
&\int_{\Gamma_1} \frac{m_1 m_2}{\bar{\rho}}\mathbf{n}_y\Psi_i^{m_1} + \int_{\Gamma_1} p\,\mathbf{n}_x\Psi_i^{m_1} + \int_{\Gamma_1} \frac{m_1 m_2}{\bar{\rho}}\mathbf{n}_x\Psi_i^{m_2} + \\
&\int_{\Gamma_1} \frac{m_2^2}{\bar{\rho}}\mathbf{n}_y\Psi_i^{m_2} + \int_{\Gamma_1} p\,\mathbf{n}_y\Psi_i^{m_2} + \\
&\int_{\Gamma_1} \frac{(E+p)\,m_1}{\bar{\rho}}\mathbf{n}_x\Psi_i^{E} + \int_{\Gamma_1} \frac{(E+p)\,m_2}{\bar{\rho}}\mathbf{n}_y\Psi_i^{E}
\end{aligned}
\tag{A.31}
$$

All terms of system of equations A.14 have been developped. Now, all of them can be merged so the general ODE system can be written as follows:

$$
\left(4\frac{\partial y_i}{\partial t} + a_i \right) + \sum_{j=1}^{N} b_{i,j}\, y_j + \sum_{j,k=1}^{N} c_{i,j,k}\, y_j y_k = 0 \tag{A.32}
$$

that yields to the Dynamical System to solve and find function y_i:

$$\frac{\partial y_i}{\partial t} = \frac{1}{4}\left(-a_i - \sum_{j=1}^{N} b_{i,j}\, y_j - \sum_{j,k=1}^{N} c_{i,j,k}\, y_j y_k\right) \tag{A.33}$$

where in next lines can be found the terms $a_i, b_{i,j}$ and $c_{i,j,k}$:

$$\begin{aligned}
a_i = &- \int_\Omega \bar{m_1}\frac{\partial \Psi_i^\rho}{\partial x} - \int_\Omega \bar{m_2}\frac{\partial \Psi_i^\rho}{\partial y} - \int_\Omega \frac{\bar{m_1}^2}{r\bar{h}o}\frac{\partial \Psi_i^{m_1}}{\partial x} - \\
&\int_\Omega \frac{\bar{m_1}\bar{m_2}}{r\bar{h}o}\frac{\partial \Psi_i^{m_1}}{\partial y} - 0.4\int_\Omega \bar{E}\frac{\partial \Psi_i^{m_1}}{\partial x}\Psi_i^{m_1} + 0.2\int_\Omega \frac{\bar{m_1}^2}{\bar{\rho}}\frac{\partial \Psi_i^{m_1}}{\partial x} + \\
&0.2\int_\Omega \frac{\bar{m_2}^2}{\bar{\rho}}\frac{\partial \Psi_i^{m_1}}{\partial x} - \int_\Omega \frac{\bar{m_1}\bar{m_2}}{\bar{\rho}}\frac{\partial \Psi_i^{m_2}}{\partial x}\Psi_i^{m_2} - \int_\Omega \frac{\bar{m_2}^2}{\bar{\rho}}\frac{\partial \Psi_i^{m_2}}{\partial y} - \\
&0.4\int_\Omega \bar{E}\frac{\partial \Psi_i^{m_2}}{\partial y} + 0.2\int_\Omega \frac{\bar{m_2}^2}{\bar{\rho}}\frac{\partial \Psi_i^{m_2}}{\partial y} + 0.2\int_\Omega \frac{\bar{m_1}^2}{\bar{\rho}}\frac{\partial \Psi_i^{m_2}}{\partial y} - \\
&1.4\int_\Omega \frac{\bar{E}\bar{m_1}}{\bar{\rho}}\frac{\partial \Psi_i^{E}}{\partial x} + 0.2\int_\Omega \frac{\bar{m_1}^3}{\bar{\rho}^2}\frac{\partial \Psi_i^{E}}{\partial x} + 0.2\int_\Omega \frac{\bar{m_2}^2\bar{m_1}}{\bar{\rho}^2}\frac{\partial \Psi_i^{E}}{\partial x} - \\
&1.4\int_\Omega \frac{\bar{E}\bar{m_2}}{\bar{\rho}}\frac{\partial \Psi_i^{E}}{\partial y} + 0.2\int_\Omega \frac{\bar{m_2}^3}{\bar{\rho}^2}\frac{\partial \Psi_i^{E}}{\partial y} + 0.2\int_\Omega \frac{\bar{m_1}^2\bar{m_2}}{\bar{\rho}^2}\frac{\partial \Psi_i^{E}}{\partial y} + \\
&0.4\int_{\Gamma_2} \bar{E}\mathbf{n}_x\Psi_i^{m_1} - 0.2\int_{\Gamma_2} \frac{\bar{m_1}^2}{2\bar{\rho}}\mathbf{n}_x\Psi_i^{m_1} - 0.2\int_{\Gamma_2} \frac{\bar{m_2}^2}{2\bar{\rho}}\mathbf{n}_x\Psi_i^{m_1} + \\
&0.4\int_{\Gamma_2} \bar{E}\mathbf{n}_y\Psi_i^{m_2} - 0.2\int_{\Gamma_2} \frac{\bar{m_1}^2}{2\bar{\rho}}\mathbf{n}_y\Psi_i^{m_2} - 0.2\int_{\Gamma_2} \frac{\bar{m_2}^2}{2\bar{\rho}}\mathbf{n}_y\Psi_i^{m_2} + \\
&\int_{\Gamma_1} m_1\mathbf{n}_x\Psi_i^{\rho} + \int_{\Gamma_1} m_2\mathbf{n}_y\Psi_i^{\rho} + \int_{\Gamma_1} \frac{m_1^2}{\bar{\rho}}\mathbf{n}_x\Psi_i^{m_1} + \int_{\Gamma_1} \frac{m_1 m_2}{\bar{\rho}}\mathbf{n}_y\Psi_i^{m_1} + \\
&\int_{\Gamma_1} p\,\mathbf{n}_x\Psi_i^{m_1} + \int_{\Gamma_1} \frac{m_1 m_2}{\bar{\rho}}\mathbf{n}_x\Psi_i^{m_2} + \int_{\Gamma_1} \frac{m_2^2}{\bar{\rho}}\mathbf{n}_y\Psi_i^{m_2} + \int_{\Gamma_1} p\,\mathbf{n}_y\Psi_i^{m_2} + \\
&\int_{\Gamma_1} \frac{(E+p)\,m_1}{\bar{\rho}}\mathbf{n}_x\Psi_i^{E} + \int_{\Gamma_1} \frac{(E+p)\,m_2}{\bar{\rho}}\mathbf{n}_y\Psi_i^{E}
\end{aligned} \tag{A.34}$$

