HAL
open science

Data-enabled, reduced-order simulation of dynamical systems with application to vehicle crash simulation

Pierre Phalippou

To cite this version:

Pierre Phalippou. Data-enabled, reduced-order simulation of dynamical systems with application to vehicle crash simulation. Mechanics [physics.med-ph]. Université de Technologie de Compiègne, 2020. English. NNT : 2020COMP2544 . tel-03207446

HAL Id: tel-03207446

https://theses.hal.science/tel-03207446v1

Submitted on 25 Apr 2021

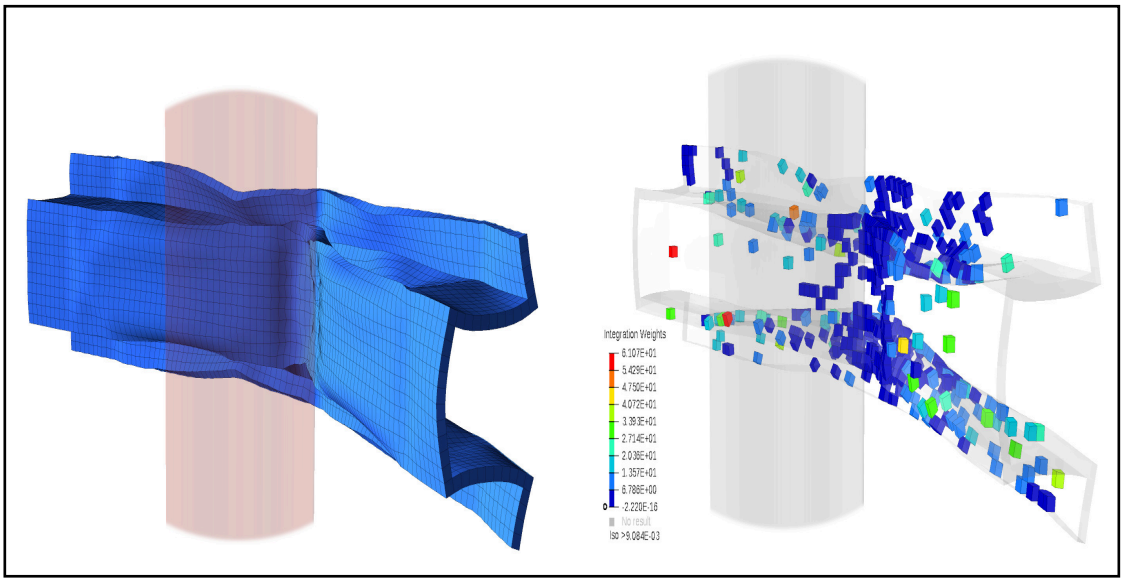
HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

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

Par **Pierre PHALIPPOU**

Data-enabled, reduced-order simulation of dynamical systems with application to vehicle crash simulation

Thèse présentée pour l'obtention du grade de Docteur de l'UTC



Soutenue le 6 mars 2020

Spécialité : Mécanique Numérique : Unité de recherche en Mécanique - Laboratoire Roberval (FRE UTC - CNRS 2012)

D2544

Data-Enabled, Reduced-Order Simulation of Dynamical Systems

with Application to Vehicle Crash Simulation

Spécialité : Mécanique Numérique

Pierre Phalippou

Laboratoire Roberval, FRE 2012 UTC-CNRS
Sorbonne Universités, Université de Technologie de Compiègne

A thesis presented for the degree of *Doctor*
Defense on the 6th of March 2020

Composition of the Jury

Juan Pedro Berror Ramirez, Altair Engineering France, France
Salim Bouabdallah (Co-supervisor)
Delphine Brancherie, Université de Technologie de Compiègne, France
Piotr Breitkopf, Université de Technologie de Compiègne, France (Supervisor)
Fabian Duddeck, Technical University of Munich, Germany (Rewiever) Pierre Kerfriden, Cardiff University, United Kingdom (Reviewer)
David Néron, École Normale Supérieure Paris-Saclay, France
Malek Zarroug, PSA Group, France (Invited)

Résumé

Lors de la conception de nouveaux véhicules, les constructeurs automobiles font face à de nombreuses contraintes telles que la sécurité, le confort, le poids et la consommation de carburant. La simulation numérique, et plus précisément la méthode des éléments finis, y est très largement utilisée car elle permet de tester l'influence des différents paramètres sur le comportement du système. Ainsi, de nombreux tests réels sur prototypes sont évités. La conception de nouveaux véhicules est néanmoins ralentie par l'utilisation de modèles de plus en plus complexes, en particulier dans les applications au crash. De nombreuses méthodes de réduction de modèles sont développées dans le but d'accélérer les simulations en tirant parti de l'énorme quantité de données disponibles et de la répétitivité des calculs réalisés avec de petites variations dans les paramètres de conception.

Cette thèse vise à appliquer de telles méthodes aux simulations de crash automobiles comprenant des nonlinéarités matérielles et géométriques ainsi que de nombreux contacts. Effectuée en étroite collaboration entre le groupe PSA, le développeur de logiciel Altair Engineering et le laboratoire de mécanique UTC-CNRS Roberval, ce projet met l'accent sur les méthodes de réduction *intrusives* nécessitant des modifications dans le code source du solveur éléments finis. Une base réduite de fonctions de forme globales est formée pendant une phase d'apprentissage *hors ligne* sur un ensemble de données et utilisée *en ligne* pour réduire la taille du modèle et permettre l'hyper-réduction des forces internes et l'accélération du temps de calcul.

Les contributions de cette thèse concernent trois aspects de la phase d'apprentissage. Les modifications apportées à la décomposition en valeurs singulières incrémentale la rendent plus facile à utiliser tout en accélérant le temps d'apprentissage des bases réduites. Le critère sparse est proposé pour la sélection des fonctions de base réduite, offrant de meilleures performances dans la phase de réduction ainsi qu'une meilleure corrélation entre l'erreur d'apprentissage *hors ligne* et l'erreur d'approximation *en ligne*. Une formulation innovante du problème d'optimisation d'hyper-réduction tirant parti de la programmation en variables mixtes est suggérée comme référence pour le développement d'heuristiques. Additionnellement, de nouvelles contraintes sur l'intégration polynomiale et la conservation du volume sont utilisées.

Tous les développements informatiques sont réalisés dans le solveur éléments finis explicite industriel Altair Radioss avec un accès complet au code source. À la connaissance de l'auteur, cette caractéristique constitue un différentiateur majeur de la présente contribution. Les difficultés relatives à cet aspect sont discutées. Enfin, des recommandations et perspectives pour l'application robuste de la méthode à l'étude de véhicules automobiles en tenue de crash sont données.

Abstract

Vehicle manufacturers face numerous constraints regarding security, comfort, weight, and fuel consumption when designing new automotive vehicles. Numerical simulation, and more precisely, the finite element method, is extensively used in the process to achieve cost reduction while allowing to test the impact of multiple design parameters on the system behavior. Nevertheless, industrial finite element models of automotive crash are growing prohibitively expensive in computation time, slowing down the design process. Numerous model order reduction methods have been developed in order to speed-up simulations by making use of the humongous amount of collected data and the repetitiveness of computations with slight changes to the design variables.

This thesis aims to apply model order reduction methods to finite element simulations of automotive crash, involving material and geometric nonlinearity as well as contact. This project, conducted in close collaboration between the car manufacturer PSA group, the software developer Altair engineering, and the computational mechanics group at the UTC-CNRS joint laboratory Roberval, targets intrusive methods necessitating modifications in the finite element solver source code. Amongst different promising methods reviewed in the state-of-the-art, our work focuses on projection-based reduced order modeling and hyper-reduction. A reduced-basis of global shape functions is built upon relevant data in the *offline* training phase and used *online* to reduce the model size, enabling the hyper-reduction of internal forces and computational speed-up.

The contributions of this thesis concern three aspects of the training phase. Developments of the incremental singular value decomposition allow for a more flexible and faster reduced-basis training. The proposed sparse criterion enables the formation of basis functions subsets with enhanced performances and better correlation between the *offline* training phase approximation error and the actual error in the *online* reduction phase. An innovative formulation of the hyper-reduction optimization problem involving mixed-integer programming, added constraints on polynomial integration, and volume conservation is suggested for the benchmarking of future heuristic methods.

All implementations are performed in the industrial explicit finite element solver Altair Radioss with complete access to the source code. To the author's best knowledge, the implementation of such a method in industrial explicit finite element solver represents a significant differentiator of the present contribution. Implementation-specific difficulties relevant to this aspect of the thesis are discussed. Finally, recommendations, as well as perspectives and further developments required for the robust application of the method to industrial numerical simulations of vehicle crash, are given.

Dedication

I dedicate this manuscript to my parents.

Declaration

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it contains no material previously published or produced by another party in fulfilment, partial or otherwise, of any other degree or diploma at another University or institute of higher learning, except where due acknowledgement is made in the text.

Pierre Phalippou

Acknowledgment

Many people supported me during my thesis, their help have been crucial in my work.

I thank my supervisor Piotr Breitkopf for accompanying me during these three years. His advices have been essential in carrying this research project through.

I praise Malek Zarroug from PSA Group who initiated this project and enabled the industrial collaboration between the software developer and car manufacturer.

I am also thankful to my industrial supervisors, Juan Pedro Berro Ramirez and Salim Bouabdallah, who both supported me in different ways and motivated me rough periods.

I am grateful to professor emeritus Pierre Villon for the interest he expressed towards my work. His scientific recommandations have been crucial in this project.

I thank Mircea, Daniel, Qiang and Lionel who helped me implementing the method in Altair Radioss.

Finally, I thank all the people working at Altair Engineering France for the amazing work environment.

Contents

1 Introduction **10**
1.1 Industrial context . . . 10
1.2 Opportunities for model order reduction in the design of new vehicles 11
1.3 Outline . . . 13
1.4 Associated journal publications and conference papers . . . 15

2 State of the art **17**
2.1 Brief review of Model Order Reduction methods . . . 18
2.1.1 Proper Generalized Decomposition . . . 18
2.1.2 Equivalent Static Load Method . . . 19
2.1.3 Projection-based Reduced Order Modeling . . . 22
2.1.4 Dimensionality reduction methods . . . 23
2.1.5 Hyper-reduction . . . 25
2.1.6 Summary . . . 26
2.2 Details on the chosen model order reduction methods . . . 27
2.2.1 Finite element formulation of the problem . . . 27
2.2.2 *A posteriori* model order reduction methods . . . 27
2.2.3 Proper Orthogonal Decomposition . . . 28
2.2.4 Energy Conserving Sampling and Weighting . . . 30
2.3 Problematics and contributions . . . 32
2.3.1 'On-the-fly' snapshots selection (Chapter 3) . . . 34
2.3.2 Sparse POD modal subsets (Chapter 4) . . . 34
2.3.3 MIP Hyper-reduction (Chapter 5) . . . 35
2.4 Organization of the manuscript . . . 35

3 'On-the-fly' snapshots selection **36**
3.1 Incremental SVD . . . 38
3.1.1 Incremental enrichment . . . 39
3.1.2 'On-the-fly' snapshot selection and truncation . . . 40
3.1.3 State-of-the-art incremental SVD algorithm . . . 41
3.1.4 State-of-the-art incremental SVD test . . . 41
3.1.5 Note on *a posteriori* centering . . . 44
3.2 Error estimator . . . 45
3.3 Proposed algorithm . . . 46
3.4 Numerical tests . . . 48
3.4.1 Taylor beam impact . . . 48
3.4.2 POD review . . . 49
3.4.3 Proposed incremental SVD tests . . . 51

3.4.4 Comparison with state-of-the-art incremental SVD 52
3.4.5 Online comparison 54
3.4.6 Proposed incremental SVD scalability 56
3.5 Conclusion 59

4 Sparse RB modal subsets 60
4.1 Data approximation and sequential criterion 61
4.2 On sub-optimality of the sequential criterion in POD applications to structural dynamics 63
4.3 Sparse subset selection criterion 67
4.4 Test on a crash box model 73
4.5 Conclusion 76

5 MIP Hyper-reduction 78
5.1 Projected hyper-reduced order model 79
5.1.1 Projected Reduced Order Model 79
5.1.2 Energy-Conserving Sampling and Weighting (ECSW) 80
5.2 Hyper-Reduced Integration 82
5.2.1 MIP formulation 82
5.2.2 Consistency constraints 83
5.2.3 Constraints reduction 84
5.3 Results and discussion 86
5.3.1 Consistency conditions 87
5.3.2 Constraints reduction 93
5.3.3 Discussion 96
5.4 Conclusion 97

6 Implementation 98
6.1 Implementation 98
6.2 Industrial Processes 100
6.2.1 ROM for full-scale crash models 101
6.2.2 Substructuring 102
6.2.3 Online adaptivity 105

7 Conclusion 107

A Notations 109

B Acronyms 112

Bibliography 114

Chapter 1

Introduction

1.1 Industrial context

The Finite Element Method (FEM) and simulation, in general, are nowadays widely used in the industry when designing new complex products. It allows testing various properties of a product numerically, avoiding expensive tests on prototypes. Such technology enables cost and time reduction in the development phase of industrial projects.

Naturally, a FE model is a representation of the reality that it approximates and there is a dilemma between computation time and solution quality when building one. When using such models for industrial design and, more specifically, in car parametric optimization for crash situations, numerous simulations are performed over and over with minor parametric changes in search of the optimal design. Some applications may turn out to be expensive in computational time and with the growing need for finer results, FE models tend to complexify in order to represent crack and contact.

Nowadays, there is a need for simulation software to explore parametric spaces faster. To this end, Model Order Reduction (MOR) methods have been developed to accelerate response time, called Reduced Order Models (ROM). Several approaches exist in the literature to build such models.

The goal of this thesis is to explore the opportunities for MOR in the design of new vehicles. While non-intrusive methods using the solver as a black box are in most cases developed internally by the car manufacturer, this thesis places the focus on intrusive ones necessitating a modification in the solver source code. Developments are directly performed in the industrial-level software with complete access to the source code provided by Altair Engineering France.

The following sections present the opportunities for MOR in car design (1.2), the outline of the thesis (1.3), and finally, the list of scientific contributions (1.4).

1.2 Opportunities for model order reduction in the design of new vehicles

Designing a new vehicle is a complex industrial process involving a multi-objective optimization in which numerous constraints concerning weight, production cost, security, fuel consumption, thermic, noise, vibration, deformation, and durability have to be satisfied. A lightweight vehicle will consume less fuel, requiring a less powerful engine and enabling furthersavings in weight and fuel consumption. Comfort is a major customer concern that entails optimization of road handling as well as noise and vibration. Last but not least, crash standards have to be fulfilled when an accident may not be avoided.

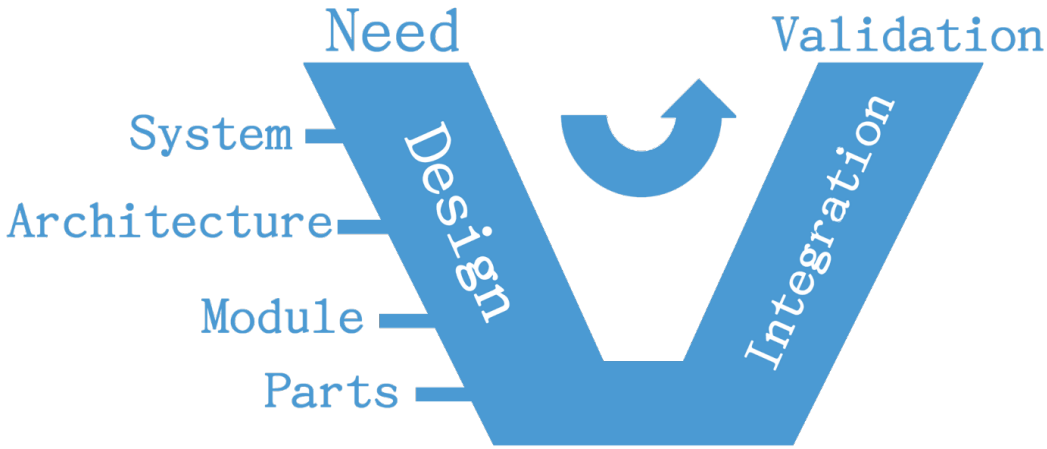


Figure 1.1: V-cycle

The V-cycle, represented in Figure 1.1, is used to summarize the different steps in the development of new vehicles, separated into two main phases: the design and the integration. The design phase goes from the global to the detailed vision of the project, while the second phase goes the opposite direction. In the first phase, vehicle drawings and requirements specification are used to go into the detailed specifications of each module and parts are designed separately. Each part is then incorporated into the vehicle for validation in the integration phase.

Finite Element Analysis (FEA) is extensively used at multiple stages of development to minimize the number of real tests, allowing to numerically study the impact of multiple design variables on the vehicle behavior and physical properties without having to build prototypes, enabling time and cost reduction. It is notably used in individual modules and parts design as well as to validate the complete vehicle integrating all subsystems. Simulation turns out to be a blocking point in the design process as it is needed to step forward in the V-cycle.

Vehicle crash simulation is the more expensive FEA application in new vehicles development. It is a complex multi-objective optimization problem considering different crash, including frontal full width rigid barrier impact, side moving deformable

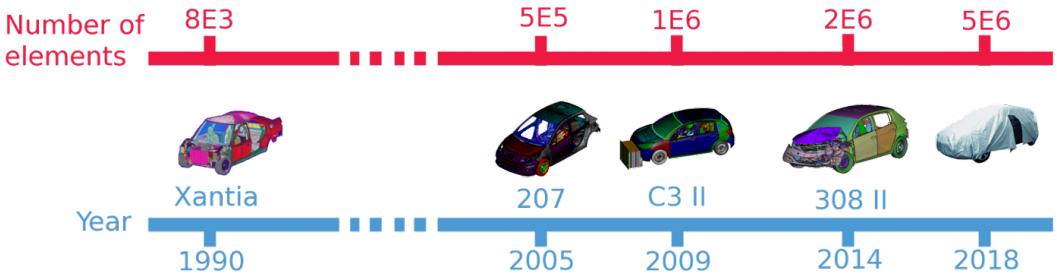


Figure 1.2: Evolution in the finite element model size used in industrial crash simulation at PSA Group.

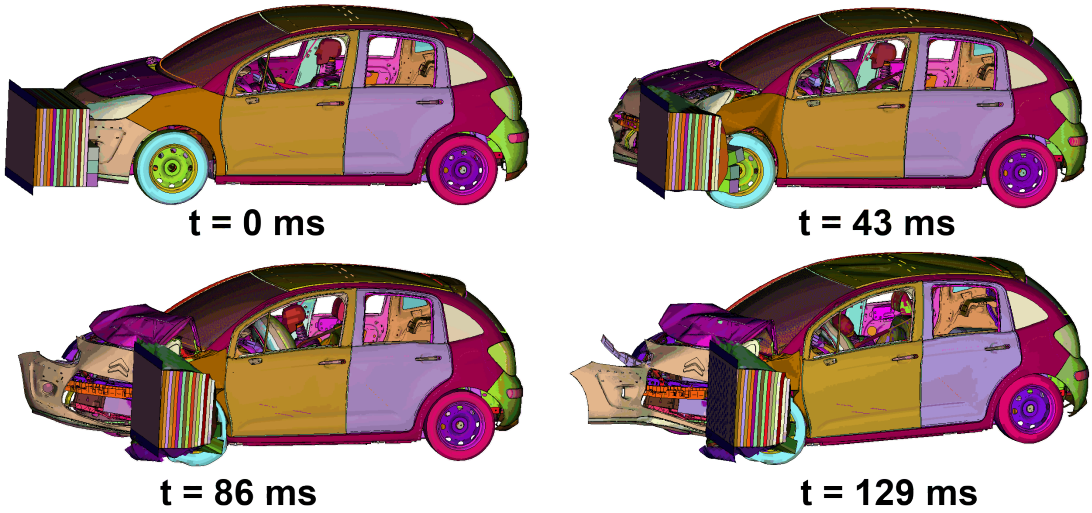


Figure 1.3: Crash simulation example of a frontal offset deformable barrier impact

barrier impact, side oblique pole impact, whiplash pedestrian subsystem tests, extensively described in [25]. Quantities of Interest (QoI) are accelerations, intrusions, and strains. Typical model parameters are material properties, shell thicknesses and test conditions (vehicle speed and attitude, dummy position). Models size, nonlinear material laws evaluation as well as contact and fracture representations, render each simulation computationally intensive. Concomitantly with the rise in computational power, models used for crash simulation are increasingly complex as illustrated by Figure 1.2 on different vehicle models at PSA Group since 1990. The number of elements is proportional to the required number of nonlinear material law evaluations at each time cycle. Thus, it is often used to assess the computational complexity of a model in such applications. Last decade, it had augmented of one order of magnitude in constant search for accuracy and, more recently, for fracture representation. An example of such simulation is provided in Figure 1.3. $T = 130 * 10^{-3}$ seconds of the model are simulated, the latter being composed of 2.5 million shell elements of average size 5 millimeters. Computations took 12 hours on 48 CPUs with a time-step of approximatively $0.5 * 10^{-6}$ second and are to be performed multiple times with little parametric variations in search of the optimal configuration.

Simulating each subsystem of the vehicle separately also is a challenging FEA application. The main difficulty concerns the boundary conditions definition as they

must take into account the potential interaction with surrounding vehicle parts. Moreover, computational costs prevent the use of stochastic methods such as Monte Carlo.

Model order reduction is increasingly used to overcome issues relative to expensive FE simulations to approximate the solution with reduced complexity. MOR methods may be categorized as intrusive or non-intrusive. Non-intrusive reduction methods build a reduced model using the FE solver as a black box. These methods are quite mature and are extensively used in industrial vehicle crashworthiness optimization to explore the design of experiment (Kriging, Non-intrusive POD, Radial Basis Functions ...). On the other hand, intrusive reduction methods require modifications in the FE solver and are thus less used in the industry where the solver source code is generally not available. Compared with non-intrusive methods, intrusive ones make use of the solver to take physic into account. They have the potential to speed-up crash simulations, sensibility and robustness analysis, and enhance boundary conditions in the study of subsystems by capitalizing on the numerous available data. However, expanding intrusive reduction methods to nonlinear transient structural dynamics problems arising in vehicle crashworthiness optimization stills remains a challenge.

In this project, the car manufacturer PSA Group and the software developer Altair Engineering France collaborate to investigate the applicability of intrusive method in the commercial Altair Radioss [3] solver in nonlinear explicit transient structural dynamics FEA.

1.3 Outline

This thesis is motivated by the need to develop intrusive model order reduction to accelerate computations in finite element industrial applications to nonlinear transient structural dynamics and, in particular, vehicle crashworthiness optimization. Such applications involve repetitive simulations of complex models with slight changes in the parameters in search of the optimal design and induce prohibitive computation times. By investing in the plethora of collected data, Projection-based Reduced Order Modeling (PROM) methods have the potential to speed-up computations needed to find the optimal design. A differentiator of the present work lies in the use of an industrial explicit FE solver in which the hyper-reduced proper orthogonal decomposition has been implemented for research purposes. It is allowed by the collaboration between the car manufacturer PSA Group, the Roberval laboratory, and the software developer Altair Engineering providing full access to the legacy explicit FE solver [3] source code in which developments presented in this thesis are performed.

Several bottlenecks to the robust application of PROM methods to vehicle crashworthiness optimization are identified:

- the computational cost associated with training the ROM over previously collected data quickly become computationally prohibitive on large models;
- the ROM approximation error is usually assessed *a priori*, which is shown to be insufficient in applications to nonlinear explicit structural dynamics;

- several options used in full-scale vehicle crash FE models are not compatible with the approach;
- the contact formulation used in industrial FE models is no reduced by the method and still represent a large portion of the computational effort.

The three following contributions of this thesis address some of the above difficulties.

The truncated Singular Value Decomposition is, in the POD, used to compute a low rank representation of the training data set, yielding the Reduced Basis (RB) used in PROM methods to reduce the model complexity. However, the SVD computations turn out infeasible on data sets involving high numbers of snapshots gathered on models with large numbers of degrees of freedom. An incremental alternative has been developed to overcome this limitation when computing low-rank representation over an extensive data set. By enriching a truncated representation of the data as soon as an observation is available, the method avoids expensive data manipulation and storage in binaries while only working with relatively small matrices, permitting considerable computational savings in the training phase. Key features of the method concern 'on-the-fly' truncation and snapshots selection. The present work proposes a new error estimator to keep track of the approximation quality. It drives truncation and snapshots selection 'on-the-fly' in place of former tolerances, simplifying the use of the algorithm along with guaranteeing nearly optimal performances in terms of approximation precision and computation time. These developments are benchmarked against the state-of-the-art incremental SVD as well as a regular SVD for data approximation, computation time, and RB performances in the online reduction phase. Moreover, such an incremental approach has the potential to enable reduced model enrichment in the *online* reduction phase.

The second contribution concerns a new sparse criterion proposed for the construction of the subsets of left singular vectors of the snapshots matrix constituting the RB. RB quality is in the literature assessed *offline* using and averaged approximation error over the training data set. Here, training data approximation error is proved to be insufficient when reported to RB's performances in the *online* reduction phase. For a minimal additional complexity, this contribution solves identified issues arising in the *online* reduction phase when using RB functions selected with the usual sequential truncation criterion in POD applications to transient structural dynamics problems. When using the sequential criterion, significant observations in the training data set are overshadowed by less meaningful, but more frequent ones. In transient problems, this behavior may result in RB that fails in the online reduction phase if, for example, the solution is poorly approximated at the time-step triggering the deformation. Rather than imposing a threshold on an averaged training data reconstruction error, as it is the case for the usual sequential singular value truncation criterion, the proposed sparse criterion imposes a threshold on each snapshot reconstruction error. On a FE crash-box benchmark, the sparse criterion is shown to circumvent this issue and to form RB exhibiting, at a given size, a better online performance.

The third contribution concerns the HR training phase with the ECSW method. The optimization problem arising in the *offline* training phase of a hyper-reduced

integration scheme is NP-hard and is, in the literature, sub-optimally solved in a greedy manner with the SNNLS algorithm. This procedure still turns out to be prohibitively expensive in computation time when training hyper-reduced integration schemes on large FE models, necessitating developments of alternative heuristics. In this context, there is a need for a reference solution for comparing different heuristics. The contribution proposes an alternative formulation of the ECSW optimization problem introducing Boolean selection variable taking account for finite element affiliation to the hyper-reduced integration scheme. The resulting optimization problem is solved using Mixed-Integer Programming (MIP), providing a reference for the validation of alternative suboptimal methods. Additionally, the impact of reducing the number of constraints and adding consistency conditions regarding polynomial integration, volume conservation, and weights minimal values is studied on an the reference example of a pierced plate in the traction model.

The manuscript is organized in the following manner. FE formulation used in industrial crashworthiness optimization is presented in Chapter 2, together with a brief state of the art of MOR techniques of interest in this context. Scientific contributions are developed in following the chapters: work on the incremental SVD for RB computation introduced in Chapter 3. Chapter 4 covers the proposed sparse criterion for the formation of the RB functions from left singular vectors and a new mixed formulation of the hyper-reduction optimization problem is presented in Chapter 5. Implementation and potential perspectives for robust integration of the method in industrial processes are discussed in Chapter 6. Notations and acronyms used throughout the manuscript are summarized in Appendix A and Appendix B, respectively.

1.4 Associated journal publications and conference papers

Scientific contributions of this thesis have been presented in academic proceedings, workshops, industrial events and published in international journals.

Different proceedings have been made in national and international academic events:

- Proceeding [13] on contribution presented in chapter 5 as been made at the 13th World Congress on Computational Mechanics (WCCM XIII).
- Presentation [59] on the modified incremental singular value decomposition of chapter 3 has been given at the 6th European Conference on Computational Mechanics (ECCM6) in Glasgow.
- The proposed sparse criterion of chapter 4 was first presented at the 14th French national congress on computational mechanics (CSMA 2019) in [61].
- Proceeding [65] on the sparse criterion at the ECCOMAS 4th Young Ivestigator Conference 2019 at Krakow [65].
- The global presentation [63] on contributions and outlooks for this work has been given at the 5th MORTech international Workshop in 2019.

We presented the work of this thesis in two industrial events:

- The NAFEMS 18 France Conference [60].
- The French Society of Automotive Engineers (SIA) 2019 numerical simulation congress [62].

Moreover, publications has been made in international scientific journal:

- Contribution on a sparse criterion, presented in chapter 4, has been published in the International Journal for Numerical Methods in Engineering in [64].
- Springer special issue

Chapter 2

State of the art

Last few decades, much attention has been being paid to the development of Model Order Reduction (MOR) methods to tackle the growing need for faster simulation in a wide variety of scientific fields. These methods rely on strong hypotheses to achieve drastic reductions in the number of unknowns of a model, thus focusing the computational effort on the Quantities of Interest (QoI). Depending on the targeted application, it is crucial to carefully choose the MOR method to apply, as: first, some methods require many developments and second, the funding hypothesis of the method may not be appropriate.

Two ways to categorize MOR methods are on 'a priori'/'a posteriori' and intrusive/non-intrusive. 'A posteriori' methods rely on previously collected data on the QoI to build a Reduced Order Model (ROM) whereas 'a priori' methods build one without any prior data on the QoI and rather rely on knowledge of the governing equations and numerical methods used in the model. Intrusive methods involve modifications in the source code of the simulation software in opposition to non-intrusive methods that use the simulation software as a black-box. An important variety of non-intrusive MOR methods is meta-modeling that use simulation results to fit a surrogate model of the Design of Experiment (DoE)(Response Surface (RS), Radial Basis Functions (RBF) [26], Kriging [23], Neural Network (NN) [57], ReCUR [50], Non-intrusive Proper Orthogonal Decomposition (NiPOD)([41, 15, 80]), thus, meta-modeling methods are 'a posteriori' non-intrusive methods. Among the intrusive MOR methods, the PGD [22, 49] approximates the QoI as a separate variable function and solves smaller independent models for each of those functions, enabling, once this work is done, the real-time evaluation of an approximated solution for any parameter configuration. Another class of intrusive methods is the Projected Reduced Order Model (PROM) that approximates the unknown as a linear combination of Reduced Basis (RB) vectors that replace a large number of shape functions originally used. A prevalent PROM method is the Proper Orthogonal Decomposition (POD) that originates from statistical data analysis [48] and found application in the analysis of turbulent flows [9]. This method build a RB from data to project the model on, yielding a ROM with fewer degrees of freedom. This method is often coupled with a method that approximates the projection of internal variables such as the Discrete Empirical Interpolation Method (DEIM) [17] and the Hyper-Reduction (HR) [73, 28].

Compared to intrusive methods, meta-modeling methods are widely used in the industry as they may easily be implemented by manufacturers without having access

to the source code of the simulation software. It is possible to couple meta-modeling and intrusive MOR methods to build a surrogate model upon a ROM simulation results. It is also possible to couple meta-modeling methods, an example of such coupling is the NiPOD: this method uses POD to reduce the dimensionality of collected simulation results and, in a second step, build a surrogate model for the reduced variables. A drawback of meta-modeling is that the physic of the problem is, in most cases, not taken into account in the surrogate model.

2.1 Brief review of Model Order Reduction methods

A large variety of reduction methods exist in the literature, all of which make strong hypothesis on the physical system to reduce its complexity.

2.1.1 Proper Generalized Decomposition

The Proper Generalized Decomposition (PGD) is an a priori model reduction method developed by F. Chinesta [22] and P. Ladevéze (LATIN-PGD) for nonlinear solid mechanics [49]. PGD approximates the solution as a finite sum of separate variables functions. Once the offline procedure complete, PGD allows real-time solving of the problem in the online phase by evaluating the basis functions for different parameter values. The method has found applications in a variety of scientific fields such as parametrized heat problem [84], parametric PDE [55], or computational rheology [21]. This method is very young, and its efficiency strongly depends on the separability of the variables.

This section briefly reviews the PGD and discusses further developments needed to apply the method to crash simulation.

Solution approximation

PGD central hypothesis is that the unknown solution $u(\{\mu\})$ of a physical problem, which depends on parameters $\{\mu\} = (\mu_1, \ldots, \mu_{n_p})^T$, may be approximated with the separate variables function

$$u(\{\mu\}) \approx \tilde{u}^{(k)}(\{\mu\}) = \sum_{i=1}^{k} \prod_{j=1}^{n_p} \phi_{i,j}(\mu_j), \tag{2.1}$$

of size k. Without loss of generality, all variables are considered as parameters. The PGD proceeds in two main steps: Basis functions $(\phi_j^i)_{i,j}$ are computed in the offline phase, and the solution (2.1) is approximated for any parameters value in the online phase. The offline phase is generally carried out by the Alternative Directions Scheme (ADS) algorithm reviewed in section 2.1.1. In the online phase, the method allows for a very fast approximation $\tilde{u}^{(k)}$ of the solution u in the online phase that may be carried out in real-time on an economic platform (phone, tablet, ...).

Alternative directions scheme

The trickiest part of the PGD method is the basis functions computation in the offline phase. This task may be carried out by the ADS algorithm, briefly presented

below, the interested reader is referred [22] where extensive information may be found.

The ADS proceeds by injecting approximation (2.1) into the equation weak form for increasing size $k = 1, \ldots$. First basis functions are initialized as constant functions, unknown functions in subsequent iterations are approached with polynomial functions. At the kth enrichment step $(\phi_{i,j})_{i,j}$ are known for $i \in [\![1, k-1]\!]$ and $j \in [\![1, \ldots, n_p]\!]$ and each function $(\phi_{k,j})_{j=1,\ldots,n_p}$ is iteratively approached with $(\phi_{n,j}^{(p)})_p$ in an inner loop on p. At this step all functions in the current weak form, except $\phi_{k,1}^{(2)}$, are known and are integrated. $\phi_{k,1}^{(2)}$ is then computed by resolving the resulting equation, using the FEM for example. The kth enrichment step continues with the other basis functions $(\phi_{k,j}^{(2)})_{j=2,\ldots,n_p}$. $(\phi_{k,j}^{(p-1)})_j$ are replaced one by one by $(\phi_{k,j}^{(p)})_j$, obtained by solving one-dimensional problems. Iterations on p in the kth enrichment step stops when Cauchy convergence is reached and $(\phi_{k,j}^{(p)})_p$ does not change anymore, $\forall j \in [\![1, k]\!]$. The same stopping criterion is used on the main loop, when further increasing approximation size k does not increase precision.

Application to car crash finite element simulation

The underlying dependence of PGD on the separability of the input space limits its application in vehicle crash to the separability of the parametric space. In the targeted nonlinear structural dynamics applications, the solution $u(\{x\}, t, \{\mu\})$ depends on the position $\{x\} = (x, y, z) \subset \Omega$, the time $t \in [0, T]$ and parameters $\{\mu\} = (\mu_1, .., \mu_{n_p})^T \in \mathbb{P}$. When the structure undergoes large deformations, space and time may not be separable, yielding the revised approximation

$$u(\{x\}, t, \{\mu\}) \approx \tilde{u}^{(N)}(\{x\}, t, \{\mu\}) = \sum_{i=1}^{N} \phi_i(\{x\}, t) \prod_{j=1}^{k} \psi_{i,j}(\mu_j). \tag{2.2}$$

This bottleneck may possibly be overcome by methods such as KPCA projecting the parameters space on a larger, possibly infinite-dimensional space in order to find hidden parameters, which will hopefully be separable (D. Gonzalez [33]).

In the ADS procedure, model parameters need to be explicitly expressed in the weak form, to be separately integrated with respect to all parameters. However, shape parameters or nonlinear material laws in specific regions of the structure may be hard to express analytically in the weak form of the equations.

Most importantly, the PGD induces the development of new software, which is beyond the scope of this thesis.

As a consequence, the PGD will not be considered in this work.

2.1.2 Equivalent Static Load Method

The Equivalent Static Load Method (ESLM) approximates the gradient of the function J to be optimized by approximating the primary system with a linear static one on which adjoint-state is performed. This method is described with more precision in [58, 31].

ESLM aims to speed-up the parametric shape optimization by giving access to the gradient. Rather than accelerating simulations, ESLM reduces the number of

simulations needed in the optimization process.

The semi-discretized equations used in the crash simulation are

$$[\mathbb{M}(\{\mu\})]\{\ddot{u}_{NL}(t,\{\mu\})\} + \{f_{int}(\{u(t,\{\mu\})\},t,\{\mu\})\} = \{f_{ext}(t,\{\mu\})\}, \tag{2.3}$$

where the dependency with respect to the parameters $\{\mu\} = (\mu_1, \ldots, \mu_{n_p})^T$ has been explicitly written. Usual parameters in the targeted application are material properties, shell thicknesses, nodes positions and CAD parameters. The tangent stiffness matrix $[\mathbb{K}_L(\{u(t,\{\mu\})\},t,\{\mu\})] = \frac{\partial\{f_{int}\}}{\partial\{u\}}(\{u(t,\{\mu\})\},t,\{\mu\})$ is defined at the current set of parameters $\{\mu\}$.

The equivalent linear static problem (2.4)

$$[\mathbb{K}_L(\{u(t,\{\mu\})\},t,\{\mu\})]\{u_L(\{\mu\})\} = \{f_{eq}(t,\{\mu\})\} \tag{2.4}$$

is written at time t. The displacement $\{u\}$ is conserved at time t in the equivalent linear static problem, explicitly

$$\{f_{eq}(t,\{\mu\}) := [\mathbb{K}_L(\{u(t,\{\mu\})\},t,\{\mu\})]\{u(t,\{\mu\})\} \quad \underset{(2.4)}{\Rightarrow} \quad \{u(t,\{\mu\})\} = \{u_L(\{\mu\})\}. \tag{2.5}$$

$[\mathbb{K}_L(\{u(t,\{\mu\})\},t,\{\mu\})]$ is computed at first time step and may be regularly updated at subsequent simulation times, which is strongly encouraged for large strain applications.

ESLM Algorithm

The ESLM algorithm 1 generates a parameter sequence $(\{\mu\}^k)_k$ converging towards the final design.

Algorithm 1: ESL Algorithm

```
Function [P] = ESLM
P^1 optimization starting point;
for k=1:Itermax do
    Solve [M({μ^(k)})]{ü_NL(t,{μ^(k)})} + {f_int({u(t,{μ^(k)})},t,{μ^(k)}) =
      {f_ext(t,{μ^(k)})};
    Evaluate J({μ^(k)},{u({μ^(k)},t)});
    Compute {f_eq(t,{μ^(k)})} = [K_L({μ^(k)})]{u({μ^(k)},t)} at different
      simulation times;
    Compute the new equivalent static linear model;
    Compute the Gradient on the design domain (see Gradient computation
      Section);
    Descend to {μ^(k+1)}; % Gradient descent, Quasi-Newton Method, etc. ...

    Test optimization shutoff parameter;
end
```

Gradient computation

The gradient is approximated with

$$\frac{dJ_{NL}}{d\{\mu\}} = \frac{\partial J_{NL}}{\partial\{\mu\}} + \frac{\partial J_{NL}}{\partial\{u\}}\frac{\partial\{u\}}{\partial\{\mu\}} \approx \frac{\partial J_{NL}}{\partial\{\mu\}} + \frac{\partial J_L}{\partial\{u_L\}}\frac{\partial\{u_L\}}{\partial\{\mu\}}, \tag{2.6}$$

where $J_{NL} := J(\{u(\{\mu\},t)\},\{\mu\})$ and $J_L := J(\{u_L(\{\mu\})\},t)$. The last term in (2.6) is obtained by performing the adjoint-state method on the linear static model. Rewriting (2.4) into the following state equation

$$G_{eq}(\{u_L(\{\mu\})\},\{\mu\}) = [K_L(\{\mu\})]\{\{u_L(\{\mu\})\} - \{f_{eq}(t,\{\mu\})\} = 0$$

leads to

$$\frac{dJ_{NL}}{d\{\mu\}} \approx \frac{\partial J_{NL}}{\partial\{\mu\}} + \lambda_L^T\frac{\partial G_{eq}}{\partial\{\mu\}}$$

where λ_L is the adjoint state obtained by solving the following adjoint-state equation (2.7).

$$[\frac{\partial G_{eq}}{\partial\{\mu\}}]^T\lambda_L = -[\frac{\partial J_L}{\partial\{\mu\}}]^T \tag{2.7}$$

Explanation :

Using the Lagrangian function

$$\mathcal{L}(\{u_L\}(\{\mu\}),\{\mu\}) = J_L(\{u_L(\{\mu\})\},\{\mu\}) + \lambda^T G_{eq}(\{u_L(\{\mu\})\},\{\mu\})$$

with

$$\frac{d\mathcal{L}}{d\{\mu\}} = (\frac{\partial J_L}{\partial\{\mu\}} + \lambda_L^T\frac{\partial G_{eq}}{\partial\{\mu\}}) + (\frac{\partial J_L}{\partial\{u_L\}} + \lambda_L^T\frac{\partial G_{eq}}{\partial\{u_L\}})\frac{\partial\{u_L\}}{\partial\{\mu\}} = 0 \tag{2.8}$$

and choosing λ_L satisfying (2.7) leads to $\frac{\partial J_L}{\partial\{u_L\}}\frac{\partial\{u_L\}}{\partial\{\mu\}} = -\lambda_L^T\frac{\partial G_{eq}}{\partial\{u_L\}}\frac{\partial\{u_L\}}{\partial\{\mu\}}$. As $\frac{dG_{eq}}{d\{\mu\}} = \frac{\partial G_{eq}}{\partial\{u_L\}}\frac{\partial\{u_L\}}{\partial\{\mu\}} + \frac{\partial G_{eq}}{\partial\{\mu\}} = 0$ one may write $\frac{\partial J_L}{\partial\{u_L\}}\frac{\partial\{u_L\}}{\partial\{\mu\}} = \lambda_L^T\frac{\partial G_{eq}}{\partial\{\mu\}}$.

Remarks and method improvement

This method has been successfully applied to car crash shape optimization for nodal intrusion and energy criterion objectives in [58, 31], providing sensibility mapping on an example such as hollow beam shape optimization for energetic absorption in a situation of impact with a rigid wall, which is illustrated on Figure 2.1.

In order to get a better gradient approximation and ensure the constraint equality between the analysis and design domains, one may want to use the secant stiffness matrix $[\mathbb{K}_S]$ in place of $[\mathbb{K}_L]$. The secant stiffness matrix is obtained by computing the Young secant modulus for each element in the model, elemental stiffness matrices, and assembling it. Equation (2.4) rewrites

$$[K_S^s(\{\mu\})]\{\{u_L\}(\{\mu\})\} = \{f_{eq}(t,\{\mu\})\}.$$

Moreover, the energy absorption is to be considered in vehicle crashworthiness optimization. To this end, nodal positions derivatives with respect to the CAD control points positions need to be computed, and the secant stiffness matrix needs to be

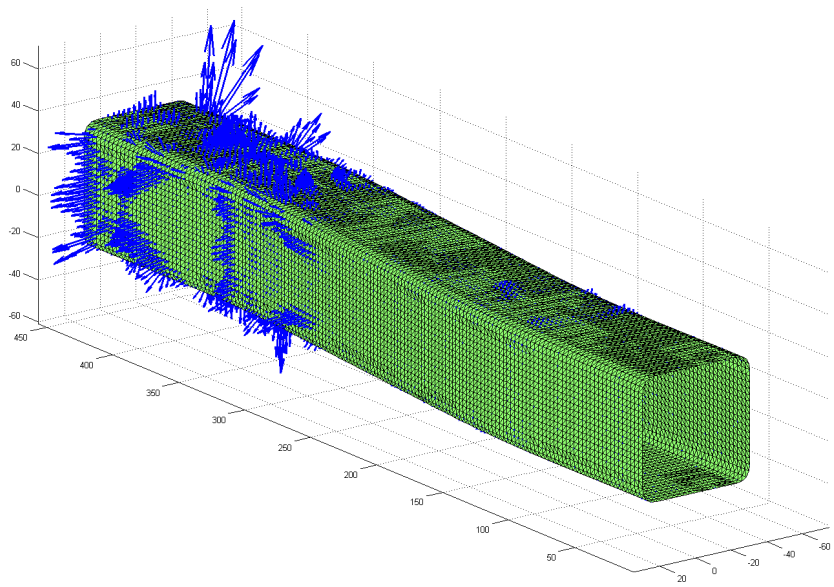


Figure 2.1: Sensitivity mapping of the of the energetic criteria wit respect to each note (seen in [31])

used to ensure constraint equality between the two domains.

The main advantage of this method is its facility to deal with optimization based on a large number of parameters and quickly provide a sensitivity mapping that does not need a parametrized CAD (see Figure 2.1 taken from [24]). However, it is essential to notice that the gradient computed using the equivalent static problem is an approximation of the real one. Hence it may not fit to a Quasi-Newton gradient optimization method or other methods based on Hessian approximation, leading to a very coarse approximation. Further developments are required to extend the method to multi-objective optimization and criterion using velocity and upper derivatives such as Head Injury Criterion (HIC). The mathematical equivalence between the dynamic and the equivalent static problems has not been demonstrated yet. A criterion showing the loss of proportionality in the loading has to be found in order to determine when the gradient approximation may not be used anymore for the descent direction computation in the optimization phase.

2.1.3 Projection-based Reduced Order Modeling

In the Finite Element (FE) method, the discrete solution $\{u\} \in \mathbb{R}^N$ is expressed as a linear combination of shape functions bounded to a single node of the mesh and locally defined in each element of the model. Projection-based Reduced Order Modeling (PROM) methods link between the local and the global formulations. PROM methods approximate the solution $\{u\}$ with a linear combination $\{\tilde{u}\}$ of global, domain spanning, Reduced Basis (RB) functions $(\{\phi_i\})_{1\leq i\leq k} \subset \mathbb{R}^N$,

$$\{u(t)\} \approx \{\tilde{u}(t)\} = \sum_{i=1}^{k} \alpha_i(t)\{\phi_i\}, \quad \forall t \in [0, T], \tag{2.9}$$

making the hypothesis that the Full Order Model (FOM) solution $\{u\}$ may be correctly approximated with a small number k of RB functions with respect to the original model size N, and thus reducing the number of unknowns.

Numerous methods address the computation of the RB, mostly based on dimensionality reduction, or low-rank approximation, methods, acting *a posteriori* on gathered training data. The Dynamic Mode Decomposition (DMD) [72, 74, 20, 79, 2] uses transient data to approximate the modes and frequencies associated with the Koopman operator. The Koopman operator [47] describes the dynamical evolution of observables in a physical system. A RB is formed using approximated modes. The Proper Orthogonal decomposition (POD), also known as Principal Component Analysis (PCA), computes the vector space of a given dimension that best approximates a set of observations. A basis of this space is used to form an orthogonal RB. Several variations of this method, such as the Balanced POD (BPOD), the Smooth Orthogonal Decomposition (SOD), have been proposed.

2.1.4 Dimensionality reduction methods

A non-exhaustive list of dimensionality reduction methods used in *a posteriori* MOR methods is presented in this section.

The singular value decomposition

A popular method used in building a low-rank approximation of the snapshot matrix is the Singular Value Decomposition (SVD), also known as the Principal Component Analysis (PCA). The thin SVD is a generalization of the eigenvalue decomposition for non-square matrices. Given a snapshot matrix $[S] = [\{u_1\}, \{u_2\}, \ldots, \{u_{n_s}\}] \in \mathbb{R}^{N \times n_s}$, where N stands for the data dimension and n_s the number of snapshots in the training data set, its singular value decomposition writes:

$$[S] = [\Phi][diag(\{s\})][\Psi]^T, \tag{2.10}$$

with $[\Phi] = [\{\phi_1\}, \{\phi_2\}, \ldots, \{\phi_m\}] \in \mathbb{R}^{N \times m}$ the matrix of left singular vectors, $[\Psi] = [\{\psi_1\}, \{\psi_2\}, \ldots, \{\psi_{n_s}\}] \in \mathbb{R}^{n_s \times m}$ the matrix of right singular vectors, $\{s\} = (s_1, s_2, \ldots, s_m)^T \in \mathbb{R}^m$ the vector of associated singular values and $m = min(N, n_s)$. $[\Phi]$ and $[\Psi]$ are both orthonormal matrices. Unlike the regular SVD, in which N left singular vectors are computed in $[\Phi]$, the thin SVD only computes m left singular vectors. Approximating a data set of size n_s, the regular SVD entails unnecessary computations if $n_s > N$. Thus, the thin SVD is preferred for data approximation. In the manuscript, what is referred to as SVD is the thin SVD.

The RB $[\Phi^{(k)}] = [\{\phi_1\}, \{\phi_2\}, \ldots, \{\phi_k\}] \in \mathbb{R}^{N \times k}$ of size k is obtained by only keeping the first k columns of $[\Phi]$. The left singular vectors in decomposition (2.10) being sorted out in descending order of associated singular values, $[\Phi^{(k)}]$ spans by construction the vector space of dimension k that best approximates the training data set $[S]$ in the sense that it minimizes the error of projection in the Frobenius norm. More precisely,

$$\frac{\|[S] - [\Phi^{(k)}][\Phi^{(k)}]^T[S]\|_F}{\|[S]\|_F} = \min_{[M] \in O(N,k)} \{\frac{\|[S] - [M][M]^T[S]\|_F}{\|[S]\|_F}\}, \tag{2.11}$$

with $\|[\bullet]\|_F = \sqrt{\sum_{i,j}[\bullet]_{ij}^2}$ the Frobenius norm and $O(N,k) := \{[M] \in \mathbb{R}^{N\times k} \mid [M]^T[M] = [Id]\}$ the space of orthogonal matrices of dimensions N by k. Approximation error (2.11) corresponds to the singular value error of truncation

$$\sqrt{\frac{\sum_{i=k+1}^{n_s} s_i^2}{\sum_{j=1}^{n_s} s_j^2}} = \frac{\|[S] - [\Phi^{(k)}][\Phi^{(k)}]^T[S]\|_F}{\|[S]\|_F}, \tag{2.12}$$

making the truncation criterion easy to use in practice.

Variations of the singular value decomposition

The SVD, presented in section 2.1.3, is based on the eigenvalue decomposition of the covariance matrix $[\Theta] = [S]^T[S]$. Variations of the SVD consist in modifying the scalar product used in the computation of these covariance matrices. In its unaltered version, the coefficient (i,j) of the covariance matrix is computed using the scalar product associated with the L^2-norm:

$$[\Theta]_{i,j} = ([S]^T[S])_{i,j} = \langle \{u_i\}, \{u_j\}\rangle_{L^2(\mathbb{R}^N)} := \sum_{k=1}^{N} \{u_i\}_k \{u_j\}_k, \tag{2.13}$$

With $\langle \bullet, \bullet\rangle_{L^2(\mathbb{R}^N)}$ the scalar product in $L^2(\mathbb{R}^N)$, the space of square-integrable functions. A first variation consists in replacing the scalar product in $L^2(\mathbb{R}^N)$ with the scalar product in $H_1(\mathbb{R}^N)$

$$\langle x, y\rangle_{H_1(\mathbb{R}^N)} := \langle x, y\rangle_{L^2(\mathbb{R}^N)} + \langle \nabla x, \nabla y\rangle_{L^2(\mathbb{R}^N)}, \quad \forall (x,y) \in \mathbb{R}^N \times \mathbb{R}^N. \tag{2.14}$$

Another possible variation of the SVD is the Kernel Principal Component Analysis (KPCA) presented hereafter.

Kernel principal component analysis

KPCA replaces the scalar product used in the SVD with a reproducing kernel inducing a mapping function that implicitly projects the data set in a vectorial space of higher, possibly infinite, dimension D in which the covariance matrix is decomposed.

Reproducing kernel:
$\kappa : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$ is a reproducing kernel if:

$$\begin{cases} \kappa(\{x\}, \bullet) \in L^2(\mathbb{R}^N), \quad \forall \{x\} \in \mathbb{R}^N \\ \langle f(\bullet), \kappa(\{x\}, \bullet)\rangle_{L^2(\mathbb{R}^N)} = f(\{x\}), \quad \forall f \in L^2(\mathbb{R}^N), \quad \forall \{x\} \in \mathbb{R}^N \end{cases}$$

Mercer's theorem :
Let $\kappa(.,.)$ a reproducing kernel. If κ is continuous, symmetric and positive definite, then it induces a mapping $\theta : \mathbb{R}^N \to \mathbb{R}^D$ such that $\langle \theta(\{x\}), \theta(\{y\})\rangle_{L^2(\mathbb{R}^D)} =$

$\kappa(\{x\},\{y\})$ for all $(\{x\},\{y\}) \in \mathbb{R}^N \times \mathbb{R}^N$

The Mercer's theorem is the cornerstone of the KPCA as it allows to computes the covariance matrix of the data set image by the mapping θ in $\mathbb{R}^D$, which may be infinite-dimensional, without knowing the mapping explicitly. Explicitly: $[\Theta]_{i,j} = \kappa(\{u_i\},\{u_j\}), \quad \forall (i,j) \in [\![1, n_s]\!]$. The idea behind KPCA is that the data set may not be separable in $\mathbb{R}^N$ but in the higher-dimensional space $\mathbb{R}^D$. The usual kernel basis functions are the following.

- $\kappa(\{x\},\{y\}) = exp(-\frac{||\{x\}-\{y\}||^2}{2\omega}), \quad \forall(\{x\},\{y\}) \in \mathbb{R}^{n^2}, \quad \omega \in \mathbb{R}^+$, radial kernel
- $\kappa(\{x\},\{y\}) = exp(-r||\{x\}-\{y\}||^2), \quad \forall(\{x\},\{y\}) \in \mathbb{R}^{n^2}, \quad r \in \mathbb{R}^+$, extended radial kernel
- $\kappa(\{x\},\{y\}) = (1 + \langle\{x\},\{y\}\rangle)^c, \quad \forall(\{x\},\{y\}) \in \mathbb{R}^{n^2}, \quad c \in \mathbb{N}^*$, polynomial kernel
- $\kappa(\{x\},\{y\}) = tanh(\langle\{x\},\{y\}\rangle + b), \quad \forall(\{x\},\{y\}) \in \mathbb{R}^{n^2}, \quad b \in \mathbb{R}$, sigmoid kernel

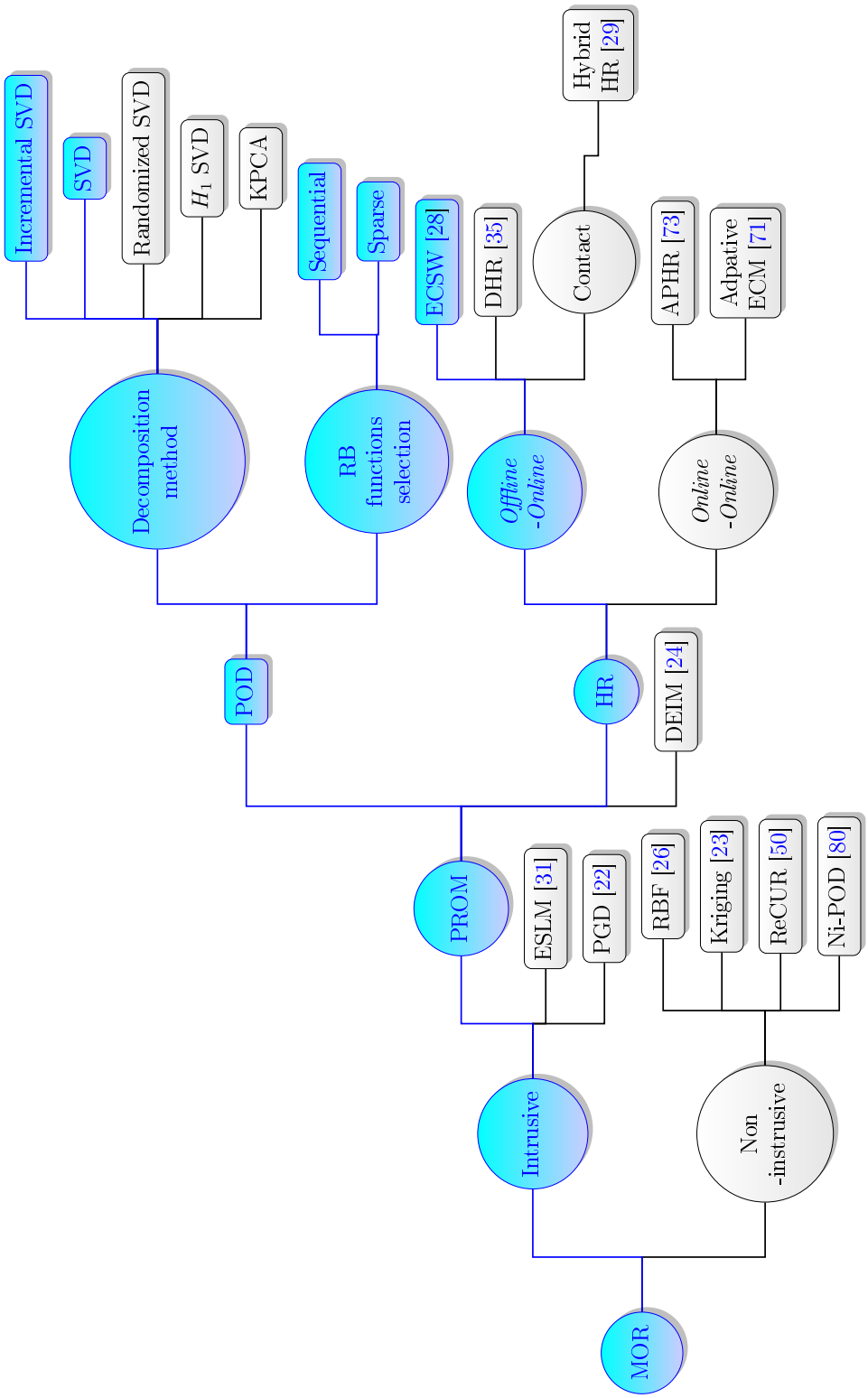
The mapping associated with a kernel basis function is closely related to its Taylor expansion. Several variations of this method may be found in the literature. The interested reader may refer to [43] for more details on the KPCA and applications to parameter space reduction.

2.1.5 Hyper-reduction

PROM applications to nonlinear problems generally yield a computational overhead in evaluating nonlinear internal variables. As a matter of fact, those variables may not be linearly projected on the reduced space of the PROM method. The full-scale solution's approximation needs to be reconstructed ahead of each nonlinear internal variables computation, which is then projected back on the reduced model, inducing more computations with respect to the unreduced model.

Different HR methods have been being developed over the last two decades [35, 73, 28] to circumvent this bottleneck, together with alternatives such as the Discrete Empirical Interpolation (DEIM) [17, 24, 77] and the magic points [51]. These methods share the same philosophy of computing the minimal required information of the full-scale internal variables while ensuring a good approximation of it once projected on the reduced space of the PROM method.

2.1.6 Summary


Incremental SVD
SVD
Randomized SVD
H_1 SVD
KPCA
Sequential
Sparse
ECSW [28]
DHR [35]
Contact
Hybrid HR [29]
APHR [73]
Adpative ECM [71]
Decomposition method
RB functions selection
Offline -Online
Online -Online
POD
HR
DEIM [24]
PROM
ESLM [31]
PGD [22]
RBF [26]
Kriging [23]
ReCUR [50]
Ni-POD [80]
Intrusive
Non -instrusive
MOR

2.2 Details on the chosen model order reduction methods

A detailed review of the reduction methods used in this thesis is proposed in this section in the scope of applications to nonlinear explicit structural dynamics. Focus is placed on PROM methods, in which a RB is used to project the FE model on, coupled with HR, which reduces the computational effort associated with nonlinear internal forces in order to achieve global speed-up in nonlinear applications. More specifically, the RB and associated hyper-reduced integration scheme are computed with the POD and the ECSW methods, respectively.

The FE formulation used in vehicle crash simulation is presented in section 2.2.1, the POD in section 2.2.3 and the ECSW in section 2.2.4.

2.2.1 Finite element formulation of the problem

In nonlinear structural dynamics, the semi-discretized FE formulation takes the following form

$$[\mathbb{M}]\{\ddot{u}(t)\} + \{f_{int}(\{u(t)\}, t)\} = \{f_{ext}(t)\}, \tag{2.15}$$

where $\{u(t)\} \in \mathbb{R}^N$ is the displacement unknown at time $t \in [0, T]$ and consists of a vector of nodal displacements at each node and in each direction. N denotes the number of degrees of freedom (DoF). $[\mathbb{M}] \in \mathbb{R}^{N \times N}$ is symmetric positive definite the mass matrix.

In the MOR field, equations (2.15) are often referred to as the Full Order Model (FOM). The considered FOM may depend on n_p parameters $(\mu_i)_{1 \leq i \leq n_p}$. In vehicle crashworthiness optimization, those parameters concern material properties and shell thicknesses. However, for simplicity, the dependency to those parameters will often be omitted.

2.2.2 *A posteriori* model order reduction methods

The POD and ECSW are *a posteriori* methods which rely on observations, called snapshots, taken from previous full-scale simulations to train a ROM used online to reduce the computation time of subsequent simulations. The global workflow for *a posteriori* methods is sketched in Figure 2.2.

In the literature, these methods are also referred to as *Offline/Online* methods. They proceed in two main phases: the ROM is computed in the *offline* training phase and used *online* to accelerate computations.

In the *offline* phase, snapshots are gathered and post-processed to obtain a ROM consisting of a reduced representation of the collected data. While these snapshots may originate from any FE model or experiment, it is essential that they are all expressed on the same mesh ahead of the ROM training step.

The ROM is then used *online* to reduce the complexity of one or multiple FE models. Approximation quality depends both on the collected data relevancy regarding the target *online* application and methods used in the ROM training step. Computational speed-up of the reduced model is the ratio between the FOM and the ROM computation times. Nevertheless, the *offline* observation and training steps induce a computational overhead that needs to be compensated *online*.

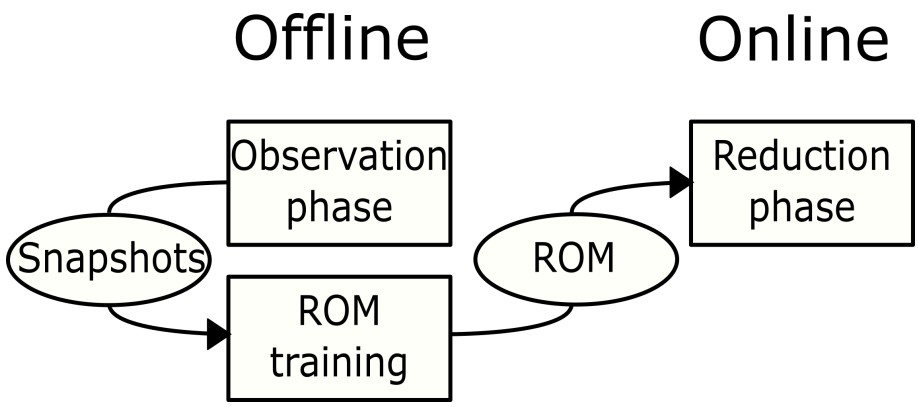


Figure 2.2: Global workflow for *a posteriori* MOR methods

2.2.3 Proper Orthogonal Decomposition

PROM performances depend on the RB capability to approximate the unknown FOM solution at all simulation times with a minimal number of functions. While the elemental shape functions used in the FOM allow the representation of any solution on the mesh, RB functions will constrain the solution to a lower-dimensional space, which may not contain the FOM solution. Thus, defining appropriate RB functions is concomitantly crucial and intricate. The Proper Orthogonal Decomposition (POD) uses *relevant a priori* collected data to construct the RB subsequently used to build the projected reduced order model (2.18). POD decomposes in two phases: collecting the data, referred to as snapshots, and computing the RB performed in the *offline* training phase, followed by the *online* reduction phase consisting in solving the resulting PROM (2.18).

In the *offline* phase, n_s snapshots $(\{u_i\})_{1\leq i\leq n_s}$ are gathered and organized in the snapshot matrix $[S] = [\{u_1\}, \ldots, \{u_{n_s}\}] \in \mathbb{R}^{N\times n_s}$. Various ways of obtaining those snapshots may be considered. They may originate from real experiments. In parametric optimization they may originate from simulations performed for a DoE. Snapshots may as well originate from previous simulation time in an online adaptive approach, or from previous vehicle models if a part of the vehicle is carried over the new model. The last two possibilities are discussed in chapter 6. While snapshots' origin depends on the process in which the POD method is used. Nevertheless, it is mandatory for snapshots to be defined on the mesh of the model to reduce.

The RB is obtained in the *offline* training phase computing a low-rank representation of previously collected snapshots. $[S]$ is approximated up to a certain precision ϵ_{rb} with the minimal number of basis vectors with the SVD. Variations of the SVD and alternative matrix approximation methods may be used in place of the SVD, those methods are discussed in section 2.1.4. RB functions and associated coefficients are arranged in the RB $[\Phi^{(k)}] = [\{\phi_1\}, \ldots, \{\phi_k\}] \in \mathbb{R}^{N\times k}$ and the vector of unknowns $\{\alpha\} = (\alpha_1, \ldots, \alpha_k)^T$, respectively, yielding the condensed form of (2.9)

$$\{u(t)\} \approx \{\tilde{u}(t)\} = [\Phi^{(k)}]\{\alpha(t)\}. \tag{2.16}$$

Injecting approximation (2.16) in the FOM (2.15) and a using a Galerkin projection on the reduced space spanned by the columns of $[\Phi^{(k)}]$ yields the so called

Reduced Order Model solved in the *online* reduction phase

$$[\Phi^{(k)}]^T[\mathbb{M}][\Phi^{(k)}]\{\ddot{\alpha}(t)\} + [\Phi^{(k)}]^T\{f_{int}([\Phi^{(k)}]\{\alpha(t)\}, t)\} = [\Phi^{(k)}]^T\{f_{ext}(t)\}. \tag{2.17}$$

Introducing notations

- $[\tilde{\mathbb{M}}] = [\Phi^{(k)}]^T[\mathbb{M}][\Phi^{(k)}] \in \mathbb{R}^{k\times k}$
- $\{\tilde{f}_{int}(\{\alpha(t)\}, t)\} = [\Phi^{(k)}]^T\{f_{int}([\Phi^{(k)}]\{\alpha(t)\}, t)\} \in \mathbb{R}^k$
- $\{\tilde{f}_{ext}(t)\} = [\Phi^{(k)}]^T\{f_{ext}(t)\} \in \mathbb{R}^k$

allows to rewrite (2.17) in the more convenient way

$$[\tilde{\mathbb{M}}]\{\ddot{\alpha}(t)\} + \{\tilde{f}_{int}(\{\alpha(t)\}, t)\} = \{\tilde{f}_{ext}(t)\}. \tag{2.18}$$

Important remarks on model (2.17) concern the stable time step and overcost in nonlinear applications using a lumped mass approach.

Stable time step

Projecting equations (2.15) using an orthonormal RB yields larger time step in model (2.18) when computed with the CFL condition. This is due to the spectral radius of $[\tilde{\mathbb{K}}]$ being smaller or equal to that of $[\mathbb{K}]$. The interested reader may refer to [7] for details and mathematical proof.

POD Computational overhead in nonlinear explicit applications

In the *online* reduction phase, when the nonlinear structural dynamics FE solver uses explicit time integration, POD generally yields a computational overhead for two reasons.

In nonlinear structural dynamics industrial FE solvers, the lumped mass approach yields a diagonal mass matrix $[\mathbb{M}]$ and no matrix inversion is needed in (2.15). Projecting the model on the reduced space yields a fully populated reduced mass matrix $[\tilde{\mathbb{M}}]$ in (2.18) which needs to be inverted. A solution is to use the metric associated with the symmetric positive definite mass matrix $[\mathbb{M}]$ in the SVD, as explained in section 2.2.3.

Also, the need to evaluate high dimensional approximations, turns out to be quite expensive in the case of nonlinear internal variables. Even though (2.18) unknown is $\{\alpha\}$, it is necessary to evaluate $\{\tilde{u}\} = [\Phi^{(k)}]\{\alpha\}$ to compute the internal forces at each time step. Depending on the complexity of the model it may as well be necessary to compute the high dimensional state variables approximations for contact interfaces, kinematic conditions, or outputs.

Symmetric positive-definite matrix metric

A simplification in the POD framework consists in using the metric implied by the mass matrix $[\mathbb{M}]$ of the FOM in the construction of the RB, resulting in the following identity:

$$[\tilde{\mathbb{M}}] = [\Phi^{(k)}]^T[\mathbb{M}][\Phi^{(k)}] = [Id_k]$$

with $[Id_k]$ the identity matrix. The same goes for any columns selection of $[\Phi]$. This is achieved by the following steps:

- premultiply all observation with $[\mathbb{M}]^{\frac{1}{2}}$

$$[\bar{S}] = [\mathbb{M}]^{\frac{1}{2}}[S],$$

- compute the singular value decomposition of $[\bar{S}]$

$$[\bar{S}] = [\bar{\Psi}]diag(\{\bar{s}\})[\bar{\Psi}]^T,$$

- multiply the resulting subspace with $[\mathbb{M}]^{-\frac{1}{2}}$

$$[\Phi] = [\mathbb{M}]^{-\frac{1}{2}}[\bar{\Phi}],$$

$$\{s\} = \{\bar{s}\},$$
$$[\Psi] = [\bar{\Psi}].$$

This way the following identity is fulfilled

$$[\Phi]^T[\mathbb{M}][\Phi] = [\bar{\Phi}]^T[\mathbb{M}]^{-\frac{1}{2}}[\mathbb{M}][\mathbb{M}]^{-\frac{1}{2}}[\bar{\Phi}] = [Id].$$

Finally, the semi-discretized ROM equation (2.18) rewrites

$$\boxed{\{\ddot{\alpha}(t)\} = [\Phi]^T(\{\tilde{f}_{ext}(t)\} - \{\tilde{f}_{int}(\{\alpha(t)\}, t)\}).} \tag{2.19}$$

When using such a trick, there are two possibilities in writing the resulting reduced order approximation $\{\tilde{u}\} \in \mathbb{R}^N$ of a state variable $\{u\} \in \mathbb{R}^N$. One possibility is to use the orthogonal projection and write

$$\{u\} \approx \{\tilde{u}\} = [\Phi]([\Phi]^T[\Phi])^{-1}[\Phi]^T\{u\}$$

which is the general case. It holds for orthonormal RB $[\Phi]^T[\Phi] = [Id]$ or any real matrix $[\Phi] \in \mathbb{R}^{N\times k}$ since $[\Phi]^T[\Phi]$ is a real symmetric matrix and is thus always reversible. This choice of approximation is consistent with the Euclidean norm error choice. However the need to reverse $[\Phi]^T[\Phi]$ makes the following oblique projection preferable.

$$\{u\} \approx \{\tilde{u}\} = [\Phi][\Phi]^T[\mathbb{M}]\{u\}$$

2.2.4 Energy Conserving Sampling and Weighting

Hyper-reduction is very effective in nonlinear structural dynamics, where internal forces evaluation with the FEM represents more than half of the total computation time. Focus is placed on the Energy Conserving Sampling and Weighting (ECSW) HR method [28], chosen for its stability properties.

The Galerkin method used in FE analysis in the divide and conquer spirit successively computes internal forces $\{f^e_{int}\} \in \mathbb{R}^N$ in each of n_e elements of the model and assembles respective contributions

$$\{f_{int}\} = \sum_{e=1}^{n_e}\{f^e_{int}\} \tag{2.20}$$

In the ROM (2.19), internal forces are projected on $[\Phi] \in \mathbb{R}^{N\times k}$

$$[\Phi]^T\{f_{int}\} = [\Phi]^T \sum_{e=1}^{n_e} \{f_{int}^e\}. \tag{2.21}$$

Hyper-reduction computes internal forces only for a subset $\mathcal{H} \subset [\![1, n_e]\!]$ of elements indexes and applies weights ζ_e^* to the elemental contributions ahead of summation and projection on the reduced space

$$[\Phi]^T\{f_{int}\} \approx [\Phi]^T \sum_{e=1}^{n_e} \zeta_e^* \{f_{int}^e\} = [\Phi]^T \sum_{e\in\mathcal{H}} \zeta_e^* \{f_{int}^e\}, \tag{2.22}$$

where $\{\zeta\}^* = (\zeta_1^*, \zeta_2^*, \ldots, \zeta_{n_e}^*)^T \in \mathbb{R}_{\geq 0}^{n_e}$ contains weights associated with all elements in the model. $\zeta_e^* = 0$ if and only if element e is not selected ($e \notin \mathcal{H}$). The ECSW method imposes also $\zeta_e^* > 0$ for selected elements to maintain the integrator positivity.

The hyper-reduced integration scheme, given by the subset of selected elements $\mathcal{H}$ and associated weights $\{\zeta^*\}$, is obtained through optimization. Given a RB $[\Phi]$ and unassembled internal forces $(\{f_{int}^e(t_i)\})_{(e,i)\in[\![1,n_e]\!]\times[\![1,n_s]\!]}$, the hyper-reduced quadrature scheme integrates the projected unassembled internal forces training data set up, to a user-defined precision, while selecting the fewest possible elements in $\mathcal{H}$. First, n_s unassembled internal forces snapshots are collected at training times $(\{t_i\}, \quad i \in [\![1, n_s]\!]$. Without loss of generality, snapshots may as well be taken at different model parameters values. Once collected, unassembled internal forces $(\{f_{int}^e(t_i)\})_{(e,i)\in[\![1,n_e]\!]\times[\![1,n_s]\!]} \subset \mathbb{R}^N$ are projected on the reduced space, yielding $([\Phi]^T\{f_{int}^e(t_i)\})_{(e,i)\in[\![1,n_e]\!]\times[\![1,n_s]\!]} \subset \mathbb{R}^k$, and are organized in the matrix

$$[G] = \begin{bmatrix} G_f(t_1) \\ G_f(t_2) \\ \vdots \\ G_f(t_{n_s}) \end{bmatrix} \in \mathbb{R}^{k*n_s\times n_e}, \tag{2.23}$$

where

$$[G_f(t_i)] = [\Phi^T][\{f_{int}^1(t_i)\}, \{f_{int}^2(t_i)\}, \ldots, \{f_{int}^{n_e}(t_i)\}] \in \mathbb{R}^{k\times n_e}, \quad \forall i \in [\![1, n_s]\!]. \tag{2.24}$$

Using this notations, the non-reduced assembly process (2.20) writes

$$\{b\} = [G]\{1_{n_e}\} \in \mathbb{R}^{k*n_s}, \tag{2.25}$$

with $\{\zeta^*\} = \{1_{n_e}\}$ corresponding to the selection of all elements with integration weights equal to 1 and $\{b\} \in \mathbb{R}^{k*n_s}$ is the 'exact' projection of all internal forces snapshots on $[\Phi]$, used as reference to train the weights. Finally, given a targeted precision τ, the optimization problem of the hyper-reduced integration scheme is stated as

$$\begin{cases} \{\zeta^*\} = \underset{\{\zeta\}\in\mathcal{A}_{ECSW}}{argmin} \ (\|\{\zeta\}\|_0) \\ \mathcal{A}_{ECSW} = \{\{\zeta\} \in \mathbb{R}_{\geq 0}^{n_e} \mid \dfrac{\|[G]\{\zeta\} - \{b\}\|_2}{\|\{b\}\|_2} \leq \tau\} \end{cases} \tag{2.26}$$

The threshold τ on the approximation precision constraint is imposed in the admissible space $\mathcal{A}_{ECSW}$ alongside weights positivity, $\| \cdot \|_0$ denoting the zero-norm associating the number of its non-zero coefficients to a vector, equivalent to the number of selected finite elements, to be minimized. However, the zero norm is not differentiable, making (2.26) NP-hard. In practice (2.26) is suboptimally solved with greedy algorithms such as SNNLS ([28])(Algorithm 2). Alternatives such as the LASSO algorithm have been compared to SNNLS in [16].

Algorithm 2: SNNLS($[G]$,$\{b\}$, $\{\zeta\}$, τ)

$n = 0$ % number of selected elements
$\{h\} = \{0\}_N$ % vector of selected elements indices
$l1$ =TRUE
while ($l1$) **do**
  $\{\mu\} = [G]^T(\{b\} - [G]\{X\})$
  $i = max_value_index(\{\mu\})$
  $n = n + 1$
  $\{h\}(n) = i$
  $l2$ =TRUE
  while ($l2$) **do**
    $\{\zeta_{tmp}\} = \underset{\{\zeta\}\in\mathbb{R}^n}{argmin}(\|[G(:, \{h(1:n)\})]\{\zeta\} - \{b\}\|_2)$ % solved with least square
    if ($\{\zeta_{tmp}\} > 0$) **then**
      $\{\zeta(\{h(1:n)\})\} = \{\zeta_{tmp}\}$
      $l2$ =FALSE
    else
      $\alpha = \underset{\{i|\{\zeta\}(\{h\}(i))>\{\zeta_{tmp}\}(i)\}}{min}(\frac{\{\zeta\}(\{h\}(i))}{\{\zeta\}(\{h\}(i))-\{\zeta_{tmp}\}(i)})$ % least feasible step to keep the solution positive
      $\{\zeta(\{h(1:n)\})\} = \{\zeta(\{h(1:n)\})\} + \alpha(\{\zeta_{tmp}\} - \{\zeta(\{h(1:n)\})\})$
      Recompute $\{h\}$ and n (eventual zeroed values)
    end
  end
  if $((n \geq size([G], 2))\ OR\ (\frac{\|[G]\{\zeta\}-\{b\}\|_2}{\|\{b\}\|_2} \leq \tau))$ **then**
    $l1 =$ FALSE
  end
end

2.3 Problematics and contributions

This thesis focuses on PROM for industrial applications to nonlinear transient structural dynamics. RBs are computed with the POD, and the projection method is coupled with the ECSW HR method to reduce nonlinear internal forces computation cost and achieve speed-up in the online reduction phase. Implementations are performed in the legacy FE solver [3] with full access to the source code. To our best knowledge, few or no works achieve global speed-up in an industrial FE solver.

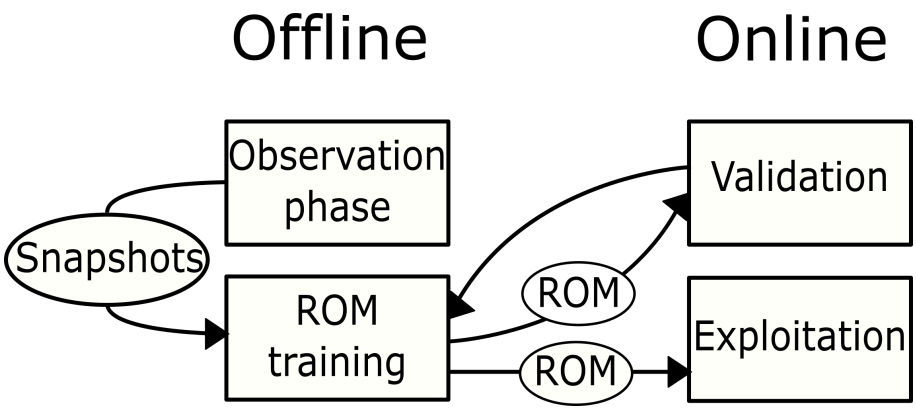


Figure 2.3: *Online* validation of the ROM *offline* training phase.

The ROM is computed *a posteriori* of data gathering in the offline training phase. Both online ROM's approximation quality and computational speed-up depend on the post-processing phase as well as training data relevancy regarding the targeted FE model to reduce. While the latter is in our work ensured by using training data originating from the FE model, issues still arise in the offline training phase regarding computation time, optimality, and *a priori* online approximation error indicators.

The following bottlenecks to the robust application of PROM and HR to industrial vehicle crashworthiness optimization are identified :

- The algorithmic complexity associated with the training phase skyrockets with increasing training data size. In order to achieve global speed-up, the offline computational effort needs to be controlled.

- The ROM approximation quality is assessed in the *offline* training phase using error indicators on the training data reconstruction. In practice, these error indicators are insufficient to assess the *online* approximation quality of a ROM.

- Compatibility issues arise between the HRPOD options, optimization, and parallelization specific to industrial solvers. Some of these options, such as weld spots, require full-scale computations of internal forces. Optimization and parallelization of the code make it challenging to access unassembled internal forces mandatory for HR training.

- None of the reduction methods reviewed in this chapter permit to reduce the contact formulation used in crash models, significantly limiting the computational speed-up on some complex models involving many contacts.

Int the present work, the ROM *offline* training phase is evaluated using an *online* validation phase ahead of subsequent use of the resulting ROM in the *online* exploitation phase, as illustrated in Figure 2.3. This approach is used in the following three contributions of this thesis, addressing the identified bottlenecks of the HRPOD *offline* training phase in applications to large-scale nonlinear structural dynamics models.

2.3.1 'On-the-fly' snapshots selection (Chapter 3)

Over the last few decades, Reduced Order Modeling (ROM) has slowly but surely inched towards widespread acceptance in computational mechanics, as well as other simulation-based fields. These methods rely on the construction of an appropriate Reduced Basis (RB), typically based on a low-rank representation of a set of "observations" made using full-field simulations. The RB is usually obtained through truncated Singular Value Decomposition (SVD). However, SVD encounters limitations when dealing with a large number of high-dimensional observations, requiring the development of alternatives such as the incremental SVD. The key advantages of this approach are a reduced computational complexity and memory requirement compared to a regular "single-pass" spectral decomposition. These are achieved by using observations to enrich the low-rank representation as and when available, to avoid having to store them. In addition, the RB may be truncated 'on-the-fly' to reduce the size of the matrices involved as much as possible and, by doing so, avoid the quadratic scale-up in computational effort with the number of observations. This chapter presents a new error estimator for the incremental SVD, which is shown to be an upper bound for the approximation error, and proposes an algorithm to perform the incremental SVD truncation and observation selection "on-the-fly", instead of tuning a prohibitively large number of frequently "hard to set" parameters. The approach, implemented in [3], is applied to the Finite Element (FE) model simulation of impact on a Taylor beam.

Key Words : Singular Value Decomposition, Principal Component Analysis, low-rank representation, snapshot selection, Model Order Reduction, Proper Orthogonal Decomposition, vehicle crash simulation

2.3.2 Sparse POD modal subsets (Chapter 4)

Projected Reduced Order Methods (PROM) such as the Proper Orthogonal Decomposition (POD), rely on the quality of the underlying Reduced Basis (RB) used to approximate the solution. The RB is generally constructed by a low-rank approximation of a set of observations, taken from full-scale simulations, through truncated Singular Value Decomposition (SVD) of the snapshot matrix. This chapter revisits the selection criterion of the RB functions in the study of dynamical systems. In opposition to truncating the set of left singular vectors, taken consecutively in decreasing order of associated singular values, the proposed method takes temporality into account, resulting in a compact, sparse subset of RB functions. Selection strategies, implemented in the reduced-order version of a legacy nonlinear explicit dynamics Finite Element (FE) code [3], are compared in both offline and online phases in terms of work of internal forces reconstruction error.

Key Words : Model Order Reduction, Proper Orthogonal Decomposition, modal selection, truncated Singular Value Decomposition, low-rank approximation, vehicle crash simulation.

2.3.3 MIP Hyper-reduction (Chapter 5)

The hyper-reduction problem for reduced-order internal forces evaluation in transient, nonlinear, explicit dynamics is reformulated, employing Mixed-Integer Programming (MIP), taking into account consistency constraints. Constraint reduction is introduced. Resulting quadratures, as well as reduced runs, are compared against the standard Energy Conserving Sampling and Weighting (ECSW) scheme, on a reference example. Rather than searching for optimal performance, the goal is to provide a benchmark solution, within a custom, Projected Reduced-Order Method (PROM) adaptation of an industrial-level legacy finite element code [3], for evaluation of heuristic hyper-reduction formulations along with a non-greedy approach.

Key Words : Finite Element Method, Model Order Reduction, crashworthiness optimization, Hyper Reduction, Internal Forces, SNNLS, LASSO, MIP.

2.4 Organization of the manuscript

Chapters 3, 4 and 5 concern the three contributions of our work which gave rise to scientific communications and include extra details with respect to the corresponding articles. Each chapter may be read independently from the rest of the manuscript.