$$
\begin{aligned}
b_{i,j} = &- \int_\Omega \Psi_j^{m_1} \frac{\partial \Psi_i^\rho}{\partial x} - \int_\Omega \Psi_j^{m_2} \frac{\partial \Psi_i^\rho}{\partial y} - 2 \int_\Omega \frac{\bar{m_1}}{\bar{\rho}} \Psi_j^{m_1} \frac{\partial \Psi_i^{m_1}}{\partial x} - \\
&\int_\Omega \frac{\bar{m_1}}{\bar{\rho}} \Psi_j^{m_2} \frac{\partial \Psi_i^{m_1}}{\partial y} - \int_\Omega \frac{\bar{m_2}}{\bar{\rho}} \Psi_j^{m_1} \frac{\partial \Psi_i^{m_1}}{\partial y} - 0.4 \int_\Omega \Psi_j^E \frac{\partial \Psi_i^{m_1}}{\partial x} + \\
&0.4 \int_\Omega \frac{\bar{m_1}}{\bar{\rho}} \Psi_j^{m_1} \frac{\partial \Psi_i^{m_1}}{\partial x} + 0.4 \int_\Omega \frac{\bar{m_2}}{\bar{\rho}} \Psi_j^{m_2} \frac{\partial \Psi_i^{m_1}}{\partial x} - \int_\Omega \frac{\bar{m_1}}{\bar{\rho}} \Psi_j^{m_2} \frac{\partial \Psi_i^{m_2}}{\partial x} - \\
&\int_\Omega \frac{\bar{m_2}}{\bar{\rho}} \Psi_j^{m_1} \frac{\partial \Psi_i^{m_2}}{\partial x} - 2 \int_\Omega \frac{\bar{m_2}}{\bar{\rho}} \Psi_j^{m_2} \frac{\partial \Psi_i^{m_2}}{\partial y} - 0.4 \int_\Omega \Psi_j^E \frac{\partial \Psi_i^{m_2}}{\partial y} + \\
&0.4 \int_\Omega \frac{\bar{m_1}}{\bar{\rho}} \Psi_j^{m_1} \frac{\partial \Psi_i^{m_2}}{\partial y} + 0.4 \int_\Omega \frac{\bar{m_2}}{\bar{\rho}} \Psi_j^{m_2} \frac{\partial \Psi_i^{m_2}}{\partial y} - 1.4 \int_\Omega \frac{\bar{m_1}}{\bar{\rho}} \Psi_j^E \frac{\partial \Psi_i^E}{\partial x} + \\
&0.6 \int_\Omega \frac{{\bar{m_1}}^2}{\bar{\rho}^2} \Psi_j^{m_1} \frac{\partial \Psi_i^E}{\partial x} + 0.4 \int_\Omega \frac{\bar{m_2} \bar{m_1}}{\bar{\rho}^2} \Psi_j^{m_2} \frac{\partial \Psi_i^E}{\partial x} + 0.2 \int_\Omega \frac{{\bar{m_2}}^2}{\bar{\rho}^2} \Psi_j^{m_1} \frac{\partial \Psi_i^E}{\partial x} - \\
&1.4 \int_\Omega \frac{\bar{m_2}}{\bar{\rho}} \Psi_j^E \frac{\partial \Psi_i^E}{\partial y} + 0.4 \int_\Omega \frac{\bar{m_2} \bar{m_1}}{\bar{\rho}^2} \Psi_j^{m_1} \frac{\partial \Psi_i^E}{\partial y} + 0.6 \int_\Omega \frac{{\bar{m_2}}^2}{\bar{\rho}^2} \Psi_j^{m_2} \frac{\partial \Psi_i^E}{\partial y} + \\
&0.2 \int_\Omega \frac{{\bar{m_1}}^2}{\bar{\rho}^2} \Psi_j^{m_2} \frac{\partial \Psi_i^E}{\partial y} - 1.4 \int_\Omega \frac{\bar{E}}{\bar{\rho}} \Psi_j^{m_1} \frac{\partial \Psi_i^E}{\partial x} - 1.4 \int_\Omega \frac{\bar{E}}{\bar{\rho}} \Psi_j^{m_2} \frac{\partial \Psi_i^E}{\partial y} + \\
&0.4 \int_{\Gamma_2} \Psi_j^E \mathbf{n}_x \Psi_i^{m_1} - 0.4 \int_{\Gamma_2} \frac{\bar{m_1}}{\bar{\rho}} \Psi_j^{m_1} \mathbf{n}_x \Psi_i^{m_1} - 0.4 \int_{\Gamma_2} \frac{\bar{m_2}}{\bar{\rho}} \Psi_j^{m_2} \mathbf{n}_x \Psi_i^{m_1} + \\
&0.4 \int_{\Gamma_2} \Psi_j^E \mathbf{n}_y \Psi_i^{m_2} - 0.4 \int_{\Gamma_2} \frac{\bar{m_1}}{\bar{\rho}} \Psi_j^{m_1} \mathbf{n}_y \Psi_i^{m_2} - 0.4 \int_{\Gamma_2} \frac{\bar{m_2}}{\bar{\rho}} \Psi_j^{m_2} \mathbf{n}_y \Psi_i^{m_2}
\end{aligned}
\tag{A.35}
$$