Chapter 3

'On-the-fly' snapshots selection

Submitted to ...

Introduction

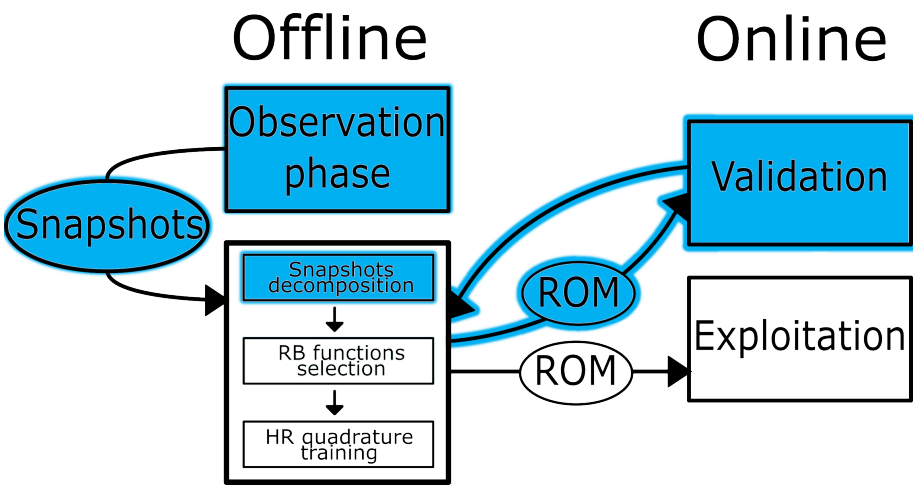
The vast majority of Model Order Reduction (MOR) methods rely on the construction of an appropriate Reduced Basis (RB). Projection-based Reduced Order Models (PROM) approximate the unknown field variable as a linear combination of a set of global and domain-spanning RB functions, that replace a large number of local, element-based shape functions. The Proper Orthogonal Decomposition (POD) is a PROM that traces its origins to statistical data analysis [48] and has found extensive application in turbulent flow modeling [75, 9, 36, 81] as well as in various other fields [67, 52, 53, 54, 68, 69, 70]. POD projects the model onto a low-rank data representation, yielding a ROM with fewer degrees of freedom. When applied to nonlinear FE models, POD is usually coupled with a method that approximates the projection of internal variables, e.g., Discrete Empirical Interpolation (DEIM) [17, 24, 77] or Hyper-Reduction (HR) [35, 73, 28].

POD builds the RB during the *offline* training phase through Singular Value Decomposition (SVD) of *previously collected* solution vector observations, called snapshots. Since the ultimate goal is a reduction in overall computation time, one needs to limit the computational complexity and memory requirements of the offline phase, since SVD, applied to a large-scale FE model, could potentially end up either expensive or infeasible. To tackle this issue, the "incremental SVD" ([11]) has recently emerged within the POD framework. In this method initially developed for streaming data analysis, SVD subsequently updates, using a mathematical identity formally presented in section 3.1, avoiding expensive data manipulation since snapshots are used as and when available.

The highlights of the algorithm presented in [56] are: adaptive, *on-the-fly* snapshot selection, and *on-the-fly* truncation. Adaptive snapshot selection identifies the simulation times at which observations must be made for the first-order differential equation solution. *On-the-fly* snapshot selection pre-evaluates the (potential) contribution of a snapshot before actually updating the RB, while *on-the-fly* truncation limits the size of the RB during subsequent enrichments. Both features enable a reduction in computational effort by avoiding the treatment of redundant observations, thus allowing the algorithm to work on smaller matrices, as discussed in detail in section 3.1.2.

However, the addition of these two features introduces new difficulties: firstly, two problem-dependent tolerances need to be set, and secondly, entailed loss of infor-

Figure 3.1: *A posteriori* MOR methods' features impacted by the incremental SVD



mation may inhibit the computation of the singular value truncation error necessary for the POD. An upper error bound for the incremental SVD method (calculated using a new weighted inner product) has been developed in [27], where the authors proved that the incremental SVD yields the exact SVD of the original data set. This error bound is incrementally computed by keeping track of singular values missed due to 'on-the-fly' calculation of the RB.

Present work, first presented by the author at the 6th European Conference on Computational Mechanics (ECCM6) in Glasgow, UK [59], builds a new upper bound estimator of the singular values truncation error using the same incremented variables. The originality consists in using the proposed incremented error estimator to monitor both 'on-the-fly' truncation and snapshot selection, as opposed to the problem-dependent tolerances in the traditional approach. The result is a relatively straightforward incremental SVD with nearly optimal 'on-the-fly' truncation and snapshot selection.

Referring to Figure 2.2 presented in chapter 2, sketching the workflow of *a posteriori* MOR methods, the features of interest of this chapter are highlighted in blue in Figure 3.1. The snapshots are to be decomposed and gathered at the same time, impacting both the observation phase and snapshots decomposition. Resulting RBs are validated *online* on the model used in the *offline* observation phase and compared to RBs computed with the traditional single-pass SVD.

The chapter is organized in the following manner. Section 3.1 presents a comprehensive review of the state-of-the-art incremental SVD. Section 3.2 presents the error estimator developed in this work. The algorithm is discussed in Section 3.3, before moving on to validation of the approach in Section 3.4, where a Taylor beam impact FE model is used to test the proposed method against the traditional incremental SVD, as well as the single-pass SVD. To evaluate the efficacy of our approach, we compare computational time, RB performances during the online reduction phase as well as the effective precision of the data approximation.

3.1 Incremental SVD

The singular value decomposition is a generalization of the eigenvalue decomposition for non-square matrices. Given a matrix $[S] \in \mathbb{R}^{N \times n_s}$ the singular value decomposition is given by

$$[S] = [\Phi][diag(\{s\})][\Psi]^T \tag{3.1}$$

In the regular SVD $[\Phi]$ and $[\Psi]$ are both square matrices and $[diag(\{s\})]$ is a matrix with zeros and dimension $N \times m$. In the thin SVD, on the other hand, $[diag(\{s\})]$ is square with size $m = min(N, n_s)$. The reason being that the dimension of the space spanned by the RB may neither exceed the number of degrees of freedom in the Full Order Model (FOM) nor the number of snapshots. Hence a regular SVD may result in unnecessary computations. Thus $[\Phi] \in \mathbb{R}^{N \times m}$ and $[\Psi] \in \mathbb{R}^{n_s \times m}$ are orthonormal matrices and $\{s\} = (s_1, s_2, \ldots, s_m)^T \in \mathbb{R}^m$ is the vector of singular values in descending order. Note that in our work the SVD refers to the "thin" SVD.

$[\Phi]$ is the matrix containing the main features of $[S]$ in decreasing order of importance. In the POD framework, if $[S]$ is a matrix of observations represented by column vectors, The complete snapshot matrix is

$$[S] = [\{u_1\}, ..., \{u_{n_s}\}] \in \mathbb{R}^{N \times n_s}, \tag{3.2}$$

the RB $[\Phi^k]$ of size k is obtained by taking the first k columns of $[\Phi]$. This RB, by construction, spans the vector space of dimension k that best approximates the data, in the sense that it minimizes the projection error given the desired precision $\epsilon_{rb} \in [0, 1]$,

$$k = \underset{l \in \mathbb{N}}{argmin}\{\epsilon_{sv}^l \leq \epsilon_{rb}\} \tag{3.3}$$

$$\epsilon_{sv}^l = \frac{|| \, [S] - [\Phi^l][\Phi^l]^T[S] \, ||_F}{|| \, [S] \, ||_F} = \sqrt{\frac{\sum_{i=l+1}^{m} s_i^2}{\sum_{j=1}^{m} s_j^2}} \tag{3.4}$$

with $|| \, [\bullet] \, ||_F = \sqrt{\sum_{i,j} [\bullet]_{ij}^2}$ being the Frobenius norm of a matrix.

When POD is applied to FE models, the snapshot matrix may become prohibitively large. To avoid storing $[S]$, the incremental SVD approach updates the RB 'on-the-fly', as and when a new observation is available, which also avoids the need to store snapshots. In addition, truncation could be performed in-between the updates, thus keeping the size of the RB reasonably small while also making room for a reduction in the number of floating point operations needed.

In subsection 3.1.1 we present the mathematical identity that allows the incremental computation of the SVD, while 'on-the-fly' snapshot selection and truncation are presented in subsection 3.1.2.

3.1.1 Incremental enrichment

The incremental SVD is a particular case of the low-rank modification (M. Brand [10]) which, given the SVD of a matrix $[S]$, computes the SVD of the updated matrix

$$[S'] = [[S], \{0_N\}] + \{a\}\{b\}^T, \tag{3.5}$$

where $\{a\} \in \mathbb{R}^N$ and $\{b\} \in \mathbb{R}^{n_s+1}$. RB enrichment appends a new snapshot $\{u_{i+1}\}$ to $[S]$ and updates it's SVD into that of $[S'] = [[S], \{u_{i+1}\}]$ using $\{a\} = \{u_{i+1}\}$ and $\{b\} = \{e_{i+1}\}$ in equation 3.5.

As mentioned in the previous section, the incremental SVD [10] computes the decomposition (3.1) of $[S]$ *without* storing the entire matrix, by updating the RB as and when a new observation is available.

The snapshot matrix at the ith iteration is given by

$$[S^{(i)}] = [\{u_1\}, ..., \{u_i\}] \in \mathbb{R}^{N\times i}.$$

The same matrix at the following iteration with one more observation added to it may be written:

$$[S^{(i+1)}] = [S^{(i)}, \{u_{i+1}\}] \in \mathbb{R}^{N\times(i+1)}. \tag{3.6}$$

The decomposition of $[S^{(i)}]$ is given by:

$$[S^{(i)}] = [\Phi^{(i)}][diag(\{s^{(i)}\})][\Psi^{(i)}]^T$$

with

- $[\Phi^{(i)}] \in \mathbb{R}^{N\times k}$ an orthonormal matrix (i.e. $[\Phi^{(i)}]^T[\Phi^{(i)}] = [Id]_{k\times k}$)
- $[\Psi^{(i)}] \in \mathbb{R}^{i\times k}$ an orthonormal matrix (i.e. $[\Psi^{(i)}]^T[\Psi^{(i)}] = [Id]_{k\times k}$)
- $\{s^{(i)}\} = (s_1^{(i)}, \dots, s_k^{(i)})^T \in \mathbb{R}^i$ with $s_j^{(i)} \neq 0$, $\forall j \in [\![1, k]\!]$

At the $(i+1)^{th}$ iteration the SVD of $[S^{(i)}]$ is known as well as N and k. Given a new observation $\{u_{i+1}\} \in \mathbb{R}^N$, the incremental SVD updates the decomposition of $[S^{(i)}]$ according to $[S^{(i+1)}] = [\Phi^{(i+1)}][diag(\{s^{(i+1)}\})][\Psi^{(i+1)}]^T$. Here the following identity [11] is used:

$$\begin{aligned}\big[[S^{(i)}], \{u_{i+1}\}\big] &= \big[[\Phi^{(i)}][diag(\{s^{(i)}\})][\Psi^{(i)}]^T, \{u_{i+1}\}\big] \\ &= \big[[\Phi^{(i)}], \{\xi\}\big] \underbrace{\begin{bmatrix} [diag(\{s^{(i)}\})] & [\Phi^{(i)^T}]\{u_{i+1}\} \\ \{0_k\}^T & \alpha \end{bmatrix}}_{:=[Q]} \begin{bmatrix} [\Psi^{(i)}] & \{0_i\} \\ \{0_k\}^T & 1 \end{bmatrix}^T,\end{aligned} \tag{3.7}$$

where

- $\alpha = \|\{u_{i+1}\} - [\Phi^{(i)}][\Phi^{(i)}]^T\{u_{i+1}\}\|_2$
- $\{\xi\} = \frac{\{u_{i+1}\} - [\Phi^{(i)}][\Phi^{(i)}]^T\{u_{i+1}\}}{\|\{u_{i+1}\} - [\Phi^{(i)}][\Phi^{(i)}]^T\{u_{i+1}\}\|}$.

Note, that $\left[[\Phi^{(i)}], \{\xi\}\right]$ and $\begin{bmatrix} [\Psi^{(i)}] & \{0_i\} \\ \{0_k\}^T & 1 \end{bmatrix}$ are orthonormal matrices. The main advantage of the incremental SVD lies in the economical diagonalization of the matrix $[Q]$.

$$[Q] = [\Phi'][diag(\{s'\})][\Psi']^T \tag{3.8}$$

yielding

$$[S^{(i+1)}] = [\Phi^{(i+1)}][diag(\{s^{(i+1)}\})][\Psi^{(i+1)}]^T$$

where

- $[\Phi^{(i+1)}] = \left[[\Phi^{(i)}], \{\xi\}\right] [\Phi']$
- $\{s^{(i+1)}\} = \{s'\}$
- $[\Psi^{(i+1)}] = \begin{bmatrix} [\Psi^{(i)}] & \{0_i\} \\ \{0_k\}^T & 1 \end{bmatrix} [\Psi']$

$[\Phi^{(i+1)}] \in \mathbb{R}^{N\times k}$ and $[\Psi^{(i+1)}] \in \mathbb{R}^{(i+1)\times k}$ are orthonormal matrices by construction, being products of orthonormal matrices each. Thus, the newly formed decomposition of $[S^{(i+1)}]$ is in fact its SVD.

Savings in computational resources (compared to the traditional SVD) occur during the bi-diagonalization of [Q], where simplifications are possible due to its particular form. Note that $[Q] \in \mathbb{R}^{k\times k}$ is a small "half-arrowhead matrix", meaning it is nearly diagonal except for its last column, which renders the SVD of $[Q]$, which is achieved by bi-diagonalization and the Golub-Kahan algorithm [32], computationnally inexpensive.

3.1.2 'On-the-fly' snapshot selection and truncation

The primary motivation for using the incremental SVD is to avoid the manipulation of a large amount of possibly redundant data while building a low-rank approximation. The two parameters ϵ_{svd} and k_{max} of this method first detailed in [56] allow for these savings:

- At the beginning of the $i+1^{th}$ iteration, the new observation $\{u_{i+1}\}$ may be rejected, if already well represented by the RB $[\Phi^{(i)}]$, so as to avoid unnecessary computations on redundant data. In the state-of-the-art algorithm [10], this is controlled by the parameter ϵ_{svd} and the new observation is skipped if $\|\{u_{i+1}\} - [\Phi^{(i)}][\Phi^{(i)}]^T\{u_{i+1}\}\|_2^2 \leq \epsilon_{svd}$.
- At the end of an iteration, $[S^{(i+1)}]$ decomposition (3.7) may be truncated to keep it as small as possible, accelerating subsequent iterations. Routinely [56], the basis is truncated 'on-the-fly' when its size exceeds a pre-determined value given by the parameter k_{max}.

Tuning ϵ_{svd} and k_{max} enables significant computational savings, compared with a single-pass SVD. When the single-pass SVD is used in the POD framework to build a RB, the right singular space $[\Psi]$ is generally not used. Such an SVD of a matrix

$[S] \in \mathbb{R}^{N \times n_s}$ will need $O(Nm^2)$ floating point operations and $O(2mN + m)$ memory. In the incremental SVD, if the RB's size does not exceed k, the method would only require $O(mNk)$ operations and $O(Nk + k)$ memory. In applications, the size k of the RB is very small in comparison with the FOM's size N and the number of observations n_s. The fundamental hypothesis of MOR is that the dimension of the underlying manifold, where the discretized solution of a computational mechanics problem "lives" and evolves, is small compared to the number of degrees of freedom. Therefore, incremental SVD is very promising for MOR. Finally, 'on-the-fly' snapshot selection would prevent computation on redundant data and thus limit the number of snapshots n_s processed, reducing the complexity as shown.

However, these two features also induce loss of information about the singular values rendering the exact final approximation error (3.4) inaccessible. An estimator for this error is developed and tested hereafter.

3.1.3 State-of-the-art incremental SVD algorithm

The state-of-the-art incremental SVD algorithm 3 as originally presented in [56] is presented in this section.

Instructions in red concern computation of the left singular subspace used in 'a posteriori' centering.

Truncation (steps 16 to 21 of algorithm 3) has been changed with respect to the original algorithm [56] prescribing additional rotations of the right and left subspaces. However, there is a mismatch between the matrices dimensions in the rotation formula and this step has been removed in our interpretation of the state-of-the-art algorithm.

3.1.4 State-of-the-art incremental SVD test

First tests of the state-of-the-art incremental SVD are presented in this section. The following analytical snapshot matrix is used to provide an easy to reproduce example of it.

$$[S] \in \mathbb{R}^{N \times n_s} \quad | \quad [S]_{ij} = i + j + j^2 + \frac{sin(mod(j, 10))}{\mid i - j \mid + 1}, \quad i \in [\![1, N]\!], \quad j \in [\![1, n_s]\!], \tag{3.9}$$

where $mod(a, b)$ denotes the remainder of the euclidean division of a by b. Expression (3.9) of the analytical snapshot matrix allow for a fast decrease of it's singular values while ensuring linearly independent columns, which is in agreement with the hypothesis that the dimension of the FE model is larger than the underlying manifold in which evolves the unknown quantity of interest.

In the following test, snapshot matrix dimensions are set to $N = 8000$ and $n_s = 200$. Different RBs are computed using the snapshot matrix and the state-of-the-art incremental SVD for varying parameters ϵ_{svd} and k_{max}. The training data approximation error $\epsilon_{sv}^{k_{max}}$, defined in equation (3.4), is computed 'a posteriori' by keeping the snapshot matrix in memory and compared to that of RBs originating from the single-pass SVD in Figure 3.2. Each curve correspond to a different value of ϵ_{svd} reported in the legend. For each value of ϵ_{svd}, 200 runs of the incremental SVD are performed for $k_{max} = k \in [\![1, 200]\!]$. The black curve is the reference error

Algorithm 3: State-of-the-art incremental SVD

Input: $[\Phi]$, $\{s\}$, $[\Psi]$, $\{u_i\}$
Output: $[\Phi]$, $\{s\}$, $[\Psi]$

```
if (size([Φ], 2) == 0) then
    if (‖{u_i}‖_2 > ε_svd) then
        {s} = ‖{u_i}‖_2
        [Φ] = {u_i} / ‖{u_i}‖_2
        [Ψ] = 1
    end
else
    p = ‖{u_i} − [Φ][Φ]^T {u_i}‖_2
    if (p > ε_svd) then
        [Q] = [ [diag({s})]   [Φ]^T {u_i} ;  0   p ]
        [[Φ'], {s'}, [Ψ']] = SVD([Q])
        % RB is expanded
        [Φ] = [[Φ], ({u_i} − [Φ][Φ]^T {u_i}) / ‖{u_i} − [Φ][Φ]^T {u_i}‖_2 ][Φ']
        {s} = {s'}
        [Ψ] = [ [Ψ]  0 ;  0  1 ] [Ψ']
        % RB is truncated
        if (size([Φ], 2) > k_max) then
            [Φ] = [Φ]_{:,1:k_max}
            {s} = {s}_{1:k_max}
            [Ψ] = [Ψ]_{:,1:k_max}
        end
        if ([Φ]^T_{:,1} [Φ]_{:,end} > ε_orth) then
            % RB has lost numerical orthogonality and is
              re-orthonormalized
            [Φ] = MGS([Φ])
        end
    end
end
```

Figure 3.2: State-of-the-art incremental SVD error of approximation

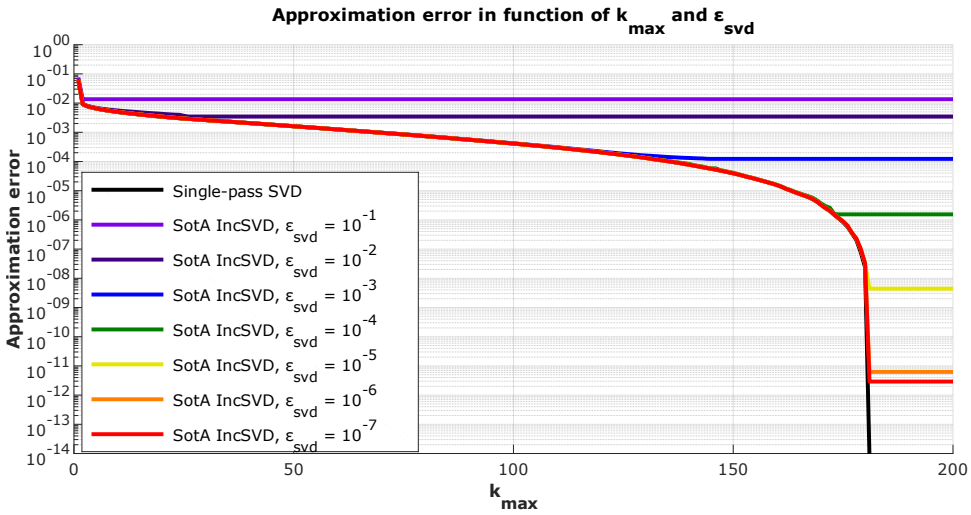
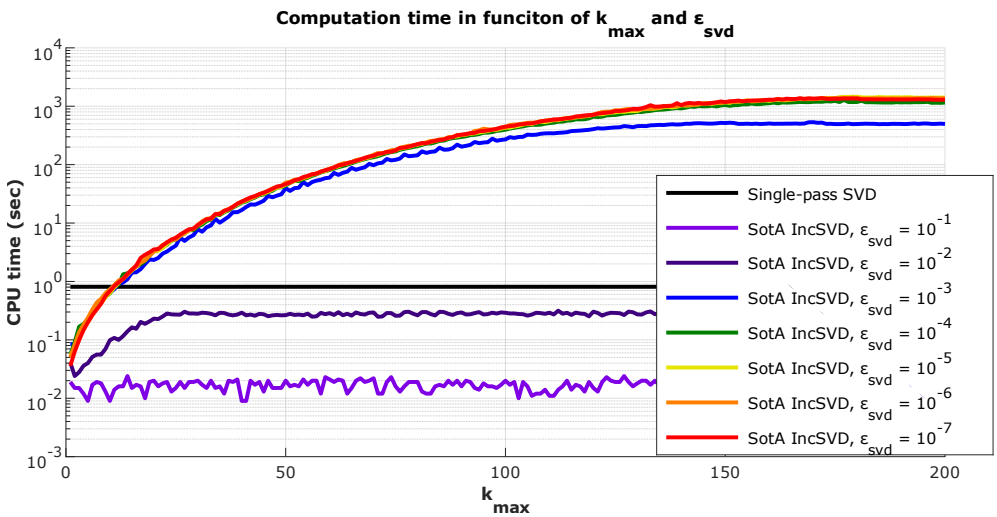


Figure 3.3: Incremental SVD computation time



obtained with RBs computed with the single-pass SVD. Associated computation times are reported on Figure 3.3. The state-of-the-art SVD is implemented in FORTRAN and computations are performed on a single Intel(R) Core(TM) i7-6820 CPU @ 2.70Ghz. The single-pass SVD used for comparison is the DGESVD provided by the Intel Math Kernel Library (Intel MKL) (version 12.1.3.300) and is used to compute only the first $min(m, n)$ left singular vectors and associated singular values wich correspond to the flags JOBU = 'S' and JOBVT = 'N' (c.f. DGESVD Documentation [1]).

ϵ_{svd} and k_{max} parameters both have a strong influence on the computation time and resulting RB training data approximation error. ϵ_{svd} enables computational savings by only selecting badly represented observations for enrichment while k_{max} control the size of the RB and enables computational savings by limiting matrices sizes in subsequent enrichment. Those two parameters also limit the resulting RB approximation quality. On this particular example, if ones want to build a RB that approximates the snapshot matrix at a precision of 10^{-2}, computation with

the state-of-the-art incremental SVD and parameters $\epsilon_{svd} = 10^{-2}$ and $k_{max} = 10$ will be one order of magnitude faster than using the single-pass SVD. Moreover, using the state-of-the-art incremental SVD prevents expensive data manipulations induced by the storage of a potentialy large snapshot matrix. Hence this method has the potential to greatly reduce CPU time needed in the computation of a RB. Nevertheless, computation times may skyrocket or, on the contrary, resulting RB may be a very poor approximation of the training data if the methods' parameters are not carefully chosen.

Two major issues arise in the practical use of the state-of-the-art incremental SVD.

- First, the snapshot matrix is not kept in memory and the produced RB training data approximation error (3.4) is unknown.
- Second, given a targeted training data approximation error, the optimal values for parameters k_{max} and ϵ_{svd} are unknown as they depend on the data nature (dimensionality and magnitude) and the method computation time may skyrocket if those two parameters are poorly set.

These two are overcame by the proposed error estimator presented in this chapter and used in place of k_{max} and ϵ_{svd}.

3.1.5 Note on *a posteriori* centering

Before presenting the modifications brought to the incremental SVD, a quick note is given regarding the possibility to center the training data *a posteriori* of the algorithm. In some POD applications, the matrix $[S]$ is centered ahead of applying the SVD to it. This is mainly motivated by the fact that the SVD, as explained in section 2.1.4, operates on the matrix $[\Theta] = [S]^T[S]$ which may be interpreted as the training data covariance matrix in the event that $[S]$ is centered.

In [56], the author claims that the snapshot matrix may be centered *a posteriori* of the RB computation with the incremental SVD using equation 3.5 with $\{a\} = \{\bar{u}\} = \frac{1}{n_s}[S]\{1_{n_s}\}$ the snapshots average and $\{b\} = \{1_{n_s}\}$, yielding

$$[S'] = [S] - \frac{1}{n_s}[S]\{1_{n_s}\}\{1_{n_s}\}^T \tag{3.10}$$

$$= [S] - \{\bar{u}\}\{1_{n_s}\}^T. \tag{3.11}$$

From our experience, *a posteriori* centering is not compatible with 'on-the-fly' snapshots selection and truncation. When using those options the incremental SVD computes a low-rank representation which does not allow an exact reconstruction of $[S]$, preventing an exact update.

The snapshots' average may be computed during the subsequent passes through the incremental SVD, nevertheless, left singular vectors are truncated 'on-the-fly' based on the decomposition of the un-centered training data. Hence, truncated singular vectors are not the same if the snapshot matrix is centered *a posteriori* or from the start.

Another issue with *a posteriori* centering lies in the need to compute the right singular vectors. Those vector are computed by the red instructions in algorithm 3

and their complexity scales with the number of snapshots rather than the RB size as $[\Psi]$ is of dimensions n_s times k_{max}.

In our work, we made the choice not to center the snapshot matrix for the following reasons. Centering the snapshot matrix is not mandatory for SVD and does not change it's range. If the training data is centered, $\{\bar{u}\}$ is to be used as a lift and PROM approximation (2.9) presented in section 2.1.3 rewrites

$$\{u(t)\} \approx \{\tilde{u}(t)\} = \sum_{i=1}^{k} \alpha_i(t)\{\phi_i\} + \{\bar{u}\}, \quad \forall t \in [0, T]. \tag{3.12}$$

We see in equation (3.12) that using a lift does not increase the approximation precision over adding an extra RB function.

3.2 Error estimator

In the original incremental SVD, 'on-the-fly' snapshot rejection and basis truncation prevent the computation of all singular values, meaning that it is impossible to compute (3.4). The modified version of the incremental SVD, proposed in this chapter, tracks the lost information (due to the missed singular values) using two variables, ϵ_{in} and ϵ_{out}.

Since the basis is truncated 'on-the-fly', the singular value associated with the truncated mode is lost, therefore, ϵ_{out} is incremented by the square of the skipped singular value:

$$\epsilon_{out} = \epsilon_{out} + s_{i+1}^2. \tag{3.13}$$

If a snapshot $\{u_{i+1}\}$ has been rejected 'on-the-fly', we can no longer calculate changes in the singular values, ϵ_{in} is incremented by the norm of projection of $\{u_{i+1}\}$ on $[\Phi^{(i)}]$:

$$\epsilon_{in} = \epsilon_{in} + \|[\Phi^{(i)}][\Phi^{(i)}]^T\{u_{i+1}\}\|_2^2, \tag{3.14}$$

and ϵ_{out} is then incremented by the error of projection of the skipped snapshot:

$$\epsilon_{out} = \epsilon_{out} + \|\{u_{i+1}\} - [\Phi^{(i)}][\Phi^{(i)}]^T\{u_{i+1}\}\|_2^2. \tag{3.15}$$

These two variables are used to compensate for the unavailable singular values, yielding the following expression for the error estimator :

$$\epsilon_{est} = \sqrt{\frac{\epsilon_{out}}{\|\{s\}\|_2^2 + \epsilon_{in} + \epsilon_{out}}}. \tag{3.16}$$

An essential condition on the error estimator is that it must be larger than the actual error. In the event that the new observation is used for enriching the RB, the error estimator does not induce any error since the square of the exact singular value associated with truncated mode is incremented in ϵ_{out}. On the other hand, if the new observation is not selected, then the error estimator satisfies the following condition:

$$\mathcal{P}^{(i)} = \left\{ \frac{\|[S^{(i)}] - [\Phi^{(i)}][\Phi^{(i)}]^T[S^{(i)}]\|_F^2}{\|[S^{(i)}]\|_F^2} \leq \frac{\epsilon_{out}^{(i)}}{\|\{s^{(i)}\}\|_2^2 + \epsilon_{in}^{(i)} + \epsilon_{out}^{(i)}} = \epsilon_{est}^{(i)} \right\}.$$

This ensures that the estimator is an upper bound on the actual error. A straightforward recursive proof of the above property is given below:

<u>initialization:</u> For the first iteration of the algorithm, $i = 1$, the error is null, $\epsilon_{in} = \epsilon_{out} = 0$ and no information has been lost (yet) due to either rejection or truncation.

<u>induction:</u> If we assume that $\mathcal{P}^{(i)}$ is true and the new observation $\{u_{i+1}\}$ has been rejected, then the RB is left unchanged ($[\Phi^{(i+1)}] = [\Phi^{(i)}]$ and $\{s\}^{(i+1)} = \{s\}^{(i)}$), $\epsilon_{out}^{(i+1)} = \epsilon_{out}^{(i)} + \|\{u_{i+1}\} - [\Phi^{(i)}][\Phi^{(i)}]^T\{u_{i+1}\}\|_2^2$ and $\epsilon_{in}^{(i+1)} = \epsilon_{in}^{(i)} + \|[\Phi][\Phi^{(i)}]^T\{u_i\}\|_2^2$. In which case, using identity (3.6):

$$\frac{\|[S^{(i+1)}] - [\Phi^{(i+1)}][\Phi^{(i+1)}]^T[S^{(i+1)}]\|_F^2}{\|[S^{(i+1)}]\|_F^2} = \frac{\|([Id] - [\Phi^{(i+1)}][\Phi^{(i+1)}]^T)[S^{(i)}, \{u_{i+1}\}]\|_F^2}{\|[S^{(i)}, \{u_{i+1}\}]\|_F^2},$$

using the additivity of the Frobenius norm, previous expression is rewritten

$$= \frac{\|([Id] - [\Phi^{(i+1)}][\Phi^{(i+1)}]^T)[S^{(i)}]\|_F^2 + \|([Id] - [\Phi^{(i+1)}][\Phi^{(i+1)}]^T)\{u_{i+1}\}]\|_2^2}{\|[S^{(i)}]\|_F^2 + \|\{u_{i+1}\}\|_2^2},$$
$$= \frac{\|([Id] - [\Phi^{(i)}][\Phi^{(i)}]^T)[S^{(i)}]\|_F^2 + \|([Id] - [\Phi^{(i)}][\Phi^{(i)}]^T)\{u_{i+1}\}]\|_2^2}{\|[S^{(i)}]\|_F^2 + \|\{u_{i+1}\}\|_2^2},$$

finally, $\mathcal{P}^{(i)}$ is used

$$\leq \frac{\epsilon_{out}^{(i)} + \|([Id] - [\Phi^{(i)}][\Phi^{(i)}]^T)\{u_{i+1}\}\|_2^2}{\|\{s^{(i)}\}\|_2^2 + \epsilon_{in}^{(i)} + \epsilon_{out}^{(i)} + \|\{u_{i+1}\}\|_2^2} = \frac{\epsilon_{out}^{(i+1)}}{\|\{s^{(i+1)}\}\|_2^2 + \epsilon_{in}^{(i+1)} + \epsilon_{out}^{(i+1)}} = \epsilon_{est}^{(i+1)},$$

which proves $\mathcal{P}^{(i+1)}$.

3.3 Proposed algorithm

Algorithm 4 incorporates the proposed incremented error estimator and corresponding 'on-the-fly' truncation and snapshot selection. ϵ_{rb} is the threshold on the singular value truncation error and is the only input argument that the user needs to set in this version of the algorithm. All other arguments are initialized in the method from steps 2 to 6 and incremented during subsequent enrichments.

The main feature in the proposed version of the algorithm is the use of the proposed error estimator for the 'on-the-fly' snapshot selection at steps 8 and 9 as well as 'on-the-fly' truncation at steps 17 to 23. The philosophy behind the algorithm is speeding up computations by rejecting the maximal number of snapshots and truncating the RB as soon as possible, in each enrichment, while *still* controlling the overall singular value error of truncation and avoiding the necessity to set the problem-dependent parameters ϵ_{svd} and k_{max}.

One potential issue is the loss of orthonormality of the RB due to subsequent enrichments (line 13). While this does not change the range of the RB, it does, however, prevent a good projection of the new observation as $\|\{u_i\}\|_2^2 = \|[\Phi][\Phi]^T\{u_i\}\|_2^2 + \|\{u_i\} - [\Phi][\Phi]^T\{u_i\}\|_2^2$ is not necessarily true if $[\Phi]$ is not orthonormal. This is why re-orthonormalization is performed (lines 23 to 26) if the scalar

Algorithm 4: Proposed incremental SVD

Input: $[\Phi]$, $\{s\}$, $\{u_i\}$, ϵ_{in}, ϵ_{out}, ϵ_{rb}
Output: $[\Phi]$, $\{s\}$, ϵ_{in}, ϵ_{out}

```
if (size([Φ], 2) == 0) then
    if (‖{u_i}‖_2 > 0) then
        {s} = ‖{u_i}‖_2
        [Φ] = {u_i} / ‖{u_i}‖_2
        ε_in = ε_out = 0
    end
else
    error = (ε_out + ‖{u_i} − [Φ][Φ]^T{u_i}‖_2^2) / (ε_in + ε_out + ‖{s}‖_2^2 + ‖{u_i}‖_2^2) − ε_rb^2
    if (error > 0) then
        α = ‖{u_i} − [Φ][Φ]^T{u_i}‖_2
        [Q] = [ [diag({s})]   [Φ]^T{u_i} ;  0   α ]
        [[Φ'], {s'}, [Ψ']] = SVD([Q])
        % RB is expanded
        [Φ] = [[Φ], ({u_i} − [Φ][Φ]^T{u_i}) / ‖{u_i} − [Φ][Φ]^T{u_i}‖_2][Φ']
        {s} = {s'}
        % RB is truncated with new error estimator
        error = (ε_out + s(end)) / (ε_in + ε_out + ‖{s}‖_2^2) − ε_rb^2
        while (error < 0) do
            ε_out = ε_out + s(end)                                   equation (3.13)
            {s} = {s}(1 : (end − 1))
            [Φ] = [Φ]_{:,1:(end−1)}
            error = error + s(end) / (ε_in + ε_out + ‖{s}‖_2^2)
        end
        if ([Φ]^T_{:,1} [Φ]_{:,end} > ε_orth) then
            % RB has lost numerical orthogonality and is
             re-orthonormalized
            [Φ] = MGS([Φ])
        end
    else
        ε_out = ε_out + ‖{u_i} − [Φ][Φ]^T{u_i}‖_2^2              equation (3.15)
        ε_in = ε_in + ‖[Φ][Φ]^T{u_i}‖_2^2                        equation (3.14)
    end
end
```

product of the first and last columns of the current RB is larger than a threshold (ϵ_{orth}), which is typically set to 10^{-14}. This is performed using the highly parallelizable Modified Gram-Schmidt (MGS) algorithm, but may also be implemented by QR factorization.

Another important remark involves "centering" the data. In some POD applications such as [82], the snapshot matrix is centered before the spectral approximation is computed, motivated, in part, by the statistical point of view interpreting $[S]^T[S]$ as the covariance matrix of the data. In our applications, data points are not centered as simply applying a "lift" to the POD approximation (2.9) will not improve the precision over adding a column to $[\Phi]$ That said, [56, 11] use 'a posteriori' centering for the incremental SVD. To this end, the right subspace $[\Psi]$ of the decomposition must be computed and stored, the size of which scales up with the number of observations and cannot be truncated 'on-the-fly'. In our experience, 'a posteriori' centering is not compatible with 'on-the-fly' truncation. Truncated/selected basis vectors as well as rejected observations during the enrichment steps would not be the same, had they been performed using snapshots that were centered from the beginning. Moreover, 'on-the-fly' truncation would deteriorate the snapshot matrix reconstruction and render the 'a posteriori' centering inaccurate. The centered POD is thus more straightforward with the regular than the incremental SVD.

3.4 Numerical tests

In this section, we validate the proposed incremental SVD approach using a Taylor beam impact model. We first give a short review of the POD method in section 3.4.2, followed by testing the approach for computation time and estimated training data approximation error in section 3.4.3. The proposed incremental SVD is benchmarked against the state-of-the-art incremental SVD first in section 3.4.4 in terms of computation time and offline training data approximation error and then in section 3.4.5 in terms of work of internal forces approximation in the online reduction phase. Finally, section 3.4.6 describes the scalability of the proposed incremental SVD with respect to the dimensions of the snapshots matrix and targeted error of approximation.

NOTE: The Taylor beam impact test was performed with the proposed incremental SVD implementation using the state-of-the-art crash simulation code Altair Radioss [3].

3.4.1 Taylor beam impact

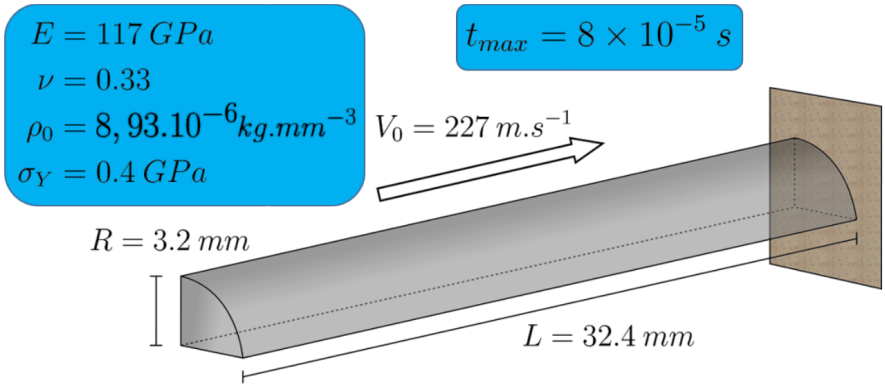
The tests have been performed on a snapshot matrix $[S]$ obtained from a Taylor beam impact simulation, using the model shown in Figure 3.4.

The beam is made of an elasto-plastic steel of density $\rho = 8.93 * 10^{-9}\ T.mm^{-3}$, Young's modulus $E = 117000\ MPa$ and Poisson's ratio $\nu = 0.33$. The hardening rule is described by the Johnson-Cook law, neglecting temperature and strain rate effects

$$\sigma_{eq} = \sigma_Y + b * \varepsilon_p^n, \tag{3.18}$$

with σ_{eq} the equivalent stress and ε_p the equivalent plastic strain. Other parameters are plastic yield stress $\sigma_Y = 400\ MPa$, plasticity hardening coefficient $b = 100\ Mpa$

Figure 3.4: Taylor beam test case



and plasticity hardening exponent $n = 1$. 8-node solid elements with one integration point are used to discretize the model into $N = 8514$ dofs. The central difference method is used for time integration along with the lumped mass approach, yielding a diagonal mass matrix.

The incremental displacements snapshots have been taken at regular intervals over the total simulation time. We begin with a brief review of the POD methods in the next subsection, followed by evaluating the RB's performance during the online reduction phase later in this section.

3.4.2 POD review

Full-order model

In nonlinear structural dynamics, the semi-discretized finite element formulation takes on the following form:

$$[\mathbb{M}]\{\ddot{u}(t)\} + \{f_{int}\}(\{u(t)\}, t) = \{f_{ext}(t)\}, \tag{3.19}$$

where $\{u(t)\} \in \mathbb{R}^N$ is the vector of nodal displacements (for all nodes and along three directions) unknown at time t. NOTE: In general, $\{u(t)\}$ may also contain shell rotations, however, that is not the primary focus of this chapter. N denotes the number of degrees of freedom (dofs), the size of the FOM. $[\mathbb{M}] \in \mathbb{R}^{N\times N}$ is a symmetric, real, positive definite mass matrix.

The FOM may depend on n_p parameters $(p_i)_{1\leq i\leq n_p}$ which are generally shell thicknesses and material properties in vehicle crash simulations. The first step consists of extracting observations, or snapshots, at various simulation times and for different parameter value sets. For convenience, the observations time t and observation parameter values $(p_1, p_2, \ldots, p_{n_p})^T$ are placed together in a vector $\{c\} = (t, p_1, p_2, \ldots, p_{n_p})^T$, that we refer to as the training configuration.

Reduced basis

The FOM solutions of training configurations $(\{c_i\})_{1\leq i\leq n_s}$ are collected in the snapshot matrix $[S] \in \mathbb{R}^{N\times n_s}$

$$[S] = [\{u(\{c_1\})\}, \{u(\{c_2\})\}, \ldots, \{u(\{c_{n_s}\})\}].$$

In the second step, we compute a basis for the vector space of minimal dimension k that is capable of approximating the data in $[S]$ to a user-defined precision ϵ_{rb}. This new vector space of dimension k is spanned by the columns of $[\Phi] \in \mathbb{R}^{N\times k}$

$$[\Phi] = [\{\phi_1\}, \{\phi_2\}, \ldots, \{\phi_k\}] \in \mathbb{R}^{N\times k}.$$

The POD approximation is then written as:

$$\{u(t)\} \approx \{\tilde{u}(t)\} = \sum_{i=1}^{k} \alpha_i(t)\{\phi_i\} = [\Phi]\{\alpha(t)\} \tag{3.20}$$

$\{\alpha(t)\} = (\alpha_1(t), \alpha_2(t), \ldots, \alpha_k(t))^T \in \mathbb{R}^k$ is the vector of ROM unknowns.

$[\Phi]$ is obtained by computing a low-rank representation of the snapshot matrix $[S]$. The RB size k is given by the desired precision ϵ_{rb} of the low-rank representation of $[S]$.

Reduced model

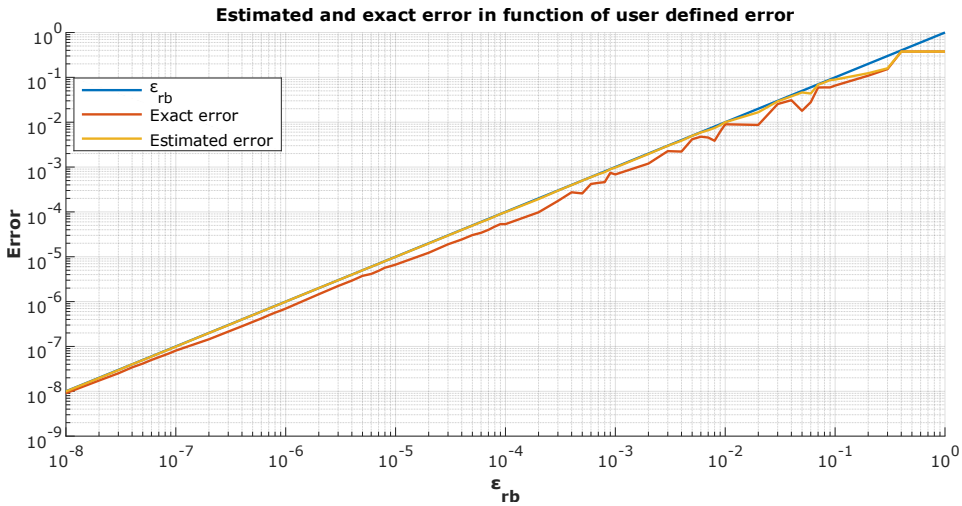
The ROM is constructed by injecting the approximation (3.20) into (3.19) and projecting the resulting equations on $[\Phi]^T$.

$$[\tilde{\mathbb{M}}]\{\ddot{\alpha}(t)\} + [\Phi]^T\{f_{int}([\Phi]\{\alpha(t)\}, t)\} = [\Phi]^T\{f_{ext}(t)\} \tag{3.21}$$

We introduce the following notation: $[\tilde{\mathbb{M}}] = [\Phi]^T[\mathbb{M}][\Phi] \in \mathbb{R}^{k\times k}$. The size of the reduced model (3.21) is k, which is much smaller than the size N of the original full order model (3.19). However, the use of POD for the reduction of explicit structural dynamics models induces computational overhead during the online phase due to the Galerkin projection used to build the PROM. Solution information at each node is mandatory in order to compute internal variables in each element. The reduced unknown $\{\alpha\}$ does not provide information explicitly at each node and the full-scale approximation needs to be computed. In order to map reduced unknowns $\{\alpha\}$ to the full-scale approximation $\{\tilde{u}\}$, defined in equation (3.20), supplementary matrix vector multiplications are performed at each time step, which negatively affects the online speed-up. Thus, we require additional reduction methods to obtain a computational speed-up during the online reduction phase in explicit nonlinear POD applications. These methods attempt to approximate the projected nonlinear internal variables operator $[\Phi]^T\{f_{int}([\Phi]\bullet, t)\}$. The most popular methods to achieve this approximation are hyper-reduction ([28, 73, 35]) and the DEIM ([17, 24, 77]).

Another important feature of the ROM (3.21) is that the stability condition for optimal time step is also projected. In the event that the nodal time step is used, the projection of the CFL condition on $[\Phi]$ leads to a larger time step, some theoretical developments on this subject may be found in [7].

Figure 3.5: Incremental SVD error estimator in function of the target error for the Taylor beam impact. In this Figure, the error estimator is compared to the exact error and the target error.



3.4.3 Proposed incremental SVD tests

In this section, we present some preliminary tests to validate the proposed algorithm, wherein we compare the error estimator (3.16) with the 'actual' error of approximation (3.4) (which requires storage of the snapshot matrix). Computation times using different values of user-defined error ϵ_{rb} are also compared with those for the single-pass SVD from the Fortran Intel MKL package. We also present a comparison of the performance of these RBs in the online reduction phase of the Taylor beam impact.

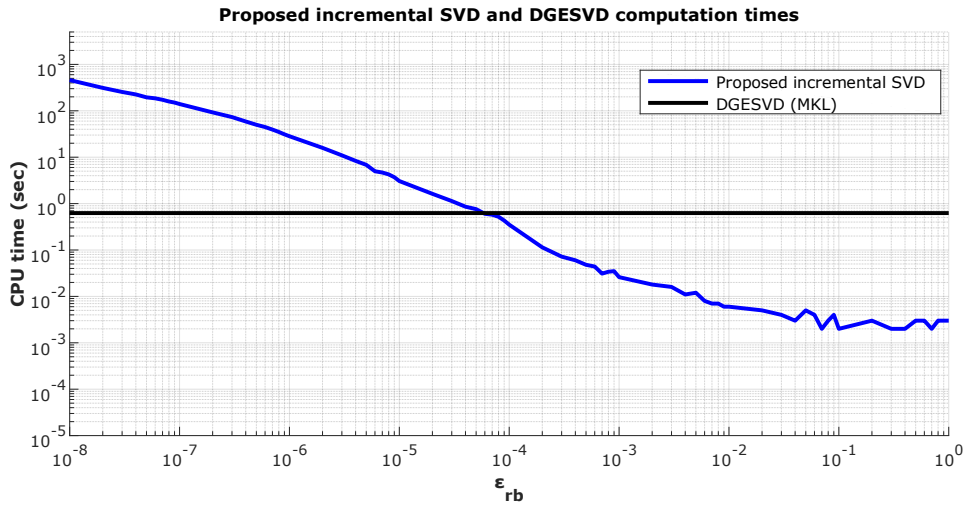
For these tests, we employ the matrix $[S] \in \mathbb{R}^{8514\times236}$ of incremental displacements snapshots obtained using the Taylor beam model runs. The snapshot selection and 'on-the-fly' truncation are now driven by keeping the error estimator (3.16) below the single target error ϵ_{rb}, in place of ϵ_{svd} and k_{max}, for the state-of-the-art incremental SVD. We test the method for $\epsilon_{rb} \in [10^{-8}, 1]$.

Figure 3.5 shows the RB's approximation error and associated error estimator values as a function of ϵ_{rb} (estimated errors in gray, real errors in black). Computation times of those RBs are compared with that of the single-pass SVD on Figure 3.6. The single-pass SVD used here is the DGESVD from the Intel MKL library provided with Intel FORTRAN compilers (version 12.1.3.300) and is used to compute only the first $m = min(n_s, N) = 236$ left singular vectors and associated singular values which correspond to the flags JOBU = 'S' and JOBVT = 'N' (c.f. DGESVD Documentation [1]).

Figure 3.5 clearly shows that the error estimator is always lower than the target ϵ_{rb} and higher than the actual approximation error. The smaller is the desired ϵ_{rb}, the more precise is the error estimator (3.16) and the 'actual' RB approximation error (3.4).

The computation times reported in Figure 3.6 show that our algorithm is faster than the single-pass SVD up to a precision of $\epsilon_{rb} = 10^{-4}$. However, the incremen-

Figure 3.6: Incremental SVD computation time in function of the target for Taylor beam impact.



tal SVD is not designed for a single-pass SVD since the computation times tends to skyrocket with very low ϵ_{rb}. It is also important to note that given a targeted precision, the incremental SVD may build a larger RB than that obtained using the single-pass SVD, since the error is only *estimated* and, in reality, may well be lower than the target error.

3.4.4 Comparison with state-of-the-art incremental SVD

Comparing the state-of-the-art incremental SVD with the proposed incremental SVD is not a straightforward task as the methods do not involve the same parameters. In particular, as explained in section 3.1.2, the state-of-the-art incremental SVD has various tolerances that have to be set to appropriate values.

We use the same snapshot matrix $[S] \in \mathbb{R}^{8514\times236}$ within this section. The algorithm is tested against the state-of-the-art incremental SVD [56] in the following manner: First, the RB (of size $k = 3$) is computed using the proposed algorithm with a user specified error $\epsilon_{rb} = 10^{-2}$. In the second phase, state-of-the-art incremental SVD runs are performed for $k_{max} = 3$ and hundred different values of ϵ_{svd} *logarithmically* spaced between 10^{-7} and 10^{2}. The re-orthonormalization parameter is set to $\epsilon_{orth} = 10^{-14}$ in both algorithms. The resulting computation times are given in Figure 3.8 and approximation errors in Figure 3.7.

In this example, both the proposed and state-of-the-art incremental SVD outperform the single-pass SVD from the Fortran Intel MKL package. This is because the data of 236 vectors may be approximated with a precision of $\epsilon_{rb} = 10^{-2}$ with only $k = 3$ basis vectors. This "very low-dimensional" approximation works in favor of the two incremental methods.

Using properly-chosen parameters within the state-of-the-art SVD ($k_{max} = 3$ and $\epsilon_{svd} = 2$), the incremental methods' performances are both comparable. However, it is not clear how to set these two parameters in practice, since they depend on the data dimensionality and magnitude as well as the target precision. k_{max} has been

Figure 3.7: Proposed and state-of-the-art incremental SVDs approximation errors for reduced basis of size $k = 3$ on snapshots from the Taylor beam impact simulations

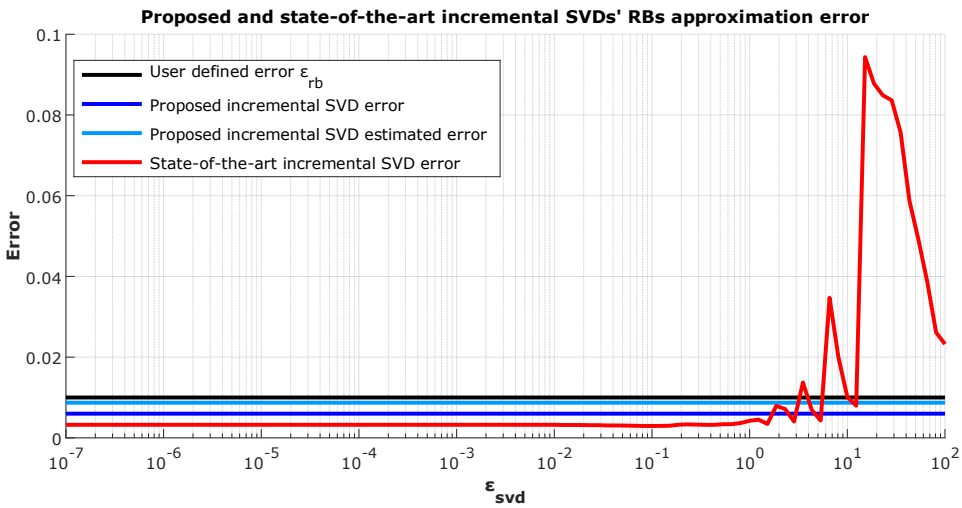
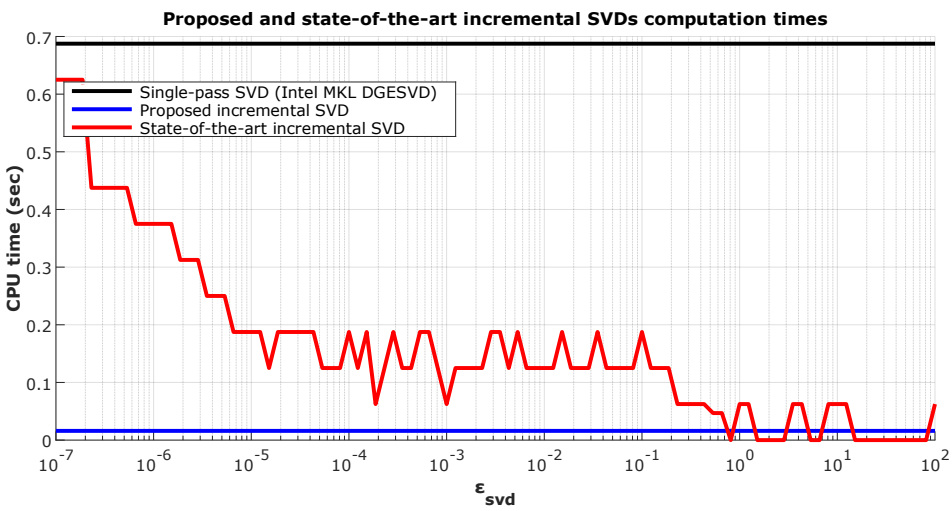


Figure 3.8: Proposed and state-of-the-art incremental SVDs computation times for reduced basis of size $k = 3$ on snapshots from the Taylor beam impact simulations



set to the minimal value allowing an error of approximation smaller than 10^{-2}. The results in Figure 3.7 show that if ϵ_{svd} is too large, then the desired error cannot be attained. On the other hand, if its value is too small, we see from Figure 3.8 that computation time would be significantly higher than that of the proposed version. Moreover, the state-of-the-art SVD approximation errors given in Figure 3.7 have been computed *a posteriori*, by storing *all* the data in memory, which is not desirable in practice.

In this example, the proposed incremental SVD shows better performance in terms of precision and computation time without having to set any parameter than the targeted singular value approximation error (3.16).

3.4.5 Online comparison

We now compare the performance of the RBs during the online reduction phase. For this, we consider 5 RBs of size $k = 6$, the first being obtained with the truncated single-pass SVD, the next three using the state-of-the-art incremental SVD with parameters $\epsilon_{svd} \in \{10^{-14}, 10^{-6}, 10^{-4}\}$ and $k_{max} = 6$ and the last RB corresponding to our proposed incremental SVD with $\epsilon_{rb} = 2 * 10^{-3}$.

Online performances are evaluated in terms of work of internal forces reconstruction. Internal work at a given time is defined by

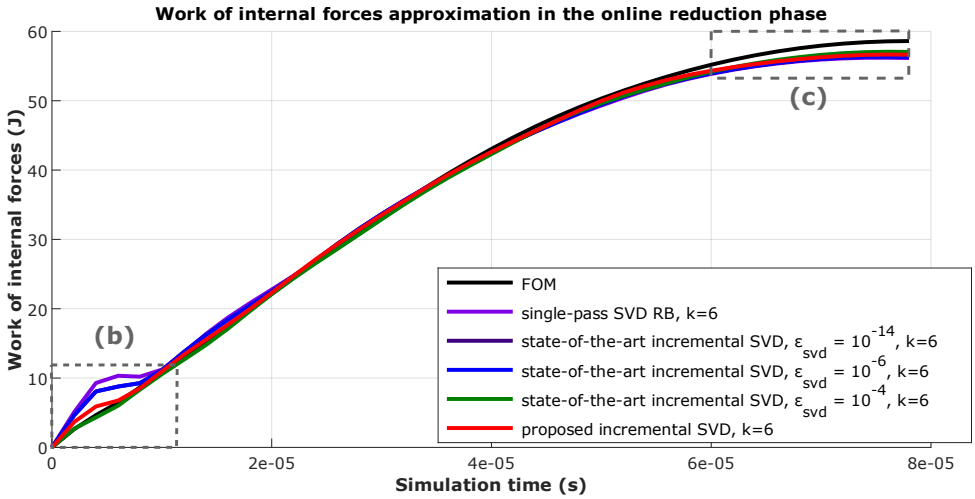
$$W_{int}(t) = \int_{\Omega} \underline{\underline{\varepsilon}} : \underline{\underline{\sigma}} dV \tag{3.22}$$

with $\underline{\underline{\varepsilon}}$ the strain tensor and $\underline{\underline{\sigma}}$ the stress tensor. Denoting W_{int} and $\tilde{W}_{int}$ the work of internal forces in the FOM and in the online reduction phase, respectively. The error ϵ_{int} on the work of internal forces considered here is

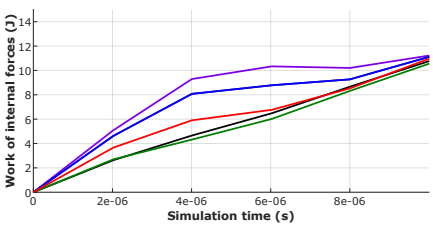
$$\epsilon_{int} = \frac{\int_0^T \mid W_{int}(t) - \tilde{W}_{int}(t) \mid dt}{\int_0^T W_{int}(t) dt}. \tag{3.23}$$

Figure 3.9 plots the variation of the work of internal forces during the reduction phase for all 5 RBs as well as for the FOM.

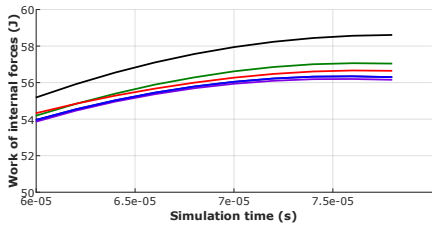
Even though the single-pass SVD is optimal as far as the singular value truncation error during the offline training phase is concerned, it is clearly not optimal for work of internal forces reconstruction during the online POD reduction phase. When snapshots are uniformly sampled from the simulation, SVD tends to preferentially emphasize those corresponding to later simulation times since they are generally larger in magnitude and redundant. On the other hand, snapshots taken at earlier simulation times tend to be smaller in magnitude and unique, therefore, they are given less importance in the SVD. As a consequence, POD approximation error is often larger during the early stages of the simulation, as seen on this example. This behavior is seen to a lesser degree in the incremental SVD as 'on-the-fly' selection avoids the stacking of redundant, larger magnitude observations by only selecting snapshots that introduce new information. In Figure 3.9, RBs obtained with the state-of-the-art incremental SVD yield a better approximation of the work of internal forces at the beginning of the reduced simulation. The RBs obtained with



(a) Work of internal forces in the entire simulation



(b) Zoom on first simulation times



(c) Zoom on last simulation times

Figure 3.9: Comparison of the work of internal forces reconstruction in the online reduction phase on the Taylor beam test case. RBs of size $k = 6$ computed with the proposed incremental SVD, the state-of-the-art SVD and the single-pass SVD are used.

$\epsilon_{svd} \leq 10^{-6}$ are identical. Increasing ϵ_{svd} from 10^{-6} to 10^{-4} results in an even better approximation. However, for $\epsilon_{svd} > 10^{-4}$, it is not possible to build RBs of size $k = 6$ as the number of snapshots selected 'on-the-fly' is insufficient. So $\epsilon_{svd} = 10^{-4}$ is the optimal value for this particular example. But this value is problem-dependent, and increasing the number of degrees of freedom N, the maximal size of the RB k_{max}, the number of snapshots n_s or switching unit system will change the optimal value for ϵ_{svd}. Although the state-of-the-art incremental SVD with optimal value of ϵ_{svd} yields the best RB in this example, the actual value itself is unknown in practice. The proposed incremental SVD involves a nearly optimal 'on-the-fly' selection criterion that only depends on the target approximation error, which is easier to use and *not* problem-dependent. The RB obtained with the proposed method approximates the work of internal forces almost as well as the state-of-the-art incremental SVD with $\epsilon_{svd} = 10^{-4}$.

We mentioned that, the singular value approximation error being only approximated in the proposed method, the RB's size may be larger than necessary. However, this is not an issue as results within this section show that the singular value approximation error does not guarantee the error that will be incurred within the online reduction phase. As a matter of fact RB computed with the proposed method performs better in the online reduction phase that the one computed using the single-pass SVD even though it produces a larger training data singular value approximation error. Finally, the test during the online POD phase shows a satisfactory performance. RB computed with the proposed method performs well in comparison with RBs of same sized obtained with state-of-the-art SVD and single-pass SVD using the same training data set.

3.4.6 Proposed incremental SVD scalability

In this subsection, we test the computation time for algorithm 4 against the single-pass SVD DGESVD from the Intel MKL FORTRAN library using snapshots matrices obtained from two different Taylor beam models. In Figure 3.11, the computation times are plotted for $[S] \in \mathbb{R}^{4752 \times n_s}$ and for $n_s \in [\![100, 1000]\!]$. A larger model of the Taylor beam impact has been used for Figure 3.13 in which $[S] \in \mathbb{R}^{18810 \times n_s}$ and $n_s \in [\![500, 8000]\!]$. The snapshots in this figure have been uniformly sampled from the simulation time interval for all values of n_s. No binary file manipulation times are taken into consideration here.

The complexity of the single-pass SVD (DGESVD) scales up according to $m = min(N, n_s)$ and thus its computation time plateaus when there are more snapshots than degrees of freedom ($n_s > N$). However, the computation time of the DGESVD grows quadratically with the number n_s of snapshots until $n_s = N$. On the other hand, the incremental SVD's complexity scales up according to the dimensionality of the approximating vectorial space, hence, the computational time stabilizes very fast. In fact, for a given target precision ϵ_{rb}, the incremental SVD computation time grows quadratically with the number of snapshots until $n_s = k$, where k is given by equation (3.3). This means that for an insufficient number of snapshots n_s and a high target precision ϵ_{svd}, the proposed incremental SVD a poorer performance than the DGESVD, until we reach the minimum number of snapshots given by equation (3.3) after which the proposed approach shows superior performance. In figure 3.13, the incremental SVD is seen to perform better than the single-pass SVD for precision up

Figure 3.10: Taylor beam impact, Scalability of incremental vs. single-pass SVD, $N = 1518$

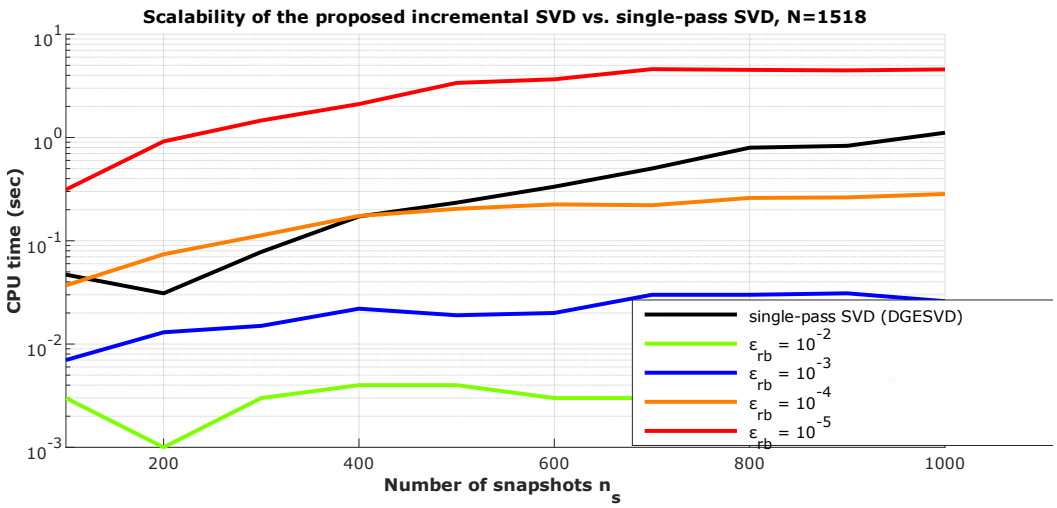


Figure 3.11: Taylor beam impact, Scalability of incremental vs. single-pass SVD, $N = 4752$

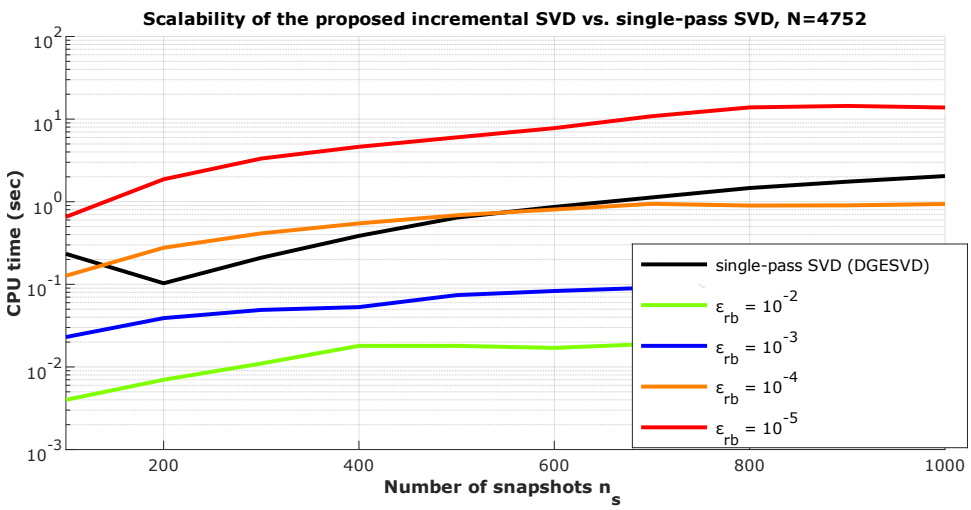


Figure 3.12: Taylor beam impact, Scalability of incremental vs. single-pass SVD, $N = 8514$

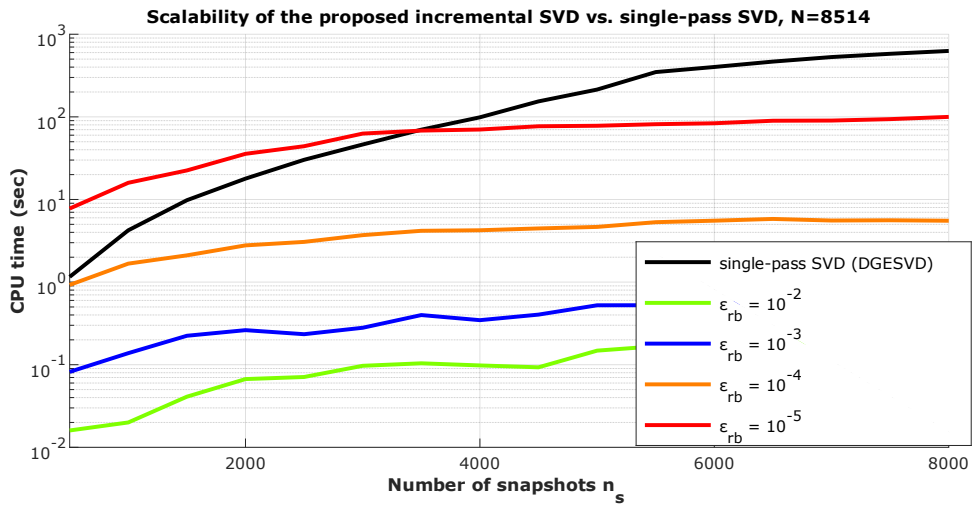
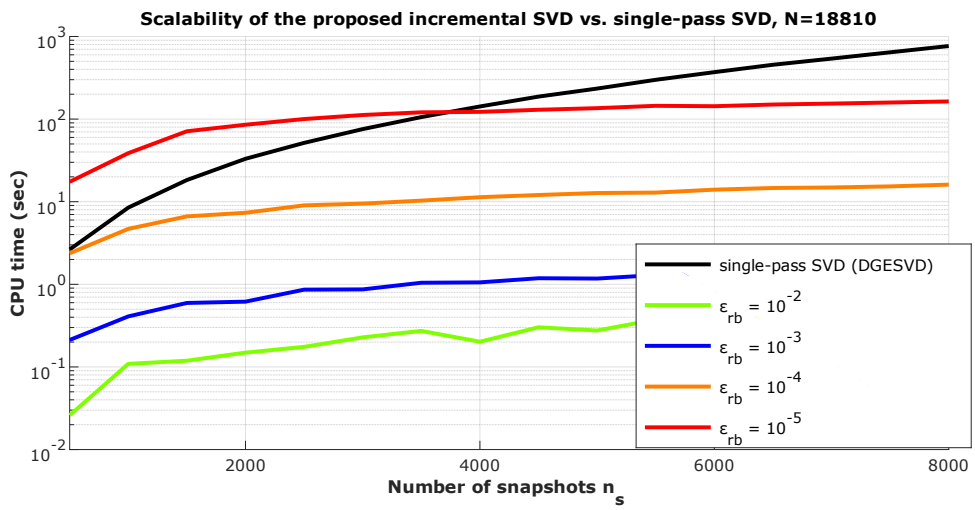


Figure 3.13: Taylor beam impact, Scalability of incremental vs. single-pass SVD, $N = 18810$



to 10^{-4}, independently of the number of snapshots n_s. For higher precision 10^{-5}, the curves are seen to intersect at $n_s = 3800$, with the incremental SVD outperforming the single-pass SVD afterwards. Note, that in the latter case, the incremental SVD's CPU time remain relatively steady after $n_s \approx 2000$, as opposed to the single-pass version. This behavior may be explained by the limited number of relevant basis vectors governing the physical system, confirming the fundamental hypothesis of reduced order modeling, as well as providing excellent scalability to our method.

3.5 Conclusion

In this chapter, we propose an approach reducing memory usage and floating point operations, compared to the regular Singular Value Decomposition (SVD), by proceeding in incremental fashion. By treating snapshots one after the other, it permits 'on-the-fly' selection for enriching and truncating the RB. By doing so, the method avoids computations on possibly redundant data, as well as unnecessary computations on data that will only be truncated later. The memory usage is minimal as only a truncated representation of the data and a single new observation need to be stored at any given time. As a result, this method is particularly efficient in approximating extensive data sets that have low underlying dimensionality, as is usually the case in computational structural dynamics.

The method is only driven by the desired singular value truncation error, avoiding parameters that depend on the nature and dimension of the data. Moreover, the use of this error estimator allows for improved snapshot selection and the 'on-the-fly' RB truncation, ultimately enhancing performance.

The only remaining parameter in our algorithm is the re-orthonormalization parameter, which has been set here to $\epsilon_{orth} = 10^{-14}$. The aim of the re-orthonormalization step between subsequent enrichments is to avoid incurring a significant error while computing $[\Phi][\Phi]^T\{u_i\}$. This means that the optimal value for ϵ_{orth} depends on the actual dimensions of the RB. Further work on optimizing this parameter should conceivably enable additional computational savings.

In the present work, results are given for a sequential algorithm. It goes without saying that a parallel implementation of the incremental SVD is possible following the lines proposed by [42].

Equipped with error estimator presented in this work, the proposed incremental SVD turns out to be more efficient and less problem-dependent than the traditional version. Using this error estimator in place of the usual tolerances makes the proposed method easy to use, and nearly optimal for training RBs for MOR. Within the POD framework, the proposed algorithm has been shown to outperform the original state-of-the-art incremental SVD as well as the single-pass SVD both in the offline training phase as well as the online reduction phase, for the presented Taylor beam impact test case.

Chapter 4

Sparse RB modal subsets

Published in the International Journal for Numerical Methods in Engineering

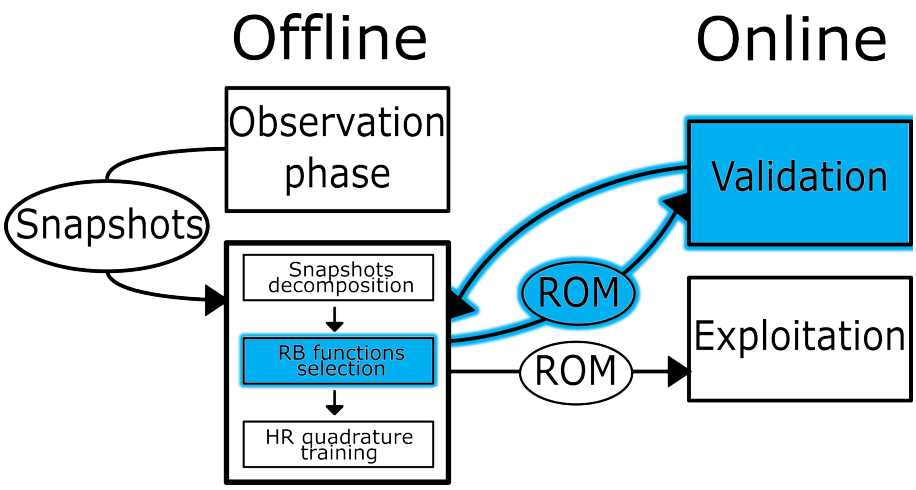
Introduction

Various methods have been proposed to build the optimal RB in different targeted applications, including changes in the norm used in the POD or post-treatment of the training data set. In [40], the H_1 norm is used in the POD low-rank approximation of the compressible Navier-Stokes equations. Different variable separation has been used in POD reduced model of generalized linear micromorphic continua in [37]. The question of whether or not the time Difference Quotients of the snapshots should be used in the generation of the POD basis functions is addressed in [39] in application to the heat equations. In the framework of the structural dynamics, the Adaptive POD (APOD) ([44]) uses the current deformation to select a subset of relevant observations to adapt the ROM. For complex models, the tied contact interface ([83]) enables the coupling between reduced and unreduced substructures. Another challenge in the POD application to structural dynamics is the representation of fracture ([46]). However, it is not clear how to adapt these methods to a particular mechanical problem, and the gains in nonlinear structural transient dynamics models are not clear. Finally, methods have been proposed to further accelerate the computation of the RB, including the incremental SVD ([4, 56, 11, 27]) and randomized algorithms ([34, 6]).

Once the training data decomposed with one of the above methods, the resulting singular vectors are truncated to form a RB subsequently used for online reduction. The Frobenius training set reconstruction error plays the role of the selection criterion of the subset of RB vectors. However, in nonlinear structural dynamics, it does not account for the transient nature of the model and may fail in the online phase, resulting in the blockage of the reduced simulation. In this chapter, we propose a sparse criterion for the selection of singular vectors in the RB accounting for the transient nature of the problem. The proposed criterion is compatible with any variant of the POD independently of the norm used in the covariance matrix computation or the nature of the snapshots.

Referring to Figure 2.2 presented in chapter 2 sketching the workflow of *a posteriori* MOR methods, the features of interest in this chapter are highlighted in blue in Figure 4.1. Snapshots are gathered and decomposed as usual, the method of selection is reviewed and resulting RBs are tested *online* against ones originating from the truncated SVD and imputed threshold on the *offline* training data-set ap-

Figure 4.1: *A posteriori* MOR methods' features impacted by the sparse criterion



proximation.

The chapter is organized as follows. Section 4.1 shortly reviews the POD and the associated RB construction process. In Section 4.2, a pierced plate in traction model is used to generate the training data set, subsequently decomposed with $L^2(\Omega)$ POD and the sequential criterion is shown to form RBs that are not optimal in the online reduction phase. The proposed sparse selection criterion, which takes advantage of the transient information contained in the right singular vectors, is introduced and illustrated in Section 4.3 on the pierced plate example and later tested on a nonlinear hollow beam crash in Section 4.4 with a $L^2(\Omega)$ POD. In the latter example, the sparse selection criterion selects a subset of singular vectors that, at the same size, allows for a better reconstruction of the work of internal forces and approximation of the displacement field in the online reduction phase.

4.1 Data approximation and sequential criterion

The FE method discretizes the displacement $u(x,t)$ over the space Ω in N DoF. In reduced-order transient structural dynamics [3], the unknown displacement $u(x,t)$, defined at each point x of the spatial domain Ω at each time $t \in [0,T]$, is decomposed over time along modal deformations functions $(\phi_h(x))_h$

$$u(x,t) = \sum_{h\in\mathbb{N}^*} \beta_h(t)\phi_h(x), \quad \forall (x,t) \in \Omega \times [0,T]. \tag{4.1}$$

In the discrete setting, separated representation (4.1) may be approximated based on n_s incremental displacement snapshots of the discretized FE solution $(\{u(t_j)\} = (u_1(t_j),\dots,u_N(t_j))^T)_{j\in[\![1,n_s]\!]} \subset \mathbb{R}^N$ taken at different training times $(t_j)_{1\le j\le n_s}$ of the full-scale simulation and stored in the snapshot matrix $[S] = [\{u(t_1)\},\dots,\{u(t_{n_s})\}] \in \mathbb{R}^{N\times n_s}$. A finite number $m = min(N,n_s)$ of orthonormal left singular vectors $(\{\phi_h\})_{h\in[\![1,m]\!]} \subset \mathbb{R}^N$, orthonormal right singular vectors $(\{\psi_h\})_{h\in[\![1,m]\!]} \subset \mathbb{R}^{n_s}$ and associated singular values $(s_h)_{1\le h\le m}$ are then identified by invoking the Singular Value Decomposition (SVD) of $[S]$ so that the discretized

decomposition (4.1) writes

$$\{u(t_j)\}_i = \sum_{h\in[\![1,m]\!]} s_h\{\psi_j\}_h\{\phi_h\}_i = \sum_{h\in[\![1,m]\!]} \beta_h(t_j)\{\phi_h\}_i, \quad \forall(i,j)\in[\![1,N]\!]\times[\![1,n_s]\!] \tag{4.2}$$

where $\{\phi_h\}_i$ refers to the ith coefficient of the hth space function $\{\phi_h\}$. Here and in the remainder of the chapter h, i and j are respectively used to index the space functions of the decomposition, the DoF and the discrete time instants.

The RB is built by selecting a subset of $(\{\phi_h\})_{h\in B}$, of size $k = \sharp(B)$, approximating the training data

$$\{u(t_j)\}_i \approx \{\hat{u}(t_j)\}_i = \sum_{h\in B\subset[\![1,m]\!]} s_h\{\psi_j\}_h\{\phi_h\}_i, \quad \forall(i,j)\in[\![1,N]\!]\times[\![1,n_s]\!]. \tag{4.3}$$

The sequential criterion approximation error of $u(x,t)$ is

$$\epsilon_{sv} = \left(\frac{\int_\Omega\int_{[0,T]}(u-\hat{u})^2 dt d\Omega}{\int_\Omega\int_{[0,T]} u^2 dt d\Omega}\right)^{\frac{1}{2}}. \tag{4.4}$$

However, the displacements of the full system usually being unknown, ϵ_{sv} is approximated offline by the reconstruction error of the training data $[S]$. In the FE framework, replacing the continuous displacement fields u and $\hat{u}$ with the time and space discretized training data $[S]$ and writing the RB in matrix form $[\Phi^{(B)}] = [\{\phi_{B(1)}\},\ldots,\{\phi_{B(k)}\}]$ yields

$$\epsilon_{sv}(B) = \frac{\|[S]-[\Phi^{(B)}][\Phi^{(B)}]^T[S]\|_F}{\|[S]\|_F} = \sqrt{1-\frac{\sum_{h\in B} s_h^2}{\sum_{h=1}^{m} s_h^2}}. \tag{4.5}$$

Given a user-defined targeted error ϵ_{rb}, the sequential criterion seeks the smallest subset $B\subset[\![1,m]\!]$ such that $\epsilon_{sv}(B)\le\epsilon_{rb}$, given by the solution of the optimization problem

For a user-prescribed targeted precision ϵ_{rb}, find $B^*\subset[\![1,m]\!]$ such that

$$\begin{cases} B^* = \underset{B\in\mathbb{B}}{argmin}(\sharp(B)) \\ \mathbb{B} = \{B\subset[\![1,m]\!] \mid \epsilon_{sv}(B)\le\epsilon_{rb}\} \end{cases} \tag{4.6}$$

With singular values sorted out in decreasing order, the minimum of $\epsilon_{sv}(B)$ for a given size of the RB k is achieved by sequentially selecting the first k modes ($B=\{1,2,\ldots,k\}$). Thus, the optimization problem (4.6) is solved by computing (4.5) for increasing values of k, until the target precision ϵ_{rb} is reached. In this article, (4.6) is referred to as the sequential criterion. It is convenient as the associated optimization problem (4.6) is easy to solve given the form (4.5) of the approximation

error. However, this criterion controls the training data $[S]$ averaged approximation precision by a RB $[\Phi^{(B)}]$ in the offline phase only. The cons of the time averaging in the sequential criterion are illustrated in the following section on a structural dynamics example of a pierced plate in traction where the truncated basis is employed in the online phase. Even though POD variations tend to improve the quality of the basis vectors (including H_1 norm POD [40], Adaptive POD [44], Difference Quotients [39]), to our best knowledge little, if any, work has been done enhancing the criterion selecting the optimal subset B regarding the online performances.

4.2 On sub-optimality of the sequential criterion in POD applications to structural dynamics

Consider the FE model of a pierced plate in traction (Figure 4.2). In explicit non-linear structural dynamics, the semi-discretized full-scale model writes

$$[\mathbb{M}]\{\ddot{u}(t)\} + \{f_{int}(\{u(t)\}, t)\} = \{f_{ext}(t)\}. \tag{4.7}$$

$[\mathbb{M}] \in \mathbb{R}^{N\times N}$ is the symmetric positive-definite mass matrix, $\{f_{int}\} \in \mathbb{R}^N$ is the vector of internal forces and $\{f_{ext}\} \in \mathbb{R}^N$ is the vector of external forces. The pierced plate is made of an elasto-plastic steel of density $\rho = 7.82 * 10^{-9}\ \ T.mm^{-3}$, Young's modulus $E = 210000\ \ MPa$ and Poisson's ratio $\nu = 0.3$. The hardening rule is described by the Johnson-Cook law, neglecting temperature and strain rate effects

$$\sigma_{eq} = \sigma_Y + b * \varepsilon_p^n, \tag{4.8}$$

with σ_{eq} the equivalent stress and ε_p the equivalent plastic strain. Other parameters are plastic yield stress $\sigma_Y = 210\ \ MPa$, plasticity hardening coefficient $b = 50\ \ Mpa$ and plasticity hardening exponent $n = 0.25$. 8-node fully integrated solid elements are used to discretize the model into $N = 444$ DoF and three planar symmetries are taken into account so that only one-eight of the plate is computed. Velocity $V_{imp} = 1000\ \ mm.s^{-1}$ is imposed at the borders of the plate, as shown in Figure 4.2, during the $T = 3 * 10^{-2}$ seconds of the simulation. The central difference method is used for time integration along with the lumped mass approach, yielding a diagonal mass matrix.