$$
\begin{aligned}
c_{i,j,k} = &- \int_\Omega \frac{1}{\bar{\rho}} \Psi_j^{m_1} \Psi_k^{m_1} \frac{\partial \Psi_i^{m_1}}{\partial x} - \int_\Omega \frac{1}{\bar{\rho}} \Psi_j^{m_1} \Psi_k^{m_2} \frac{\partial \Psi_i^{m_1}}{\partial y} + \\
&0.2 \int_\Omega \frac{1}{\bar{\rho}} \Psi_j^{m_1} \Psi_k^{m_1} \frac{\partial \Psi_i^{m_1}}{\partial x} + 0.2 \int_\Omega \frac{1}{\bar{\rho}} \Psi_j^{m_2} \Psi_k^{m_2} \frac{\partial \Psi_i^{m_1}}{\partial x} - \\
&\int_\Omega \frac{1}{\bar{\rho}} \Psi_j^{m_1} \Psi_k^{m_2} \frac{\partial \Psi_i^{m_2}}{\partial x} - \int_\Omega \frac{1}{\bar{\rho}} \Psi_j^{m_2} \Psi_k^{m_2} \frac{\partial \Psi_i^{m_2}}{\partial y} + \\
&0.2 \int_\Omega \frac{1}{\bar{\rho}} \Psi_j^{m_1} \Psi_k^{m_1} \frac{\partial \Psi_i^{m_2}}{\partial y} + 0.2 \int_\Omega \frac{1}{\bar{\rho}} \Psi_j^{m_2} \Psi_k^{m_2} \frac{\partial \Psi_i^{m_2}}{\partial y} + \\
&0.6 \int_\Omega \frac{\bar{m_1}}{\bar{\rho}^2} \Psi_j^{m_1} \Psi_k^{m_1} \frac{\partial \Psi_i^E}{\partial x} + 0.2 \int_\Omega \frac{\bar{m_1}}{\bar{\rho}^2} \Psi_j^{m_2} \Psi_k^{m_2} \frac{\partial \Psi_i^E}{\partial x} + \\
&0.4 \int_\Omega \frac{\bar{m_2}}{\bar{\rho}^2} \Psi_j^{m_2} \Psi_k^{m_1} \frac{\partial \Psi_i^E}{\partial x} + 0.2 \int_\Omega \frac{\bar{m_2}}{\bar{\rho}^2} \Psi_j^{m_1} \Psi_k^{m_1} \frac{\partial \Psi_i^E}{\partial y} + \\
&0.6 \int_\Omega \frac{\bar{m_2}}{\bar{\rho}^2} \Psi_j^{m_2} \Psi_k^{m_2} \frac{\partial \Psi_i^E}{\partial y} + 0.4 \int_\Omega \frac{\bar{m_1}}{\bar{\rho}^2} \Psi_j^{m_1} \Psi_k^{m_2} \frac{\partial \Psi_i^E}{\partial y} - \\
&1.4 \int_\Omega \frac{1}{\bar{\rho}} \Psi_j^E \Psi_k^{m_1} \frac{\partial \Psi_i^E}{\partial x} - 1.4 \int_\Omega \frac{1}{\bar{\rho}} \Psi_j^E \Psi_k^{m_2} \frac{\partial \Psi_i^E}{\partial y} - \\
&0.2 \int_{\Gamma_2} \frac{1}{\bar{\rho}} \Psi_j^{m_1} \Psi_k^{m_1} \mathbf{n}_x \Psi_i^{m_1} - 0.2 \int_{\Gamma_2} \frac{1}{\bar{\rho}} \Psi_j^{m_2} \Psi_k^{m_2} \mathbf{n}_x \Psi_i^{m_1} - \\
&0.2 \int_{\Gamma_2} \frac{1}{\bar{\rho}} \Psi_j^{m_1} \Psi_k^{m_1} \mathbf{n}_y \Psi_i^{m_2} - 0.2 \int_{\Gamma_2} \frac{1}{\bar{\rho}} \Psi_j^{m_2} \Psi_k^{m_2} \mathbf{n}_y \Psi_i^{m_2}
\end{aligned}
\tag{A.36}
$$

Finally, we bring solution y_i from A.33 into A.15 and get the wanted approximate solutions ρ, ρu, ρv and E for new problem parameter values.

Appendix B

POD in relation to SVD

In this appendix one of the most powerful method of data analysis for multivariate and non linear phenomena will be described. Essentially, POD is a linear procedure that takes a given collection of input data and creates an orthogonal basis constituted by functions estimated as the solutions of an integral eigenvalue problem. These eigenfunctions are optimal in terms of representation of the energy present within the data, and only a small number of them are necessary to represent the main underlying information. Reduced Order Modelling by Proper Orthogonal Decomposition is based on a Galerkin projection of the governing equations onto subspaces spanned by those POD basis functions (also known as modes) yielding a simple set of ordinary differential equations (ODEs).

In the context of affine parameter dependence, which the operator is expressible as the sum of products of parameter-dependent functions and parameter-independent operators, allows an offline/online decomposition and is the basis for efficient reduced simulation. Then, suppose that we want to approximate a vector function $u(x,t)$ over some domain of interest Ω as a finite sum in the separated variables form:

$$u(x,t) \approx \sum_{j=1}^{N} y_j(t)\Psi_j(x) \tag{B.1}$$

In the applied cases that will be found in this thesis, numerical simulations of fluid flows equations, x can be considered as a spatial coordinate and t as a temporal coordinate, or even more, as any parameter variable in which solution physics depends on (Mach number, Angle of Attack...).

The approximation (B.1) becomes exact as $N \longrightarrow +\infty$, and is not unique. A classic way to solve this problem is to use for the basis functions $\Psi_j(x)$, functions given a priori, for example Fourier series, Legendre polynomials or Chebyshev polynomials. An alternative approach could be to determine the functions $\Psi_j(x)$ that are naturally intrinsic for the approximation of the function $u(x,t)$. This particular approach corresponds to the Proper Orthogonal Decomposition.

Related to the $\Psi_j(x)$ selected basis functions will be the $y_j(t)$ functions. Different $y_j(t)$ sequence corresponds to each choice of basis functions $\Psi_j(x)$, and each $y_j(t)$ will depend only in each associated $\Psi_j(x)$, for each j value, and not on the other $\Psi(x)$. Then, for selecting basis functions $\Psi_j(x)$, it will be useful to be orthonormal one to each other:

$$\int_{\Omega} \Psi_{j_1}(x)\Psi_{j_2}(x)\mathrm{d}x = \delta_{j_1 j_2} \tag{B.2}$$

where

$$\delta_{j_1 j_2} = \begin{cases} 0 & for j_1 \neq j_2 \\ 1 & for j_1 = j_2 \end{cases} \tag{B.3}$$

is the Kroenecker delta symbol, then

$$y_j(t) = \int_{\Omega} u(x,t)\Psi_j(x)\mathrm{d}x \tag{B.4}$$

As mentioned before, the accuracy of the approximation (B.1) depends on the size of N, and therefore, we would like to find, for any specific N size, the sequence of orthonormal functions $\Psi_j(x)$ which better approximates $u(x,t)$ in a least square sense. As it will be shown in next subsection, this is what the Singular Value Decomposition (SVD) do. Now consider that we can measure (experimentally or numerically) at N_t different values of t, M realizations of $u(x,t)$ at M different locations $x_1, x_2, ..., x_M$. The approximation problem (B.1) is then equivalent to finding the orthonormal functions $\{\Psi_j(x)\}_{j=1}^{N}$ with $N \leq N_t$ that solve:

$$\min \sum_{i=1}^{N_t} \parallel u(x,t_i) - \sum_{j=1}^{N}[u(x,t_i), \Psi_j(x)]\Psi_j(x) \parallel_2^2, \tag{B.5}$$

where $\parallel \Delta \parallel_2$ define the norm associated to the usual L^2 inner product $(.,.)$, but as it is discussed in chapter 2, other inner product can be used, too. Remind that, for any vector $v \in \mathbb{R}^M$ we have

$$v = \begin{pmatrix} v_1 \\ \vdots \\ v_M \end{pmatrix} \implies \parallel v \parallel_2 = (v,v)^{1/2} = \sqrt{v^T v} = \sqrt{v_1^2 + \cdots + v_M^2} \tag{B.6}$$

The method for solving the (B.5) problem is to arrange the data set into a $M\mathrm{x}N_t$ matrix, called the snapshot matrix A.

$$A = \begin{pmatrix} u(x_1,t_1) & u(x_1,t_2) & \dots & u(x_1,t_{N_t}) \\ u(x_2,t_1) & u(x_2,t_2) & \dots & u(x_2,t_{N_t}) \\ \vdots & \vdots & \vdots & \\ u(x_M,t_1) & u(x_M,t_2) & \dots & u(x_M,t_{N_t}) \end{pmatrix}, \quad A \in \mathbb{R}^{M\mathrm{x}N_t} \tag{B.7}$$

where each of the columns $A_{:,i} \in \mathbb{R}^M$ is a solution $u(x,t_i)$ for an input value of t_i. We note that, if the snapshot data are assumed to be linearly independent, the snapshot data matrix has full column rank.

Regarding to the minimization problem (B.5), the solutions are given by the truncated Singular Value Decomposition (SVD) of length N of the matrix A. For this reason, in next subsection, short review of the SVD is presented.

B.1 Definition of SVD

Let A be a general complex MxN_t matrix. The Singular Value Decomposition (SVD) of A is the factorization [34]:

$$A = U\Sigma V^* \tag{B.8}$$

where U and V are (non-unique) unitary MxM and N_txN_t matrices, respectively, i.e. $UU^* = I_M$ and $VV^* = I_{N_t}$ and $\Sigma = \text{diag}(\sigma_1, ..., \sigma_r)$ with $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r \geq 0$ where $r = \min(M, N_t)$. The rank of A equals the number of nonzero singular values it has. Here, V^* denotes the adjoint matrix of V defined as the conjugate transpose of V. Remind that for a unitary matrix $A^{-1} = A^*$. If $A \in \mathbb{R}^{M\text{x}N_t}$ then $V^* = V^T$, and V is said to be orthogonal.

The σ_i are called the singular values of A (and of A^*), the first r columns of $V = (v_1, v_2, ..., v_{N_t})$ are the right singular vectors, and the first r columns of $U = (u_1, u_2, ..., u_M)$ are the left singular vectors. Since the singular values are arranged in a specific order, the index i of the i-th singular value will be called the singular value number.

B.2 Relationships Between SVD and Eigenvalue Problems

Now is presented how the singular values and the right and left singular vectors of a rectangular matrix A can also be computed by solving symmetric eigenproblems with, e.g., the matrices A^*A or AA^*, instead of computing the SVD of A.

Let $A = U\Sigma V^*$ be a singular value decomposition of $A \in \mathbb{R}^{M\text{x}N_t}$. Then

$$\begin{aligned} A^*A &= V\Sigma^T U^* U\Sigma V^* \\ &= V\Sigma^2 V^* \end{aligned} \tag{B.9}$$

where Σ^2 is a diagonal matrix. Since A^*A is an Hermitian matrix, its eigenvalue decomposition can be written:

$$\begin{aligned} A^*A &= W\Lambda W^{-1} \\ &= W\Lambda W^* \end{aligned} \tag{B.10}$$

where W is an N_txN_t unitary matrix. By comparing the two expression of A, we conclude that

$$\Sigma^2 = \Lambda \tag{B.11}$$

and

$$W = V \tag{B.12}$$

In other words

$$\sigma_i = \sqrt{\lambda_i} \tag{B.13}$$

and (V, Λ) is the eigenvector–eigenvalue decomposition of $A^*A \in \mathbb{R}^{N_t\text{x}N_t}$.

The same development applied to the matrix AA^* leads to

$$\begin{aligned} AA^* &= U\Sigma V^* V\Sigma^T U^* \\ &= U\Sigma^2 U^* \end{aligned} \tag{B.14}$$

so (U, Λ) is the eigenvector–eigenvalue decomposition of $AA^* \in \mathbb{R}^{M\text{x}M}$.

B.3 The covariance matrix

Regard again to the snapshot matrix A (B.7), and remember the objective we are trying to achieve: finding an orthonormal basis, dimensionally smaller than A, which represents as best as possible the information contained in those snapshots. In other words, we are searching a process which compares all the data from Snapshots and keeps the main trends of the system. This process will try to remove redundant information, and will keep the most important directions (modes) in which "things happen". An easy way to identify redundant data is by considering the covariance between data sets. The covariance measures the statistical dependence/independence between two variables. Obviously, strongly statistically dependent variables can be considered as redundant observations of the system. Specifically, consider two sets of measurements with zero means expressed in row vector form:

$$\vec{a} = [a_1 \quad a_2 \quad ... \quad a_n] \qquad \vec{b} = [b_1 \quad b_2 \quad ... \quad b_n] \tag{B.15}$$

where the subscript denotes the sample number. The *variances* of $\vec{a}$ and $\vec{b}$ are given by

$$\sigma_a^2 = \frac{1}{N_t} a a^T \tag{B.16}$$

$$\sigma_b^2 = \frac{1}{N_t} b b^T \tag{B.17}$$

while the *covariance* between these two data sets is given by

$$\sigma_{ab}^2 = \frac{1}{N_t} a b^T \tag{B.18}$$

In snapshot matrix (B.7) there aren't just two vectors, but potentially quite a number of experiments and data that would need to be correlated and checked for redundancy and so needs to be checked for covariance. The appropriate covariance matrix for this case is then

$$C_A = \frac{1}{N_t} A A^T \tag{B.19}$$

The covariance matrix C_A is a square, symmetric $M\text{x}M$ matrix whose diagonal represents the variance of particular measurements. The off-diagonal terms are the covariances between measurement types. Thus C_A captures the correlations between all possible pairs of measurements. Redundancy is thus easily captured since if two data sets are identical (identically redundant), the off-diagonal term and diagonal term would be equal since $\sigma_{ab}^2 = \sigma_a^2 = \sigma_b^2$ if $\vec{a} = \vec{b}$. Thus, large off-diagonal terms correspond to redundancy while small off-diagonal terms suggest

that the two measured quantities are close to being statistically independent and have low redundancy. It should also be noted the meaning of the diagonal terms. Large diagonal terms, or those with large variances, typically represent what we might consider the dynamics of interest (where "things happen") since the large variance suggests strong fluctuations in that variable. Thus the covariance matrix is the key component to understanding the entire data analysis. Summarizing, the covariance matrix:

- C_A is a square, symmetric M x M matrix.
- The diagonal terms of C_A are the variances for particular measurements. By assumption, large variances correspond to dynamics of interest, whereas low variances are assumed to correspond to uninteresting dynamics.
- The off-diagonal terms of C_A are the covariances between measurements. Indeed, the offdiagonals capture the correlations between all possible pairs of measurements. A large offdiagonal term represents two events that have a high degree of redundancy, whereas a small off-diagonal coefficient means there is little redundancy in the data, i.e. they are statistically independent.
- It is also possible to compute the covariance matrix as $C_A = \frac{1}{N_t} A^T A$, and in that case resulting matrix would be N_txN_t dimension.