Snapshots taken from this model are decomposed with SVD and a subset of k left singular vectors is selected to project the FOM (4.7) on, yielding

$$[\tilde{\mathbb{M}}]\{\ddot{\alpha}(t)\} + \{\tilde{f}_{int}(\{\tilde{u}(t)\}, t)\} = \{\tilde{f}_{ext}(t)\} \tag{4.9}$$

refered to as the Reduced Order Model (ROM) and producing the approximate solution $\{\tilde{u}\}$

$$\{\tilde{u}(t)\} = \sum_{i=1}^{k} \alpha_i(t)\{\phi_{B(i)}\}. \tag{4.10}$$

The ROM unknowns are the $(\alpha_i(t))_{1\leq i\leq k}$, $[\tilde{\mathbb{M}}] = [\Phi^{(B)}]^T[\mathbb{M}][\Phi^{(B)}] \in \mathbb{R}^{k\times k}$ is the projection of the mass matrix of the original FOM on the reduced space. Likewise, $\{\tilde{f}_{int}\} = [\Phi^{(B)}]^T\{f_{int}\}$ and $\{\tilde{f}_{ext}\} = [\Phi^{(B)}]^T\{f_{ext}\}$ are the internal and external forces projections onto the reduced space. In practice, RB functions are orthonormalized

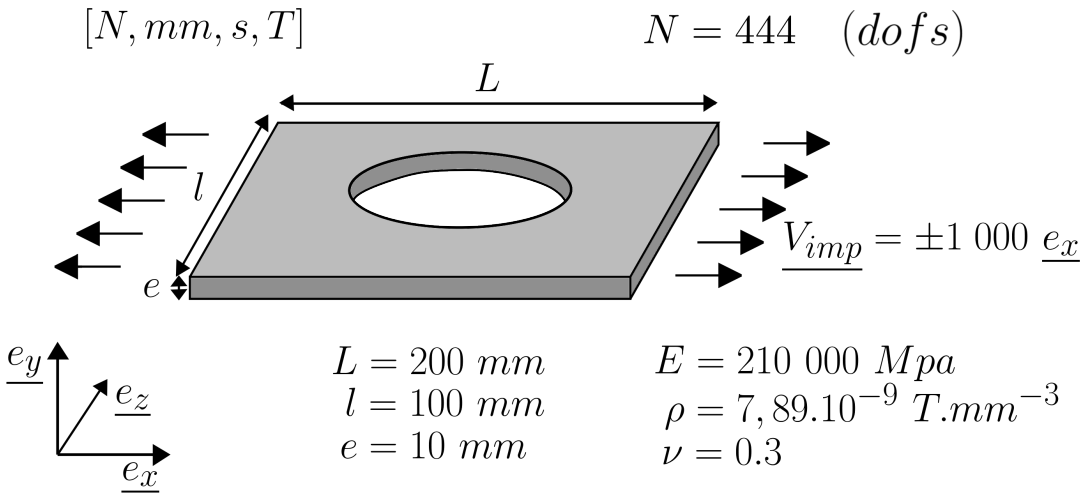


Figure 4.2: Pierced plate in traction model.

with respect to the metric induced by the mass matrix $[\mathbb{M}]$ so that $[\tilde{\mathbb{M}}] = [Id_k]$, where $[Id_k]$ is the identity matrix of $\mathbb{R}^{k\times k}$.

The ROM approximation $\{\tilde{u}\}$ (4.10), computed in the online reduction phase, is not to be confused with the training data projection on the RB $\{\hat{u}\}$ (4.3). While $\{\hat{u}\}$ is obtained by projecting $[S]$ on the RB in the off-line training phase, $\{\tilde{u}\}$ is obtained by expanding the online ROM solution $\{\alpha\}$ to the high dimensional space. Main difference is that for $\{\tilde{u}\}$, the error induced by the use of a RB propagates as the reduced simulation progresses in time. On the contrary, considering that each snapshot $(\{u(t_j)\})_j$ in $[S]$ is projected *separately* on the RB, $\{\hat{u(t_j)}\}$ may be a very poor approximation of $\{u(t_j)\}$ at a given training time j and perfectly represent it after at subsequent training times. In reference POD, the RB modes are selected for their ability to minimize the sequential criterion approximation error (4.5) of the training data approximation $\{\hat{u}\}$, however, this does not imply the same error for the ROM approximation $\{\tilde{u}\}$ in the online reduction phase.

This difference between $\{\hat{u}\}$ and $\{\tilde{u}\}$ is illustrated by performing reduced runs (4.9) for different RBs and comparing online performances with the offline training data approximation. In our application, the work of internal forces W_{int} and the displacement field $\{u\}$ are used to evaluate online performances of a RB. Internal work at a given time is defined by

$$W_{int} = \int_{\Omega} \underline{\underline{\varepsilon}} : \underline{\underline{\sigma}} dV \tag{4.11}$$

with $\underline{\underline{\varepsilon}}$ the strain tensor and $\underline{\underline{\sigma}}$ the stress tensor. Denoting W_{int} the work of internal forces in the FOM and $\tilde{W}_{int}$ the work of internal forces obtained in the online reduction phase, the online error ϵ_{int} considered in this chapter is

$$\epsilon_{int} = \frac{\int_0^T \mid W_{int}(t) - \tilde{W}_{int}(t) \mid dt}{\int_0^T W_{int}(t) dt}. \tag{4.12}$$

W_{int} is a convenient quantity of interest to estimate online performances of a RB, as it is easy to post-process and allows the detection of potential issues in the reduction phase quickly. In all our applications, W_{int} has shown good correlation with

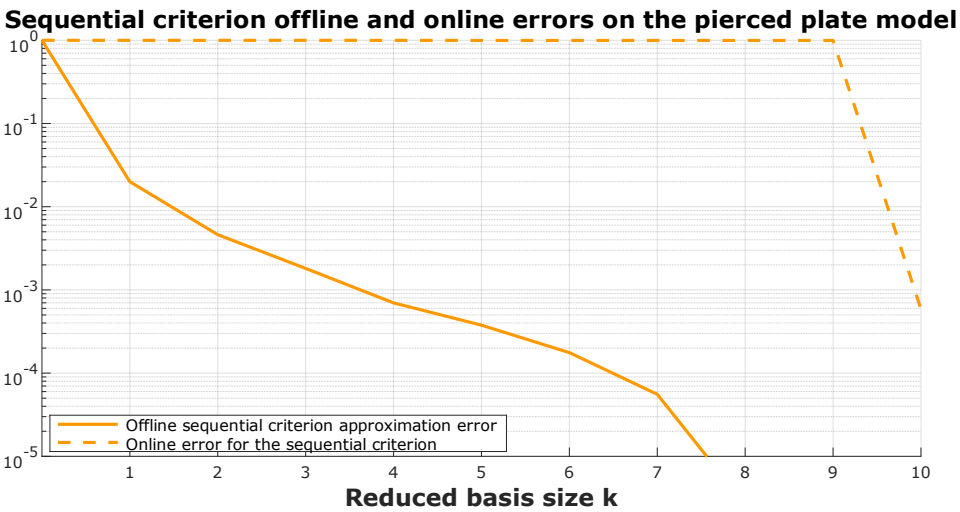


Figure 4.3: Offline sequential criterion approximation error (black curve) and associated online work of internal forces approximation error (gray curve), in function of the size k of the RB k.

Modes selected with the sequential criterion for different values of ϵ_{rb}, associated training data sequential criterion approximation error and online work of internal forces reconstruction error												
$\epsilon_{sv} \leq \epsilon_{rb}$	$\{\phi_1\}$	$\{\phi_2\}$	$\{\phi_3\}$	$\{\phi_4\}$	$\{\phi_5\}$	$\{\phi_6\}$	$\{\phi_7\}$	$\{\phi_8\}$	$\{\phi_9\}$	$\{\phi_{10}\}$	ϵ_{sv}	ϵ_{int}
$5*10^{-2}$	✓										$1.99*10^{-2}$	$9.99*10^{-1}$
$1*10^{-2}$	✓	✓									$4.61*10^{-3}$	$9.99*10^{-1}$
$2*10^{-3}$	✓	✓	✓								$1.81*10^{-3}$	$9.99*10^{-1}$
$1*10^{-3}$	✓	✓	✓	✓							$6.98*10^{-4}$	$9.99*10^{-1}$
$5*10^{-4}$	✓	✓	✓	✓	✓						$3.77*10^{-4}$	$9.96*10^{-1}$
$2*10^{-4}$	✓	✓	✓	✓	✓	✓					$1.76*10^{-4}$	$9.96*10^{-1}$
$6*10^{-5}$	✓	✓	✓	✓	✓	✓	✓				$5.56*10^{-5}$	$9.96*10^{-1}$
$5*10^{-6}$	✓	✓	✓	✓	✓	✓	✓	✓			$2.63*10^{-6}$	$9.96*10^{-1}$
$1*10^{-11}$	✓	✓	✓	✓	✓	✓	✓	✓	✓		$8.38*10^{-12}$	$9.96*10^{-1}$
$1*10^{-12}$	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	0	$5.85*10^{-4}$

Table 4.1: Sequential subsets for different values of threshold ϵ_{rb} and associated offline and online errors ϵ_{sv} (4.5) and ϵ_{int} (4.12), respectively.

the online displacement field error. However, a low approximation error of the work of internal forces does not necessarily imply the same for the displacement field.

In the following test, a snapshot matrix $[S] \in \mathbb{R}^{444\times 10}$ storing $n_s = 10$ displacement snapshots, uniformly sampled from the full-scale pierced plate in traction FE model, is used to reduce the model. Ten basis vectors and associated singular values are computed with SVD and used to build a POD reduced order model.

Figure 4.3 plots the offline training data sequential criterion approximation error (4.5) (continuous orange) as well as associated work of internal forces approximation error (4.12) (dotted orange) in the online reduction phase for ten RBs composed of sequentially selected subsets of $k = 1, \ldots, 10$ left singular vectors, obtained using the sequential criterion. These results are reported in the last two columns of Table 4.1 together with the corresponding prescribed error ϵ_{rb} imposed on the sequential criterion approximation error and associated sequentially selected subsets of the basis vectors. Work of internal forces in the online reduction phase for each case is compared to that of the FOM in Figure 4.4. The final configuration for

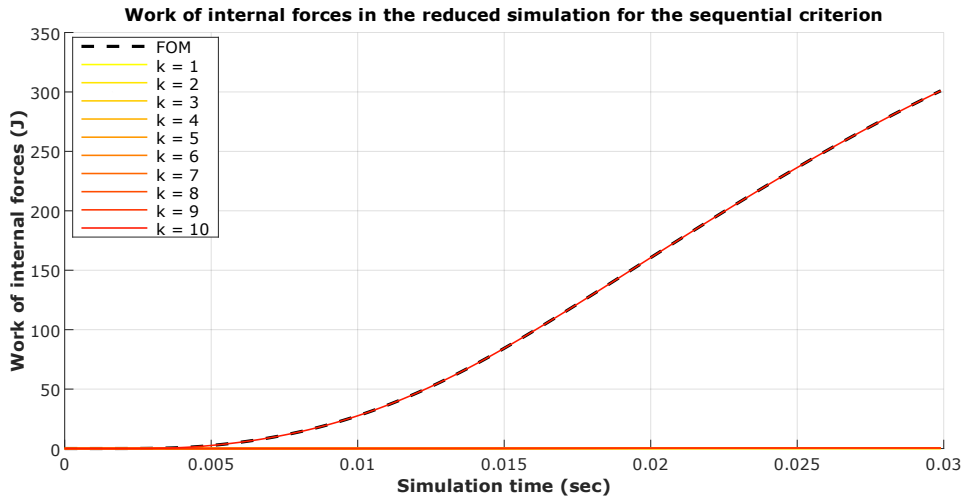


Figure 4.4: Work of internal forces in the reduced run for different sequential RBs of size k is quasi-null in all reduced runs except for the non-truncated RB of size $k = 10$ (red curve) that perfectly fits the work of internal forces in the reference FOM (black dotted curve).

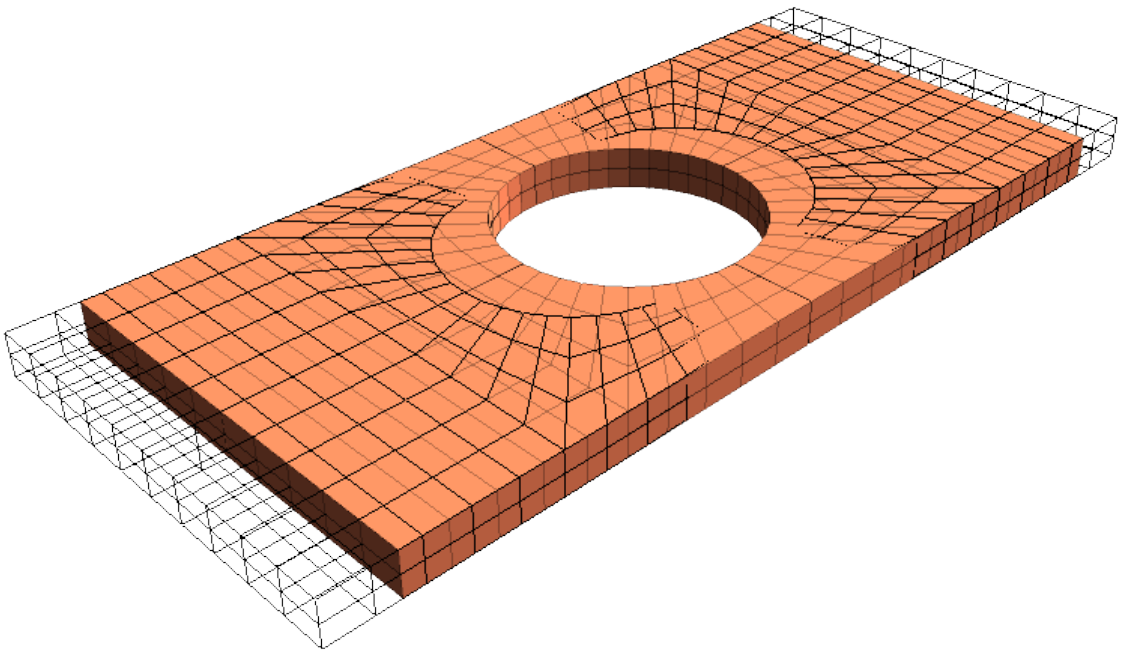


Figure 4.5: Final displacement fields comparison between the reference FOM (black wireframe grid) and the reduced model (solid orange) for a RB of size 8 formed using the sequential criterion, putting to evidence the lag between the FOM and the ROM.

the sequential RB of size 8 is shown with solid orange elements in Figure 4.5 and compared to the reference final configuration of the FOM represented with a black wireframe grid.

$\epsilon_{sv}(B)$ falls below 1% with only two basis vectors out of ten ($[\Phi^{(B)}] = [\{\phi_1\}, \{\phi_2\}]$). This confirms the funding hypothesis of PROM that the dimension of the underlying manifold, where the discretized solution of a computational mechanics problem evolves, is small compared to the number of DoF. Unlike the offline training error, the work of internal forces approximation error in reduced runs does not decrease when enriching the RB. Looking at the work of internal forces in Figure 4.4, we notice that it is actually null for all reduced runs except for the last one which is in a good agreement with the reference FOM. These results extend to the displacement field: the ROM approximation $\{\tilde{u}\}$ is in perfect agreement with the FOM when all ten RB functions are selected and null for RBs of size 1 to 9 as seen in Figure 4.5 for a RB of size 8.

This example shows that imposing a low threshold on the training data sequential criterion approximation error does not imply the same for the online reduction phase. A significant concern in the POD application to nonlinear structural dynamics is to find the possibly sparse RB of minimal size that allows for a given approximation quality in the online reduction phase and, in doing so, avoid expensive offline training of a ROM that will fail in the online reduction phase. On this example, despite the contribution of $\{\phi_{10}\}$ to the offline sequential criterion approximation error (4.5) being nearly null, it's absence in the RB seems to prevent the ROM from computing any deformation. This behavior is better understood in the following section 4.3 where a sparse criterion for the selection of the left singular vectors appearing in the training data decomposition is proposed.

4.3 Sparse subset selection criterion

This section is organized as follows: the previous results are interpreted, a sparse selection criterion is proposed and then used to form RBs that, based on the same training data, improves the online approximation compared with the previous results. To understand the offline/online discrepancy phenomenon observed in section 4.2, we first take a look at the transient contribution $\beta_h(t_j)$ of each left singular vector to the reconstruction of $\{u(t_j)\}$. The training data (4.2), and their approximation with a subset B of modes (4.3), rewrite

$$u_i(t_j) = \sum_{h=1}^{m} \sigma_h \{\phi_h\}_i \{\psi_j\}_h = \sum_{h=1}^{m} \beta_h(t_j)\{\phi_h\}_i, \quad \forall (i,j) \in [\![1, N]\!] \times [\![1, n_s]\!], \tag{4.13}$$

and,

$$\{u(t_j)\} \approx \{\hat{u}(t_j)\} = \sum_{h \in B} \beta_h(t_j)\{\phi_h\}, \quad \forall j \in [\![1, n_s]\!], \tag{4.14}$$

respectively.

Once the spatial discretization performed, $\beta_h(t_j)$ at each training time t_j, $1 \leq j \leq n_s$, give relative transient contributions of $\{\phi_h\}$, $1 \leq h \leq m$

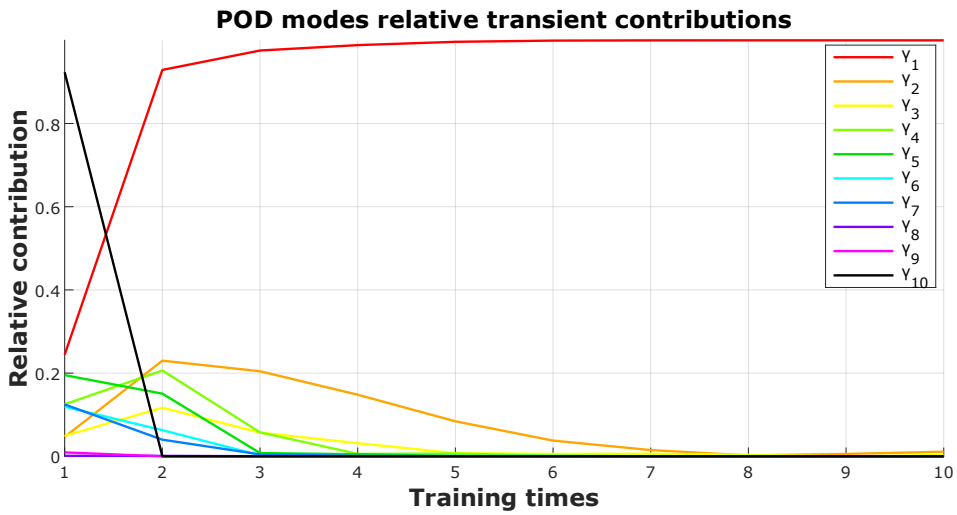


Figure 4.6: Relative transient contributions (4.15) of the ten left singular vectors for the pierced plate in traction test case.

$$\gamma_h^j = \sqrt{\frac{\beta_h^2(t_j)}{\sum_{l=1}^{m} \beta_l^2(t_j)}}, \quad \forall (h,j) \in [\![1,m]\!] \times [\![1,n_s]\!], \tag{4.15}$$

reported in Figure 4.6 for the pierced plate test case presented in the section 4.2. As expected, first left singular vectors contribute in average more as they are associated with the highest singular values. However, this is not true when looking at their contribution to the solution reconstruction at each training time separately. In Figure 4.6, $\{\phi_{10}\}$ turns out to contribute significantly more than any other basis vector at the first training time and to have negligible contribution afterward. Once averaged in time by the sequential criterion approximation error (4.4), contribution of $\{\phi_{10}\}$ vanishes. Apparently, $\{\phi_{10}\}$ represented in Figure 4.7 as a displacement field with respect to the initial configuration, corresponds to the elongation of the first elements impacted by the imposed velocity, essential to represent the initial condition in the online reduction phase. This observation is consistent with the observation that truncating $\{\phi_{10}\}$ prevents the pierced plate from stretching in the reduced simulation, as observed in section 4.2.

A novel sparse criterion is proposed hereafter to overcome this limitation associated with the sequential criterion, guaranteeing that the projection error $\{\hat{u}\}$ in $L^2(\Omega)$ norm meets the target approximation precision ϵ_{RB} at each training time. We introduce the sparse criterion approximation error

$$\epsilon_{sp} = \sup_{t\in[0,T]} \left(\left(\frac{\int_\Omega (u(\underline{x},t) - \hat{u}(\underline{x},t))^2 d\underline{x}}{\int_\Omega u^2(\underline{x},t) d\underline{x}} \right)^{\frac{1}{2}} \right) \tag{4.16}$$

Replacing the, in practice unknown, continuous displacement field u in (4.16) with

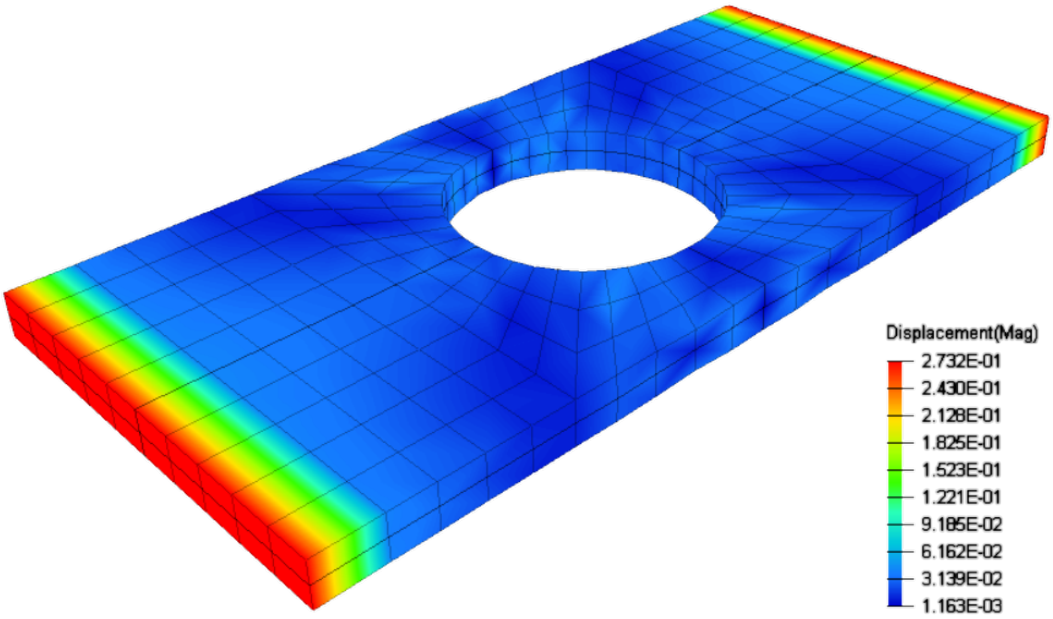


Figure 4.7: Tenth left singular vector for the pierced plate in traction FE model represented as a displacement field with respect to the model initial configuration.

the training data $(\{u(t_j)\})_{j\in[\![1,n_s]\!]}$ yields

$$\epsilon_{sp}(B) = \max_{j\in[\![1,n_s]\!]} \left(\frac{\|\{u(t_j)\} - [\Phi^{(B)}][\Phi^{(B)}]^T\{u(t_j)\}\|_2}{\|\{u(t_j)\}\|_2} \right) = \max_{j\in[\![1,n_s]\!]} \left(\sqrt{1 - \frac{\sum_{h\in B} \beta_h(t_j)}{\sum_{h=1}^{m} \beta_h(t_j)}} \right), \tag{4.17}$$

where the last form is obtained using (4.14) together with the orthonormality of the left singular vectors.

To approximate the training data up to a given target ϵ_{rb} while selecting the fewest possible basis vectors, the modified optimization problem (4.6) writes

Given a user-prescribed targeted precision ϵ_{rb}, find $B^* \subset [\![1,m]\!]$ such that

$$\begin{cases} B^* = \underset{B\in\mathbb{S}}{argmin}(\sharp(B)) \\ \mathbb{S} = \{B \subset [\![1,m]\!] \mid \epsilon_{sp}(B) \leq \epsilon_{rb}\}. \end{cases} \tag{4.18}$$

The admissible space $\mathbb{S}$ is rewritten using

$$\epsilon_{sp}^2 = \max_{j\in[\![1,n_s]\!]} (1 - \frac{\sum_{h\in B} \beta_h^2(t_j)}{\sum_{l=1}^{m} \beta_l^2(t_j)}) = 1 - \min_{j\in[\![1,n_s]\!]} (\frac{\sum_{h\in B} \beta_h^2(t_j)}{\sum_{l=1}^{m} \beta_l^2(t_j)}) \Rightarrow 1 - \epsilon_{sp}^2 = \min_{j\in[\![1,n_s]\!]} (\sum_{h\in B)} (\gamma_h^j)^2),$$

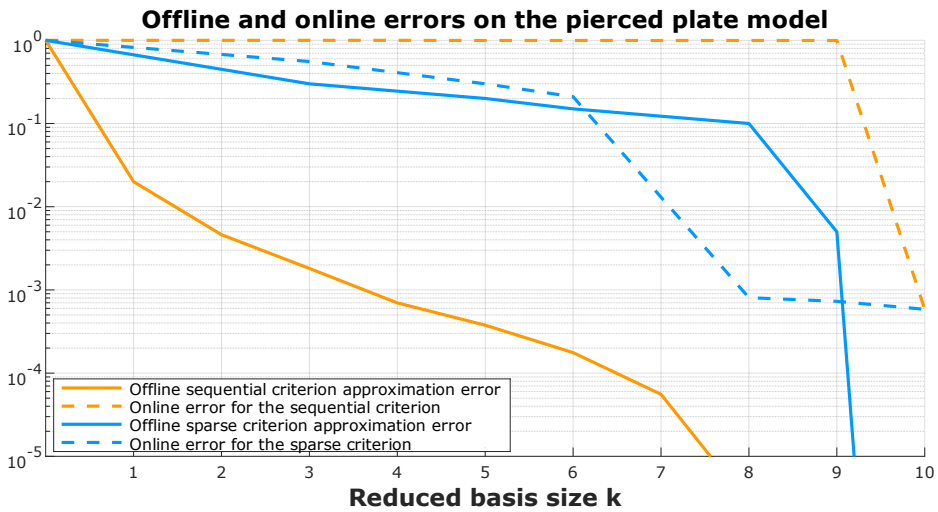


Figure 4.8: Offline (black) and online work of internal forces (gray) approximation errors in function of the size of the RB k for both sequential (continous curve) and sparse criteria (dotted curve).

which finally leads to the following form of the optimization problem

Given a user-prescribed targeted precision ϵ_{rb}, find $B^* \subset [\![1, m]\!]$ such that

$$\begin{cases} B^* = \underset{B \in \mathbb{S}}{argmin}(\sharp(B)) \\ \mathbb{S} = \{B \subset [\![1, m]\!] \mid \sum_{h \in B} (\gamma_h^j)^2 \geq 1 - \epsilon_{rb}^2, \quad \forall j \in [\![1, n_s]\!]\}. \end{cases} \tag{4.19}$$

Unlike (4.6), optimization problem (4.19) does not have an explicit solution. However, it may be conveniently stated for use with combinatorial optimization tools using the form

$$\begin{cases} \{\xi^*\} = \underset{\{\tilde{\xi}\} \in \mathbb{S}}{argmin}(\|\{\tilde{\xi}\}\|_1) \\ \mathbb{S} = \{\{\tilde{\xi}\} \in \{0, 1\}^m \mid ([\Gamma]\{\tilde{\xi}\})_j \geq 1 - \epsilon_{rb}^2, \quad \forall j \in [\![1, n_s]\!]\}, \end{cases} \tag{4.20}$$

where $\{\tilde{\xi}\}$ is a boolean vector in $\{0, 1\}^m$ such that the hth entry of the vector is 1 if, and only if, $\{\phi_h\}$ is selected. $[\Gamma] \in \mathbb{R}^{n_s \times m}$ coefficients are given by $[\Gamma]_{j,h} = (\gamma_h^j)^2, \quad \forall (h, j) \in [\![1, m]\!] \times [\![1, n_s]\!]$, $\|\bullet\|_1$ stands for the L_1 norm that associates to a vector the sum of its coefficients, corresponding in the considered application to the number of selected left singular vectors. In the present work, (4.20) is solved with the simplex algorithm implementation [38].

The proposed sparse criterion is tested on the pierced plate in traction model introduced in section 4.2.

Figure 4.8 plots the work of internal forces (4.12) (gray) approximation error in the online reduction phase for sparse RBs (dotted curve) together with the two

Modes selected with the sparse criterion for different values of ϵ_{rb}, associated training data sparse criterion approximation error and online work of internal forces reconstruction error												
$\epsilon_{sp} \leq \epsilon_{rb}$	$\{\phi_1\}$	$\{\phi_2\}$	$\{\phi_3\}$	$\{\phi_4\}$	$\{\phi_5\}$	$\{\phi_6\}$	$\{\phi_7\}$	$\{\phi_8\}$	$\{\phi_9\}$	$\{\phi_{10}\}$	ϵ_{sp}	ϵ_{int}
$3 * 10^{-1}$	✓	✓								✓	$3 * 10^{-1}$	$5.56 * 10^{-1}$
$2 * 10^{-1}$	✓	✓		✓	✓					✓	$2 * 10^{-1}$	$3 * 10^{-1}$
$1.5 * 10^{-1}$	✓	✓		✓	✓	✓				✓	$1.5 * 10^{-1}$	$2.1 * 10^{-1}$
$1 * 10^{-1}$	✓	✓	✓	✓	✓	✓	✓			✓	$1 * 10^{-1}$	$8.07 * 10^{-4}$
$5 * 10^{-3}$	✓	✓	✓	✓	✓	✓	✓		✓	✓	$5 * 10^{-3}$	$7.29 * 10^{-4}$
$1 * 10^{-10}$	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	$1 * 10^{-16}$	$5.85 * 10^{-4}$

Table 4.2: Sparse subsets of modes selected for different values of the threshold ϵ_{rb}, associated offline sparse criterion ϵ_{sp} (4.17) and online work of internal forces ϵ_{int} (4.12) errors.

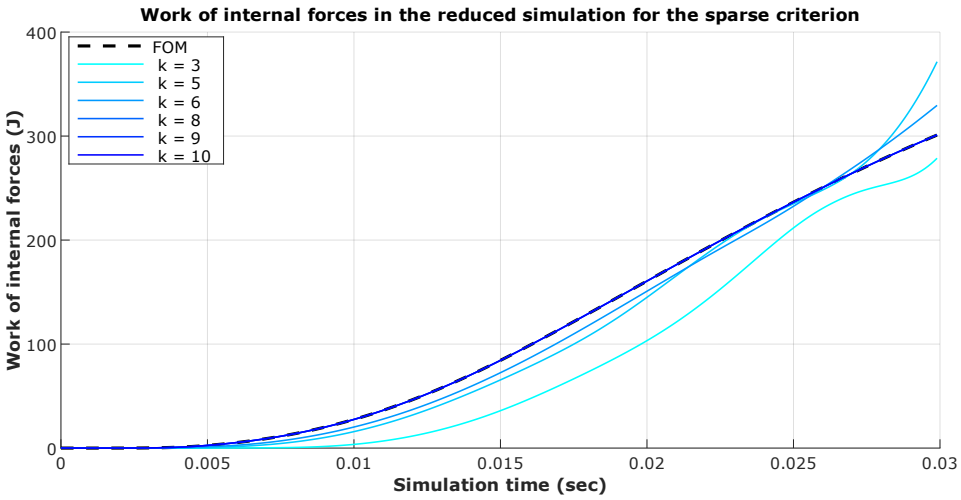


Figure 4.9: Work of internal forces in the online reduction phase for the sparse RBs.

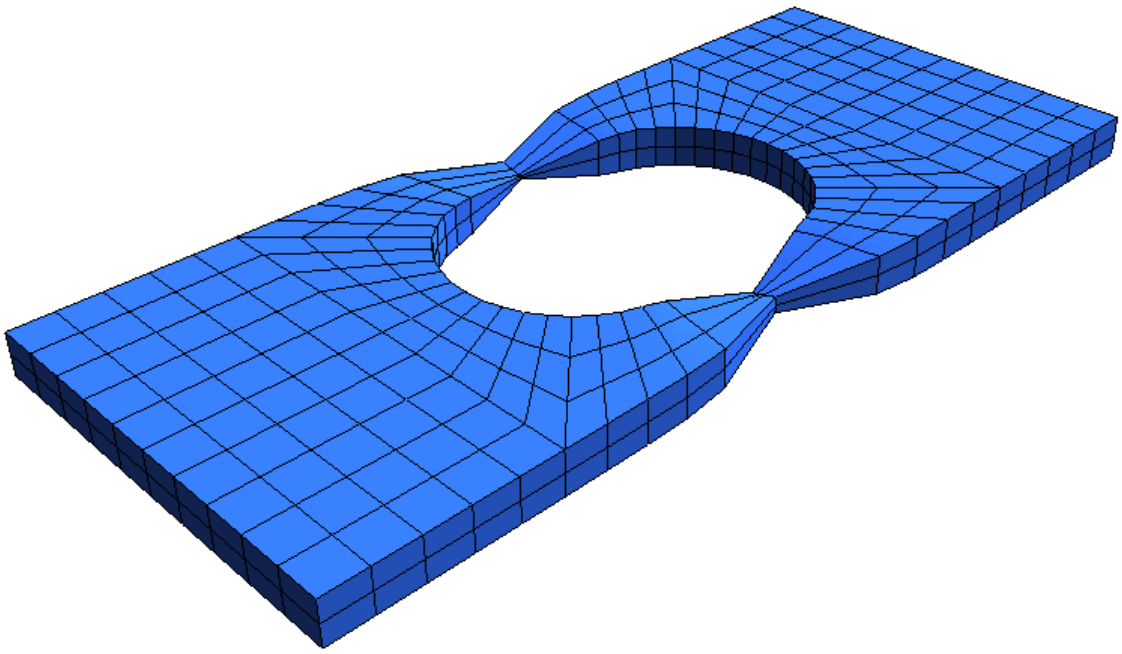


Figure 4.10: Final configuration displacement fields comparison between the reference FOM (black wireframe grid) and the reduced model (solid blue) for a sparse RB of size 8, putting to evidence an improved agreement between the FOM and the ROM.

curves already present in Figure 4.3. Corresponding subsets of basis vectors selected using the proposed criterion are presented in Table 4.2 as well as the corresponding target error ϵ_{rb}, sparse criterion approximation error ϵ_{sp} in the offline training phase for the formed RB and the work of internal forces approximation error ϵ_{int} in the online reduction phase. The work of internal forces in the online phase is compared to that of the FOM in Figure 4.9. The final displacement fields of the FOM (black wireframe grid) and the ROM associated with the sparse RB of size 8 (solid gray) are represented in Figure 4.10.

These results show that, when compared with the sequential criterion, the sparse criterion selects a subset of non-consecutive basis vectors allowing for better online performance. In Figure 4.8, the average relative error of the online approximation of the work of internal forces is, for any given size of the RB, better with the sparse criterion. Offline and online curves do not coincide, as the offline approximation error is, for both criteria, not directly related to the work of internal forces. However, for the sparse criterion, offline approximation error and online work of internal forces error are in better agreement as they strictly decrease together when enriching the RB, which was not the case with the sequential criterion. Furthermore, the offline and online indicators are closer to each other for the sparse than for the sequential criterion. As seen in table 4.2, the sparse criterion selects $\{\phi_{10}\}$ earlier than the sequential one, resulting in a higher sequential criterion approximation error in the offline phase but a better work of internal forces approximation online. In Figure 4.9, reduced simulations start producing work of internal forces with only 3 RB functions and runs are in good agreement with that of the FOM with only 5 RB functions as opposed to the sequential criterion unable to trigger the imposed velocity and

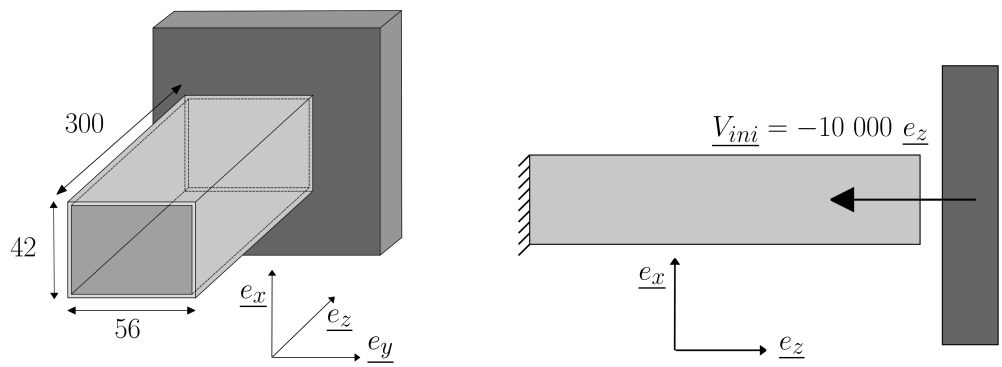


Figure 4.11: Scheme of the crash box model.

to produce work of internal forces until the RB reaches the full set of ten basis functions (Figure 4.4). In Figure 4.9, the work of internal forces approximation error in the online phase is delayed, in opposition to the sequential criterion tending to emphasize the importance of snapshots taken at late simulation times as they generally are redundant and of higher magnitudes compared with early stages of the simulation. Displacement field in the final configuration of the reduced simulation (solid gray in Figure 4.10), for a RB of size 8, is in agreement with that of the FOM (black wireframe grid) which was not the case for the sequential RB of the same size (Figure 4.5).

4.4 Test on a crash box model

Both criteria are now tested for the reduction of the more complicated and representative case of a hollow beam impact (crash box) model, presented in Figure 4.11, consisting of a hollow beam of dimensions $300 \times 42 \times 56$ and thickness of 1.5 millimeters. The material is elasto-plastic steel with density $\rho = 7.89 * 10^{-9}\ T.mm^{-3}$, Young's modulus $E = 210000\ MPa$, Poisson's ratio $\nu = 0.3$, plastic yield stress $\sigma_Y = 792MPa$, plasticity hardening coefficient $b = 510$ and plasticity hardening exponent $n = 0.26$. The hardening rule is described with the same Johnson-Cook law (4.8) as in the pierced plate model. The beam is impacted by a wall of 0.45 Tons, modeled as a rigid body on which initial velocity of 10000 $mm.s^{-1}$ is imposed. Fully integrated 8-node solid elements are employed and $N = 15432$ DoF.
The central difference method is used for time integration along with the lumped mass approach.

RBs from size 12 to 15 are built from 214 incremental displacement snapshots, taken uniformly in time, with both selection methods. Work of internal forces obtained with different reduced runs is plotted in Figure 4.12 as well as the FOM reference. Selected subsets of modes for each RB, associated sequential criterion approximation errors of reconstruction of the training data and work of internal forces errors in the online reduction phase are reported in Table 4.3 for the sparse criterion and in Table 4.4 for the sequential criterion, respectively. A comparison of the final displacement field between both selection methods against the FOM reference is provided in Figure 4.13.

The proposed sparse criterion outperforms the sequential one in the online re-

Subsets of modes obtained with the sequential criterion for different sizes of the RB, associated training data sequential criterion approximation error and online work of internal forces reconstruction error							
size of the RB k	$\{\{\phi_1\},\ldots,\{\phi_{12}\}\}$	$\{\phi_{13}\}$	$\{\phi_{14}\}$	$\{\phi_{15}\}$	$\{\phi_{16}\}$	ϵ_{sv}	ϵ_{int}
12	✓					$1.25 * 10^{-5}$	$3.15 * 10^{-1}$
13	✓	✓				$1.08 * 10^{-5}$	$3.24 * 10^{-1}$
14	✓	✓	✓			$9.2 * 10^{-6}$	$3.31 * 10^{-1}$
15	✓	✓	✓	✓		$8.1 * 10^{-6}$	$1.82 * 10^{-1}$
16	✓	✓	✓	✓	✓	$7.2 * 10^{-6}$	$1.81 * 10^{-1}$

Table 4.3: Sequential subsets of modes of increasing size, associated offline error on training data ϵ_{sv}, and online error of the work of internal forces ϵ_{int} for the crash box model.

Subsets of modes obtained with the sparse criterion for different size of the RB, associated training data sparse criterion approximation error and online work of internal forces reconstruction error									
size of the RB k	$[\{\phi_1\},\ldots,\{\phi_{10}\}]$	$\{\phi_{11}\}$	$\{\phi_{12}\}$	$\{\phi_{13}\}$	$\{\phi_{15}\}$	$\{\phi_{16}\}$	$\{\phi_{18}\}$	ϵ_{sp}	ϵ_{int}
12	✓		✓			✓		$4.47 * 10^{-4}$	$3.63 * 10^{-2}$
13	✓		✓		✓	✓		$4 * 10^{-4}$	$3.66 * 10^{-2}$
14	✓	✓	✓			✓	✓	$3.16 * 10^{-4}$	$2.36 * 10^{-2}$
15	✓	✓	✓		✓	✓	✓	$3 * 10^{-4}$	$3 * 10^{-2}$
16	✓	✓	✓	✓	✓	✓	✓	$2.25 * 10^{-4}$	$3.22 * 10^{-2}$

Table 4.4: Sparse Subsets of modes for different sizes of the RB, associated offline sparse criterion approximation error on training data ϵ_{sp}, and online error of the work of internal forces ϵ_{int} for the crash box model.

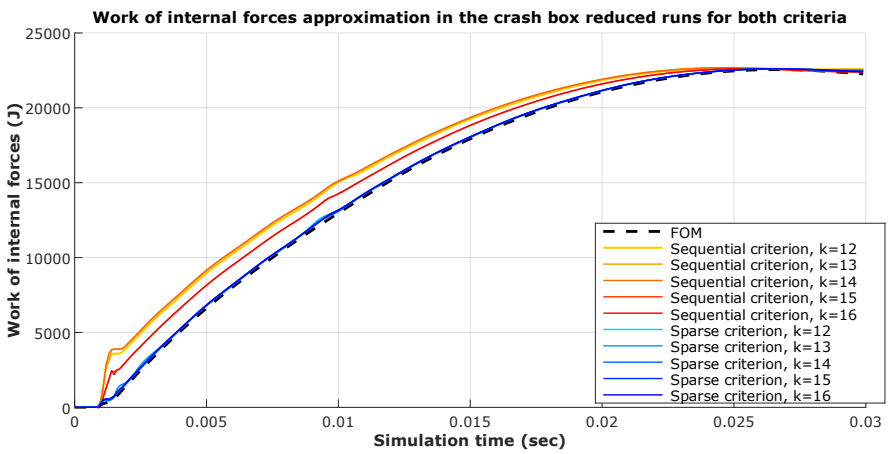


Figure 4.12: Comparison of the work of internal forces in the crash box model for sequential and sparse RBs.