The insight given by the covariance matrix leads to our ultimate aim of: removing redundancy and identifying those signals with maximal variance. Thus, in a mathematical sense we are simply asking to represent C_A so that the diagonals are ordered from largest to smallest and the off-diagonals are zero, i.e. our task is to diagonalize the covariance matrix. This is exactly what the SVD does, thus allowing it to becomes the tool of choice for data analysis and dimensional reduction. In fact, the SVD diagonalizes and each singular direction captures as much energy as possible as measured by the singular values σ_j.

The key idea behind the diagonalization is simply this: there exists an ideal basis in which the C_A can be written (diagonalized) so that in this basis, all redundancies have been removed, and the largest variances of particular measurements are ordered. In the language being developed here, this means that the system has been written in terms of its Principal Components, or in a Proper Orthogonal Decomposition (POD).

B.4 Diagonalization by Eigenvectors and Eigenvalues

The most straightforward way to diagonalize the covariance matrix is by making the observation that AA^T is a square, symmetric MxM matrix, i.e. it is self-adjoint so that the M eigenvalues are real and distinct. Linear algebra provides theorems which state that such a matrix can be rewritten as

$$AA^T = S\Lambda S^{-1} \tag{B.20}$$

where the matrix S is a matrix of the eigenvectors of AA^T arranged in columns. Since it is a symmetric matrix, these eigenvector columns are orthogonal so that

ultimately the S can be written as a unitary matrix with $S^{-1} = S^T$. Recall that the matrix Λ is a diagonal matrix whose entries correspond to the M distinct eigenvalue of AA^T.

This suggests that instead of working directly with the matrix A, we consider working with the transformed variable, or in the principal component basis,

$$Y = S^T A \tag{B.21}$$

For this new basis, we can then consider its covariance

$$\begin{aligned} C_Y &= \frac{1}{N_t} Y Y^T \\ &= \frac{1}{N_t} S^T A A^T S \\ &= \frac{1}{N_t} S^T S \Lambda S^T S \\ &= \frac{1}{N_t} \Lambda \end{aligned} \tag{B.22}$$

which is clearly diagonal, with eigenvalues on it.

In this basis, the principal components are the eigenvectors of AA^T with the interpretation that the jth diagonal value of C_Y is the variance of A along x_j, the jth column of S.

B.5 Diagonalization by SVD

A second method for diagonalizing the covariance matrix is the SVD method. The SVD can diagonalize any matrix by working in the appropriate pair of bases U and V. Now, instead of working directly with the matrix A, we consider working with the transformed variable Y, in the principal component basis, defined as

$$Y = U^* A \tag{B.23}$$

where U is the unitary transformation associated with the SVD: $A = U\Sigma V^*$. For this new basis, just as in the eigenvalue/eigenvector formulation, we can then consider its covariance:

$$\begin{aligned} C_Y &= \frac{1}{N_t} Y Y^T \\ &= \frac{1}{N_t} U^* A A^* U \\ &= \frac{1}{N_t} U^* U \Sigma V^* V \Sigma^T U^* U \\ &= \frac{1}{N_t} \Sigma^2 \end{aligned} \tag{B.24}$$

This makes explicit the connection shown in (B.11) between the SVD and the eigenvalue method, namely that $\Sigma^2 = \Lambda$. This gives the SVD method for producing the principal components. As regarded, matrices can be diagonalized via either an eigenvalue decomposition or an SVD decomposition. However, there are three key differences in the diagonalization process making the SVD method more robust:

- The SVD performs the diagonalization using two different bases, U and V, while the eigenvalue method uses a single basis S.
- The SVD method uses an orthonormal basis while the basis vectors in S, while linearly independent, are not generally orthogonal.
- Finally, the SVD is guaranteed to exist for any matrix A while the same is not true, even for square matrices, for the eigenvalue decomposition.

Now comes the last, and most formative, property associated with the SVD: low dimensional approximations to high degree of freedom or complex systems. In linear algebra terms, this is also known as low rank approximations. The interpretation of the theorems associated with these low dimensional reductions are critical for the use and implementation of the SVD. Thus we consider the following:

Theorem. *A is the sum of r rank-one matrices*

$$A = \sum_{j=1}^{r} \sigma_j \vec{u}_j \vec{v}_j^* \tag{B.25}$$

There are a variety of ways to express an m x n matrix A as a sum of rank-one matrices. The bottom line is this: the Jth partial sum captures as much of the matrix A as possible. Thus the partial sum of the rank-one matrices is an important object to consider. This leads to the following theorem:

Theorem. *For any N so that $0 \leq N \leq r$, we can define the partial sum*

$$A = \sum_{j=1}^{N} \sigma_j \vec{u}_j \vec{v}_j^* \tag{B.26}$$

And if $N = \min\{m, n\}$, define $\sigma_{N+1} = 0$ Then

$$||A - A_N||_2 = \sigma_{J+1} \tag{B.27}$$

Interpreting this theorem is critical. Geometrically, we can ask what is the best approximation of a hyper-ellipsoid by a line segment? Simply take the line segment to be the longest axis, i.e. that associated with the singular value σ_1. Continuing this idea, what is the best approximation by a two-dimensional ellipse? Take the longest and second longest axes, i.e. those associated with the singular values σ_1 and σ_2. After r steps, the total energy in A is completely captured. Thus the SVD gives a type of least-square fitting algorithm, allowing us to project the matrix onto low dimensional representations in a formal, algorithmic way. Herein lies the ultimate power of the method.

Bibliography

[1] N. Akkari et al. "A mathematical and numerical study of the sensitivity of a reduced order model by POD (ROM–POD), for a 2D incompressible fluid flow". In: *Journal of Computational and Applied Mathematics* 270 (2014), pp. 522–530.

[2] Alessandro Alla, Carmen Gräßle, and Michael Hinze. "A residual based snapshot location strategy for POD in distributed optimal control of linear parabolic equations". In: *IFAC-PapersOnLine* 49.8 (2016), pp. 13–18.

[3] David Amsallem, Matthew J Zahr, and Charbel Farhat. "Nonlinear model order reduction based on local reduced-order bases". In: *International Journal for Numerical Methods in Engineering* 92.10 (2012), pp. 891–916.

[4] Athanasios C. Antoulas. "A survey of model reduction methods for large-scale systems". In: *American Mathematical Society* 280 (2006), pp. 1–28.

[5] Patricia Astrid. "Reduction of Process Simulation Models: a Proper Orthogonal Decomposition Approach". PhD thesis. 2004, p. 250.

[6] Jeanne a Atwell. "Proper Orthogonal Decomposition for Reduced Order Control of Partial Differential Equations". PhD thesis. 2000.

[7] Nadine Aubry et al. "The dynamics of coherent structures in the wall region of a turbulent boundary layer". In: *Journal of Fluid Mechanics* 192 (1988), pp. 115–173. DOI: 10.1017/S0022112088001818.

[8] Timothy J. Baker. "Mesh adaptation strategies for problems in fluid dynamics". In: *Finite Elements in Analysis and Design* 25.3 (1997). Adaptive Meshing, Part 2, pp. 243–273.

[9] Etienne Balmès. "Parametric families of reduced finite element models. Theory and applications". In: *Mechanical Systems and Signal Processing* 10.4 (1996), pp. 381–394.

[10] Maxime Barrault et al. "An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations". In: *Comptes Rendus Mathematique* 339.9 (2004), pp. 667–672.

[11] Ulrike Baur, Peter Benner, and Lihong Feng. "Model Order Reduction for Linear and Nonlinear Systems: A System-Theoretic Perspective". In: *Archives of Computational Methods in Engineering* 21.4 (2014), pp. 331–358.

[12] Ulrike Baur et al. "Interpolatory Projection Methods for Parameterized Model Reduction". In: *SIAM Journal on Scientific Computing* 33.5 (2011), pp. 2489–2518.

[13] Peter Benner, Serkan Gugercin, and Karen Willcox. “A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems”. In: *SIAM Review* 57.4 (2015), pp. 483–531.

[14] G. Berkooz, P. Holmes, and JL. Lumley. “The proper orthogonal decomposition in the analysis of turbulent flows”. In: *Annual Review of Fluid Mechanics* 25 (1993), pp. 539–575.

[15] Dimitris Bertsimas and John N Tsitsiklis. *Introduction to linear optimization*. Vol. 6. Athena Scientific Belmont, MA, 1997.

[16] B.N. Bond and L. Daniel. “A Piecewise-Linear Moment-Matching Approach to Parameterized Model-Order Reduction for Highly Nonlinear Systems”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 26.12 (2007), pp. 2116–2129.

[17] S. Boyaval et al. “Reduced Basis Techniques for Stochastic Problems”. In: *Archives of Computational Methods in Engineering* 17.4 (2010), pp. 435–454.

[18] T. Bui-Thanh, K. Willcox, and Omar Ghattas. “Model Reduction for Large-Scale Systems with High-Dimensional Parametric Input Space”. In: *SIAM Journal on Scientific Computing* 30.6 (2008), pp. 3270–3288.

[19] T Bui-Thanh, Murali Damodaran, and Karen Willcox. “Proper Orthogonal Decomposition Extensions for Parametric Applications in Compressible Aerodynamics”. In: *21st AIAA Applied Aerodynamics Conference*. Reston, Virigina: American Institute of Aeronautics and Astronautics, 2003, pp. 1–11.

[20] Tan Bui-Thanh. “Model-Constrained Optimization Methods for Reduction of Parameterized Large-Scale Systems”. PhD thesis. MIT, 2007.

[21] Kevin Carlberg, Charbel Bou-Mosleh, and Charbel Farhat. “Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations”. In: *International Journal for Numerical Methods in Engineering* 86.2 (2011), pp. 155–181.

[22] Saifon Chaturantabut. “Temporal localized nonlinear model reduction with a priori error estimate”. In: *Applied Numerical Mathematics* 119 (2017), pp. 225–238.

[23] Saifon Chaturantabut and Danny C Sorensen. “Nonlinear Model Reduction via Discrete Empirical Interpolation”. In: *SIAM Journal on Scientific Computing* 32.5 (2010), pp. 2737–2764.

[24] Erik Adler Christensen, Morten Brøns, and Jens Nørkær Sørensen. “Evaluation Of Proper Orthogonal Decomposition – Based Decomposition Techniques”. In: 21.4 (2000), pp. 1419–1434.

[25] Yunfei Chu, Mitchell Serpas, and Juergen Hahn. “State-preserving nonlinear model reduction procedure”. In: *Chemical Engineering Science* 66.17 (2011), pp. 3907–3913.

[26] L. Cordier and M. Bergmann. “Proper orthogonal decomposition: an overview”. In: *Technical Report Lecture series 2002-04 and 2003-04 on post-processing of experimental and numerical data,, Von Karman Institute for Fluid Dynamics*. 2003.

[27] Vít Dolejsí and Jirí Felcman. “Anisotropic mesh adaptation for transonic and supersonic flow simulation”. In: *Proceedings of ALGORITMY Conference on Scientific Computing, 78–85 2002.* 2002, pp. 78–85.

[28] Martin Drohmann, Bernard Haasdonk, and Mario Ohlberger. “Adaptive reduced basis methods for nonlinear convection–diffusion equations”. In: *Finite Volumes for Complex Applications VI Problems & Perspectives.* Springer, 2011, pp. 369–377.

[29] Jens L. Eftang, Anthony T. Patera, and Einar M. Rønquist. “An ”hp” Certified Reduced Basis Method for Parametrized Elliptic Partial Differential Equations”. In: *SIAM Journal on Scientific Computing* 32.6 (2010), pp. 3170–3200.

[30] Jens L. Eftang and Benjamin Stamm. “Parameter multi-domain ‘hp’ empirical interpolation”. In: *International Journal for Numerical Methods in Engineering* 90.4 (2012), pp. 412–428.

[31] F. Fang et al. “Non-linear Petrov–Galerkin methods for reduced order hyperbolic equations and discontinuous finite element methods”. In: *Journal of Computational Physics* 234 (2013), pp. 540–559.

[32] Lihong Feng, Athanasios C Antoulas, and Peter Benner. “Some a posteriori error bounds for reduced-order modelling of (non-)parametrized linear systems”. In: *ESAIM: Mathematical Modelling and Numerical Analysis* 51.6 (2017), pp. 2127–2158.

[33] Michel Fortin. “Scientific Computing and Applications”. In: ed. by Peter Minev and Yanping Lin. Commack, NY, USA: Nova Science Publishers, Inc., 2001. Chap. Anisotropic Mesh Adaptation Through Hierarchical Error Estimators, pp. 53–65.