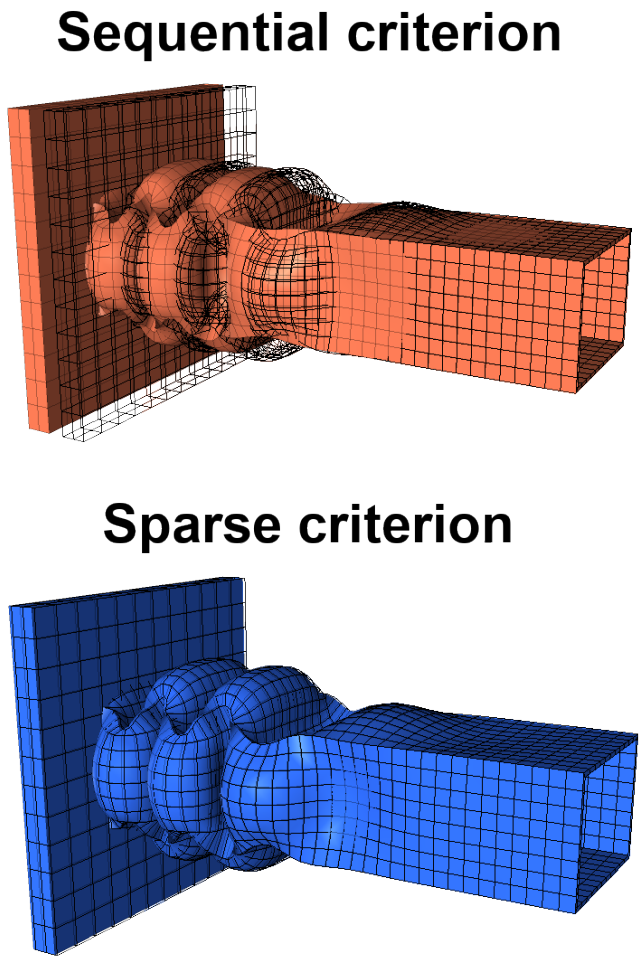


Figure 4.13: Final configuration displacement fields comparison between the FOM (black wireframe grid) and the reduced model (solid elements) for RB of size 12 for both sequential and sparse criteria. The lag between the two models apparent for the sequential criterion is visually cancelled for the proposed sparse criterion.

construction of both the work of internal forces (Figure 4.12) and displacement field (Figure 4.13). For sequential RBs, work of internal forces within reduced runs start differing from the reference FOM at initial contact time with the rigid wall. As stated for the pierced plate example, this behavior may be explained by the tendency of the sequential criterion to emphasize the importance of snapshots taken at the end of the simulation. The online approximation error compensates later in the simulation when considering the work of internal forces but not in the displacement field, represented in Figure 4.13 at the final stage of the simulation, appearing since the first, poorly approximated by the RB, contact with the rigid wall. The proposed sparse criterion contributes to resolving this issue, by prescribing the same precision for all snapshots, regardless of their magnitude. As a consequence, for a given size, sparse RBs induce one order of magnitude lower work of internal forces approximation error within the reduced run, as seen in Tables 4.3 and 4.4, moreover, offline and online errors are in better accordance with each other. Sparse RBs are still associated with relatively large singular values. 214 left singular vectors have been computed on this example; however, only the first 18 modes are used in the considered RBs. An interesting feature of the sparse criterion is observed with decreasing values of the user-prescribed ϵ_{rb} in Table 4.4 when already selected modes are removed from subsequent subsets B when incrementing k. When enriching the RB from size $k = 13$ to size $k = 14$, the sparse criterion adds $\{\phi_{18}\}$ and $\{\phi_{12}\}$ to the subset of RB funstions and removes $\{\phi_{15}\}$. Results in Table 4.3 and Table 4.4 highlight this behavior that remains to be understood. Sparse RBs of size $k = 12$ and $k = 13$ offer better online work of internal forces approximation than the sequential one of size $k = 16$, even though the latter contains all sparse basis functions. This result has been verified multiple times e.g., when removing $\{\phi_{11}\}$ and $\{\phi_{13}\}$ from the RB of size $k = 16$ formed with the sequential criterion enhances online approximation of the work of internal forces but not the displacement field approximation. Both selection criteria are implemented in a custom version of the state-of-the-art crash simulation code Altair Radioss [3].

4.5 Conclusion

This chapter proposes a sparse criterion to constitute subsets of RB modes, originating from the training data decomposition, for efficient online simulations. The sparse criterion ensures a target approximation quality for each snapshot as opposed to the sequential one which averages the approximation quality over all snapshots. Thus, it is insensible to redundant snapshots with high orders of magnitude and avoids bad approximation of possibly important isolated snapshots with low orders of magnitude.

Both criteria are tested for the selection of basis vectors originating from the usual L^2 SVD of incremental displacement snapshots taken from a pierced plate in traction model and a hollow beam impact model. Given an appropriate training data set and an associated SVD, the sparse criterion results in subsets of RB functions that allow for a more accurate approximation in the online POD reduced simulations in nonlinear transient structural dynamics.

For the pierced plate in traction and the hollow beam impact models, sparse RBs perform better in terms of online approximation of both the work of internal

forces and final configuration displacement field than sequential RBs of same sizes.

Nevertheless, several topics remain open: The choice of metrics to compare ROM performances in the reduced phase in terms of approximation quality depends on the quantity of interest. The time-averaged relative error of the work of internal forces used here as an indicator of ROMs approximation quality in the online reduction phase is not a perfect comparator: this error is not strictly decreasing with increasing user-prescribed precision for both selection methods, and the work of internal forces may be underestimated in the online reduction phase, as is the case for the pierced plate, as well as overestimated, as is the case for the crash box. In the second example, enriching a RB does not necessarily enhance the ROMs approximation of the work of internal forces, so performances in the online reduction phase are also compared in terms of final configuration displacement field approximation. Another perspective concerns parametric variations between the offline training phase and the online reduction phase which are not taken into account in this chapter. Finally, the sparse criterion induces a computational overhead as it requires right SVD vectors and solving of a combinatorial optimization problem. The combinatorial optimization problem, which may be efficiently solved due to the low dimensionality of the data. A greedy algorithm could avoid both the computation of the right singular vectors and avoid solving a combinatorial optimization problem for bigger models.

Chapter 5

MIP Hyper-reduction

Submitted to Advanced Modeling and Simulation in Engineering Sciences

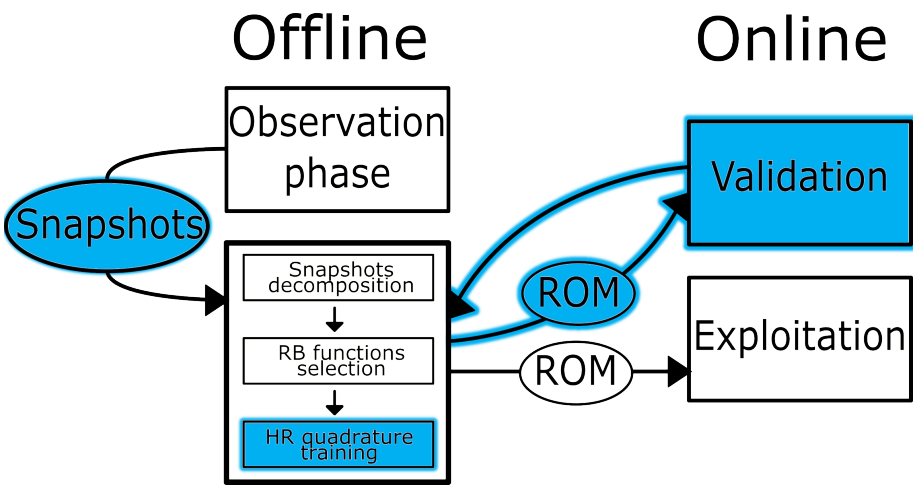
Introduction

When applied to explicit nonlinear dynamics ([45, 46]), POD does not reduce the complexity of evaluating internal variables and entails a computational overhead in the 'online' reduction phase due to the necessity of computing internal forces over all elements. Hyper-Reduction (HR) methods are combined to PROM to tackle the internal forces computational complexity and achieve 'online' speed-up by selecting a representative subset of elements for internal forces approximation.

Miscellaneous HR methods exist in the literature: The Discrete Empirical Interpolation Method (DEIM)([17, 24, 77]) reduces the complexity of evaluating nonlinear internal variables combining projection and interpolation. DEIM originates from the Empirical Interpolation Method ([8]) and aims to approximate the projection of the internal forces by using only a few of their components. Other HR approaches directly approximate the projected internal forces by building a reduced spatial integration scheme. The multidimensional 'a priori' hyper-reduction (APHR)([73]) creates a reduced integration domain without any knowledge from the FOM. In the present chapter, we consider the Energy-Conserving Sampling and Weighting (ECSW)([28]) formulation of an optimization problem, based on data collected from the FOM, to build a hyper-reduced integration scheme whose particularity is to impose integration weights positivity ensuring energy conservation of the reduced integrator. ECSW is an 'a posteriori' method as it relies on data from full-scale simulations to train a hyper-reduced integration scheme, as opposed to 'a priori' methods which use the knowledge of the physical problem rather than training data ([73]). The optimization problem that arises in the ECSW is NP-hard and is in practice suboptimally solved by mean of the Sparse Non-Negative Least Square (SNNLS) greedy algorithm. Similar numerical integration schemes, in which it is necessary to preserve the consistency and compatibility during the shape functions integration, arise in Galerkin meshless methods such as Diffuse Elements ([14]), and Element Free Galerkin ([66, 30, 18]).

Present work is motivated by the observation that, proceeding greedily by subsequent enrichments, the SNNLS algorithm becomes computationally expensive in building large hyper-reduced integration schemes , and that a reference method is needed to compare performances of heuristic approaches. Reformulating the ECSW optimization problem, an alternative linear optimization problem is stated, solved using Mixed Integer Programming (MIP), and illustrated on a benchmark FE model.

Figure 5.1: *A posteriori* MOR methods' features impacted by the MIP formulation



Different formulations incorporating consistency conditions and constraint reduction are tested. Resulting integration schemes are compared in terms of number and position of integration points, and error in the work of internal forces within the 'online' reduction phase.

Referring to Figure 2.2 presented in chapter 2 sketching the workflow of *a posteriori* MOR methods, the features of interest in this chapter are highlighted in blue in Figure 5.1. The optimization problem used in *offline* HR training is reviewed, and additional snapshots on integration points position and elemental volumes are taken to impose new constraints on polynomial integration and volume conservation. Resulting HR quadratures are used *online* in the validation phase, comparing the reconstructed work of internal forces to that obtained with the SNNLS greedy procedure.

The chapter is organized as follows. Section 5.1 briefly reviews PROM and ECSW methods in a nonlinear explicit structural dynamics framework. Section 5.2 develops the theory behind the proposed linear MIP formulation, which is then tested in Section 5.3 on a pierced plate under uniform tension FE model. Results are then discussed, and recommendations are made regarding future developments.

5.1 Projected hyper-reduced order model

5.1.1 Projected Reduced Order Model

We focus on semi-discretized equations used in nonlinear explicit FE solvers

$$[\mathbb{M}]\{\ddot{u}(t)\} + \{f_{int}(\{u(t)\}, t)\} = \{f_{ext}(t)\}, \tag{5.1}$$

where $\{u(t)\} \in \mathbb{R}^N$ is the vector of nodal Degrees of Freedom (DoF), $\{f_{int}\} \in \mathbb{R}^N$ is the vector of nonlinear internal forces and $\{f_{ext}\} \in \mathbb{R}^N$ is the vector of external forces. N denotes the number of DoF and is referred to as the FOM size. $[\mathbb{M}] \in \mathbb{R}^{N\times N}$ is the symmetric positive-definite mass matrix. In explicit nonlinear structural dynamics, industrial FE solvers frequently use a lumped mass approach yielding diagonal $[\mathbb{M}]$.

Hence there is no need to inverse the mass matrix, and the computational effort concentrates on the internal and contact forces evaluation.
The explicit central difference method is used for integration in time domain.

Given a RB $[\Phi] \in \mathbb{R}^{N\times k}$ of size k, the ROM approximates the FOM solution $\{u\}$ with $\{\tilde{u}\}$,

$$\{u(t)\} \approx \{\tilde{u}(t)\} = \sum_{i=1}^{k} \alpha_i(t)\{\phi_i\} = [\Phi]\{\alpha(t)\}. \tag{5.2}$$

Above approximation is injected into the FOM (5.1) which is then projected on the RB, yielding the PROM

$$[\tilde{\mathbb{M}}]\{\ddot{\alpha}(t)\} + [\Phi]^T\{f_{int}([\Phi]\{\alpha(t)\}, t)\} = [\Phi]^T\{f_{ext}(t)\}. \tag{5.3}$$

$\{\alpha\} \in \mathbb{R}^k$ is the ROM unknown and $[\tilde{\mathbb{M}}] = [\Phi]^T[\mathbb{M}][\Phi] \in \mathbb{R}^{k\times k}$ is the reduced mass matrix. $[\Phi]$ is usually $[\mathbb{M}]$-orthonormalized so that $[\tilde{\mathbb{M}}]$ is the identity matrix $[Id_k] \in \mathbb{R}^{k\times k}$. The ROM is of size k, much smaller than N,

$$\{\ddot{\alpha}(t)\} = [\Phi]^T\Big(\{f_{ext}(t)\} - \{f_{int}([\Phi]\{\alpha(t)\}, t)\}\Big). \tag{5.4}$$

Projecting the FOM (5.1) employing an orthogonal matrix $[\Phi]$ potentially increases the critical time step ensuring numerical stability. A rigorous mathematical proof of this property is provided in [7]. However, in applications to explicit structural dynamics, PROM methods generally yield a computational overhead. First, there is no real gain in reducing the model size to k when the lumped mass approach is used in the FOM (5.1) as a diagonal mass matrix inverse is computed with negligible computational effort. Second, internal forces have to be computed for all elements, involving the time-consuming integration of the constitutive law, and two additional steps are required: expansion of reduced DoF $\{u\} = [\Phi]\{\alpha\}$ and contraction of internal forces $\{\tilde{f}_{int}\} = [\Phi]^T\{f_{int}\}$. The same goes for kinematic conditions, as well as any requested output.

However, as internal forces vector of size N is to be projected on the reduced space of size k spanned by the columns of the RB $[\Phi]$ and $k << N$, it may not be necessary to compute the whole internal forces vector to have a good approximation of its projection on the reduced space, and this is where hyper-reduction takes place. The following section briefly reminds the ECSW scheme motivating this work.

5.1.2 Energy-Conserving Sampling and Weighting (ECSW)

The Galerkin method used in FE analysis in the divide and conquer spirit successively computes internal forces $\{f^e_{int}\} \in \mathbb{R}^N$ in each of n_e elements of the model and assembles respective contributions

$$\{f_{int}\} = \sum_{e=1}^{n_e}\{f^e_{int}\} \tag{5.5}$$

In the ROM (5.4), internal forces are projected on $[\Phi] \in \mathbb{R}^{N\times k}$

$$[\Phi]^T\{f_{int}\} = [\Phi]^T\sum_{e=1}^{n_e}\{f^e_{int}\}. \tag{5.6}$$

Hyper-reduction computes internal forces only for a subset $\mathcal{H} \subset [\![1, n_e]\!]$ of elements indexes and applies weights ζ_e^* to the elemental contributions ahead of summation and projection on the reduced space

$$[\Phi]^T\{f_{int}\} \approx [\Phi]^T \sum_{e=1}^{n_e} \zeta_e^* \{f_{int}^e\} = [\Phi]^T \sum_{e \in \mathcal{H}} \zeta_e^* \{f_{int}^e\}, \tag{5.7}$$

where $\{\zeta\}^* = (\zeta_1^*, \zeta_2^*, \ldots, \zeta_{n_e}^*)^T \in \mathbb{R}_{\geq 0}^{n_e}$ contains weights associated with all elements in the model. $\zeta_e^* = 0$ if and only if element e is not selected ($e \notin \mathcal{H}$). The ECSW method imposes also $\zeta_e^* > 0$ for selected elements to maintain the integrator positivity.

The hyper-reduced integration scheme, given by the subset of selected elements $\mathcal{H}$ and associated weights $\{\zeta^*\}$, is obtained through optimization. Given a RB $[\Phi]$ and unassembled internal forces $(\{f_{int}^e(t_i)\})_{(e,i) \in [\![1,n_e]\!] \times [\![1,n_s]\!]}$, the hyper-reduced quadrature scheme integrates the projected unassembled internal forces training data set up, to a user-defined precision, while selecting the fewest possible elements in $\mathcal{H}$. First, n_s unassembled internal forces snapshots are collected at training times $(\{t_i\}, \quad i \in [\![1, n_s]\!]$. Without loss of generality, snapshots may as well be taken at different model parameters values. Once collected, unassembled internal forces $(\{f_{int}^e(t_i)\})_{(e,i) \in [\![1,n_e]\!] \times [\![1,n_s]\!]} \subset \mathbb{R}^N$ are projected on the reduced space, yielding $([\Phi]^T\{f_{int}^e(t_i)\})_{(e,i) \in [\![1,n_e]\!] \times [\![1,n_s]\!]} \subset \mathbb{R}^k$, and are organized in the matrix

$$[G] = \begin{bmatrix} G_f(t_1) \\ G_f(t_2) \\ \vdots \\ G_f(t_{n_s}) \end{bmatrix} \in \mathbb{R}^{k*n_s \times n_e}, \tag{5.8}$$

where

$$[G_f(t_i)] = [\Phi^T][\{f_{int}^1(t_i)\}, \{f_{int}^2(t_i)\}, \ldots, \{f_{int}^{n_e}(t_i)\}] \in \mathbb{R}^{k \times n_e}, \quad \forall i \in [\![1, n_s]\!]. \tag{5.9}$$

Using this notations, the non-reduced assembly process (5.5) writes

$$\{b\} = [G]\{1_{n_e}\} \in \mathbb{R}^{k*n_s}, \tag{5.10}$$

with $\{\zeta^*\} = \{1_{n_e}\}$ corresponding to the selection of all elements with integration weights equal to 1 and $\{b\} \in \mathbb{R}^{k*n_s}$ is the 'exact' projection of all internal forces snapshots on $[\Phi]$, used as reference to train the weights. Finally, given a targeted precision τ, the optimization problem of the hyper-reduced integration scheme is stated as

$$\begin{cases} \{\zeta^*\} = \underset{\{\zeta\} \in \mathcal{A}_{ECSW}}{argmin} \; (\|\{\zeta\}\|_0) \\ \mathcal{A}_{ECSW} = \{\{\zeta\} \in \mathbb{R}_{\geq 0}^{n_e} \mid \dfrac{\|[G]\{\zeta\} - \{b\}\|_2}{\|\{b\}\|_2} \leq \tau\} \end{cases} \tag{5.11}$$

The threshold τ on the approximation precision constraint is imposed in the admissible space $\mathcal{A}_{ECSW}$ alongside weights positivity, $\| \bullet \|_0$ denotes the zero-norm associating the number of its non-zero coefficients to a vector, equivalent to the number of selected finite elements, to be minimized. However, the zero norm is

not differentiable, making (5.11) NP-hard. In practice (5.11) is suboptimally solved with greedy algorithms such as SNNLS ([28])(Algorithm 2). Alternatives such as the LASSO algorithm have been compared to SNNLS in [16]. In the present work, (5.11) is reformulated and solved using MIP optimization, proposing a non-greedy reference for testing quadrature schemes.

5.2 Hyper-Reduced Integration

5.2.1 MIP formulation

MIP solves problems involving both integer and real variables based on a combinatorial Branch And Bound algorithm for integer unknowns and simplex algorithm for real variables. (5.11) is transformed into a problem that may be solved using MIP in the following steps. An additional boolean unknown $\{\xi\} \in \{0,1\}^{n_e}$ is introduced to take account of the elements affiliation to $\mathcal{H}$

$$\xi_e = \begin{cases} 1 & , \quad e \in \mathcal{H} \\ 0 & , \quad e \in [\![1, n_e]\!] \backslash \mathcal{H}. \end{cases}$$

Mixed optimization variables are thus $(\{\zeta\}, \{\xi\}) \in \mathbb{R}^{n_e}_{\geq 0} \times \{0,1\}^{n_e}$. The link between the non-negative real weights and the Boolean selectors is: if $\xi_e = 0$ for a given element e, the element is not selected and $\zeta_e = 0$ is imposed. A maximal value for the weights is prescribed

$$-\infty \leq \zeta_e - \zeta_{max}\xi_e \leq 0, \quad \forall e \in [\![1, n_e]\!], \tag{5.12}$$

bounding each selected element weight with ζ_{max} and each unselected element weight with 0 and linearizing the problem. Together with the weight non-negativity condition, (5.12) ensures

$$(5.12) \cap (\{\zeta\} \in \mathbb{R}^{n_e}_{\geq 0}) \Rightarrow \begin{cases} \zeta_e = 0 & \text{if} \quad \xi_e = 0 \\ 0 \leq \zeta_e \leq \zeta_{max} & \text{if} \quad \xi_e = 1. \end{cases}$$

Similarly, a minimal value ζ_{min} is imposed on each selected element weight by the constraint

$$0 \leq \zeta_e - \zeta_{min}\xi_e \leq \infty, \quad \forall e \in [\![1, n_e]\!]. \tag{5.13}$$

(5.13) imposed with $\zeta_{min} = 1$ ensures that each selected element counts at least for itself while avoiding over-fitting.

Additionally, defining ξ_{min} and ξ_{max}, the minimal and maximal number of elements in $\mathcal{H}$, respectively $(1 \leq \xi_{min} \leq \xi_{max} \leq n_e)$

$$\xi_{min} \leq \sum_{e \in E} \xi_e \leq \xi_{max}, \tag{5.14}$$

allows to target prescribed intervals for the number of selected elements. This feature, in contrary to usual greedy algorithms progressively enriching the selection, allows to directly start searching for solutions from a given ξ_{min} or assess unattainability of the targeted precision for a given ξ_{max}, reducing computation time by

shrinking the admissible space.

Finally, for a given threshold τ, the MIP optimization problem writes

$$\begin{cases} (\{\zeta^*\},\{\xi^*\}) = \underset{(\{\zeta\},\{\xi\})\in\mathcal{A}_{MIP}}{argmin} \left(\sum_{e\in[\![1,n_e]\!]} \xi_e\right) \\ \mathcal{A}_{MIP} = \{(\{\zeta\},\{\xi\}) \in \mathbb{R}^{n_e}_{\geq 0} \times \{0,1\}^{n_e} \mid \begin{bmatrix} \{b\}-\tau \\ -\{\infty\} \\ \xi_{min} \end{bmatrix} \leq [C] \begin{bmatrix} \{\zeta\} \\ \{\xi\} \end{bmatrix} \leq \begin{bmatrix} \{b\}+\tau \\ \{0_{n_e}\} \\ \xi_{max} \end{bmatrix}\}. \end{cases} \tag{5.15}$$

Note, that the threshold τ on the hyper-reduced quadrature scheme precision is now imposed in L^∞ norm and not in L^2 norm in contrary to the ECSW optimization problem (5.11). The linear constraint matrix is

$$[C] = \begin{bmatrix} [G] & \{0_{n_e}\}^T \\ [Id] & [diag(\zeta_{max}\{1_{n_e}\})] \\ \{0_{n_e}\}^T & \{1_{n_e}\}^T \end{bmatrix}, \tag{5.16}$$

with $[G]$ defined by equation (5.8).

5.2.2 Consistency constraints

When internal forces functions are strongly varying in space and time, such as in car-crash analysis, exact domain integration may not be possible. Exactness in the projected Galerkin hyper-reduced scheme is not guaranteed even if the RB functions possess sufficient completeness to represent the solution. A similar problem arises in the Element Free Galerkin context ([19, 14]), where the exactness in the Galerkin approximation, conditioned by the numerical verification of volume and divergence equalities, is met provided additional zero and first-order integration constraints, respectively. The present work also investigates the impact of adding consistency conditions at training times $(t_i)_{i\in[\![1,n_s]\!]}$ such as exact volumic and polynomial integration up to a given degree. Volume preservation writes

$$\sum_{e=1}^{n_e} \zeta_e \omega_e(t_i) = \sum_{e=1}^{n_e} \omega_e(t_i), \tag{5.17}$$

where $\omega_e(t_i)$ denotes the volume of element e at training time t_i, $i \in [\![1,n_s]\!]$, without loss of generality, sampled at the same training times as unassembled internal forces. Constraint (5.17) is expressed in the MIP formalism by appending $[G_v]$ to (5.16)

$$[G_v] = \begin{bmatrix} \omega_1(t_1) & \dots & \omega_{n_e}(t_1) \\ \vdots & & \vdots \\ \omega_1(t_{n_s}) & \dots & \omega_{n_e}(t_{n_s}) \end{bmatrix} \in \mathbb{R}^{n_s\times n_e}. \tag{5.18}$$

In a similar fashion, first degree polynomials integration is imposed at training times $t_i, \quad i \in [\![1, n_s]\!]$

$$\begin{cases} \sum_{e=1}^{n_e} \zeta_e \omega_e(t_i) g_1^{(e)}(t_i) = \sum_{e=1}^{n_e} \omega_e(t_i) g_1^{(e)}(t_i) & (5.19) \\ \sum_{e=1}^{n_e} \zeta_e \omega_e(t_i) g_2^{(e)}(t_i) = \sum_{e=1}^{n_e} \omega_e(t_i) g_2^{(e)}(t_i) & (5.20) \\ \sum_{e=1}^{n_e} \zeta_e \omega_e(t_i) g_3^{(e)}(t_i) = \sum_{e=1}^{n_e} \omega_e(t_i) g_3^{(e)}(t_i). & (5.21) \end{cases}$$

A single integration point per element is considered at the center of element e, $\{g^{(e)}(t_i)\} = (g_1^{(e)}(t_i), g_2^{(e)}(t_i), g_3^{(e)}(t_i))^T$ at training time t_i. Constraints on first degree polynomials are prescribed by further appending the following matrix $[G_p]$ to $[G]$

$$[G_p] = \begin{bmatrix} \omega_1(t_1) g_1^{(1)}(t_1) & \dots & \omega_{n_e}(t_1) g_1^{(n_e)}(t_1) \\ \vdots & & \vdots \\ \omega_1(t_1) g_1^{(1)}(t_1) & \dots & \omega_{n_e}(t_{n_s}) g_1^{(n_e)}(t_{n_s}) \\ \omega_1(t_1) g_2^{(1)}(t_1) & \dots & \omega_{n_e}(t_1) g_2^{(n_e)}(t_1) \\ \vdots & & \vdots \\ \omega_1(t_1) g_2^{(1)}(t_1) & \dots & \omega_{n_e}(t_{n_s}) g_2^{(n_e)}(t_{n_s}) \\ \omega_1(t_1) g_3^{(1)}(t_1) & \dots & \omega_{n_e}(t_1) g_3^{(n_e)}(t_1) \\ \vdots & & \vdots \\ \omega_1(t_1) g_3^{(1)}(t_1) & \dots & \omega_{n_e}(t_{n_s}) g_3^{(n_e)}(t_{n_s}) \end{bmatrix}, \tag{5.22}$$

while $\{b\}$ in the left and right members of (5.15) is computed using equation (5.10) with modified $[G]$.

5.2.3 Constraints reduction

Overfitting is a significant concern when training a hyper-reduced integration scheme over a set of collected data. Redundant snapshots may eclipse others and result in inadequate internal forces approximation in the online reduction phase. Moreover, an excessively large data set may lead to an unnecessarily large number of linear constraints, deteriorating performances of the MIP solver. A constraint reduction in the optimization problem (5.15) is thus proposed to address those two issues.

In this section and the remainder of the chapter, $[G]$ dimensions are denoted $m \times n_e$. Data is decomposed using Singular Value Decomposition (SVD). Note, that as the SVD decomposes the second dimension of the matrix and, in the present case, the constraints are to be reduced, the decomposition is performed on $[G]^T$ of a presumably lower number of lines

$$[G]^T = [\Upsilon][diag(\{\lambda\})][\Pi]^T, \tag{5.23}$$

yielding $[\Upsilon]$ of dimensions $n_e \times n_e$ and $[\Pi]$ of dimensions $m \times n_e$ as well as $[\lambda] = diag(\lambda_1, \lambda_2, ..., \lambda_{n_e}) \in \mathbb{R}^{n_e \times n_e}$ containing singular values $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$ arranged in descending order. Using (5.23), the constraint $[G]\{\xi\} = [G]\{1_{n_e}\}$ writes

$$[\Pi][diag(\{\lambda\})][\Upsilon]^T\{\zeta\} = [\Pi][diag(\{\lambda\})][\Upsilon]^T\{1_{n_e}\} \Rightarrow [\Upsilon]^T\{\zeta\} = [\Upsilon]^T\{1_{n_e}\}. \tag{5.24}$$

Keeping only the first $l < n_e$ columns of $[\Upsilon]$ yields the first form of the reduced constraints,

$$[\tilde{\Upsilon}]^T\{s\} = [\tilde{\Upsilon}]^T\{1_{n_e}\}, \tag{5.25}$$

with

$$[\tilde{\Upsilon}] = [\Upsilon]_{:,1:l} \in \mathbb{R}^{n_e \times l}, \tag{5.26}$$

where l is chosen so that $\lambda_1 \geq \cdots \geq \lambda_l > 0$. As a consequence, $[\tilde{\Upsilon}]$ has l linearly independent columns and the rank theorem yields

$$\begin{aligned} ker([\tilde{\Upsilon}]^T) &= n_e - rank([\tilde{\Upsilon}]^T) \\ &= n_e - rank([\tilde{\Upsilon}]) \\ &= n_e - l > 0. \end{aligned}$$

$[\tilde{\Upsilon}]^T$ kernel is non-empty and may be identified with QR factorization.

$$[\tilde{\Upsilon}] = [[Q_1], [Q_2]] \begin{bmatrix} [R] \\ \{0_{(n_e-l)\times l}\} \end{bmatrix} \tag{5.27}$$

with $[Q_1] \in \mathbb{R}^{n_e \times l}$ and $[Q_2] \in \mathbb{R}^{n_e \times (n_e - l)}$ orthonormal matrices and $[R] \in \mathbb{R}^{l \times l}$ an upper triangular matrix. Constraints (5.25) rewrite

$$[\tilde{\Upsilon}]^T\{\zeta\} = [\tilde{\Upsilon}]^T\{1_{n_e}\} \Rightarrow [\tilde{\Upsilon}]^T(\{\zeta\} - \{1_{n_e}\}) = \{0_l\}.$$

Substituting variable $\{\zeta\} = \{1_{n_e}\} + \{w\}$

$$\Rightarrow [\tilde{\Upsilon}]^T\{w\} = \{0_l\}$$

and injecting QR factorization (5.27) yields

$$\begin{aligned} &\Rightarrow ([R]^T[Q_1]^T + [0_{(n_e-l)\times l}]^T[Q_2]^T)\{w\} = \{0_l\} \\ &\Rightarrow [Q_1]^T\{w\} = \{0_l\} \\ &\Rightarrow \exists\{\chi\} \in \mathbb{R}^{(n_e-l)} \text{ such that } \{w\} = [Q_2]\{\chi\}. \end{aligned}$$

The final form of the unknown integration weights is

$$\{\zeta\} = \{1_{n_e}\} + [Q_2]\{\chi\}. \tag{5.28}$$

For every $\{\chi\} \in \mathbb{R}^{n_e - l}$, $\{\zeta\}$ defined by equation (5.28) verifies implicitly (5.25). Constraint on weights non-negativity writes

$$(\{\zeta\} \in \mathbb{R}^{n_e}_{\geq 0}), \Rightarrow -\{1_{n_e}\} \leq [Q_2]\{\chi\} \leq \infty \tag{5.29}$$

and constraint (5.12) implies

$$(5.12) \Rightarrow -\{\infty\} \leq [Q_2]\{\chi\} - \zeta_{max}\{\xi\} \leq -\{1_{n_e}\}. \tag{5.30}$$

Finally, the reduced optimization problem is

$$\begin{cases} (\{\chi^*\}, \{\xi^*\}) = \underset{(\{\chi\},\{\xi\})\in\mathcal{A}_{MIP}}{argmin} \left(\sum_{e=1}^{n_e} \xi_e\right) \\ \mathcal{A}_{RMIP} = \{(\{\chi\}, \{\xi\}) \in \mathbb{R}^{(n_e-l)}_{\geq 0} \times \{0,1\}^{n_e} \mid \begin{bmatrix} -\{1_{n_e}\} \\ -\infty \\ \xi_{min} \end{bmatrix} \leq [C_r] \begin{bmatrix} \{\chi\} \\ \{\xi\} \end{bmatrix} \leq \begin{bmatrix} \infty \\ -\{1_{n_e}\} \\ \xi_{max} \end{bmatrix}\}, \end{cases} \tag{5.31}$$

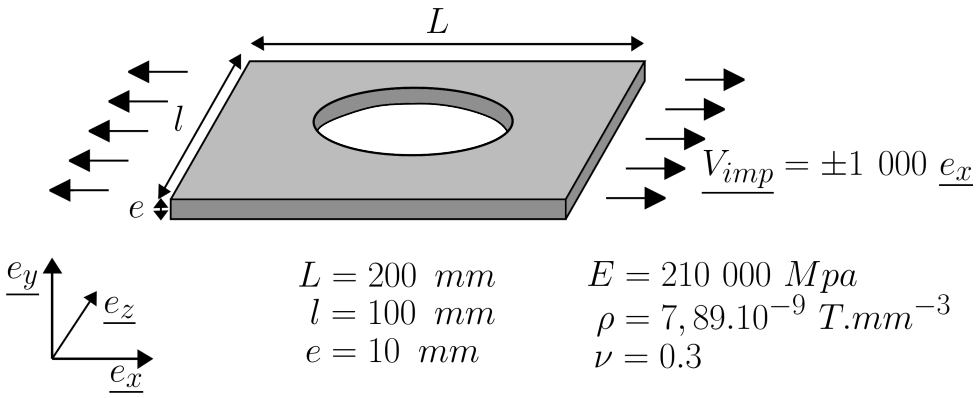


Figure 5.2: Test case model scheme

with the reduced constraint matrix

$$[C_r] = \begin{bmatrix} [Q_2] & \{0_{n_e}\}^T \\ [Q_2] & [-diag(s_{max})] \\ \{0_{(n_e-l)}\}^T & \{1_{n_e}\}^T \end{bmatrix}. \tag{5.32}$$

In this problem there are $2n_e + 1$ linear constraints and $2n_e - l$ MIP unknowns, yielding more contraints than unknowns. Yet, depending on the value of ξ_{min}, ξ_{max} and s_{max}, it is always possible to find a solution. With $(\xi_{min}, \xi_{max}) = (1, n_e)$ and $s_{max} \geq 1$, a trivial solution is $\{\chi\} = \{0\}$ and $\{\xi\} = \{1_{n_e}\}$. Non-trivial solutions exist for every ξ_{max} greater or equal to $(n_e - l)$, $\xi_{min} = 1$. As a matter of fact, the kernel of $[\tilde{\Upsilon}]$, being spanned by the columns of $[Q_2]$ of dimension l, it is possible to satisfy the constraints by selecting less than $n_e - l$ elements.

5.3 Results and discussion

Approaches proposed in the present work are implemented in the legacy FE solver Altair Radioss [3] modified for research purposes. The mixed optimization problems are solved with CPLEX [38]. In this section, different hyper-reduced integration schemes are computed with the state-of-the-art SNNLS algorithm and compared with the proposed approaches in terms of numbers and positions of selected elements, offline training data, and online work of internal forces approximation errors.

The model used for comparison is presented in Figure 5.2 using the unit system $[T, mm, s, MPa]$. The plate is of dimensions $1000 \times 2000 \times 100 \quad mm^3$ and is composed of steel of density $\rho = 7.89E - 9 \quad T.mm^{-2}$ modeled with Johnson-Cook elasto-plastic material law. The plate is stretched with imposed velocity $\boldsymbol{V} = \pm 10000.\{e_x\} \quad mm.s^{-1}$. Parameters of the material law are: Young modulus $E = 210000$ MPa, Poisson ratio $\mu = 0.3$, yield stress $\sigma_Y = 290$ MPa, hardening parameter $b = 50$ and hardening exponent $n = 0.25$. Three planar symmetries allow simulating only an eighth of the plate using $N = 444$ degrees of freedom and $n_e = 55$ elements. Final configuration and Von Mises constraints are represented in Figure 5.3. No parametric variations are considered in this work and the training times correspond to different discrete times $(t_i)_{i \in [\![1, n_s]\!]}$ of the same simulation. In this test case, an RB $[\Phi]$ of size eight is used, and $n_s = 1148$ unassembled internal

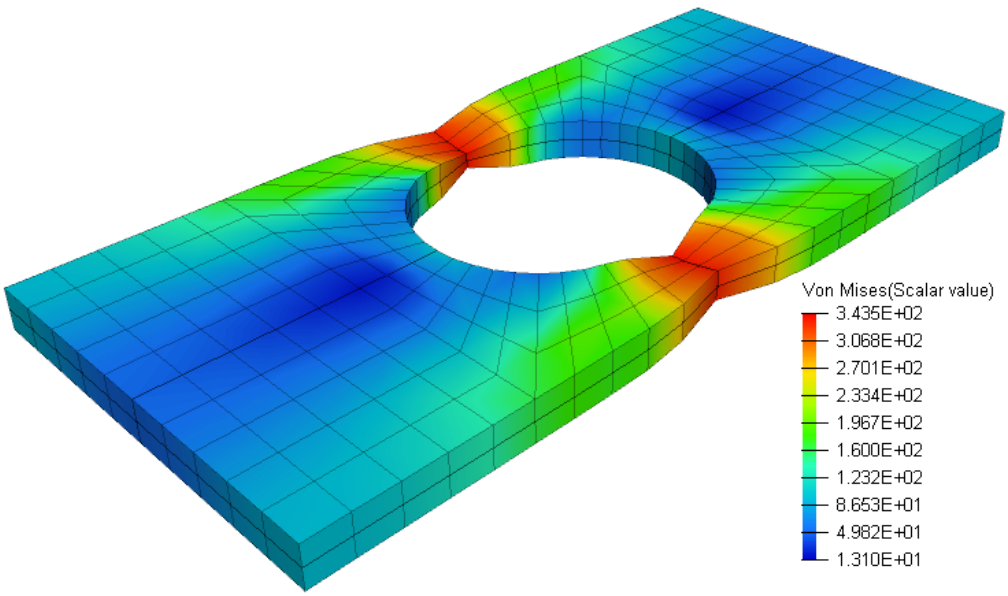


Figure 5.3: Von Mises constraints in the pierced plate in traction model

forces snapshots are taken uniformly during the simulation. For consistency conditions, elements' centers and volumes are gathered at the same simulation times as unassembled forces snapshots.

Results obtained in this example are presented in two groups. The impact of adding consistency conditions is first studied in section 5.3.1 while constraints reduction is tested in section 5.3.2.

5.3.1 Consistency conditions

In this section, the following hyper-reduced formulations are tested both in the offline training phase, in terms of training data approximation error at given quadratures size, and in the online reduction phase, in terms of the work of internal forces reconstruction. These tests include:

- SNNLS greedy Algorithm 2 solving the ECSW optimization problem (5.11);
- MIP optimization problem (5.15);
- MIP + V, MIP optimization problem (5.15) with the consistency conditions (5.18) on volume integration;
- MIP + V + P1, MIP optimization problem (5.15) with the consistency conditions (5.18) on volume integration and (5.22) on first order polynomial integration;
- MIP + G1, MIP optimization problem (5.15) with a minimal weight value $\zeta_{min} = 1$.

The reduced run using the RB $[\Phi]$ of size eight without hyper-reduction is used as reference to investigate the impact of the different hyper-reduction methods.

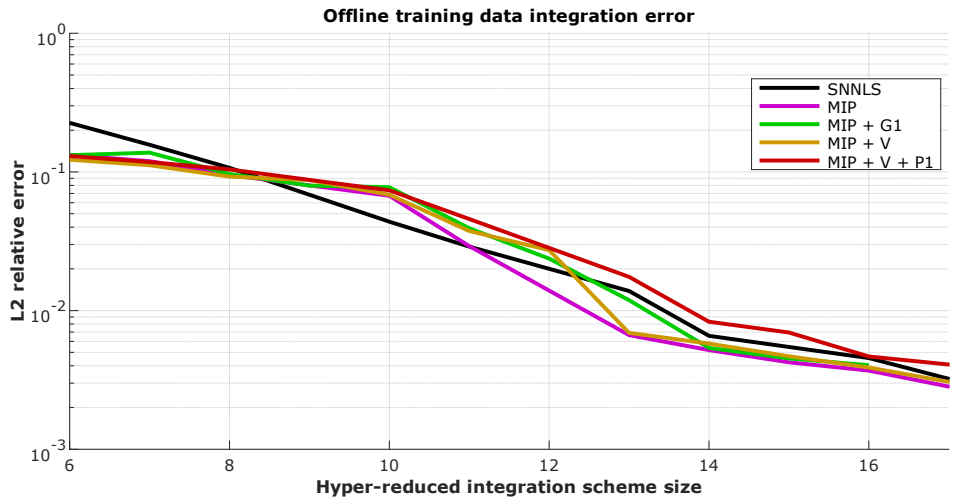


Figure 5.4: Offline training data relative L2 error (5.33) for SNNLS and proposed hyper-reduced integration schemes selecting from 6 to 17 elements.

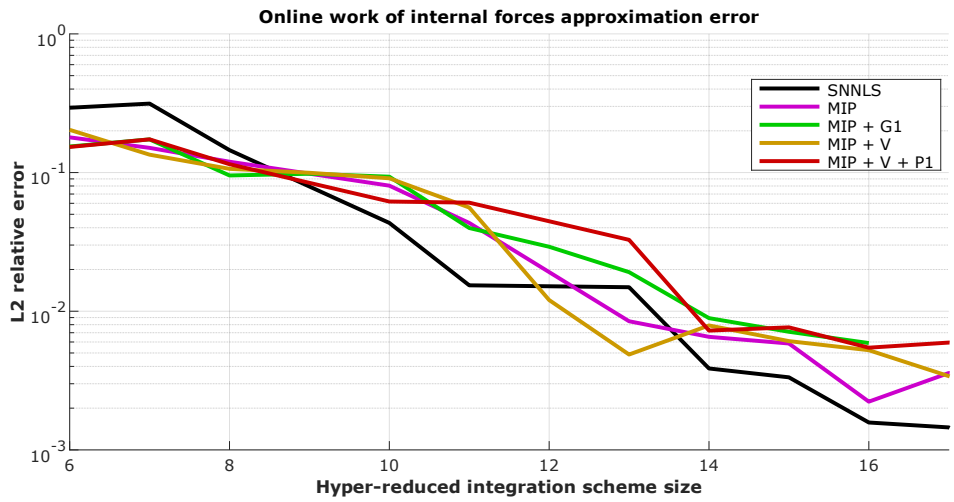


Figure 5.5: Online approximation error of the work of internal forces approximation error (5.35) for SNNLS and proposed methods hyper-reduced integration schemes sizes ranging from 6 to 17.

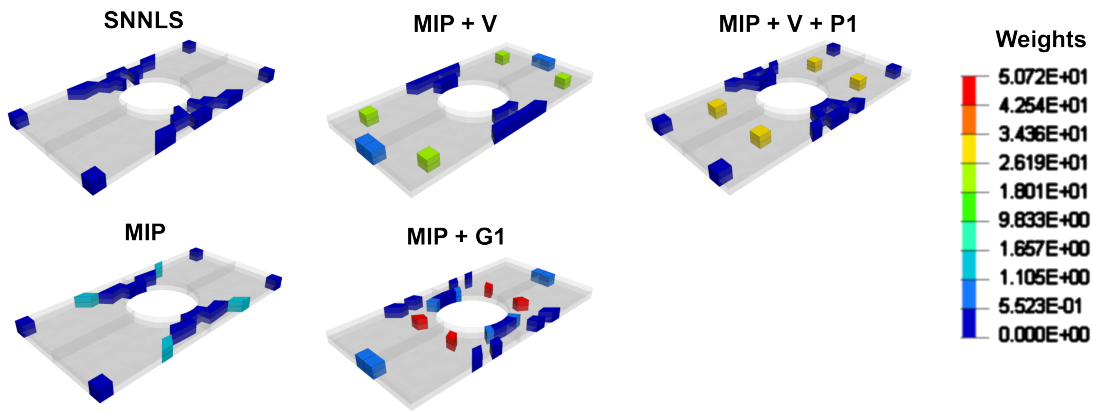


Figure 5.6: Element selection for the hyper-reduced integration schemes of size seven computed with the different methods.

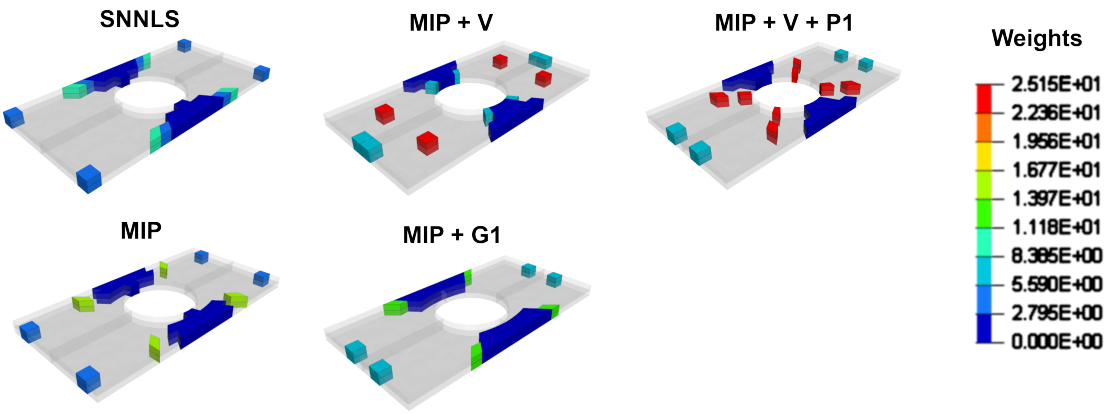


Figure 5.7: Element selection for the hyper-reduced integration schemes of size ten computed with the different methods.

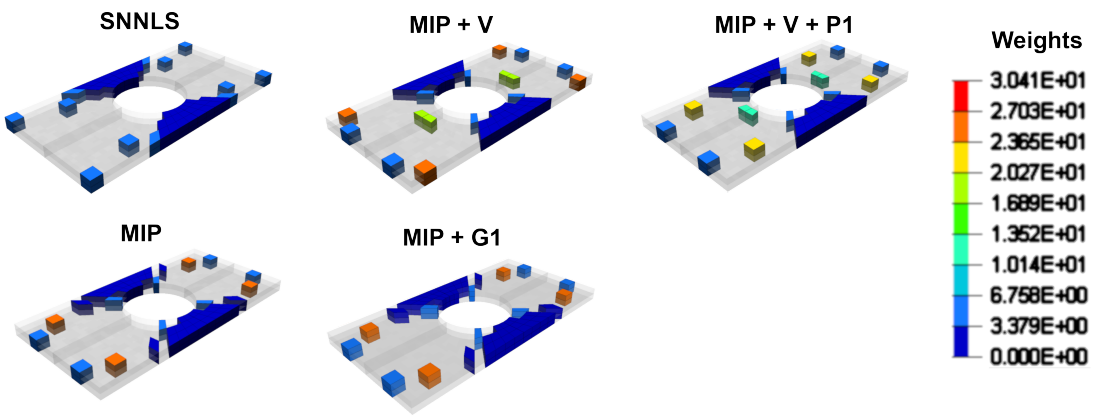
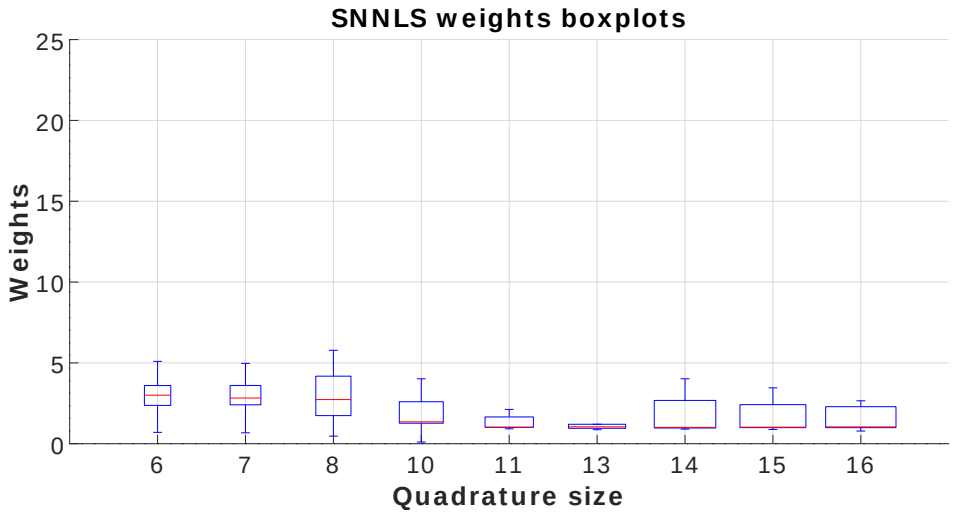
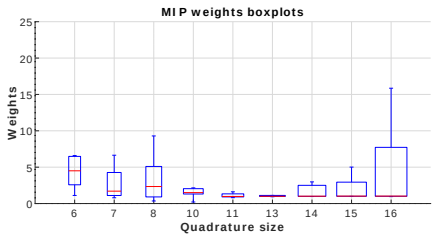


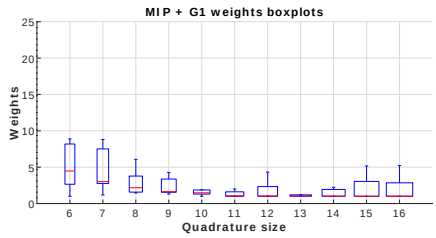
Figure 5.8: Element selection for the hyper-reduced integration schemes of size fourteen computed with the different methods.



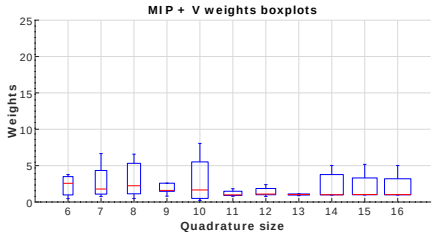
(a) SNNLS



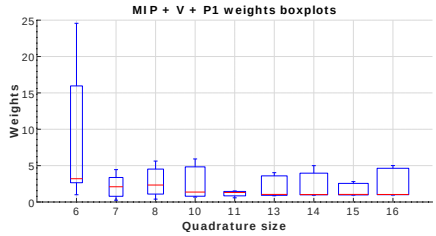
(b) MIP



(c) MIP + G1



(d) MIP + V



(e) MIP + V + P1

Figure 5.9: Hyper-reduction weights boxplots for consistency constraints formulations in comparison with the SNNLS and MIP methods.

Figure 5.4 shows the error on unassembled internal forces snapshots training data

$$\epsilon_{hr} = \frac{\|[G]\{\zeta\} - \{b\}\|_2}{\|\{b\}\|_2}. \tag{5.33}$$

Consistency constraints are not appended to $[G]$ when computing this error for respective hyper-reduced quadratures.

Figure 5.5 shows the approximation error of work of internal forces within the online reduction phase. The work of internal forces W_{int} is defined as the integral over the domain Ω of the tensor dot product between the stress $\underline{\underline{\varepsilon}}$ and the strain $\underline{\underline{\sigma}}$ tensors

$$W_{int}(t) = \int_{\Omega} \underline{\underline{\varepsilon}} : \underline{\underline{\sigma}} dV. \tag{5.34}$$

The approximation error of the work of internal forces ϵ_{int} is defined as

$$\epsilon_{int} = \frac{\int_0^T \mid W_{int}(t) - \tilde{W}_{int}(t) \mid dt}{\int_0^T W_{int}(t) dt}, \tag{5.35}$$

with W_{int}, and $\tilde{W}_{int}$, the work of internal forces of the reference and the hyper-reduced simulations, respectively.

Element subsets $\mathcal{H}$ selected by the different methods are compared for hyper-reduction schemes of size 7, 10, and 14 in Figures 5.6, 5.7, and 5.8. On those examples, and as mentioned above, results are extended to the full pierced plate through three planar symmetries. This choice allows for a better comparison of element selection as, on a full model, hyper-reduction may indiscriminately select among symmetric elements. Consequently, each selected element is represented eight times in the symmetric parts of the model.

Hyper-reduced integration weights obtained with the different formulations are represented in boxplots in Figure 5.9.

Figure 5.4 assesses the performances of the greedy SNNLS algorithm in the offline training phase. Among the hyper-reduced quadratures of size seven, the SNNLS algorithm solution offers the less accurate approximation in the offline phase. However, for larger hyper-reduced quadratures, proposed MIP formulations do not always offer better offline approximations than the SNNLS. This is in part due to the incapacity to express the constraints on training data integration in L2 norm in the proposed linear methods, thus, they don't share the same admissible space with the SNNLS. Moreover, the MIP formulations do not minimize the offline training data approximation error but only keeps it under a prescribed threshold. On the other hand, quadratures obtained with the MIP approach may also be unattainable with the SNNLS algorithm due to its greedy nature. Unassembled training data approximation quality is overall quite similar between the different approaches and adding consistency constraints does not deteriorate the training data approximation at a given quadrature size.

Online work of internal forces approximation presented in Figure 5.5 showing a good overall performances of the SNNLS algorithm. Even though it offers the worst online performances when less than seven elements are selected, it outperforms the

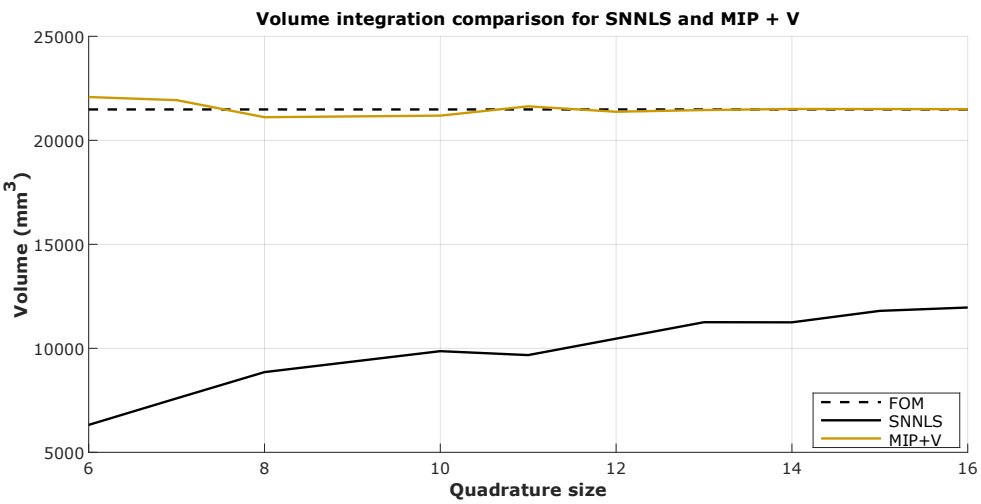


Figure 5.10: Volume integration for quadrature originating from SNNLS and MIP+V

proposed methods for larger quadratures size. Adding consistency constraints seems to deteriorate online performances of larger integration schemes. A good correlation between offline and online performances is observed.

SNNLS and MIP select elements in a similar pattern, as shown in Figures 5.6, 5.7, and 5.8. Most selected elements are located in the necking zone near the hole and few elements, with larger integration weights, summarize internal forces behavior on the outer parts of the model. Element selection at size 7, presented in Figure 5.6, is quite similar for the SNNLS and MIP methods but tends to diverge as more elements are selected. As a matter of fact, the SNNLS greedy procedure allows the deselection of one element in subsequent enrichment only if the associated weight is set to zero when computing the least feasible step (lines 15, 16 and 17 of SNNLS Algorithm 2). Thus, on the three figures, elements selected by the SNNLS at a given quadrature size are still selected in larger integration schemes, which is not the case for the proposed methods.

Boxplots in Figure 5.9 show that the proposed method including consistency conditions on polynomial integration suffer from weight overfitting as it consistently yield most significant integration weights. On this example, adding contraints on volume integration prevents overfitting, as observed comparing the MIP quadrature of size 16 with the MIP+V quadrature of the same size. On the other hand, weights computed with the SNNLS procedure are uniform.

The volume integration is tested for quadratures originating from the SNNLS and MIP + V on Figure 5.10. In this figure, volume obtained by integrating with different quadratures is averaged over all $n_s = 1148$ training times. Quadratures computed with the SNNLS do not preserve the volume while the proposed MIP+V formulation does independently of the number of elements. Results show that it is possible to add constraints to the hyper-reduced quadrature without deteriorating offline and online performances.

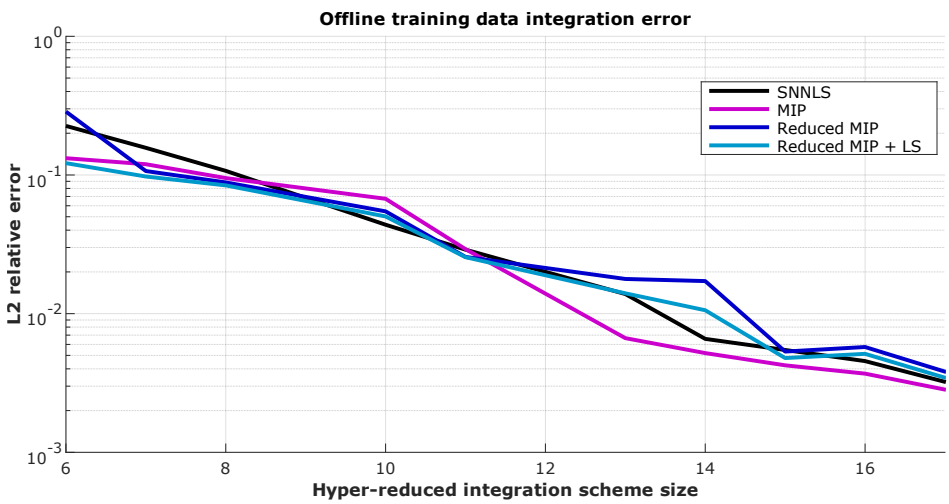


Figure 5.11: Offline training data relative L2 error (5.33) of SNNLS and proposed hyper-reduced methods. Hyper-reduced integration schemes sizes range from 6 to 17.

5.3.2 Constraints reduction

In this section, results are presented for the two following formulations of the hyper-reduction problem without consistency condition:

- Reduced MIP optimization problem (5.31);
- Reduced MIP + LS, optimization problem (5.31), the element selection $\{\xi\}$ is kept but weights are recomputed by a single pass of the SNNLS algorithm (least squares problem on the full matrix $[G]$ and least feasible step to ensure weight positivity).

Figure 5.11 plots the offline unassembled internal forces data approximation errors for hyper-reduced quadratures size ranging from 6 to 17 while associated online work of reconstructed internal forces is presented in Figure 5.12. Selected elements are illustrated in Figures 5.13, 5.14 and 5.15 for integration schemes of size seven, ten and fourteen. Weights are represented in boxplots in Figure 5.16.

While offline performances, plotted in Figure 5.11, are very similar between the different methods, reducing constraints has improved online performances with respect to the results obtained in the previous section where consistency conditions are used. Online results regarding Reduced MIP and the Reduced MIP + LS show that being accurate over the training data does not necessarily imply good online performances. While the Reduced MIP + LS always approximates the training data better than the Reduced MIP, since the weights have been optimized on the non-reduced training data in the first method, it does not yield better online performances. In Figure 5.11 the Reduced MIP is less accurate on the training data than the Reduced MIP + LS as the element selection is kept in the latter, but weights are re-optimized on the training data. Yet, Figure 5.12 shows that it does not imply better performances online. The version without weights re-optimization shows

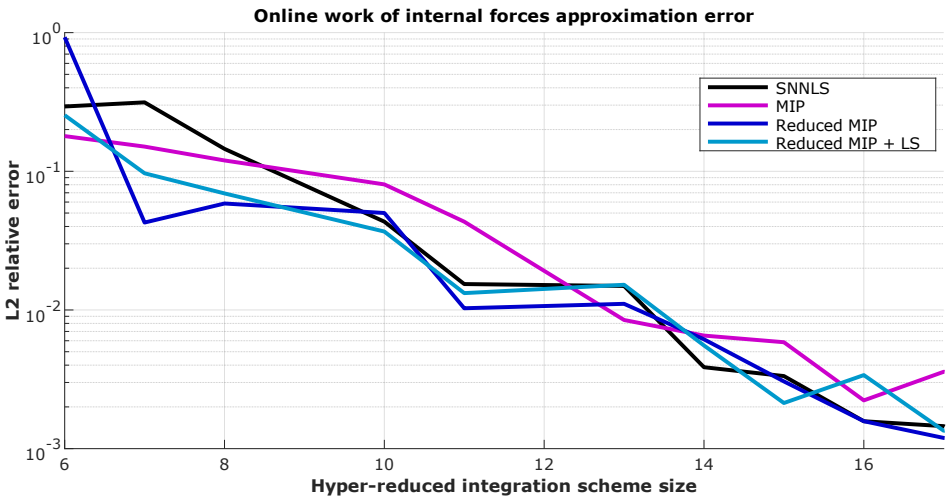


Figure 5.12: Online work of internal forces approximation error (5.35) of SNNLS and proposed hyper-reduced integration schemes. Hyper-reduced integration schemes sizes range from 6 to 17.

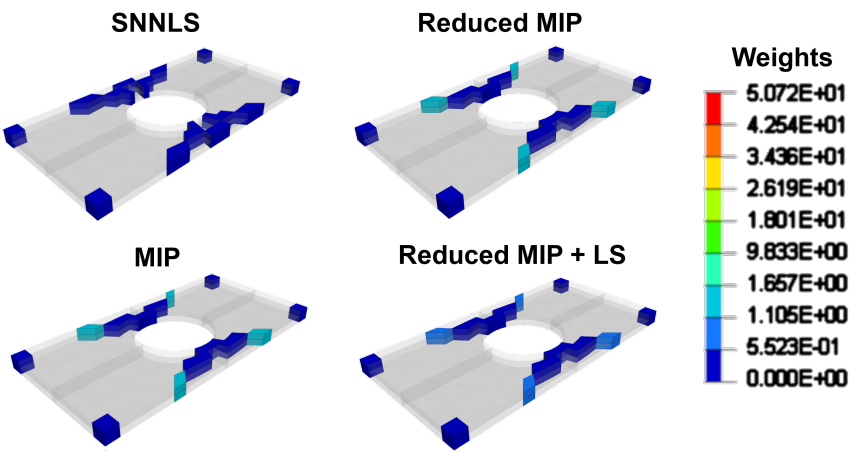


Figure 5.13: Element selection for the hyper-reduced integration schemes of size seven computed with the different methods.

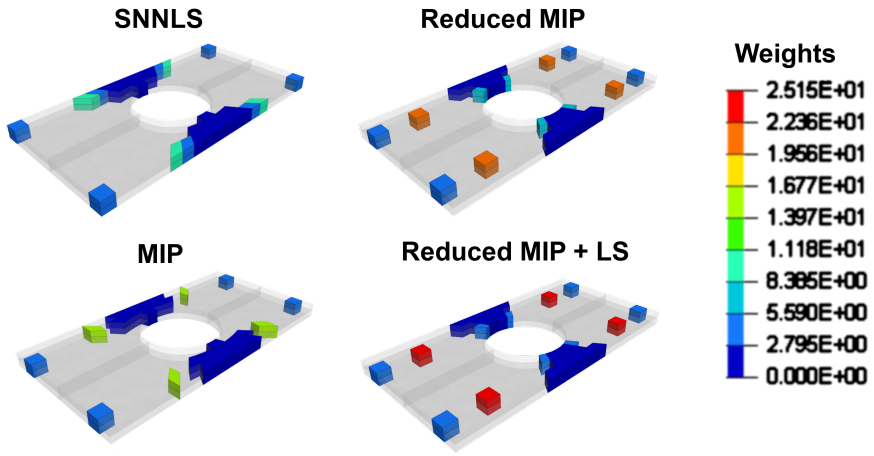


Figure 5.14: Element selection for the hyper-reduced integration schemes of size ten computed with the different methods.

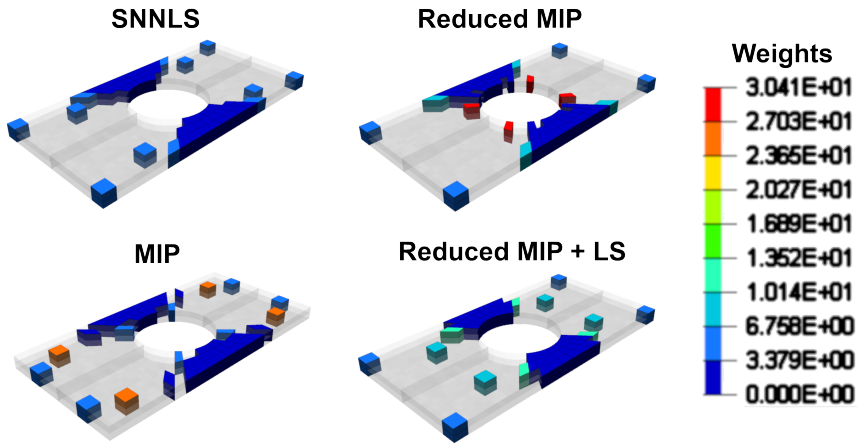
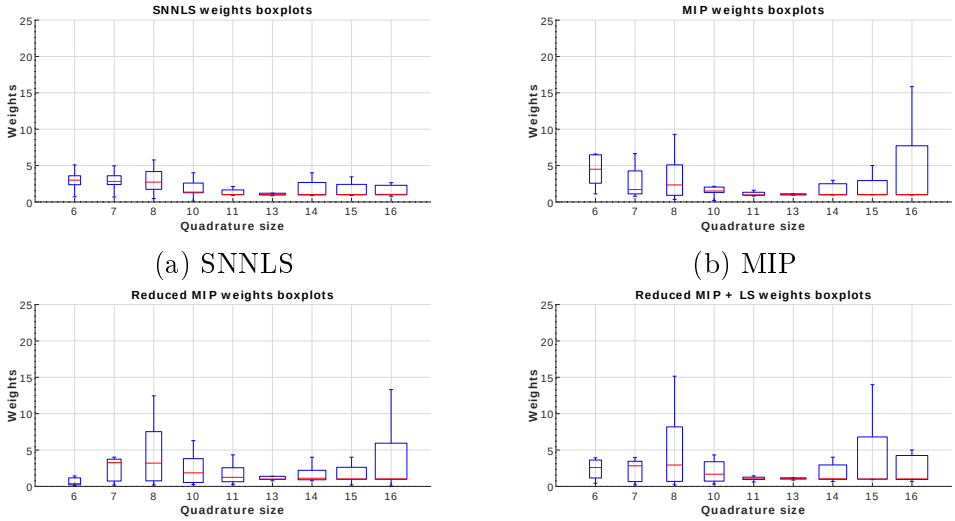


Figure 5.15: Element selection for the hyper-reduced integration schemes of size fourteen computed with the different methods.



(a) SNNLS (b) MIP

(c) Reduced MIP (d) Reduced MIP + LS

Figure 5.16: Hyper-reduction weights boxplots for constraints reduction formulations in comparison with the SNNLS and MIP methods.

slightly better online performances for hyper-reduced quadratures of size 7, 8, 11, 12, 13, 16, and 17 than other methods.

Elements are selected in a similarly by the different approaches, as observed in Figures 5.13, 5.14, and 5.15 where most selected elements concentrate in the necking zone around the hole. However, different quadratures tend to diverge as more elements are selected. By contrast with the proposed method, the SNNLS does not deselect elements when quadrature size increases. In Figure 5.15, Reduced MIP and the Reduced MIP + LS did not select the same elements. The Reduced MIP + LS solution corresponds, in fact, to the Reduced MIP for 15 elements in which one element has been unselected by the single pass in the SNNLS algorithm.

Weights comparison in Figure 5.16 shows overfitting issues of both reduced approaches in some cases. Overfitting issue is most pronounced in quadratures of small and large sizes, while quadratures of size eleven and thirteen exhibit uniform weight repartition regardless of the method. Neither constraints reduction nor adding weights re-optimization seem to circumvent this issue. Overall, the SNNLS algorithm still provides the most homogeneous weights repartition.

5.3.3 Discussion

The SNNLS algorithm shows out to be an excellent heuristic to solve the optimization problem arising in the ECSW training phase. It provides good performances on the pierced plate example both in offline and online phases compared with the proposed approach.

Adding consistency conditions to the MIP formulation seems to deteriorate online performances. On the contrary, constraints reduction improved online performances.

Proposed approaches suffer from overfitting compared with SNNLS. Further testing should include the effect of the maximal weight value ζ_{max}, introduced in section 5.2, on both offline and online performances.

Computation times associated with the training phase of the proposed MIP approaches are, on the small pierced plate example, close to that of the SNNLS, even for a large number of snapshots. However, MIP approaches quickly become computationally unaffordable for increasing numbers of elements, preventing the treatment of a more significant test case. At the moment, small improvements of the online performances observed in some cases do not counterbalance the growth in training complexity.

5.4 Conclusion

Compared with the proposed MIP approach, SNNLS is shown to be a good heuristic for solving the optimization problem arising in the ECSW in both the offline training data set reconstruction and online work of internal forces prediction. Tests on a pierced plate in traction model show similar performances between the suboptimal hyper-reduced quadratures obtained with the SNNLS algorithm and quadratures obtained with the proposed MIP method, both in offline and online phases. Variations of the proposed method, incorporating consistency conditions, allow for volume preservation and polynomial integration as well as constraints reduction. However, the computational overhead of the proposed approach with respect to the SNNLS is not compensated by the online performance gains.

Alternative formulations of the ECSW optimization problem still merit research as the greedy nature of the SNNLS algorithm renders computations prohibitively long on large models. The introduction of Boolean selection variable enables new constraints on the hyper-reduced integration scheme size. Such a feature may leverage the explosion in complexity from which suffers the SNNLS greedy algorithm when dealing with large models but induces more developments requiring further research. Moreover, if a constraint on the number of selected elements is used, objective functions may be changed to the training data set reconstruction error.

Chapter 6

Implementation

In the scope of this work, the Hyper-Reduced Proper Orthogonal Decomposition (HRPOD) is implementated in the state-of-the-art finite element explicit solver Altair Radioss [3]. Altair Engineering France provided full access to the solver source code creating a separate branch of the version 2017.1 for the development of the method for research purposes.

Section 6.1 provides general recommendations in implementing the method from the programming point of view. Current bottlenecks to HRPOD applications to industrial FE models, together with outlooks, are presented in Section 6.2.

6.1 Implementation

Developments required in the HRPOD implementation separate into two phases represented in Figure 6.1: during the training phase, data are collected and used to compute a ROM used in the *online* reduction phase to speed-up computations.

The offline training phase decomposes further into the snapshots gathering and the snapshots post-processing steps.

First, solution and unassembled internal forces snapshots are gathered and stored in binaries. A buffer is used to minimize the access to binary files. An unassembled internal forces snapshot is represented by a matrix of size $N \times n_e$ in which each column is the contribution of the corresponding element to the global internal forces vectors (Section 2.1.5). As a consequence, before being projected on the RB, unassembled internal forces snapshots are highly sparse. The choice has been made to store those snapshots using the Compressed Sparse Row (CSR) sparse format to save memory. This choice is motivated by the facility to append further unassembled internal forces snapshots in lines. In the FE method, internal forces elemental contributions are computed in each element with respect to the local Degrees of Freedom (DoF) indexation and then added to the general internal forces vector indexing DoFs globally. Depending on the code structure, the inverse connectivity matrix is very likely to be used to reorganize unassembled internal forces snapshots.

The post-process consists of reading snapshots in binaries, computing the ROM, and writing it into binaries for *online* use. The ROM is given by a RB and associated hyper-reduced quadrature for the integration of internal forces. The RB is computed first as it is needed to train the hyper-reduction scheme. Translational and rotational DoF are separated in each solution snapshots and two different RBs

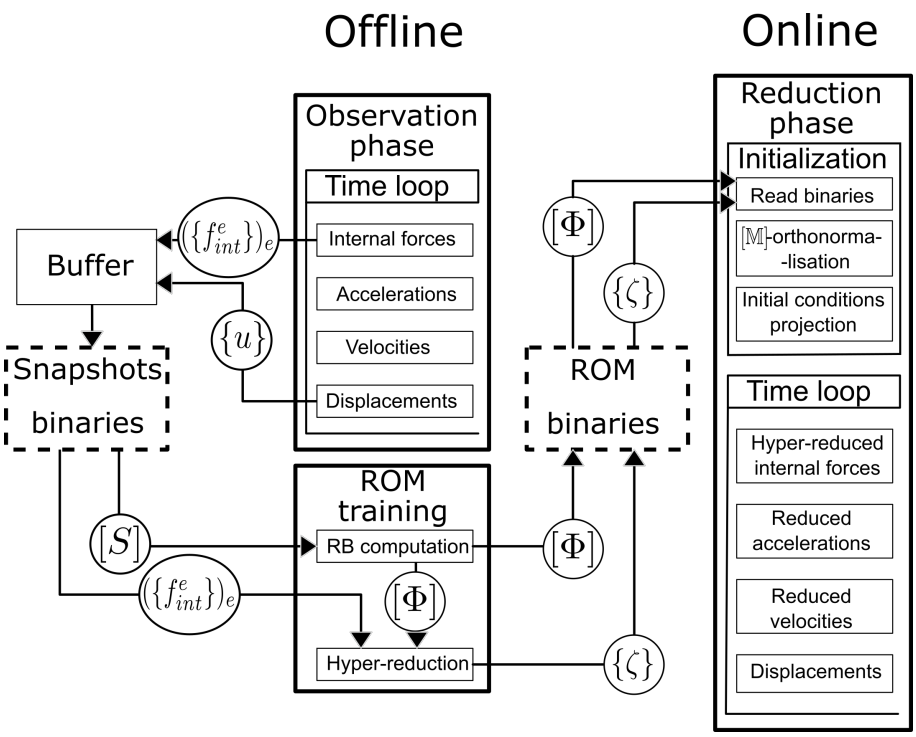


Figure 6.1: Flowchart of the HRPOD *offline* training and *online* reduction procedures.

are computed. They may be of different sizes and are orthonormalized in the *online* reduction phase with respect to the metric induced by the mass and inertia matrices. Once RBs computed, unassembled internal forces are projected on the reduced space and used to train hyper-reduction. This projection induces the multiplication of fully populated and sparse matrices. The snapshot post-process phase ends by writing the RBs and associated hyper-reduced quadrature into binaries.

The *online* reduction phase uses the ROM in a simulation to speed-up computations.

The reduced run is initialized by reading the ROM binaries, orthonormalizing the RBs to the metrics induced by the mass, for translational DoF, and inertia, for rotational DoF, matrices. The oblique projection, introduced at the end of section 2.2.3, is then used to project all initial conditions onto the reduced space. The oblique projection matrix $[\Phi]^T[\mathbb{M}]$ is computed once and stored in memory for subsequent projections. Additional initializations regarding the data structure for parallelism and code optimization are performed here.

Modifications in the time loop include hyper-reduced internal forces, stable time-step, accelerations, and velocities computations. The internal forces routine is modified to only take into account elements selected in the hyper-reduced quadrature. Associated weights are applied to the assembly. The nodal time-step is computed after the internal forces by projecting the equivalent nodal stiffness onto the reduced space. Reduced accelerations are simply obtained by projecting internal forces on $[\Phi]^T$, according to the ROM semi-discretized equations (2.19) introduced in section 2.2.3. High dimensional accelerations are approximated using reduced accelerations only if required for animation, penalization, or any other feature to avoid

unnecessary computations. Similarly, unnecessary projections are avoided when computing velocities. When kinematic conditions are prescribed, they are projected onto the reduced space ahead of velocites computations. Finally, reduced velocities are obtained by integrating reduced accelerations and used to approximate high dimensional velocities.

Multiple features of the solver require special care in the *online* reduction phase. At first they are handled by computing full-scale approximations or projecting additional full-scale variables on the reduced space. However, it must be done only if necessary as it induces a computational overhead. In some cases, additional methods exist, such as hyper-reduction for nonlinear internal forces, to avoid extra computations. In other cases, computing the needed full-scale variables will not work has they are poorly approximated by the ROM, and alternatives need to be found as discussed in Section 6.2.2.

6.2 Industrial Processes

Several bottlenecks are still hindereing a robust integration of the HRPOD in industrial crashworthiness optimization.

The offline training phase computational cost increases exponentially with the model size. As shown in chapter 3, the incremental SVD equipped with the newly developed sequential error estimator alleviates this issue for the RB training. The algorithm is yet to adapt to the sparse criterion proposed in chapter 4. Most effort in the training phase concerns the hyper-reduced integration schemes' computation, particularly intensive due to the greedy nature of the SNNLS algorithm and the complexity of the associated optimization problem. Prior work on an alternative non-greedy method is presented in chapter 5.

Advanced features such as airbags, springs, weld spots, and adaptive mass scaling present in crash models require further developments. Moreover, optimizing the method to answer industrial standards regarding performance and parallelization incurs considerable developments in the source code.

A significant concern arising in the industrial use of the HRPOD in crashworthiness optimization is the relevance of training data to the targeted application.

The parametric variation between the *offline* and *online* phases has not been studied in this thesis. A reduced model trained on snapshots collected at given parameters values will deliver good *online* approximation as long as the deformation is comparable to the one observed in the training data. As a consequence, the performance of HRPOD in such a study is problem-dependent. Thus, we choose to leave this matter to the industrial process. Nevertheless, two outlooks are worth investigating to overcome the training data relevancy bottleneck: substructuring and online adaptivity.

Substructuring aims to extend HRPOD to multi-domain, allowing to use MOR on standard parts of the structure intended to behave similarly at different parameters values while using the FOM on the remainder of the structure. This outlook and associated required developments are presented in section 6.2.2.

The online adaptivity concept is to develop auto-correcting ROM in the *online* reduction phase when the RB does not represent the model solution at a given time

step anymore. Section 6.2.3 is devoted to the different leads in the development of such a feature.

6.2.1 ROM for full-scale crash models

Industrial FE crash models make use of a lot of different options such as contact interfaces, airbags, weld spots, and adaptive mass scaling as well as invoking a wide variety of different element formulations. Incorporating these options in the HRPOD framework requires considerable software developments. Moreover, applying the HRPOD approach to a large-scale model requires special care concerning the training data manipulation and source code parallelization.

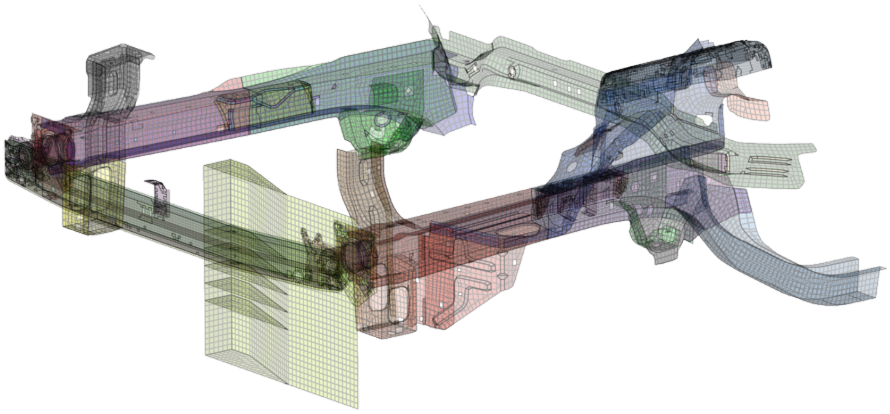


Figure 6.2: Bumper plus frame rail frontal impact on a rigid barrier FE model (Courtesy of the PSA Group)

In our work, we tried to apply the HRPOD to a full-scale assembly of bumper and frame rail model provided by the PSA group presented in Figure 6.2. This model is composed of 39865 nodes, $n_e = 37510$ shell elements with three different formulations and 396 spring elements yielding $N = 239190$ DoF. A rigid body with 1.5 tons of added mass takes account of the rest of the vehicle. The assembly impacts the rigid wall with initial velocity $v_0 = 35 \quad km.h^{-1}$. $T = 1.25 * 10^{-1}$ seconds are simulated, taking about 18450 seconds of computations for the full-scale simulation on a single CPU.

No speed-up is achieved on this example as hyper-reduction is not used for two reasons.

The hyper-reduction training phase is prohibitively expensive in computation time. For an inputted precision of one percent, the training of a hyper-reduced quadrature over 189 unassembled internal forces snapshots takes more than three days.

Moreover, when using the hyper-reduced quadrature in the *online* reduction phase, issues are encountered at weld spots where a precise computation of internal forces is needed.

The first $5 * 10^{-2}$ seconds of the simulation may be pretty well approximated by small RBs of size 27 and 100 for translational and rotational DoFs, respectively,

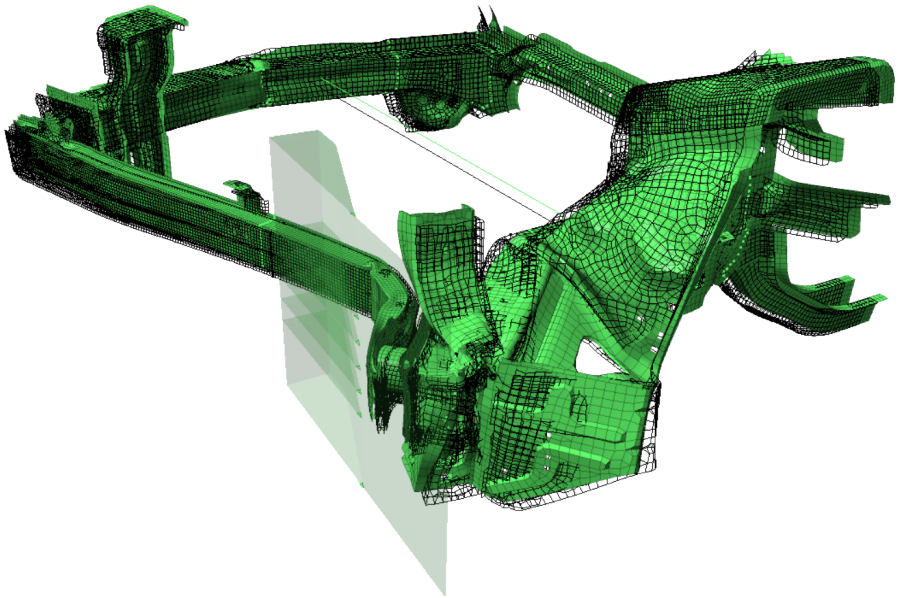


Figure 6.3: Bumper plus frame rail frontal impact on a rigid barrier ROM and FOM. The ROM is represented with solid green elements and the FOM with the black wireframe grid at $5 * 10^{-2}$ seconds. RBs of size 27 and 100 are used for translational and rotational DoF, respectively.

as shown in Figure 6.3. In this example, the RB was computed with the proposed incremental SVD presented in chapter 3. Figure 6.3 shows the reduced model approximation with solid green elements and the reference FOM in the black wireframe grid.