[34] Gene H Golub and Charles F Van Loan. *Matrix Computations.* 2013, p. 780.

[35] G. Gori et al. “Experimental assessment of the open-source SU2 CFD suite for ORC applications”. In: *4TH INTERNATIONAL SEMINAR ON ORC POWER SYSTEMS.* Ed. by Dossena, V and Guardone, A and Astolfi, M. Vol. 129. Energy Procedia. 4th International Seminar on Organic Rankine Cycle Power Systems (ORC), Politecnico Milano Bovisa Campus, Milano, ITALY, SEP 13-15, 2017. 2017, 256–263.

[36] Martin A. Grepl and Anthony T. Patera. “A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations”. In: *ESAIM: Mathematical Modelling and Numerical Analysis* 39.1 (2005), pp. 157–181.

[37] Bernard Haasdonk, Markus Dihlmann, and Mario Ohlberger. “A training set and multiple bases generation approach for parameterized model reduction based on adaptive grids in parameter space”. In: *Mathematical and Computer Modelling of Dynamical Systems* 17.4 (2011), pp. 423–442.

[38] Bernard Haasdonk and Mario Ohlberger. “Reduced basis method for finite volume approximations of parametrized linear evolution equations”. In: *ESAIM: Mathematical Modelling and Numerical Analysis* 42.2 (2008), pp. 277–302.

[39] W. G. Habashi et al. “Anisotropic mesh adaptation: Towards user-independent, mesh-independent and solver-independent CFD solutions: Part I: General principles”. In: *Internat. J. Numer. Methods Fluids* 32 (2000), pp. 725–744.

[40] Jan S Hesthaven, Gianluigi Rozza, Benjamin Stamm, et al. *Certified reduced basis methods for parametrized partial differential equations*. Springer, 2016.

[41] P. Holmes, JL. Lumley, and G. Berkooz. *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge university press, 1998.

[42] Philip Holmes, John L Lumley, and Gal Berkooz. *Turbulence, Coherent structurs, dynamical systems and symmetry*. 2012.

[43] D.B.P. Huynh et al. “A successive constraint linear optimization method for lower bounds of parametric coercivity and inf–sup stability constants”. In: *Comptes Rendus Mathematique* 345.8 (2007), pp. 473–478.

[44] A Iollo. “Remarks on the approximation of the Euler equations by a low order model”. In: (1997), pp. 1–28.

[45] Angelo Iollo, Stéphane Lanteri, and Jean Antoine Désidéri. “Stability properties of POD-Galerkin approximations for the compressible Navier-Stokes equations”. In: *Theoretical and Computational Fluid Dynamics* 13.6 (2000), pp. 377–396.

[46] K. Ito and S.S. Ravindran. “A Reduced-Order Method for Simulation and Control of Fluid Flows”. In: *Journal of Computational Physics* 143.2 (1998), pp. 403–425.

[47] I. Kalashnikova and M. F. Barone. “On the stability and convergence of a Galerkin reduced order model (ROM) of compressible flow with solid wall and far-field boundary treatment”. In: *International Journal for Numerical Methods in Engineering* 83.10 (2010), pp. 1345–1375.

[48] Michal A. Kopera and Francis X. Giraldo. “Analysis of adaptive mesh refinement for IMEX discontinuous Galerkin solutions of the compressible Euler equations with application to atmospheric simulations”. In: *Journal of Computational Physics* 275 (2014), pp. 92–117.

[49] K. Kunisch and S. Volkwein. “Galerkin Proper Orthogonal Decomposition Methods for a General Equation in Fluid Dynamics”. In: *SIAM Journal on Numerical Analysis* 40.2 (2002), pp. 492–515.

[50] K. Kunisch and Stefan Volkwein. “Galerkin proper orthogonal decomposition methods for parabolic problems”. In: *Numerische Mathematik* 90.1 (2001), pp. 117–148.

[51] Karl Kunisch and Stefan Volkwein. “Optimal snapshot location for computing POD basis functions”. In: *ESAIM: Mathematical Modelling and Numerical Analysis* 44.3 (2010), pp. 509–529.

[52] Nathan Kutz. *Data-Driven modelling and scientific computation*. 2013.

[53] O. Lass and S. Volkwein. “Adaptive POD basis computation for parametrized nonlinear systems using optimal snapshot location”. In: *Computational Optimization and Applications* 58 (2014), pp. 645–677.

[54] M. M. Loeve. *Probability Theory*. Van Nostrand, Princeton, NJ, 1988.

[55] JL. Lumley. "The Structure of inhomogeneous turbulent flows". In: *A.M. Yaglom and V.I. Tatarski (Eds.), Atmospheric Turbulence and Radio Wave Propagation* (1967), pp. 166–178.

[56] John Leask Lumley. "The structure of inhomogeneous turbulent flows". In: *Atmospheric turbulence and radio wave propagation* (1967).

[57] Martin Meckesheimer et al. "Computationally Inexpensive Metamodel Assessment Strategies". In: *AIAA Journal* 40.10 (2002), pp. 2053–2060.

[58] J. Mercer. "Functions of positive and negative type and their connection with the theory of integral equations". In: *Philos. Trans. Roy. Soc. London* (1909), pp. 415–446.

[59] B. Moore. "Principal component analysis in linear systems: Controllability, observability, and model reduction". In: *IEEE Transactions on Automatic Control* 26.1 (1981), pp. 17–32.

[60] Nirmal J. Nair and Maciej Balajewicz. "Transported snapshot model order reduction approach for parametric, steady-state fluid flows containing parameter-dependent shocks". In: *International Journal for Numerical Methods in Engineering* 117.12 (2019), pp. 1234–1262.

[61] Nguyen Ngoc Cuong, Karen Veroy, and Anthony T. Patera. "Certified Real-Time Solution of Parametrized Partial Differential Equations". In: *Handbook of Materials Modeling: Methods*. Ed. by Sidney Yip. Dordrecht: Springer Netherlands, 2005, pp. 1529–1564.

[62] Holger Nobach et al. "Review of Some Fundamentals of Data Processing". In: *Springer Handbook of Experimental Fluid Mechanics*. Ed. by Cameron Tropea, Alexander L. Yarin, and John F. Foss. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 1337–1398.

[63] Robert O'Connor and Martin Grepl. "Offline error bounds for the reduced basis method". In: *Model Reduction of Parametrized Systems*. Springer, 2017, pp. 121–136.

[64] AT Patera and G Rozza. "Reduced Basis Approximation and A Posteriori Error Estimation for Parametrized Partial Differential Equations. Version 1.0, Copyright MIT 2006, to appear in (tentative rubric) MIT Pappalardo Graduate Monographs in Mechanical Engineering". In: *tentative rubric) MIT Pappalardo Graduate Monographs in Mechanical Engineering* (2006).

[65] B. Peherstorfer et al. "Localized Discrete Empirical Interpolation Method". In: *SIAM Journal on Scientific Computing* 36.1 (2014), A168–A192.

[66] J Peraire et al. "Adaptive remeshing for compressible flow computations". In: *Journal of Computational Physics* 72.2 (1987), pp. 449–466.

[67] Janet S. Peterson. "The Reduced Basis Method for Incompressible Viscous Flow Calculations". In: *SIAM Journal on Scientific and Statistical Computing* 10.4 (1989), pp. 777–786.

[68] M. Pini et al. “SU2: the Open-Source Software for Non-ideal Compressible Flows”. In: *1ST INTERNATIONAL SEMINAR ON NON-IDEAL COMPRESSIBLE-FLUID DYNAMICS FOR PROPULSION & POWER*. Vol. 821. Journal of Physics Conference Series. 1st International Seminar on Non-Ideal Compressible-Fluid Dynamics for Propulsion and Power (NICFD), Varenna, ITALY, OCT 19-21, 2016. 2017.

[69] René Pinnau. “Model Reduction via Proper Orthogonal Decomposition”. In: *Model Order Reduction: Theory, Research Aspects and Applications* (2008), pp. 95–109.

[70] C. Prud'homme et al. “Reliable Real-Time Solution of Parametrized Partial Differential Equations: Reduced-Basis Output Bound Methods”. In: *Journal of Fluids Engineering* 124.1 (2002), p. 70.

[71] Thomas H. Pulliam. *The Euler Equations. Revised Notes November 1994*. 1994.

[72] A Quarteroni. “Reduced Basis Approximation For Parametrized Partial Differential Equations”. In: *Numerical Models For Differential Problems* (2009), pp. 547–579.

[73] A Quarteroni, A Manzoni, and F Negri. *Reduced Basis Methods for Partial Differential Equations - An Introduction*. 2016.

[74] Alfio Quarteroni, Gianluigi Rozza, and Andrea Manzoni. “Certified reduced basis approximation for parametrized partial differential equations and applications”. In: *Journal of Mathematics in Industry* 1.1 (2011), p. 3.

[75] Muruhan Rathinam and Linda R. Petzold. “A New Look at Proper Orthogonal Decomposition”. In: *SIAM Journal on Numerical Analysis* 41.5 (2003), pp. 1893–1925.

[76] Lakhdar Remaki and Wagdi G Habashi. “3-D Mesh Adaptation on Multiple Weak Discontinuities and Boundary Layers”. In: *SIAM Journal on Scientific Computing* 28.4 (2006), pp. 1379–1397.

[77] Lakhdar Remaki et al. “A High Order Finite Volume-HLLC Solver and Anisotropic Delaunay Mesh Adaptation”. In: *47th AIAA Aerosp. Sci. Meet. Incl. New Horizons Forum Aerosp. Expo.* Aerospace Sciences Meetings. American Institute of Aeronautics and Astronautics, Jan. 2009. DOI: doi:10.2514/6.2009-1498. URL: http://dx.doi.org/10.2514/6.2009-1498.

[78] G. Rozza, D. B.P. Huynh, and A. T. Patera. “Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: Application to transport and continuum mechanics”. In: *Archives of Computational Methods in Engineering* 15.3 (2008), pp. 229–275.

[79] Gianluigi Rozza. “Reduced-basis methods for elliptic equations in sub-domains with a posteriori error bounds and adaptivity”. In: *Applied Numerical Mathematics* 55.4 (2005), pp. 403–424.

[80] Feriedoun Sabetghadam and Alireza Jafarpour. “alpha Regularization of the POD-Galerkin dynamical systems of the Kuramoto–Sivashinsky equation”. In: *Applied Mathematics and Computation* 218.10 (2012), pp. 6012–6026.

[81] Wilhelmus H a Schilders, Henk a Van Der Vorst, and Joost Rommes. *Model Order Reduction: Theory, Research Aspects and Applications*. Vol. 13. 2008, p. 471.

[82] L. Sirovich. "Turbulence and the dynamics of coherent structures, parts I-III". In: (1987), pp. 561–590.

[83] Lawrence Sirovich. "Turbulence and the dynamics of coherent structures part i: coherent structures*". In: *Quarterly of Applied Mathematics* XLV.3 (1987), pp. 561–571.

[84] *SU2, the Open-Source CFD Code*. URL: `https://su2code.github.io/`.

[85] R. Verfürth. "A posteriori error estimation and adaptive mesh-refinement techniques". In: *Journal of Computational and Applied Mathematics* 50.1 (1994), pp. 67–83.

[86] K Veroy and AT Patera. "Certified real-time solution of the parametrized steady incompressible Navier–Stokes equations: rigorous reduced-basis a posteriori error bounds". In: *International Journal for Numerical Methods in Fluids* 47.8-9 (2005), pp. 773–788.

[87] Karen Veroy et al. "A Posteriori Error Bounds for Reduced-Basis Approximation of Parametrized Noncoercive and Nonlinear Elliptic Partial Differential Equations". In: *16th AIAA Computational Fluid Dynamics Conference* June (2003), pp. 1–18.

[88] Stefan Volkwein. "Proper orthogonal decomposition: Theory and reduced-order modelling". In: *Lecture Notes, University of Konstanz* 4.4 (2013).

[89] Kyle Washabaugh et al. "Nonlinear Model Reduction for CFD Problems Using Local Reduced-Order Bases". In: *42nd AIAA Fluid Dynamics Conference and Exhibit*.

[90] D. Xiao et al. "Non-linear Petrov–Galerkin methods for reduced order modelling of the Navier–Stokes equations using a mixed finite element pair". In: *Computer Methods in Applied Mechanics and Engineering* 255 (2013), pp. 147–157.

[91] Raul Yondo, Esther Andrés, and Eusebio Valero. "A review on design of experiments and surrogate models in aircraft real-time and many-query aerodynamic analyses". In: *Progress in Aerospace Sciences* 96 (2018), pp. 23–61.

[92] Zhao Zhan, Wagdi G. Habashi, and Marco Fossati. "Local Reduced-Order Modeling and Iterative Sampling for Parametric Analyses of Aero-Icing Problems". In: *AIAA Journal* 53.8 (2015), pp. 2174–2185.

[93] QS. Zhang, YZ. Liu, and SF. Wang. "The identification of coherent structures using proper orthogonal decomposition and dynamic mode decomposition". In: *Journal of Fluids and Structures Vol* 49 (2014), pp. 53–72.