However, we currently fail to compute RBs of reasonable sizes to approximate the rest of the simulation. Using RBs of size 200 for both translational and rotational DoF, the first $5{*}10^{-2}$ seconds of the phenomenon are perferctly approximated, as shown in Figure 6.4. However, the approximation at the final simulation time $T = 1.25 * 10^{-1}$ seconds is quite bad, as seen in Figure 6.5. On both figures, the reduced model is represented with solid blue elements and the reference FOM with a black grid. Larger RBs result in prohibitive computation times, hardly compensated by hyper-reduction in the *online* reduction phase at the moment.

We are currently considering two non-exclusive strategies to overcome this limitation. Using different RBs for parts of the domain and eventually simulating the most complicated part of the domain with at full-scale is discussed in Section 6.2.2. Moreover, different parts of the domain may switch between the FOM and ROM at different moments of the simulation, as discussed in Section 6.2.3.

6.2.2 Substructuring

Applying reduction to a given part of the domain while simulating the rest of the domain with the FOM may help in various applications.

When studying the solution behavior under parametric variations, some parts of

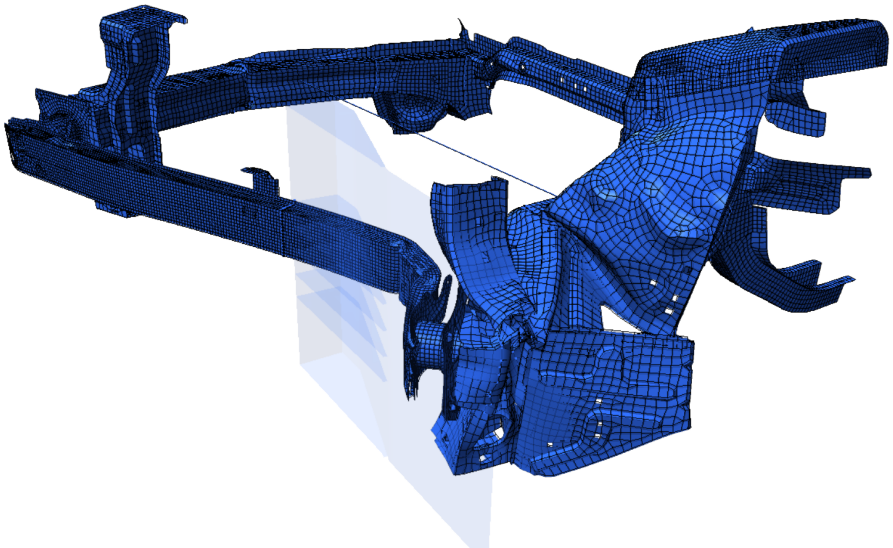


Figure 6.4: Bumper plus frame rail frontal impact on a rigid barrier ROM and FOM. The ROM is represented with solid green elements and the FOM with a black grid at $5 * 10^{-2}$ seconds. RBs of size 200 and 200 are used for both translational and rotational DoF.

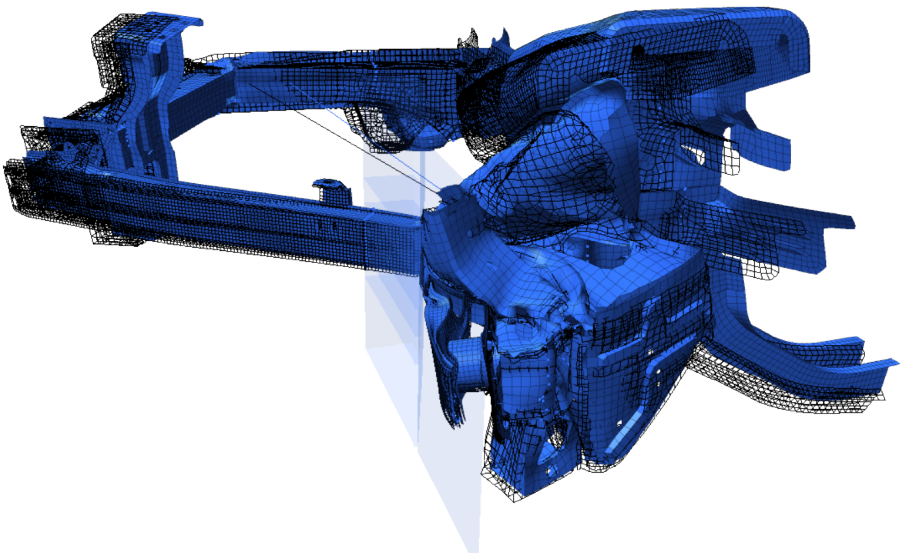


Figure 6.5: Bumper plus frame rail frontal impact on a rigid barrier ROM and FOM. The ROM is represented with solid green elements and the FOM with a black wireframe grid at final simulation time $T = 1.25 * 1-^{-1}$ seconds. RBs of size 200 and 200 are used for both translational and rotational DoF.

the model are intended to give similar results at different parameters value. Thus, it appears possible to train a ROM for these parts on a relevant data set. In this context, substructuring also enables the use of data collected from previous projects. New vehicles often carry over parts from previous designs on which extensive data are accessible from simulations or experiments. These data may be used to build a

ROM specific to the part and used in the FE model of a new vehicle. If training data from experimentations and FE simulations gathered from previous projects are to be used, it is necessary to transfer them to the new mesh.

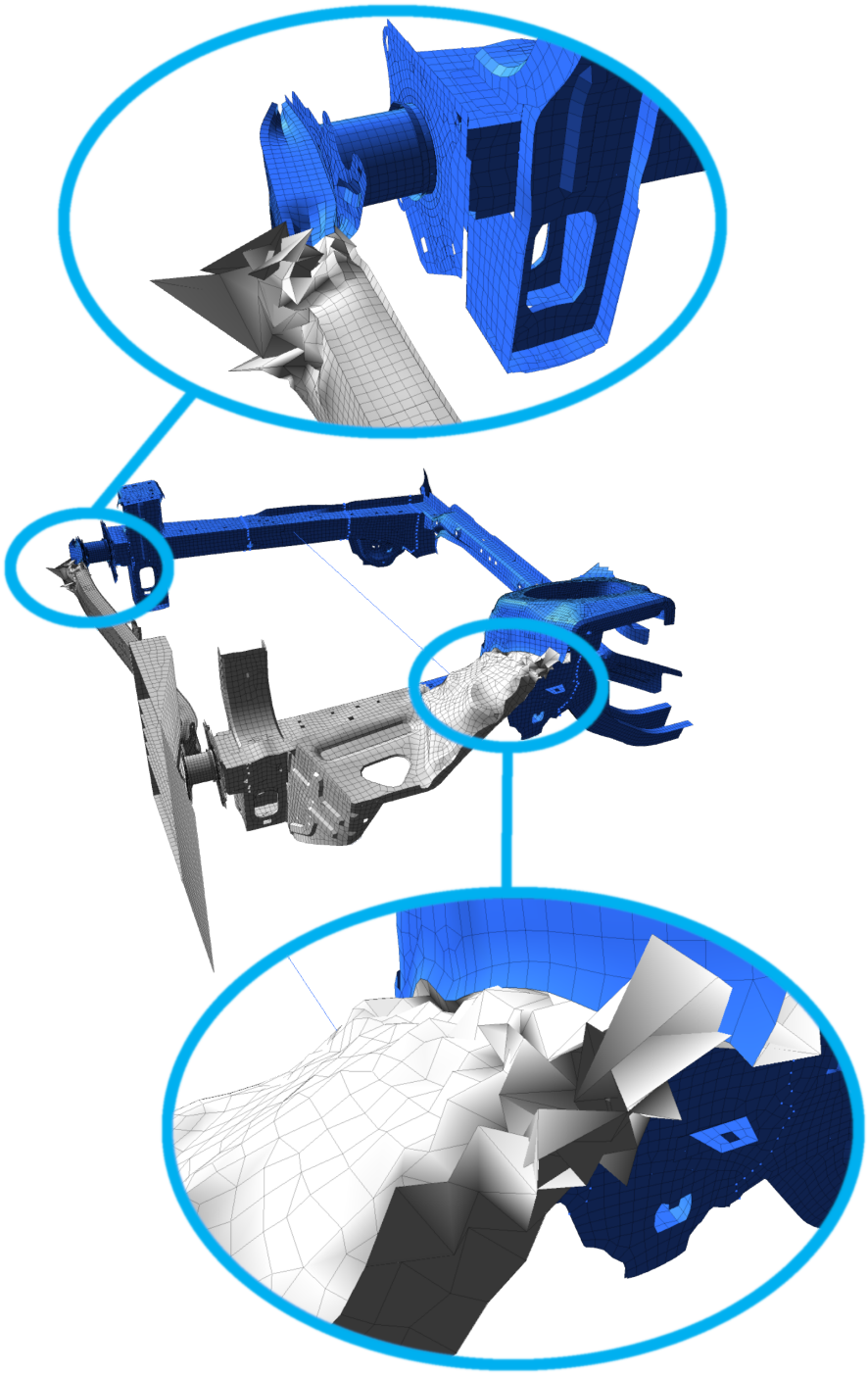


Figure 6.6: Bumper multi-domains simulation. The blue part is simulated using two RBs of size 100 for translational and rotational DoF, while the gray part is simulated at full-scale. Zooms on interfaces show instabilities.

We tried to apply the multi-domain approach to the bumper plus frame rail

model presented in Figure 6.2. The model splits into two parts. The front of the bumper is simulated at full-scale, while the ROM is used to simulate the back of the bumper. The idea is to reduce the model as much as possible and simulate at full-scale areas where the solution is very likely to change with the model parameters.

We have successfully applied this approach to models involving solid elements only. However, we have encountered new difficulties with shells as instabilities quickly appeared in the full-scale domain for elements close to the reduced domain, preventing the simulation from terminating correctly. This issue is illustrated in Figure 6.6, with the full-scale domain presented in gray and the reduced one in blue. Zooms are performed on interfaces between the two domains where instabilities appear.

These results show, that the interface between the reduced and unreduced domains needs to be reformulated. For shell elements, the coupling between domains makes use of accelerations, which are poorly approximated by the ROM. As a matter of fact, the funding hypothesis of PROM, that the solution evolves in a manifold embedded in a vectorial space of much lower dimension than the FOM, is correct for displacements but not necessarily for accelerations. An adapted formulation of the interface requires further theoretical developments.

6.2.3 Online adaptivity

A promising lead to the robust application of model order reduction in DoE exploration lies in online adaptivity. As discussed earlier, the HRPOD performance depends on the RB's ability to approximate the solution. Being able to detect *online* when the approximation deteriorates in a given region of the domain at given simulation time and to update the ROM *online* at minimal computational cost would be a significant step toward robust applications of the method to industrial DoE exploration.

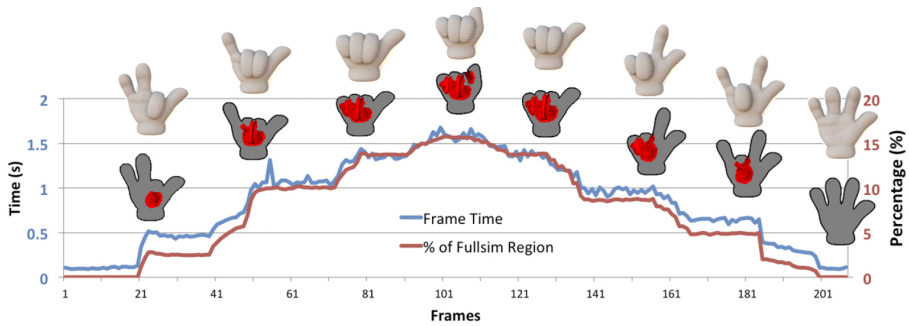


Figure 6.7: Simulation time per frame for a fist clenching and unclenching. The fingers clench in a different sequence from which they unclench, so novel collision configurations unseen during training are encountered. The FOM regions are drawn in red and the ROM regions in gray [76].

This concept originates from HRPOD applications to computer graphics ([5, 78, 76, 12]). In [76], contacts not present in the training data set appear in the reduction phase and a region of full-scale simulation is decided *online* in a certain radius around the contact interface, as represented in Figure 6.7. Doing so, avoids significant error due to unexpected configurations, while accelerating the simulation in

regions in which the ROM is still valid. Contact is used as a trigger, referred to as *oracle*, for online adaptivity, consisting in defining a region of the domain in which full-scale computation is required.

The *oracle* identifies when and where the ROM is invalid while avoiding full-scale computations and unnecessary updates. Different *oracles* and online adaptations may be considered in applications to explicit nonlinear structural dynamics. Contact is not the only event which may results in ROM invalidity and other *oracles*, based on deformations or total energy loss for example, need to be developed.

In automotive design application, it is not clear how to restrain the ROM to a region of the domain. Using condensation, as is the case in [76], is likely to be computationally unaffordable, requiring the development of alternative methods. Moreover, online adaptivity may also consists in enriching the ROM. While enriching the RB is straightforward with the incremental SVD presented in Chapter 3, updating the hyper-reduced integration scheme incrementally remains an open question.

Chapter 7

Conclusion

A significant literature, reviewed in the present work, allows to consider applications of intrusive reduced order industrial explicit solver to automotive vehicle crashworthiness optimization. The project has been oriented toward projection-based reduced order modelling methods conjointly with hyper-reduction of nonlinear internal forces. Based on an implementation of the HRPOD in the legacy solver Altair Radioss, different bottlenecks to the robust application of the method in industrial processes have been identified and addressed in original contributions mainly focusing on the following two aspects of the hyper-reduced projected reduced order model training phase. The *offline* training phase computational cost is to be controlled to achieve global speed-up. *Online* validation is used to compare *offline* training methods and provide a better correlation between both training data approximation and effective approximation quality in the reduction phases.

The overall objective being computational savings, the computational cost of the *offline* training phase is to be addressed. More importantly, the singular value decomposition used in the computation of a reduced basis may reveal to be infeasible when applied to large-scale training data. Incremental alternatives for the decomposition of large-scale data have been developed for video streaming and recently brought to the model order reduction field. In this work, incremental reduced basis method is further adapted to projected reduced order modeling and an error estimator is proposed for 'on-the-fly' truncation and snapshots selection. The resulting algorithm is nearly optimal in approximating the data, and more "user-friendly" than the reference version. When building low-rank approximations, the algorithm outperforms the traditional single-pass SVD in terms of computation time by several orders of magnitude. *Online* testing on a Taylor beam impact example showed enhanced performances of reduced basis computed with the proposed incremental SVD. In the future, such incremental approach may be used to compute and correct reduced order models in the *online* reduction phase.

The ROM's validation in the *online* reduction phase. In nonlinear structural dynamics, the error criterion used in the POD *offline* training phase is shown to be insufficient to assess the *online* approximation quality when the training data is relevant to the targeted application. As a matter of fact, the averaging nature of this criterion may result in bad representation of some snapshots, even when a very low threshold is imposed on the ROM's *offline* training data approximation error, resulting in considerable approximation error in the *online* reduction phase. A sparse

criterion is proposed in this thesis to circumvent this issue. By imposing a threshold on the representation of all snapshots and using combinatorial optimization to build the smallest subset of RB functions, this error criterion shows on different examples a better agreement between the approximation errors observed both in the *offline* training and in the *online* reduction phases.

The optimization problem of the ECSW hyper-reduction training phase is solved in a greedy manner by the SNNLS algorithm. This suboptimal procedure turns out to be prohibitively computationally expensive in applications to large training data sets. An alternative formulation of hyper-reduction optimization problem is proposed and solved with Mixed-Integer Programming on an academic example. While the SNNLS algorithm shows to be a good heuristic, the developed MIP formulation may be used to further develop heuristic alternatives. The present work further investigates the impact of reducing or adding constraints, such as polynomial integration and volume conservation, on both *offline* and *online* performances.

Applications to full-scale FE models provided by the industrial partner PSA group revealed some important bottlenecks of the HRPOD regarding the *offline* training phase feasibility on large models. Training an hyper-reduced integration scheme turns out to be prohibitively expensive due to the greedy nature of the SNNLS algorithm. Some features used in industrial vehicle crash simulations, such as weld spots, requires full-scale computations and need to be correctly handled in the *online* reduction phase. Additionally, regions of the domain undergoing complex deformations are hardly approximated with RB of small size. These bottlenecks motivated outlooks for this project.

Further research topics, concerning online adaptivity and substructuring are identified. The structure may undergo unexpected deformations in the *online* reduction phase, being able to detect such event and adapt the ROM "on the fly" would be a major advantage of intrusive reduction methods over non-intrusive ones. Moreover, having different ROM for different regions of the model and while simulating other regions at full-scale may help industrial applications of the method as some regions of the model may be too challenging to reduce or because training data is not available on the whole domain.

The developments are available in a dedicated research branch of Altair Radioss code.

Appendix A

Notations

In the manuscript, curly brackets designate vectors and square brackets designate matrices. Following notations are sorted out in alphabetical order.

$\sharp(\bullet)$ - cardinality of a set
$\{0_N\}$ - column vector of size N with all coefficients equal to zero
$\{1_N\}$ - column vector of size N with all coefficients equal to one
$\{\alpha\}$ - reduced unknown displacement, $\alpha \in \mathbb{R}^k$
B - subset of k indices of left singular vectors forming the RB
b - plasticity hardening coefficient
$\{b\}$ - assembled training data
$(\beta_i(t_j))_{(i,j)}$ - projection of singular vector $\{\phi_i\}$ on snapshot $\{u(t_j)\}$, $(i,j) \in [\![1,m]\!] \times [\![1,n_s]\!]$
$[C]$ - linear constraints matrix
$[C_r]$ - reduced linear constraints matrix
$(c_i)_i$ - training configurations, $i \in [\![1,n_s]\!]$
E - Young's modulus
$\underline{\underline{\varepsilon}}$ - strain tensor
ε_p - equivalent plastic strain
ϵ_{est} - error estimator in the proposed incremental SVD algorithm
ϵ_{hr} - training data approximation error
ϵ_{in} - part of discarded snapshots represented by $[\Phi^k]$
ϵ_{int} - work of internal forces approximation error
ϵ_{orth} - tolerance for reorthonormalization in the incremental SVD
ϵ_{out} - part of discarded snapshots non-represented by $[\Phi^k]$
ϵ_{rb} - user-specified threshold on the basis function selection criterion
ϵ_{sp} - sparse criterion approximation error
ϵ_{sv} - sequential criterion approximation error
ϵ_{sv}^k - singular value truncation error for a reduced basis of size k
ϵ_{svd} - tolerance for snapshot selection in the state-of-the-art incremental SVD
$\{f_{ext}\}$ - space discretized external forces, $\{f_{ext}\} \in \mathbb{R}^N$

$\{\tilde{f}_{ext}\}$ - reduced external forces, $\tilde{f}_{ext} = [\Phi^{(B)}]^T f_{ext} \in \mathbb{R}^k$
$\{f_{int}\}$ - space discretized internal forces, $\{f_{int}\} \in \mathbb{R}^N$
$\{f^e_{int}\}$ - internal forces in element e
$\{\tilde{f}_{int}\}$ - reduced internal forces, $\tilde{f}_{int} = [\Phi^{(B)}]^T f_{int} \in \mathbb{R}^k$
$\|[\bullet]\|_F$ - Frobenius norm of a matrix, $\|[\bullet]\|_F = (\sum_{i,j} [\bullet]^2_{i,j})^{\frac{1}{2}}$
$[G]$ - matrix of unassembled training data
$[G_f]$ - matrix of unassembled internal forces at all training times
$[G_v]$ - matrix of elements volumes at all training times
$[G_p]$ - Constraint matrix on polynomial integration
$\{g^{(e)}\}$ - center of element e, $\{g^{(e)}\} \in \mathbb{R}^3$
$(\gamma_i(t_j))_{(i,j)}$ - relative contribution of singular vector $\{\phi_i\}$ to snapshot $\{u(t_j)\}$
$\mathcal{H}$ - subset of elements in the hyper-reduced integration scheme, $\mathcal{H} \subset [\![1, n_e]\!]$
$[\mathbb{K}]$ - tangent stiffness matrix, $[\mathbb{K}] \in \mathbb{R}^{N \times N}$
k - Reduced basis size
k_{max} - Maximum size for the reduced basis in the state-of-the-art incremental SVD
$[\mathbb{M}]$ - symmetric positive-definite mass matrix, $[\mathbb{M}] \in \mathbb{R}^{N \times N}$
$[\tilde{\mathbb{M}}]$ - reduced mass matrix, $[\tilde{\mathbb{M}}] = [\Phi^{(B)}]^T [\mathbb{M}] [\Phi^{(B)}] \in \mathbb{R}^{k \times k}$
m - number of singular vectors in the training data POD decomposition, $m = min(N, n_s)$
N - number of degrees of freedom in the FE space discretization
n - plasticity hardening exponent
n_e - number of elements in the model
n_p - number of parameters of the FE model
n_s - number of snapshots in the training data set
ν - Poisson's ratio
Ω - considered spatial domain, $\Omega \subset \mathbb{R}^3$
ω_e - volume of element e
$(p_i)_i$ - parameters of the FE model, $i \in [\![1, n_p]\!]$
$[\Phi]$ - left singular vectors of $[S]$, $[\Phi] = [\phi_1, \ldots, \phi_{n_s}] \in \mathbb{R}^{N \times m}$
$[\Phi^{(B)}]$ - reduced basis formed by the left singular vectors selected according to the subset of indices B, $[\Phi^{(B)}] = [\{\phi_{B(1)}\}, \ldots, \{\phi_{B(k)}\}] \in \mathbb{R}^{N \times k}$
$[\Phi^k]$ - reduced basis of size k
$[\Psi]$ - right singular vectors of $[S]$, $[\Psi] = [\psi_1, \ldots, \psi_m] \in \mathbb{R}^{n_s \times m}$
$\mathbb{R}_{\geq 0}$ - set of non-negative real numbers
ρ - density
$[S]$ - training data set, $[S] = [\{u(t_1)\}, \ldots, \{u(t_{n_s})\}] \in \mathbb{R}^{N \times n_s}$
$\{s\}$ - singular values of $[S]$ in decreasing order, $\{s\} = (s_1, \ldots, s_m)^T \in \mathbb{R}^m$

$\underline{\underline{\sigma}}$ - stress tensor
σ_{eq} - equivalent stress
σ_Y - plastic yield stress
t - time, $t \in [0, T]$
T - final simulation time
$\{t_i\}$ - training times, $i \in [\![1, n_s]\!]$
$u(x, t)$ - the continuous unknown displacement field
$\{u\}$ - space discretized unknown displacement field, $\{u\} \in \mathbb{R}^N$
$\{\tilde{u}\}$ - displacement field approximation in the reduced model, $\tilde{u} = [\Phi^k]\alpha \in \mathbb{R}^N$
$\{\hat{u}\}$ - displacement projection on the reduced basis, $\hat{u} = [\Phi^{(B)}][\Phi^{(B)}]^T u \in \mathbb{R}^N$
V_{imp} - imposed velocity
V_0 - initial velocity
W_{int} - work of internal forces
x - point in Ω
$\{\xi\}$ - boolean vector selecting the subset of RB functions
$\{\zeta\}$ - weights of the hyper-reduced integration scheme
ζ_{max} - maximal weights value

Appendix B

Acronyms

ADS	Alternative Directions Scheme
APHR	A Priori Hyper-Reduction
APOD	Adaptive Proper Orthogonal Decomposition
BPOD	Balanced Proper Orthogonal Decomposition
CAD	Computer-Aided Design
CFL	Courant-Friedrichs-Lewy
CPU	Central Processing Unit
CSR	Compressed Sparse Row
DEIM	Discrete Empirical Interpolation Method
DoE	Design of Experiment
DoF	Degree of Freedom
ECM	Empirical Cubature Method
ECSW	Energy Conserving Sampling and Weighting
ESLM	Equivalent Static Load Method
FE	Finite Element
FEA	Finite Element Analysis
FEM	Finite Element Method
FOM	Full Order Model
HIC	Head Injury Criterion
HR	Hyper-Reduction
HRPOD	Hyper-Reduced Proper Orthogonal Decomposition
KPCA	Kernel Principal Component analysis
LASSO	least Absolute Shrinkage and Selection Operator
LATIN	LArge Time INcrement model
LS	Least Square
MGS	Modified Gram-Schmidt
MIP	Mixed-Integer Programming
MKL	Math Kernel Library
MOR	Model Order Reduction
NiPOD	Non-intrusive Proper Orthogonal Decomposition
NN	Neural Network
NP	Non-Polynomial

PCA	Principal Component Analysis
PDE	Parametric Differential Equation
PGD	Proper Generalized Decomposition
POD	Proper Orthogonal Decomposition
PROM	Projected Reduced Order Model
QoI	Quantity of Interest
RB	Reduced Basis
RBF	Radial Basis Function
ReCUR	Regressive CUR
ROM	Reduced Order Model
RS	Response Surface
SNNLS	Sparse Non-Negative Least Square
SOD	Smooth Orthogonal Decomposition
SVD	Singular Value Decomposition

Bibliography

[1] *Reference Manual for Intel® Math Kernel Library (Intel® MKL)*, 2019.

[2] Alessandro Alla and J Nathan Kutz. Nonlinear model order reduction via dynamic mode decomposition. *SIAM Journal on Scientific Computing*, 39(5):B778–B796, 2017.

[3] ALTAIR Engineering, 1820 E Big Beaver Rd, Troy, MI 48083, United States. *Altair Radioss 2017 Reference Guide*, 2017.

[4] A. Levey amd M. Lindenbaum. Sequential karhunen-loeve basis extraction and its application to images. *IEEE Transactions on Image Processing*, 90:1371–1374, 2000.

[5] Steven S An, Theodore Kim, and Doug L James. Optimizing cubature for efficient integration of subspace deformations. In *ACM transactions on graphics (TOG)*, volume 27, page 165. ACM, 2008.

[6] C Bach, D Ceglia, L Song, and F Duddeck. Randomized low-rank approximation methods for projection-based model order reduction of large nonlinear dynamical problems. *International Journal for Numerical Methods in Engineering*, 118(4):209–241, 2019.

[7] C Bach, L Song, T Erhart, and F Duddeck. Stability conditions for the explicit integration of projection based nonlinear reduced-order and hyper reduced structural mechanics finite element models. *arXiv preprint arXiv:1806.11404*, 2018.

[8] M. Barrault, Y. Maday, N.C. Nguyen, and A.T. Patera. An 'empirical interpolation' method: Application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique*, 339(9):667–672, 2004.

[9] G. Berkooz, P. Holmes, and John L. Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual Review fo Fluid Mechanics*, 25:539–575, 1993.

[10] M. Brand. Fast low-rank modifications of the thin singular value decomposition. *Linear Algebra Appl*, 415:20–30, 2006.

[11] Matthew Brand. Incremental singular value decomposition of uncertain data with missing values. In *European Conference on Computer Vision*, pages 707–720, Berlin/Heidelberg, Germany, 2002. Springer.

[12] Christopher Brandt, Elmar Eisemann, and Klaus Hildebrandt. Hyper-reduced projective dynamics. *ACM Transactions on Graphics (TOG)*, 37(4):80, 2018.

[13] P. Breitkopf, P. Phalippou, S. Bouabdallah, and P. Villon. Incremental POD and custom integration schemes for hyper-reduced automotive crash analysis. In 13th *World Congress on Computational Mechanics (WCCM XIII)*, New York, USA, July 2018.

[14] P. Breitkopf, A. Rassineux, J.-M. Savignat, and P. Villon. Integration constraint in diffuse element method. *Comput Meth Appl M*, 193(12-14):1203–1220, 2004.

[15] Fabien Casenave, Alexandre Ern, and Tony Lelièvre. A nonintrusive reduced basis method applied to aeroacoustic simulations. *Advances in Computational Mathematics*, 41(5):961–986, 2015.

[16] T. Chapman, P. Avery, P. Collins, and C. Farhat. Accelerated mesh sampling for the hyper reduction for nonlinear computational models. *Int. J. Numer. Meth. Engng*, 2017.

[17] S. Chaturantabut. *Nonlinear Model reduction via Discrete Empirical Interpolation.* PhD thesis, Rice University, 2011.

[18] Jiun-Shyan Chen, Michael Hillman, and Marcus Rüter. An arbitrary order variationally consistent integration for galerkin meshfree methods. *International Journal for Numerical Methods in Engineering*, 95(5):387–418, 2013.

[19] Jiun-Shyan Chen, Cheng-Tang Wu, Sangpil Yoon, and Yang You. A stabilized conforming nodal integration for galerkin mesh-free methods. *International journal for numerical methods in engineering*, 50(2):435–466, 2001.

[20] Kevin K Chen, Jonathan H Tu, and Clarence W Rowley. Variants of dynamic mode decomposition: boundary condition, koopman, and fourier analyses. *Journal of nonlinear science*, 22(6):887–915, 2012.

[21] Francisco Chinesta, Amine Ammar, Adrien Leygue, and Roland Keunings. An overview of the proper generalized decomposition with applications in computational rheology. *Journal of Non-Newtonian Fluid Mechanics*, 166(11):578–592, 2011.

[22] Francisco Chinesta, Roland Keunings, and Adrien Leygue. *The proper generalized decomposition for advanced numerical simulations: a primer.* Springer Science & Business Media, Berlin/Heidelberg, Germany, 2013.

[23] N. Cressie. Spatial prediction and ordinary kriging. *Mathematical Geology*, 20(4):405–421, 1988.

[24] R.J. Dedden. Model order reduction using the discrete empirical interpolation method. Master's thesis, Delft University of Technology, 2012.

[25] EUROPEAN NEW CAR ASSESSMENT PROGRAMME (Euro NCAP). *FILM & PHOTO PROTOCOL, Version 1.2*, december 2016.

[26] H. Fang. Global response approcimation with radial basis functions. *Journal of Engineering Optimization*, 38(4):407–424, 2006.

[27] Hiba Fareed and John R Singler. Error analysis of an incremental POD algorithm for PDE simulation data. *arXiv preprint arXiv:1803.06313*, 2018.

[28] Charbel Farhat, Philip Avery, Todd Chapman, and Julien Cortial. Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency. *International Journal for Numerical Methods in Engineering*, 98(9):625–662, 2014.

[29] Jules Fauque, Isabelle Ramière, and David Ryckelynck. Hybrid hyper-reduced modeling for contact mechanics problems. *International Journal for Numerical Methods in Engineering*, 115(1):117–139, 2018.

[30] G. Fougeron, G. Pierrot, and D. Aubry. Recovery of differentiation/integration compatibility of meshless operators via local adaptation of the point cloud in the context of nodal integration. In *Proceedings of the 7th European Congress on Computational Methods in Applied Sciences and Engineering*, 2016.

[31] Laurent Genest. *Optimisation de forme par gradient en dynamique rapide*. PhD thesis, Lyon, 2016.

[32] G. H. Golub and C. F. Van Loan. *Matrix Computations Third Edition*. The Johns Hopkins University Press, 1996.

[33] D. Gonzalez, J. V. Aguado, E. Cueto, E. Abisset-Chavanne, and F. Chinesta. kPCA-Based parametric solutions within the PGD framework. *Springer ARCH COMPUTAT METHODS ENG*, 2016.

[34] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

[35] Joaquin Alberto Hernandez, Manuel Alejandro Caicedo, and Alex Ferrer. Dimensional hyper-reduction of nonlinear finite element models via empirical cubature. *Computer methods in applied mechanics and engineering*, 313:687–722, 2017.

[36] Philip Holmes, John L Lumley, Gahl Berkooz, and Clarence W Rowley. *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge university press, 2012.

[37] Martin Horák, David Ryckelynck, and Samuel Forest. Hyper-reduction of generalized continua. *Computational Mechanics*, 59(5):753–778, 2017.

[38] IBM. ILOG CPLEX version 12.1.7.0. https://www.ibm.com/products/ilog-cplex-optimization-studio, 2017.

[39] Traian Iliescu and Zhu Wang. Are the snapshot difference quotients needed in the proper orthogonal decomposition? *SIAM Journal on Scientific Computing*, 36(3):A1221–A1250, 2014.

[40] Angelo Iollo, Stéphane Lanteri, and J-A Désidéri. Stability properties of POD–galerkin approximations for the compressible navier–stokes equations. *Theoretical and Computational Fluid Dynamics*, 13(6):377–396, 2000.

[41] Emiliano Iuliano and Domenico Quagliarella. Aerodynamic shape optimization via non-intrusive POD-based surrogate modelling. In *2013 IEEE Congress on Evolutionary Computation*, pages 1467–1474. IEEE, 2013.

[42] M.A. Iwen and B.W. Ong. A distributed and incremental SVD algorithm for agglomerative data analysis on large networks. *SIAM Journal on Matrix Analysis and Applications*, 37(4):1699–1718, 2016.

[43] I. Jaffel, O. Taouali, M. Faouzi Harkat, and H. Messaoud. Moving window KPCA with reduced complexity for nonlinear dynamic process monitoring. *ISA TRANSACTIONS*, 2016.

[44] Steffen Kastian, Dieter Moser, Stefanie Reese, and Lars Grasedyck. Proper orthogonal decomposition (POD) combined with hierarchical tensor approximation (HTA) in the context of uncertain parameters. *arXiv preprint arXiv:1901.10719*, 2019.

[45] Pierre Kerfriden, Pierre Gosselet, Sondipon Adhikari, and Stephane Pierre-Alain Bordas. Bridging proper orthogonal decomposition methods and augmented newton–krylov algorithms: an adaptive model order reduction for highly nonlinear mechanical problems. *Computer Methods in Applied Mechanics and Engineering*, 200(5-8):850–866, 2011.

[46] Pierre Kerfriden, Olivier Goury, Timon Rabczuk, and Stephane Pierre-Alain Bordas. A partitioned model order reduction approach to rationalise computational expenses in nonlinear fracture mechanics. *Computer methods in applied mechanics and engineering*, 256:169–188, 2013.

[47] Bernard O Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences of the United States of America*, 17(5):315, 1931.

[48] DD. Kosambi. Statistics in function space. *Journal of Indian Mathematical Society*, 7:76–88, 1943.

[49] P. Ladevèze, J.-C. Passieux, and D. Néron. The LATIN multiscale computational method and the proper generalized decomposition. *Computer Methods in Applied Mechanics and Engineering*, 199(21-22):1287–1296, 2010.

[50] Yves Le Guennec, J-P Brunet, F-Z Daim, Ming Chau, and Yves Tourbier. A parametric and non-intrusive reduced order model of car crash simulation. *Computer Methods in Applied Mechanics and Engineering*, 338:186–207, 2018.

[51] Yvon Maday, Ngoc Cuong Nguyen, Anthony T Patera, and George SH Pau. A general, multipurpose interpolation procedure: the magic points. 2007.

[52] Anna Madra, Piotr Breitkopf, Balaji Raghavan, and François Trochu. Diffuse manifold learning of the geometry of woven reinforcements in composites. *Comptes Rendus Mécanique*, 346(7):532–538, 2018.

[53] L Meng, P Breitkopf, Balaji Raghavan, G Mauvoisin, O Bartier, and X Hernot. On the study of mystical materials identified by indentation on power law and voce hardening solids. *International Journal of Material Forming*, pages 1–16, 2018.

[54] Liang Meng, Piotr Breitkopf, Guénhaël Le Quilliec, Balaji Raghavan, and Pierre Villon. Nonlinear shape-manifold learning approach: concepts, tools and applications. *Archives of Computational Methods in Engineering*, 25(1):1–21, 2018.

[55] David Modesto, Sergio Zlotnik, and Antonio Huerta. Proper generalized decomposition for parameterized helmholtz problems in heterogeneous and unbounded domains: Application to harbor agitation. *Computer Methods in Applied Mechanics and Engineering*, 295:127–149, 2015.

[56] G. M. Oxberry, T. Kostova-Vassilevska, B. Arrighi, and K. Chand. Limited-memory adaptive snapshot selection for proper orthogonal decomposition. Technical Report LLNL-TR-669265, Lawrence Livermore National Laboratory, 2015.

[57] M. Papadrakakis, M. Lagaros, and Y. Tsompanakis. Structural optimization using evolution strategies and neural networks. *Computer Methods in Applied Mechanics and Engineering*, 156(1-4):309–333, 1998.

[58] Gyung-Jin Park. Technical overview of the equivalent static loads method for non-linear static response structural optimization. *Structural and Multidisciplinary Optimization*, 43(3):319–337, 2011.

[59] P. Phalippou, S. Bouabdallah, P. Breitkopf, and P. Villon. 'on the fly' snapshot selection for hyper-reduced proper orthogonal decomposition with application to nonlinear dynamic. In 6th *European Conference on Computational Mechanics (ECCM6)*, Glasgow, UK, June 2018.

[60] P. Phalippou, P. Breitkopf, S. Bouabdallah, P. Villon, and M. Zarroug. Hyper réduction pour l'optimisation en dynamique rapide des structures. In *NAFEMS 18 France Conference*, Paris, France, November 2018.

[61] P. Phalippou, P. Breitkopf, S. Bouabdallah, P. Villon, and M. Zarroug. Critère temporel pour la sélection des modes POD et application à la dynamique des structures. In 14eme *colloque national en calcul des structures (CSMA 2019)*, France, May 2019.

[62] P. Phalippou, P. Breitkopf, S. Bouabdallah, P. Villon, and M. Zarroug. Hyper-reduced proper orthogonal decompostion for automotive crashworthiness design. In *CONGRÈS: SIA Simulation numérique*, ENSTACA Saint-Quentin en Yvelines, France, April 2019.

[63] P. Phalippou, P. Breitkopf, S. Bouabdallah, P. Villon, and M. Zarroug. Incremental POD and custom integration schemes for hyper-reduced nonlinear dynamics. In *MORTech* 2019 - 5th *international Workshop - Reduced Basis, POD and PGD Model Reduction Techniques*, FIAP Jean Monnet Conference Center - Paris, France, November 2019.

[64] P. Phalippou, P. Breitkopf, S. Bouabdallah, P. Villon, and M. Zarroug. Sparse POD modal subsets for reduced-order nonlinear explicit dynamics. *International Journal for Numerical Methods in Engineering*, 2019.

[65] P. Phalippou, P. Breitkopf, S. Bouabdallah, P. Villon, and M. Zarroug. Sparse POD modes selection for reduced-order nonlinear explicit dynamics. In *ECCOMAS* 5th *Young Investigator Conference*, Krakow, Poland, September 2019.

[66] M.A. Puso, J.S. Chen, E. Zywicz, and W. Elmer. Meshfree and finite element nodal integration methods. *Comput Meth Appl M*, 74(3):416–446, 2008.

[67] Balaji Raghavan and Piotr Breitkopf. Asynchronous evolutionary shape optimization based on high-quality surrogates: application to an air-conditioning duct. *Engineering with Computers*, 29(4):467–476, Oct 2013.

[68] Balaji Raghavan, Piotr Breitkopf, Yves Tourbier, and Pierre Villon. Towards a space reduction approach for efficient structural shape optimization. *Structural and Multidisciplinary Optimization*, 48(5):987–1000, 2013.

[69] Balaji Raghavan, Mohamed Hamdaoui, Manyu Xiao, Piotr Breitkopf, and Pierre Villon. A bi-level meta-modeling approach for structural optimization using modified POD bases and diffuse approximation. *Computers & Structures*, 127:19–28, 2013.

[70] Balaji Raghavan, Manyu Xiao, Piotr Breitkopf, and Pierre Villon. Implicit constraint handling for shape optimisation with POD-morphing. *European Journal of Computational Mechanics*, 21(3-6):325–336, 2012.

[71] IBCM Rocha, FP van der Meer, and LJ Sluys. An adaptive domain-based POD/ECM hyper-reduced modeling framework without offline training. *Computer Methods in Applied Mechanics and Engineering*, 358:112650, 2020.

[72] Clarence W Rowley, Igor Mezić, Shervin Bagheri, Philipp Schlatter, and Dan S Henningson. Spectral analysis of nonlinear flows. *Journal of fluid mechanics*, 641:115–127, 2009.

[73] David Ryckelynck, Florence Vincent, and Sabine Cantournet. Multidimensional a priori hyper-reduction of mechanical models involving internal variables. *Computer Methods in Applied Mechanics and Engineering*, 225:28–43, 2012.

[74] Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.

[75] Lawrence Sirovich. Turbulence and the dynamics of coherent structures. i. coherent structures. *Quarterly of applied mathematics*, 45(3):561–571, 1987.

[76] Yun Teng, Mark Meyer, Tony DeRose, and Theodore Kim. Subspace condensation: full space adaptivity for subspace deformations. *ACM Transactions on Graphics (TOG)*, 34(4):76, 2015.

[77] Paolo Tiso, Rob Dedden, and Daniel Rixen. A modified discrete empirical interpolation method for reducing non-linear structural finite element models. In *ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Two Park Avenue, New York 10016-5990, United States, 2014. American Society of Mechanical Engineers Digital Collection.

[78] Christoph Von Tycowicz, Christian Schulz, Hans-Peter Seidel, and Klaus Hildebrandt. An efficient construction of reduced deformable objects. *ACM Transactions on Graphics (TOG)*, 32(6):213, 2013.

[79] Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015.

[80] D Xiao, F Fang, AG Buchan, CC Pain, IM Navon, and A Muggeridge. Non-intrusive reduced order modelling of the navier–stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 293:522–541, 2015.

[81] Manyu Xiao, Piotr Breitkopf, Rajan Filomeno Coelho, Catherine Knopf-Lenoir, Maryan Sidorkiewicz, and Pierre Villon. Model reduction by CPOD and kriging. *Structural and multidisciplinary optimization*, 41(4):555–574, 2010.

[82] Manyu Xiao, Piotr Breitkopf, Rajan Filomeno Coelho, Pierre Villon, and Weihong Zhang. Proper orthogonal decomposition with high number of linear constraints for aerodynamical shape optimization. *Applied mathematics and computation*, 247:1096–1112, 2014.

[83] Lei Zhou, Jaan-Willem Simon, and Stefanie Reese. Proper orthogonal decomposition for substructures in nonlinear finite element analysis: coupling by means of tied contact. *Archive of Applied Mechanics*, 88(11):1975–2001, 2018.

[84] Sergio Zlotnik, Pedro Díez, David Modesto, and Antonio Huerta. Proper generalized decomposition of a geometrically parametrized heat problem with geophysical applications. *International Journal for Numerical Methods in Engineering*, 103(10):737–758, 2015.